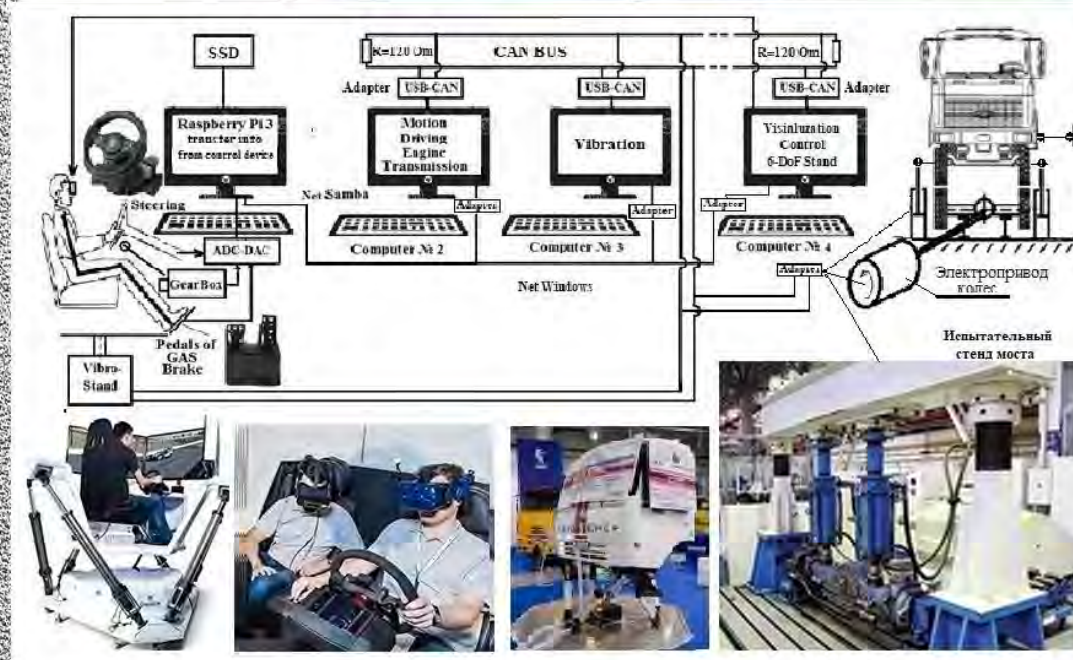


В.Г. Михайлов

МОДЕЛИРОВАНИЕ АВТОМОБИЛЯ СО СМЕШАННОЙ РЕАЛЬНОСТЬЮ С СИМУЛЯТОРОМ



В. Г. МИХАЙЛОВ

**МОДЕЛИРОВАНИЕ АВТОМОБИЛЯ
СО СМЕШАННОЙ РЕАЛЬНОСТЬЮ
С СИМУЛЯТОРОМ**

Минск 2025

УДК 629.33.028(075.8)
ББК 39.33-01я7+32.973

Автор: В.Г. Михайлов

Рецензенты:

Зав. кафедрой “Тракторы” БНТУ, докт. техн. наук, проф. В.П. Бойков;
д.т.н., проф. кафедры “Тракторы и автомобили” БГАТУ, Ю.Д. Карпиевич

Рассматриваются вопросы полунатурного моделирования автомобиля со смешанной реальностью с симулятором для учета фактора водителя, дорожной обстановки, определения его функциональных параметров, оценки долговечности узлов с заданием реальной дороги путем использования более совершенных моделей, пакетов C/C++, Matlab/Simulink. RoadRunner, UnrealEngine, применения микропроцессорных систем, GPS для определения макро и микро профиля дороги.

Предназначена для специалистов конструкторских бюро, научных сотрудников, преподавателей, аспирантов, которые занимаются проектированием, моделированием и разработкой динамических систем, испытаниями колесных машин, роботов. Может быть полезна специалистам других областей, а также в качестве учебного пособия.



Михайлов Владимир Георгиевич, канд. техн. наук
05.05.03, РБ, г. Минск, e-mail sapr7@mail.ru.

Специалист в области автомобилестроения, испытаний подвесок, рам, пневматики, гидравлики, тензометрирования, разработки систем CALS/PLM (PDM, ERP) на Oracle, программ на C/C++, Java, мини-компьютеров (Arduino/Raspberry), микроконтроллеров, моделирования динамических систем в пакете Matlab/Simulink в режиме реального времени с использованием модуля S-Function Builder, смешанной реальности, симуляторов, оценки напряженно-деформированного состояния в пакете ANSYS.

ПЕРЕЧЕНЬ СОКРАЩЕНИЙ И УСЛОВНЫХ ОБОЗНАЧЕНИЙ

ANSYS – программное средство для проведения исследований напряженного состояния твердотельных объектов путем создания конечно элементной модели, динамики газов и жидкостей, динамического соударения объектов.

CAN BUS – сеть для обмена информацией по двум проводам.

GPS – система спутниковой навигации.

MSC.ADAMS – программное средство для моделирования динамических систем транспортных средств и других изделий.

3D пакеты – программные пакеты для создания трехмерной модели 3D-проектирования.

Creo – программное средство для создания 3D-моделей и проектирования в них (пакет высокого уровня).

UNITY – пакет для создания мультимедийных, игровых задач.

Unreal Engine – пакет для создания мультимедийных, игровых задач.

Luforia – пакет для создания мультимедийных, имитационных задач.

S-Function Builder – модуль, обеспечивающий возможность создания пользовательской программы на языке C, связанной с процессом моделирования в MatLab/Simulink.

Raspberry Pi 3/4/400 – 4-ядерный одноплатный мини-компьютер на основе 32-ти битной операционной системы Rasbian.

VR/AR/MR – виртуальная/дополненная/смешанная реальность – компьютерная технология создания искусственного цифрового мира, максимально похожего на реальный. Виртуальная реальность создаёт своё окружение, не взаимодействуя с внешним миром. Дополненная реальность накладывает вспомогательные объекты на наше окружение. Смешанная реальность встраивает виртуальные объекты в окружение и подстраивает их.

КЭМ – конечно-элементная модель, используемая в пакетах инженерного анализа для проведения расчетов напряженно-деформированного состояния.

КМ – Колесные машины.

ПС – программное средство – программа для компьютера.

ПО – программное обеспечение – комплекс программных и технических средств.

ТС – транспортное средство.

Штабная машина – специализированный автомобиль повышенной проходимости с установленным на него модулем управления войсками.

НДС – напряженно-деформированное состояние.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ.....	6
ОБЩАЯ ХАРАКТЕРИСТИКА РАБОТЫ.....	14
ГЛАВА 1 ОБЗОР И АНАЛИЗ РАБОТ ПО МОДЕЛИРОВАНИЮ АВТОМОБИЛЕЙ.....	18
1.1 Обзор работ по теме.....	18
1.2 Обзор применяемых специализированных программных пакетов	21
1.3 Обзор работ по моделирование движения и вибронагруженности машин с применением современных программно-технических средств.....	22
1.4 О моделях движения автомобиля.....	26
1.5 Оценка моделей колебаний автомобиля.....	28
1.6 Обобщение проблем, требующих решения.....	29
ВЫВОДЫ ПО ГЛАВЕ.1.....	30
ГЛАВА 2 ИСТОЧНИКИ ВОЗМУЩАЮЩЕГО ВОЗДЕЙСТВИЯ НА АВТОМОБИЛЬ И МЕТОДЫ ИХ ПОЛУЧЕНИЯ.....	31
2.1 Дорога.....	31
2.2 Получение продольного профиля дороги с помощью GPS.....	37
2.3 Получение макро и микропрофиля дорог другими методами.....	54
2.4 Моделирование воздействия работы двигателя на автомобиль.....	60
2.5 Работа и вибрации восьмицилиндрового V-образного двигателя.....	63
Выводы по главе 2.....	73
ГЛАВА 3 МОДЕЛИ И МЕТОДИКА МОДЕЛИРОВАНИЯ АВТОМОБИЛЯ СО СМЕШАННОЙ РЕАЛЬНОСТЬЮ С СИМУЛЯТОРОМ.....	75
3.1 Модели динамической системы “дорога–автомобиль–водитель”.....	75
3.2 Моделирование упругих элементов в пакете MSC.ADAMS.....	78
3.3 Модель гидравлического амортизатора.....	81
3.4 Модели пневмогидравлического цилиндра подвески.....	85
3.5 Колебательная модель колесной машины.....	90
3.6 Пространственная модель колебательной системы колесной машины..	105
3.7 Модель дизель-генераторной установки и ее влияние на вибрации штабной машины.....	108
3.8 Устойчивость колесных машин.....	113
3.9 Управляемость колесной машины.....	117
3.10 Модель в пакете MSC.ADAMS.....	121
3.11 Модель движения колесной машины по реальной дороге.....	121
3.12 Модель рулевого управления.....	131
3.13 Методика моделирования.....	143
Выводы по главе 3.....	162
ГЛАВА 4 ЭКСПЕРИМЕНТАЛЬНЫЕ ИССЛЕДОВАНИЯ.....	163
4.1 Применение микропроцессорных и мини-компьютерных систем.....	163
4.2 Результаты исследования дорог через ускорения поддресоренных неподдресоренных масс.....	172
4.3 Исследование движения на местных дорогах с помощью GPS.....	178
4.4 Исследование движения в городе и магистральных трассах.....	180

4.5 Исследования элементов систем поддрессоривания.....	184
4.6 Характеристики подвески сиденья.....	192
4.7 Экспериментальное исследование влияния подвески сиденья.....	198
4.8 Исследования поперечной устойчивости штабной машины.....	207
4.9 О лабораторном варианте имитационного моделирования.....	208
ВЫВОДЫ ПО ГЛАВЕ 4.....	211
ГЛАВА 5 МОДЕЛИРОВАНИЕ ДВИЖЕНИЯ, КОЛЕБАНИЙ И	
ДОЛГОВЕЧНОСТИ КОЛЕСНЫХ МАШИН.....	211
5.1 Моделирование движения колесной машины.....	211
5.2 Влияние вибраций на скорость движения.....	222
5.3 Продольные ускорения и результаты моделирования.....	229
5.4 Продольные ускорения от перемещений мостов.....	230
5.5 Определение долговечности узлов автомобиля на стадии проектирования.....	231
5.6 Результаты имитационного моделирования автомобиля на макетном образце с участием водителя в режиме ручной КПП и АКПП.....	242
Выводы по главе 5.....	244
ГЛАВА 6 ПРОГНОЗИРОВАНИЕ ДВИЖЕНИЯ КОЛЕСНЫХ МАШИН...	246
6.1. Задание маршрута и режима движения с помощью карт.....	246
6.2 Оценка эксплуатационных характеристик колесной машины.....	253
Выводы по главе 6.....	259
ГЛАВА 7 МОДЕЛИРОВАНИЕ АВТОМОБИЛЯ СО СМЕШАННОЙ РЕАЛЬНОСТЬЮ С СИМУЛЯТОРОМ.....	260
7.1 Имитационное моделирование автомобиля и рулевого управления....	262
7.2 Выбор реализации имитационного моделирования автомобиля.....	263
7.3 Использование 6-DOF платформы для моделирования в Simulink и испытаний ТС.....	266
7.4 Кинематическая модель платформы Stewart	270
7.5 Моделирование 6-DOF платформы в Simulink.....	273
7.6 Реализация смешанной реальности.....	280
7.7 Реализация визуализации в Simulink (VDBS), Blender, UnrealEngine, RoadRunner.....	309
7.8 Обеспечение быстродействия моделирования.....	317
7.9 Концепция использования смешанной реальности для эргономики....	326
ЗАКЛЮЧЕНИЕ.....	332
БИБЛИОГРАФИЧЕСКИЙ СПИСОК.....	334
ПРИЛОЖЕНИЯ.....	347

ВВЕДЕНИЕ

Сейчас при осуществлении компьютерного имитационного моделирования с задействованием испытательного оборудования (рисунок В.1) начали интенсивно использовать смешанную реальность (MR) с симулятором (рисунок В.2). Это обусловлено необходимостью учета фактора водителя, дорожной обстановки, значительного снижения затрат на создание дорогостоящей перемещающейся рельсовой системы с установленным на нее симулятором, с размещением в нем легкового автомобиля или части кабины, а также специального огромного помещения с его сложной инфраструктурой.



Рисунок В.1. – Имитационные испытания автомобилей за рубежом



Рисунок В.2. – Использование фирмами BMW (а), Volvo (б) смешанной реальности

Современный симулятор (рисунок В.1, справа, рисунок В.2) со смешанной реальностью представляет собой имитатор рабочего места водителя (сиденье, руль, педали, органы управления), установленный на шестиосный (6-DoF) вибростенд, управляемый с компьютеров, на которых осуществляется моделирование движения, колебаний, управляемости автомобиля, визуализация дорожной обстановки, реализуемая в шлеме виртуальной реальности у водителя. Шлем представляет собой внутренний монитор 4K/5K в виде очков, датчики перемещений и связан по сети либо WF с компьютерной техникой. Изображение объекта в шлеме меняется в соответствии перемещением автомобиля и поворотами человека благодаря

датчикам на шлеме, руке и связи маркеров с 3D моделью. Вместо шлема могут использоваться видеопанели, проекторы, мониторы, но с меньшим эффектом.

Смешанная реальность (MR) представляет собой компьютерную технологию создания виртуального мира эксплуатации техники (сочетания виртуального смешанного с реальным) с задействованием и погружением в него человека с его имитационным управлением машиной на основе программно-технического комплекса, включая иные способы визуализации, иллюзии движения по дороге. Осуществляется путем использования шлема VR или видеоз экранов/панелей и “искусственной” руки с перчаткой, их позиционирования, виртуальной визуализации внешней и внутренней обстановки машины, вибрационного и звукового воздействия на симуляторе в сочетании с моделированием автомобиля.

Смешанная реальность применяется для создания у испытуемого ощущения движения по реальной дороге с учетом обстановки на ней, управления им ТС на основе зрительной иллюзии, вибрационного и звукового воздействия, влияющие на режим движения и нагруженность автомобиля.

Использование MR позволяет проводить исследования автомобиля с учетом фактора водителя, дорожной обстановки, обзорности, эргономики рабочего места водителя, пассажира и прогнозировать поведение человека за рулем, а также обсуждать и согласовывать проект в электронном виде, когда он еще не воплощен в металле, включая работу дилеров с клиентами по сети.

Путем сочетания методов имитационного полунатурного моделирования (ИМ) на симуляторе со смешанной реальностью (с водителем, шлемом виртуальной реальности VR) [2–4] можно получить в лабораторных условиях более точные данные параметров, нагруженности машины, необходимые для расчетов ее долговечности, проработать на моделях большое количество дорожных ситуаций и вариантов конструкции с имитированием реальных условий эксплуатации с последующим их испытанием в стендовых условиях, отработать эргономику, значительно сократить время создания новых моделей машин, перенеся на них основную тяжесть работ по созданию техники.

Различается реализация смешанной реальности для обучающих тренажеров и исследовательских симуляторов.

Первые создаются на основе мультимедийных пакетов UnrealEngine (UE), Unity, их стандартных компонентов с упрощенной имитацией движения путем визуального изменения генерируемой внешней дорожной обстановки (зрительный эффект от движения соседнего поезда, создающий иллюзию движения стоящего) и задаваемого управления ТС. Акцент в них направлен на визуализацию обстановки на дороге, а не на вибрации. Тренажеры создаются под конкретную машину и включают в себя ее кабину и органы ее управления.

Примерами отечественных разработок являются обучающие тренажеры вождения БелаЗ, КамаЗ, трактора Беларус-1523 (рисунок В.3), использующие упрощенные модели, простые решения, не требующие точности. Для их реализации в пакетах UnrealEngine, Unity достаточно одного компьютера.



Рисунок В.3. – Тренажеры Камаз (а), трактора Беларус-1523 (б), сделанные в РФ

На них можно имитировать условия эксплуатации, движение ТС с имитацией управления, достаточные для обучения. Однако процесс создания MR довольно сложен, трудоемок, включающий разработку 3D модели, размещения ее на сцене в пакетах, задания окружения дороги, их параметров, связей и программирование. На освоение UnrealEngine с применением C/C++ потребуются потратить два года.

Для простых автотренажеров обычно используются механические виброплатформы (рисунок В.4), на которых невозможно воспроизвести реальный спектр вибрации (в основном наклоны платформы) или без них.

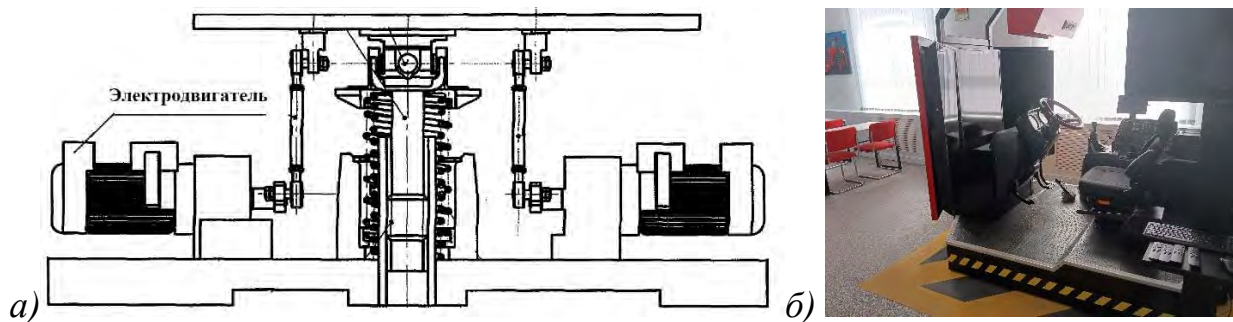


Рисунок В.4 – Механическая виброплатформа (а), тренажер трактора в БГАТУ (б)

На симуляторах в отличие от тренажеров используются 6-DoF вибростенды, более совершенные модели и решения, нацеленные на сложные

задачи исследований, доводки автомобилей путем одновременного моделирования подсистем автомобиля (рисунок В.5) с учетом их связей, что требует специальных подходов и больших ресурсов компьютеров.



Рисунок В.5. – Структурная схема связей основных подсистем

Реализовать исследовательский симулятор на одном компьютере с одним из пакетов (Simulink, UnrealEngine, Unity) не получается, решать эти задачи необходимо комбинированием различных пакетов и подходов. Хотя в UnrealEngine задается масса автомобиля, кривая крутящего момента двигателя, передаточное число трансмиссии, радиус колеса этого все же недостаточно для полноценного моделирования автомобиля. Также невозможно корректно оценить топливную экономичность, вибронагруженность автомобиля. Все нацелено на внешние визуальные эффекты для игр, включая наклоны, опрокидывание автомобиля через параметры подвески. Получить данные по нагруженности автомобиля на тренажерах с UE невозможно т.к. нельзя задать макро и микропрофиль дороги.

Поэтому UnrealEngine, Unity больше подходят для создания тренажеров для обучения водителей, игровых целей. Для проведения исследовательских работ требуется учитывать все, реализовывать более высокую точность моделирования, использовать иные решения включая в т.ч. указанные пакеты.

В некоторых случаях возможно использование тренажеров с 6-DoF платформой в качестве симуляторов, но для этого потребуется переделка и создание специального программного обеспечения.

Моделирование автомобиля с MR и симулятором следует рассматривать в совокупности основных взаимосвязанных ключевых подсистем

(рисунок В.5) с заданием совместного массива макро и микропрофиля дороги синхронно с визуализацией обстановки в MR, воздействием вибростенда на водителя при осуществлении задач:

- движения и оценки топливной экономичности автомобиля;
- вибронегруженности;
- колебаний в трансмиссии;
- управляемости автомобиля.

Возможен вариант реализации смешанной реальности на макетном образце без вибростенда (рисунок В.6), но он не дает такого эффекта как со стендом. Основное его преимущество – упрощение реализации, сокращение числа компьютеров.



Рисунок В.6. – Схема имитационного моделирования (а), макетный настольный образец (б) и использованные в работе модели (в, г)

Следует отметить, что при использовании смешанной реальности с игровыми, обучающими симуляторами на основе пакетов UnrealEngine, Unity классического моделирования колебаний нет, они заменяются наклонами, перемещениями испытуемого объекта для создания иллюзии колебаний.

Преимуществом симуляторов в отличие от тренажеров является возможность варьирования 3D модели автомобиля без физического изменения рабочего места водителя или самой машины т.е. они более универсальны.

Спектр решаемых задач при исследовательском моделировании шире, позволяющий охватить всю цепочку создания автомобиля, включая **проведение полунатурных испытаний**. На заводах обычно ограничиваются минимальным объемом стендовых, лабораторно-дорожных, пробеговых испытаний колесной техники, а также ресурсными испытаниями узлов на основе блоков нагружения. Но этого бывает недостаточно, чтобы выявить причины конструктивных недоработок. Так пока на МАЗе не провели стендовые испытания не могли понять причину выхода подшипников промежуточного вала среднего моста, дребезжания сиденья, выхода лампочек

фар из строя. Сейчас проще это решить путем проведения имитационных испытаний на 6-DoF симуляторе с задействованием смешанной реальности с воспроизведением реальных эксплуатационных нагрузок на основе моделирования движения и колебаний автомобиля (рисунок В.7) в сочетании со стендовыми ресурсными испытаниями, позволяющими сократить сроки доводки машины, что является новизной предлагаемого решения.

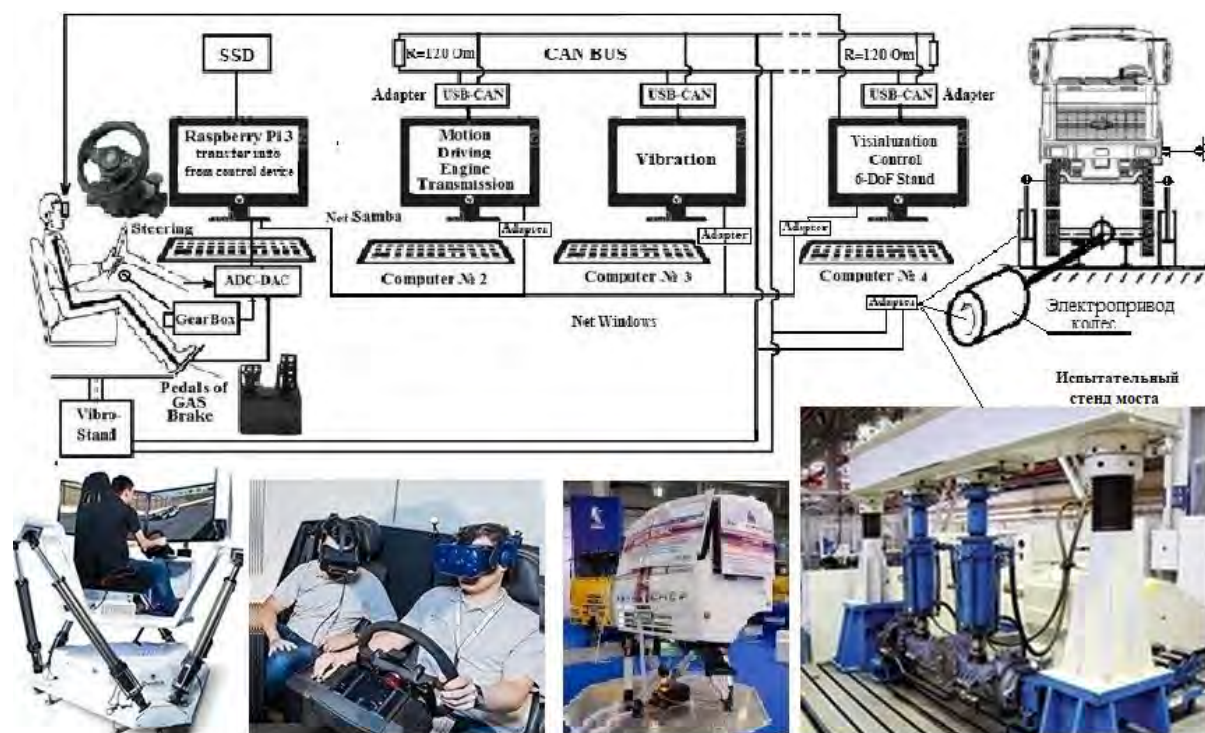


Рисунок В.7. – Предлагаемая схема моделирования автомобиля со смешанной реальностью с симулятором в сочетании с испытательными стендами

Важным в таком моделировании являются более полное имитирование условий эксплуатации, включая визуализацию дорожной обстановки, режим движения, вибрации, звук на основе использования реального макро и микропрофиля дороги, более совершенных моделей, обеспечивающих итоговую погрешность не более 10–15 % по спектру на реальной дороге против 100% и более при использовании простых моделей, линеаризации на игровых, обучающих тренажерах.

Создание исследовательских симуляторов обуславливает проведение экспериментального и теоретического исследования всех этих элементов. Особое внимание следует обратить на применение более корректных моделей упругих и демпфирующих элементов, включая трение в подвеске, задание реального возмущения дороги, определяющих точность моделирования.

Необходимо решить вопрос синхронизации визуализации дорожной обстановки с его реальным перемещением в процессе движения автомобиля.

Поэтому в данной работе акцентировано внимание на создание исследовательского симулятора, хотя многое применимо и к тренажерам.

Имитационное моделирование с задействованием смешанной реальности и симулятора требует реализации большого комплекса решений:

- создания и использования более совершенных моделей;
- моделирования движения, колебаний и управляемости автомобиля;
- имитирования и визуализации дорожной обстановки синхронно с воздействием дороги (макро и микропрофиля) на водителя и автомобиль;
- конструктивного исполнения самой системы;
- разработки программно-технического обеспечения моделирования в режиме реального времени, получение через АЦП информации с органов управления автомобиля и передача ее по сети на другие компьютеры, включая обмен информацией между ними и исполнительными устройствами вибростенда, практической реализации всего этого;

Новизной предлагаемых решений является комбинированное использование более совершенных моделей, различных сетей, разделения вычислений по компьютерам для обеспечения необходимого быстродействия, осуществление моделирования движения, колебаний, управляемости автомобиля, управления симулятором в пакете Simulink, а для визуализации дорожной обстановки применение C/C++, RoadRunner, UnrealEngine на основе смешанной реальности для получения более корректных результатов, позволяющих отработать конструкцию автомобиля на электронных моделях, что невозможно при использовании одних пакетов UnrealEngine или Unity. Предлагаемое решение такого моделирования позволяет варьировать:

- внутренний дизайн машины путем изменения ее 3D модели;
- различные категории дорог, их макро и микропрофили;
- дорожную обстановку;
- ограничения на дорогах (различные ситуации, пробки и т.п.);
- использование различных характеристик подвесок, шин;
- характер вождения водителей;
- различные климатические условия и др.

Полученные в результате такого моделирования данные могут использоваться, как для непосредственного, так и для проведения ускоренных стендовых испытаний узлов и деталей автомобиля и отладки ее систем.

Однако такая реализация представляет сложную задачу и пока в СНГ не изучена.

Без разработки основополагающих компонентов системы это не решить, в т.ч. найти решение обеспечения быстродействия моделирования этих трех задач в реальном времени ввиду ограниченных возможностей существующих

компьютеров при использовании более корректных моделей и возмущений. Поэтому в работе уделено большое внимание рассмотрению этих вопросов.

Для имитационных испытаний с MR с симулятором целесообразно использование 6-DoF платформы на базе электромеханических актуаторов (рисунок В.8), представляющих механизм преобразования вращательного движения в поступательное с помощью электродвигателей с электронными блоками управления, обеспечивающих реальный спектр вибраций.

E12 Series:MB-E-6DOF/12/500KG

PAYLOAD WEIGHT 500 kg / 1,02 lb		
Specifications	Heave (max): ± 0.19 m (± 7.5 in)	Degrees of Freedom: Six
	Acceleration	Velocity
	Surge ± 6.0 m/s ² (± 0.61 g)	± 0.60 m/s (± 23.6 in/s)
	Sway ± 6.0 m/s ² (± 0.61 g)	± 0.60 m/s (± 23.6 in/s)
	Heave ± 8.0 m/s ² , ± 8.2 g	± 0.50 m/s (± 19.7 in/s)
	Roll $\pm 300^\circ$ /s ²	$\pm 40.0^\circ$ /s
	Pitch $\pm 300^\circ$ /s ²	$\pm 50.0^\circ$ /s
Yaw $\pm 500^\circ$ /s ²	$\pm 50.0^\circ$ /s	

E12 Series:MB-E-6DOF/12/1500KG

PAYLOAD WEIGHT 1,500 kg / 3,307 lb		
Specifications	Heave (max): ± 0.19 m (± 7.5 in)	Degrees of Freedom: Six
	Acceleration	Velocity
	Surge ± 6.0 m/s ² (± 0.61 g)	± 0.60 m/s (± 23.6 in/s)
	Sway ± 6.0 m/s ² (± 0.61 g)	± 0.60 m/s (± 23.6 in/s)
	Heave ± 8.0 m/s ² (± 0.82 g)	± 0.50 m/s (± 19.7 in/s)
	Roll $\pm 300^\circ$ /s ²	$\pm 40.0^\circ$ /s
	Pitch $\pm 300^\circ$ /s ²	$\pm 50.0^\circ$ /s
Yaw $\pm 500^\circ$ /s ²	$\pm 50.0^\circ$ /s	

a)

b)

a)

б)

Рисунок В.8. – Конструкция 6-DoF платформы на электроактуаторах Moog (а), ее электрический цилиндр (б)

Но для их использования требуется разработать математическое описание кинематики 6-DoF платформы, программы преобразования сигналов для электроактуаторов для передачи информации на нее по сети CAN BUS.

Принимая во внимание, что в СНГ вопросами такого моделирования для исследовательских целей на качественно новом уровне только сейчас серьезно начинают заниматься (нет высококвалифицированных кадров), да и сами методы компьютерного моделирования, требуют совершенствования. Многие вопросы получения макро и микропрофилей дорог, необходимые для моделирования, использования совершенных моделей и других аспектов, реализации новых методов являются актуальными для нас и требующих своего разрешения поскольку зарубежные фирмы не желают раскрывать свои секреты. Несмотря на развитие Matlab/Simulink, задействование новых пакетов (RoadRunner, UnrealEngine) для визуализации, многие вопросы их совместного использования еще недостаточно проработаны и решены. Все это свидетельствует об актуальности данного направления работы. Без этого невозможно дальнейшее развитие отечественного авто и тракторостроения.

Решению этих вопросов посвящена данная монография.

ОБЩАЯ ХАРАКТЕРИСТИКА РАБОТЫ

Связь работы с крупными научными программами (проектами) и темами. Монография представляет обобщение научных работ автора, накопленных знаний, опыта на МАЗе, МЗКТ, Мидивисаны при выполнении различных государственных и отраслевых программ и вне их.

Цель и задачи исследования. Целью является разработка научных основ и методология моделирования системы “дорога–автомобиль–водитель” на качественном новом уровне для решения задач движения, колебаний, управляемости, нагруженности, долговечности, топливной экономичности автомобиля с учетом воздействия вибраций и дороги на водителя путем имитационного моделирования в RealTime с использованием смешанной реальности с симулятором.

Для достижения указанной цели сформулированы следующие **задачи**:

1. Создать более совершенные модели совместного движения и колебаний автомобиля для оценки на стадии проектирования их тягово-скоростных свойств, экономичности, плавности хода, нагруженности узлов и реализовать моделирование на симуляторах, включая смешанную реальность.

2. Разработать новые методы визуализации дороги, получения макропрофилей дорог, режимов движения с помощью GPS, микропрофилей через ускорения неподрессоренных и поддрессоренных масс, нагрузок на мост, необходимые для моделирования движения и колебаний автомобилей.

3. Провести комплексные исследования тягово-скоростных свойств, затрат топлива на примере колесных машин специального назначения на различных дорогах с учетом воздействия вибраций на водителя;

4. Выявить источник продольных и поперечных ускорений автомобиля и предложить способ их реализации при моделировании.

5. Разработать рекомендации по выбору основных параметров, прогнозированию передвижения автомобилей по различным дорогам на примере специальных машин на основе полученных данных, спутниковых карт местности, снимков с учетом на них дорожной обстановки.

Научная новизна. Научная новизна работы заключается в:

– в создании научных основ и методологии моделирования системы “дорога–автомобиль–водитель” на качественно новом уровне для решения задач движения, колебаний, управляемости, нагруженности, долговечности, топливной экономичности автомобиля с учетом воздействия вибраций дороги, ее обстановки на водителя на основе полунатурного моделирования со смешанной реальностью с симулятором и решениях по их реализации.

- разработке новых методов получения и использования совместных массивов продольных профилей дорог на основе GPS, микропрофиля по ускорениям неподрессоренных и поддрессоренных масс и нагрузкам на мост в процессе движения автомобиля, характеристик восьми типов автомобильных дорог, осуществление оценки, критериев ровности, качества этих дорог;

- создании более совершенных моделей (трения, амортизаторов, рессор, увода шины) и методика совместного моделирования автомобиля на их основе с расхождением не более 10–15 % с учетом спектра ТС;

- использовании модуля Simulink S-Function Builder и разработанной в нем программы на C/C++ для повышения быстродействия за счет занесения в ОП совместного массива дороги (96000 точек) и реализации через нее управления процессом моделирования, логикой переключения КПП на основе баланса сил, ускорения ТС, точек снижения скорости, передачу звука, информации на стенды, что дало возможность осуществить режим Real Time;

- разработке метода оценки долговечности узлов колесных машин через цепочку этапов: моделирования движения и колебаний автомобиля, получения по ним нагрузок, напряжений в узлах через КЭМ модель, далее расчет долговечности по усталостным характеристикам материалов и систематизации данных напряжений методом дождя, позволившие оценить долговечность картера заднего моста МАЗ в 1275 тыс. км;

- предложенной оригинальной схеме с распределением вычислений по компьютерам и методе полунатурных испытаний автомобиля (включая рулевое управление) с использованием стенда с беговыми барабанами на основе задних мостов с шинами, скорость вращения которых управляется по результатам моделирования на компьютерах;

- методе реализации моделирования более сложных моделей в т.ч. пространственных (трудно реализуемых в режиме реального времени) на основе моделирования вначале на плоских моделях с последующим использованием полученных записей для пространственных моделей, позволяющих проводить оценку напряженно-деформированного состояния рам, кабин, мостов и других узлов, испытывающих крутящие нагрузки;

- разработке нового более простого метода визуализации MR на основе видеосъемки дорожной обстановки при фиксированной скорости движения автомобиля ($v = 20$ м/с) с последующим ее воспроизведением в программе с управляемой частотой просмотра синхронно с процессом в Simulink;

- комбинированном использовании пакетов: для моделирования движения, колебаний, управляемости автомобиля – Simulink, а для визуализации обстановки MR – RoadRunner, UnrealEngine и их интеграции;

- предложенной схеме и модели работы дизеля V8 (от сил сгорания топлива) и его реализации в Simulink как источника воздействия на

трансмиссию, ТС, объясняющая его продольные и поперечные вибрации;

- в полученных графиках для простого прогнозирования передвижения колесных машин на различных дорогах в чрезвычайных ситуациях на основе моделирования движения в этих условиях;

- проведении на основе предложенных моделей и подходов со смешанной реальности с симулятором всестороннего исследования движения, колебаний, управляемости и экономичности колесной машины.

Объект исследования – колесные машины, включая специальные на базе автомобильных шасси повышенной проходимости.

Предмет исследования — научные основы и методология моделирования движения, управляемости, колебаний, оценки долговечности, нагруженности автомобиля, параметров движения, топливной экономичности, колебаний на различных дорогах с учетом фактора воздействия вибраций на водителя на основе новых подходов получения и использования реальных микро и макропрофилей дорог, режимов движения, полученных с помощью GPS, реализация имитационного моделирования со смешанной реальности.

Положения, рассматриваемые в монографии:

- научные основы и методология моделирования системы “дорога–автомобиль–водитель” на качественно новом уровне для решения задач движения, колебаний, управляемости, нагруженности, долговечности, топливной экономичности автомобиля, реализация моделирования со смешанной реальности с симулятором и решения по их реализации;

- методы получения продольных профилей дорог на основе данных GPS, микропрофиля по ускорениям неподдрессоренных и поддрессоренных масс, нагрузкам на мост в процессе движения автомобиля и создании на их основе объединенных массивов восьми типов автомобильных дорог, необходимых для моделирования автомобиля. Разработка предложений по оценке и критериям ровности и качества дорог;

- создание более совершенных моделей совместного движения и колебаний автомобиля в Matlab/Simulink с расхождением 10–15 % при использовании разработанной программы на языке С в модуле S-Function Builder и задании реального макро и микропрофиля различных дорог для оценки их тягово-скоростных свойств, экономичности, плавности хода, нагрузок колесных машин;

- результаты комплексного исследования движения, колебаний и топливной экономичности колесной машины на основе предложенных моделей, методов, подходов, полученных массивов дорог;

- оценка виброн нагруженности колесных машин и допустимого времени работы в экстремальных ситуациях согласно ИСО 2631-5 2018;

- выявление источника продольных ускорений и способа их реализации;

- предложения по совершенствованию подвески колесной машины, выбору ее оптимальных параметров.

- разработка метода прогнозирования движения автомобиля на незнакомых участках дорог на основе карт и информации с интернета при моделировании и оценки машины без симулятора;

- анализе кинематики 6-DoF платформы, создание ее модели, определении необходимых усилия электроактуаторов (20280 Н) и выборе подходящих на рынке платформ (MOOG/Parker)

- реализация имитационного моделирования со смешанной реальностью в т.ч. в комбинации Simulink с RoadRunner, UnrealEngine, звуком в шлеме VR(MR) с симулятором при создании автомобиля.

Вклад и участие автора. Данная работа выполнена лично Михайловым В.Г. и обобщает опыт его работы на МАЗе, МЗКТ, ООО “Мидивисана” и вне их.

Апробация результатов работы и ее использование. Основные результаты работы публиковались, докладывались автором, обсуждались на национальных и международных научно-технических конгрессах, конференциях ОИМ НАН РБ и конференции по военно-техническому сотрудничеству России и Республики Беларуси, в научных журналах БНТУ, МАДИ, НАМИ, НГТУ. Использование результатов приведено в заключении.

Опубликованность результатов работы. Основные результаты опубликованы в 52 работах (12,3 авторских листа) в изданиях, включенных в Перечень научных изданий ВАК Республики Беларусь для опубликования результатов научных исследований и в четырех монографиях.

Структура и объем работы. Работа состоит из введения, общей характеристики работы, семи глав, посвященных анализу выполненных работ, источникам воздействия на систему “дорога-автомобиль-водитель”, получению макро и микропрофиля дорог, разработке моделей этой системы, экспериментальным исследованиям элементов систем поддрессирования и их влиянию, поперечной устойчивости колесной машины, моделированию движения, вибраций, влиянию их на скорость движения, выявлению причин продольных и поперечных ускорений, определению долговечности узлов автомобиля на стадии проектирования, прогнозированию движения колесных машин, реализации имитационного моделирования и смешанной реальности, заключению, библиографического списка и одиннадцати приложений. Полный объем монографии – 506 страниц, 207 рисунков, 32 таблицы, объем текста – 268 стр. (17,7 а.л.). Количество использованных библиографических источников – 143.

ГЛАВА 1

ОБЗОР И АНАЛИЗ РАБОТ ПО МОДЕЛИРОВАНИЮ АВТОМОБИЛЕЙ

1.1 Обзор работ по теме

Исследованиями динамики, устойчивости транспортных средств занимались Д.А. Антонов, В.Б. Альгин, В.Л. Афанасьев, С. В. Бахмутов, Е. В. Балакина, В.П. Бойков, В.С. Васильев, Л.Л. Гинцбург, Г.В. Гольдин, А.И. Гришкевич, В.В. Гуськов, Б.М. Додонов, В. П. Жигарев, А. С. Литвинов, М.В. Келдыш, В.И. Кольцов, Я.М. Певзнер, Е.А. А.Т. Скойбеда, В.С. Чудаков, Б.С. Фалькевич, А.А. Хачатуров, В.П. Жигарев, Е.И. Яковлев, и др. Благодаря им заложены основы теории движения, устойчивости и расчета автомобилей. Основным инструментом расчетов тогда (1960–1980 гг.) являлось моделирование на аналоговых и цифровых вычислительных машин (АВМ и ЦВМ), что сдерживало применение сложных пространственных моделей машин. По плавности хода грузовых автомобилей, исследованию упругих и демпфирующих характеристик подвесок также имеется много работ, выполненных в 1970-1980 гг. А.И. Гришкевичем, Ю.Ю. Беленьким, А.М. Мариничем, А.И. Петрович, Р.И. Фурунжиевым, А.Д. Дербаремдикером, Ф.Д. Коневым, Я.М. Певзнером, Г.Г. Гридасовым, А.Е. Плетневым, И.Г. Пархиловским, В.И. Кольцовым и др. с использованием плоских моделей.

Однако в этих работах и других открытых источниках нет данных по вибрациям и нагрузкам, действующим на специальные автомобили и кузов-контейнер штабной машины в реальных условиях эксплуатации. В настоящее время в связи насыщением кузовов-контейнеров вычислительной и электронной аппаратурой вопрос ее надежного функционирования из-за вибраций и связанный с этим вопрос прочности контейнера становится все более актуальным. На данном этапе развития компьютерной техники возможно в большей степени на стадии проектирования решать эти вопросы на основе моделирования движения и колебаний автомобиля и на их основе оценивать напряженно-деформированное состояние конструкции и проводить оценку ее долговечности, перенося основную тяжесть на них и лабораторно-имитационные испытания, резко сокращая дорожные экспериментальные исследования в различных условиях эксплуатации.

Необходимо отметить, что после развала СССР значительно снизился интерес к этой тематике ввиду отсутствия финансирования. Произошла смена поколения людей, которые не знают, что было сделано до них и они часто повторяют ошибочные пути прошлого.

Остановимся на ключевых моментах: движения и колебаний автомобиля.

Колебательные модели автомобилей. До конца 1990-х годов, да и сейчас, при расчетах в основном используют упрощенную плоскую модель автомобиля (рисунок 1.2а) с простой моделью “вязкого” трения в подвеске, которая не позволяет рассчитывать нагрузки, связанные с кручением.

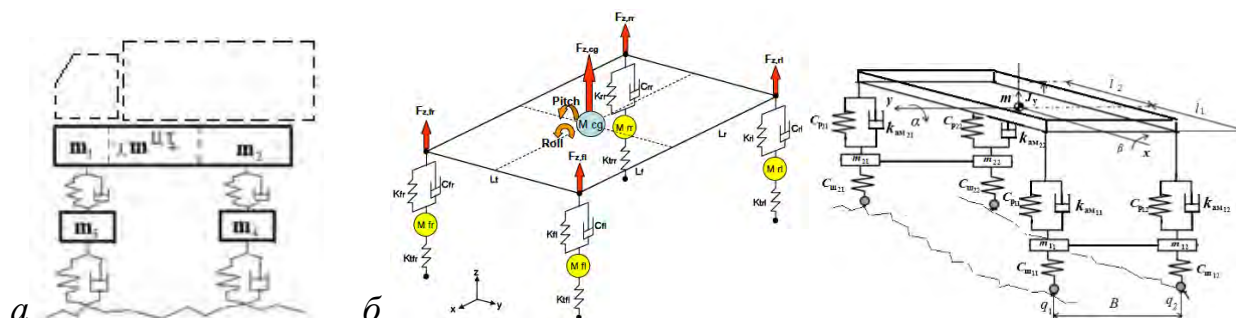


Рисунок 1.2. – Плоская (а) и пространственные модели автомобиля (б)

Расчеты плавности хода сводились в основном к оценке вертикальных вибраций при переезде препятствий. Также делались расчеты при использовании случайного возмущения с генераторов белого шума с оценкой вибраций по общей полосе частот 0–22,4 Гц. Использовались либо линейные модели без трения или с “вязкой” моделью, которые не отражали реальной картины. Сейчас в СНГ из-за отсутствия данных по реальным характеристикам дорог (макро и микропрофиля), характеристик подвески исследования по плавности хода вообще перестали проводиться. В то же время за рубежом такие исследования на ведущих фирмах ведутся, но информации в открытых источниках об этом нет. Косвенно об этом свидетельствуют установленные ISO 2631 жесткие нормы на вибрации [16].

Пространственные модели использовались крайне редко, при этом подрессоренную массу рассматривали как твердое тело без учета податливостей рамы, кузова (рисунок 1.2б).

Это связано с ограниченными возможностями в то время вычислительной техники (АВМ, ЦВМ). В настоящее время для расчета прочности каркасных конструкций, поперечно-угловых вибраций возможно и целесообразно использовать пространственную модель, так как наибольшие напряжения возникают при кручении, учитывать податливости рамы и контейнера, а также тип подвески. Но для этого необходимо располагать данными по дорогам и основным элементам динамической системы.

Следует упомянуть работу [17], в которой еще в 1981 г. рассмотрены динамические схемы отдельных агрегатов автомобиля: подвески, фрикционного сцепления, редукторного узла, гидротрансформатора и движителя и приведены дифференциальные уравнения, описывающие функционирование этих агрегатов (рисунок 1.3). Но до реализации движения и колебаний автомобиля тогда дело не дошло, но их схемы актуальны и сейчас.

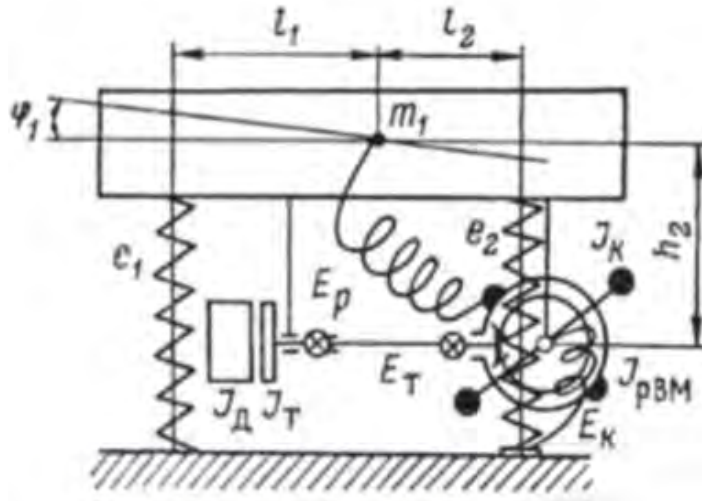


Рисунок 1.3. – Совместная динамическая модель колебаний и трансмиссии

Интересная динамическая схема трансмиссии предложена в работе [18] для моделирования переходных процессов при неустановившихся режимах (рисунок 1.4). Но в ней не хватает модели двигателя с учетом динамического сгорания топлива, обуславливающее неравномерное вращение.

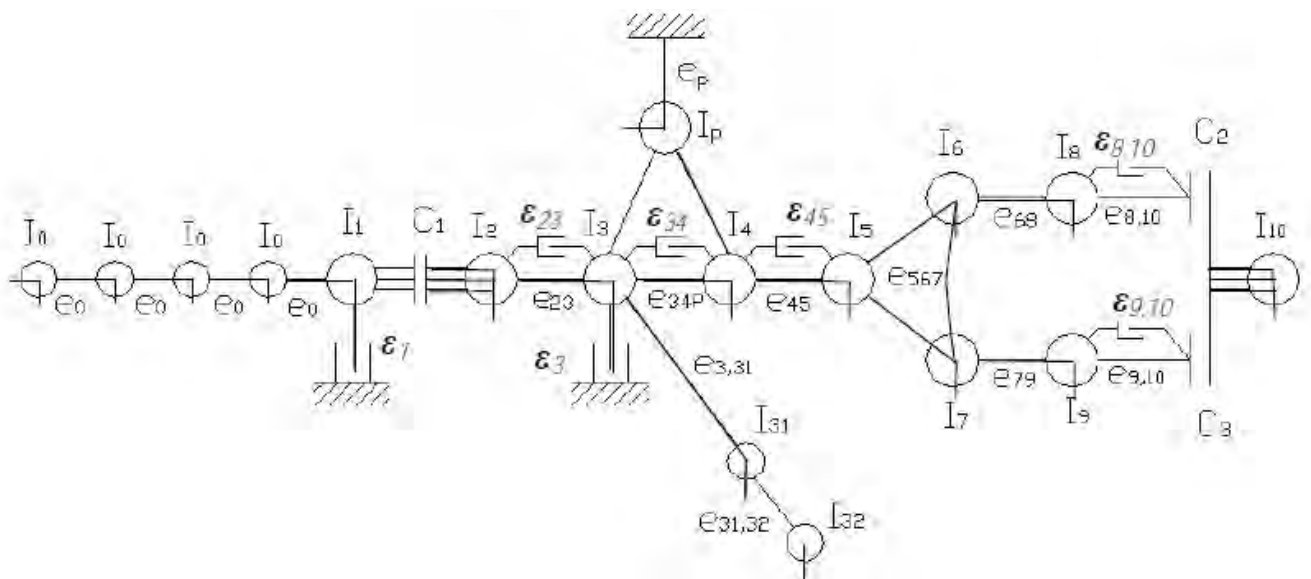


Рисунок 1.4. Схема приведенной динамической системы машинного агрегата автомобиля с колесной формулой 4 × 2

1.2 Обзор применяемых специализированных программных пакетов

Для динамических расчетов и кинематического анализа механических систем в большинстве организаций СНГ (ГАЗ, МАЗ, ОИМ и др.) применяются специализированные программные пакеты. Одним из них является программный комплекс MSC.ADAMS, имеющий библиотеку с широким набором типовых компонентов, которые используются для построения сложной модели механического объекта [19–21, 24–26]. После формирования такой модели компьютерная математическая модель, необходимая для кинематического или динамического расчета объекта, формируется автоматически. Пакет MSC.ADAMS имеет возможность учета податливостей компонентов исследуемого изделия. Для этого упругие характеристики его частей определяются в пакетах (например, MSC.NASTRAN), а затем в специальном формате передаются в MSC.ADAMS и включаются в виртуальную модель. Полученные в MSC.ADAMS усилия, действующие на элементы системы, затем могут быть переданы в конечно-элементную систему и использованы в качестве исходных данных для определения напряженного состояния деталей. Такая технология обеспечивает более точное моделирование динамических систем. Недостатками MSC.ADAMS – использование упрощенных моделей трения и невозможность задать требуемые объемы массивов реальных профилей и микропрофилей дорог по точкам, что ограничивает сферу решаемых задач по динамике автомобиля.

Альтернативным направлением является использование пакета Matlab/Simulink, который широко применяется для моделирования движения, колебаний, устойчивости автомобилей. Последний требует написания дифференциальных уравнений динамической системы. Далее на их основе в пакете составляются графические блок-схемы интеграторов, сумматоров, нелинейных элементов, соответствующие динамической системе, после чего задача решается численным методом на компьютере. Возмущение дороги обычно задается с помощью генераторов белого шума и фильтров или синусоидальных сигналов.

С помощью этого пакета в ОИМ НАН РБ проводились моделирование колебаний автобуса [19] на синусоидальных неровностях и исследования его динамики движения и торможения [24, 25]. ОАО “Мидивисана” использовала Matlab/Simulink для оценки вибронагруженности и плавности хода штабной машины с применением реального микропрофиля дороги [26]. Пакет Matlab/Simulink больше подходит для значительно большего круга задач, а также проведения имитационных исследований, когда надо сопоставить или получить определенные характеристики модели. Например, дважды проинтегрировав сигнал ускорений в Simulink, можно получить

сигнал виброперемещений, который затем воспроизвести на вибростенде и исследовать поведение реального объекта в стендовых условиях. Также на Simulink путем сложной обработки сигналов усилий и ускорений на оси автомобиля можно получить продольный и микропрофиль дороги для моделирования ТС.

1.3 Обзор работ по моделирование движения и вибронагруженности машин с применением современных программно-технических средств

1.3.1 Моделирование движения и вибронагруженности машин

Начиная примерно с 1998 г., с внедрением пакета инженерного анализа MSC.ADAMS на персональных компьютерах появился ряд интересных научных работ по его использованию, посвященных исследованиям колебаний, устойчивости, рулевого управления, определению нагрузок в раме, кузове легкового и грузового автомобиля [19–21, 24].

Представляют интерес работы по моделированию движения автомобиля по дороге, поперечной и продольной устойчивости сочлененного самосвала большой грузоподъемности [25], выполненные В.Б. Альгиным, С.В. Харитончиком, А.Г. Выгонным, А.Н. Колесникович, Р.Э. Шейбак, А.В. Калинин в Объединенном институте машиностроения (ОИМ) НАН Беларуси в пакете MSC.ADAMS.

Подходы по оценке параметров вибронагруженности колесных машин. При создании автомобилей, как и штабных машин, необходимо располагать данными о действующих нагрузках как в дорожных условиях, так и в других. Их можно получить как экспериментально, так и расчетным путем. В работах [26, 27] рассматривается получение возникающего воздействия от дороги на несущие системы конструкций мобильных транспортных средств путем моделирования микропрофилей дорог и колебаний автомобиля.

В работе [27] приведены таблицы, в которой указаны технические скорости движения в дорожных условиях нормальной эксплуатации (таблица 1.1).

Допустимые значения скорости в зависимости от типа покрытия, оцениваемые по толчкомеру, приведены в таблице 1.2.

Таблица 1.1. – технические скорости движения в типичных дорожных условиях нормальной эксплуатации

Группы грузовых автомобилей	Техническая скорость, км/ч			
	По дорогам с ровным усовершенствованным покрытием		По дорогам с щебеночным, булыжным или гравий. покрытием	По дорогам в сухом состоянии и по горным дорогам
	Интенсивное	Неинтенсивное		
Малой грузоподъемности	45–50	60–70	20–35	25–35
Одиночные, грузоподъемностью до 8 т	38–42	50–60	25–35	20–30
Автопоезда всех видов и одиночные, грузоподъемностью свыше 8 т	28–32	40–50	20–30	15–25

Таблица 1.2. – Допустимые значения скорости в зависимости от типа покрытия

Тип покрытия	Показания толчкометра, S , см/км	Допустимая скорость движения $V_{\text{доп}}$, км/ч
Асфальт, бетон	50–75	90
	150	80
	300	50
Грунтовое, укрепленное битумом	150	60
	300	40
Щебеночное	100	65
	250	50
	600	30
Булыжная мостовая	300	45
	500	40
	1000	20

Отмечается, что именно ровность дороги является основным доминирующим фактором, влияющим на формирование скоростного режима движения тягача на грунтовых дорогах.

Хотя в работе [27] и приводятся общие колебательные уравнения без схемы тягача и делаются попытки связать высоту неровности микропрофиля, она все же не дает ответа по определению скорости движения. Для этого необходимо совместное моделирование движения и колебаний автомобиля с учетом двигателя, трансмиссии, подвески и влияния вибраций на водителя. Пока эти работы ведутся раздельно в основном по моделированию движения автомобиля [19, 21, 24, 25]. Исследования же по совместному моделированию

движения и нагруженности автомобиля отсутствуют [26–45]. Для проведения таких работ требуются массивы макро и микропрофилей дорог, которых у большинства исследователей нет.

Необходимо учитывать, что моделирование колебаний является ключом для определения нагрузок, как на саму машину, так и на водителя и работающей в ней персонал. Отсюда требуется изучение вопросов: как более точно получить данные по вибрациям, действующие на колесную и штабную машину и ее кузов-контейнер в дорожных условиях. Для этого необходимо располагать данными по макро и микропрофилю дорог и режиму движению автомобилей. Для их получения необходимо использование новых подходов, например, GPS.

Учет характеристик элементов. Важную роль в точности расчетных исследований играет правильный выбор расчетной схемы модели и корректное задание характеристик упругих и демпфирующих элементов, а также модели трения [14]. В большинстве исследований по плавности хода и устойчивости грузовых автомобилей листовую рессору моделируют как упругий элемент, параллельно соединенный с элементом “сухого” трения, что далеко не в полной мере отражает сложный характер сил трения в рессоре и служит источником значительного расхождения расчетных и экспериментальных данных, особенно при малом уровне деформации рессоры. Использование модели кулоновского трения приводит к неустойчивости колебательной системы, и ее сейчас не применяют. Известна модель “вязкого” трения [43, 45], которая дает приемлемую сходимость (расхождение 10–15 %) при переезде препятствия (высотой более 100 мм), движении по грунтовым дорогам и дорогам с булыжным покрытием. Интересна модель листовой рессоры, предложенная в работе [46] и реализованная в виде блок-схемы на АВМ, которая позволяет корректно моделировать ее динамические характеристики на небольших прогибах. Фактически, это модель трения с параллельным включением дополнительных упругих элементов с ограничением их усилий при небольших деформациях рессоры при смене направления движения вследствие блокировки проскальзывания листов. Однако в работе [46] не дано физического обоснования и математического описания модели, из-за чего многими она не была понята, не развивалась и в СССР, СНГ не использовалась, кроме [14]. Но ее, судя по видео [1], сейчас использует фирма Мерседес-Бенц.

По данным работ [14, 47–49] тип амортизатора (двухтрубный, однотрубный газонаполненный) и расчетная модель трения оказывают существенное влияние на форму амплитудно-частотной характеристики колебательной системы и ее виброзащитные свойства при реальном дорожном воздействии. Особенно сильно влияет модель трения на асфальтовых дорогах

(среднеквадратичная высота неровностей 1–1,2 см²). Так использование математических моделей [43–45] с “вязким” трением приводит к большим погрешностям (до 30–50 %) по среднеквадратичным ускорениям в октавных полосах частот. При амплитудах воздействия $Z > 20$ мм влияние модели трения сказывается меньше [47–49].

В работе [47] автором предложена новая модель трения для подвесок ТС в виде сочетания последовательного соединения элементов трения, упругих и демпфирующих элементов, учитывающая динамическое изменение величин трения (во время смены направления движения и установившегося движения), и дано ее математическое описание. Эта модель обеспечивает хорошую сходимость (расхождение 2–8 %) в октавных полосах частот при моделировании вертикальных колебаний на асфальтовом и булыжном шоссе, корректность амплитудно-частотной характеристики при малых возмущениях и объясняла смещение пика резонансной частоты системы вправо, в то время как при моделях с “вязким” трением расхождение по ускорениям на асфальтовом шоссе достигало 40–50 %. В дальнейшем она была усовершенствована для Simulink [48].

Данную модель трения [48] целесообразно применять при моделировании рессор и в других устройствах. Имеются и другие модели неупругого сопротивления в листовой рессоре, но они либо сложны и неудобны для практического использования, либо не отражают наиболее важных сторон процесса диссипации энергии колебаний при деформировании рессоры и также могут вносить погрешности в расчеты колебаний автомобиля.

В то же время по данным работ [28, 63, 67, 73] для задач моделирования переезда препятствий, устойчивости с их большими величинами деформации и расчетов напряженного состояния кузовов-контейнеров в наиболее нагруженных режимах, можно обойтись более простыми моделями трения и использованием пакета MSC.ADAMS с моделью “вязкого” трения.

В работе [29] рассматривается модель шины при переезде препятствий на основе использования 3D пакетов, ANSYS, Simulink, SimPack (рисунок 1.5).

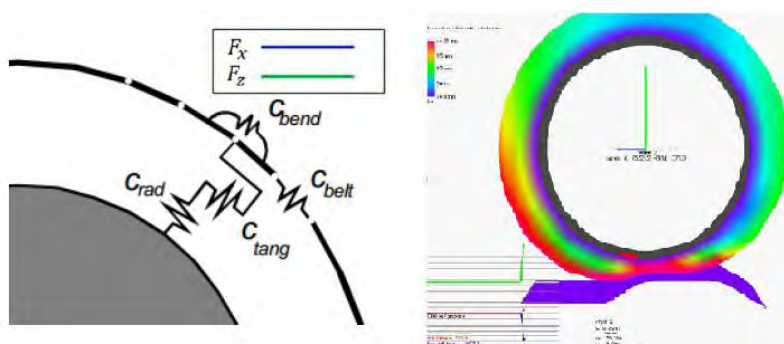


Рисунок 1.5. – Модель и напряженно-деформированное состояние шины

Похоже, основная цель работы все же была оценка напряженно-деформированного состояния шины, возникающие моменты и динамические нагрузки, демонстрация овладения магистрами этих пакетов поскольку нет математического описания и коэффициентов элементов.

По данным работ [55, 57, 60] при решении задач движения и колебаний проще и целесообразнее рассматривать шину в виде упругого элемента с "вязким" трением с точечным контактом (рисунок 1.6).

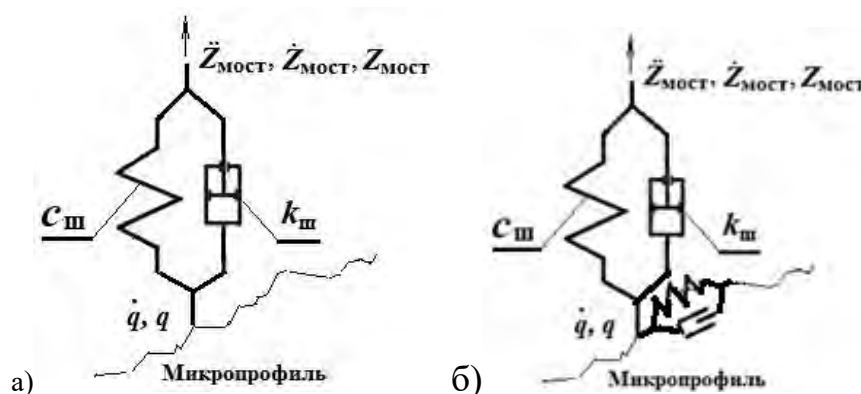


Рисунок 1.6. – Модель шины в вертикальной (а) и в продольной плоскости (б) для моделирования автомобиля

1.4 О моделях движения автомобиля

Для проведения моделирования движения автомобиля обычно используют классическую математическую модель как сумму сил, действующих на него [19, 24, 32, 33, 62] (см. подраздел 3.3). Наиболее сложными и ключевыми моментами ее реализации является задание угла наклона дороги, режима движения автомобиля и логики переключения КПП. Сейчас в основном угол наклона дороги определяют на основе макропрофиля, используя данные радарного сканирования земли, имеющиеся на некоторых сайтах [32, 33, 84] и задаваемые через MS.Excell в Matlab/Simulink. Данный подход использует ОИМ НАН РБ, Нижегородский технический университет, МАЗ и другие. К недостаткам метода следует отнести невысокую точность данных на доступных сайтах (до 1 м по высоте и точность указания точки месторасположения 20–30 %). Хотя современные спутники могут обеспечить точность от нескольких до десяти сантиметров, но они имеются на закрытых сайтах. Кроме того, в этих работах используется большой шаг точек замеров макропрофиля (250–400 м), который по данным работы [14] приводит к погрешности до 50% на местных дорогах. В то же время такой шаг приемлем для магистральных трасс, где осуществлено выравнивание определенных

участков трассы по высоте. В работе [14] отмечается необходимость использования при расчетах значительных массивов пути от 10 до 100 км и шага замеров 50 м, чтобы уменьшить погрешность моделирования.

Из проскальзывающей в интернете информации известно, что зарубежные фирмы широко используют моделирование движения автомобиля с учетом дорожных условий и информацию о наклоне дороги для выбора оптимальной передачи АКПП [30]. В то же время нет данных о комплексных подходах по одновременному решению задач движения и вибронагруженности автомобиля в дорожных условиях. За рубежом в интернете [21, 27–28, 34–39] в основном преобладают публикации преподавателей, магистров, использующих известные положения теории автомобиля, не внесших ничего существенно нового в ее развитие. Единственное, что они широко используют пакеты трехмерного проектирования, MS.ADAMS, Matlab/Simulink для решения классических задач при упрощенном воздействии дороги (трапецеидальном, синусоидальном). К единственной серьезной работе можно отнести [34], где рассматривается раздельное моделирование процессов работы двигателя и трансмиссии, проведенной еще в 1995 г. с использованием Simulink на рабочих станциях. Однако в ней не рассмотрены совместные колебательные процессы в двигателе и трансмиссии.

Вторым ключевым моментом является осуществление логики переключения передач. В классическом варианте рекомендуется переключать передачи на высшую при числе оборотов 0.9 от максимальных, а на низшую при 900 об/мин [60]. В работах ОИМ НАН РБ [30, 31] рекомендуют использование реверсивных характеристик с учетом загрузки двигателя (дизеля) при оборотах двигателя 1500–1600¹мин, что обеспечивает более минимальный расход топлива при одновременном снижении средней скорости движения. Данный подход по оценке автора [14] в случае совместного моделирования движений и колебаний автомобиля приводит к неустойчивости колебательной системы и остановке процесса моделирования в MatLab/Simulink. Это связано с реализацией сложных нелинейностей в системе и скачками реального макропрофиля, заданного мелким шагом пути (0,125 м) при шаге моделирования 0,001 с. Факт такого резкого изменения макропрофиля замечен на графиках работы [32]. К тому же в работах ОИМ НАН РБ используются данные по углу наклона дороги, заданные с большим шагом, где такое воздействие нивелируется по сравнению с использованием высоты при совместном макро и микропрофиле.

Поэтому для совместного моделирования движения и колебаний в работах [57–58] рекомендуется использование совместного макро и микропрофиля дороги, заданного в виде большого массива (96000 точек) и

заносимого в оперативную память компьютера и расчетом наклона дороги по усредненным значениям. Этот подход представляется более перспективным.

Третьим ключевым моментом является оценка топливной экономичности машины. Здесь представляет интерес использование подхода ОИМ НАН РБ (Р.Э. Шейбак), предложившего задание воздействия водителя на педаль газа как разность скорости движения по пути, пересчитанных оборотов с использованием ПИД-регулятора и характеристики двигателя. Единственным некорректным в подходе является использование внешней характеристики двигателя при определении топливной экономичности. Представляется, что необходимо учитывать степень загрузки двигателя (частичные загрузки), исходя из реальных затрат на движение при подъемах, ускорениях, исключая спуски, с учетом макропрофиля дороги.

Из отрывочных данных в интернете по этим направлениям следует, что на ведущих автомобильных фирмах ведется подобная серьезная работа, включая использование виртуальной/смешанной реальности для задач движения и колебаний автомобиля [74–77], являющая их Know how.

1.5 Оценка моделей колебаний автомобиля

По данным работ [14, 56] для совместного моделирования движения и колебаний автомобиля вполне достаточно использование плоской модели. Важно только применение более корректной модели трения в подвеске и задание реального микропрофиля дороги по точкам на основе использования модуля S-Function Builder Simulink с занесением их массивов в оперативную память компьютера, позволяющее реализовать режим реального времени.

Нерешенным является вопрос, как моделировать продольные и поперечные колебания автомобиля. Сейчас все сосредоточено на вертикальных колебаниях. И нигде нет предложений по корректной модели продольных и поперечных колебания автомобиля.

В работах [28, 30, 72] продольные колебания пытались схематично представить, как воздействие микропрофиля на шину в виде упругого элемента в продольном направлении. Непонятно, как задавать возмущение и что является источником: то ли скорость изменения высоты. Эта проблема так и не была решена.

В дальнейшем в данной работе она была смоделирована. И оказалось, что данный подход ошибочен и было предложено иное объяснение и реализация.

Важным является оценка долговечности колесных машин, которую можно осуществить через цепочку этапов: по колебаниям автомобиля выйти

на нагрузки, далее через 3D модель выявить связь нагрузок и напряжений в узлах, и по усталостным характеристикам материалов и систематизации данных напряжений оценить долговечность. Вот почему в работе акцентировано внимание на колебаниях автомобиля, зависящих от движения, его режима и наоборот.

1.6 Обобщение проблем, требующих решения

С учетом всего вышеизложенного в данной работе сконцентрировано внимание на рассмотрении вопросов оценки таких параметров как тягово-скоростных, топливной экономичности, плавности хода, управляемости и устойчивости. Анализ состояния вопроса показывает, что существует ряд нерешенных проблем, а именно:

- несовершенство моделей движения, колебаний автомобиля, приводящие к некорректности расчетов;
- получение и задание возмущения реальной дороги и ее обстановки, режима движения, обеспечивающие более корректную точность расчетов;
- проведение совместного моделирования движения и вибронагруженности автомобиля с учетом дорожных условий и режима движения;
- выявление какой режим эксплуатации является определяющим с точки зрения вибронагруженности колесных машин специального назначения;
- что обуславливает продольные колебания и как их реализовать;
- какие модели и подходы целесообразно использовать для расчета динамических нагрузок, действующих на колесные машины;
- использование микропроцессорные системы для исследования автомобиля;
- как реализовать смешанную реальность при исследовании автомобиля.

Сейчас важно уметь на стадии проектирования с достоверной точностью оценивать параметры будущих и существующих машин, что требует применения новых методов моделирования, базирующихся на системе “дорога-автомобиль-водитель”, использования более совершенных моделей, численных методов решения дифференциальных уравнений с реализацией реального возмущения.

Необходимость таких работ обусловлена значительным расхождением (до 100 %) с экспериментом [127] при оценке нагруженности при использовании спектральной плотности дороги, передаточных функций в MSC.ADAMS. Значительное расхождение (до 50%) имело место и при численных методах на существующих тогда моделях с использованием “вязкого трения” [45].

Поэтому разработка научных основ и методология моделирования автомобиля в рамках системы “дорога–автомобиль–водитель” на качественном новом уровне для решения задач движения, колебаний, управляемости, нагруженности, долговечности, топливной экономичности автомобиля с задействованием симулятора и смешанной реальности является актуальной задачей.

ВЫВОДЫ К ГЛАВЕ 1

1. Выполнены обзор, анализ применяемых методов моделирования, используемых моделей, подходов при их реализации и испытаниях и имеющихся проблем. Рассмотрены сферы применения современных программно-технических средств, области применения плоской и пространственной модели колесной машины, необходимость корректного учета характеристик элементов подвески, включая трение, податливости рамы, шин, задания реального возмущения.

2. Выявлено отсутствие необходимых для проведения моделирования автомобилей на качественно новом уровне более корректных их реальных моделей, массивов продольных и микропрофилей дорог, режимов движения без чего это невозможно.

3. Обоснован перечень подлежащих рассмотрению в работе основных задач, подходов и инструментальных средств для их решения, использования более совершенных моделей, новых методов получения и реализации реального возмущения дороги, включая GPS, визуализации дороги, учета фактора водителя на основе имитационного моделирования со смешанной реальностью на симуляторе.

ГЛАВА 2

ИСТОЧНИКИ ВОЗМУЩАЮЩЕГО ВОЗДЕЙСТВИЯ НА АВТОМОБИЛЬ И МЕТОДЫ ИХ ПОЛУЧЕНИЯ

Более тщательное рассмотрение в главе источников возмущающего воздействия на автомобиль обусловлено:

- существенным влиянием воздействия дороги на колебания КМ;
- некорректностью получаемых результатов при синусоидальном воздействии на автомобиль в исследованиях 1960–1975 гг. [41, 57] и при использовании случайного возмущения воздействия дороги как следующего этапа развития моделирования 1975–1985 гг. [57];
- расхождением результатов моделирования и эксперимента по спектру даже при использовании микропрофиля дороги, заданного по точкам [43];
- необходимостью получения более корректных результатов расчета колебаний в т.ч. продольных и поперечных и долговечности узлов.

Микропрофиль дороги, заданный по точкам, впервые использовался еще в 1965 г. [43] при моделировании колебаний автопоезда на ЦВМ. В качестве критериев корректности в работе использовались величина максимальных ускорений шасси при переезде препятствий высотой 100 мм и среднеквадратические значения вертикальных ускорений на булыжной дороге по общей полосе 0–22,4 Гц. Расхождение составило 10–15%. Дальнейшее развитие работ выявило значительное расхождение по спектру (в октавных полосах частот) (до 50%) результатов расчетов и эксперимента несмотря на использование такого микропрофиля [43, 45] и его $\tan$ наклона в качестве источника продольных вибраций автомобиля. Все это обусловило необходимость совершенствования моделирования и более тщательного рассмотрения источников возмущающего воздействия, к которым относятся:

- дорога, ее продольный (макро) профиль, микропрофиль и шероховатость ее поверхности;
- неравномерность вращения колес, их эксцентриситет;
- неуравновешенность колес, вращающихся частей двигателя, колебательные процессы в них и трансмиссии;
- взрывной характера сгорания топлива, обуславливающий неравномерную работу двигателя.

2.1 Дорога

2.1.1 Профиль дороги делится на две составляющие: продольный/макро и микропрофиль [52, 53]. К продольному профилю относятся неровности ($<0,7$ Гц) с длиной волны 50–100 м и более. Продольный макропрофиль

представляет собой зависимость изменения высоты по длине дороги в виде массива, заносимого по точкам с шагом 50–100 м (рисунок 2.1).

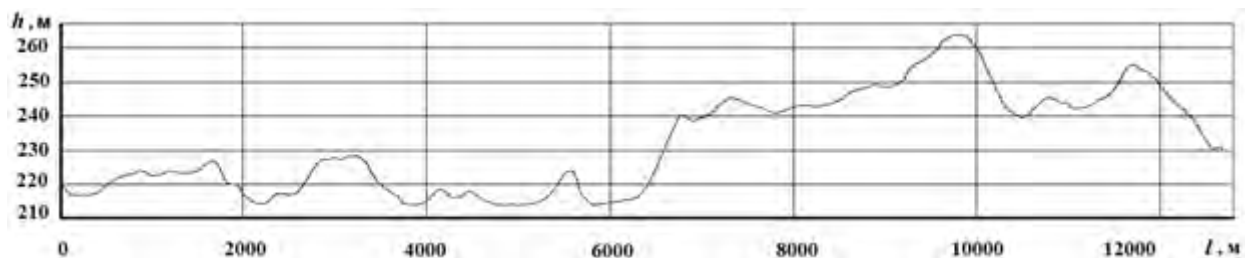


Рисунок 2.1. – Продольный (макро) профиль дороги

Продольный профиль оказывает влияние на тяговую динамику колесных машин в виде сопротивления наклону дороги, но практически не влияет на вибрации. Шероховатость (длина волны 0,05 м и менее) оказывает влияние только на сопротивление качения, шум и не влияет на вибрации.

Микропрофиль – неровности с длиной волны 10 м и менее с воздействием $>0,7$ Гц, влияющие на колебания автомобиля. Величина шага при моделировании обычно задается 0,1–0,15 м (рисунок.2.2). Ранее при замерах использовался шаг 0,5 м [43, 45].

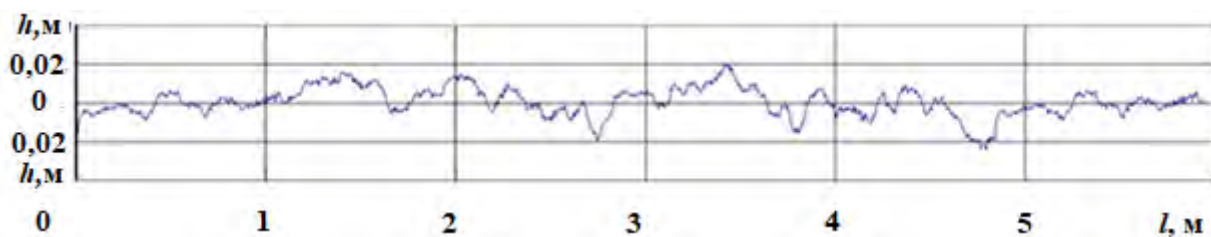


Рисунок 2.2. – Микропрофиль дороги

Его воздействие также сказывается и на динамике автомобиля в виде изменения сопротивления движения. Данное разделение во многом условно и зависит от скорости движения и собственной частоты подвески.

Использование макро и микропрофиля дорог является ключевым моментом в проведении современных расчетов транспортных средств, так как позволяет более адекватно оценивать их параметры и нагруженность автомобиля. Необходимо учитывать, что моделирование колебаний численными методами в Simulink реально возможно лишь в полосе 0–28 Гц.

На скорость движения помимо тягово-скоростных свойств также влияют:

- дорожная обстановка: движение встречных и попутных автомобилей;
- ограничение скорости на отдельных участках, населенных пунктах, дорожная разметка, ограничивающая обгоны;

- повороты дороги и подъемы, переезд путей, мостов на которых приходится снижать скорость;
- состояние дороги, количество полос, манера движения водителя;
- фактор водителя.

Без учета воздействия реальной дороги и реакции на нее водителем невозможно провести оценку динамики и нагруженности автомобиля.

Идеальным вариантом является использование симуляторов (рисунок 2.3), которые позволяют воспроизводить вибрационное воздействие и дорожную обстановку и проводить на их основе исследования. Но симуляторы на рельсовой системе (рисунок 2.3) дорогие (~ 20 млн. \$).

К сожалению, из-за сложной экономической ситуации, отсутствия высококвалифицированных кадров, косности и нежелания что-то делать симуляторы не получили широкого применения на автозаводах СНГ.

За рубежом такие симуляторы больше применяются для задач управляемости легковых автомобилей, отладки рулевого управления и подвески. Из рекламных видеоматериалов зарубежных фирм не видно, что они используют в своих исследованиях разбитые дороги. Внешняя обстановка задается у них с помощью компьютерной графики, где не отражено состояние дорожного полотна. Да и высокое качество их дорог не делает актуальным проведение исследования вибрационного воздействия.

Сейчас для имитационного моделирования больше используются 6-DoF платформы на электроактуаторах без рельсовой системы (рисунок 2.3).



Рисунок 2.3. – Использование симуляторов за рубежом в автомобилестроении

Исследование, проведенное на настольном макетном образце имитатора (рисунок 2.4), показало, что он не позволяет полностью воссоздать обстановку реальной дороги и в машине из-за отсутствия вибрационного воздействия [14]. И такая же ситуация с дополненной и смешанной реальностью.

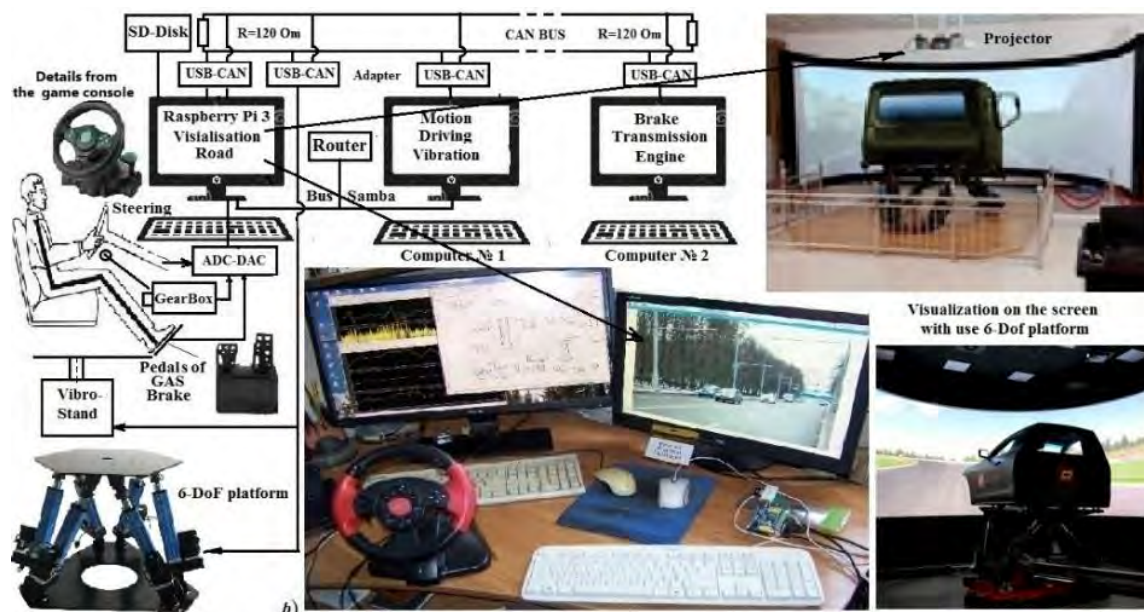


Рисунок 2.4. – Использование настольного макетного образца имитатора и симуляторов на базе 6-DoF платформах с экранами

Важным моментом при моделировании является каким способом задавать возмущение дороги. Оно может быть задано двумя способами: путем использования генераторов белого шума с последующей фильтрацией [52–59] и использования реального возмущения дороги [42–45], замеренного и заданного по точкам. Для реализации первого требуется наличие статистических характеристик дорог, их математической аппроксимации. Данный подход широко использовался на аналоговых машинах (АВМ) в проблемной лаборатории автомобилей БПИ и в работах Гришкевича А.И. [40, 46], но сейчас практически не применяется.

Сейчас больше применяют использование реального возмущения макропрофиля дороги [43–46], заданного с помощью MS.Excel по точкам (до 1024 точек) в Matlab\Simulink на основе данных радарного сканирования земли (ОИМ НАН РБ, НГТУ, МАЗ). При задании по точкам [43, 45, 72] получаются более корректные результаты (погрешность 10–15%) при расчетах движения и плавности хода автомобилей. При радарном сканировании в зависимости от шага может наблюдаться ступенчатый характер изменения высоты профиля дороги (провалы значений высот, рисунок 2.5) [30].

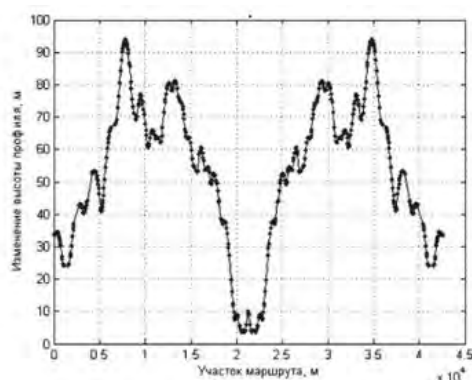


Рисунок 2.5. – Продольный профиль участка маршрута “Заславль–Радошковичи–Заславль”

На корректность расчетов влияют: способы задания возмущения дороги, спектр, форма воздействия, его характер и закон распределения. Спектр возмущения оказывает существенное влияние на колебания: чем больше доля высокочастотных, тем меньше низкочастотные вибрации, и наоборот.

Ранее для микропрофиля дороги широко использовались сигналы с помощью генераторов белого шума (ГБШ) и фильтров, но это применимо лишь для фиксированной скорости движения, поскольку требуется для каждой скорости подбирать и корректировать параметры фильтров, что крайне неудобно. Данный вариант хотя и обеспечивает приемлемое расхождение с экспериментом в октавных полосах частот 2, 4, 8, 16 Гц в пределах 8–15 % (рисунок 2.6, приведенный в моей работе [14]), однако для специальных машин, эксплуатирующихся в режиме переменной скорости, он не подходит.

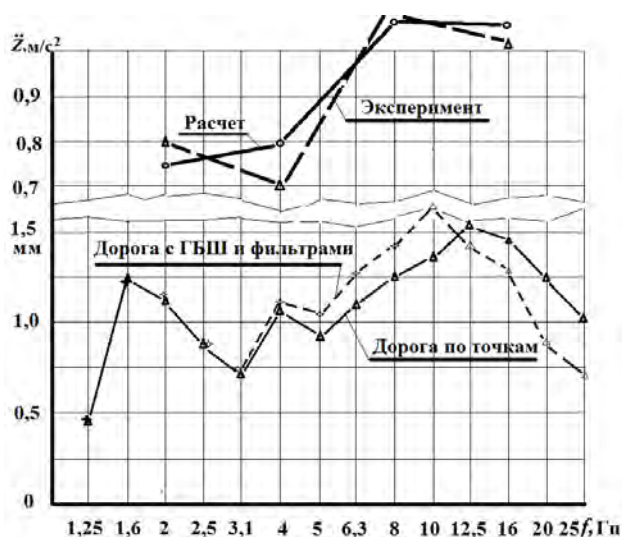


Рисунок 2.6. – Среднеквадратический спектр микронеровностей дороги в третьоктавных полосах частот и вертикальных ускорений на сиденье водителя в октавных полосах частот

Следует отметить, что ступенчатый характер сигнала ГБШ требует хорошей фильтрации т.к. при использовании сложных зависимостей трения он

может привести к остановке процесса моделирования численными методами. ГБШ больше подходят для применения на линейных моделях.

Ранее для замеров микропрофиля дороги широко применялся нивелир, а также использовался метод “качающегося” маятника, установленного на буксируемой тележке, с последующим преобразованием сигнала через передаточную функцию двухмассовой колебательной системы [52–54]. Результатом последнего является получение характеристики спектральной плотности дорог и их аппроксимация. Затем на основе уже их с помощью генератора белого шума (ГБШ) и передаточных функций фильтров формировалось случайное возмущение дороги, которое использовалось при расчетах. Недостатками такого подхода являются двойное преобразование, снижающее точность и необходимость изменения параметров фильтров от скорости, что усложняет моделирование.

Сейчас основным пакетом с помощью которого ведется моделирование движения и колебаний является Matlab/Simulink. В модуле Adams/Vibration пакета MSC.ADAMS имеется ограничение по количеству задаваемых точек реального профиля дороги (несколько десятков). Для проведения расчетов на реальных дорогах необходим охват участков от 10 до 100 км пути. Это приводит к необходимости задания порядка 1000 точек и более. Как показано в работе автора [14] необходимо задавать с шагом как минимум 100 м. Получить на сайтах меньший шаг не удастся. Увеличении шага до 250–400 м по данным работы [14] приводит к двухкратной погрешности на местных дорогах. Данный шаг может использоваться только для магистральных дорог I-й категории и не подходит для условий эксплуатации колесных машин повышенной проходимости.

Поэтому в работе ориентировались на использование метода получения экспериментальных данных по определению макропрофиля дороги и режима движения колесной машины специального назначения с помощью GPS и применения его при моделировании.

В большинстве работ, публикуемых в интернете, применяются те же модели, характеристики и подходы, что и в 1960–1980 гг. на основе упрощенных линейных однопараметрических зависимостей, передаточная функция и гармоническое возмущение, что не дает приемлемой точности.

В пакете MSC.ADAMS можно задать гармоническое, ступенчатое, трапецеидальное возмущение. Предусмотрена возможность задания стационарных случайных нагрузок и импульсного широкополосного воздействия через использование методов линеаризации. Однако по ним нет работ. В пакете Matlab/Simulink стандартной функции задания микропрофиля дороги по точкам также нет. Поэтому в работах, связанных с ним, в основном используется гармоническое возмущение либо случайное возмущение,

формируемое с помощью генератора белого шума (ГБШ). Пакет Matlab/Simulink имеет гораздо больше возможностей для моделирования нестандартных ситуаций. Сейчас на Matlab/Simulink можно подключить компилятор языка C/C++ в т.ч. MS Visual Studio и через модуль S-Function Builder программно реализовать возмущение по точкам с занесением в память.

Но большинство исследователей по-прежнему применяют передаточную функцию и гармоническое возмущение, приводящие к большому расхождению и также не дают при расчетах оценки вибраций в октавных и третьоктавных полосах частот [52–54]. Такие подходы являются нонсенсом на нынешнем этапе. Все это в комплексе приводит к значительному расхождению с экспериментом и потере доверия к получаемым данным при расчетах.

Более оптимальным является использование задания возмущения с помощью разработанной программы (приложение А), реализуемой на языке C на компиляторе MS Visual Studio v.2022 в модуле S-Function Builder Simulink (ранее использовался TDM-GCC (версии 5.1.0.2/10.3.0)) [14]

2.2 Получение продольного профиля дороги с помощью GPS

2.2.1 Для проведения расчетов движения и нагруженности автомобиля необходимо располагать массивами продольных профилей и микропрофилей дорог. К сожалению, полученные в 70–80 годах статистические данные по микропрофилям дорог [50, 51] во многом устарели, так как изменился характер интенсивности эксплуатации дорог, а также из-за смены поколений вычислительной техники утеряна непосредственная информация по ним. Кроме того, использование их затруднено в современных пакетах из-за ступенчатости сигнала с генератора “белого” шума, требующего сложной дополнительной фильтрации и необходимости перестройки параметров фильтров при изменении скорости. Также необходимо учитывать ухудшение параметров дорог (микропрофиля) в процессе эксплуатации. Поэтому важно отслеживать состояния дорог, особенно при проведении сравнительных испытаний, расчетов на плавность хода и тяговой динамики автомобиля.

Массив продольного профиля дороги можно получить следующими способами:

- на основе строительных, геодезических карт дорог, их обработки;
- проведения замеров дорог с помощью геодезического оборудования, нивелиров [50];
- использования дорогостоящих мобильных лабораторий, позволяющих фиксировать наклоны дорог в процессе движения с помощью гироскопов, а также “качающегося” маятника [50, 51];

- регистрации параметров CAN шины электронного блока управления силовым агрегатом движущегося автомобиля и положения автомобиля на трассе с помощью датчика GPS и преобразований на основе уравнений тяговой динамики [83–86];

- использования данных космического радарного сканирования земли [87, 88], имеющиеся в интернете;

Чтобы получить микропрофиль дорог требуется использование дорогостоящих мобильных лабораторий, позволяющих фиксировать перемещение поддрессоренных и неподдрессоренных частей автомобиля и расстояния от рамки на поддрессоренной части до полотна дороги с помощью радарных датчиков и последующего математического преобразования [86].

Эти методы сложны и затратны, а ранее применяемые методы и подходы получения микропрофиля дорог с помощью качающегося маятника уже не отвечают современному уровню. Последний более и менее еще подходит для дорог с большими значениями высот микронеровностей дорог (порядка 5–10 см², которых становится все меньше), но уже не годится для замеров микропрофилей с среднеквадратических значений высот 1–2 см².

Основным способом оценки ровности дорог, применяемыми строителями и эксплуатационщиками дорог, является метод IRI в основе которого среднеарифметические значения вычисляемого параметра высоты неровности (q_0/L), как разности перемещений кузова и высоты до неровности дороги [50, 51]. Его метод базируется на регистрации ускорений (с последующим преобразованием в виброскорость и перемещение) поддрессоренной и неподдрессоренной масс движущегося автомобиля (при $v=80$ км/ч). База блоков замеров дороги L принята равной 3 м при шаге 0,125 м. Отметим, что в IRI фактически вместо высоты неровности дороги q_0 используется перемещение Z_n неподдрессоренной массы автомобиля, что может быть корректно только до 4–5 Гц, а не областей больше 7,4 Гц для которых рассчитывается показатель IRI.

Сейчас делаются попытки заменить перемещения неподдрессоренной массы непосредственно на неровность дороги, для чего начали использоваться ИК-радарные либо ультразвуковые (сонары) датчики расстояний [86]. Недостатком метода IRI с позиций моделирования колебаний автомобиля является использование блоков коротких участков замеров (с базой 3 м). Это соответствует частоте возмущения 7,4 Гц и выше при скорости 80 км/ч. В то время как собственная частота подвески автомобиля составляет 1,2–1,6 Гц у легковых и 2–3 Гц у грузовых. Для задач моделирования требуется использование массивов данных виброскорости и виброперемещений микропрофиля ($\dot{q}_0, q_0$) в полосе частот 0,7–22,4 Гц с длиной участков дороги желательно не менее 1000 м с дискретным заданием по точкам и последующей

их интерполяцией. Из-за этого данные IRI в виде блоков с базой замеров 3 м не подходят для моделирования колебаний автомобиля.

Все это обуславливает необходимость разработки новых методов получения продольного профиля и микропрофиля дороги, пригодных, как для оценки их ровности, так и моделирования колебаний автомобиля.

2.2.2 Метод применения GPS, предложенный в данной работе, является одним из способов получения макропрофиля дороги. Он также может быть использован помимо получения массивов продольных профилей дорог также и для режимов движения, необходимых для моделирования движения автомобиля, военной техники (ВТ) и оценки их эксплуатационных характеристик на различных дорогах и трассировки полетов БЛА. Ниже использованы материалы, приведенные в совместных работах автора [65, 66].

Системы спутниковой навигации, включают наземные станции, передающий модуль на спутниках, приемный модуль (GPS, Glonass и др.) и компьютерную или микропроцессорную технику (рисунок 2.7). Сигнал с модуля GPS подается на последовательный порт GRIO Raspberry Pi 3/4 и обрабатывается программой. На компьютер с Windows сигнал может быть подан через USB (его эмулирование).

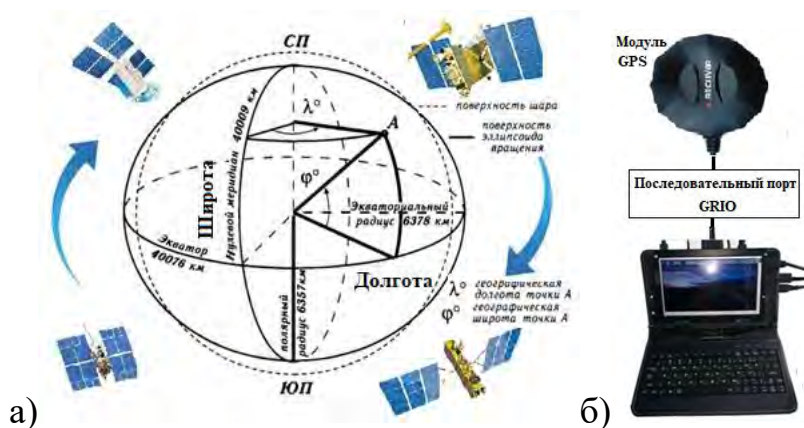


Рисунок 2.7. – Спутниковая система позиционирования и схема Земли (а), модуль GPS GN-803G, laptop на базе Raspberry Pi 3 b (б)

Сейчас уже имеются геодезические приемные модули с системой дифференциальной коррекции от базовых станций (рисунок 2.8), обеспечивающие измерения с погрешностью от несколько до десяти сантиметров, но они пока дорогие (~ 3000–17000 \$ (Leica GS)).



Рисунок 2.8. – Принцип измерения спутниковыми приемниками трехмерного вектора

Высокая точность геодезического ГНСС оборудования [59, 60] достигается за счет базовых наземных станций, координаты которых заранее определены с очень высокой точностью и за счет выполнения процедур корректуры, приведенных на рисунке 2.8.

Недостатком является необходимость стоять на месте замера (от часа до нескольких десятков секунд). В новых моделях время может быть меньше.

Для замеров профилей дорог целесообразно использование более дешевых модулей класса UBOX-M8N–UBOX-M10N (стоимостью ~ 20 \$), которые обеспечивают погрешность в движении 1–2 м, у смартфонов (7–15 м) [64–66]. Представляет интерес для этих целей использование приемного модуля GPS (GN-803G) ввиду его приемлемой точности (1–2 м) путем регистрации параметров движущегося автомобиля с помощью GPS. Однако получаемые данные со спутников требуют преобразования в метры, поскольку информация с них выдается в виде сообщений местоположения в градусах широты и долготы, а скорость в виде миль/с. И только высота выдается в метрах. Проблемным моментом является определение пройденного пути: он должен быть рассчитан исходя из других параметров GPS. Эта задача решена в навигационных приборах, но непонятна специалистам других областей. Кроме того, необходимо привести данные к постоянному шагу из-за переменной скорости движения. Также необходимо учитывать различие систем отсчета: за рубежом базовыми осями широты считается меридиан Лондона (по Гринвичу), а долготы – Экватор и центр Земли в WGS84 (GPS). В СНГ базовым для широты считается Пулковский меридиан. Для пересчета в метры за рубежом используются упрощенные зависимости, более подходящие для Европы, а в СНГ более сложные согласно ГОСТ 32453-2017 [56]. И при преобразовании получают различные данные.

Недостаточно освещенным вопросом для специалистов, не связанных с GPS, является как все это практически реализовать и какой метод целесообразнее использовать для преобразования данных, чтобы точнее

рассчитать пройденный путь, местоположение транспортного средства в метрах. Должна быть привязка высоты профиля к пути, а не по времени, учитывая движение автомобиля с переменной скоростью по различным дорогам, имеющих сложную траекторию.

Как показала проработка вопроса: пройденной путь можно получить с помощью GPS двумя способами:

– путем определения разности двух точек проекций позиционирования и их суммирования (рисунок 2.9).

– путем квантования скорости в единицу времени (частоты опроса GPS f в Гц), эквивалентную пройденному пути ($\Delta l = v \cdot f$) и их последующего суммирования.

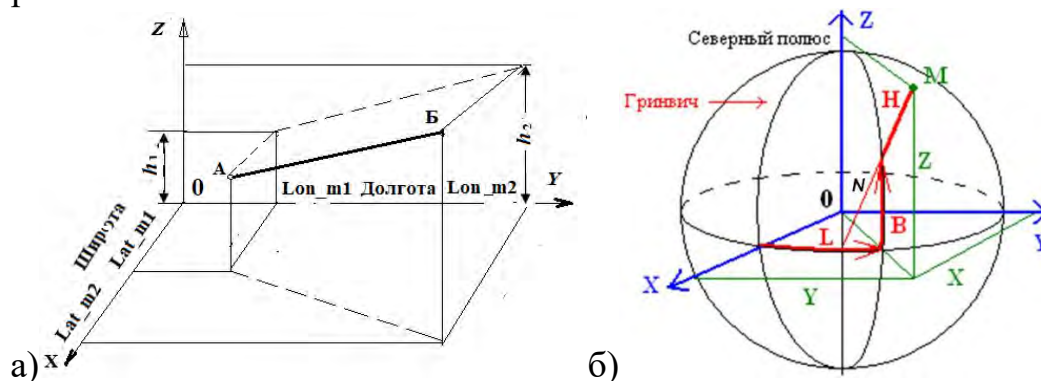


Рисунок 2.9. – Схема определения пути по двум точкам позиционирования (а) и геоцентрические координаты (б)

Величина пройденного пути между двумя точками с учетом высот равняется

$$\Delta l = \sqrt{(\text{lon_m2} - \text{lon_m1})^2 + (\text{lat_m2} - \text{lat_m1})^2 + (h_2 - h_1)^2},$$

где lon_m1 , lon_m2 – проекции координат долготы, lat_m1 , lat_m2 – проекции координат широты, h_1 , h_2 – высоты этих точек в метрах.

Для перехода от геодезических координат к геоцентрическим (рисунок 2.7) используются следующие формулы ГОСТ 32453-2017 (п.5.1.1):

$$\begin{aligned} x &= (N+H) \cdot \cos B \cdot \cos L, \\ y &= (N+H) \cdot \cos B \cdot \sin L, \\ z &= (N+H - e^2 \cdot N) \cdot \sin B, \end{aligned}$$

Здесь L – долгота, B – широта в радианах, H – высота в м, N – так называемый радиус кривизны первого вертикала:

$$N = \frac{a}{\sqrt{1 - e^2 \cdot \sin^2 B}},$$

где: $e = \sqrt{1 - \left(\frac{b}{a}\right)^2}$ – эксцентриситет эллипсоида Земли, $a = 6378137.0$ м, $b = 6356752.3142$ м – радиусы эллипсоида.

Для получения более точных плоских прямоугольных координат в принятой на территории СНГ проекции Гаусса-Крюгера и карт СК-42, СЛ-95 используют геодезические координаты на эллипсоиде Красовского, рассчитываемые по сложным формулам ГОСТ 32453-2017 (п.5.4.1) [61].

В работе [67] с помощью смартфона и бесплатной программы «AndroSensor» получены ускорения, угловые скорости (по трем осям), долгота, широта и скорость движения автомобиля и его крены в зависимости по времени. При преобразовании данных в MS Excel и MatLab использовались упрощенные формулы ($x = 62547,1826$, $y = 110595,543$ м), применимые для московской области.

Однако работ по получению с помощью GPS продольного профиля дорог в открытых источниках нет, как и рассмотрения решения их реализации.

2.2.3 Передаваемая информация со спутника

Информация со спутника системы GPS передается в виде GGA, GLL, GSA, GSV, VTG, RMC, ZDA, GNS сообщений согласно стандарту NMEA-0183 [60, 61] в текстовой форме ASCII с полями, разделенными запятой (.). Например:

\$GPVTG,333.93,T,,M,0.343,N,0.635,K,A30

\$GPGGA,043025.00,3510.97111,S,13512.82950,E,1,11,0.79,37.5,M,-4.4,M,,6D

\$GPGSA,A,3,26,03,04,02,06,30,08,07,09,27,16,,1.35,0.79,1.10,0B

\$GPGSV,3,1,12,02,06,248,22,03,19,018,31,04,52,098,44,05,00,215,73

Сообщения по протоколу NMEA-0183 содержат следующую информацию:

- GGA — данные местоположения объекта;
- GLL — географические координаты широта/долгота;
- GNS — данные местоположения GNSS;
- GSA — геометрический фактор ухудшения точности и активные спутники;
- GSV — видимые спутники;
- RMC — минимальный рекомендованный набор данных;
- VTG — скорость и курс относительно Земли;
- ZDA — время и дата.

Каждая строка сообщений начинается с символа '\$'. Следующие два знака определяют идентификатор, указывающий на спутник – источник сообщения. Например, сообщения от GPS начинаются с GP, от ГЛОНАСС – GL, от Галилео – GA, от BeiDou – GB*, от других GNSS – GN. Следующие три знака определяют вид и содержание сообщения.

Примечание. Модуль GPS GN-803G может работать одновременно с GPS и Glonass.

Для задач исследования движения автомобиля необходимо использовать сообщения GGA (таблица 2.1) и RMC (таблица 2.2), которые дополняют друг друга. В RMC отсутствуют данные о высоте над уровнем моря, а в GGA нет данных по скорости и направлению.

Таблица 2.1. – Формат данных сообщения GGA

Имя	Пример	Размерность	Описание
1	2	3	4
Message ID	\$GPGGA		GGA protocol header
UTC Time	161229.487		hhmmss.sss
Latitude	3723.2475		ddmm.mmmm
N/S Indicator	N		N=north or S=south
Longitude	12158.3416		dddmm.mmmm (degrees & min)
E/W Indicator	W		E=east or W=west
Position Fix Indicator	1		See Table 1-4
Satellites Used	07		Range 0 to 12
HDOP	1.0		Horizontal Dilution of Precision
MSL Altitude	9.0	Meters	
Units	M	Meters	
Geoid Separat		meters	
Units	M	meters	

Таблица 2.2. – Структура сообщения RMC

Поле N	Имя	Размерность Unit	Формат	Образец	Описание
1	xxRMC	-	string	\$GPRMC	RMC Message ID (xx = current Talker ID, see NMEA Talker IDs table)
2	time	-	hhmmss.ss	083559.00	UTC time. See the section UTC representation in the Integration manual for details.
3	lat	-	ddmm mmmm	4717.11437	Data validity status, see position fix flags description
4	NS	-	character	N	North/South indicator
5	lon	-	ddmm mmmm	00833.91522	Longitude (degrees and minutes), see format description
6	EW	-	character	E	East/West indicator
7	spd	kno ts	numeric	0.004	Speed over ground
8	cog	deg	numeric	77.52	Course over ground

Пример. \$GPRMC, 083559.00, A, 4717.11437, N, 00833.91522, E, 0.004, 77.52, 091202,, A,V*57.

Алгоритм получения данных заключается в считывании и идентификации строки, затем считывании с нее полей данных, контролируемых по счетчику.

2.2.4 Подключение модуля GPS GN-803G к Raspberry Pi 3/4

Более целесообразным по быстродействию для мини-компьютера Raspberry является подключения GPS-модуля через интерфейс GPIO по сравнению с USB. Однако при низких частотах квантования (1 Гц) разница практически незаметна. На рисунке 2.10 приведена схема подключения модуля **GN-803G** к Raspberry 3/4. Для подключения GPS-модуля к мини-компьютеру, необходимо соединить:

GPS модуль	Raspberry P 3/4
VCC	3.3V (пин 1)
RX	TXD/GPIO 14 (пин 8)
TX	RXD/GPIO 15 (пин 10)
GND	Ground (пин 6)

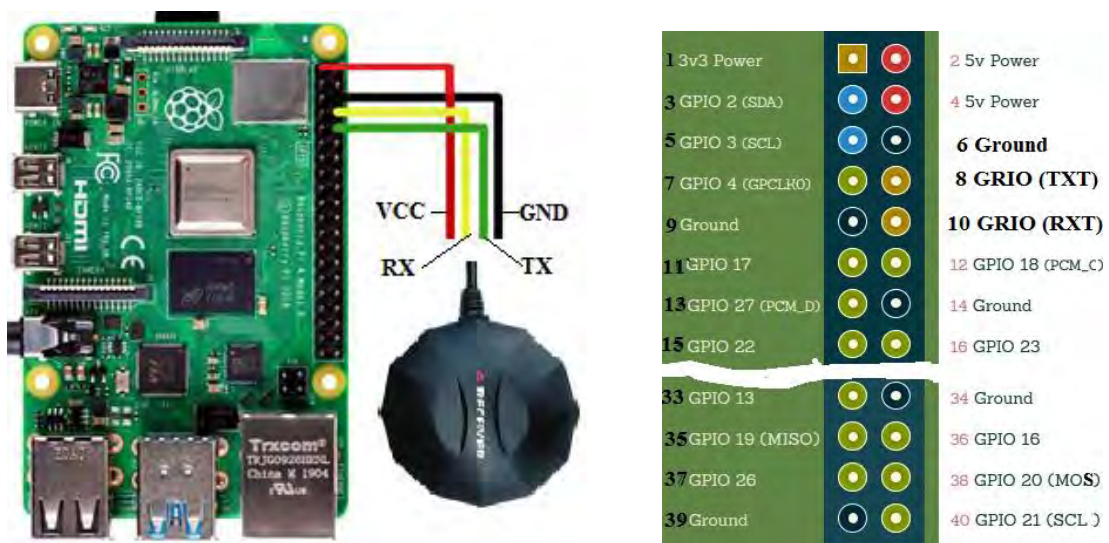


Рисунок 2.10. – Схема подключения модуля GN-803N к Raspberry P 3/4

2.2.5 Методы преобразования данных для местонахождения объекта

Рассмотрим методы преобразования данных для получения местонахождения объекта.

2.2.5.1 Геодезический метод обработки данных

В основе геодезического метода [59–61], применяемого за рубежом и более подходящего для западной Европы, используется соответствие расстояния 111320 м, приходящееся на 1° геодезического положения объекта (рисунок 2.7а).

Зная значение в градусах и умножая на этот параметр, получим расстояние от Гринвича и Экватора в метрах. Необходимо также учитывать в какой форме представлены данные со спутника. Для расчетов необходимы данные в десятичной форме. Если информация представлена отдельно в градусах (D_{dec}, D), минутах (M), секундах (S), то она должна быть преобразована в градусы по формуле

$$D_{dec} = D + \frac{M}{60} + \frac{S}{3600}.$$

Для большинства приемников и модулей GPS (Garmin, Navitel, смартфонов и др.), работающих по стандарту NMEA-0184 информация выдается в текстовом формате в виде: например, широта 02734.50963, долгота 5354.9707. Первые три (два) символа (027 53) соответствуют градусам, начиная с четвертого (третьего) минутам с секундами в десятичном виде. Для преобразования требуется программно выделить их части и произвести расчет по формуле

$$D_{dec} = D + \frac{M+S}{60}.$$

Приближенно расстояние можно определить по следующим образом:

$$\begin{aligned} \text{lon_m} &= 111320 \cdot D_{dec \text{ lon}} \\ \text{lat_m} &= 111320 \cdot D_{dec \text{ lat}} \end{aligned}$$

где lon_m – получаемое значение широты/Latitude в метрах, lat_m – значение долготы/Longitude в метрах. $D_{dec \text{ lon}}$ – угол широты в градусах, $D_{dec \text{ lat}}$ – угол долготы в градусах.

2.3.5.2 Метод проекции Меркатора

Метод проекции Меркатора [58, 59], разработанный для морской навигации, базируется на использовании углов долготы, широты и радиуса Земли. Местоположение в метрах определяется следующими формулами:

$$X = \alpha \cdot \text{lon},$$

$$Y = a \cdot \ln \left[\tan\left(\frac{\pi}{4} + \frac{\text{lat}}{2}\right) \cdot \left(\frac{1-e \cdot \sin(\text{lat})}{1+e \cdot \sin(\text{lat})}\right)^{e/2} \right]$$

где: lon/lat - долгота/широта в радианах, e – эксцентриситет эллипса Земли в метрах.

$$e^2 = 2 \cdot f - f^2$$

где: f - коэффициент сжатия Земли, $a = 6378137.0$ м, $b = 6356752.3142$ м – радиусы эллипсоида.

$$f = \frac{a-b}{b}.$$

Либо выразить эксцентриситет e через полуоси,

$$e = \sqrt{1 - \left(\frac{b}{a}\right)^2}.$$

Для расчета использовались функции на C/C++, приведенные в работе [48].

2.2.3.3 Метод по cos угла

В работе [59] предлагается определять расстояние по следующим формулам

$$\begin{aligned} \text{lon_m} &= \text{lon_d} \cdot (111412.84 \cdot (\cos(\lambda) - 93.5 \cdot \cos(3 \cdot \lambda) + 0.118 \cdot \cos(5 \cdot \lambda))), \\ \text{lat_m} &= \text{lat_d} \cdot (111132.92 - 559.82 \cdot \cos(2 \cdot \varphi) - 1.175 \cdot \cos(4 \cdot \varphi) - 0.0023 \cdot \cos(6 \cdot \varphi)) \end{aligned}$$

где λ , φ – соответственно широта и долгота в радианах (рисунок 2.7а), lon_m, lat_m – получаемое значение расстояния широты и долготы в метрах.

2.2.5.4 Метод коррекции углов, получаемых с спутника

Метод коррекции углов [59, 60], базируется на том факте, что радиус Земли на экваторе составляет 6 378 137,0 м при соответствующей окружности 40 075 016,7 м. Экватор разделен на 360 градусов долготы. Таким образом каждый один градус на экваторе составляет 111 320 м. При перемещении от экватора к полюсу уменьшается длина Δ_{lon} приближаясь к нулю на полюсе (таблица 2.3).

Таблица 2.3. – Изменения угла долготы от угла широты

φ	0°	15°	30°	45°	60°	75°	90°
Δ_{lat}	110574 м	110649 м	110.852	111.133	111412	111618	111694
Δ_{lon}	111.320 м	107551 м	96486 м	78847 м	55800	28902	0.000

В работе [60] предлагается градусы долготы умножать на косинус широты. При сферической форме Земли радиус $a = 6\,367\,449$ м при долготе φ соответствует $a \cdot \cos \varphi$ и длина одного градуса составляет $\Delta_{lon} = \frac{\pi}{180} \cdot a \cdot \cos \varphi$.

При рассмотрении Земли как эллипсоида, длину от одного градуса угла предлагается рассчитывать как,

$$\Delta_{lon} = \frac{\pi \cdot a \cdot \cos \varphi}{180 \sqrt{1 - e^2 \cdot \sin^2 \varphi}},$$

где e – эксцентриситет эллипсоида, равный $e^2 = \frac{a^2 - b^2}{a^2}$ (b / a составляет 0,99664719).

Альтернативной является формула

$$\Delta_{lon} = \frac{\pi}{180} \cdot a \cdot \cos \beta$$

где $\tan \beta = \frac{b}{a}$, β назван уменьшенной (или параметрической) широтой.

2.2.5.5 Метод определения пройденного пути, продольного профиля дороги, его высоты и режима движения автомобиля по параметру скорости

Предложенный в этой работе метод определения пройденного расстояния, продольного профиля дороги, его высоты и режима движения автомобиля, базируется на регистрации параметров скорости и высоты движущегося объекта по дороге, получаемых с модуля GPS и последующих их математических преобразований. Использовано соответствие параметра скорости (v) пройденному пути ($\Delta l = v \cdot f$) в единицу времени с учетом частоты квантования данных (в данном случае $f=1$ Гц, соответствующему шагу в 1 сек.) и суммировании их значений ввиду более высокой точности этого параметра (0,05 м/с).

Примечание. Значение скорости перемещения объекта в GPS выдается в виде миль/с. Для перевода в м/с это значение необходимо умножать на 0,5144.

2.2.5.6 Программа считывания и обработки данных

За основу при разработке ПО взята программа NMEA-GPS-master.zip [68], которая только считывает и выводит на экран определенную информацию. Она была переработана для получения отсутствующей в программе информации по RMC, расчету текущего положения в метрах, сведения данных в единую бинарную структуру данных и записи из нее требуемой информации в файл на карту памяти для последующей обработки в Matlab/Simulink.

Исходный текст программы, реализованный на языке C/C++ в Raspberry P 3b+, приведен в приложении Б.

Полученные с помощью модернизированной программы данные в дальнейшем подвергались преобразованию для приведения к фиксированному шагу с помощью полинома Лагранжа 2-го порядка по трем точкам [69] переменного шага.

2.2.5.7 Приведение полученных данных к фиксированному шагу

При проведении исследований оценка методов осуществлялась на основе данных, полученных на автомобиле, движущемся с переменной скоростью, т.к. сложно в условиях дорожного движения обеспечить постоянную скорость. Из-за этого шаг квантования по длине получается также переменным. Для приведения полученных данных к постоянному шагу в

работе использовалась коррекция данных с помощью полинома Лагранжа 2-го порядка по трем точкам [69] (погрешность 0,5 %).

$$y = L2(x) = (((x - x_1) \cdot (x - x_2)) / ((x_0 - x_1) \cdot (x_0 - x_2))) \cdot y_0 + (((x - x_0)(x - x_2)) / ((x_1 - x_0) \cdot (x_1 - x_2))) \cdot y_1 + (((x - x_0) \cdot (x - x_1)) / ((x_2 - x_0) \cdot (x_2 - x_1))) \cdot y_2;$$

где x – рассчитываемое значение пути в м, x_0 – значение пути на один шаг меньше,

x_1 – значение пути на один шаг больше, y – рассчитываемое значение пути в м, y_0 – значение пути на один шаг меньше, y_1 – значение пути на один шаг больше.

В дальнейшем при последующем их использовании и моделировании автомобиля на основе уже скорректированного массива данных применялся метод интерполяции CatMull-Rom [70] по 4-м точкам при фиксированном шаге. Фрагмент разработанной программы на C/C++ приведен ниже.

```
n=u/x/ts;
```

```
t=u-n;
```

```
t2=t*t;
```

```
t3=t*t*t;
```

```
P0=rd[n-1][1];
```

```
P1=rd[n][1];
```

```
P2=rd[n+1][1];
```

```
P3=rd[n+2][1];
```

```
h = 0.5 * ((2*P1) + (-P0 + P2) * t + (2*P0 - 5*P1 + 4*P2 - P3) * t2 + (-P0 + 3*P1 - 3*P2 + P3) * t3),
```

где x – текущее значение пути, n – целостное значение от деления x/ts , u – десятичное значение от деления x/ts , ts – шаг квантования массива (5 м), t , t_2 , t_3 – множители, P_0 , P_1 , P_2 , P_3 – полиномы CatMull-Rom, $rd[]$, h – массив высот продольного профиля дороги.

2.2.6 Результаты сравнения данных разных методов

Ввиду того, что применяются различные методы преобразования и базовые точки отсчета координат в СНГ (Пулковские координаты), а за рубежом (Гринвича и Экватора) предложено оценивать точность метода определяемого местоположения по получаемому расстоянию на 1° широты и долготы. В качестве которого взяты для примера координаты Минска с GPS (5354.97907, 2734.50963). Выбор такого подхода сравнения обусловлен необходимостью упрощения практических расчетов при движении

автомобиля, ВТ и военном деле с использованием карт и при расчете артиллерийской стрельбы. Как показывает обсуждение на геодезических форумах при небольших изменениях относительных расстояний (до 475 км) на большинстве методов на плоскости получаются примерно близкие относительные расстояния с разницей менее 1 м и с уменьшением расстояния точность увеличивается. Для поставленной в работе цели и расстояний до 100 км это приемлемо.

Это подтверждается проведенными в работе исследованиями. Такой подход может быть распространен и на другое местоположение в СНГ, для которого необходимо провести аналогичные расчеты и определить коэффициенты преобразования. Данные по приходящемуся расстоянию на 1° при разных методах приведены в таблице 2.4.

Таблица 2.4. – Приходящееся расстояние долготы и широты на 1° в разных методах

φ	0°	15°	30°	45°	60°	75°	90°
Δlat	110574 м	110649 м	110.852	111.133	111412	111618	111694
Δlon	111.320 м	107551 м	96486 м	78847 м	55800	28902	0.000

Как видно из таблицы полученные результаты при этих методов значительно отличаются. Это связано использованием различных систем базовых точек отсчета в СНГ и за рубежом в картографии, электронных картах и для спутников. Например, в СНГ все еще используются военные карты СК-42 и СК-95, построенные от геодезического пункта с фиксированными наземными и угловыми координатами (Пулковские координаты), а WGS84 (GPS) и ГСК-2011 построены расчетах на эллипсоиде, центр которого совпадает с центром масс Земли. Из-за чего получаемые в них координаты (в градусах и метрах) отличаются. Сейчас в качестве единой государственной системы координат в СНГ для использования при осуществлении геодезических и картографических работ установлена геодезическая система координат ГСК-2011. Она предусматривает использование как пространственных координат ГОСТ 32453-2017 по п. 5.1.1, так и плоскостных по п. 5.4.1.

Как видно из таблицы 2.1 параметры широты на большинстве методов близки в пределах 1 % за исключением по п. 5.1.1 ГОСТ 32453-2017. В отношении долготы параметры более значительно разнятся от 20559,24032 до 191318,22375 м. Учитывая использование еще старых карт СК-42, СК-95 представляется более целесообразным использовать данные параметров, получаемые по формулам п. 5.4.1 ГОСТ 32453-2017, который принят в ОДКБ. На основании проведенных расчетов на территории РБ при определении

относительных расстояний и привязке к картам предложено пользоваться упрощенной линейной формулой $lat_m = 112297,6196 \cdot D_{dec\ lat}$, $lon_m = 63215,444 \cdot D_{dec\ lon}$. Данные коэффициенты получаются при изменении на 1° широты и долготы по формуле п. 5.4.1 ГОСТ 32453-2017 для координат Минска.

Определение пройденного расстояния и продольного профиля дороги, на основе получаемых данных с модуля GPS GN-803G, осуществлялась с помощью мини-компьютера Raspberry Pi 3b+. Регистрация проводилась при движении автомобиля по следующему замкнутому маршруту на дорогах различного качества [14], показанному на рисунке 2.11 и последующей обработки записей программами на C/C++ и Matlab/Simulink на стационарном компьютере.

Проведенные замеры по маршруту в близи Минска, показали пройденное расстояние 42363.7 и 29520.2 м соответственно для географического метода и по скорости при показании спидометра 29,9 км, что свидетельствует о корректности предложенного метода по скорости.

Результаты обработки данных при движении по маршруту сведены в таблицу 2.5.

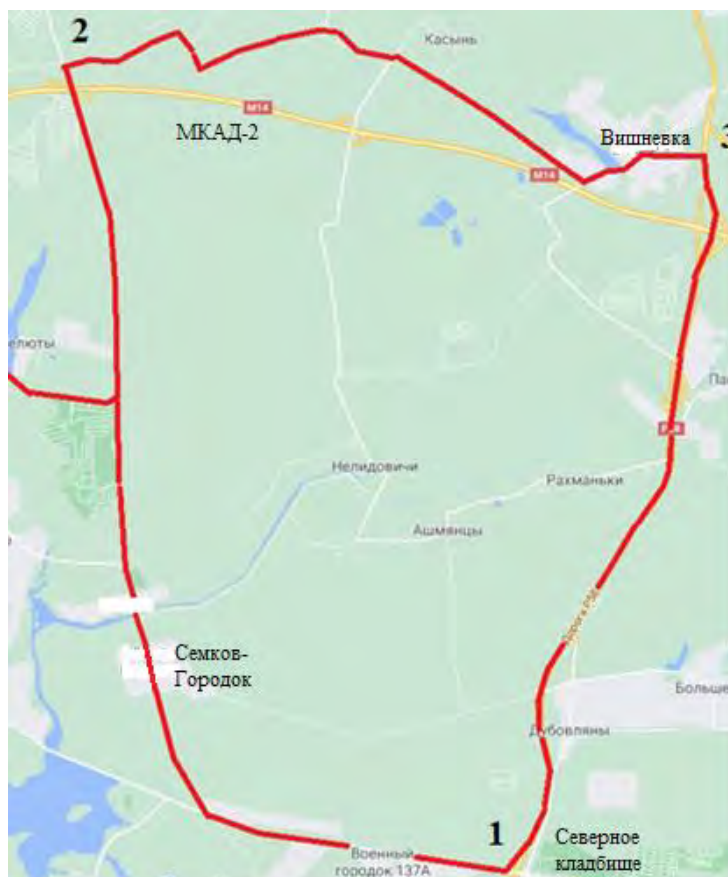


Рисунок 2.11. – Маршрут движения 1–2–3–1

Таблица 2.5. – Сравнение пройденного расстояния при различных методах определения пройденного пути

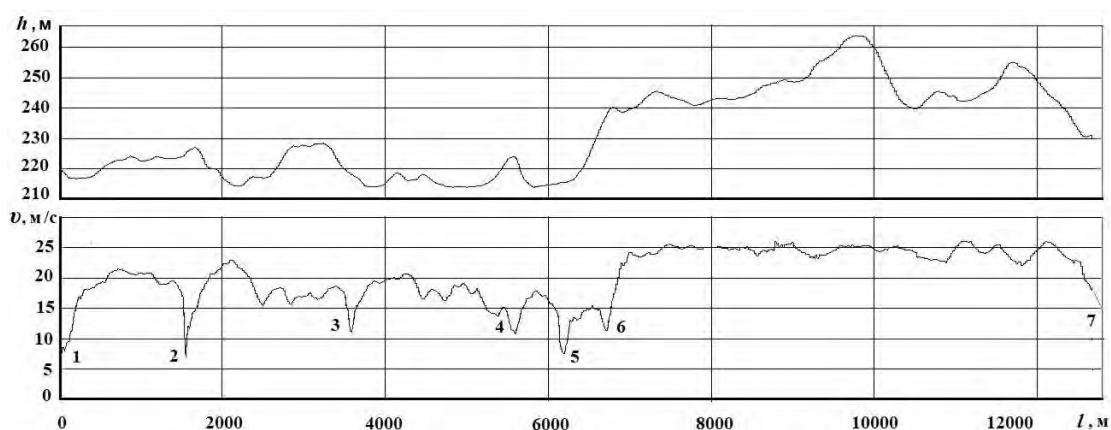
		Параметры точек		Пройденный путь на участках, м				
		Широта	Долгота	Вариант методов обработки				
				1	2	3	4	5
1	Нач	02731.89508	5359.13900	15827	21786	18121	17244	12657
	Кон	02727.16147	5404.20944					
2	Нач	02727.16163	5404.20891	13466	15554	26551	14695	8422
	Кон	02733.93791	5403.64511					
3	Нач	02733.95086	5403.60849	10299	10365	11334	12303	9073
	Кон	02731.83292	5359.12580					
		Суммарный пройденный путь, м		39592	47705	39706	44142	30152

Примечание. 1. Суммарный пройденный путь составляет 30,2 и 31 км соответственно по электронной карте [7] и по электронному спидометру автомобиля.

2. Вариант методов обработки 1,2, 3, 4, 5,6 соответствуют методам 2.1; 2.2; 2.3; 2.4; 2.5.

Из таблицы следует, наиболее лучшее совпадение данных имеет место при оценке пройденного пути при суммировании значений пути исходя из скорости (в м/с) с учетом частоты квантования (1 Гц), получаемой с GPS (расхождение 0,15 %). В остальных вариантах различие составляет 30–47%. Причина такого различия заключается в малом шаге изменения квантования замеров (8–25 м, зависящем от скорости), при которых погрешность позиционирования модуля GPS GN-803G (1–2 м) начинает сказываться. При уменьшении шага (зависящем и от частоты квантования), точность ухудшается. Не случайно на автомобильных навигаторах используется частота 1 Гц, хотя их модули могут работать до 10 Гц.

Результаты свидетельствуют, что при использовании на транспортных средствах, БЛА предпочтителен метод определения пройденного пути по скорости GPS. Для иллюстрации на рисунке 2.12 представлены графики изменения высоты h профиля и скорости движения v в зависимости от пройденного пути l .



1–старт, 2– искусств. препятствие, 3 – поворот, 4 – ограничение v_a , 5– выбоина, 6 – узкий мост

Рисунок 2.12. – Изменение скорости движения автомобиля от пройденного пути l , высоты (h) продольного профиля дороги и дорожной обстановки

Имеющие место экстремумы на графике скорости обусловлены резким изменением скорости при торможении перед переездом препятствий, при повороте дороги, перед мостом и медленно движущим впереди автомобилем на подъеме, проезде населенных пунктов и отражают реальный режим движения. Полученные графики свидетельствуют о необходимости учета этих дорожных факторов при моделировании автомобиля.

По характеру изменения высоты данного участка дороги, их графики совпадают с данными работы автора [14] и по совокупности свидетельствуют о корректности примененного метода. Он может применяться и для систем ГЛОНАСС, Galileo, BeiDou. Полученные данные по дорогам планируется использовать при моделировании движения колесных машин.

Из рисунка 2.12 видно, что скорость движения автомобиля зависит как от профиля дороги, дорожных условий, так возможностей самого автомобиля. На подъемах она замедляется, а на спуске увеличивается. Наблюдаемые экстремумы на графике скорости обусловлены резким изменением скорости при торможении перед переездом препятствий, повороте дороги, перед мостом и медленно движущим впереди автомобилем на подъеме, проезде населенных пунктов и свидетельствуют о корректности примененного метода.

Данные GPS по боковым отклонениям позволяют определить повороты на трассе и могут использоваться для определения и задания ограничивающих точек режима движения. Но все же для окончательного задания режима движения необходимо учитывать еще фактор водителя и дорожную обстановку, которую можно решать с помощью синхронной видеозаписи по предложенному методу, приведенному в главе 7.

Как видно из анализа предложенный метод получения по скорости на основе GPS продольного профиля дороги и режима движения автомобиля, ВТ позволяет более точнее определять их на стадии проектирования, проведения

военных игр, моделирования и поддержки принятия решений действительные средние скорости, пройденный путь и другие технические показатели ТС.

2.3 Получение макро и микропрофиля дорог другими методами

2.3.1 Основываясь на фактах, что датчики ускорений линейно реагируют на небольшие наклоны и при двойном интегрировании ускорений в спектре сигнала присутствует инфранизкочастотная составляющая, характерная для наклона дороги предложены два новых метода получения данных по продольному профилю и микропрофилю дорог [14, 55, 56, 65, 66]:

– **использование данных ускорений и наклонов** на кузове и над задней осью по двум осям X и Y , интегрирования их с экспонентным сглаживанием с коэффициентом K и получения микропрофиля дороги в широкой полосе 0,1–25 Гц, затем выделение полосы сигнала 0,1–0,63 Гц, соответствующей продольному профилю дороги);

– на основе регистрации **продольного угла наклона автомобиля** (по h), учитывая, что

$$h = K_{\pi} \cdot v \cdot \int_0^t \alpha dt$$

где $K_{\pi} = 0,0176$ – коэффициент перевода угла из градусов в радианы; α – угол наклона автомобиля; v – скорость автомобиля.

Получение микропрофиля дороги на движущемся автомобиле по первому методу базируется на следующей схеме колебаний задней части автомобиля в поперечной плоскости (рисунок 2.13).

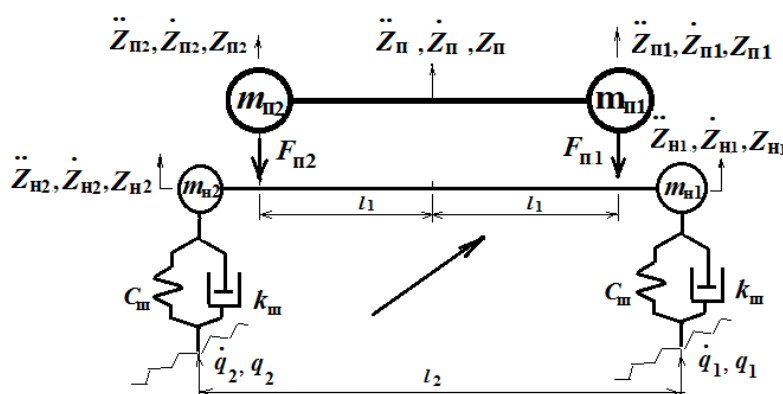


Рисунок 2.13. – Схема колебаний задней части автомобиля в поперечной плоскости

Колебания неподрессоренной задней части автомобиля можно описать следующими дифференциальными уравнениями

$$\begin{aligned}\dot{q}_1 &= \frac{1}{k_{ш}} (0,5m_n \ddot{Z}_n - C_{ш} \cdot (q_1 - Z_{n1}) - k_{ш} \dot{Z}_{n1} - K_1 (F_1 + F_2)) ; \\ \dot{q}_2 &= \frac{1}{k_{ш}} (0,5m_n \ddot{Z}_n - C_{ш} \cdot (q_2 - Z_{n2}) - k_{ш} \dot{Z}_{n2} - K_1 (F_1 + F_2)) ; \\ F_1 &= 0,5m_{п} \ddot{Z}_{п} + K_1 l_1 a_{п} ; \quad F_2 = 0,5m_{п} \ddot{Z}_{п} - K_1 l_1 a_{п} ; \\ \dot{Z}_{п1} &= \dot{Z}_{п} + l_1 \dot{a}_{п} ; \quad \dot{Z}_{п2} = \dot{Z}_{п} - l_1 \dot{a}_{п} ; \\ \dot{Z}_{н2} &= \dot{Z}_{н1} - l_1 \dot{a}_{н} ; \quad Z_{н2} = Z_{н1} - l_2 a_{н} ; \quad K_1 = 2l_1 / l_2 ;\end{aligned}$$

где $m_{п}$, m_n , $\ddot{Z}_{п}$, $\dot{Z}_{п}$, $Z_{п}$, $\ddot{Z}_{н}$, $\dot{Z}_{н}$, $Z_{н}$ – масса, ускорение, виброскорость, перемещение поддрессоренных и неподдрессоренных частей задней части автомобиля; $F_{п1}$, $F_{п2}$ – усилие в подвеске; $C_{ш}$, $k_{ш}$ – жесткость и коэффициент демпфирования шины, l_1 и l_2 – размеры.

Решая эти уравнения в пакете Matlab/Simulink с использованием записей ускорений и углов наклона автомобиля можно получить продольный профиль и микропрофиль дороги.

Для проведения исследований (рисунок 2.14) использовалась следующая аппаратура: микрокомпьютер Raspberry Pi 3, два датчика MPU-6050 (погрешность 0,2 %).



Рисунок 2.14. –Схема записи и обработки параметров дороги

Методика измерений заключалась в замерах и регистрации ускорений и углов наклона на кузове и над задней осью по двум осям X и Y , получении $\arctg$ углового наклона, двойной фильтрации сигналов и записи их на SD-карту [55, 56, 72]. Запись параметров производилась при постоянной скорости движения 72 км/ч (20 м/с) в раннее утреннее время, когда менее загружены трассы. Выбор скоростей движения определялся исходя из аналогичных соображений метода IRI (80 км/ч) по оценке ровности дорог – несоздания помех движению.

Используя текущие значения ускорений и данных гироскопа по указанным осям с помощью функции $\arctg(\alpha) = \frac{x}{\sqrt{x^2+z^2}}$ в мини-компьютере определялось значение ускорения углового поворота кузова автомобиля, которое корректировалось на основе данных гироскопа.

Затем путем его фильтрации фильтром Butterworth (40 дБ, в полосе до 0.7 Гц) и интегрирования в пакете Mathlab/Simulink определялся угол наклона кузова, который принимался за угол наклона дороги. Данный сигнал в дальнейшем подвергался дифференцированию для получения скорости высоты профиля дороги, необходимой для моделирования тяговой динамики. Для устранения шумов и дрейфа сигнала датчиков MPU-6050 использовались фильтрация внутри датчика и экспонентное сглаживание сигнала в Raspberry. Регистрация данных с датчика производилась на SD-карту в виде структуры бинарного файла. Далее с помощью программы на языке C в блоке модуля S-Function Builder (приложение В) они заносились в электронный массив данных MatLab/SimuLink и здесь еще раз фильтровались фильтром Butterworth. Блок-схема получения продольного и микропрофиля дороги приведена на рисунке 2.15. С помощью константы “Скорость/Speed” и интегратора “Integrator” задается скорость движения (считывания массива данных ускорений или углов уклona).

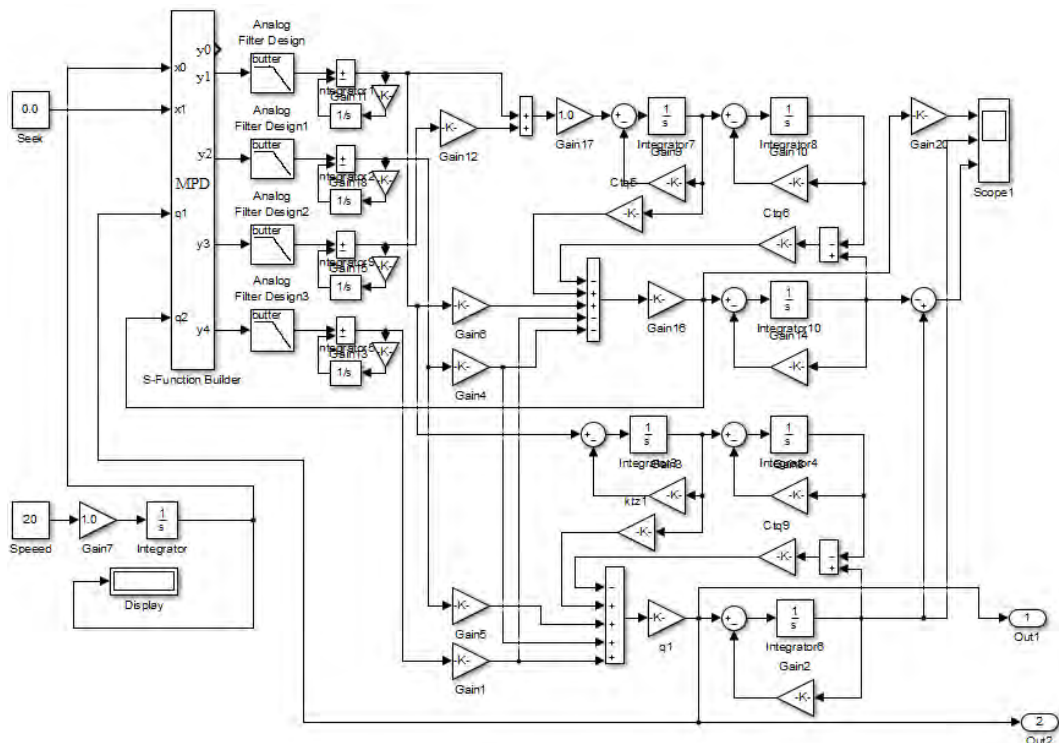


Рисунок 2.15. – Блок-схема получения и обработки высоты и углов наклона дороги

2.3.2 Получение макро и микропрофиля дороги через тензометрирование усилий и ускорений на балке моста

Имеется еще один метод получения макро и микропрофиля дороги через тензометрирование усилий и ускорений на балке моста, который был опробован автором в 1983 г. на МАЗе (рисунок 2.16) с осуществлением обработки данных на аналоговой машине.

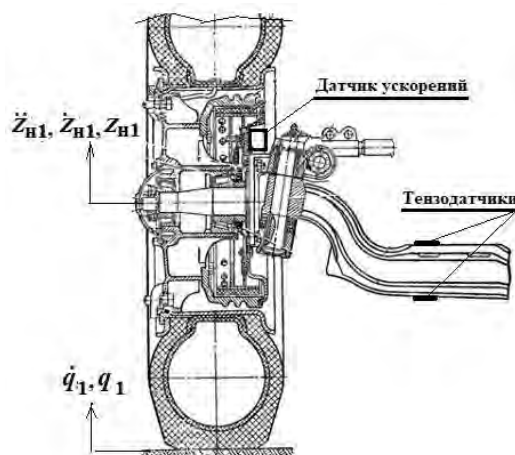


Рисунок 2.16. – Регистрация профилей дорог путем тензометрирования усилий и ускорений на балке переднего моста

Он требует использования датчика ускорений с жидкостным демпфированием и наклеенных двух тензодатчиков по мостовой схеме для фиксации только изгибающих усилий и регистрирующей аппаратуры.

Микропрофиль дороги определяется следующим дифференциальным уравнением,

$$\dot{q}_1 = \frac{1}{k_{ш}} (m_{н} \ddot{z}_{н} - C_{ш} \cdot q_1 + C_{ш} \cdot q_{н} - k_{ш} \dot{z}_{н1}) ;$$

где $m_{н}$, $\ddot{z}_{н}$, $\dot{z}_{н1}$, $z_{н}$ – соответственно вертикальные ускорения, скорость, перемещение неподрессоренной массы балки (ее оси), q_1 , $\dot{q}_1$, $q_{н}$ – соответственно вертикальные ускорения, скорость, высота профиля дороги под шиной.

Блок-схема получения микропрофиля дороги приведена на рисунке 2.17.

В зависимости от параметров экспоненциального сглаживания интеграторов может быть получен либо совмещенный макро и микропрофиль дороги или только один микропрофиль. Во втором случае можно наложить микропрофиль на макропрофиль, полученный с помощью GPS.

Если наклеить тензодатчики в продольной плоскости, то можно еще получить зависимость сопротивления качения от микропрофиля дороги.

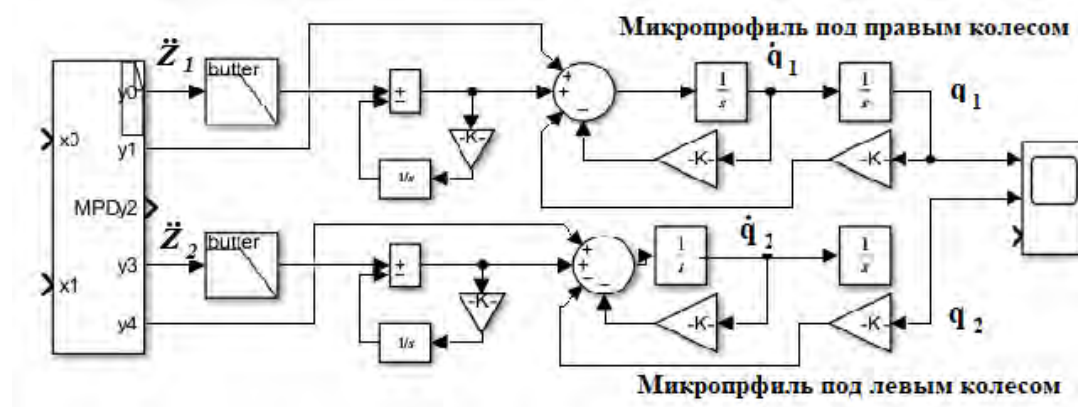


Рисунок 2.17. – Блок-схема для получения микропрофиля дороги

Для обработки данных требуется использование Simulink с модулем S-Function Builder. Последний используется для ввода данных с датчиков, а процесс интегрирования осуществляется в Simulink.

2.3.3 Возможен вариант получения макро и микропрофиля дороги на основе смартфона, учитывая, что современные смартфоны имеют в своем составе 3-х осевой датчик ускорений, который можно использовать для регистрации макро и микропрофиля дороги. Для этого требуется установить бесплатную программу AndroSensor.apk на смартфон, указать в ней необходимые параметры записи [67]. Экспериментальные данные ускорений по трем осям на смартфоне записываются в формате CSV с разделителями. Эти данные необходимо программным путем затем преобразовать в формат float. Рассматривая и моделируя заднюю часть автомобиля в виде двухмассовой системы с использованием ускорений поддрессоренной части в виде внешнего возмущения, можно определить скорость высоты макро и микропрофиля дороги решая следующее дифференциальное уравнение

$$\dot{q}_d = \frac{1}{k_{ш}} (m_{п} \ddot{Z}_{п} - C_{ш} \cdot q_1 + C_{ш} \cdot q_n - k_{ам} \dot{Z}_n) .$$

Интегрируя $\dot{q}_d$ можно получить значение высоты макро и микропрофиля дороги. Это можно сделать в программе на языке C в S-Function Builder Simulink аналогично блок-схеме верхнего ряда усилителей и интеграторов (рисунок 2.3), подобрав время задержки, соответствующее шагу 0,125 м и перезаписать их в бинарный массив файла. Недостатком такого метода является невысокая точность метода (до 15 %) из-за упрощения модели, точности датчика смартфона и возможность регистрации только одной стороны колеи либо обобщенной по центру оси X.

2.3.4 Специалистами фирмы Volvo предложен интересный метод по получению уклонов дорог [84–86]. Он основывается на регистрации параметров CAN-шины электронного блока управления силовым агрегатом движущегося автомобиля и положения автомобиля на трассе с помощью датчика GPS. Полученные данные (скорость, крутящий момент, текущая передача КПП, координаты точек сканирования GPS) записывались на laptop. Регистрация параметров производилась с шагом 12,5 м. Длина записанного участка составляла 15 км. Число точек сканирования 1200. Затем на основе обработки этих данных рассчитывался угол наклона дороги, с использованием тягового баланса сил автомобиля и значения его массы без учета аэродинамического фактора. В дальнейшем полученный массив данных используется при расчетах тяговой динамики и оптимизации параметров автомобилей этой фирмы. Вместо laptop вполне можно использовать более дешевые Arduino Due или Raspberry.

2.3.5 Формирование сигнала микропрофиля дороги по точкам

Для формирования сигнала микропрофиля дороги на основе массивов замеров разработаны программы на языке C/C++ для S-Function Builder Simulink. Последние считывают данные замеров микропрофиля дороги с фиксированным шагом (0,125 м) для левой и правой колеи и заносят их в бинарный массив памяти. Затем в процессе обращения к S-Function по значению длины дороги x и этого массива вычисляется значение высоты неровности методами интерполяции по ближайшим точкам к значению x длины дороги.

Откомпилированная программа на языке C представляет собой разновидность DLL. Она работает на порядок быстрее чем на языке Matlab. При использовании 64-битного компилятора языка C, начиная с версии Matlab/Simulink 2015b, информация по дороге может заноситься в виде большого массива float rd [48000][2] по двум колеям (48 тысяч записей) либо 96000 тысяч записей по одной (float rd [96000]) и при необходимости реально реализуемый объем может быть увеличен за счет подкачки информации.

Как показал анализ и исследования автора [14, 55, 56], более рациональным все же является применение объединенного макро и микропрофиля дороги. В данном случае использовалось суммирование, полученного в работах [55, 66] микропрофиля с помощью микропроцессорных систем на движущемся автомобиле и продольного профиля дороги, полученного с помощью GPS. Интерполяции промежуточных значений

рассчитывалась по методу **Catmull-Rom** по четырем точкам, предложенному на фирме Ситроен [70].

Сформированный таким путем объединенный макро и микропрофиль дороги использовался при проведении совместного моделирования движения и колебаний автомобиля.

2.4 Моделирование воздействия работы двигателя на автомобиль

2.4.1 Необходимость рассмотрения работы двигателя (дизеля V8) обусловлена предположением, что он является источником возмущения на трансмиссию и автомобиль в области 8–31 Гц продольных и поперечных вибраций из-за совпадения диапазонов частот, а также требованиями ISO 2631 проводить оценку вибраций не только в вертикальном, но и в продольном и поперечном направлении. Эти вопросы изучены слабо.

По данным МАЗа [81], в которых участвовал автор, уровни продольных и поперечных вибраций в 3–4 раза меньше вертикальных, но они все же вносят свою лепту (15–25%) по приведенным вибрациям по 3-м плоскостям согласно ISO-2631-2018, а также влияют на критерии комфорта.

И если с вертикальными колебаниями автомобиля все ясно: источником их является микропрофиль дороги и путем моделирования можно обеспечить совпадение с экспериментом 5–15 % с учетом спектра [99], то механизм продольных и поперечных вибраций в полосе 8–31 Гц не изучен,

Неясно как моделировать продольные и поперечные вибрации. В работах [29, 30] их предлагалось рассматривать как воздействие микропрофиля дороги через $\tan$ угла наклона по аналогии с вертикальными. Но это подходит только для переезда больших по высоте препятствий ($h > 100$ мм) и не подходит для реальной дороги.

Анализ работ по динамике двигателей [90–92] свидетельствует, что практически полностью отсутствуют работы по оценке воздействия двигателя на вибрации грузового автомобиля и трансмиссию.

Согласно этих работ динамика двигателя имеет сложный характер из-за наполняемости цилиндров, наличия зазоров, трения, взрывного процесса сгорания топлива. Но вопросы совместного моделирования двигателя, трансмиссии и подвески пока еще не решены из-за сложности их процессов. Нет их объединенной модели, особенно модели работы двигателя при переменных оборотах и нагрузке.

Сейчас с внедрением специализированных пакетов сделаны некоторые первые шаги в этом направлении. Так в работе [109] предлагалось рассматривать эти вибрации как результат совместной работы двигателя 4L и трансмиссии путем реализации следующей схемы (рисунок 2.18).

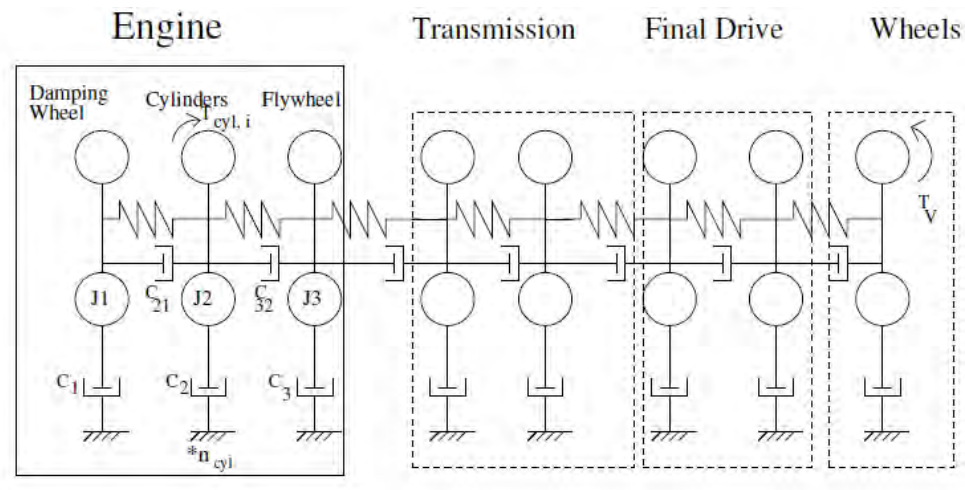


Рисунок 2.18. – Динамическая модель двигателя, трансмиссии с приводом на колеса

Недостатком работы является отсутствие математической модели работы двигателя, его спектра возмущения и параметров модели.

В работе [110] сделана попытка смоделировать процесс на специализированном пакете AVL/EXCITE (рисунок 2.19) путем создания КЭМ и его динамической модели 4-х тактного рядного двигателя без математического описания, аналогично технологии MS.ADAMS.

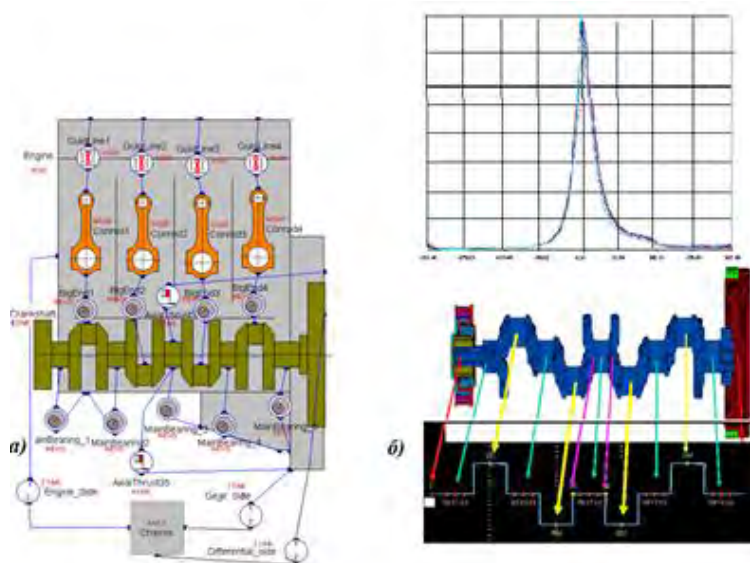


Рисунок 2.19. – Созданная в пакете AVL/EXCITE модель двигателя, трансмиссии и автомобиля (1), и изменение давления в цилиндре при работе двигателя (б)

И на основании сформированных связей в пакете и передаточных функций проведена оценка спектра колебаний без связи с дорогой (с ее макро и микропрофилем). Полезными являются график изменения давления в цилиндре при сгорании топлива от угла поворота коленвала и подход, который

может быть применен в MS.ADAMS в сочетании с моделью, созданной в 3D пакете. Но неясно как реализовать это при переменных оборотах ДВС при движении по реальной дороге.

Обобщая отметим, что оценка вибраций в продольном и поперечном направлении при моделировании как у нас в СНГ, так и за рубежом по открытым источникам пока серьезно не проводится, за исключением экспериментальных исследований [100, 108, 109]. Причиной этого – отсутствие моделей и неясность как реализовать источник возмущения с меняющимися параметрами. Хотя можно предположить, что на зарубежных автомобильных фирмах этот вопрос решен, но материалы по нему не публикуют. Из-за чего необходимо проведение собственных исследований и оценивать вибрации в совокупности по всем направлениям.

2.4.2 Источники вибраций и ситуация с вибронегруженностью

Как показывает проработка вопроса по этому направлению источниками вибраций в этой области частот могут быть:

- неуровновешенностью самого двигателя и колебания его на резино-металлических опорах;
- неравномерная работа двигателя, связанная со взрывным характером сгорания топлива в цилиндрах,
- колебательные процессы в трансмиссии.

Вопрос вибронегруженности особенно актуален для автомобилей повышенной проходимости, имеющих высокий уровень вибраций в условиях движения по разбитой гравийной дороге (рисунок 2.20).

Как видно из графика на раме над задними мостами основной уровень вибраций на разбитом гравийном шоссе приходится на третьоктавы 12,5–25 Гц и достигает по приведенным среднеквадратическим вертикальным ускорениям значений $2,45 \text{ м/с}^2$. Последние меньше действительных. Это связано с жесткой задней подвеской и отсутствием амортизаторов в ней. С учетом продольных и поперечных вибраций общий уровень получается выше. При моделировании движения по асфальтовым дорогам уровень приведенных вертикальных вибраций на сиденье составляет соответственно 1,07 и 1,2 м/с^2 на дорогах хорошего и среднего качества при средних скоростях 78 и 66 км/ч. По ISO это соответствует ощущению дискомфорта.

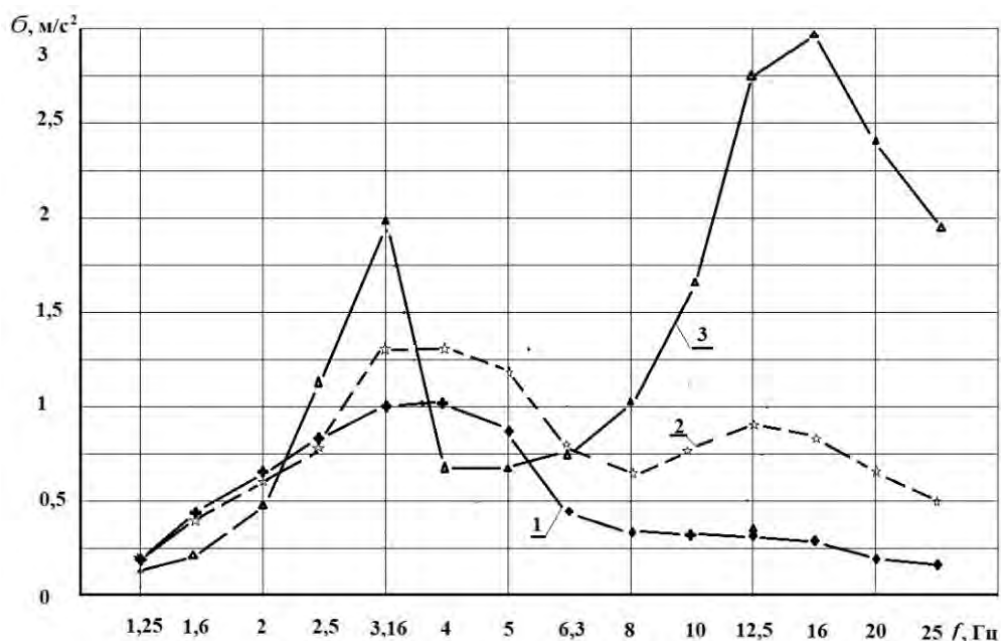


Рисунок 2.20. – Изменение уровней вибраций МАЗ-6317 в третьоктавных полосах частот на сиденье (1), на раме над передним мостом (2), на раме над осью балансира задней подвески (3) при моделировании при скорости движения $v=40$ км/ч по разбитому гравийному шоссе

Следует отметить, что даже у легковых зарубежных автомобилей по данным работы [100] наблюдается высокий уровень приведенных вибраций, замеренных экспериментально. Так, уровень вибраций зарубежных легковых автомобилей при средней скорости 50 км/ч при смешанных поездках составляет 1–1,4 м/с². К сожалению, не указывается какова доля в них продольных и поперечных вибраций и свидетельствует о их неизученности.

Все это требует проведения серьезных экспериментальных исследований, что доступно только автозаводам, располагающих возможностями и оборудованием. Из-за их отсутствия попробуем решить этот вопрос путем моделирования.

2.5 Работа и вибрации восьмицилиндрового V-образного двигателя

2.5.1 Рассмотрим работу восьмицилиндрового V-образного двигателя, которые наиболее чаще применяются на грузовых автомобилях. Примером является двигатель ЯМЗ-7511, применяемый на автомобилях МАЗ с порядком работы, приведенной в таблице 3.5.

Угол между рядами цилиндров составляет $\gamma = 90^\circ$; коленчатый вал несимметричный и имеет кривошипы, расположенные в двух взаимно перпендикулярных плоскостях с углом 90° .

ЯМЗ-7511 работает с очередностью 1-5-4-2-6-3-7-8, интервал

последовательности вспышек - 90 градусов (рисунок 2.21).

Таблица 3.5. – Порядок работы восьмицилиндрового V-образного двигателя ЯМЗ-7511

Полуобороты коленвала	Углы поворота коленвала	Цилиндры							
		1	2	3	4	5	6	7	8
Первый	0	Раб.ход	Впуск	Выпуск	Сжатие	Сжатие	Впуск	Выпуск	Раб.ход
	90	Сжатие	Впуск	Выпуск	Раб.ход	Сжатие	Впуск	Выпуск	Сжатие
Второй	180	Выпуск	Сжатие	Впуск	Выпуск	Сжатие	Впуск	Выпуск	Сжатие
	270	Сжатие	Впуск	Выпуск	Раб.ход	Сжатие	Впуск	Выпуск	Сжатие
Третий	360	Впуск	Сжатие	Выпуск	Впуск	Сжатие	Впуск	Выпуск	Сжатие
	450	Сжатие	Впуск	Выпуск	Раб.ход	Сжатие	Впуск	Выпуск	Сжатие
Четвертый	540	Выпуск	Сжатие	Впуск	Выпуск	Сжатие	Впуск	Выпуск	Сжатие
	630	Сжатие	Впуск	Выпуск	Раб.ход	Сжатие	Впуск	Выпуск	Сжатие
	720	Впуск	Сжатие	Выпуск	Впуск	Сжатие	Впуск	Выпуск	Сжатие

Так, если в первом цилиндре происходит рабочий цикл, то через 90° поворота коленвала, рабочий цикл будет уже в 5 цилиндре. Через следующие 90° поворота коленвала, рабочий цикл будет уже в 4 цилиндре и т.д. Для полного оборота коленвала нужно $(360/90)$ 4 рабочих хода. Шатунные шейки коленвала располагаются под углом 90° т. е. крестообразно. Такое чередование способствует его более равномерному вращению.

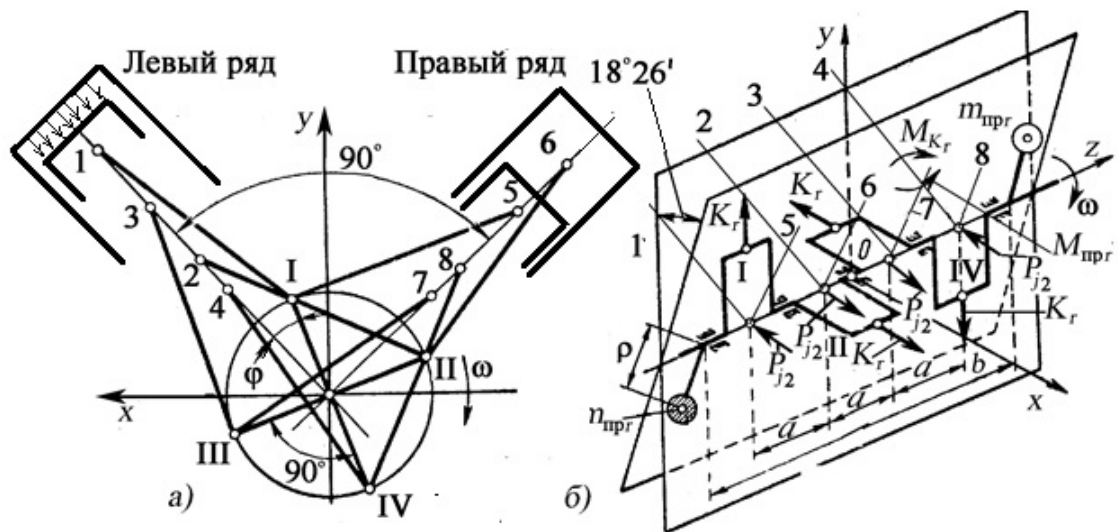


Рисунок 2.21. – Порядок работы цилиндров V8-образного двигателя

2.5.2 Силы, действующие в кривошипно-шатунном механизме (КШМ)

Схема КШМ, рассматриваемого ДВС, показана на рисунке 2.22.

Для удобства анализа будем рассматривать двигатель ЯМЗ-7511, повернутый на 45° вправо (рисунок 2.22, б) в виде четырех условно «состыкованных» 2-х цилиндровых V-образных двигателей (отсеков).



a – схема двигателя; *б* – уравнивание момента центробежных сил и первого порядка

**Рисунок 2.22. – Восьмицилиндровый V-образный двигатель
с углом развала цилиндров 90°**

На рисунке 3.47*б* цифрами 1...4 обозначен левый ряд, 5...8 – правый ряд цилиндров, а I – IV – номера кривошипов.

На каждом кривошипе, возникает действующая в плоскости кривошипа по его радиусу сила

$$K_r = (m_r + m_j) r \omega^2$$

представляющая собой сумму центробежной силы вращающихся неуравновешенных масс кривошипа $m_r = (m_k + 2m_2)$ (на одной шейке располагается два шатуна, часть массы которых m_2 относится к оси шатунной шейки) и равнодействующие сил инерции первого порядка возвратно-поступательно движущихся масс $m_j = 2m_{\pi} + 2m_1$:

где m_{π} – масса поршневого комплекта, включающая массы собственно поршня, поршневых колец, поршневого пальца и заглушек; m_1 – часть массы шатуна, отнесенная к оси поршневого пальца.

Равнодействующая R_{F_r} четырех сил K_r равна нулю поскольку углы между кривошипами, вдоль которых действуют эти силы, равны 90°.

В этом случае моменты от каждой пары шатунов, расположенных на одной шатунной шейке, не учитываются. Равнодействующая R_K четырех сил $K_r = (m_r + m_j)r\omega^2$ равна нулю, но ввиду несимметричности вала момент M_{K_T} этих сил не равен нулю. Величину этого момента M_{K_T} можно определить, найдя моменты сил K_r , действующих как пары сил в плоскостях кривошипов

I и IV и II и III соответственно (в горизонтальной и вертикальной плоскостях). Момент сил в кривошипах I и IV равен $\bar{M}_{K_r}^{I,IV} = -3K_r\alpha$ (вектор $\bar{M}_{K_r}^{I,IV}$ расположен в горизонтальной плоскости), а в кривошипах II и III ($\bar{M}_{K_r}^{II,III} = K_r\alpha$ располагается в вертикальной плоскости).

Результирующий момент M_{K_r} (рисунок 2.23,а) получается геометрическим сложением моментов $\bar{M}_{K_r}^{I,IV}$ и $\bar{M}_{K_r}^{II,III}$, или $M_{K_r} = \sqrt{(-3K_r\alpha)^2 + (K_r\alpha)^2} = \sqrt{10} \alpha K_r = 3,162 \alpha K_r$.

Плоскость действия результирующего момента M_k определяется углом α , который она составляет с плоскостью первого кривошипа: $\operatorname{tg} \alpha = \frac{\alpha K_r}{3\alpha K_r} = \frac{1}{3}$, откуда $\alpha = 18^\circ 26'$.

Этот момент может быть уравновешен установкой противовесов на продолжении щек каждого кривошипа либо двух внешних противовесов, установленных на хвостовике и носке коленчатого вала в плоскости действия момента M_{K_r} (см. рисунок 2.23 б).

В последнем случае масса каждого внешнего противовеса определится из условия равенства моментов M_{K_r} и M_r ;

$$m_{\text{пр}} \rho r \omega^2 b = \sqrt{10} \alpha K_r,$$

$$\text{Откуда } m_{\text{пр}} = 3,162 \frac{a}{b} \frac{K_r}{\rho \omega^2} = 3,162 \frac{a}{b} \frac{r}{\rho} (m_r + m_j),$$

где m_r и m_j имеют то же значение, что и в зависимости $K_r = (m_r + m_j) r \omega^2$.

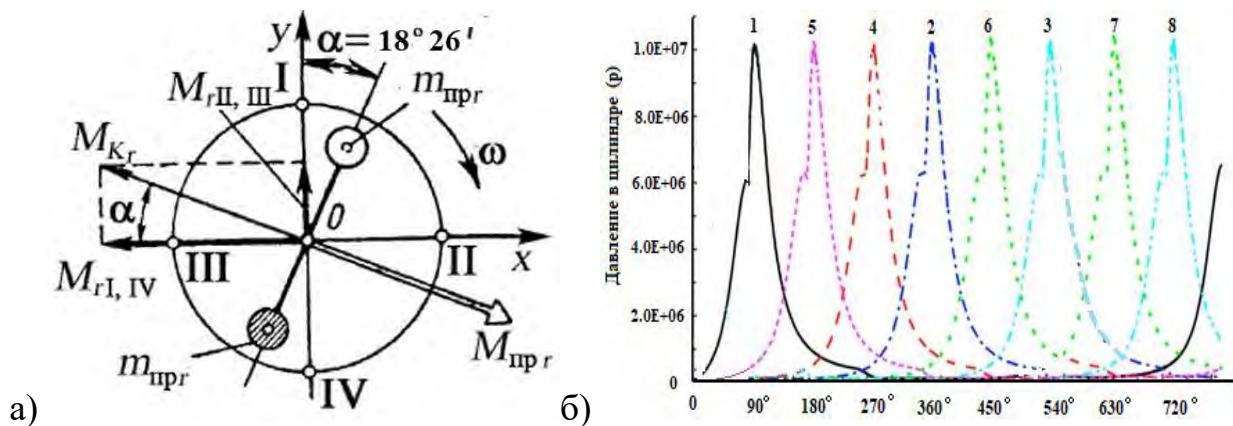


Рисунок 2.23. – Определение результирующего вектора M_R (а), изменение давления в цилиндрах (б)

Равнодействующая сил инерции второго порядка R_{j2} для каждой пары цилиндров 1 и 5; 2 и 6; 3 и 7; 4 и 8 в соответствии с расположением кривошипов могут быть определены как

$$R_{j2} = \sum P_{j2} = -\sqrt{2} \lambda C \cos 2\varphi - \sqrt{2} \lambda C \cos 2(90^\circ + \varphi) - \sqrt{2} \lambda C \cos 2(270^\circ + \varphi) - \sqrt{2} \lambda C \cos 2(180^\circ + \varphi) = -\sqrt{2} \lambda C (\cos 2\varphi - \cos 2\varphi - \cos 2\varphi + \cos 2\varphi) = 0.$$

Все эти параллельные силы лежат в горизонтальной плоскости, проходящей через ось вала, и всегда равны по величине, но попарно противоположны по знаку (см. рисунок 3.47, б). Следовательно, $R_{j2} = 0$.

Моменты сил R_{j2} попарно кривошипов I и IV, а также II и III относительно центра вала (начала координат 0) также создают взаимно уравновешенные моменты. Поэтому результирующий момент сил инерции второго порядка $\sum M_{j2} = 0$.

На основании вышеизложенного можно сделать вывод, что после установки нащечных противовесов для уравновешивания момента от сил инерции 1-го порядка и центробежного момента $\sum M_2$, двигатель V8 с углом развала цилиндров 90° полностью уравновешен от действия всех инерционных сил и моментов. И основную причину вибраций следует искать в неравномерной работе двигателя от действия взрывного характера сгорания топлива.

2.5.3 Для реализации возмущения от действия взрывного сгорания топлива двигателя аппроксимируем индикаторную диаграмму в виде синусоиды (рисунок 2.23,б) по передаваемому от нее усилию F_k , на трансмиссию и колеса автомобиля. Абсолютное значение синусоиды от угла поворота коленвала φ (уг) передаваемого от двигателя в реализуемой программе на языке C/C++ можно описать следующей зависимостью

$$vs = f_{abs}(\sin(2 * \varphi + M_{PI}/2)).$$

Ниже приведен фрагмент программы на C/C++ в модуле S-Function Builder Simulink, реализующий это.

```
//-----Моделирование воздействия двигателя
if (vt >= 720)
{
vt = 0.0;
nlc = 86400 / ndvs; // nlc = 720.0 * 120 / ndvs;
stc = 720.0 / nlc; // кол-во точек на цикл 90 град
vt = stc / tm;
}
```



```

if (vt<720.0) //Выполнение цикла
{
yg=(M_PI*vt)/180.0;
vs=(sin(2*yg+M_PI/2));
fka=1.41*vs*Fk;
if (fka>0.0)
*y4=fka;
else
*y4=-fka;
}
vt++;

```

где vt – счетчик циклов коленвала; stc – шаг моделирования; n/c – число колебаний в цикле, yg – угол поворота; $ndvs$ – обороты двигателя, vs – значение синусоиды в зависимости от угла yg , Fk – меняющееся усилие тяги двигателя с учетом передаточного числа трансмиссии.

Определенную сложность представляет вычисление при меняющихся оборотах двигателя поскольку автомобиль в реальных условиях движется с переменной скоростью и перевод их в углы φ . Текущие мгновенные обороты двигателя $ndvs$ можно определить из моделирования движения автомобиля [14, 99] и их пересчитать в углы поворота коленвала по следующим формулам,

$$\omega = n/60,$$

$$\varphi = n/720.$$

Величину F_k , являющиеся функцией φ , предложено моделировать в модуле S-Function Builder как повторяющийся цикл 720° (два оборота коленвала) с учетом циклов моделирования Simulink, его шага со счетчиком с переменным значением шага $ch=(n/60) t_m$.

где ch – текущее значение счетчика (целая величина), n – текущие обороты двигателя, t_m – шаг моделирования Simulink.

При шаге моделирования $t_m=0,0005$ с процесс цикла 720° при оборотах $n=1200^{-1}$ разбивается на 80 значений (шагов), а при $n=1800^{-1}$ на ~ 53 . Этого достаточно для моделирования. При достижении конечного значения ch счетчик обнуляется, вычисляется новое значение исходя из текущего значения оборотов n и процесс повторяется.

И моделируя таким образом получаем источник воздействия двигателя на трансмиссию и автомобиль. Значение продольных вибраций определяется как $a_{\text{продол}} = 1.41 * vs * F_k / m_a$.

Примечание. F_k – представляет меняющиеся значение силы тяги, определяемого при моделировании движения автомобиля.

2.5.3 Передача вибраций на автомобиль от двигателя

Пульсации двигателя воздействуют на трансмиссию и шины, вызывая колебания в продольной и поперечной плоскости автомобиля (рисунок 2.24).

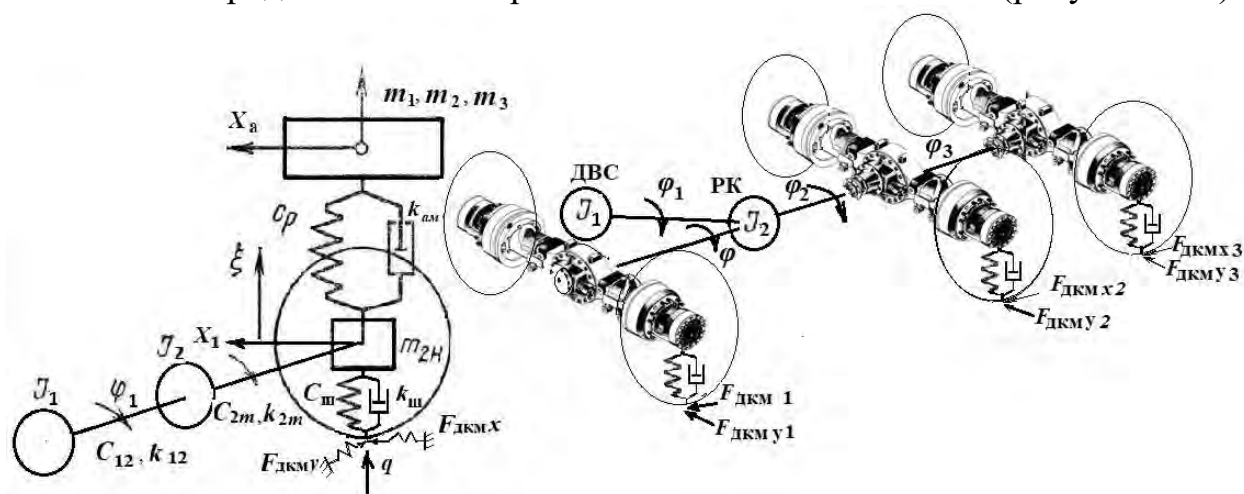


Рисунок 2.24. – Схема передачи динамических моментов двигателя, вызывающих продольные и поперечные колебания автомобиля

Для реализации моделирования воздействия двигателя могут быть использованы две следующие схемы, представленные на рисунке 2.25.

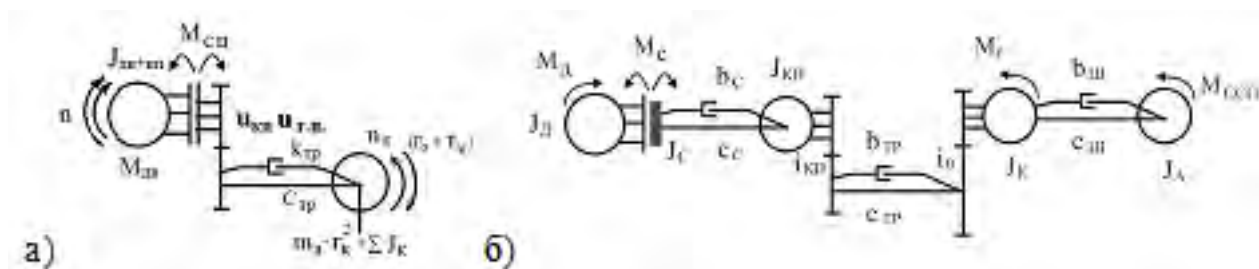


Рисунок 2.25. – Расчетная схема динамической схемы трансмиссии

В случае первой (рисунок 3.50а) более упрощенной учитывается суммарный момент инерции двигателя, сцепления и КПП, податливости и демпфирование трансмиссии, момент колес и автомобиля. Во втором случае (рисунок 2.25 б) дополнительно податливости и демпфирование сцепления и шин, позволяющие определить крутящие моменты в трансмиссии при переключении передач.

Возможен также вариант использования при моделировании колебательной схемы трансмиссии (см. рисунок 3.43) с применением в качестве возмущения крутящего момента двигателя, формируемого аналогично описанному выше подходу.

Предложенный подход формирования воздействия двигателя является новым, позволяющим вести моделирование моментов и нагрузок ДВС при переменной скорости движения автомобиля и усилий, развиваемых двигателем. Он не тормозит моделирование в Simulink, позволяя реализовать в нем режим “реального” времени.

Учитывая, что основной задачей имитационного моделирования является определение нагрузочного режима в процессе движения ограничимся и воспользуемся первой схемой, которую можно описать следующими уравнениями

$$J_{\text{двсц}} \cdot \ddot{\psi} = M_{\text{дв}} - m_a \cdot r_k^2 - (F_k - F_B - F_{\psi} - F_j) \cdot r_k$$

где F_k – сила тяги $F_k = \frac{M_{\text{дв}} \cdot u_{\text{кп}} \cdot u_{\text{вм}} \cdot \eta_T}{r_k}$; $F_B = K_B \cdot A_a \cdot v_a^2$; $M_{\text{дв}}$ — от текущих

крутящих моментов двигателя внутреннего сгорания (ДВС), Н·м; η_T – КПД трансмиссии на соответствующей передаче КП; r_k – радиус качения ведущих колес автомобиля, м; $u_{\text{кп}}$ – текущее значение передаточного числа КП; $u_{\text{вм}}$ – передаточное число главной передачи ведущего моста; F_B – сила сопротивления воздуха; K_B – коэффициент обтекаемости автомобиля, Н·с²/м⁴; A_a — лобовая площадь автомобиля, м²,

$v_a = \frac{\pi}{30} \cdot \frac{n \cdot r_k}{u_{\text{кп}} \cdot u_{\text{вм}}}$ – текущая скорость движения автомобиля, м/с, n – текущая частота вращения коленчатого вала ДВС, мин⁻¹; $F_{\psi} = 9,81 \cdot \psi \cdot m_a$ – сила дорожного сопротивления; где m_a – полная масса автомобиля, кг; $\psi = f \cos \alpha + \sin \alpha$ – коэффициент дорожного сопротивления, α – угол наклона текущего элементарного участка маршрута, f – коэффициент сопротивления качению автомобиля как одной материальной точки, $f_{\text{ш}}$ – коэффициент сопротивления качению шин; $F_j = m_a \cdot a \cdot \delta$ – сила сопротивления разгону; где a – ускорение автопоезда м/с²,

$\delta = \frac{J_{\text{двсц}} \cdot \eta_T \cdot (u_{\text{кп}} \cdot u_{\text{вм}})^2}{m_a \cdot r_k^2} + \frac{\sum J_k}{m_k \cdot r_k^2}$ — коэффициент учета вращающихся масс,

$J_{\text{двсц}}$ – момент инерции вращающихся масс двигателя и сцепления, кг·м²,

J_k – суммарный момент инерции колес автомобиля и приводных валов, кг·м².

2.5.4 О дополнительном источнике продольных и поперечных ускорений

Дополнительным источников продольных и поперечных ускорений

автомобиля также может быть воздействие вертикальных сил на мост в поперечно-угловой плоскости, вызывающих его поворот и воздействие на трансмиссию согласно схемы на рисунке 2.24. Согласно ей математическую модель вертикальных и продольных колебаний можно описать следующими дифференциальными уравнениями

$$m_a \ddot{Z}_0 - (F_{п1} + F_{п2}) + (F_{пcab1} + F_{пcab2}) = 0;$$

$$m_{cab} \ddot{Z}_{cab} - (F_{пcab1} + F_{пcab2} + F_{псид}) = 0;$$

$$m_a \rho_{y1}^2 \ddot{\beta}_1 + (F_{п1} L_1 - F_{п2} L_2) - (F_{пcab1} (L_1 + L_4) + F_{пcab2} (L_1 + L_5)) = 0;$$

$$m_{cab} \rho_{y1}^2 \ddot{\beta}_2 + (F_{пcab1} (L_1 + L_4) + F_{пcab2} (L_1 + L_5)) = 0;$$

$$m_{н1} \ddot{Z}_{н1} + (F_{ш1} - F_{п1}) = 0;$$

$$m_{н2} \ddot{Z}_{н2} + (F_{ш2} - F_{п2}/2) = 0;$$

$$m_{н3} \ddot{Z}_{н3} + (F_{ш2} - F_{п2}/2) = 0;$$

$$\beta_1 = \frac{Z_1 - Z_2}{2L_1 + 2L_2}; \beta_2 = \frac{Z_{cab1} - Z_{cab2}}{2L_5 + 2L_4};$$

$$\dot{\beta}_1 = \frac{\dot{Z}_1 - \dot{Z}_2}{2b_{ш1} + 2b_{ш1}}; \dot{\beta}_2 = \frac{\dot{Z}_{cab1} - \dot{Z}_{cab2}}{2L_5 + 2bL_4};$$

$$F_{ш1} = F_{сш1} (q_{1л}) + F_{кш1} (\dot{q}_{1л});$$

$$F_{ш2} = F_{сш2} (q_{2л}) + F_{кш2} (\dot{q}_{2л});$$

$$F_{ш3} F_{ш1} = F_{сш3} (q_{3п}) + F_{кш3} (\dot{q}_{3л});$$

$$F_{п1} = F_{п1} (L_1 \beta_1 - L_2 \beta_1) - F_{ам1} (L_1 \dot{\beta}_1 - L_2 \dot{\beta}_1) + F_{тр1};$$

$$F_{п2} = F_{п2} (L_2 \beta_2 - L_{п2} \frac{(\beta_{к2л} - \beta_{к3л})}{2}) + F_{ам2} (L_2 \dot{\beta}_2 + L_2 \frac{\dot{\beta}_{к2л} - \dot{\beta}_{к3л}}{2}) + F_{тр2},$$

где $F_{сш1} = F_{п1} (L_1 \beta_1 - L_2 \beta_1) - F_{ам1} (L_1 \dot{\beta}_1 - L_2 \dot{\beta}_1) + F_{тр1};$

$$F_{п2} = F_{п2} (L_2 \beta_2 - L_{п2} \frac{(\beta_{к2л} - \beta_{к3л})}{2}) + F_{ам2} (L_2 \dot{\beta}_2 + L_2 \frac{\dot{\beta}_{к2л} - \dot{\beta}_{к3л}}{2}) + F_{тр2},$$

$F_{сш2}, F_{сш3}$ – упругая характеристика шин переднего, среднего заднего моста в виде зависимости от относительной их деформации;

$F_{п1}, F_{п2}$, – упругая характеристика рессор переднего моста и балансирной подвески в виде зависимости от их относительной деформации;

$F_{тр1}, F_{тр2}$ – характеристика трения рессор переднего моста и балансирной подвески в виде зависимости интеграла от относительной скорости

перемещения с ограничениями ее величины; $F_{ам1}$ – характеристика амортизаторов переднего моста в виде зависимости от относительной скорости перемещения;

$F_{п1}, F_{п2}$ – суммарная характеристика рессоры, амортизатора и трения переднего моста и балансирной подвески; m_a, m_{cab} – масса подрессоренной части автомобиля, кабины;

$m_{н1}, m_{н2}, m_{н3}$ – неподрессоренная масса переднего, среднего, заднего моста;

L_1, L_2 – расстояние между центром тяжести и передней и задней подвесками;

$J_y = m_a \rho_{y1}^2, J_{y2} = m_{cab} \rho_{y2}^2$ – момент инерции подрессоренной массы автомобиля, кабины в продольной плоскости. $\ddot{\varphi} = x_i$

Усилие от вертикальными сил, передаваемое через трансмиссию представляет разность воздействия правой и левой стороны автомобиля и составляет

$$F_{xi} = \frac{B_i F_M \dot{i}_M}{2r_k} - \frac{B_i F_M \dot{i}_M}{2r_k}$$

где F_M – сумма сил, действующих на мост от действия микропрофиля дороги каждой стороны $F_{ш1} = F_{сш1}(q_{1л}) + F_{кш1}(\dot{q}_{1л})$ и противодействия подвески $F_{п1} = F_{р1}(L_1 \beta_1 - L_2 \beta_1) - F_{ам1}(L_1 \dot{\beta}_1 - L_2 \dot{\beta}_1) + F_{тр1}$;

$\dot{i}_M$ – передаточное число ведущего моста;

B_i – расстояние между колесами моста.

При моделировании рассматривались колебания правой и левой стороны машины при воздействии соответствующих микропрофилей дороги. Более идеальным было бы использование пространственной модели [26,74]. Но это очень сложная и трудоемкая для Simulink задача в режиме реального времени с учетом оценки спектра и реальных нелинейностей [14]. Поэтому выбран подход – рассматривать автомобиль, разделенный на две части в поперечной плоскости при разном микропрофиле. В процессе моделирования усилие записывалось в файл (через модуль To File). После чего эти усилия воспроизводятся через (через модуль From File) и обрабатывались по спектру.

Возможны другие источники воздействия: такие как эксцентриситет и неравномерность, вращения колес, их неуравновешенность могут быть введены в систему в виде синусоидального возмущения в соответствии скорости движения.

Более сложным является учет вращающихся частей двигателя, которые все же могут быть определены из кинематического анализа в пакете MS.ADAMS и записаны в файл для последующего использования.

В подразделе п.3.7 рассматриваются в качестве примера учет колебаний дизель-генераторной установки, как один из возможных путей решения.

Более сложным является моделирование колебательных процессов, вызванных работой двигателя и трансмиссии. Здесь необходимо совместное моделирование движения, колебаний автомобиля и колебательных процессов в трансмиссии с использованием схем подобных рисунку 2.24, 2.25. К сожалению, по данному вопросу нет исследований в открытых источниках и предложений как это осуществить.

Выводы по главе 2

1. Рассмотрены возможности и области применения GPS для исследований автомобиля, проведена оценка точности определения местоположения и перемещения техники с помощью различных методов, предложен метод обработки данных и получения продольного профиля дороги, скорости (режима) движения автомобиля с помощью GPS (по параметрам скорости и высоте движущегося объекта на поверхности Земли).

2. Выявлено, что рассмотренные зарубежные методы определения местоположения дают значительные расхождения (до 2-х раз), что связано с использованием различных систем базовых точек отсчета в СНГ и за рубежом в т.ч. в картографии, электронных картах. При оценке пройденного пути по разности перемещений двух точек проекций позиционирования и их суммирования при небольших шагах квантования (8–20 м, зависящих от скорости) рассмотренные методы также приводят к расхождению до 30–47 %.

3. Для получения продольного профиля дороги в зависимости от пройденного пути, режима перемещения автомобилей, ВТ, БЛА более корректным является использования данных GPS по скорости и высоте, обеспечивающие минимальную погрешность (0,15 %), что подтверждается характером изменения, $h=f(l)$, $v_a=f(l)$ и свидетельствуют о корректности предложенного метода. Также при моделировании движения автомобиля необходимо учитывать дорожные факторы, влияющие на режим движения.

4. Разработана методика и программное обеспечение на языке C/C++, которые обеспечивают получение данных по продольному профилю дорог, режиму движения автомобиля с помощью модуля GPS GN-803G с точностью от 0,6 до 2 м, скорости движения с погрешностью в 0,05 м/сек, курсового направления до 0,3° достаточную для исследований движения в дорожных условиях и моделирования автомобиля в Matlab/Simulink.

5. На территории РБ при определении относительных расстояний и привязке к картам при преобразовании можно пользоваться упрощенной линейной формулой $lat_m = 112297,6196 \cdot D_{dec\ lat}$, $lon_m = 63215,444 \cdot D_{dec\ lon}$.

5. На основании проведенного анализа предложено использование задания внешнего возмущения в виде объединенного макро и микропрофиля дороги, полученного с помощью микропроцессорных систем на движущемся автомобиле [14, 55, 56] в сочетании с применением интерполяции промежуточных значений по методу интерполяции Catmull-Rom по пяти точкам, что обеспечивает наилучшее совпадение результатов расчета и эксперимента.

7. Предложен метод задания и хранения большого массива (96000 точек) макро и микропрофиля дороги, подкачку дополнительной информации, осуществления переменной задержки времени воздействия на оси с помощью разработанной программы в модуле S-Function Builder Simulink, считывание и обмен информации с органов управления на основе мини-компьютера Raspberry и АЦП, важные для реализации моделирования динамической системы автомобиля.

8. Проанализированы предположительные причины продольных и поперечных вибраций автомобиля, связанные с неуравновешенностью самого двигателя, его работой со взрывным характером сгорания топлива в цилиндрах и колебательными процессами в трансмиссии.

9. Установлено, что после установки противовесов двигатель V8 с углом развала цилиндров 90° полностью уравновешен от действия всех инерционных сил и моментов.

10. Предложены схема и модель вибраций двигателя (дизеля V8), базирующуюся на основе индикаторного воздействия сил сгорания топлива в виде дискретной аппроксимации синусоиды с переменной частотой от оборотов и развиваемых усилий двигателя в качестве источника возмущения на трансмиссию и автомобиль, обуславливающая его продольные и поперечные колебания.

11. Разработана процедура пересчета оборотов двигателя в угол поворота коленвала в виде повторяющегося цикл 720° (два оборота коленвала) на основе счетчика колебаний с переменным значением $ch=(n/60) \cdot t_m$ с учетом шага моделирования, что позволило смоделировать вибрационное воздействие по продольным и поперечным вибрациям на трансмиссию и автомобиль при движении по реальной дороге.

12. Дополнительно рассмотрен вариант продольных ускорений автомобиля, вызываемый поперечно-угловыми поворотами моста под действием вертикальных вибраций левого и правого микропрофиля дороги относительно карданных валов, передающих моменты, необходимые для движения автомобиля.

ГЛАВА 3

МОДЕЛИ И МЕТОДИКА МОДЕЛИРОВАНИЯ АВТОМОБИЛЯ СО СМЕШАННОЙ РЕАЛЬНОСТЬЮ С СИМУЛЯТОРОМ

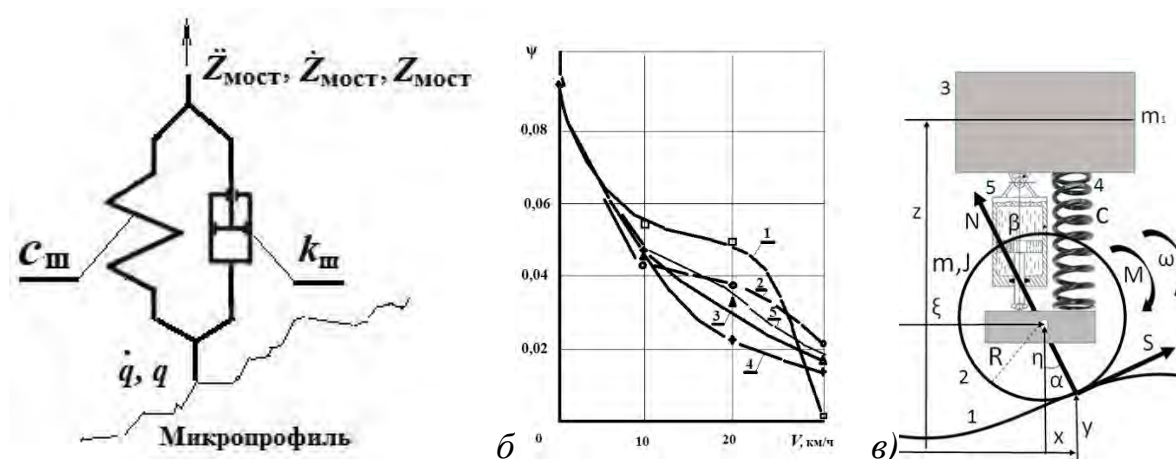
3.1 Модели системы “дорога–автомобиль–водитель”

Корректность моделирования помимо источников возмущения зависит также от используемых моделей, их связей, рассматриваемых ниже.

3.1.1 Шины

Для колесных машин специального назначения применяются как двухбортные, так и одно бортные шины. Чаще применяют однобортные шины. Они позволяют повысить проходимость на мягких грунтах.

Автомобильная шина обладает упругостью в радиальном, боковом и тангенциальном направлениях, а также демпфированием (рисунок 3.1).



1 – статическая нагрузка; 2 – 50 % статической нагрузки;

3 – +50% статической нагрузки; 4 –+ статическая нагрузка; 5 – среднее значение

**Рисунок 3.1. – Модель шины (а) и затухание в ней при сбрасывании
на барабанном стенде (б)**

Значения коэффициента демпфирования шины определяют экспериментально по собственным затухающим колебаниям динамической системы, рассматривая шину в виде упругого элемента. Приведенные в работе [57] исследования собственных колебаний масс, опирающихся на вращающееся и неподвижное колесо, показали большое различие в коэффициентах затухания при вращающихся и неподвижных колесах. На рисунке 3.2б приведены графики, взятые из работы [57], по относительным коэффициентам демпфирования шины 12,00 – 18 при давлении воздуха 1,4 и 2,8 кг/см², при

неподвижной шине и вращающейся со скоростью 21 км/ч.

С увеличением скорости вращения относительный коэффициент демпфирования уменьшается и при скорости 20 км/ч становится пренебрежимо малым. Учитывая малое затухание во вращающейся шине, в расчетных схемах представляется возможным затухание в шинах не учитывать.

На основе вышеизложенного при расчетах колебаний автомобиля шину можно рассматривать в виде упругого элемента с "вязким" трением с точечным контактом, что и было применено в работе.

Жесткость шины можно определить по формуле Бидермана В. Л.,

$$C_{\text{ш}} = \frac{F}{f_{\text{ш}}} \left(2 - C \frac{F}{p_{\text{ш}} \cdot f_{\text{ш}}} \right),$$

где F – нагрузка на шину, $f_{\text{ш}}$ – прогиб при нагрузке, $p_{\text{ш}}$ – давление воздуха.

Коэффициент C зависит от размеров шины; его значения, полученные экспериментально, следующие:

Размеры шины	Коэффициент C	Размеры шины	Коэффициент C
6,00–16	0,0067	7,50–20	0,00
7,50–16	0,0059	260–20	0,00

3.1.2 Упругие и направляющие элементы подвески

В качестве упругих элементов подвески для автомобилей в основном применяются рессорные и пружинно-рычажные подвески (рисунок 3.2). Рессора служит одновременно и направляющим элементом.

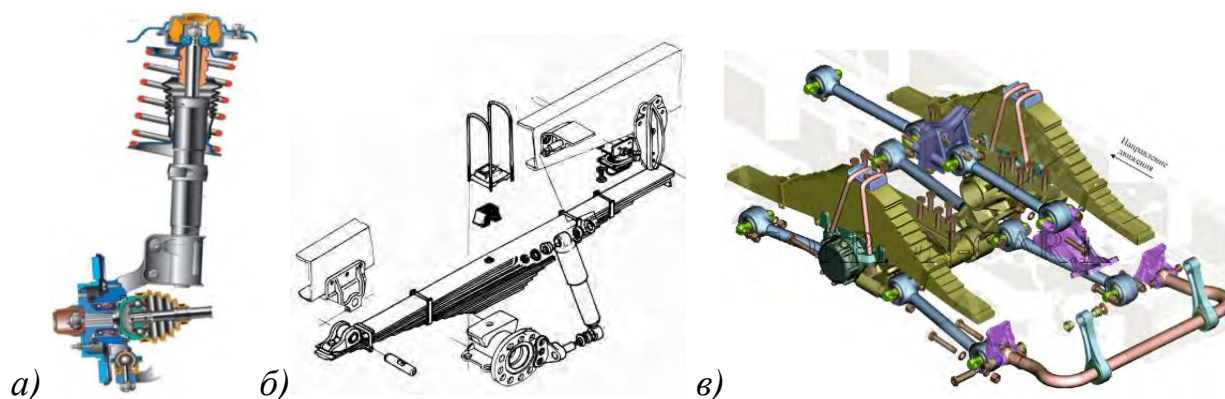


Рисунок 3.2. – Пружинно-рычажная передняя подвеска легкового автомобиля (а), рессорная передняя (б) и задняя (в) подвеска автомобиля МАЗ-6317

Также могут применяться пневматические и гидропневматические подвески (МАЗ-547, МЗКТ-6922) и пружинные типа Мак-Ферсон (рисунок 3.2а).

Характеристика пружинной подвески имеет линейную зависимость от прогиба. Динамическая характеристика рессоры представляет собой сложную зависимость $F_p = f(\dot{\Delta}, \Delta, \omega)$. Характеристику упругого элемента необходимо рассматривать как сумму характеристик упругого элемента и трения. Однако большинство исследователей применяет простую зависимость $F_p = f(\Delta)$. В работе [47] предложена блок-схема реализации на аналоговых машинах модели рессоры без ее математического описания, которая обеспечивает лучшее совпадение по спектру. Однако многим модель была непонятна и на нее не обратили внимание, из-за этого она не получила применения. На основе проведенных анализов работ автора [14, 47–48] и исследований (см. п.4.5.3) предложено ее математическое описание в виде

$$\begin{aligned}
 F_{p,ст} &= c_{p,ст} (Z_{нм} - Z_{пм}); \\
 F_{тр1} &= \begin{cases} c_{дин1} \int_0^t (\dot{Z}_{нм} - \dot{Z}_{пм}) dt & \text{если } F_{тр1} > c_{од,дин} \int_0^t (\dot{Z}_{нм} - \dot{Z}_{пм}) dt > -F_{тр1}; \\ F_{тр1}, & (-) \quad c_{под,дин1} \int_0^t (\dot{Z}_{пм} - \dot{Z}_{нм}) dt > F_1; \\ -F_{тр1}, & (-) \quad c_{од,дин1} \int_0^t (\dot{Z}_{нм} - \dot{Z}_{пм}) dt < -F_1; \end{cases} \\
 F_{тр2} &= \begin{cases} c_{под,дин2} \int_0^t (\dot{Z}_{нм} - \dot{Z}_{пм}) dt & \text{если } F_{тр2} > c_{под,дин} \int_0^t (\dot{Z}_{нм} - \dot{Z}_{пм}) dt > -F_{тр2}; \\ F_{тр2}, & (-) \quad c_{под,дин2} \int_0^t (\dot{Z}_{пм} - \dot{Z}_{нм}) dt > F_2; \\ -F_{тр2}, & (-) \quad c_{под,дин2} \int_0^t (\dot{Z}_{нм} - \dot{Z}_{пм}) dt < -F_2; \end{cases} \\
 F_{ам} &= \begin{cases} k_{в,тр} (\dot{Z}_{нм} - \dot{Z}_{пм}) & \text{если } F_{тр5max} > k_{сп} (\dot{Z}_{нм} - \dot{Z}_{пм}) > -F_{ам,маx}; \\ F_{ам,маx} & (-) \quad k_{сп} (\dot{Z}_{сид} - \dot{Z}_{гр}) > F_{ам,маx}; \\ -F_{ам,маx} & (-) \quad k_{сп} (\dot{Z}_{сид} - \dot{Z}_{гр}) < -F_{ам,маx}; \end{cases}
 \end{aligned}$$

$$F_{рес} = F_{p,ст} + F_{тр1} + F_{тр2} + F_{ам},$$

где $c_{p,ст}$, – жесткость рессоры в статике; $c_{р,дин1}$, $c_{р,дин2}$ – жесткость рессоры при блокировке подвески; $F_{тр}$, $F_{тр2}$, – величина усилия при трении 1, 2; $F_{р1max}$, $F_{р2max}$ – максимальная величина трения 1, 2;

$k_{в,тр}$ – коэффициент “вязкого” трения;

$F_{ам}$ – максимальная величина “вязкого” трения.

Этому описанию соответствует модель на рисунке 3.3.

Предложенная модель и ее описание является новым и важным компонентом в обеспечении корректности моделирования колебаний и нагрузок автомобиля.

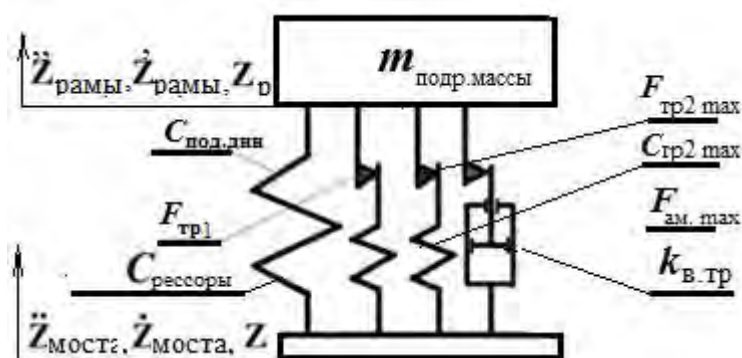


Рисунок 3.3. – Модель упругого элемента

3.2 Моделирование упругих элементов в пакете MSC.ADAMS

В пакете MSC.ADAMS с помощью модуля Chassis можно рассчитать упругую характеристику рессоры ТС, рассматривая ее как совокупность масс, связанных между собой упругими элементами типа Beam (балка) с учетом “вязкого” трения. Для этого в нем задаются параметры упругости и геометрические параметры сечения: площадь поперечного сечения, моменты инерции по всем осям и длина листов рессоры. С помощью утилиты LeafSpringEditor модуля MSC.ADAMS/Chassis рессора создается путем задания профиля каждого листа, механических свойств материала и других геометрических параметров рессоры. Модели рессор показаны на рисунках 3.4 и 3.5. На рисунке 3.4 показано окно создания рессоры в утилите LeafSpringEditor.

После задания параметров производится упрощение полученной рессоры в модуле MSC.ADAMS/Avview. В результате на стадии проектирования можно рассчитать характеристику рессоры в совокупности с моделью “вязкого” трения. На рисунке 3.6 приведены рассчитанные упругие характеристики моделей рессор в сравнении с экспериментальными характеристиками рессор. Как видно из рисунка, имеет место хорошее совпадение характеристик при статическом нагружении.

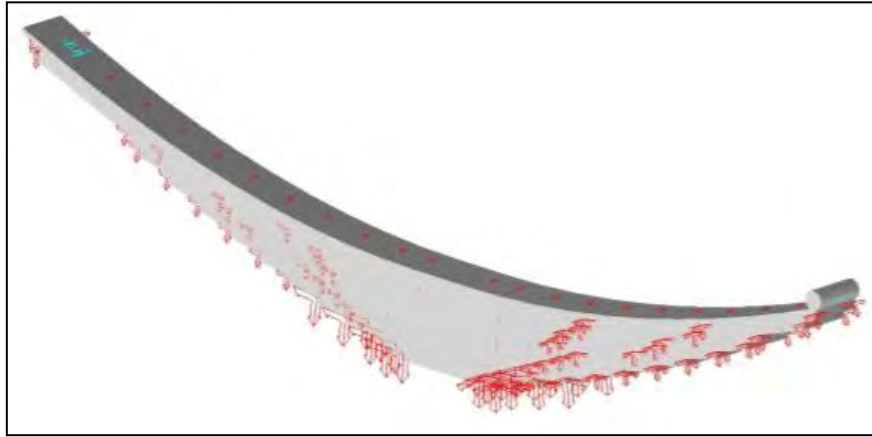


Рисунок 3.4. – Модель рессоры передней подвески

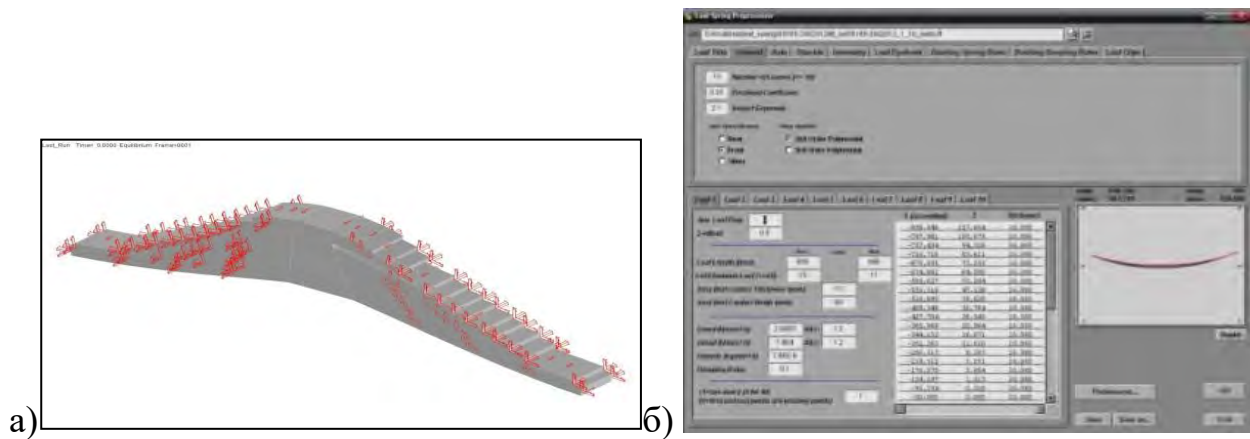


Рисунок 3.5. – Модель рессоры задней подвески (а) и окно утилиты (б)

LeafSpringEditor

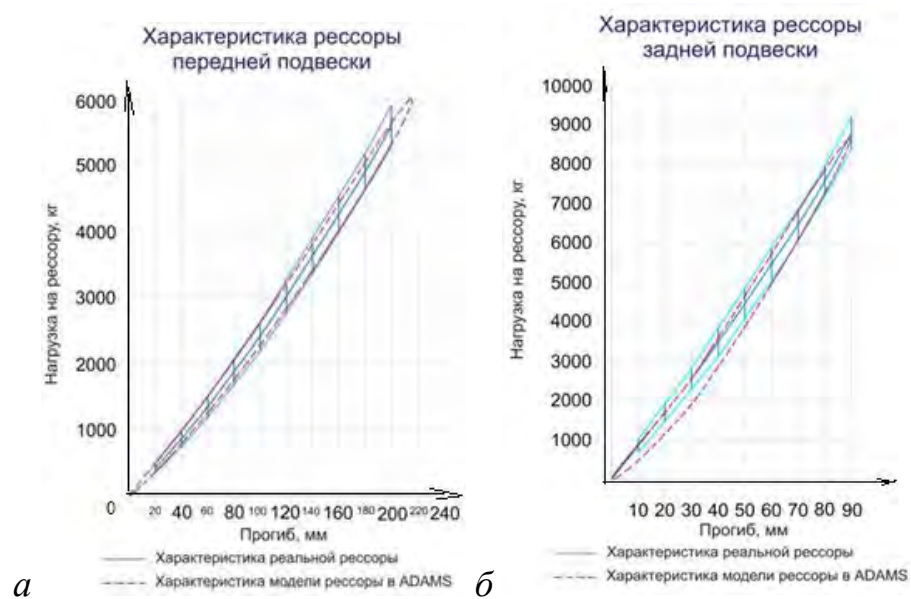


Рисунок 3.6. – Характеристики рессор передней (а) и задней (б) подвески

Модели передней и задней подвески колесной машины, создаваемые в пакете MSC.ADAMS из стандартных элементов и используемые при проведения моделирования, приведены на рисунке 3.7.

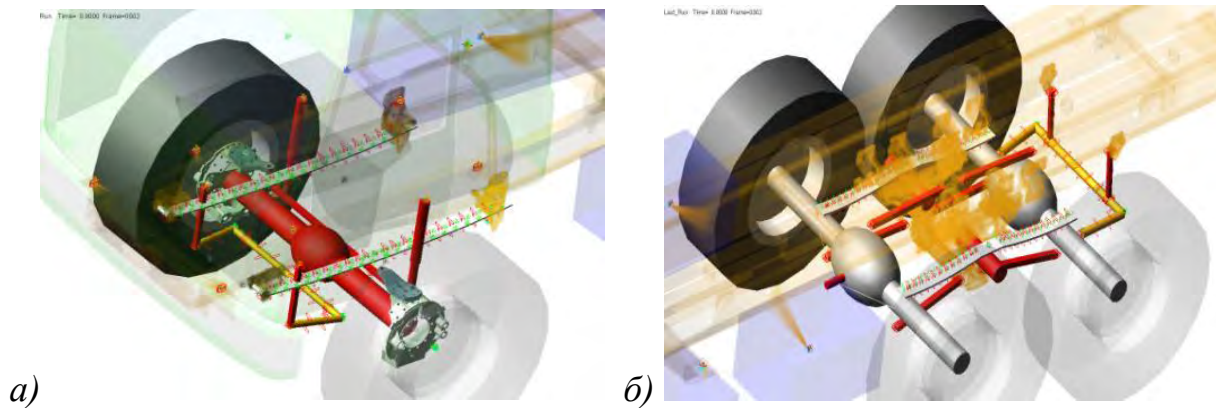
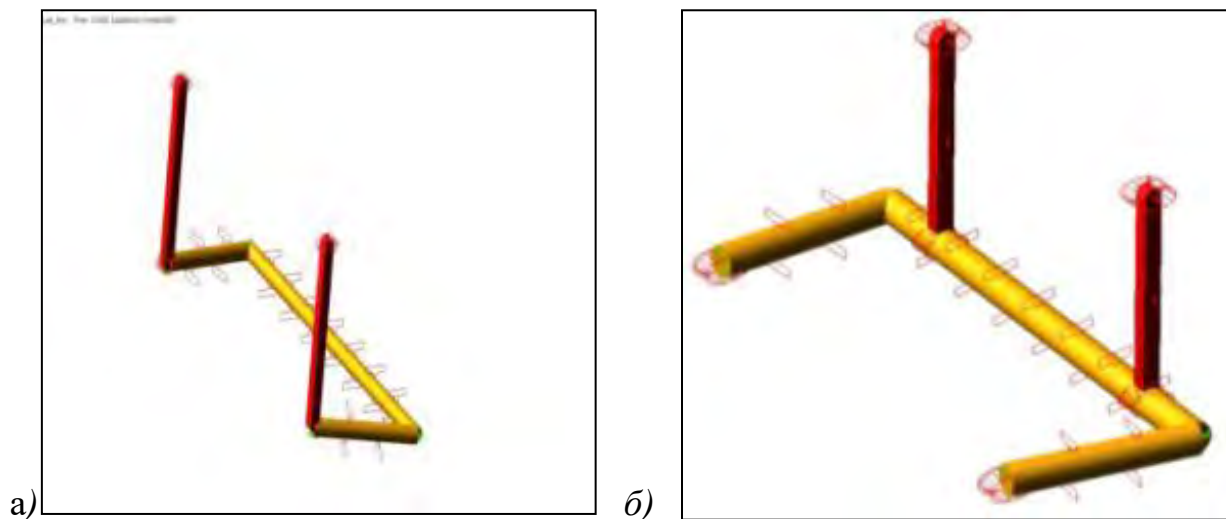


Рисунок 3.7 – Модель передней (а) и задней (б) подвески

3.3.2 Моделирование стабилизаторов. С помощью функции DiskreteFlexibleLink в пакете MSC.ADAMS можно смоделировать стабилизаторы поперечной устойчивости автомобиля, их характеристику как совокупность масс, связанных между собой упругими элементами типа Beam (балка). Модели стабилизаторов показаны на рисунках 3.8.



а – передней подвески;

б – задней подвески

Рисунок 3.8 – Модель стабилизатора

3.3 Модель гидравлического амортизатора

Важным элементом конструкции подвески ТС, влияющим на вибрации, является амортизатор. В настоящее время в подвесках транспортных средств, включая сиденье, как наиболее технологичные применяются исключительно только гидравлические телескопические амортизаторы. Они бывают двух типов: двухтрубные и однотрубные газонаполненные. Большинство исследователей рассматривают их сопротивление как зависимость от относительной скорости $F_{амор} = f(\dot{\Delta})$, что не всегда так.

На двухтрубных амортизаторах, функционирующих в нормальном режиме, сложно получить симметричную характеристику, так как сопротивление сжатия обуславливается малым объемом дросселирующей жидкости, равной объему перемещаемого штока. Обычно усилие сжатия в них составляет 10–20 % от усилия отбоя. Из-за слабых усилий сжатия в подвесках могут происходить пробои. Во многом из-за этого в подвесках сидений и легковых автомобилей (кроссоверах, внедорожниках) начали применяться газонаполненные амортизаторы, обеспечивающие большие усилия сжатия, вопросы теории которых еще недостаточно проработаны.

Действительная характеристика гидравлического амортизатора в ряде случаев имеет более сложную многопараметрическую зависимость $F_{амор} = f(\dot{\Delta}, \Delta, \omega)$, отличающуюся от общепринятой $F_{амор} = f(\dot{\Delta})$.

3.3.1 Разработка динамических моделей гидравлических телескопических амортизаторов

Гидравлический амортизатор (рисунок 3.9) является одним из основных элементов системы поддрессирования транспортного средства,

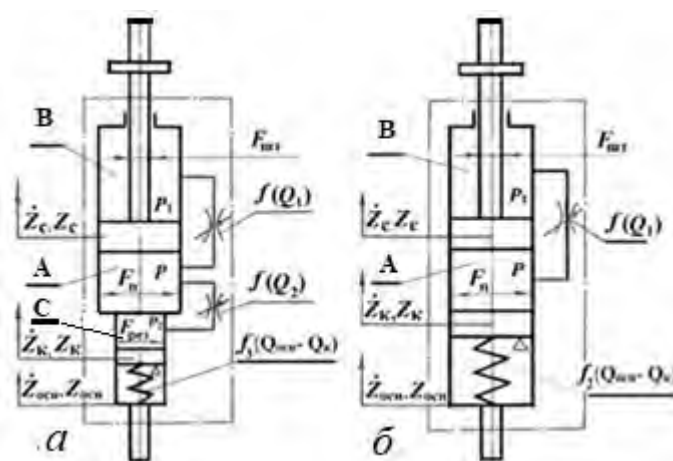


Рисунок 3.9. – Принципиальные схемы гидравлических амортизаторов:
двухтрубного (а) и газонаполненного (б)

включая сиденье и кабину, от которого во многом зависит отсутствие пробоев подвески и обеспечение минимальных ускорений на рабочем месте водителя.

Как показано в работах автора [14, 47–49] характеристики двух основных типов гидравлических амортизаторов: двухтрубного с переменным направлением потока жидкости и однострубнога газонаполненного, приведенных на рисунке 3.9 можно описать следующими уравнениями.

Для аномального цикла работы **двухтрубного амортизатора** уравнение будет иметь вид

$$f_2(Q_2) = f_1(Q_{\text{общ}} - Q_2) - \frac{1}{F_{\text{рез}}} f(Z_{\text{осн}} - Z_{\text{к}}).$$

Решение этого уравнения относительно высшей производной Q можно получить с помощью пакета Matlab/Simulink и определить исходные значения для расчета усилия амортизатора,

$$F_{\text{амор}} = F_{\text{п}} p = F_{\text{п}} f_2(Q_2) + \frac{F_{\text{п}}}{F_{\text{рез}}} f(Z_{\text{осн}} - Z_{\text{к}}).$$

При нормальной работе амортизатора на ходе сжатия

$$\begin{aligned} Q_1 &= (F_{\text{п}} + F_{\text{шт}}) (\dot{Z}_{\text{осн}} - \dot{Z}_{\text{сид}}); \\ Q_2 &= F_{\text{шт}} (\dot{Z}_{\text{осн}} - \dot{Z}_{\text{сид}}). \end{aligned}$$

Усилие сжатия амортизатора

$$F_{\text{амор}} = F_{\text{п}} p - (F_{\text{п}} - F_{\text{шт}}) p_1 = F_{\text{п}} p - (F_{\text{п}} - F_{\text{шт}}) (p - f_1(Q_1)),$$

или

$$F_{\text{амор}} = F_{\text{шт}} p + (F_{\text{п}} - F_{\text{шт}}) f_1(Q_1),$$

$$\text{где } p = f_2(Q_2) + \frac{1}{F_{\text{рез}}} f_3(Z_{\text{осн}} - Z_{\text{к}}) = f_2[F_{\text{шт}} (\dot{Z}_{\text{осн}} - \dot{Z}_{\text{сид}}) + \frac{1}{F_{\text{рез}}} f_3(\frac{F_{\text{шт}}}{F_{\text{рез}}}(Z_{\text{осн}} - Z_{\text{к}}))].$$

а) При ходе отбоя двухтрубного амортизатора при разрыве потока жидкости

$$F_{\text{амор}} = F_{\text{п}} f_2(Q_2) + \frac{F_{\text{шт}}}{F_{\text{рез}}} f_3(Z_{\text{осн}} - Z_{\text{к}}),$$

Исходные параметры Q_2 , $Z_{\text{осн}} - Z_{\text{к}}$ находятся при решении дифференциального уравнения

$$f_2(Q_2) = f_1(Q_{\text{общ}} - Q_2) - \frac{F_{\text{п}}}{F_{\text{рез}}} f_3(Z_{\text{осн}} - Z_{\text{к}}),$$

б) Когда в полости В отсутствует разрыв потока жидкости

$$F_{\text{амор}} = (F_{\text{п}} - F_{\text{шт}}) f_5(\dot{Z}_{\text{осн}} - \dot{Z}_{\text{сид}}) - \frac{F_{\text{п}}}{F_{\text{рез}}} f_3\left[\frac{F_{\text{шт}}}{F_{\text{рез}}}(Z_{\text{осн}} - Z_{\text{к}})\right],$$

где $f_5(\dot{Z}_{\text{осн}} - \dot{Z}_{\text{сид}})$ – гидравлическая характеристика потерь давления клапана отбоя.

Для газонаполненного амортизатора случая (а) уравнение сжатия будет иметь вид:

для нелинейной зависимости $\Delta p_1 = f_1(Q_1$

$$f_1[F_{\text{п}}(\dot{Z}_{\text{осн}} - \dot{Z}_{\text{сид}})] = \frac{1}{F_{\text{п}}} f_2(Z_{\text{осн}} - Z_{\text{к}}),$$

для линейной зависимости $\Delta p_1 = f_1(Q_1)$

$$f_1(F_{\text{п}}\dot{Z}_{\text{осн}}) - f_1(F_{\text{п}}\dot{Z}_{\text{сид}}) = \frac{1}{F_{\text{п}}} f_2(Z_{\text{осн}} - Z_{\text{к}}),$$

или

$$f_1(F_{\text{п}}\dot{Z}_{\text{осн}}) = f_1(F_{\text{п}}\dot{Z}_{\text{сид}}) + \frac{1}{F_{\text{п}}} f_2(Z_{\text{осн}} - Z_{\text{к}}).$$

Решая эти уравнения относительно высших производных, $\dot{Z}_{\text{к}} - \dot{Z}_{\text{сид}}$, $\dot{Z}_{\text{к}}$, найдем исходное значение $Z_{\text{осн}} - Z_{\text{к}}$ для расчета усилия сжатия амортизатора:

$$F_{\text{амор}} = F_{\text{п}} p = f_2(Z_{\text{осн}} - Z_{\text{к}}),$$

Для случая (б) уравнение сжатия газонаполненного амортизатора будет иметь вид

$$p = -f_1(Q_1) + \frac{1}{F_{\text{п}}} f_2(Z_{\text{осн}} - Z_{\text{к}}) = \frac{1}{F_{\text{п}}} f_2\left[\frac{F_{\text{шт}}}{F_{\text{п}}}(Z_{\text{осн}} - Z_{\text{к}}) - f_1[(F_{\text{п}} - F_{\text{шт}})(Z_{\text{осн}} - Z_{\text{к}})]\right];$$

$$p_1 = -f_1(Q_1) + \frac{1}{F_{\text{п}}} f_2(Z_{\text{осн}} - Z_{\text{к}}) = \frac{1}{F_{\text{п}}} f_2\left[\frac{F_{\text{шт}}}{F_{\text{п}}}(Z_{\text{осн}} - Z_{\text{к}}) - f_1[(F_{\text{п}} - F_{\text{шт}})(Z_{\text{осн}} - Z_{\text{к}})]\right];$$

$$p = \frac{1}{F_{\text{п}}} f_2\left[\frac{F_{\text{шт}}}{F_{\text{п}}}(Z_{\text{осн}} - Z_{\text{к}})\right];$$

$$F_{\text{амор}} = f_2 \left[\frac{F_{\text{шт}}}{F_{\text{п}}} (Z_{\text{осн}} - Z_{\text{к}}) - \frac{F_{\text{п}} - F_{\text{шт}}}{F_{\text{п}}} f_2 \left[\frac{F_{\text{шт}}}{F_{\text{п}}} (Z_{\text{осн}} - Z_{\text{к}}) \right] + (F_{\text{п}} - F_{\text{шт}}) f_1 [(F_{\text{п}} - F_{\text{шт}})] + \right. \\ \left. (F_{\text{п}} - F_{\text{шт}}) (Z_{\text{осн}} - Z_{\text{к}}) \right];$$

$$F_{\text{амор}} = \frac{F_{\text{шт}}}{F_{\text{п}}} f_2 \left[\frac{F_{\text{шт}}}{F_{\text{п}}} (Z_{\text{осн}} - Z_{\text{к}}) \right] + (F_{\text{п}} - F_{\text{шт}}) f_1 [(F_{\text{п}} - F_{\text{шт}}) (\dot{Z}_{\text{осн}} - \dot{Z}_{\text{сид}})],$$

где $f_2 \left[\frac{F_{\text{шт}}}{F_{\text{п}}} (Z_{\text{осн}} - Z_{\text{к}}) \right]$ – характеристика компенсационной камеры

относительно положения статического равновесия амортизатора.

При отбое газонаполненного амортизатора также возможны два варианта:

в полости Б еще при сжатии возник разрыв потока жидкости; в этом случае уравнением отбоя является уравнение сжатия (случай а)

$$f_1 [(F_{\text{п}} (\dot{Z}_{\text{осн}} - \dot{Z}_{\text{сид}}))] = \frac{1}{F_{\text{п}}} f_2 (Z_{\text{осн}} - Z_{\text{к}}), \\ F_{\text{амор}} = f_2 (Z_{\text{осн}} - Z_{\text{к}}).$$

В полости В отсутствует разрыв потока жидкости. Тогда

$$F_{\text{амор}} = (F_{\text{п}} - F_{\text{шт}}) f_{\text{отб}} [(F_{\text{п}} - F_{\text{шт}}) (\dot{Z}_{\text{осн}} - \dot{Z}_{\text{сид}}) - \frac{F_{\text{шт}}}{F_{\text{п}}} f_2 \left[\frac{F_{\text{шт}}}{F_{\text{п}}} (Z_{\text{осн}} - Z_{\text{к}}) \right]].$$

3.4.2 Проверка предложенной модели амортизатора

Проверка предложенной модели газонаполненного амортизатора КамАЗ проведена путем моделирования колебаний водителя на сиденье грузового автомобиля МАЗ-5336 (рисунок 3.10), движущегося по асфальтированному шоссе со скоростью 70 км/ч с использованием реального возмущения пола кабины [47].

Выбор данного метода проверки обусловлен возможностью применения более упрощенной модели по сравнению с полной моделью автомобиля. Это позволяет исключить возможную некорректность всей модели и основных элементов подвески: рессор, рамы, двигателя, трансмиссии и их характеристик. Использовалась динамическая модель системы “сиденье-водитель” в виде трехмассовой модели человека (масс груди и таза), подушки сиденья и массы подрессоренных частей сиденья (см. п. 3.6.2).

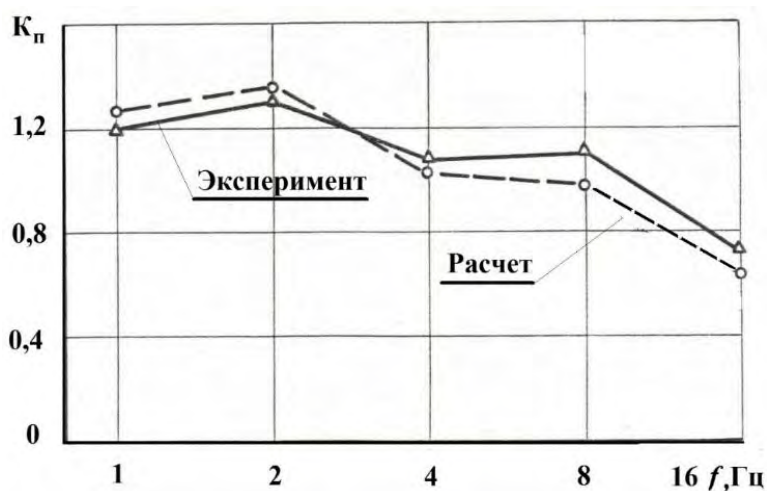


Рисунок 3.10. – Сравнение виброзащитных свойств поддрессоренного сиденья с газонаполненным амортизатором при движении по асфальтированному шоссе в октавных полосах частот

Виброзащитные свойства сиденья оценивались по коэффициенту передачи вибраций в октавных полосах частот, представляющему отношение среднеквадратических ускорений на подушке сиденья водителя к уровню ускорений пола кабины. Как видно из рисунка 3.10, где приведены результаты моделирования, предложенная модель обеспечивает хорошее совпадение с экспериментом (расхождение 5-12 % в октавных полосах частот 1, 2, 4, 8, 16 Гц против 30–60 % у упрощенных моделей [14, 41, 43, 51, 79]), что свидетельствует о ее корректности.

3.4 Модель пневмогидравлического цилиндра подвески

Гидропневматический цилиндр подвески представляет собой объединенный в один узел упругий элемент и амортизатор (рисунок 3.11).

В качестве упругого элемента выступает сжатый азот под высоким давлением, находящийся в верхней части цилиндра. Применяются два типа цилиндров: с противодавлением (рисунок 3.12а) и без противодавления. Первая конструкция применяется на большегрузных самосвалах и специальной внедорожной большегрузной технике. Примером является конструкция цилиндра самосвала БелАЗ-7540 (рисунок 3.12). Цилиндр воспринимает только вертикальные нагрузки, а в качестве направляющих используется система рычагов.

Работа пневматического цилиндра основана на принципе сжатия и расширения газа в соответствии с политропическим законом:

$$pV^n = \text{const},$$

где V - объем газа; p - давление газа; n - показатель политропы.

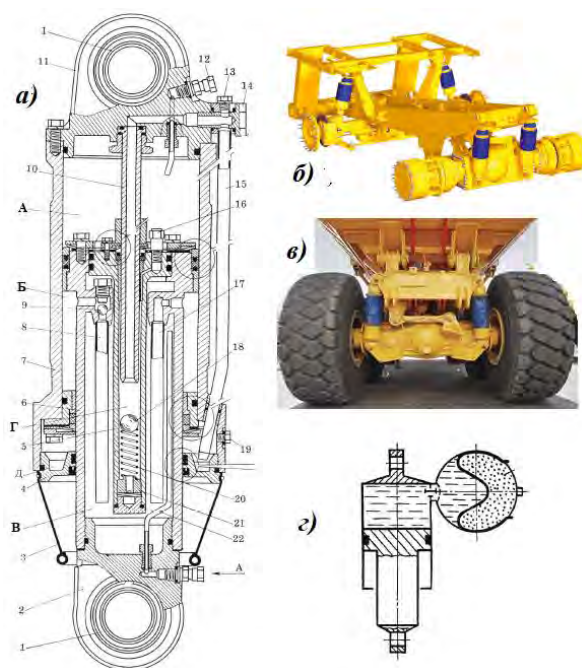


Рисунок 3.11. – Пневмогидравлический цилиндр (а), подвеска автомобиля БелАЗ (б, в), пневмогидравлический цилиндр без противодавления (г)

Показатель политропы зависит от теплоотдачи: при изотермическим процессе $n = 1$, при ухудшении теплопередачи с повышением скорости процесса значение n возрастает и может достигать 1,3... 1,4.

Упругая характеристика цилиндра подвески рассчитывается по заданным нагрузкам на цилиндр подвески для снаряженного автомобиля [22]:

$$p_x = p_0 \left(\frac{L_0}{(L_0 - \Delta x)} \right)^n$$

где Δx – отклонения поршня от начального положения L_x , L_0 , соответствующего, положению снаряженного автомобиля. В качестве положительного направления принят ход сжатия, отрицательного отбой). p_x и p_0 - давление в цилиндре, соответственно, в промежуточном и начальном положениях поршня.

В зависимости от скорости сжатия газа различают статическую ($n = 1$) и динамическую ($n = 1,2.. 1,3$) упругую характеристику.

При моделировании упругую характеристику цилиндра с учетом ниже изложенного по растворимости газа можно рассматривать приближенно как однопараметрическую $F = f(\Delta x)$. Демпфирующую характеристику в цилиндрах с противодавлением также можно рассматривать приближенно

как однопараметрическую $F = f(\Delta x)$.

При использовании цилиндров без противодействия (рисунки 3.12в, 3.13) его необходимо рассматривать как газонаполненный амортизатор, описываемый дифференциальными уравнениями (см. пп.3.4.1). Этим можно объяснить изменение жесткости в характеристике цилиндра, показанное на рисунке 3.12, взятое из работы [23].

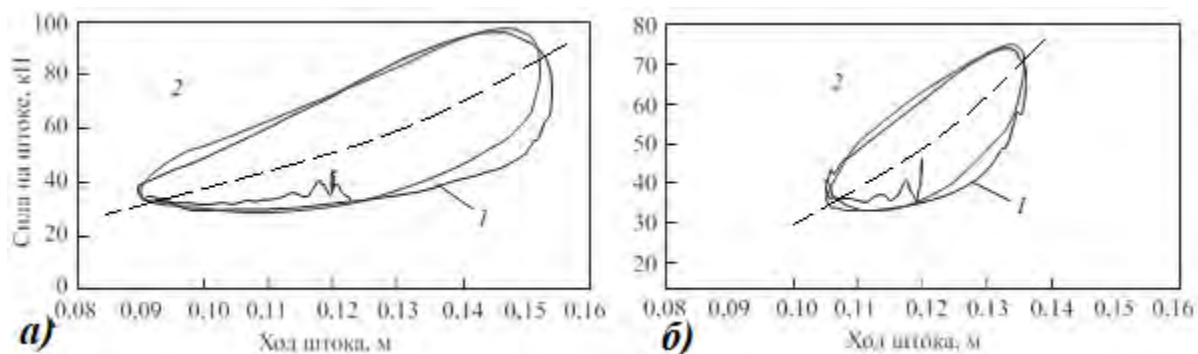


**Рисунок 3.12. – Динамические характеристики цилиндра, полученные при стендовых испытаниях (1) и моделировании (2), для различных режимов нагружения
а – $f = 1$ Гц, $A = 30$ мм; б – $f = 2$ Гц, $A = 15$ мм**

В цилиндрах с противодействием пневмогидравлической подвески карьерных самосвалов БелАЗ газовая и жидкая фаза не разделены, поэтому под воздействием давления газ частично растворяется в масле. В следствии растворения газа упругая характеристика изменяется т.к. уменьшается его количество, участвующее в процессе работы цилиндра. Согласно работы [22] при расчете параметров зарядки рабочей полости цилиндра подвески необходимо учитывать растворимость газа и пользоваться первоначальной статической упругой характеристикой с избыточным содержанием азота, называемой зарядной:

$$p_x = p_0 \frac{FL_0 + kV_{ж}}{F(L_0 - \Delta x)},$$

где F – активная площадь поршня; k – коэффициент растворимости азота, причем $k = V_p / V_{ж}$, где V_p – объем газа при некотором давлении p , который может раствориться в объеме жидкости $V_{ж}$ до ее полного насыщения (коэффициент k уменьшается при увеличении плотности жидкости, в которой растворяется газ, и практически не зависит от давления).

Из-за медленного растворения газа в качестве расчетного принимают положение поршня цилиндра подвески, соответствующее снаряженному автомобилю. Зарядную характеристику используют для того, чтобы определить, на сколько необходимо повысить начальное давление p_0 в

цилиндре (в растянутом положении), если он заправляется газом вне автомобиля, или на сколько должна быть увеличена первоначальная высота L_0 столба газа, если цилиндр заправляется на снаряженном автомобиле.

Отмечено изменение упруго-демпфирующих свойств цилиндра (в повышении ее жесткости) в ходе эксплуатации самосвалов БелАЗ при длительной непрерывной работе под нагрузкой (транспортировка грузов на большие расстояния), причем упругие свойства восстанавливались через относительно небольшой промежуток времени после разгрузки самосвала.

Для учета такого изменения в работе [22] предложено объем газовой полости в цилиндре подвески для снаряженного автомобиля при заданной величине L_0 определять по выражению $V_0 = L_0 \cdot F$. При нормальных условиях (ну). этот газ займет объем

$$V_{0,ну} = V_0 \frac{p_0}{p_{ну}},$$

где $p_{ну}$ - давление при нормальных условиях (0,1 МПа).

С целью компенсации растворимости в снаряженном состоянии в цилиндр подвески заправляется дополнительное количество газа V_{p0} . Приведенный к нормальным условиям этот газ будет занимать объем

$$V_{p,ну} = V_0 \frac{p_0}{p_{ну}} = kV_{ж} \frac{p_0}{p_{ну}}.$$

В результате получим суммарное количество газа, заправляемого в цилиндр подвески:

$$V_{\Sigma,ну} = V_{0,ну} + V_{p,ну} = (L_0 F + kV_{ж}) \frac{p_0}{p_{ну}},$$

что соответствует объему газовой полости при давлении p_0 , равном

$$V_{\Sigma,0} = V_{\Sigma,ну} \frac{p_0}{p_{ну}} = L_0 F + kV_{ж},$$

и изменению начального значения L_0 на величину $kV_{ж}/F$.

Для промежуточного положения подвески при давлении p_x приведенный к нормальным условиям, участвующий в формировании упругой характеристики объем газа $V_{ну x}$ по причине растворимости уменьшается на величину $V_{рну x}$ которую можно определить $V_{рну x} p_{ну} = V_{рх} p_x$ из уравнения состояния газа

$$V_{р,ну x} = V_{рх} \frac{p_x}{p_{ну}} = k V_{ж} \frac{p_x}{p_{ну}},$$

Тогда

$$V_{\text{нyx}} = -V_{\Sigma, \text{ny}} - V_{p \text{ny}} = (L_0 F + k V_{\text{ж}}) \frac{p_0}{p_{\text{ny}}} - k V_{\text{ж}} \frac{p_x}{p_{\text{ny}}}$$

Если сжать этот газ давлением p_x до объема $V_x = (L_0 - \Delta x)F$, то можно записать

$$V_{\text{ny}x} p_{\text{ny}} = V_x p_x$$

откуда получается зависимость, устанавливающую связь между степенью загрузки цилиндра (p_x) и отклонением Δx координаты поршня от снаряженного L_0 положения в случае полного насыщения масла азотом:

$$p_x = p_0 \frac{FL_0 + kV_{\text{ж}}}{F(L_0 - \Delta x) + kV_{\text{ж}}}.$$

В работе [22] приведены рассчитанные по различным зависимостям характеристики для задних цилиндров карьерного самосвала БелАЗ-7513 грузоподъемностью 130 тонн (рисунок 3.13). Из рисунка видно, что на ходе отбоя и сжатия до 0,05 м характеристики практически совпадают, но при учете растворимости азота наблюдается более существенное (по сравнению с другими характеристиками) отклонение положения поршня от снаряженного состояния при равном нагружении цилиндра подвески.

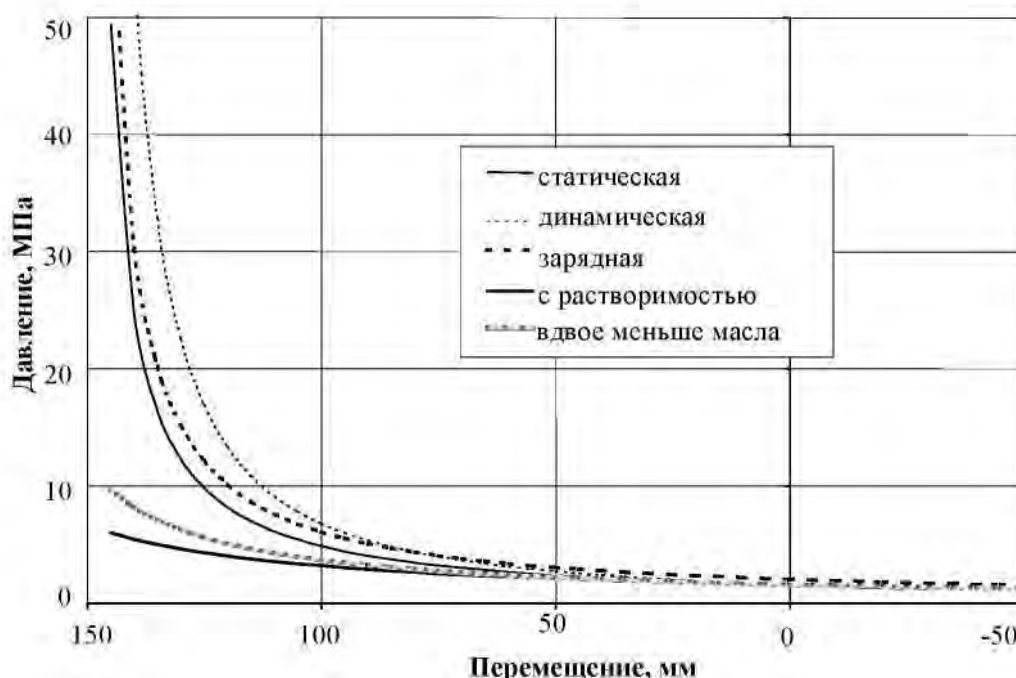


Рисунок 3.13. – Расчетные характеристики цилиндра

Состоянию груженого автомобиля соответствует давление в цилиндре около 10 МПа и перемещение около 125 мм по статической упругой

характеристике. Если нагрузка действует длительное время и происходит растворение газа, то при полном насыщении масла азотом увеличивается ход поршня и он максимально приближается к упору, что соответствует более жесткому участку упругой характеристики и может служить объяснением изменения свойств подвески в указанных условиях эксплуатации.

3.5 Колебательная модель колесной машины

3.5.1 При проведении расчетов плавности хода и нагрузок может использоваться как плоская модель, так и пространственная. Плоская модель более простая. Ее применение оправдано при расчетах плавности хода, тяговой динамики. Пространственная модель более сложная. Ее целесообразно применять при расчетах нагруженности рам, кабин, платформ, кузовов-контейнеров, устойчивости в реальных условиях движения, получения нагрузочных режимов, требующих учета кручения. Следует отметить, что общие схемы колебательной модели основной подвески (без кабины и сиденья) как легкового, так и грузового автомобиля практически ничем не отличаются. Отличие только в параметрах модели (см. рисунок 3.14) и их компонентов.

Для проведения моделирования разработана [74, 75, 77], показанная на рисунке 3.14, плоская модель автомобиля (колесной машины 6×6), учитывающая заднюю балансирную подвеску, кабину с поддрессированием спереди и сзади и поддрессоренное сиденье с моделью человека, обладающие новизной их связей.

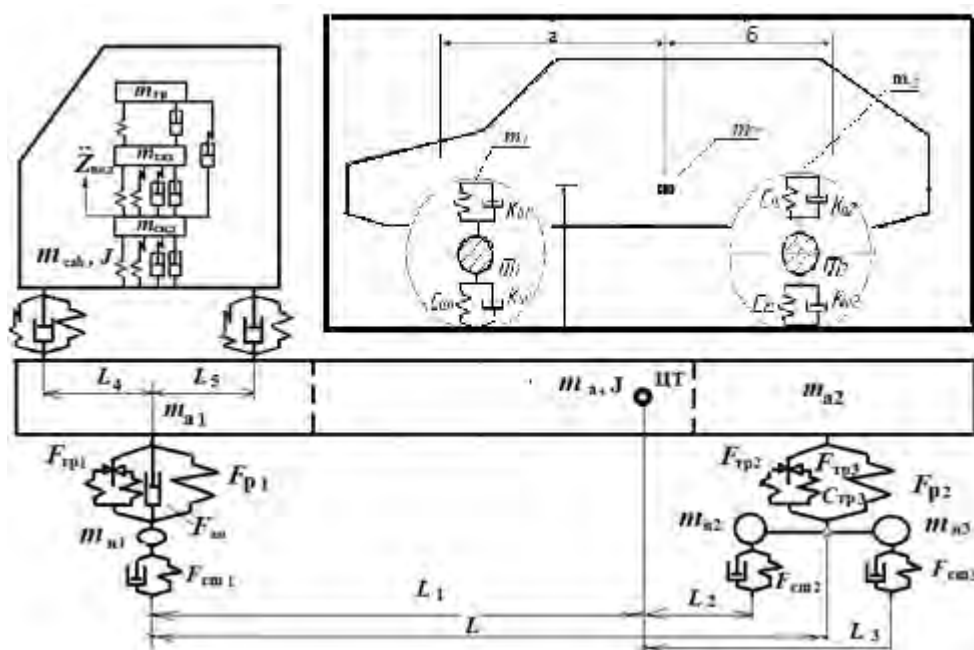


Рисунок 3.14. – Полная плоская модель автомобиля 6×6

Разработанную модель можно описать следующей системой дифференциальных уравнений:

$$\begin{aligned}
 m_a \ddot{Z}_0 - (F_{п1} + F_{п2}) + (F_{пcab1} + F_{пcab2}) &= 0; \\
 m_{cab} \ddot{Z}_{cab} - (F_{пcab1} + F_{пcab2} + F_{псид}) &= 0; \\
 m_a \rho_{y1}^2 \ddot{\beta}_1 + (F_{п1} L_1 - F_{п2} L_2) - (F_{пcab1} (L_1 + L_4) + F_{пcab2} (L_1 + L_5)) &= 0; \\
 m_{cab} \rho_{y1}^2 \ddot{\beta}_2 + (F_{пcab1} (L_1 + L_4) + F_{пcab2} (L_1 + L_5)) &= 0; \\
 m_{н1} \ddot{Z}_{н1} + (F_{ш1} - F_{п1}) &= 0; \\
 m_{н2} \ddot{Z}_{н2} + (F_{ш2} - F_{п2}/2) &= 0; \\
 m_{н3} \ddot{Z}_{н3} + (F_{ш2} - F_{п2}/2) &= 0;
 \end{aligned}$$

$$\begin{aligned}
 \beta_1 &= \frac{Z_1 - Z_2}{2L_1 + 2L_2}; \quad \beta_2 = \frac{Z_{cab1} - Z_{cab2}}{2L_5 + 2L_4}; \\
 \dot{\beta}_1 &= \frac{\dot{Z}_1 - \dot{Z}_2}{2b_{ш1} + 2b_{ш1}}; \quad \dot{\beta}_2 = \frac{\dot{Z}_{cab1} - \dot{Z}_{cab2}}{2L_5 + 2b_{ш4}};
 \end{aligned}$$

$$\begin{aligned}
 F_{ш1} &= F_{сш1} (q_{1л}) + F_{кш1} (\dot{q}_{1л}); \\
 F_{ш2} &= F_{сш2} (q_{2л}) + F_{кш2} (\dot{q}_{2л}); \\
 F_{ш3} &= F_{сш3} (q_{3л}) + F_{кш3} (\dot{q}_{3л}); \\
 F_{п1} &= F_{п1} (L_1 \beta_1 - L_2 \beta_1) - F_{ам1} (L_1 \dot{\beta}_1 - L_2 \dot{\beta}_1) + F_{тр1}; \\
 F_{п2} &= F_{п2} (L_2 \beta_2 - L_{п2} \frac{(\beta_{к2л} - \beta_{к3л})}{2}) + F_{ам2} (L_2 \dot{\beta}_2 + L_2 \frac{\dot{\beta}_{к2л} - \dot{\beta}_{к3л}}{2}) + F_{тр2},
 \end{aligned}$$

где $F_{сш1}, F_{сш2}, F_{сш3}$ – упругая характеристика шин переднего, среднего заднего моста в виде зависимости от относительной их деформации;

$F_{п1}, F_{п2}$, – упругая характеристика рессор переднего моста и балансирной подвески в виде зависимости от их относительной деформации;

$F_{тр1}, F_{тр2}$ – характеристика трения рессор переднего моста и балансирной подвески в виде зависимости интеграла от относительной скорости перемещения с ограничениями ее величины; $F_{ам1}$ – характеристика амортизаторов переднего моста в виде зависимости от относительной скорости перемещения;

$F_{п1}, F_{п2}$ – суммарная характеристика рессоры, амортизатора и трения переднего моста и балансирной подвески; m_a, m_{cab} – масса подрессоренной части автомобиля, кабины;

$m_{н1}, m_{н2}, m_{н3}$ – неподрессоренная масса переднего, среднего, заднего моста;

L_1, L_2 – расстояние между центром тяжести и передней и задней подвесками;

$J_y = m_a \rho_{y1}^2, J_{y2} = m_{cab} \rho_{y2}^2$ – момент инерции подрессоренной массы автомобиля, кабины в продольной плоскости.

На рисунке 3.15 приведена плоская модель автопоезда в составе двухосного тягача и двухосного полуприцепа.

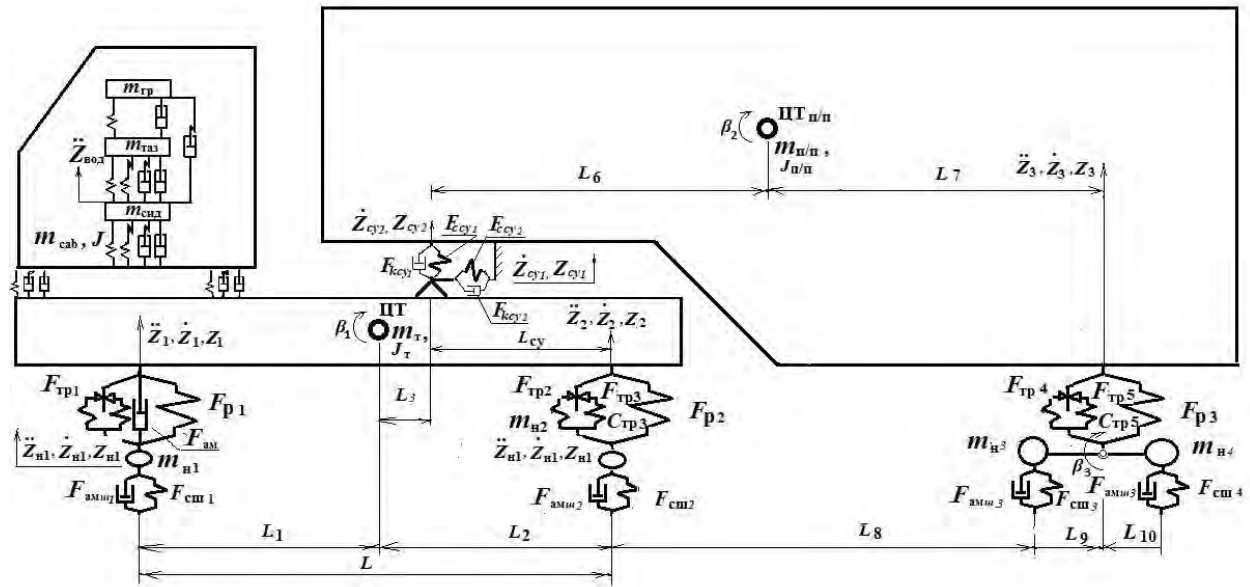


Рисунок 3.15. – Плоская модель автопоезда

Ее колебательную модель можно описать следующими уравнениями

$$m_a \ddot{Z}_T - (F_{п1} + F_{п2} - F_{cy}) - (F_{пcab1} + F_{пcab2}) = 0;$$

$$m_{пп} \ddot{Z}_{пп} - (F_{cy} + F_{пп}) = 0;$$

$$m_{cab} \ddot{Z}_{cab} - (F_{пcab1} + F_{пcab2} + F_{псид}) = 0;$$

$$m_{т} \rho_{y1}^2 \ddot{\beta}_1 + (F_{п1} L_1 - F_{п2} L_2) - (F_{пcab1} (L_1 + L_4) + F_{пcab2} (L_1 + L_5)) = 0;$$

$$m_{пп} \rho_{y2}^2 \ddot{\beta}_2 + (F_{cy} L_6 - F_{пп} L_7) = 0;$$

$$m_{cab} \rho_{y3}^2 \ddot{\beta}_3 + (F_{пcab1} (L_1 + L_4) + F_{пcab2} (L_1 + L_5)) = 0;$$

$$m_{нт1} \ddot{Z}_{н1} + (F_{н1} - F_{п1}) = 0;$$

$$m_{нт2} \ddot{Z}_{н2} + (F_{н2} - F_{п2}) = 0;$$

$$m_{н3} \ddot{Z}_{н3} + (F_{н3}/2 - F_{п3}/2) = 0;$$

$$m_{н4} \ddot{Z}_{н4} + (F_{н4}/2 - F_{п4}/2) = 0;$$

$$\beta_1 = \frac{Z_1 - Z_2}{2L_1 + 2L_2}; \dot{\beta}_1 = \frac{\dot{Z}_1 - \dot{Z}_2}{2b_{ш1} + 2b_{ш1}};$$

$$\beta_2 = \frac{Z_{cy} - Z_{пп}}{2L_1 + 2L_2}; \dot{\beta}_2 = \frac{\dot{Z}_{cy} - \dot{Z}_{пп}}{2b_{ш1} + 2b_{ш1}};$$

$$\beta_3 = \frac{Z_{ccab1} - Z_{ccab2}}{2L_5 + 2L_4}; \dot{\beta}_3 = \frac{\dot{Z}_{cab1} - \dot{Z}_{cab2}}{2L_5 + 2bL_4};$$

$$F_{ш1} = F_{см1} (q_{1л}) + F_{кш1} (\dot{q}_{1л});$$

$$F_{ш2} = F_{сш2} (q_{2л}) + F_{кш2} (\dot{q}_{2л});$$

$$F_{ш3} = F_{сш3} (q_{3п}) + F_{кш3} (\dot{q}_{3л});$$

$$F_{ш4} = F_{сш4} (q_{4л}) + F_{кш4} (\dot{q}_{4л});$$

$$F_{п1} = F_{р1} (Z_{н1} - L_2 \beta_1) - F_{ам1} (L_1 \dot{\beta}_1 - L_2 \dot{\beta}_1) + F_{тр1} (L_1 \dot{\beta}_1 - L_2 \dot{\beta}_1);$$

$$F_{п2} = F_{р2} (L_2 \beta_2 - L_{р2} \frac{(\beta_{к2л} - \beta_{к3л})}{2}) + F_{ам2} (L_2 \dot{\beta}_2 + L_2 \frac{\dot{\beta}_{к2л} - \dot{\beta}_{к3л}}{2}) + F_{тр2},$$

$$F_{п3} = F_{р3} (L_9 \beta_3 - L_{р3} \frac{(\beta_{к2л} - \beta_{к3л})}{2}) + F_{ам3} (L_9 \dot{\beta}_3 + L_9 \frac{\dot{\beta}_{к3л} - \dot{\beta}_{к4л}}{2}) + F_{тр4} + F_{тр5};$$

Для проведения расчетных исследований на основе рисунка 3.14 использована следующая модель подрессоренного сиденья, описанная в п.3.6.2 (рисунок 3.17а). Она учитывает параметры биодинамической модели человека, подушки и подвески сиденья. Особенностью и новизной данной модели является учет сложной модели трения в основной подвеске, подвеске кабины и подушке сиденья, сочетающей упругий и демпфирующий характер сил трения с ограничением их максимальных усилий. Это позволяет имитировать переменную жесткость в подвеске и подушке сиденья и лучше аппроксимировать виброзащитные свойства сиденья [14, 47, 48].

При моделировании для подвесок использовалась комбинированная модель трения [48, 51, 53] с упругими элементами и “вязкого” трения, которая более точно описывает динамическую характеристику рессорной подвески, кабины, сиденья и колебания при малых значениях возмущения.

Моделирование вибраций проводилось путем решения нелинейных дифференциальных уравнений с помощью пакета MatLab/Simulink.

Эффективность подрессоривания автомобиля оценивалась по среднеквадратическим ускорениям на сиденье водителя в третьоктавных полосах частот (1–25 Гц).

Параметры амортизатора кабины приняты: усилие сжатия 176 Н, усилие отбоя 1080 Н при скорости 0,25 м/с. Они соответствуют амортизаторам 20.5001010-03, 11.2905005-53, 53212-2905006-01, 5310-50001730, применяемым в подвесках кабины МАЗ и КамАЗ.

Жесткость передней и задней рессоры соответственно составляла 2000 и 4000 Н/м. Сопротивление амортизатора передней подвески при скорости 0,5 м/с: на сжатие 800 Н, на отбой 7500 Н.

Момент инерции автомобиля 6647 кг/м², подрессоренная масса автомобиля 17000 кг, масса кабины 750 кг.

Значения жесткости упругих элементов соответствуют собственным колебаниям кабины 1,5, 2,5, 4, 5 Гц.

Параметры сиденья соответствуют оптимальным параметрам, полученными в работе [47]: жесткость подвески 4000 кН/м, $F_{тр} = 25$ Н, $k_{ам} = 1500$ Н·с/м.

При моделировании подвесок использовалась комбинированная модель трения [14, 47, 48, 75-77] с упругими элементами и “вязкого” трения, которая более точно описывает динамическую характеристику рессорной подвески, сиденья и колебания при малых значениях возмущения и обеспечивает хорошую сходимость в октавных полосах частот.

Моделирование вибраций проводилось с помощью пакета MatLab/Simulink и модуля S-Function Builder, в которых осуществлялся расчет нелинейных дифференциальных уравнений численными методами.

3.6.2 Сиденье водителя

Акцентирование внимания к поддрессированию сиденья обусловлено тем, что на машинах повышенной проходимости применяются жесткие основные подвески, возможности совершенствования, которых ограничены. Поэтому на многих грузовых автомобилях, включая специальные, широко используются поддрессоренные сиденья. Их применение началось еще в конце 1960 годов прошлого столетия. Типичным представителем является сиденье Bostrom Viking-300 (рисунок 3.16а), применяемое до сих пор на автомобилях КамАЗ.

В качестве упругого элемента применяется торсион в виде двух пластин или горизонтально расположенные пружины. Преимуществами данной схемы являются ее компактность по высоте, простота, технологичность и дешевизна конструкции из-за применения штампованных деталей и сварки. Величина трения в подвеске без амортизатора составляет порядка 50 Н.

Данное сиденье использует рычажно-ножничную схему с торсионной подвеской с однотрубным газонаполненным амортизатором [14, 47]. Подобная схема конструкции применяется и на автомобилях МАЗ, МЗКТ. Данная схема оказалась удачной и она широко распространена на новейших сиденьях Bostrom Viking серии 301, 4000 и других фирм.

В отличие 300 (301) серии на сиденье 4000 серии применяется двухтрубный амортизатор ($F_{тр} = 1-2$ Н) [47], который в отличие от газонаполненного ($F_{тр} = 15$ Н) не увеличивает вибрации, но имеет пробои.

Помимо рычажно-ножничной схемы также широко применяются четырех-рычажные схемы подвесок (см. рисунок 3.17, б, д, е). Регулировка высоты сиденья осуществляется за счет поджатия пружины или давления в пневмоэлементе. Однако они имеют большую высоту и менее технологичны

из-за использования литого остова (из алюминия) или сварной конструкции и требуют механической обработки. В ряде конструкций (сиденье IFA 50W1) в осевых шарнирах использовались бронзовые втулки, что незначительно снизило трение (до 40 Н).



а – подрессоренное сиденье Viking-300; *в, г* – сиденья с ножнично-рычажной пневмоподвеской с рукавным элементом; *б, д, е* – с четырехрычажной схемой подвески

Рисунок 3.16. – Конструкции подрессоренных сидений

Сейчас все более широко начали использоваться сиденья с пневматической подвеской (см. рисунок 3.17, б–е) в первую очередь на автопоездах для обеспечения удобства их регулировок при смене водителей. Они имеют автономный компрессор, питающийся от бортовой сети 12 или 24 В. На них, как правило, устанавливается двухтрубный гидравлический амортизатор. Различие механических и пневматических систем подрессоривания сидений только в использованном упругом элементе (торсион, пружина или пневмоэлемент) поскольку для сидений с пневмоподвеской используются те же ножнично-рычажная и четырехрычажная схемы подвесок. С точки зрения виброзащитных свойств кинематические схемы подвесок одинаковы и выбор конкретной конструкции больше определяется компоновочными соображениями по кабине. Следует отметить, что подрессоренные сиденья вследствие своей кинематики защищают только от вертикальных колебаний и борются с продольными

вибрациями (8–22,4 Гц) можно только за счет компоновки и подбора параметров автомобиля (распределения поддрессоренных масс, базы автомобиля, смещения седельного устройства, снижения высоты центра тяжести, размеров шин и т. п.).

Применение поддрессоренных сидений обусловлено сильным увеличением вибраций на подушках (коэффициент передачи вибраций $K_{\text{п}}=3,5$) в полосе частот 3,5–5 Гц при отсутствии подвески [14, 46, 76].

На стендовых испытаниях [47] при постоянных синусоидальных амплитудах воздействия (10, 20 мм) сиденье Viking-4000 с двухтрубным амортизатором показывало неплохие виброзащитные свойства: ослабление вибраций начиналось с 2,5 Гц и на 4 Гц оно имело $K_{\text{п}}=0,5$. Однако в реальных дорожных условиях их эффективность оказалась гораздо хуже, особенно с газонаполненным амортизатором. Так, $K_{\text{п}}$ на сиденье КамАЗ (Viking-300) в октавных полосах частот 2, 4, 8, 16 Гц составлял соответственно 1,2–1,3; 0,96; 0,95; 0,75. Как выяснено автором в работах [32, 44] во многом это обусловлено газонаполненным амортизатором с большим внутренним трением (до 15 Н), работающим как дополнительный упругий элемент, и малыми амплитудами возмущения в средней и высокочастотной области (октавные полосы 4, 8, 16 Гц).

3.5.3 Модель поддрессоренного сиденья

При создании модели сиденья и водителя исходили из того, что сиденье защищает от вибраций только в вертикальном направлении. Продольные вибрации наблюдаются в области октавных полос 4, 8, 16 Гц. В этих областях, как правило, вибрации на автомобилях укладываются в допустимые нормы. Также необходимо учитывать, что существующие нормы предельно допустимых уровней вибраций получены на основе гармонического возмущения и субъективной оценки испытуемых. Из-за чего предлагаемые в ГОСТ 31191.1-2004 (ИСО 2631-1:1997) значения ощущения комфорта по приведенным вибрациям и диапазону разброса (от 0,315 до 0,63 м/с², оцениваемое как легкое ощущение дискомфорта, от 0,5 до 1 м/с² – приемлемое ощущение дискомфорта, от 0,8 до 1,6 м/с² – отчетливое ощущение дискомфорта) во многом условны и относительны. Поэтому, учитывая превышение допустимых норм по вертикальным вибрациям, в работе ориентировались только на них, их предельно допустимые уровни и относительный характер их изменения,

Исходя из этого предложена и использовалась следующая модель поддрессоренного сиденья, показанная на рисунке 3.17а, обладающая новизной

связей масс и используемого математического описания характеристик элементов. Применение модели (рисунок 3.17б) нецелесообразно, так как масса этих внутренних органов мала и дальнейшее увеличение количества масс не оправданно. Выбор предложенной модели основан на анализе работы автора [47]. Данная модель обеспечивает хорошую сходимость с экспериментом (погрешность 5–15 % в октавных полосах частот) [14, 75–78], имеет три подрессоренные массы: головы, груди и тела человека (при полной массе водителя 80 кг) и подрессоренной массы сиденья.

Математическая модель составлена для положения статического равновесия на основе принципа Д’аламбера.

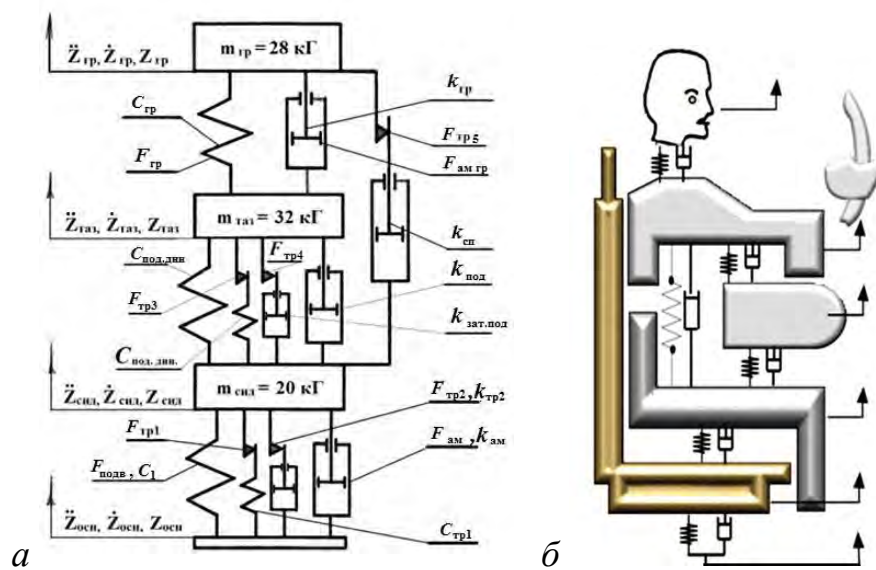


Рисунок 3.17. –Динамическая модель системы “сиденье-водитель”

Данная модель описывается следующими дифференциальными уравнениями:

$$\begin{aligned}
 m_{gp} \ddot{Z}_{gp} - (F_{gp} + F_{тр5}) &= 0; \\
 m_{таз} \ddot{Z}_{таз} + F_{gp} &= 0; \\
 m_{сид} \ddot{Z}_{сид} + (F_{подв} - F_{тр5}) &= 0; \\
 F_{gp} &= C_{gp} (Z_{таз} - Z_{gp}) + k_{gp} (\dot{Z}_{таз} - \dot{Z}_{gp}); \\
 F_{таз} &= C_{таз} (Z_{сид} - Z_{таз}) + k_{таз} (\dot{Z}_{сид} - \dot{Z}_{таз}) + F_{тр5} (\dot{Z}_{таз} - \dot{Z}_{сид}); \\
 F_{тр3} &= \begin{cases} C_{под,дин} \int_0^t (\dot{Z}_{сид} - \dot{Z}_{таз}) dt; & \text{если } F_{тр1} > C_{под,дин} \int_0^t (\dot{Z}_{сид} - \dot{Z}_{таз}) dt > -F_{тр3}; \\ F_{тр3}, & (-) \\ -F_{тр3}, & (-) \end{cases}
 \end{aligned}$$

$$F_{\text{тр4}} = \begin{cases} [k_{\text{зат.под}} (\dot{Z}_{\text{сид}} - \dot{Z}_{\text{таз}}), & \text{если } F_{\text{тр4max}} > k_{\text{зат.под}} (\dot{Z}_{\text{сид}} - \dot{Z}_{\text{таз}}) > -F_{\text{тр4max}}; \\ F_{\text{тр4max}}, & (-) \quad k_{\text{зат.под}} (\dot{Z}_{\text{сид}} - \dot{Z}_{\text{таз}}) > F_{\text{тр4max}}; \\ [-F_{\text{тр4max}}, & (-) \quad k_{\text{зат.под}} (\dot{Z}_{\text{сид}} - \dot{Z}_{\text{таз}}) < -F_{\text{тр4max}}; \end{cases}$$

$$F_{\text{тр5}} = \begin{cases} [k_{\text{сп}} (\dot{Z}_{\text{сид}} - \dot{Z}_{\text{гр}}), & \text{если } F_{\text{тр5max}} > k_{\text{сп}} (\dot{Z}_{\text{сид}} - \dot{Z}_{\text{гр}}) > -F_{\text{тр5max}}; \\ F_{\text{тр5max}}, & (-) \quad k_{\text{сп}} (\dot{Z}_{\text{сид}} - \dot{Z}_{\text{гр}}) > F_{\text{тр5max}}; \\ [-F_{\text{тр5max}}, & (-) \quad k_{\text{сп}} (\dot{Z}_{\text{сид}} - \dot{Z}_{\text{гр}}) < -F_{\text{тр5max}}; \end{cases}$$

$$F_{\text{ам.под}} = k_{\text{под}} (\dot{Z}_{\text{сид}} - \dot{Z}_{\text{таз}});$$

$$F_{\text{тр1}} = \begin{cases} [C_{\text{тр1}} \int_0^t (\dot{Z}_{\text{осн}} - \dot{Z}_{\text{сид}}) dt & \text{если } F_{\text{тр1}} > C_{\text{тр1}} \int_0^t (\dot{Z}_{\text{осн}} - \dot{Z}_{\text{сид}}) dt > -F_{\text{тр1max}}; \\ F_{\text{тр1max}}, & (-) \quad C_{\text{тр1}} \int_0^t (\dot{Z}_{\text{осн}} - \dot{Z}_{\text{сид}}) dt > F_{\text{тр1max}}; \\ [-F_{\text{тр1max}}, & (-) \quad C_{\text{тр1}} \int_0^t (\dot{Z}_{\text{осн}} - \dot{Z}_{\text{сид}}) dt < -F_{\text{тр1max}}; \end{cases}$$

$$F_{\text{тр2}} = \begin{cases} [k_{\text{тр2}} (\dot{Z}_{\text{осн}} - \dot{Z}_{\text{сид}}), & \text{если } F_{\text{тр2max}} > k_{\text{тр2}} (\dot{Z}_{\text{осн}} - \dot{Z}_{\text{сид}}) > -F_{\text{тр2max}}; \\ F_{\text{тр2max}}, & (-) \quad k_{\text{тр2}} (\dot{Z}_{\text{осн}} - \dot{Z}_{\text{сид}}) > F_{\text{тр2max}}; \\ [-F_{\text{тр2max}}, & (-) \quad k_{\text{тр2}} (\dot{Z}_{\text{осн}} - \dot{Z}_{\text{сид}}) < -F_{\text{тр2max}}; \end{cases}$$

$$F_{\text{ам.}} = k_{\text{ам}} (\dot{Z}_{\text{осн}} - \dot{Z}_{\text{сид}});$$

$$F_{\text{под.стат.}} = C_1 (Z_{\text{осн}} - Z_{\text{сид}}),$$

Где $C_{\text{под.стат.}}$, $C_{\text{под.дин}}$ – статическая и динамическая жесткость подушки;
 $k_{\text{зат.под.}}$, $k_{\text{ам}}$ – коэффициенты затуханий в подушке;
 $F_{\text{тр3}}$, $F_{\text{тр4}}$ – максимальные значения трения, ограничивающего усилия от динамической жесткости и демпфирующего элемента;

$\dot{Z}_{\text{осн}}$, $\dot{Z}_{\text{таз}}$, $\dot{Z}_{\text{гр}}$ – текущее значение виброскорости основания, таза и груди;

$Z_{\text{осн}}$, $Z_{\text{таз}}$, $Z_{\text{гр}}$ – текущее значение перемещений основания, таза и груди.

$\ddot{Z}_{\text{осн}}$, $\ddot{Z}_{\text{таз}}$, $\ddot{Z}_{\text{гр}}$ – текущее значение ускорений основания, таза и груди.

$F_{\text{подв}}$ – упругая характеристика подвески сиденья в виде зависимости от относительной деформации;

$F_{\text{тр1}}$, $F_{\text{тр3}}$ – характеристика трения подвески и подушки сиденья в виде зависимости интеграла от относительной скорости перемещения с ограничениями ее величины;

$F_{\text{тр2}}$, $F_{\text{тр4}}$, $F_{\text{тр5}}$ – характеристика “вязкого” трения подвески, подушки сиденья и трения о спинку в виде зависимости от относительной скорости перемещения с ограничениями ее величины;

$F_{\text{тр1max}}$, $F_{\text{тр2max}}$, $F_{\text{тр3max}}$, $F_{\text{тр4max}}$, $F_{\text{тр5max}}$ – максимальные значения трения подвески, подушки сиденья и трения о спинку;

$F_{\text{ам}}$ – характеристика амортизаторов подвески сиденья в виде зависимости от относительной скорости перемещения;

$m_{\text{гр}}$, $m_{\text{таз}}$, $m_{\text{сид}}$ – подрессоренная масса груди, таза, сиденья.

3.5.4 Модель водителя на спальном месте

В настоящее время все больше внимания уделяется вопросам безопасности движения автотранспортных средств. По данным американских исследователей [80] в 80% несчастных случаев водителей грузовиков со смертельным исходом является усталость и засыпание за рулем. Одной из мер противодействия этому является соблюдение регламента ЕСТР труда и отдыха водителей. Согласно ему продолжительность управления между любыми двумя периодами ежедневного отдыха или между ежедневным периодом отдыха и еженедельным периодом отдыха, именуемая ниже "ежедневная продолжительность управления", не должна превышать девяти часов (рисунок 3.18).

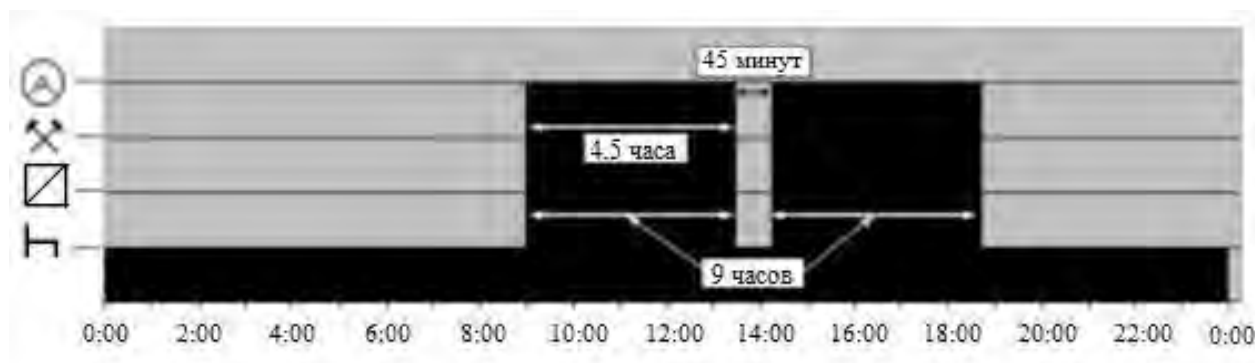


Рисунок 3.18. – График работы и отдыха водителя

После управления в течение четырех с половиной часов водитель должен сделать перерыв по крайней мере на сорок пять минут, если не наступает период отдыха (ст.7.1).

Если продолжительность рейса более 12 часов, то поездка должна осуществляться двумя водителями. ЕСТР рекомендует при 30 часовой поездке придерживаться следующего режима труда и отдыха (рисунок 3.20), сменяясь за рулем каждые 4,5 часа, после 19 часов поездки водители сменяются через 1 час, после чего должен быть обязательный непрерывный отдых 10 часов либо не менее 8 часов. В процессе движения в течение 4,5 часов невождения второй водитель находится в состоянии отдыха + готовности. В это время он может бодрствовать на месте пассажира либо спать, что не регламентировано ЕСТР. Большинство современных грузовых автомобилей оборудуются спальными местами, которые позволяют спать обоим водителям на стоянке

либо дают возможность второму водителю-сменщику отдохнуть в процессе движения (рисунок 3.20).



Рисунок 3.19. – График работы и отдыха при поездке двух водителей



Рисунок 3.20. – Спальное место современного грузового автомобиля

В качестве спального места используется специальный лежак с матрасом. Роль упругого и демпфирующего элемента матраса выполняет латексная губчатая резина 1-й и 2-й категории твердости, которая по данным исследований подушек значительно усиливает вибрации [47, 76].

Информация о вибрациях водителя на спальном месте во время отдыха при движении в открытых источниках отсутствует и исследование этого является новым так как водитель может проводить значительную часть времени в пути и на спальном месте и вибрации отрицательно сказываются на его состоянии.

С точки зрения биомеханики [14, 41, 47] тело человека можно рассматривать как колебательные системы с упругими связями. Собственные частоты внутренних органов приходятся на диапазон 3–6 Гц. При воздействии на человека внешних колебаний таких частот происходит возникновение резонансных явлений во внутренних органах, способных вызвать травмы, разрыв артерий, летальный исход. Собственные частоты колебаний плечевого

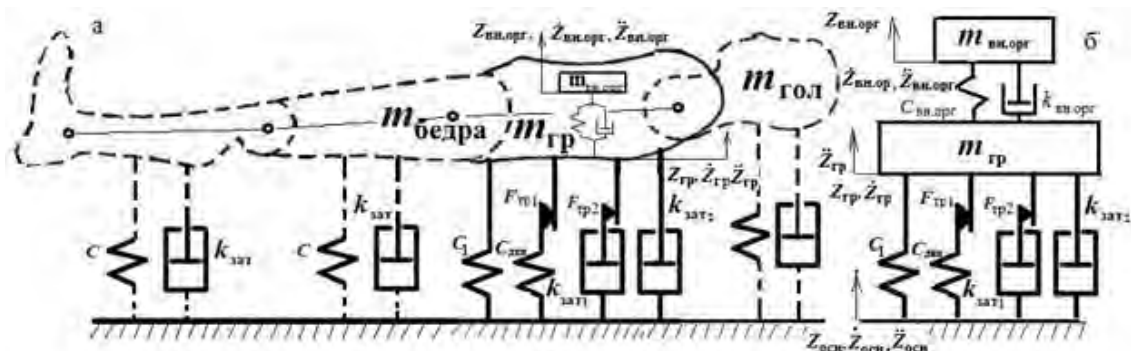
пояса, бедер и головы относительно опорной поверхности составляют 4–6 Гц, головы относительно плеч 25–30 Гц, глазных яблок 50 Гц. Собственные частоты колебаний тела в положении лежа составляют 3–6 Гц, в положении стоя 5–12 Гц, колебаний грудной клетки 5–8 Гц. Воздействуя на человека, вибрация угнетает центральную нервную систему, вызывая чувство тревоги и страха, ухудшает зрение и качество работы человека-оператора и физиологическое и функциональное состояния организма человека. Это проявляется в повышении утомляемости, увеличении времени двигательной и зрительной реакции, нарушении вестибулярных реакций и координации движений.

Вибрации отрицательно влияют на отдых водителя, возможность сна и естественно на безопасность движения и данных по ним нет. Поэтому представляет интерес оценить уровень вибраций на спальном месте с учетом их третьоктавного спектра путем моделирования. Это позволит рассмотреть выбор мер по их снижению.

3.5.5 Колебательная модель водителя на спальном месте

Водителя на спальном месте можно представить в виде следующей модели с шарнирами (рисунок 3.21), обладающей новизной связей элементов.

Наибольший интерес представляет часть тела, относящаяся к груди. Здесь расположены сердце, печень, почки, которые наиболее чувствительны к колебаниям и имеют резонанс в области 5–7 Гц. По данным работ [41, 51] эту часть тела можно представить в виде массы $m_{гр}$, соединенной с пружиной и амортизатором. Данных по модели головы в горизонтальном положении нет. Нога человека в виде бедра и голени не является чувствительным к вибрациям элементом. Хотя они и совершают колебания, однако, учитывая шарнирную связь, их в колебательной модели можно не рассматривать.



а – полная модель водителя на спальном месте, б – упрощенная модель

Рисунок 3.21. – Модель водителя на спальном месте

То есть при рассмотрении модели человека в горизонтальном состоянии лежа можно ограничиться двухмассовой моделью груди и внутренних органов (сердца, печени, почек) (см. рисунок 3.20б), что является новым. Масса последних в сумме составляет 2 кг. Приняты следующие колебательные параметры для линейной модели внутренних органов, соответствующие собственной частоте их колебаний 7 Гц: $c_{\text{вн.орг}} = 3864 \text{ Н/м}$, $k_{\text{вн.орг}} = 31 \text{ Н·с/м}$.

Подушку спального места можно представить в виде двух пружин, усилие одной из которых ограничено величиной трения и двух амортизаторов, усилие одного из них ограничено величиной трения.

Математическую модель колебаний и усилия упругих, демпфирующих элементов прокладки спального места можно описать следующими уравнениями, базируясь на анализе работ [14, 47]

$$\begin{aligned}
 m_{\text{вн.орг}} \ddot{Z}_{\text{вн.орг}} + c_{\text{вн.орг}} (\dot{Z}_{\text{вн.орг}} - \dot{Z}_{\text{гр}}) + k_{\text{вн.орг}} (Z_{\text{вн.орг}} - Z_{\text{гр}}) &= 0; \\
 m_{\text{гр}} \ddot{Z}_{\text{гр}} + F_1 + F_{\text{тр1}} + F_{\text{тр2}} &= 0; \\
 F_{\text{тр1}} = \begin{cases} [C_{\text{дин}} \int_0^t (\dot{Z}_{\text{осн}} - \dot{Z}_{\text{гр}}) dt, & \text{если } F_{\text{тр1}} > c_{\text{дин}} \int_0^t (\dot{Z}_{\text{осн}} - \dot{Z}_{\text{гр}}) dt > -F_{\text{тр1}}; \\ F_{\text{тр1}}, & (-) \quad c_{\text{дин}} \int_0^t (\dot{Z}_{\text{осн}} - \dot{Z}_{\text{гр}}) dt > F_{\text{тр1}}; \\ -F_{\text{тр1}}, & (-) \quad c_{\text{дин}} \int_0^t (\dot{Z}_{\text{осн}} - \dot{Z}_{\text{гр}}) dt < -F_{\text{тр1}}; \end{cases} \\
 F_{\text{тр2}} = \begin{cases} [k_{\text{зат1}} \int_0^t (\dot{Z}_{\text{осн}} - \dot{Z}_{\text{гр}}) dt & \text{если } F_{\text{тр2}} > k_{\text{зат1}} \int_0^t (\dot{Z}_{\text{осн}} - \dot{Z}_{\text{гр}}) dt > -F_{\text{тр2}}; \\ F_{\text{тр2}}, & (-) \quad k_{\text{зат1}} \int_0^t (\dot{Z}_{\text{осн}} - \dot{Z}_{\text{гр}}) dt > F_{\text{тр2}}; \\ -F_{\text{тр2}}, & (-) \quad k_{\text{зат2}} \int_0^t (\dot{Z}_{\text{осн}} - \dot{Z}_{\text{гр}}) dt < -F_{\text{огр}}; \end{cases} \\
 F_1 = c_1 (Z_{\text{осн}} - Z_{\text{гр}}); \quad F_{\text{зат2}} = k_{\text{зат2}} (\dot{Z}_{\text{осн}} - \dot{Z}_{\text{гр}}),
 \end{aligned}$$

где c_1 , $c_{\text{дин}}$ – жесткость прокладки подушки, определенная при статическом и динамическом нагружении;

$k_{\text{зат1}}$, $k_{\text{зат2}}$ – коэффициенты затуханий в прокладке;

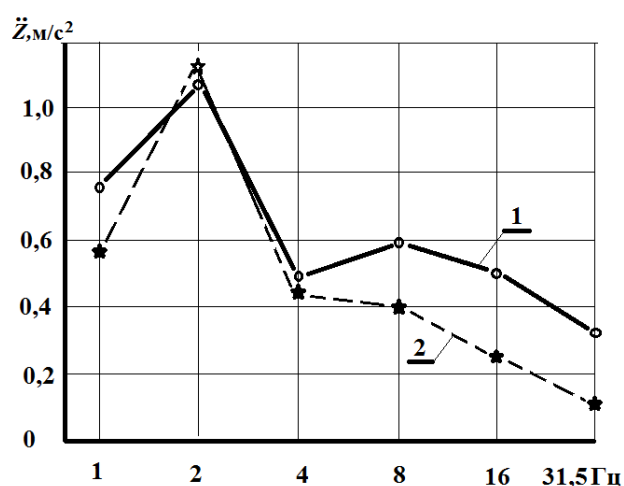
F_1 , $F_{\text{тр1}}$, $F_{\text{тр2}}$ – усилия от жесткости прокладки подушки, максимальные значения трения, ограничивающего усилия от динамической жесткости и демпфирующего элемента; $m_{\text{гр}}$, $m_{\text{вн.орг}}$ – груди, внутренних органов; $\dot{Z}_{\text{осн}}$, $\dot{Z}_{\text{гр}}$, $\dot{Z}_{\text{вн.орг}}$ – текущее значение виброскорости основания, груди, внутренних органов; $Z_{\text{осн}}$, $Z_{\text{гр}}$, $Z_{\text{вн.орг}}$ – текущее значение перемещений основания, груди, внутренних органов.

Значения $Z_{\text{осн}}$ и $\dot{Z}_{\text{осн}}$ берутся из колебательной модели автомобиля.

3.5.6 Подрессоривание кабины

Грузовые автомобили с компоновкой “кабина над двигателем” имеют повышенный уровень вибраций на рабочем месте водителя по сравнению с классической – кабина за двигателем. Это связано, как с более высоким расположением водителя, так и с особенностями этой компоновки, за счет которой стремятся увеличить нагрузку на переднюю ось и повысить грузоподъемность автомобиля. Последнее часто приводит к ухудшению коэффициента распределения поддрессоренных масс. Особенно неблагоприятная ситуация по вибрациям складывается с седельными тягачами, у которых седельное устройство смещено вперед по отношению к задней подвеске и сильное воздействие оказывает полуприцеп [45, 77, 81], а также с автомобилями повышенной проходимости. Поэтому для этих грузовых автомобилей используются все возможные способы снижения вибронагруженности рабочего места водителя, включая применение поддрессоривания кабины и сиденья.

Минский автомобильный завод (МАЗ), базируясь на опыте зарубежных фирм и на основании расчетных исследований в институте математики АН БССР в 1980 г. в процессе создания семейства МАЗ-6422 [81], ожидал путем поддрессоривания кабины (рисунок 3.22) добиться снижения вибраций в первую очередь в октавных полосах частот 2, 4 Гц, уровни вибраций на опытных образцах превышали санитарные нормы [47, 14, 81].



1 – без поддрессоривания кабины; 2 – с поддрессориванием кабины

Рисунок 3.22. – Вибрации под сиденьем при поддрессоривании кабины (2) и без него (1) при движении по булыжному шоссе со скоростью $v = 70$ км/ч

Реальный эффект от подрессоривания кабины с пружинной подвеской в 1984–1985 гг. получился следующим: уровень вибраций в октавных полосах частот 2, 4 Гц на сиденьи с газонаполненным амортизатором оказался выше на 5–10 % чем без подрессоривания кабины. И лишь в октавных полосах частот 8, 16 Гц наблюдалось небольшое снижение вибраций на 10–20 %. Снижение вибраций на полу кабины в октавных полосах частот 8, 16 Гц оказалось больше (30–40 %). Обеспечить соответствие санитарным нормам на седельных тягачах в дальнейшем удалось за счет применения малолистовой рессоры с малым трением, использования импортного подрессоренного сиденья с пневмоподвеской и проведения испытаний на участке дороги с более совершенным покрытием. При этом у малолистовой рессоры был меньший статический прогиб (160 мм), чем у многолистовой (180–190 мм). Последнее обстоятельство указывает на значительное влияние снижения трения в рессорной подвеске.

В настоящее время на автомобилях нового семейства МАЗ-6430 применена пневматическая подвеска кабины (рисунок 3.23) с использованием импортных элементов со встроенным гидравлическим двухтрубным амортизатором, обеспечивающая более низкую собственную частоту колебаний кабины.

По логике это должно привести к снижению вибраций, но этому пока нет экспериментального подтверждения. Пробои подвески кабины при переездах трамвайных путей и больших неровностей, ям, волн на дорогах отрицательно воспринимаются водителем и вынуждают снижать скорость движения. На данный момент у разработчиков нет единого мнения и четкого понимания, какими путями добиваться снижения вибраций.

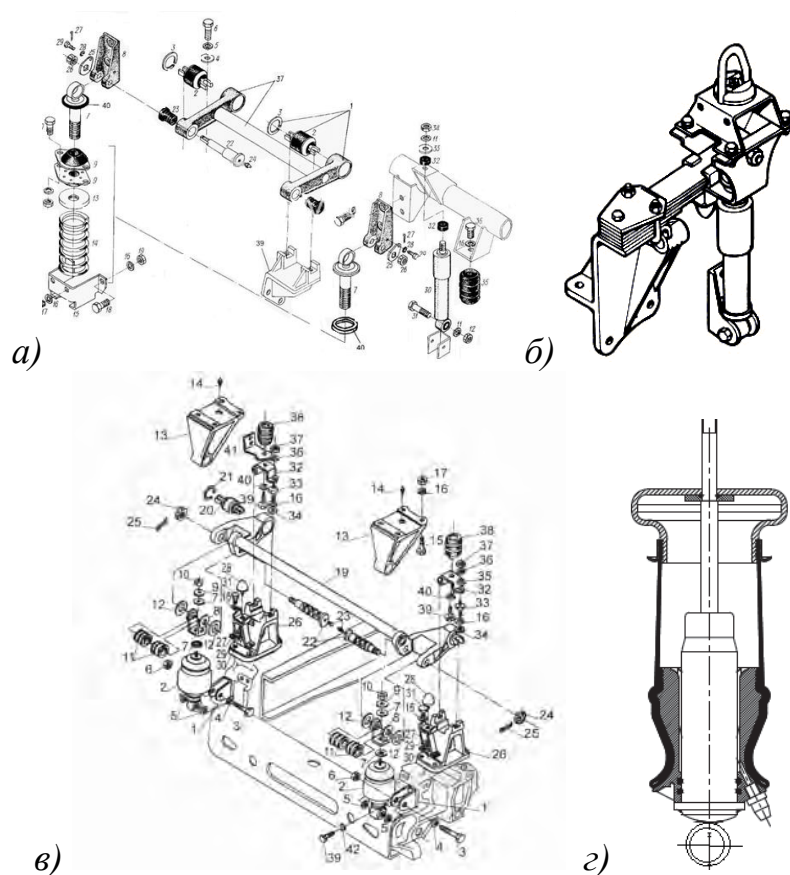


Рисунок 3.23. – Конструкции подвесок кабины автомобилей МАЗ-6422 (а), КамАЗ-4310 (б), МАЗ-6430 с пневмоподвеской (в), пневмоэлемент с амортизатором (г)

3.6 Пространственная модель колебательной системы колесной машины

С учетом, что для ряда узлов автомобиля определяющими являются крутящие нагрузки разработана пространственная динамическая модель автомобиля 6×6 [26, 83]. Данная схема, показанная на рисунках 3.24 и 3.25, широко применяется на самосвалах, пожарных, штабных машинах и БТР.

Предложенная пространственная модель отличается от известных (в виде единого тела) (рисунок 3.25 справа) наличием двух приведенных подрессоренных масс $m_{п1}$, $m_{п2}$ в продольной плоскости, которые могут совершать как вертикальные, так и продольно-угловые и поперечно-угловые колебания. Также учтена кинематика задней балансирной подвески.

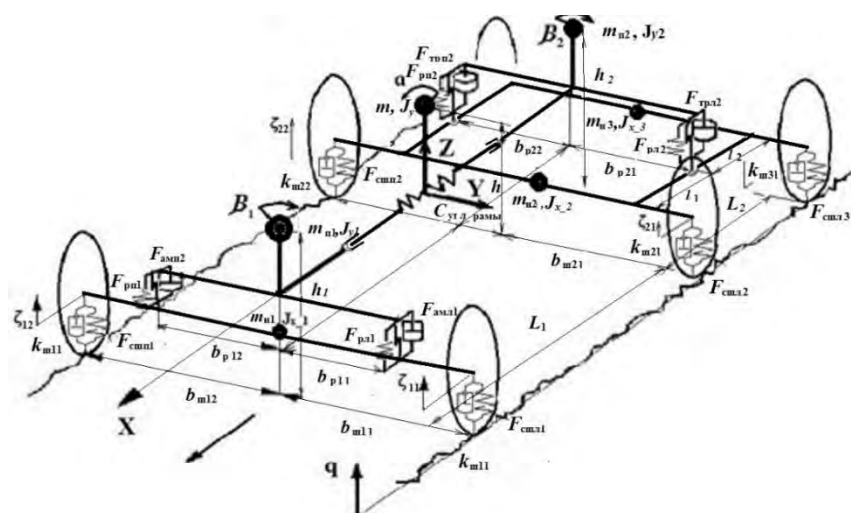


Рисунок 3.24. – Пространственная динамическая модель колесной машины на базе шасси 6×6

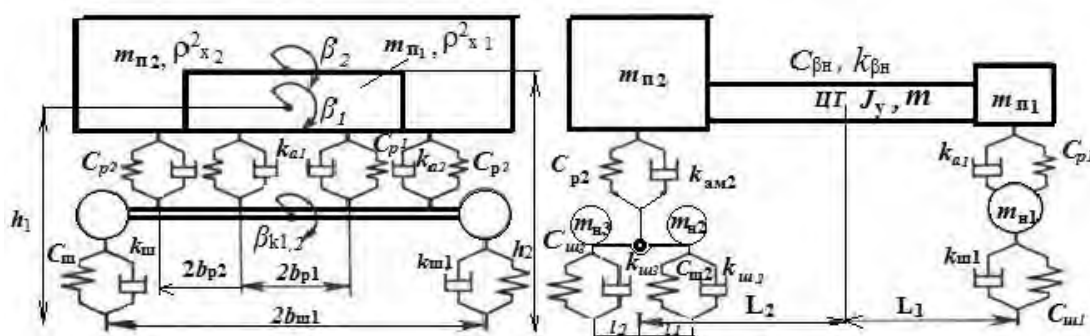


Рисунок 3.25. – Упрощенная схема колебательной системы колесной машины 6×6

Динамическая модель учитывает упругие и демпфирующие характеристики шин, рамы, передних, средних, задних подвесок, суммарную податливость рамы с контейнером, кинематику подвески. Это позволяет моделировать как вертикальные, так и поперечно-угловые колебания колесной машины и определять силы и моменты, действующие на раму, платформу, кабину и кузов-контейнер.

Модель может быть описана системой дифференциальных уравнений:

$$\begin{aligned}
 m_{n1} \ddot{Z}_1 + (F_{n11} + F_{n12}) + m_{n3} \ddot{Z}_2 &= 0; \\
 m_{n2} \ddot{Z}_2 + (F_{n21} + F_{n22}) + m_{n3} \ddot{Z}_1 &= 0; \\
 m_{n1} \rho_{x1}^2 \ddot{\beta}_1 + (F_{n11} - F_{n12}) b_{p1} + M_{\beta n} &= 0; \\
 m_{n2} \rho_{x2}^2 \ddot{\beta}_2 + (F_{n21} - F_{n22}) b_{p2} + M_{\beta n} &= 0; \\
 m_{n1} \rho_{x1}^2 \ddot{\beta}_{k1} + (F_{n11} - F_{n12}) b_{p1} + (F_{n11} - F_{n12}) b_{m1} &= 0; \\
 m_{n2} \rho_{x2}^2 \ddot{\beta}_{k2} + (F_{n21} - F_{n22}) b_{p2} + (F_{n21} - F_{n22}) b_{m2} &= 0;
 \end{aligned}$$

$$m_{\text{ш3}} \rho_{\text{ш3}}^2 \ddot{\beta}_{\text{k3}} + (F_{\text{пл3}} - F_{\text{шп3}}) b_{\text{p3}} + (F_{\text{шл3}} - F_{\text{шп3}}) b_{\text{ш3}} = 0;$$

$$\beta_{\text{k1}} = \frac{\zeta_{1\text{л}} - \zeta_{1\text{п}}}{2b_{\text{ш1}}};$$

$$\beta_{\text{k2}} = \frac{\zeta_{2\text{л}} - \zeta_{2\text{п}}}{2b_{\text{ш2}}};$$

$$\beta_{\text{k3}} = \frac{\zeta_{3\text{л}} - \zeta_{3\text{п}}}{2b_{\text{ш3}}};$$

$$\dot{\beta}_{\text{k1}} = \frac{\dot{\zeta}_{1\text{л}} - \dot{\zeta}_{1\text{п}}}{2b_{\text{ш1}}};$$

$$\dot{\beta}_{\text{k2}} = \frac{\dot{\zeta}_{2\text{л}} - \dot{\zeta}_{2\text{п}}}{2b_{\text{ш2}}};$$

$$\dot{\beta}_{\text{k3}} = \frac{\dot{\zeta}_{3\text{л}} - \dot{\zeta}_{3\text{п}}}{2b_{\text{ш3}}};$$

$$M_{\beta\text{н}} = c_{\beta\text{н}} b_{\text{p1}}^2 (\beta_1 - \beta_2) + k_{\beta\text{н}} b_{\text{p2}}^2 (\dot{\beta}_1 - \dot{\beta}_2);$$

$$F_{\text{шл1}} = F_{\text{сшл1}} (b_{\text{k1}} \beta_{\text{k1}} - q_{1\text{л}}) + F_{\text{кшл1}} (b_{\text{k1}} \dot{\beta}_{\text{k1}} - \dot{q}_{1\text{л}});$$

$$F_{\text{шп1}} = -F_{\text{сшп1}} (b_{\text{k1}} \beta_{\text{k1}} - q_{1\text{п}}) + F_{\text{кшп1}} (b_{\text{k1}} \dot{\beta}_{\text{k1}} - \dot{q}_{1\text{л}});$$

$$F_{\text{шл2}} = F_{\text{сшл2}} (b_{\text{k2}} \beta_{\text{k2}} - q_{2\text{л}}) + F_{\text{кшл2}} (b_{\text{k2}} \dot{\beta}_{\text{k2}} - \dot{q}_{2\text{л}});$$

$$F_{\text{шп2}} = -F_{\text{сшп2}} (b_{\text{k2}} \beta_{\text{k2}} - q_{2\text{п}}) + F_{\text{кшп2}} (b_{\text{k2}} \dot{\beta}_{\text{k2}} - \dot{q}_{2\text{л}});$$

$$F_{\text{шл3}} = c_{\text{шл3}} (b_{\text{k1}} \beta_{\text{k3}} - q_{3\text{п}}) + F_{\text{кшл3}} (b_{\text{k1}} \dot{\beta}_{\text{k3}} - \dot{q}_{3\text{л}});$$

$$F_{\text{шп3}} = -c_{\text{шп3}} (b_{\text{k3}} \beta_{\text{k3}} - q_{3\text{п}}) + F_{\text{кшп3}} (b_{\text{k2}} \dot{\beta}_{\text{k3}} - \dot{q}_{3\text{л}});$$

$$F_{\text{пл1}} = F_{\text{рл1}} (b_{\text{p1}} \beta_1 - b_{\text{p1}} \beta_{\text{k1}}) - F_{\text{амл1}} (b_{\text{p1}} \dot{\beta}_1 - b_{\text{p1}} \dot{\beta}_{\text{k1}});$$

$$F_{\text{пп1}} = -F_{\text{рл1}} (b_{\text{p1}} \beta_1 - b_{\text{p1}} \beta_{\text{k1}}) - F_{\text{амл1}} (b_{\text{p1}} \dot{\beta}_1 - b_{\text{p1}} \dot{\beta}_{\text{k1}});$$

$$F_{\text{пл2}} = F_{\text{рл2}} (b_{\text{p2}} \beta_2 - b_{\text{p2}} \frac{(\beta_{\text{k2л}} - \beta_{\text{k3л}})}{2}) + F_{\text{амл2}} (b_{\text{p2}} \dot{\beta}_2 + b_{\text{p2}} \frac{\dot{\beta}_{\text{k2л}} - \dot{\beta}_{\text{k3л}}}{2});$$

$$F_{\text{пп2}} = -F_{\text{рл2}} (b_{\text{p2}} \beta_2 - b_{\text{p2}} \frac{(\beta_{\text{k2л}} - \beta_{\text{k3л}})}{2}) + F_{\text{амп2}} (b_{\text{p2}} \dot{\beta}_2 + b_{\text{p2}} \frac{\dot{\beta}_{\text{k2п}} - \dot{\beta}_{\text{k3п}}}{2}),$$

где $F_{\text{шп1}}, F_{\text{шл1}}$ – упругая характеристика правой (левой) шины переднего моста в виде зависимости от относительной их деформации; $F_{\text{шп2}}, F_{\text{шл2}}$ – упругая характеристика правой (левой) шины среднего моста в виде зависимости от их относительной деформации; $F_{\text{шп3}}, F_{\text{шл3}}$ – упругая характеристика правой (левой) шины третьего моста в виде зависимости от их относительной деформации;

$F_{\text{рп1}}, F_{\text{рл1}}$ – упругая характеристика правой (левой) рессоры переднего моста в виде зависимости от их относительной деформации; $F_{\text{трп1}}, F_{\text{трл1}}$ – характеристика трения правой (левой) рессоры переднего моста в виде зависимости от относительной скорости перемещения; $F_{\text{амп1}}, F_{\text{амл1}}$ – характеристика правого (левого) амортизатора переднего моста в виде зависимости от относительной скорости перемещения; $F_{\text{рп2}}, F_{\text{рл2}}$ – упругая характеристика правой (левой) рессоры задней тележки в виде зависимости от их относительной деформации; $F_{\text{трп2}}, F_{\text{трл2}}$ – характеристика трения правой

(левой) рессоры задней тележки в виде зависимости от относительной скорости перемещения; $F_{\text{пш1}}, F_{\text{пш1}}$ – суммарная характеристика рессоры, амортизатора и трения переднего моста; $F_{\text{пш2}}, F_{\text{пш2}}$ – суммарная характеристика рессоры и трения задней тележки;

$m_{\text{п1}}$ – приведенная масса передней подрессоренной части колесной машины,

$m_{\text{п1}} = m \frac{L_2^2 + p_y^2}{L^2}$; $m_{\text{п2}}$ – приведенная масса задней подрессоренной части колесной

машины, $m_{\text{п2}} = m \frac{L_1^2 + p_y^2}{L^2}$; $m_{\text{п3}}$ – приведенная масса связи передней и задней

подрессоренной части колесной машины, $m_{\text{п3}} = m \frac{l_2^2 l_1^2 + p_y^2}{L^2}$;

$m_{\text{н1}}$ – неподрессоренная масса переднего моста колесной машины;

$m_{\text{н2}}$ – неподрессоренная масса среднего моста колесной машины;

$m_{\text{н3}}$ – неподрессоренная масса заднего моста колесной машины;

$c_{\beta\text{н}}$ – жесткость рамы и контейнера на кручение,

$k_{\beta\text{н}}$ – коэффициент демпфирования рамы и контейнера при кручении;

$b_{\text{кр1}}, b_{\text{кр2}}$ – расстояние между рессорами, амортизаторами;

$b_{\text{ш1}}, b_{\text{ш2}}$ – колея (расстояние) между шинами;

J_y – момент инерции подрессоренной массы колесной машины в продольной плоскости; m – подрессоренная масса колесной машины в продольной плоскости;

$J_{x1} = m_1 \rho_{x1}^2$ – момент инерции неподрессоренных масс переднего моста в поперечной плоскости;

$J_{x2} = m_2 \rho_{x2}^2$ – момент инерции неподрессоренных масс среднего моста в поперечной плоскости.

$J_{x3} = m_3 \rho_{x3}^2$ – момент инерции неподрессоренных масс заднего моста в поперечной плоскости.

Новизна и отличие предложенной модели и системы дифференциальных уравнений от других моделей [26, 81] заключается в описании пространственной модели автомобиля с учетом ее сложных связей и особенностей. Представленная модель в процессе исследований реализована в пакете MSC.ADAMS для решения задачи оценки поперечной устойчивости путем воспроизведения процесса «опрокидывания на стенде» и получения значений максимальных нагрузок и ускорений, действующих на штабную машину и кузов-контейнер при переезде препятствий.

3.7 Модель дизель-генераторной установки и ее воздействие на вибрации штабной машины

3.7.1 Особенностью штабных машин является применение автономной дизель-генераторной установки (ДГУ), размещенной в кузове-контейнере

(рисунок 3.26). Она в основном используется, когда автомобиль не движется. Данная дизель-генераторная установка (7–20 кВт) устанавливается на специальную рамку, разработанную изготовителем ДГУ, через четыре виброгасящие резиновые опоры. Для ДГУ (до 10-15 кВт) обычно используется одноцилиндровый двигатель.



Рисунок 3.26. – Общий вид дизель-генераторной установки, компоновка ее в кузове-контейнере штабной машины

Параметры этих виброопор обычно выбирает изготовитель ДГУ, исходя из оптимального виброгашения колебаний при стационарном режиме работы установки (1500 мин^{-1} , 25 Гц), закрепляемой на жестком массивном фундаменте. Это не всегда учитывает специфические условия ее эксплуатации. Кроме того, в ряде случаев изготовителями кузовов-контейнеров применяется еще дополнительное подрессоривание ДГУ с ее рамкой через дополнительные виброопоры, размещенные под этой рамкой.

В практике имел место единичный случай, когда из-за несогласованных конструктивных решений сопоставщиков (поставщиков ДГУ) и неучета параметров основной подвески автомобиля наблюдались резонансные вибрации при работающей ДГУ на полу кузова-контейнера при неподвижном автомобиле. И хотя уровень вибраций не превышал допустимые санитарные нормы, он ухудшал комфортные условия для персонала, работающего с компьютерной техникой и в процессе отдыха. Для уменьшения вибраций потребовалось применить виброопоры большей жесткости, поскольку использование виброопор уменьшенной жесткости приводило к еще большему увеличению вибраций.

Анализ большого количество работ по динамике двигателей [90–92] показал, что практически полностью отсутствуют работы по оценке воздействия работающей ДГУ на вибрации грузового автомобиля с кузовом-контейнером и размещенную в них аппаратуру и персонал. Имеющиеся работы по двигателям внутреннего сгорания (ДВС) больше посвящены

процессам наполняемости цилиндров, сгорания топлива, системам управления двигателей, а также колебаниям в трансмиссии. В основном это исследования по двигателям легковых автомобилей (4-х цилиндровых бензиновых) [90, 91] и морских судов (многоцилиндровых дизельных) [92]. Результаты этих экспериментальных исследований свидетельствуют, что вращение коленвала отличается от равномерного. Динамика двигателя имеет сложный характер из-за наполняемости цилиндров, наличия зазоров, трения и взрывного процесса сгорания топлива [90, 91]. Моделирование двигателей проводится, как правило, с помощью пакета Matlab/Simulink или методов линеаризации. Следует отметить, что использование численных методов интегрирования при уменьшении шага для охвата широкого диапазона частот (28–50 Гц и более) не обеспечивает достаточной точности или приводит к остановке процесса вычислений, а использование методов линеаризации не позволяет учесть многие нелинейности. Исходя из практики расчетов колебаний автомобиля и движения, более или менее адекватно (с погрешностью 10–15 %) можно смоделировать процесс колебаний только в диапазоне 0–28 Гц [14, 47]. В силу вышеперечисленного из-за сложности процессов и разнообразия конструкций двигателей при моделировании колебаний в системе “автомобиль–ДГУ” целесообразно ориентироваться на упрощенную модель, учитывающую преобладающие инерционные силы ДГУ.

Параметры демпфирования виброопор можно рассчитать, используя выведенные в работах автора [47, 14, 76] формулы для подушек сидений, учитывая их идентичный характер изменения.

3.7.2 Динамическая и математическая модель колебательной системы с ДГУ

Для проведения данного исследования использовался метод компьютерного моделирования. Применена следующая модель, представленная на рисунке 3.27).

Схема и предложенное математическое описание в совокупности с моделью поддрессирования автомобиля являются новыми [93].

Использование плоской модели обуславливалось поперечным расположением ДГУ и применением одинаковых виброопор как спереди, так и сзади на установке. Это позволяет значительно упростить модель. Важное значение имеет, какую модель трения в подвеске использовать, учитывая небольшие амплитуды колебаний, происходящие в средне и высокочастотной области (4–28 Гц).

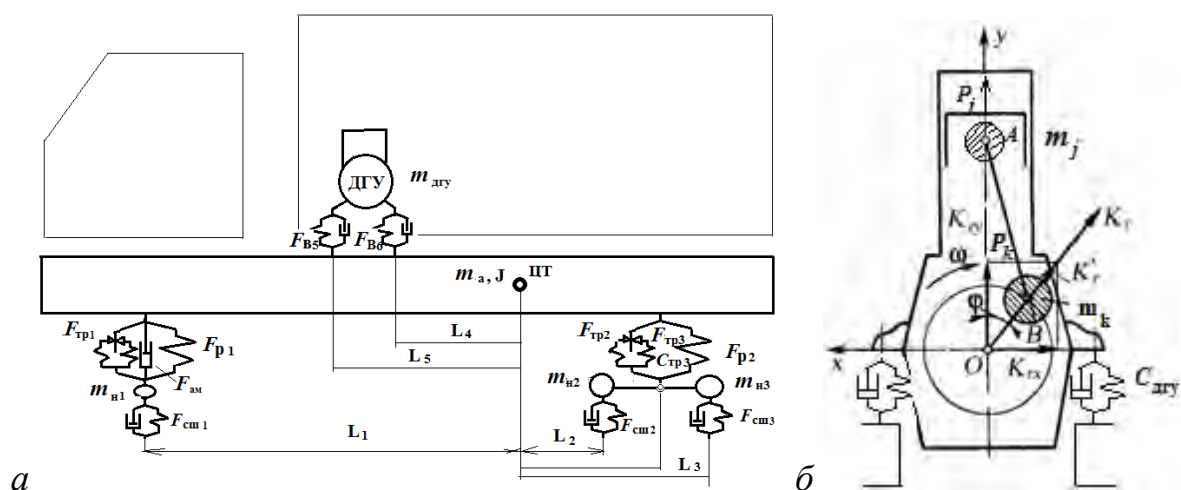


Рисунок 3.27. – Колебательная модель автомобиля (а)
и системы виброизоляции ДГУ (б)

Применена модель упругого трения [48], показанная на рисунке 3.28, которая более точно описывает динамическую характеристику подвески и колебания при малых значениях возмущения.

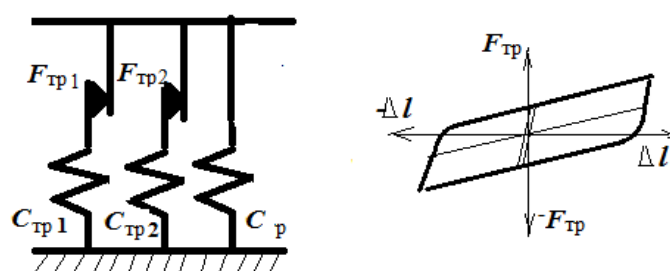


Рисунок 3.28. – Используемая модель трения и ее характеристика

Данная модель представляет собой совокупность последовательно включенных упругих элементов, максимальные значения усилий, которые ограничиваются трением. Благодаря этому реализуется переменность жесткости рессоры от величин ее относительного прогиба и скорости.

Схема динамического воздействия ДГУ представлена на рисунке 3.27б.

Для упрощения модели принято равномерное вращение коленвала, силы динамического воздействия от рабочих процессов в двигателе из-за их сложности не рассматривались. Учитывались преобладающие силы инерции движущихся масс, которые были сведены к двум силам: силе инерции возвратно-поступательно движущейся приведенной массы поршня $m_{\text{п}}$ ($P_{\text{п}}$) и центробежной силе инерции вращающихся масс коленвала $m_{\text{к}}$ ($P_{\text{к}}$). Последние согласно [91] можно описать следующими уравнениями.

$$P_j = -m_r j_{\Pi} = -m_j r \omega^2 \left(\cos \varphi + \frac{r}{l} \cos 2 \varphi \right);$$

$$P_k = +m_r r \omega^2 \sin \varphi,$$

где $m_r = 440$ кГ, $r = 0,04$ м, $\omega = 1500$ мин⁻¹, $l = 0,145$ м – соответственно приведенная масса, радиус кривошипа, частота вращения, , длина шатуна, φ – угол поворота коленвала.

Модель колебаний автомобиля с ДГУ описана следующими математическими уравнениями [93], реализованными на основе блок-схемы (приложение Г):

$$m_{a1} \ddot{Z}_0 - (F_{\Pi 11} + F_{\Pi 12}) + (F_{\Pi дгу1} + F_{\Pi дгу2}) = 0;$$

$$m_{дгу} \ddot{Z}_{дгу} - (F_{\Pi дгу1} + F_{\Pi дгу2}) = 0;$$

$$m_{a1} \rho_{x1}^2 \ddot{\beta}_1 - (F_{\Pi 1} L_1 - F_{\Pi 2} L_2) + (F_{\Pi дгу1} L_{дгу1} + F_{\Pi дгу2} L_{дгу1}) = 0;$$

$$m_{дгу1} \rho_{x1}^2 \ddot{\beta}_2 + (F_{\Pi дгу1} L_4 - F_{\Pi дгу2} L_5) + (F_{\Pi дгу1} L_4 + F_{\Pi дгу2} L_5) = 0;$$

$$m_{н1} \ddot{Z}_{н1} + (F_{ш1} - F_{рп1}/2) = 0;$$

$$m_{н2} \ddot{Z}_{н2} + (F_{ш2} - F_{рп2}/2) = 0;$$

$$m_{н3} \ddot{Z}_{н3} + (F_{ш1} - F_{рп3}/2) = 0;$$

$$\beta_1 = \frac{Z_1 - Z_2}{2L_1 + 2L_2}; \quad \beta_2 = \frac{Z_{дгу1} - Z_{дгу2}}{2L_5 + 2L_4};$$

$$\dot{\beta}_1 = \frac{\dot{Z}_1 - \dot{Z}_2}{2b_{ш1} + 2b_{ш1}}; \quad \dot{\beta}_2 = \frac{\dot{Z}_{дгу1} - \dot{Z}_{дгу2}}{2L_5 + 2bL_4};$$

$$F_{шл1} = F_{сшл1} (q_{1л}) + F_{кшл1} (\dot{q}_{1л});$$

$$F_{шл2} = F_{сшл2} (q_{2л}) + F_{кшл2} (\dot{q}_{2л});$$

$$F_{шл3} = F_{сшл3} (q_{3п}) + F_{кшл3} (\dot{q}_{3л});$$

$$F_{п1} = F_{рп1} (L_1 \beta_1 - L_2 \beta_1) - F_{амл1} (L_1 \dot{\beta}_1 - L_2 \dot{\beta}_1) + F_{тр1};$$

$$F_{п2} = F_{рп2} (L_2 \beta_2 - L_{р2} \frac{(\beta_{к2л} - \beta_{к3л})}{2}) + F_{амл2} (L_2 \dot{\beta}_2 + L_2 \frac{\dot{\beta}_{к2л} - \dot{\beta}_{к3л}}{2}) + F_{тр2},$$

где $F_{сш1}, F_{сш2}, F_{сш3}$ – упругая характеристика шин переднего, среднего заднего моста в виде зависимости от относительной их деформации; $F_{р1}, F_{р2}$, – упругая характеристика рессор переднего моста и балансирной подвески в виде зависимости от их относительной деформации;

$F_{тр1}, F_{тр2}$ – характеристика трения рессор переднего моста и балансирной подвески в виде зависимости интеграла от относительной скорости перемещения с ограничениями ее величины;

$F_{ам1}$ – характеристика амортизаторов переднего моста в виде зависимости от относительной скорости перемещения;

$F_{шл1}, F_{шл2}$ – суммарная характеристика рессоры, амортизатора и трения переднего моста и балансирной подвески;

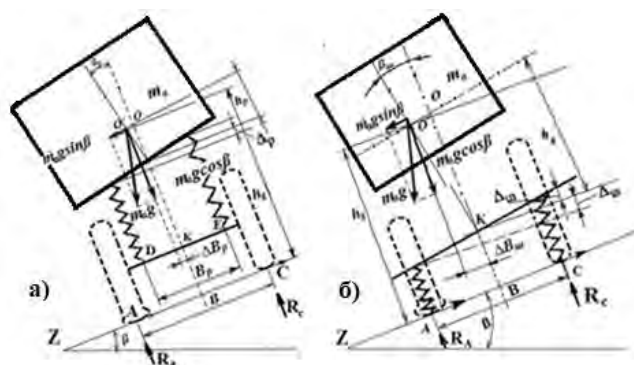
m_a – масса подрессоренной части штабной машины;
 $m_{н1}, m_{н2}, m_{н3}$ –неподрессоренная масса переднего, среднего, заднего моста штабной машины;

L_1, L_2 – расстояние между центром тяжести и передней и задней подвесками;
 J_y – момент инерции подрессоренной массы штабной машины в продольной плоскости;
 m – подрессоренная масса штабной машины в продольной плоскости;
 $J_{x1} = m_{п1} \rho_{x1}^2$ – момент инерции подрессоренной массы в поперечной плоскости;
 $J_{x2} = m_{п1} \rho_{x2}^2$ – момент инерции подрессоренной массы дизель-генераторной установки в поперечной плоскости.

Данная компьютерная модель реализована в пакете MatLab/Simulink.

3.8 Устойчивость колесных машин

3.8.1 Ниже рассмотрено определение статической устойчивости аналитическим путем. Машина упрощенно представлена в виде массы m_a , соответствующей задней подрессоренной массе колесной машины (рисунок 3.29).



а – при деформации рессор, б – при деформации шин

Рисунок 3.29. – Схема определения статической поперечной устойчивости колесной машины

В ней учтены деформации рессор и шин, а также различные центры крена на рессорах и шинах при наклоне колесной машины, изменения конструктивных параметров. В этом состоит новизна модели, по которой получено аналитическое решение [14].

При проведении расчета принято допущение, что поперечная составляющая силы тяжести при наклоне подрессоренной массы за счет деформации упругих элементов не меняет своего направления при смещении центра масс в сторону наклона колесной машины.

Опрокидывающий момент, создаваемый поперечной составляющей силы тяжести, равен

$$M_{оп} = m_a g h_g \sin \beta . \quad (3.1)$$

Опрокидывающий момент при блокировке подвески имеет величину, равную $M_{\text{оп}} = m_a g \cos \beta$, а восстанавливающий момент уменьшается за счет изменения его плеча на величину, обусловленную смещением центра масс при наклоне подрессоренной массы машины относительно центра крена. Для упругих элементов подвески восстанавливающий момент:

для рессор,

$$M_B = m_a g \cos \beta \left(\frac{B}{2} - \Delta B_p \right); \quad (3.2)$$

для шин,

$$M_B = m_a g \cos \beta \left(\frac{B}{2} - \Delta B_{\text{ш}} \right), \quad (3.3)$$

где m_a – подрессоренная масса;

ΔB_p и $\Delta B_{\text{ш}}$ – величины смещения центра масс машины и вертикальной составляющей силы тяжести при наклоне подрессоренной массы машины относительно оси крена при деформации соответственно рессор и шин.

Крен подрессоренной массы колесной машины рассматривался отдельно для рессор и шин, так как наклон подрессоренной массы машины на упругих элементах подвески и шинах происходит относительно их различных центров крена.

Для рессор центр крена подрессоренной массы находится на плоскости, проходящей через оси верхних кронштейнов рессор, а для шин ось крена подрессоренной массы находится на уровне осей колес.

Величину предельного угла статической поперечной устойчивости при учете деформации упругих элементов подвески и шин можно представить как разность между максимальным углом поперечной устойчивости при отсутствии рессор и шин и углами крена колесной машины при наличии деформации рессор и шин.

При деформации рессор подвески или шин подрессоренная масса поворачивается на угол β_y относительно их оси крена. При неизменном опрокидывающем моменте восстанавливающий момент

$$M_B = m_a g \cos \beta \left(B/2 - \Delta B_y \right); \quad (3.4)$$

$$\Delta B_y = h_k \operatorname{tg} \beta_y, \quad (3.5)$$

где ΔB_y – величина смещения центра масс; h_k – плечо поворота подрессоренной массы относительно оси крена; B – расстояние между упругими элементами (рессорная база в случае подвески или колея в случае

упругих шин).

Величина $\operatorname{tg}\beta_y$ определяется из геометрического подобия треугольников с углами при вершине β_y :

$$\operatorname{tg}\beta_y = \Delta_{\text{уп}} / (0,5B_y), \quad (3.6)$$

где $\Delta_{\text{уп}}$ – деформация упругих элементов.

Деформация упругого элемента $\Delta_{\text{упр}}$ определяется из равенства двух моментов:

а) момента относительно оси крена от поперечной составляющей силы тяжести

$$M_1 = m_a g h_k \sin \beta; \quad (3.7)$$

б) момента, возникающего при дополнительной деформации упругих элементов, вызванной поперечным наклоном поддрессоренной массы

$$M_2 = \Delta_{\text{упр}} C_{\text{упр}} B_y,$$

где $C_{\text{упр}}$ – суммарная вертикальная (радиальная) жесткость упругих элементов одной стороны машины.

Из равенства $M_1 = M_2$ следует

$$\Delta_{\text{упр}} = m_{\text{п}} g h_k \sin \beta / (C_{\text{упр}} B_{\text{уп}}).$$

Решая совместно уравнения (3.1–3.7) при $M_{\text{оп}} = M_{\text{в}}$, получим формулу для определения угла статической устойчивости для рассматриваемой массы машины с учетом деформации упругих элементов

$$\operatorname{tg}\beta_{\text{п.мах}} = B/2h_g - 2m_a g h_k^2 \sin \beta / (h_g C_{\text{уп}}).$$

В итоге при деформации упругих элементов подвески формула для определения поперечной устойчивости при деформации упругих элементов подвески-имеет вид

$$\operatorname{tg}\beta_{\text{р.мах}} = B/2h_g - 2m_a g h_p^2 \sin \beta / (h_g C_p B_p^2),$$

а при упругой деформации шин

$$\operatorname{tg}\beta_{\text{ш.мах}} = B/2h_g - 2 g h_{\text{ш}}^2 \sin \beta / (h_g C_{\text{ш}} B^2).$$

где h_p – радиус крена подрессоренной массы при деформации рессор; C_p – радиальная жесткость упругих элементов подвески; B_p – расстояние между площадками рессор; $h_{ш}$ – радиус крена подрессоренной массы при деформации шин; $C_{ш}$ – радиальная жесткость шин; B – колея. $\beta_{p.max}$, $\beta_{ш.max}$ – максимальный угол деформации на рессорах, шине.

Таким образом, наибольший угол статической устойчивости автомобиля с учетом деформации упругих элементов подвески и шин равен

$$\operatorname{tg} \beta_{п.max} = \frac{B}{2h_g} - 2m_{п}g \left(\frac{h_p^2 \sin \beta}{h_g C_p B_p^2} + \frac{h_{ш}^2 \sin \beta}{h_g C_{ш} B^2} \right); \quad (3.8)$$

$$\eta_0 = \operatorname{tg} \beta \Rightarrow \alpha_{c.y} = \operatorname{arctg} \eta_0;$$

$$\alpha_{c.y} = \operatorname{arctg} \left(\frac{B}{2h_g} - 2m_{п}g \left(\frac{h_p^2 \sin \beta}{h_g C_p B_p^2} + \frac{h_{ш}^2 \sin \beta}{h_g C_{ш} B^2} \right) \right). \quad (3.9).$$

Следует также учитывать возможность уменьшения угла статической устойчивости на величину $\alpha_{c.y} = 57,3 \frac{q_1 - q_2}{B}$ из-за попадания колеса автомобиля в выбоину дороги. Так, при разности высот неровностей 100 мм угол статической устойчивости может уменьшиться на $2,3^\circ$.

3.8.2 Устойчивость колесной машины при повороте

Рассмотрим упрощенную схему для определения предельной скорости установившегося криволинейного движения колесной машины по условию опрокидывания (рисунок 3.30).

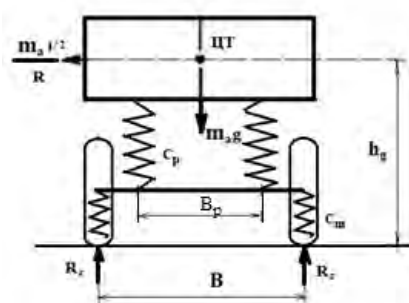


Рисунок 3.30. – Схема для определения скорости опрокидывания колесной машины при маневре «Поворот»

Момент опрокидывания создается поперечной составляющей центробежной силы скорости v ;

$$M_{\text{оп}} = m_a \frac{v^2 h_g}{R}; \quad (3.10)$$

восстанавливающий момент – силой тяжести

$$M_{\text{оп}} = m_a g B/3. \quad (3.11)$$

Приравнивая правые части уравнений (3.10) и (3.11), после преобразования получим

$$v = \sqrt{\frac{g BR}{2 h_g}}.$$

Учет влияния деформации подвески и шин на положение центра масс при действии поперечной силы выражается в уменьшении плеча восстанавливающего момента, поэтому критическая скорость с учетом влияния жесткостей упругих элементов подвески и шин

$$v_{\text{кр}} = \sqrt{\frac{gBR}{2(h_g + \frac{h_{\text{кр}}^2 m_a}{2c_p B_p^2} + \frac{h_{\text{ш}}^2 m_a}{2c_{\text{ш}} B^2})}}, \quad (3.12)$$

где $h_{\text{кр}}$, $h_{\text{ш}}$ – плечо крена подрессоренной массы при деформации подвески и шин соответственно; m_a – подрессоренная масса колесной машины;

c_p , B_p – жесткость подвески и расстояние между площадками рессор соответственно; $c_{\text{ш}}$, B – жесткость шин и колея соответственно; R – радиус поворота; $g = 9,81 \text{ м/с}^2$.

3.9 Управляемость колесной машины

Управляемость автомобиля является важным показателем колесной машины. Для упрощения описания системы уравнений движение автомобиля будем рассматривать его как плоско-параллельное движение твердого тела на плоскости (рисунок 3.31).

При моделировании управляемости принято использование модели шины в виде точечного контакта, как сочетание параллельного включения упругого и демпфирующего элемента в вертикальной и поперечной плоскости (рис. 3.32).

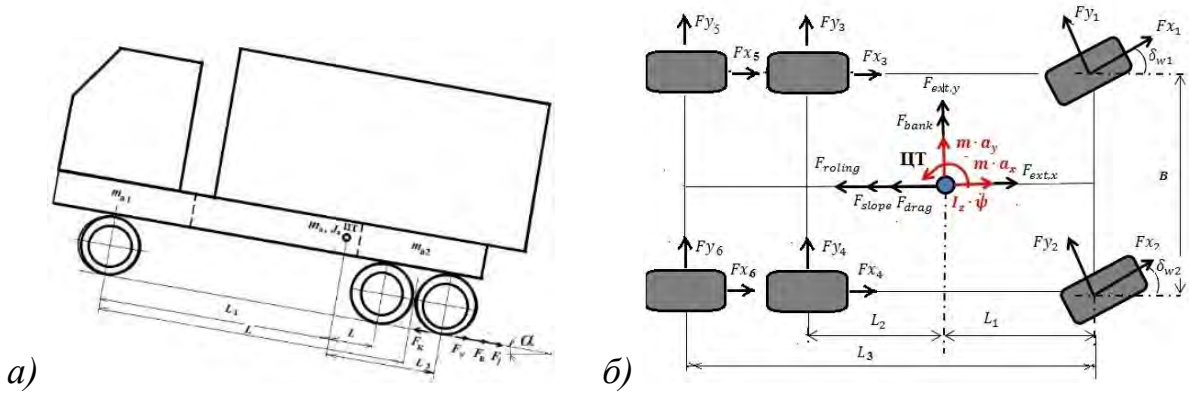


Рисунок 3.31. – Схема сил, действующих на автомобиль при движении (а) и управлении ТС (б)

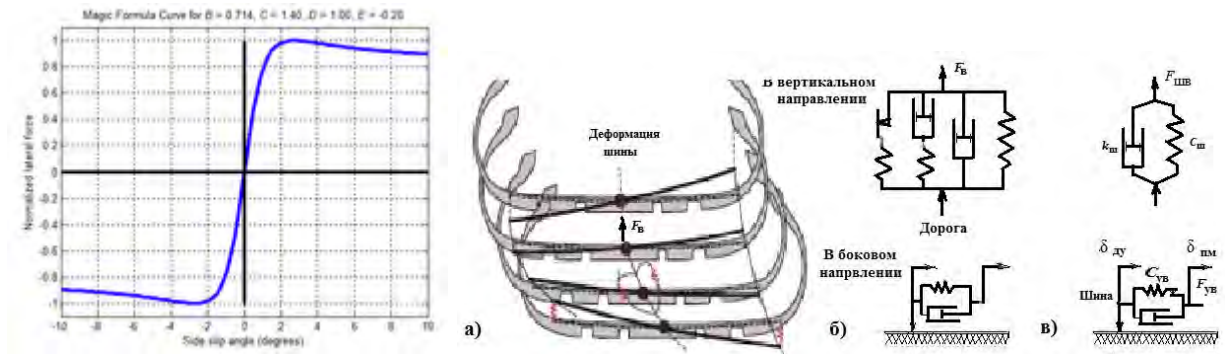


Рисунок. 3.32. – Модель шины

Как показал анализ графиков зависимостей в работах [51], [54], [94] увод следует рассматривать как зависимость от скорости угла, а не самого угла как делают большинство исследователей т.е. в виде интеграла от скорости. И представить боковой увод как параллельное включения нескольких упругих элементов с ограничением их максимальных величин по аналогии с моделью трения [14, 48]. В этом новизна модели, которую можно описать следующими уравнениями,

$$F_{увш} = \begin{cases} C_{уш.дин1} \int_0^t (\dot{\delta}_{прк} - \dot{\delta}_{па}) dt & \text{если } C_{уш.дин1} \int_0^t (\dot{\delta}_{прк} - \dot{\delta}_{па}) dt > F_{огр.усил} \\ F_{огр.усил} & \\ -F_{огр.усил} & \text{если } C_{уш.дин1} \int_0^t (\dot{\delta}_{прк} - \dot{\delta}_{па}) dt < -F_{огр.усил}; \end{cases}$$

На основании второго закона Ньютона движение и управляемость ТС может быть описана следующими тремя дифференциальными уравнениями:

в направлении продольной оси x

$$\begin{aligned}
m \cdot a_x &= F_{x1} \cdot \cos(\delta_{w1}) - F_{y1} \cdot \sin(\delta_{w1}) + F_{x2} \cdot \cos(\delta_{w2}) - F_{y2} \cdot \sin(\delta_{w2}) \\
&+ F_{x3} + F_{x4} + F_{x5} + F_{x6} - F_B - F_9; \\
a_x &= \dot{v}_x + v_y \cdot \dot{\psi}; \\
m \cdot a_x &= 2 \cdot F_{x1} \cdot \cos(\delta_{w1}) - 2 \cdot F_{y1} \cdot \sin(\delta_{w1}) + 4 \cdot F_{x3} - F_B - F_9;
\end{aligned}$$

где $F_{x1} \dots F_{x6}$ – усилие на колесах;

F_B – сила сопротивления воздуха при движении автомобиля

$$F_B = K_B \cdot A_a \cdot v_a^2;$$

K_B — коэффициент обтекаемости автомобиля, $\text{Н} \cdot \text{с}^2 / \text{м}^4$; A_a — лобовая площадь автомобиля, м^2 ; v_a — текущая скорость движения автопоезда, м/с ; v_x — производная продольной скорости, v_y — боковая скорость; $\dot{\psi}$ — угловая скорость вращения автомобиля вокруг ЦТ;

F_9 — сила дорожного сопротивления

$$F_9 = 9,81 \cdot g \cdot m_a$$

m_a — полная масса автомобиля, кг ; g — коэффициент дорожного сопротивления,

$$g = f_k \cdot \cos \alpha + \sin \alpha$$

α — угол наклона текущего элементарного участка маршрута, f_k — коэффициент сопротивления качению шин; F_j — инерционная сила сопротивления разгону,

$$F_j = m_a \cdot a \cdot \gamma$$

a — ускорение автомобиля м/с^2 , λ — коэффициент учета вращающихся масс,

$$\lambda = \frac{J_{\text{eng}} \cdot \eta_T \cdot (u_{\text{кп}} \cdot u_{\text{вм}})^2}{m_a \cdot r_k^2} + \frac{\sum J_k}{m_k \cdot r_k^2};$$

J_{eng} — момент инерции вращающихся масс двигателя и сцепления, $\text{кг} \cdot \text{м}^2$,

J_k — суммарный момент инерции колес автопоезда и приводных валов, $\text{кг} \cdot \text{м}^2$.

в направлении поперечной оси y

$$\begin{aligned}
m \cdot a_y &= (F_{x1} \cdot \sin(\delta_{w1}) + F_{x2} \cdot \sin(\delta_{w2}) + F_{y1} \cdot \cos(\delta_{w1}) + F_{y1} \cdot \cos(\delta_{w2}) \\
&+ F_{x2} \cdot \sin(\delta_{w2}) + F_{x2} \cdot \sin(\delta_{w2}) + F_{y2} \cdot \cos(\delta_{w2}) + F_{y2} \cdot \cos(\delta_{w2}) + F_{gr}; \\
F_{gr} &= -m \cdot g \cdot \sin(\theta) \\
m \cdot a_y &= (2 \cdot F_{x1} \cdot \sin(\delta_{w1}) + 2 \cdot F_{y1} \cdot \cos(\delta_{w1}) + F_{gr};
\end{aligned}$$

$$a_y = \dot{v}_y + v_x \cdot \dot{\psi};$$

θ – угол поперечного наклона дороги;

$\dot{v}_y$ – производная продольной скорости,

v_x – боковая скорость; $F_{xi} \dots F_{xi}$ – усилия на колесах.

Вращение автомобиля относительно оси Z вокруг ЦТ

$$J_a \cdot \ddot{\psi} = [-F_{x1} \cdot \cos(\delta_{w1}) - F_{y1} \cdot \sin(\delta_{w1}) - F_{x2} \cdot \cos(\delta_{w2}) - F_{y2} \sin(\delta_{w2}) \sin(\delta_{w2}) - F_{x3} + F_{x4} - F_{x5} + F_{x6}] \cdot \frac{B}{2}.$$

При равенстве сил $F_{x1} = F_{x2}, F_{y1} = F_{y2}, F_{x3} = F_{x4} = F_{x5} = F_{x6}$

$$J_a \cdot \ddot{\psi} = [-2 \cdot F_{x1} \cdot \cos(\delta_{w1}) - 2 \cdot F_{y1} \cdot \sin(\delta_{w1})] \cdot \frac{B}{2}.$$

Значение для $\sin$ углов δ_{w1}, δ_{w2} целесообразно заменить на δ_{wi} (в радианах), чтобы учесть изменение знака величины. И принять углы $\delta_{w3} \dots \delta_{w6} = 0$. Значение $\cos(\delta_{w1..2})$ меняются в пределах $1 \dots 0,94$ при $0 \dots 10^\circ$. Их также можно заменить на 1, включая $\cos(\delta_{w3..6}) = 1$.

Как видно из уравнений для их решения требуется определить усилия на колесах с учетом углов поворота передних колес. Они могут быть определены исходя из тяговой динамики автомобиля, описываемой известным дифференциальным уравнением второго порядка [18, 19] (см. п.2.12).

$$\frac{dV_a}{dt} = \frac{dv_x}{dt} = \frac{\sum_0^6 F_k - F_B - F_g}{m_a \cdot \delta}.$$

где F_k – сила тяги с учетом угла поворота управляемых колес и распределения массы на них; $F_{xi} = K_i \cdot F_k \cdot \delta$; для 1-го, 2-го колеса, $F_{yi} = K_i \cdot F_k$ – для 3-го... 6-го колеса

$$F_{ki} = \frac{M \cdot u_{кп} \cdot u_{вм} \cdot \eta_T}{r_k};$$

K_1 – коэффициент распределения массы на переднюю ось; $K_{2,3}$ – коэффициенты распределения массы на ведомые оси; M – текущий крутящих моментов двигателя внутреннего сгорания (ДВС), Н·м, η_T – КПД трансмиссии на соответствующей передаче КП; r_k – радиус качения ведущих колес автомобиля, м; $u_{кп}$ – текущее значение передаточного числа КПП; $u_{вм}$ – передаточное число главной передачи ведущего моста; v_a – текущая скорость движения автомобиля, м/с,

$$v_a = \frac{\pi}{30} \cdot \frac{n \cdot r_k}{u_{kp} \cdot u_{bm}};$$

n – текущая частота вращения коленчатого вала ДВС, мин⁻¹.

Предложенные модель и подход является новыми в реализации моделирования управляемости автомобиля с учетом его движения. Данная модель использовалась при моделировании в Matlab/Simulink. Для MS.ADAMS использовалась модель, составленная из его типовых элементов аналогично рисунка 3.33.

3.10 Модель в пакете MSC.ADAMS

Динамическая модель колесной машины с установленным кузовом-контейнером на шасси МАЗ-631705-374 в пакете MSC.ADAMS создавалась на основе его типовых библиотек с учетом схемы рисунка 3.25. Общий вид модели колесной машины с кузовом-контейнером показан на рисунке 3.33.

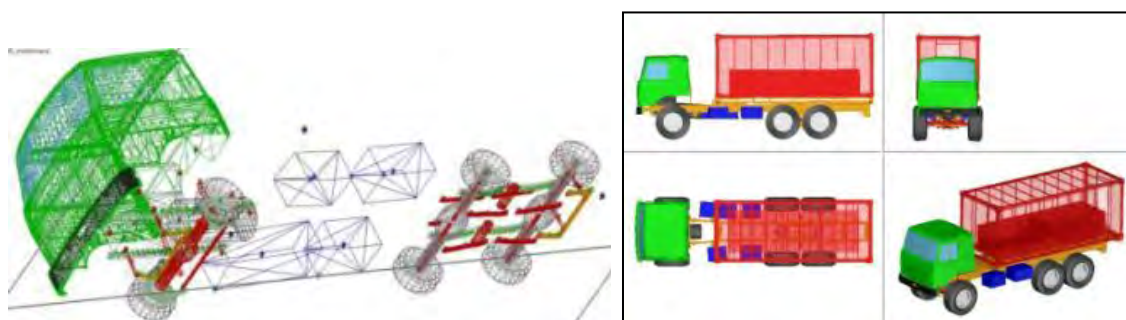


Рисунок 3.33. – Общий вид модели колесной машины с кузовом-контейнером

Предложенная пространственная модель отличается от известных (в виде единого тела) (рисунок 3.25) наличием двух приведенных поддресоренных масс $m_{п1}$, $m_{п2}$ в продольной плоскости, которые могут совершать как вертикальные, так и продольно-угловые и поперечно-угловые колебания. Также учтена кинематика задней балансирной подвески.

С помощью этой модели в пакете MSC.ADAMS путем моделирования определены нагрузки, действующие на контейнер в наиболее тяжелых дорожных условиях (режим переезда препятствий) [26].

3.11 Модель движения колесной машины по реальной дороге

3.11.1 На движущуюся по дороге колесную машину действуют пять сил: сила тяги F_k , сила сопротивления воздуха F_v , сила дорожного сопротивления,

F_ψ , сила инерционного сопротивления разгону F_j , сила торможения F_T . (рисунок 3.34).

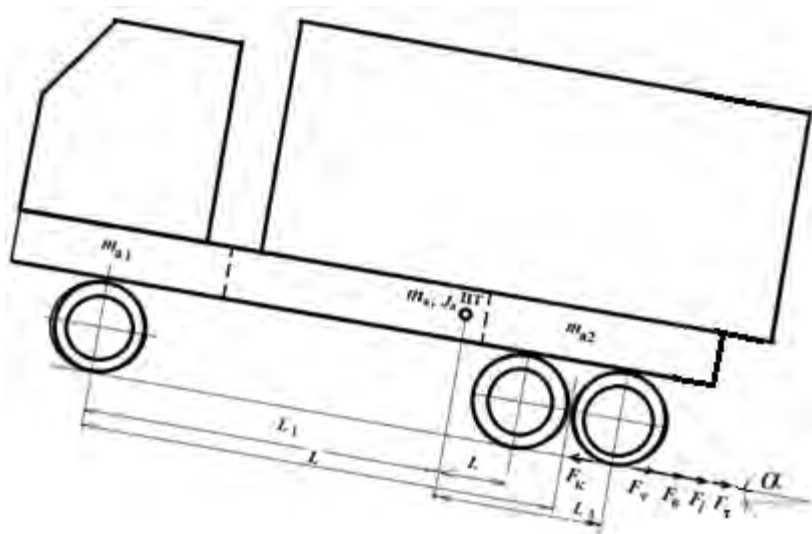


Рисунок 3.34. – Динамическая модель движения колесной машины

На основании принципа Даламбера, рассматривая машину как одну материальную точку, динамику движения можно описать следующим уравнением [32, 33, 57]

$$\frac{dV_a}{dt} = \frac{F_k - F_B - F_\psi - F_T}{m_a \delta},$$

где F_k – сила тяги:

$$F_k = \frac{M u_{\text{кп}} u_{\text{вм}} \eta_T}{r_k};$$

M – текущий крутящий момент двигателя внутреннего сгорания (ДВС), Н·м,

η_T – КПД трансмиссии на соответствующей передаче КП;

r_k – радиус качения ведущих колес автомобиля, м;

$u_{\text{кп}}$ – текущее значение передаточного числа КП;

$u_{\text{вм}}$ – передаточное число главной передачи ведущего моста;

F_B – сила сопротивления воздуха

$$F_B = K_B A_a v_a^2,$$

где K_B — коэффициент обтекаемости колесной машины, Н·с²/м⁴;

A_a — лобовая площадь автомобиля, м²,

v_a — текущая скорость движения колесной машины, м/с,

$$v_a = \frac{\pi}{30} \cdot \frac{n r_k}{u_{\text{кп}} \cdot u_{\text{вм}}};$$

где n – текущая частота вращения коленчатого вала ДВС, мин⁻¹.

F_{ψ} – сила дорожного сопротивления

$$F_{\psi} = 9,81 \psi m_a$$

где m_a – полная масса автомобиля, кг;

ψ – коэффициент дорожного сопротивления,

$$\psi = f_{\text{ш}} \cos \alpha + \sin \alpha$$

где α – угол наклона текущего элементарного участка маршрута, f – коэффициент сопротивления качению автомобиля как одной материальной точки, $f_{\text{ш}}$ – коэффициент сопротивления качению шин.

При малых значениях углов наклона (до 10–15°) $\sin \alpha \equiv \tan \alpha$ и $\sin \alpha$ можно заменить на отношение изменения высоты к пути $\Delta h / \Delta l$, $\cos \alpha$ на 1.

$$\text{Отсюда } \psi = f_{\text{ш}} + \alpha.$$

F_j – сила сопротивления разгону.

$$F_j = m_a a \delta$$

где a – ускорение автопоезда м/с²,

δ – коэффициент учета вращающихся масс:

$$\delta = \frac{J_{\text{eng}} \eta_{\text{т}} (u_{\text{кп}} u_{\text{вм}})^2}{m_a r_k^2} + \frac{\sum J_k}{m_k r_k^2};$$

где J_{eng} – момент инерции вращающихся масс двигателя и сцепления, кг·м², J_k – суммарный момент инерции колес колесной машины и приводных валов, кг·м².

$F_{\text{т}} = K v_a$ – сила торможения.

Новым в уравнениях является учет и представление силы торможения, как зависимости от скорости v_a на основе анализа полученных графиков режимов движения. Значение $K=4000$ Н с/м определено из распечаток (таблица 3.3). Среднее значение пути торможения составило $l = 100$ м.

Таблица 3.3 – Изменения скорости перед препятствиями, поворотами

Параметр	Значение К						
l , м	50	100	135	105	80	145	85
v_a , м/с	10,3	11,1	10,0	10,4	11,5	10,3	10,3

Сложность решения этих уравнений заключается в необходимости получения и задания массива продольного профиля дороги, дорожных условий, реализации многопараметрических зависимостей двигателя и логики действий водителя по переключению передач [32, 33]. Для этих целей обычно используются стандартные средства и процедуры MatLab, значительно усложняющие реализацию в Simulink и замедляющие процесс моделирования.

Кроме того, как указывалось выше при моделировании движения необходимо учитывать дорожные условия:

- ограничение скорости на отдельных участках, населенных пунктах;
- повороты дороги и подъемы, при которых приходится снижать скорость;
- движение встречных и попутных автомобилей;
- состояние дороги (ее микропрофиль).

Использовался модуль S-Function Builder, через программу которого на C/C++ реализовалось совместное моделирование движения и колебаний автомобиля, логика переключения передач, что является новым и наиболее эффективным для решения такого класса задач. Такой подход позволяет формировать возмущение дороги, дорожные ограничения, решать задачи в реальном масштабе времени за счет отказа от применения Excel и интерпретируемых процедур Matlab, реализовать логику управления, переключения КПП, вести моделирование автомобиля с переменной скоростью на основе дорожных условий, вибрационного воздействия на водителя и тягово-скоростных свойств автомобиля [99].

Моделирование движения автомобиля реализовано с помощью блок-схемы на рисунке 3.35, включающей двигатель. Управление процессом осуществляется за счет получаемой разности заданного сигнала по пути и сигнала по скорости, который подается на блок PID-регулятора $\int Pl(s)$ и множительный блок Product1. В зависимости от числа оборотов двигателя, рассчитываемых исходя из текущей скорости автомобиля, передаточных чисел КПП, главной передачи, размера колес, определяется крутящий момент и усилия на колесах. В модуле P_expl учитываются потери мощности на вентилятор, генератор, глушитель. Расход топлива рассчитывался с помощью характеристик двигателя и его загрузки.

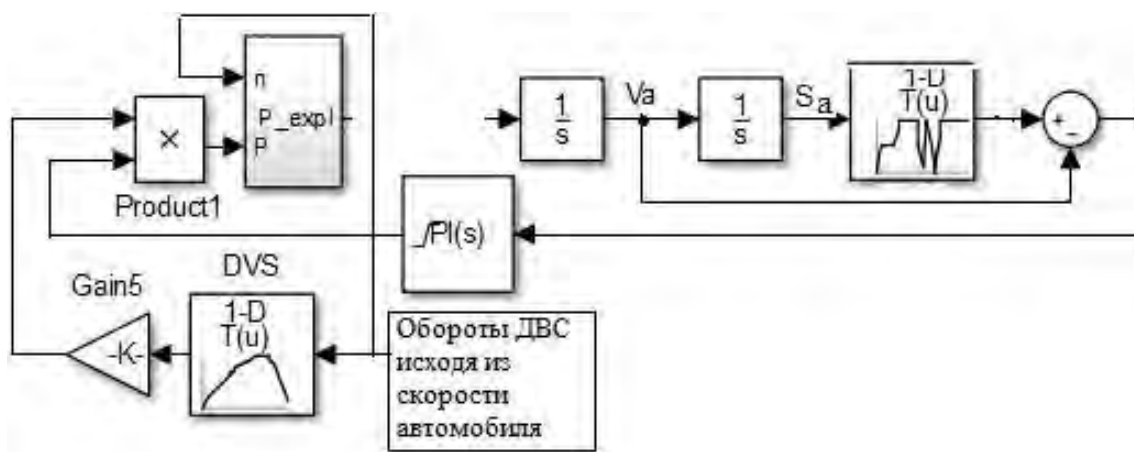


Рисунок 3.35. – Блок-схема моделирования режима движения автомобиля

Данное простое решение позволяет задать желаемый режим движения автомобиля с учетом пути и дорожных ограничений.

Новым в логике переключения передач является использование при моделировании движения ускорений при переходе на высшую и низшую передачу. Фрагмент основной части программы приведен в приложении Б.

Первоначально предполагалось использование рекомендаций по оптимальному переключению передач (рисунок 3.36), предложенному в работах [32, 33] исходя из минимизации расхода топлива на основе логики переключения передач КПП с использованием инверсированных зависимостей загрузки двигателя ($\sim 71\text{--}76\%$ от максимальных оборотов двигателя).

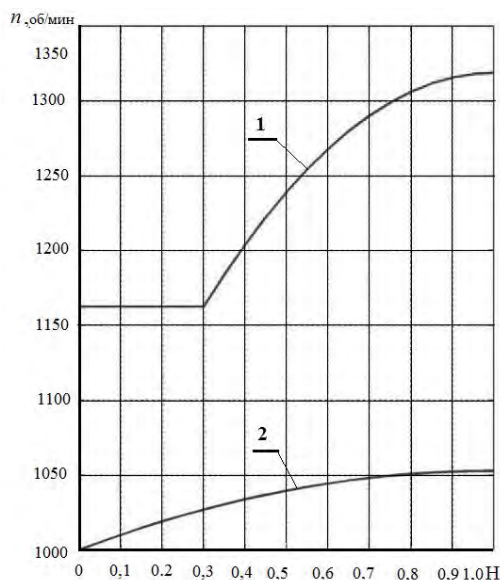


Рисунок 3.36. — Границы работы двигателя (а), задающие область переключения КПП (1– вверх, 2 – вниз) и плоская колебательная модель автомобиля бхб (б)

Исследования данного подхода показали, что при одновременном моделировании движения и колебаний с малым шагом интегрирования происходит остановка процесса моделирования. И от этого варианта пришлось отказаться.

Предложена **новая логика** переключения передач КП, основанная на использовании ускорений и оборотов двигателя (перехода на высшую передачу при 80% от максимальных оборотов двигателя).

Выбор такого решения базируется на промелькнувшей информации, что Toyota использует ускорения для более плавного переключения передач на АКП.

Переключение КПП вверх (фрагмент кода) осуществляется когда

```
if (a2>0.5&&ndvs>1700&&chm!=1)
```

```
а вниз когда
```

```
//Переключение КПП вниз
```

```
if (a2<-0.2&&ndvs<1100&&chm!=1)
```

где a_2 – продольное ускорение автомобиля в момент переключения;

$ndvs$ – обороты двигателя, chm – триггер включения нейтральной передачи.

Особенностью переключения является то, что процесс происходит через нейтральную передачу, при которой $F_k = 0$. Но сам процесс моделирования не останавливается: подсчитывается определенное число циклов моделирования для его отключения. Это позволяет не тормозить процесс моделирования и обеспечивает режим реального времени. Это является **новым** по сравнению с [57].

Управляющее скоростью воздействие осуществлялось с помощью PID-регулятора $\int Pl(s)$ и множительного блока Product1.

Учитывались потери мощности на вентилятор, генератор, глушитель рассчитываемые в модуле P_expl. Расход топлива рассчитывался с помощью многопараметрических характеристик двигателя и его загрузки (рисунок 3.37).

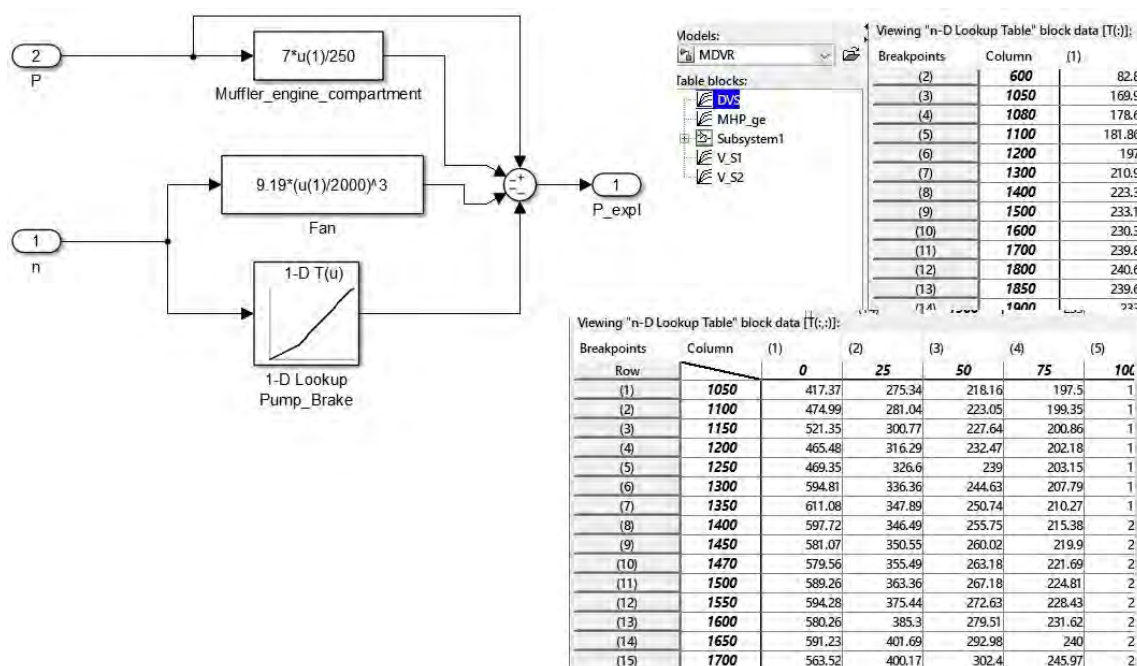


Рисунок 3.37. – Учет потерь мощности и характеристики двигателя

Используемая при расчетах топливной экономичности затрачиваемая мощность двигателя рассчитывалась исходя из усилий, необходимых для преодоления подъема F_ψ , ускорений в процессе его преодоления F_j , сопротивления качения F_ψ , сопротивления воздуха

$$F_{\text{затрач}} = F_j + F_\psi + F_B$$

где F_j – сила сопротивления разгону;

F_ψ – сила дорожного сопротивления;

F_B – сила сопротивления воздуха;

F_k – сила тяги:

$$N_{\text{затр}} = \frac{M_{\text{зат}} n_{\text{двиг}}}{9550} = \frac{F_{\text{зат}} R_{\text{кол}} v_{\text{авто}}}{9550} =$$

Ниже приведен фрагмент программы на языке C для реализации этого.

```
Fk=(89010.53*Ndv*u_kp[n_kp])/ndvs; //9550*N*u_rk*u_kp*Um*kpd;//
9550*N*1.107*6.59*u_kp[]/ndvs*rk // Fk= (M*u_kp*u_gp*kpd)/rk; Сила тяги
nдвиг= 30 *v_a (i_рк*i_кп*i_мост) / (3.14 * r_кол );
Nt=0.0010*(Fa2+Fy_2+Fy1+Fv)*v_a+Neq;
```

Необходимый для расчетов движения угол наклона дороги получается путем деления изменения высот к шагу пути

память заносится информация о дорожных условиях. На данном этапе режим скоростей движения задан в виде массива V_S , в который введены ограничения скорости (связанные с населенными пунктами, поворотами, препятствиями). Дополнительно в модель поступает информация о пробое подвески (x_3), что вызывает снижение скорости на 10 % на последующей длине участка (100 м).

В качестве колебательной модели (модуль *VibrModelAuto*) использовалась плоская модель автомобиля МАЗ-6317 (6×6) с кузовом-контейнером, учитывающая сложную модель трения в рессорах, описанная в п. 3.5 и в работе [74]. Эта модель колебаний показала хорошую сходимость по ускорениям с расхождением 5–15 % в октавных полосах частот (2, 4, 8, 16 Гц) при сравнении с экспериментальными данными (1980 г.) автомобиля МАЗ-6303 при движении по булыжному шоссе Плещеницы–Вилейка. Для информации, при использовании "вязкой" модели трения в подвеске расхождение достигает 30–60 %, а при линейной модели на основе передаточной функции нескольких раз. Разработанную модель целесообразно использовать при расчетах динамической нагруженности мостов, элементов подвески, вибрационных испытаний кабин. В то же время для оценки нагруженности рам, платформ, кузовов-контейнеров, испытывающих крутящие нагрузки, требуется применение пространственной модели, рассмотренной в п. 3.5.7.1 и работах [26,74].

В качестве объекта исследований использовались данные по автомобилю МАЗ-6317 с кузовом-контейнером, имеющие следующие параметры: масса автомобиля и контейнера 17000 кг, площадь лобового сопротивления $A_b = 9.7 \text{ м}^2$; коэффициент сопротивления качению $f = 0.003903$; коэффициент сопротивления воздуха $K_b = 0.41202$; двигатель ЯМЗ 7512 (243 кВт); мощность для вентилятора в номинальном режиме 9,19 кВт; коробка передач ЯМЗ-202; номера передач $N_{\text{кп}} = [0, 1, 2, 3, 4, 5, 6, 7, 8]$; передаточные числа 0,0; 8,73; 6,13; 4,51; 3,50; 2,78; 1,75; 1,00; 0,795; передаточное число ведущего моста $u_m = 3.45$; радиус качения ведущего колеса, $r_k = 0.522 \text{ м}$; время переключения передачи $t = 2.0 \text{ с}$; момент вращающихся масс двигателя и сцепления $J_m = 4.51 \text{ кг} \cdot \text{м}^2$; суммарный момент инерции колес и приводных валов $J_k = 251.136 \text{ кг} \cdot \text{м}^2$; суммарная максимальная тормозная сила $F_{\text{т}} = 13000 \text{ Н}$; плотность топлива $\rho_o = 830 \text{ кг/м}^3$.

С помощью предложенной модели проведено моделирование движения колесной машины (автомобиль 6×6) по реальной местной дороге под Минском: Белробот – Приморье – Семков городок – Чучаны – МКАД2 общей протяженностью 12 км (4 км асфальтобетонного разбитого шоссе, 8 км асфальтированного шоссе хорошего качества).

Предложенный подход совместного моделирования движения и колебаний по дороге с реальным макро и микропрофилем является новым, позволяющим рассчитывать реальные скорости движения, экономичность на реальной дороге с учетом вибрационного воздействия на водителя.

Учитывая возможность вести моделирование в реальном масштабе времени при частоте многоядерного процессора компьютера более 3,5 Гц на данной модели можно проводить имитационные стендовые испытания систем управления АКПП, их отладку, и всего автомобиля, а также реакций водителя, как это делают ведущие зарубежные фирмы.

3.11.2 Учет переменного сопротивления качения

При моделировании движения автомобиля необходимо учитывать изменение значения коэффициента качения, который в пределах даже одной дороги может меняться. Значения коэффициента качения зависят от нагрузки, размеров шины, давления в ней, числа слоев корда в шинах, скорости движения и других факторов [45, 65]. Влияние скорости движения начинает сказываться при $v_x > 80\text{--}100$ км/ч. Но в целом параметры вышеперечисленных факторов для конкретного грузового автомобиля и шины в условиях эксплуатации за исключением коэффициента сопротивления качения можно принять постоянными. Наименее изучено изменение коэффициента сопротивления качения от качества дорожного покрытия (ее микропрофиля).

В процессе обработки полученных экспериментальных данных продольных и вертикальных ускорений, высоты микропрофиля дороги замечено, что их отношение коррелируется со значениями коэффициента сопротивления качения (таблица 3.4, рисунок 3.39).

Для дорог с твердым покрытием (асфальт, бетон) за исключением разбитой грунтовой дороги (5) наблюдается линейная зависимость, которую можно описать следующим уравнением

$$f_k = K \cdot \sigma + f_0$$

где K – коэффициент влияния микропрофиля дороги;

σ – среднеквадратическое текущее значение микропрофиля дороги,
 f_0 – коэффициент сопротивления качения шины на стенде.

Для широкопрофильной шины 16.00 R20 приняты значения: $K=0,25$; $f_0=0,0135$. Данная зависимость позволяет учесть влияния микропрофиля дороги на коэффициент сопротивления качения. И исходя из этого рассчитывалась тяговая динамика автомобиля.

Таблица 3.4 – Значения среднеквадратических величин перемещений, скорости, ускорений микропрофиля дорог

Частота, Гц	Значения среднеквадратических величин перемещения, скорости, ускорений микропрофиля дорог														
	По перемещению, мм					По скорости см/с					По ускорению, м/с ²				
	Номер дороги														
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Значения в октавных и широких полосах частот															
2	0,49	0,45	1,9	2,22	3,19	0,69	0,68	2,5	2,96	4,24	0,27	0,36	0,69	0,63	0,95
4	0,44	0,6	1,9	2,41	4,03	1,5	2,05	5,84	7	1,21	1,09	1,54	2,6	2,8	4,42
8	0,75	0,92	3,43	1,71	6,13	5,18	7,22	19,04	19,7	28,2	5,2	7,25	12,2	12,1	14,6
16	1,09	1,5	2,99	3,03	3,95	14,1	15,9	20,9	20,2	22	12,7	18,2	17,4	16,2	15,8
2-16	14,8	1,91	5,29	5,8	8,92	12,5	17,6	29	29,2	38,0	13,8	19,6	21,4	20,4	22
8-16	1,32	176	4,55	4,79	7,3	12,4	17,5	28,3	28,2	36,8	13,7	19,6	21,2	20,2	21,6

Примечание. 1 – асфальтовое шоссе Вишневка–Новинки, 2 – бетонное шоссе МКАД-2, 3 – шоссе Белробот–Семков городок– перекресток МКАД-2, 4 – перекресток МКАД-2 – Вишневка, 5 – гравийная разбитая дорога

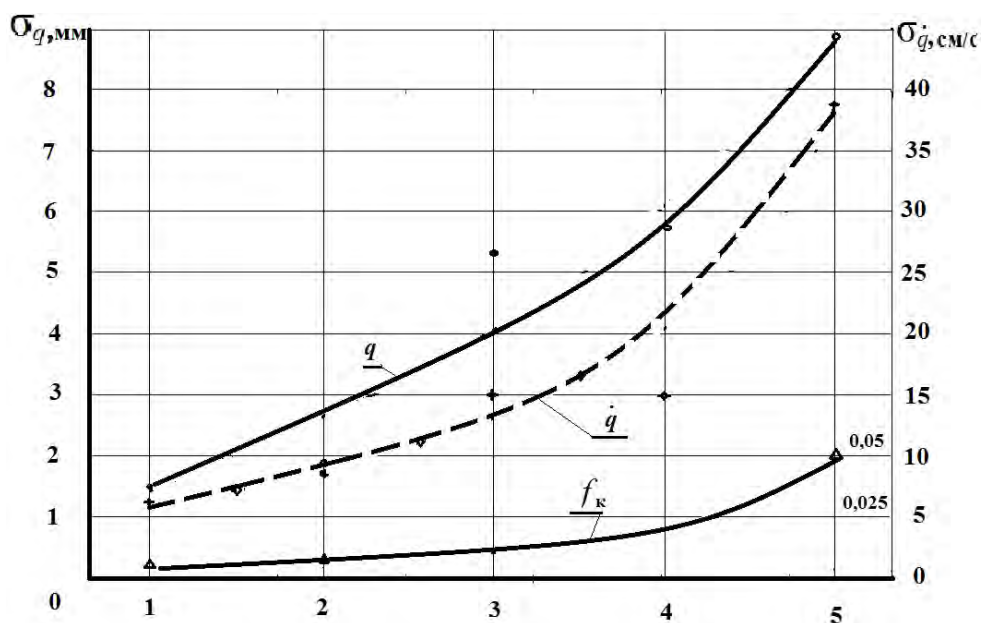
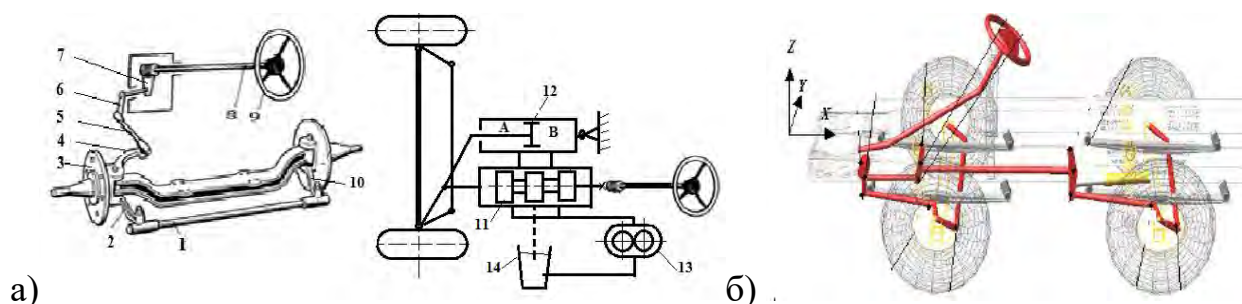


Рисунок 3.39. – Изменение среднеквадратической высоты, скорости микропрофиля дороги и коэффициента сопротивления качения на различных дорогах

3.12 Модель рулевого управления

3.12.1 Рулевое управление автомобиля представляет собой совокупность механизмов, предназначенных для поворота управляемых колес, обеспечивающих движение автомобиля в заданном направлении. Их общие схемы рулевого управления грузовых автомобилей приведены на рисунке 3.40.



а) 1 - поперечная тяга; 2, 10 - нижний рычаг; 3 - поворотная цапфа; 4 - верхний рычаг; 5 - продольная тяга; 6 - сошка рулевого привода; 7 - рулевая передача; 8 - рулевой вал; 9 - рулевое колесо, 11 – распределитель, 12 – исполнительный цилиндр, 13 – гидронасос, 14 – бак

Рисунок 3.40 – Общие схемы рулевого управления грузового автомобиля с одной управляемой осью (а) и с двумя управляемыми осями (б)

При вращении водителем рулевого колеса усилие передается посредством рулевого механизма, тяг и рычагов на поворотные кулаки, которые поворачиваются на определенный угол (заданный водителем), изменяя направление движения автомобиля.

Для облегчения управления автомобилем применяются гидро или электрические усилители руля. На грузовых автомобилях применяются гидроусилители следующих типов: интегрированные в механизм и полуинтегрального типа. Последний применяется в рулевом механизме МАЗ-64229-3400010 с распределителем.

Особенностью кинематики рулевого управления является его связанность с подвеской автомобиля, что обуславливает применение упругих элементов в шарнирах тяг. Из-за чего и неправильно подобранных параметров золотников, а также кинематики возможны резонансные колебания в рулевом управлении, что наблюдалось на самосвале МЗКТ-65151 со схемой (рисунок 3.43б). Для устранения которых пришлось пожертвовать чувствительностью/информативностью управления.

Для проведения расчетов рулевого управления сейчас используются пакеты MS.ADAMS и Simulink, ранее ЦВМ, АВМ. В работе [39] рассмотрено определение параметров рулевого управления и производительности насоса при повороте колес “на месте” с использованием стандартных элементов и кинематических связей в пакете MS.ADAMS.

Более сложным вопросом является моделирования управляемости автомобиля на реальной дороге с учетом гидроусилителя руля и колебаний подвески. В работе [39], выполненной также в пакете MS.ADAMS, сделана попытка рассмотреть взаимосвязь ходов рессоры и рулевого управления при синусоидальных возмущениях от дороги. Однако последнее в неполной мере отражают реальный процесс воздействия.

Второй особенностью рулевого управления является установка осей колес под небольшим углом для обеспечения стабилизации прямолинейного движения (рисунок 3.41).

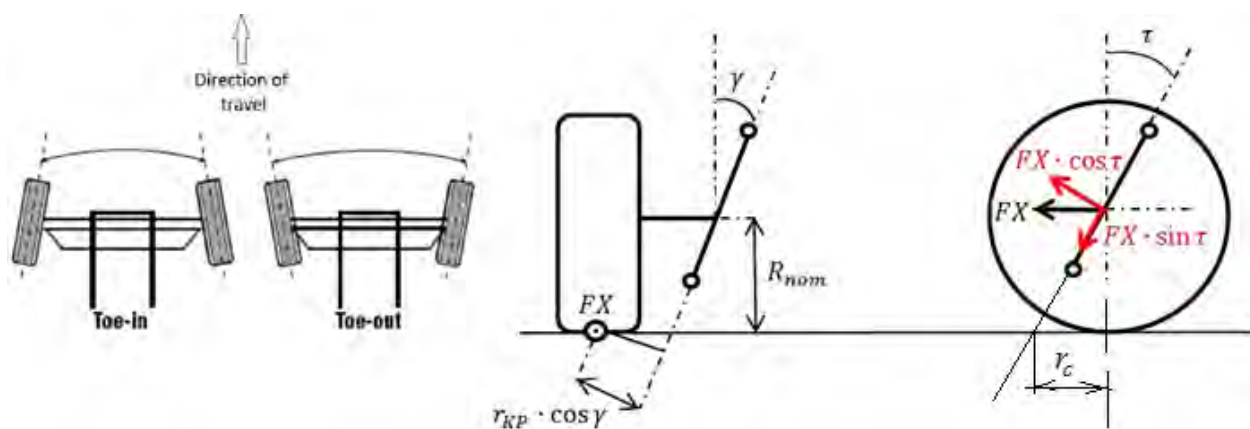


Рисунок 3.41 – Схема установки колес для стабилизации прямолинейного движения.

3.12.2. Усилия и моменты, действующие в рулевом управлении

При определении их необходимо учитывать, что поперечная сила не приложена к центру участка контакта шины, а на расстоянии от него, как показано на рисунке 3.44.

Расчетный момент сопротивления повороту на поворотных кулаках управляемых колес при повороте "на месте" может быть определен по формуле:

$$M_p = M_f + M_\varphi + M_h + M_{тр}, \quad (3.13)$$

где M_f – момент сопротивления качению колес; M_φ – момент сопротивления скольжению, возникающий при повороте "на месте", M_h – момент сопротивления за счет углов поворота шкворня; $M_{тр}$ – момент сопротивления трению в шарнирах рулевой трапеции и поворотного устройства.

$$M_f = 2 \cdot m_k \cdot f \cdot \alpha \quad (3.14)$$

где m_k – масса, приходящаяся на одно управляемое колесо; f – коэффициент сопротивления качению, $f \approx 0,0018$; α – радиус обкатки колеса вокруг шкворня, см

$$M_\varphi = 2 \cdot m_k \cdot f \cdot r_{обк} \quad (3.15)$$

Для большинства автомобилей $r_{обк} \approx 40\text{--}60$ мм, для грузовых автомобилей $r_{обк} \approx 60\text{--}100$ мм.

$$M_{\varphi} = 2 \cdot m_k \cdot \varphi \cdot r_{\psi} \quad (3.16)$$

где φ – коэффициент сцепления колес с дорожным покрытием, $\varphi \approx 0,8$;

r_{ψ} – плечо момента трения скольжения (при повороте), $r_{\psi} \approx 0,14 \cdot r_c$;

r_c – радиус качения свободного колеса.

Момент сопротивления подъему передней части колес определяется по формуле

$$M_h = m_k \cdot r_{обк} \cdot n \cdot [\lambda \cdot (\sin \alpha_{ср} + \sin \beta_{ср}) + \gamma \cdot (\cos \alpha_{ср} + \cos \beta_{ср})], \quad (3.17)$$

где n – число управляемых мостов; $\alpha_{ср}$ и $\beta_{ср}$ – средний угол поворота внутренних и наружных колес автомобиля (можно принимать 0,5 от наибольшего угла);

λ и γ – углы наклона шкворня вбок и назад.

Момент трения в шкворневом устройстве и в шарнирах рулевой трапеции M_t может быть рассчитан путем деления суммы моментов $M_f + M_{\varphi} + M_h$ на кпд шкворневого устройства $\eta_{шу}$ и кпд шарниров рулевой трапеции. По данным МАЗ величину $\eta_{шу}$ можно принять равной 0,95, а величину $\eta_{рт}$ равной 0,94.

Исходя из этого

$$M_p = \frac{M_f + M_{\varphi} + M_h}{\eta_{шу} \cdot \eta_{рт}} = \frac{M_f + M_{\varphi} + M_h}{0,95 \cdot 0,94} \quad (3.18)$$

Зная момент на поворотных кулаках, можно определить усилие на рулевом колесе, необходимое для поворота управляемых колес "на месте",

$$P_{рк} = \frac{M_p}{i_{рм} \cdot i_{рп} \cdot \eta_{рм} \cdot \eta_{рк}} \quad (3.19)$$

где $i_{рм}$ – передаточное число рулевого механизма; $i_{рп}$ – передаточное число рулевого привода; $\eta_{рм}$ – кпд рулевого механизма; $\eta_{рк}$ – кпд рулевого колеса и шарниров колонки.

$$i_{рп} = \frac{l_{рычага цапфы}}{l_{сошки}} \quad (3.20)$$

3.12.3 Динамическая модель рулевого управления с гидроусилителем

Динамическую модель рулевого управления с гидроусилителем можно представить в виде, показанном на рисунке 3.42.

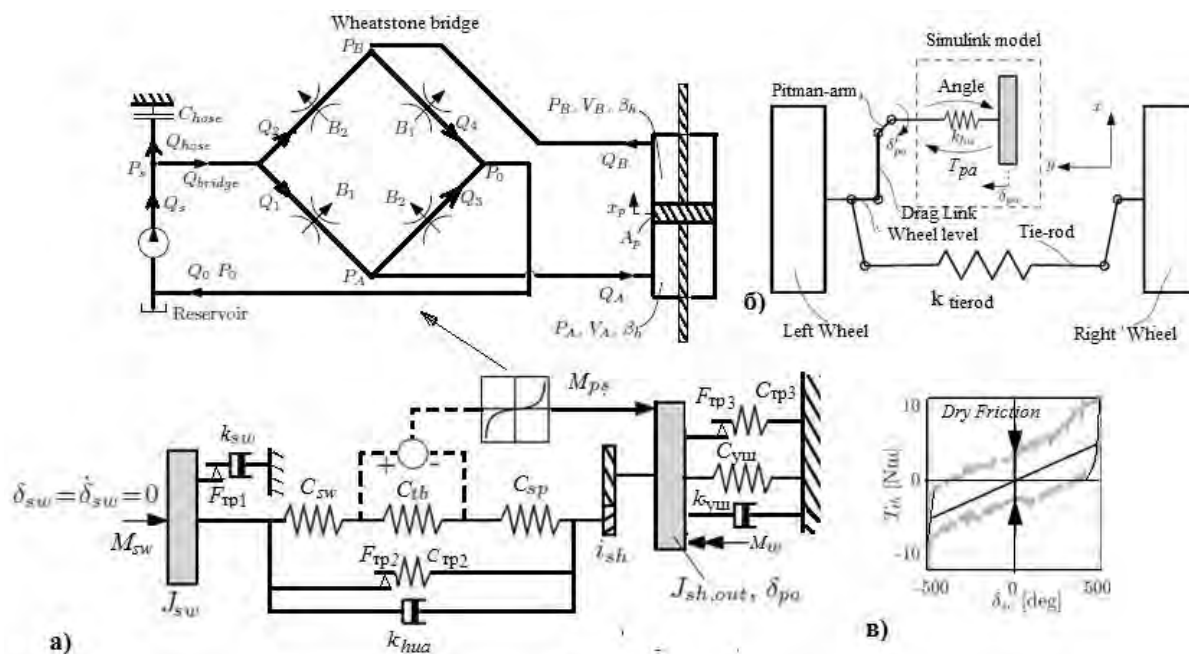


Рисунок 3.42 – Предлагаемая модель рулевого управления с гидроусилителем (а, б) и характер трения в ее системе (в)

Данная модель может быть описана следующими уравнения:

$$J_{sw} \ddot{\delta}_{sw} = M_{sw} - C_{tp1}(\delta_{sw} - \delta_{pa} \cdot i_{sh}) - k_{hua} \cdot (\dot{\delta}_{sw} - \dot{\delta}_{pa} \cdot i_{sh}) - \dot{\delta}_{sw} k_{sw} \quad (3.21)$$

$$J_w \cdot \ddot{\delta}_{pa} = C_{yvu} \cdot i_{sh}(\delta_{sw} - \delta_{pa} \cdot i_{sh}) + d_{hua} \cdot i_{sh}(\dot{\delta}_{sw} - \dot{\delta}_{pa} \cdot i_{sh}) \cdot i_{sh} + M_{ps} - C_{out} \cdot \delta_{pa} - k_{ha} \cdot \dot{\delta}_{pa} \quad (3.22)$$

где J_{sw} – момент инерции рулевого колеса, включая вал, карданный привод, часть рулевого механизма; J_w – момент инерции колес вокруг шкворня; $\ddot{\delta}_{sw}$, $\dot{\delta}_{sw}$, δ_{sw} – соответственно угловое ускорение, скорость, угол поворота рулевого колеса; $\ddot{\delta}_w$, $\dot{\delta}_w$, δ_w – соответственно угловое ускорение, скорость, угол поворота управляемых колес, $-C_{sw}$, C_{tb} , C_{sp} – крутильная жесткость соответственно рулевого колеса, вала, карданов; торсиона золотника; привода трапеции; i_{sh} – передаточное число рулевого механизма.

Важным фактором является правильное описание трения в системе рулевого управления. В работах [19, 139, 140] используются разновидности модели “сухого” трения и линейного сопротивления. Первая неадекватно описывает процесс и приводит к неустойчивости системы и ее автоколебаниям, вторая не учитывает нелинейности.

В тоже время приведенные в работах [139–140] экспериментальные характеристики (рисунок 3.42в [140]) и кадры видео [1] свидетельствуют о характере трения в характеристике увода шины аналогичным работе [51]. Последние лучше описывается зависимостью в виде интеграла от относительной скорости с ограничением максимальных величин

$$F_{\text{увш}} = \begin{cases} C_{\text{уш.дин1}} \int_0^t (\dot{\delta}_{\text{прк}} - \dot{\delta}_{\text{па}}) dt & \text{если } C_{\text{уш.дин1}} \int_0^t (\dot{\delta}_{\text{прк}} - \dot{\delta}_{\text{па}}) dt > F_{\text{огр.усил}} \\ -F_{\text{огр.усил}} & \text{если } C_{\text{уш.дин1}} \int_0^t (\dot{\delta}_{\text{прк}} - \dot{\delta}_{\text{па}}) dt < -F_{\text{огр.усил}} \end{cases} \quad (3.23)$$

$$F_{\text{рес}} = F_{\text{р.ст}} + F_{\text{тр1}} + F_{\text{тр2}} + F_{\text{ам}};$$

где $C_{\text{ру.ст}}$ – приведенная жесткость рулевого управления в статике, $C_{\text{ру.дин1}}$, $C_{\text{ру.дин2}}$ – жесткость рессоры при блокировке подвески, $F_{\text{тр1}}$, $F_{\text{тр2}}$ – величина усилия при трении 1, 2, $F_{\text{ру1max}}$, $F_{\text{ру2max}}$ – максимальная величина трения, $k_{\text{в.тр}}$ – коэффициент “вязкого” трения, $F_{\text{м в.тр}}$ – максимальная величина трения.

При использовании этой модели трения при колебаниях автомобиля на реальной дороге обеспечивается погрешность 5–15 % с учетом спектра. В то время при линейной модели и “сухого” трения погрешность достигает 50–100 %.

Следует отметить, что такой же вид имеет характеристика увода шины [21], которую также целесообразно описать подобной зависимостью.

С учетом необходимости моделирования в реальном масштабе времени при ограниченных возможностях компьютеров (4-5 ГГц) увеличение числа элементов модели и жесткостей промежуточных частей рулевого механизма и гидравлического привода, жесткости шлангов и жидкости путем учета их масс не оправдано поскольку они имеют собственные частоты выше 15–30 Гц. В противном случае каждый элемент потребует описывать дифференциальным уравнением 3-го порядка, (с учетом трения), что приводит к уравнениям 10-го порядка и более. Также необходимо учитывать, что реально в упомянутых пакетах можно моделировать процессы только с частотой до 20–30 Гц. Поэтому более целесообразным представляется ограничиться предлагаемой моделью, где податливости рулевой трапеции и другие объединены с характеристикой увода шины. В дальнейшем более целесообразным является использование полунатурного моделирования, описанного в подразделе 7.3.

Для оценки фактора гидроусилителя руля рассмотрим баланс расхода жидкости и давления в его системе с использованием моста Уитстона (рисунок 3.42(а)), где p_s – давление подачи, давление в камерах B , p_B , A , p_A ,

давление в отсеке B , p_B , и обратное давление p_0 . Поток через систему обозначен Q_{bridge} и Q_0 , потоки к полостям обозначены Q_{xA} и Q_{xB} .

Используя баланс сохранения расходов входящих и выходящих узлов, показанных на рисунке 3.42 имеем

$$\begin{cases} -Q_s + Q_{hose} + Q_{bridge} \\ Q_1 + Q_2 - Q_{bridge} \\ -Q_1 + Q_2 + Q_x \end{cases} = 0 \quad (3.24)$$

$$\begin{cases} Q_1 - Q_2 - Q_x \\ -Q_1 - Q_2 + Q_{bridge} \end{cases}$$

Входящий приток рассматривается как отрицательный:

$$Q_1 + Q_2 - Q_{bridge} = 0$$

Перепад давления в золотнике по дросселирующим отверстиям, их сопротивления связаны следующими зависимостями:

$$\begin{cases} p_s - p_A \\ p_s - p_B \\ p_B \end{cases} = \begin{cases} Q_1^2 \cdot R_1^2 \\ Q_2^2 \cdot R_2^2 \\ Q_1^2 \cdot R_1^2 \end{cases} \quad (3.25)$$

$$\{ p_A \} = \{ Q_2^2 \cdot R_2^2 \}$$

Эквивалентное сопротивление полного моста определяется как

$$R_{bridge} = \frac{1}{2} \sqrt{R_1^2 + R_2^2}.$$

Используя это определение, выражение для Q_1 и Q_2 может быть записано.

$$\begin{cases} 2Q_1 - Q_{bridge} - Q_x \\ 2Q_2 - Q_{bridge} + Q_x \\ Q_s - Q_{bridge} - Q_{hose} \end{cases} = \begin{cases} 0 \\ 0 \\ 0 \end{cases} \quad (3.26)$$

$$\begin{cases} p_A \\ p_B \\ p_s \end{cases} = \begin{cases} Q_2^2 \cdot R_2^2 \\ Q_1^2 \cdot R_1^2 \\ Q_1^2 \cdot R_1^2 + Q_2^2 \cdot R_2^2 \end{cases}$$

Учитывая симметричность, избыточность уравнений (3.24) и равенств в втором и пятом ряду $Q_1 + Q_2 + Q_{bridge}$, и в третьем и четвертом ряду $Q_1 + Q_2 + Q_x$ при помощи явного определения p_A и p_B , приведенного в (3.67), p_s может быть записано как $p_s = Q_1^2 \cdot R_1^2 + Q_2^2 \cdot R_2^2$.

Это приводит к следующему результату

$$Q_1 = \frac{\sqrt{p_B}}{R_1}; \quad Q_2 = \frac{\sqrt{p_A}}{R_2}; \quad Q_{bridge} = \frac{\sqrt{p_s}}{R_{bridge}} \quad (3.27)$$

Две первых записи в (3.69) определены для Q_x и связаны с давлениями при помощи выражения в (3.68). С учетом сжимаемости в полостях, поток перемещаемый в камеру А не обязательно равен потоку от полости В. Для различия потоков, перемещаемых в камеру А и поток из полости В используются два выражения для Q_x

$$Q_x = \left\{ \begin{array}{l} Q_{bridge} - 2 \cdot Q_2 = \frac{\sqrt{p_s}}{R_{bridge}} - 2 \cdot \frac{\sqrt{p_A}}{R_2} = Q_{xA} \\ 2 \cdot Q_1 - Q_{bridge} = 2 \cdot \frac{\sqrt{p_B}}{R_1} - \frac{\sqrt{p_s}}{R_{bridge}} = Q_{xB} \end{array} \right\} \quad (3.28)$$

Производные давления могут быть вычислены путем использования уравнения непрерывности расхода жидкости:

$$-\sum Q = \frac{dV}{dt} + \frac{V}{\beta} \cdot \frac{dp}{dt} \quad (3.29)$$

где V – объем полости. Объемы полостей определяются как:

$$V_A = V_{A0} + F \cdot x \text{ и } V_B = V_{B0} - F \cdot x, \quad (3.30)$$

где – V_{A0} и V_{B0} начальные объемы полостей, F – площадь поршня и x – перемещение поршня. Применяя уравнение (3.71), к p_s , p_A , и p_B и произведя замену их из уравнений (3.69) и (3.70) вытекает

$$\frac{dp_s}{dt} = \frac{Q_s - Q_{bridge}}{C_{hose}} = \frac{Q_s - \frac{\sqrt{p_s}}{R_{bridge}}}{C_{hose}}, \quad (3.31)$$

$$\frac{dp_A}{dt} = \left(+Q_{xF_A} - \frac{V_A}{dt} \right) \cdot \frac{\beta}{V_A} = \frac{\left(\frac{\sqrt{p_s}}{R_{bridge}} - 2 \cdot \frac{\sqrt{p_A}}{R_2} - F \cdot \dot{x} \right) \cdot \beta}{V_{A0} + V_{A0} \cdot x}; \quad (3.32)$$

$$\frac{dp_B}{dt} = \left(-Q_{xF_B} - \frac{V_B}{dt} \right) \cdot \frac{\beta}{V_B} = \frac{\left(\frac{\sqrt{p_s}}{R_{bridge}} - 2 \cdot \frac{\sqrt{p_B}}{R_2} + F \cdot \dot{x} \right) \cdot \beta}{V_{B0} + F \cdot x}; \quad (3.33)$$

3.12.4 Уравнения динамики рулевого управления в упрощенном виде

3.12.4.1 Привод

Входным параметром для привода является угол поворота рулевого управления колеса α , а выходным – перемещение золотника $x_{\text{зол}}$ распределителя. Считая, что привод обладает большой жесткостью и малой массой по сравнению с другими элементами, имеем

$$x_{\text{зол}} = \frac{l_c}{i_{\text{рм}}} \cdot \alpha \quad (3.34)$$

где l_c – расчетная длина сошки, $i_{\text{рм}}$ – передаточное число рулевого механизма.

3.12.4.2 Распределитель и трубопроводы

Распределитель (рисунок 3.42а) регулирует поступление жидкости в рабочие полости силового цилиндра. Его характеристика представляет зависимость между расходом жидкости Q в рабочие полости цилиндра и величиной смещения золотника относительно корпуса

$$\Delta x = x_{\text{зол}} - x_{\text{корп}} \quad (3.35)$$

Для распределителя с отрицательным перекрытием характеристика имеет вид, показанной на рисунке 3.42в.

Расходная характеристика обычно снимается при давлении линии питания

$$P_H = 0,5 (P_A + P_B), \quad (3.36)$$

где P_A , P_B – давление жидкости в полостях А и В цилиндра гидроусилителя, Н/м².

Распределитель характеризуется коэффициентом усиления, определенным для перепада давления $0,5 (P_A + P_B)$.

$$K = \frac{dQ}{dx} \quad (3.37)$$

Он представляется собой наклон (крутизну) характеристики $Q = f(\Delta x)$ в средней части. Пусть Q_a , Q_b – расходы в рабочих полостях А и В силового цилиндра. Тогда уравнения расхода

$$\begin{aligned} Q_a &= F_{\text{ц}} \cdot \frac{dx_{\text{п}}}{dt} + \left(\frac{V}{\beta} + e \right) \cdot \frac{dP_a}{dt}; \\ Q_b &= -F_{\text{ц}} \cdot \frac{dx_{\text{п}}}{dt} + \left(\frac{V}{\beta} + e \right) \cdot \frac{dP_b}{dt}; \end{aligned} \quad (3.38)$$

где $x_{\text{п}}$ – перемещение поршня, см; $F_{\text{ц}}$ – рабочая площадь силового цилиндра, м². V – объем полости цилиндра в среднем положении поршня, м³; β – коэффициент сжимаемости жидкости, для стандартной жидкости, используемой в гидросистемах, $\beta = 17000$ Н/м²; e – объемная податливость гидросистемы, включающая шланги и трубопроводы, равная $e = \frac{dV}{dp}$, м³/Н.

Первые слагаемые правой части уравнений (3.80) соответствуют изменению объемов полостей А и В силового цилиндра при перемещении поршня, а вторые слагаемые – изменению объемов гидросистемы вследствие ее податливости.

$$Q_a = K \cdot \Delta x \sqrt{\frac{P_a - P_b}{0,5 \cdot P_H}}; \quad Q_b = -K \cdot \Delta x \sqrt{\frac{P_b}{0,5 \cdot P_H}}; \quad (3.39)$$

при повороте направо

$$Q_a = -K \cdot \Delta x \sqrt{\frac{P_a}{0,5 \cdot P_H}}; \quad Q_b = -K \cdot \Delta x \sqrt{\frac{P_a - P_b}{0,5 \cdot P_H}}, \quad (3.40)$$

при повороте налево.

Исходя из условия неразрывности потока жидкости

$$Q_a + Q_b = 0; \quad (3.41)$$

уравнение (3.83) удовлетворяется при,

$$P_H = P_a + P_b; \quad (3.42)$$

то есть давления P_a и P_b остаются симметричными относительно $0,5 \cdot P_H$.

Исходя из этого можно написать

$$p_a = 0,5 \cdot p_H + \frac{P_a - P_B}{2}; P_B = 0,5 \cdot p_H + -\frac{P_a - P_B}{2},$$

В результате уравнения (3.83, 2.84) приводятся к виду

$$Q_a = -Q_B = K \cdot \Delta x \sqrt{1 - \varepsilon \frac{P_a - P_B}{P_H}} \quad (3.43)$$

где $\varepsilon = +1$ при повороте вправо и $\varepsilon = -1$ при повороте влево.

Подставляя 2.85 в 2.81 имеем

$$F_{\text{сц}} \cdot \frac{dx_{\text{п}}}{dt} + \left(\frac{V}{\beta} + e\right) \cdot \frac{dp_a}{dt} = K \cdot \Delta x \sqrt{1 - \varepsilon \frac{P_a - P_B}{P_H}}, \quad (3.44)$$

$$F_{\text{сц}} \cdot \frac{dx_{\text{п}}}{dt} + \left(\frac{V}{\beta} + e\right) \cdot \frac{dp_a}{dt} = K \cdot \Delta x \sqrt{1 - \varepsilon \frac{P_a - P_B}{P_H}}. \quad (3.45)$$

Вычитая 2.87 из 2.86

$$2 \cdot F_{\text{сц}} \cdot \frac{dx_{\text{п}}}{dt} + \left(\frac{V}{\beta} + e\right) \cdot \frac{dp_a}{dt} = 2 \cdot K \cdot \Delta x \sqrt{1 - \varepsilon \frac{P_a - P_B}{P_H}}. \quad (3.46)$$

или

$$2 \cdot F_{\text{сц}}^2 \cdot \frac{dx_{\text{п}}}{dt} + \left(\frac{V}{\beta} + e\right) \cdot \frac{dM}{dt} = 2 \cdot F_{\text{сц}} \cdot l_{\text{сц}} \cdot Q_{\Delta x} \sqrt{1 - \varepsilon \frac{P_a - P_B}{P_H}}. \quad (3.47)$$

так как

$$M = (P_a + P_B) \cdot F_{\text{сц}} \cdot l_{\text{сц}}; M < M_H = P_H \cdot F_{\text{сц}} \cdot l_{\text{сц}},$$

где $l_{\text{сц}}$ – расчетная длина рычага, связанного с силовым цилиндром, представляет собой момент со стороны гидроцилиндра, действующий на колеса автомобиля

$$Q_{\Delta x} = \frac{dQ}{dt} \cdot \Delta x. \quad (3.48)$$

3.12.4.3 Силовой цилиндр

Перемещение поршня $x_{\text{п}}$ цилиндра равно

$$x_{\text{п}} = l_{\text{ц}} \cdot \theta \quad (3.49)$$

где θ – текущее значение усредненного угла поворота управляемых колес.

С учетом этого равенства уравнение 2.87 принимает вид

$$2 \cdot F_{\text{сц}}^2 \cdot l_{\text{ц}}^2 \cdot \frac{d\theta}{dt} + \left(\frac{V}{\beta} + l_{\text{т}} \right) \cdot \frac{dM}{dt} = 2 \cdot F_{\text{сц}} \cdot l_{\text{сц}} \cdot Q_{\Delta x} \sqrt{1 - \varepsilon \frac{p_a - p_b}{P_H}}. \quad (3.50)$$

3.12.4.4 Управляемые колеса

Учитывая, что колебания управляемых колес происходит с относительно малыми амплитудами (до 3–4°), уравнение движения для колес можно записать в виде

$$J\ddot{\theta} + r \cdot \dot{\theta} + c \cdot \theta = M - M_p, \quad (3.51)$$

где J – момент инерции колес относительно осей шкворней, Н·м/м²,
 θ – крутильная жесткость шины с учетом стабилизирующего момента, возникающего вследствие наклона шкворней, Н·м;
 r – коэффициент трения в шинах колес, гидросистеме и приводе, Н·м/с;
 M_p – момент, действующий на колеса со стороны реактивных камер распределителя руля, Н·м. Учитывая, что $M_p \ll M$, им можно пренебречь.

3.12.4.5 Обратная связь

Соединяет управляемые колеса с корпусом распределителя и обеспечивает следящее действие системы.

В результате смещение золотника относительно корпуса распределителя равно

$$\Delta x = x_{\text{зол}} - x_{\text{корп}} = \frac{l_{\text{ас}} - P_b}{l_{\text{рм}}} \cdot \alpha - l_p \cdot Q_1, \quad (3.52)$$

где l_p – расчетная длина рычага, связанного с распределителем. Таким образом, система описывается следующими уравнениями движения;

$$\begin{aligned} 2 \cdot F_{\text{сц}}^2 \cdot l_{\text{ц}}^2 \cdot \frac{d\theta}{dt} + \left(\frac{V}{\beta} + l_{\text{т}} \right) \cdot \frac{dM}{dt} &= 2 \cdot F_{\text{сц}} \cdot l_{\text{сц}} \cdot Q_{\Delta x} \sqrt{1 - \varepsilon \frac{M}{M_H}}, \\ J \cdot \frac{d^2 Q}{dt^2} + r \cdot \dot{\theta} + c \cdot \frac{d\theta}{dt} + c \cdot Q &= M, \\ \Delta x &= \frac{l_c}{l_{\text{рм}}} \alpha - l_p \cdot Q \end{aligned} \quad (3.53)$$

Для определения динамических характеристик рулевого управления (устойчивость, быстродействие и т.п.) в Matlab/Simulink уравнение представлено в виде

$$\frac{dM}{dt} = \alpha_1 \cdot Q_{\Delta x} \sqrt{M_H - \varepsilon \cdot M} - \alpha_2 \cdot \frac{d\theta}{dt}; \quad (3.54)$$

$$\frac{d^2 Q}{dt^2} = \frac{1}{J} M - \frac{r}{J} \frac{d\theta}{dt} - \frac{c}{J} \cdot Q = 0; \Delta x = \frac{l_c}{l_{pm}} \alpha - l_p \cdot Q_1, l_{ц}^2, \quad (3.55)$$

$$\alpha_1 = \frac{2 \cdot F_{сц} \cdot l_n \cdot Q_n}{\left(\frac{V}{\beta} + \varepsilon\right) \sqrt{M_n}}; \quad \alpha_2 = \frac{2 \cdot F_{сц}^2 \cdot l_{ц}^2}{\left(\frac{V}{\beta} + \varepsilon\right)}. \quad (3.56)$$

3.13 Методика моделирования

На основании проведенного анализа опубликованных работ, собственных исследований и сформулированных задач предлагается следующая методика моделирования, показанная схематично на рисунке 3.51.



Рисунок 3.43. – Методика моделирования динамической системы “дорога–автомобиль–водитель”

Предлагаемая методика включает:

- разработку новых методов получения продольных и микропрофилей дорог, как основного источника возмущения на автомобиль и водителя, характеристик основных элементов подвески (глава 2);
- создание более совершенных моделей системы “дорога–автомобиль–водитель”, обеспечивающих расхождение не более 10–15 % против 50 % у существующих моделей и методик, что дает возможность оценивать машину путем имитационного моделирования со смешанной реальностью;
- разработку метода совместного моделирования движения и колебаний автомобиля с учетом реального макро и микропрофиля дороги, режима движения, характеристик двигателя, топливной экономичности с использованием модуля S-Function Builder Simulink и реализацию через него управления процессом моделирования, логики переключения передач, передачу информации на стенды для ресурсных испытаний;
- разработку метода оценки долговечности узлов колесных машин через цепочку этапов: моделирования движения и колебаний автомобиля,

получения по ним нагрузок, определение по последним напряжений в узлах через КЭМ модель, далее расчет долговечности по усталостным характеристикам материалов и систематизации данных напряжений методом дожда на примере оценки долговечности картера заднего моста;

- выявление источника продольных и поперечных ускорений автомобиля и разработку метода их реализации при моделировании;

- проведение на основе предложенных моделей и подходов всестороннего расчетного исследования движения, колебаний, управляемости и экономичности колесной машины и влияния параметров системы на них;

- разработку нового более простого метода визуализации дорожной обстановки при имитационном моделировании со смешанной реальностью на основе ее видеосъемки при фиксированной скорости движения автомобиля ($v = 20$ м/с) с последующим ее воспроизведением в программе с управляемой частотой просмотра синхронно с процессом в Simulink.

- выбор направления, подходов и программных средств реализации этой системы, включая имитационное моделирование и смешанную реальность;

- прогнозирование передвижения автомобилей по различным дорогам на примере специальных машин на основе полученных при эксперименте данных, карт местности из интернета, снимков с учетом на них дорожной обстановки.

- использование Matlab/Simulink с модулем S-Function Builder, мини-компьютера Raspberry, MC, организацию сети обмена информацией.

Предложенная методика включает вышеперечисленные методы, рассматриваемые в главах 2–7 и в подпунктах ниже. Она также учитывает условия эксплуатации автомобиля.

3.13.1 Дорожные факторы, влияющие на нагруженность машины

При проектировании автомобилей повышенной проходимости, самосвалов и штабных машин с кузовом-контейнером, необходимо учитывать специфические условия их эксплуатации – использование их вне дорожной сети: на проселочных, грунтовых, лесных дорогах, в поле, каменистых перекатах рек, в условиях природных, техногенных аварий. Кроме того, необходимо учитывать состояние местных дорог в течении длительного времени, многие из которых могут находиться в разбитом состоянии особенно после зимы (рисунок 3.44). Глубина выбоин 5–30 см. Степень воздействия на автомобиль (штабную машину), его узлы, части во многом зависит от высоты (глубины) неровности дороги, частоты волны, скорости, распределения движения по различным дорогам.



Рисунок 3.44. – Разбитые дороги с выбоинами, проломами и бездорожье

По данным работ [95, 96], максимальная высота неровностей может достигать 23,2–29,7 см. Кроме того, имеются лесные и сильно разбитые грунтовые дороги.

Нагруженность автомобилей, в том числе и штабных машин, во многом зависит также от распределения пробега по разным типам дорог. По данным работ [95, 98], пробег по дорогам Российской Федерации грузовых автомобилей распределяется согласно данным таблицы 3.1. Эти данные можно использовать при расчетах специальных и военных машин.

Для получения более достоверной оценки распределения движения по различным дорогам для штабных машин использовался программно-информационный комплекс принятия решения ГИС ВН, разработанный Институтом проблем информатики НАН Беларуси для нужд Министерства обороны. Он позволяет задать и смоделировать маршрут движения из точки А в точку Б на основе электронных топографических карт с учетом ограничений по движению и определить протяженность участков дорог различных категорий либо вручную по электронным картам просчитать протяженность заданного маршрута для различных дорог.

В результате проведенного моделирования маршрутов и обработки данных получены следующие значения распределения движения штабных машин по различным дорогам в мирных условиях (таблица 3.2) [97].

Анализ данных моделирования по распределению движения штабных машин по территории Республики Беларусь показал, что доля скоростных, обычных асфальтовых, грунтовых дорог составляет в среднем соответственно 11,4; 69,8; 16,9 %. Полученные данные, с одной стороны, отражают специфику дорожной сети Республики Беларусь, а с другой, они больше соответствуют условиям мирного времени, когда войска перемещаются в основном по дорогам общего пользования из пунктов их базирования на полигоны в процессе учений.

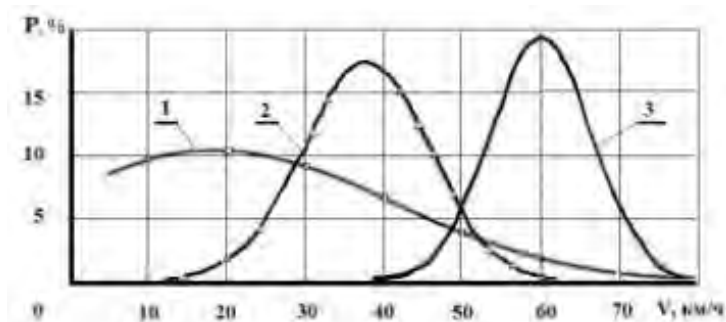
Таблица 3.1 – Распределение пробега грузовых автомобилей по разным видам дорог Российской Федерации

Относительный пробег грузовых автомобилей по дорогам РФ, км (%)					Относительное время действия процесса нагружения при заданной скорости движения			
Категория дороги	Северо-запад европ. части		Европейская часть	Восточный регион	$v_{i,}$ км/ч	$x_i = t_i / T$		
	Автомобили общ. назначения	Автомобили повыш. проходимости	Автомобили общего назначения	Автомобили общего назначения		Асфальт	Щебенка	Грунтовка
1	2	3	4	5	6	7	8	9
I	–	–	0.87	4	5			0,326
II	48	12.5	60.5	30.8	10			0,182
1	2	3	4	5	6	7	8	9
III	52	57.5	37.7	60.3	15		0,047	0,129
IV	–	30	0.93	4.9	20		0,111	0,098
					25			0,076
					30		0,193	0,058
					35		0,239	0,043
					40		0,209	0,031
					45	0,031	0,129	0,022
					50	0,114	0,055	0,015
					55	0,247	0,017	0,010
					60	0,300		0,006
					65	0,209		0,004
					70	0,081		
					75	0,019		

Таблица 3.2 – Распределение движения штабных машин по различным дорогам

Тип дороги	Пробег, км, по маршрутам 1—16																Среднее значение, км
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	
Общий пробег	33,2	174	137	136	193	129	149	86	139	71	85	70	55	65	250	330	131
Скоростная	16,5				1,78								15	9	150	100	18,3
Обычная асфальтовая дорога	12,2	168	136	13	183	125	80	65	134	30	63	29	30	41	80	200	94,6
Грунтовая	4,42	5,48	0,7	0	8,2	4,47	709	21	4,92	42	21	41	4,4	10	15	20	17,03
	Доля %																%
Скоростная	49,7					0,85							27,3	13,8	60	30,3	11,4
Обычная асфальтовая дорога	36,7	96,8	99,2	100	94,9	96,6	53,5	76,1	96,3	41,6	73,2	41,6	54,5	63,1	32	60,6	69,8
Грунтовая	13,3	3,14	0,8		4,2	3,34	46,9	23,9	3,54	58,3	24,5	58,3	8,04	15,4	6	6,6	16,9

Для условий военного времени, природных или техногенных катастроф для штабных машин целесообразно ориентироваться на данные исследования [98] самосвала «Урал-5557», предназначенного для эксплуатации в условиях бездорожья (рисунок 3.45).



1 – грунтовая, 2 – щебенка, 3 – асфальт

Рисунок 3.45. – Распределение скорости движения на дорогах

Последний в данном случае рассматривается как наиболее близкий аналог по условиям эксплуатации. Воспользуемся полученными экспериментальными данными по этому автомобилю [98] для обобщенной качественной оценки, связывающей тип автомобильной дороги и вероятностную скорость движения.

Полученные данные по автомобилю «Урал-5557» свидетельствуют, что изменение скоростей движения колесной машины хорошо описывается нормальным законом распределения. Существуют нижняя граница скорости движения, определяемая минимальной скоростью устойчивого движения автомобиля $v_{min}=5$ км/ч, и верхняя граница, определяемая техническими возможностями автомобиля и дорожными условиями $v_{max}^T = 75$ км/ч, что близко к скоростям колесной машины.

Используя эти данные и считая закон распределения мгновенных скоростей движения усеченным нормальным, можно определить математическое ожидание средней скорости колесной машины для разных типов дорог, среднеквадратическое отклонение (СКО) средней скорости и с их помощью вычислить долю ξ_i от общего пробега колесной машины с конкретной скоростью v_i . Значения ξ_i вычисляется как вероятность появления случайной величины в заданном интервале значений.

Полученные в работах [95, 96] данные по микропрофилю дорог в поле, представленные на рисунке 3.46, показывают, что возмущение в целом носит синусоидальный характер с наличием небольших высокочастотных гармоник.

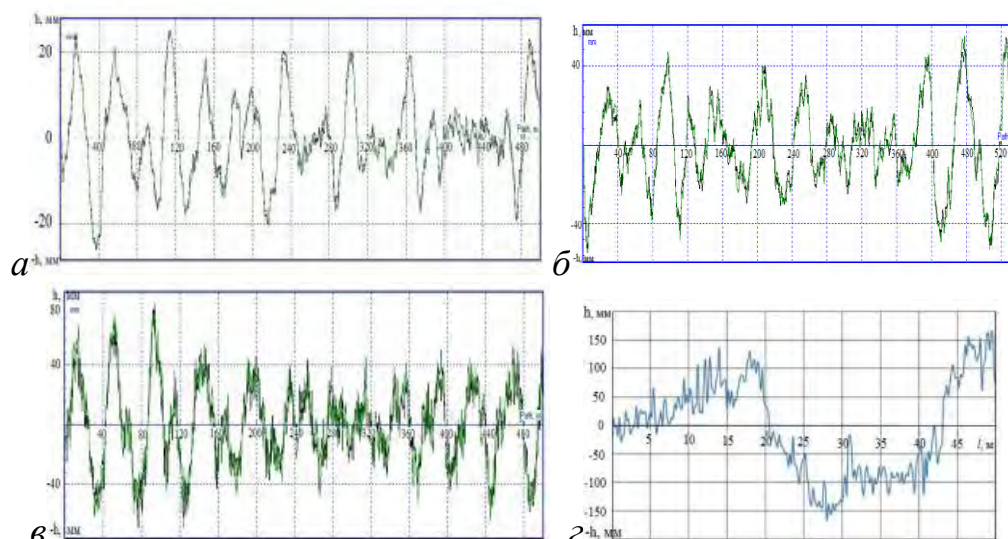


Рисунок 3.46. – Микропрофиль цементобетонного (а), асфальтового шоссе в удовлетворительном состоянии (б), булыжного шоссе (в) и заболоченного поля (г)

Низкочастотное возмущение, имеющее большую амплитуду, в основном и определяет наибольшие напряжения в кузове-контейнере, вызывающие его деформацию. Максимальные пиковые значения деформаций не превышают 22 мм на асфальтобетоне и 80 мм – на булыжном разбитом шоссе. Максимальное значение на заболоченном поле 150 мм.

Сравнивая результаты распределения движения по дорогам I–IV категории и соответствующие им значения максимальных неровностей, можно отметить, что в условиях мирного времени на этих дорогах не будет столь значительных нагрузок, которые могут иметь место в случае движения по разбитым дорогам или при переезде препятствий высотой 100 (одиночных), 250 мм (диагональных). Кроме того, надо учитывать, что в случае военных действий состояние дорог может резко ухудшиться вследствие бомбежек, авиационных обстрелов колонн из пушек и пулеметов, а также отсутствия возможностей поддержания дорог в нормальном состоянии. Также необходимо учитывать возможность задействования военных с их техникой, в том числе использование штабных машин, для координации работ при устранении природных катастроф, техногенных аварий. Величины неровностей в этих условиях будут большими – порядка 100–250 мм.

Что касается кузова-контейнера, то необходимо принять во внимание, что по данным эксплуатации, определяющим фактором является все же деформируемость его конструкции при кручении, не приводящая к повреждаемости, но отрицательно сказывающаяся на заклинивании дверей и выдвижных частей контейнера. Чаше это происходит, когда одновременно

имеет место большие значения и разность неровностей дорожного покрытия под левым и правым колесом машины. Поэтому для моделирования нагруженности штабных машин больше подходит режим переезда препятствий, особенно диагональных. В этом режиме проводится большинство ускоренных ресурсных испытаний военной колесной техники. Этот режим позволяет значительно сократить время моделирования (более чем в 20 раз), поскольку для переезда препятствий достаточен участок 10–20 м в отличие от моделирования колебаний колесной машины с заданным микропрофилем дороги, для которого требуется участок длиной 200–1000 м. Исходя из вышеизложенного для моделирования выбран режим переезда препятствий: одиночных (100 мм) и диагональных (230 мм), применяемых на автополигонах НАМИ (Москва) и в г. Бронницы.

3.13.2 Моделирование системы “дорога–автомобиль–водитель”

3.13.2.1 Использование Matlab/Simulink и модуля S-Function Builder

Для реализации имитационного моделирования в режиме “реального” времени необходимо использование 64-битного Matlab/Simulink с модулем S-Function. Builder. Данный модуль позволяет создать пользовательскую программу на языке C в виде своеобразной DLL [14], позволяющую реализовать внешнее возмущение от дороги в виде огромного массива (96000 точек) макро и микропрофиля дороги, переменную задержку времени воздействия на оси, логику переключения передач КПП, учет дорожных факторов, режима движения автомобиля, обмен информации с его органами управления через Raspberry чего не возможно осуществить в пакете MSC.ADAMS, что является новизной предлагаемого метода.

Использование Matlab/Simulink заключается в создании блок-схемы на основе разработанных дифференциальных уравнений, описывающих динамические процессы в подсистемах автомобиля.

Рассмотрим реализацию в Simulink на примере упрощенной двухмассовой системы колебаний водителя на сиденье, описанной следующим уравнением

$$m_{\text{вод}} \ddot{Z}_{\text{вод}} - C_{\text{сид}} (Z_{\text{пола}} - Z_{\text{вод}}) - k_{\text{амор}} (\dot{Z}_{\text{пола}} - \dot{Z}_{\text{вод}}) = 0;$$

или

$$\ddot{Z}_{\text{вод}} = \frac{1}{m_{\text{вод}}} (C_{\text{сид}} (Z_{\text{пола}} - Z_{\text{вод}}) + k_{\text{амор}} (\dot{Z}_{\text{пола}} - \dot{Z}_{\text{вод}})) = 0;$$

Для решения этого уравнения требуется два интегратора. Используя Simulink Library Brower, перетаскиваем на созданную новую модель два интегратора, три усилителя, два сумматора и два входа. Соединяем их согласно рисунка 3.47. Усилители используются для установки коэффициентов масштабирования $\frac{1}{m_{\text{вод}}}$, $C_{\text{сид}}$, $k_{\text{амор}}$, а сумматоры для суммирования возмущающего воздействия.

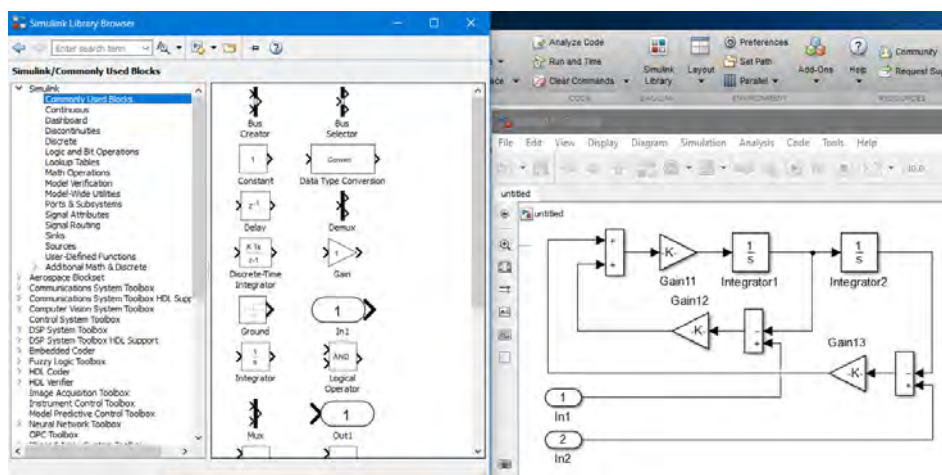


Рисунок 3.47. – Блок-схема колебаний водителя на сиденье

Начиная с версии Matlab/Simulink R2023 S-Function Builder используется компилятор MS Visual C/C++ 2022 v. 17.7 и новее (v. 17.12) для реализации программы на языке C. Компилятор на языке C (TDM-GC-64) использовался на более ранних версиях Matlab/Simulink R2015–R2022.

Вторым ключевым моментом является использование мини-компьютера Raspberry Pi 3/4/5 для реализации программы для считывания информации с органов управления автомобиля ввиду того, что программное обеспечение АЦП сильно замедляет процесс моделирования при реализации моделирования на одном компьютере.

В приложении Д рассматривается подробная инструкция по установке программного обеспечения на Raspberry Pi 3/4 в связи тем, что имеющиеся инструкции в интернете написаны и сконфигурированы под язык Python. Они не годятся для установки библиотеки визуализации OpenCV 4 и компилятора GCC 8.2.0/10 (C/C++) на Raspberry. Сложным вопросом также является организация работы с сетью. Путем анализа информации, собранной по крупицам, проведенному тестированию найдены их решения. Для чего пошагово подробно расписаны установка и настройка пакета OpenCV для C/C++, библиотеки АЦП_ЦАП bcm2835 и подходы по реализации

моделирования и создания программы, обеспечивающей высокое быстродействие. Освоение библиотек `bcm2835` и `OpenCV`, позволяющих распознавать объекты, является необходимым этапом для дальнейшего создания системы управления беспилотными автомобилями.

3.13.3 Настройка модуля S-Function Builder

Запускаем Matlab/Simulink R2024, затем Simulink Library. На окно с программой переносим модуль S-Function Builder. Дважды быстро нажав на него, начинаем его настраивать. В первую очередь дадим имя S-function name, например, `MPD` (рисунок 3.48). Рекомендуется давать программе Simulink *.slx имя отличное от S-function name.

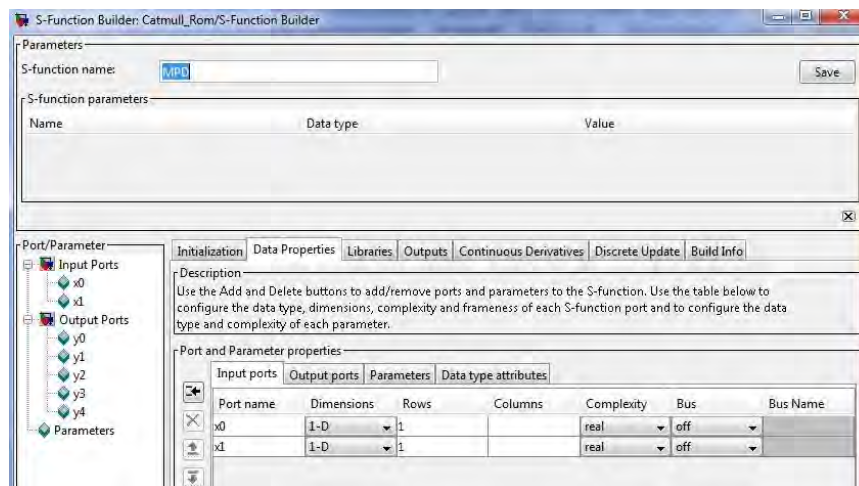


Рисунок 3.48. – Меню модуль S-Function Builder

В подменю `Data Properties` задаем входные и выходные параметры/переменные. Нажимаем `Save` и модуль S-Function Builder создает пустой шаблон программы, например `mpd.c`. В эту программу/необходимо внести свои переменные и сам реализующий текст на языке C/C++ и сохранить ее. Фрагмент внесенной части программы и места их расположения приведены в приложении Г.

3.13.4 Реализация возмущения дороги

Путем тестирования выяснено, что в S-Function Builder можно реализовать массив для хранения данных 384000 байт (`float rd[96000]`). При значении свыше происходит сбой.

Чтобы исключить многократную загрузку данных на каждом шаге вычисления используется счетчик `if (ch==0) {}`, в тело которого помещается текст программы. Он отслеживает переход на следующий блок загрузки.

Параметр 11892 зависит от шага записи данных. В данном случае массив сформирован с шагом $t_s=0,125$ м, 160 Гц. Значение 11892 получается, как $96000/8=12000$. Значение 8 определяется как количество точек на метр пути (1 м/0,125 м). Немного уменьшаем 12000 до 11892, чтобы не было переполнение массива 384000 байт (96000 записей).

Интерполирование значений промежуточных точек выполнено методом Catmull Rom, который наиболее подходит для этих целей. Другие методы (Ньютона, квадратической интерполяции) показали худшие результаты [99].

Компиляция программы выполняется в главном меню MATLAB командой `mex mpd.c`. После сообщения об ее успешной компиляции можно выполнять обычный процесс моделирования в Simulink.

На этот модуль можно возложить еще логические операции, например, логику управления двигателем и переключением передач КПП, как это реализовано автором при проведении совместного моделирования движения и нагруженности автомобиля (рисунок 3.49).

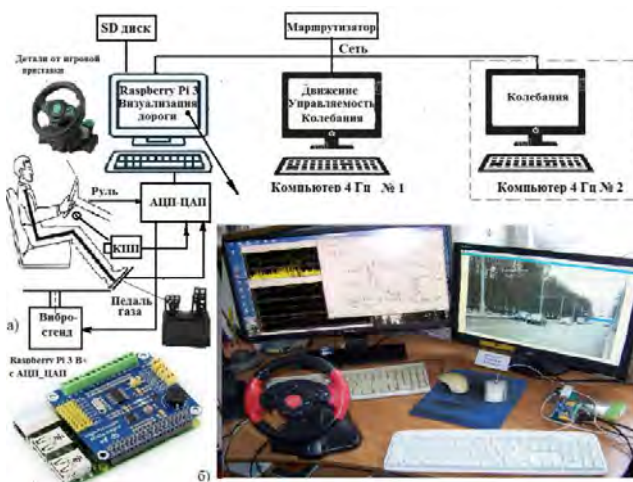


Рисунок 3.49. – Схема реализации имитационного моделирования автомобиля с использованием сети Samba с визуализацией на Raspberry

Укажем, что использование S-Function Builder при шаге интегрирования 0,001 и 16-ти интеграторах позволяет проводить моделирование в реальном масштабе времени (в полосе частот 0–28 Гц) при использовании компьютеров с многоядерными процессорами с частотой около 4 ГГц.

Вопрос получения внешнего возмущения представляет собой отдельную задачу. Макро и микропрофиль дорог можно получить путем замеров ускорений подрессоренных и неподдресоренных масс автомобиля с последующим их преобразованием в Matlab/Simulink [28, 43, 62, 70, 83–85].

Помимо этого, возможны и другие решения. Так в ТВ программе National Geographic промелькнул сюжет об возможном использовании для управления беспилотными автомобилями данных сканирования дорожного полотна на его глубину и создании их карт по типу навигационных.

3.13.5 Визуализация дорожной обстановки с использованием Raspberry

Зарубежные фирмы для визуализации дорожной обстановки используют анимацию, но она требует больших затрат и привлечение сторонних фирм. В рамках выполнения работы предложено применение нового более простого метода визуализации на основе видеосъемки дорожной обстановки на видеорегистратор/смартфон при фиксированной скорости движения автомобиля ($v = 20$ м/с) синхронно с записью продольного профиля и микропрофиля дороги, осуществленного с помощью Raspberry [33, 34, 35] и последующем ее воспроизведения с помощью разработанной автором программы на C/C++ с использованием библиотеки OpenCV с управляемой частотой просмотра синхронизированной с процессом моделирования в Simulink (приложение Е). Примененный в работе подход и программные решения, включая синхронизацию изображения являются новыми, обладающим научной новизной.

Предлагаемый подход по реализации имитационного моделирования заключается в использовании разработанных программ на C/C++, реализованных в S-Function Builder и Raspberry Pi 3. Данные программы обеспечивают задание внешнего воздействия дороги (продольного профиля и микропрофиля дороги), логику переключения передач КПП и обмен информацией между компьютерами [33–35]. Обмен информации между компьютерами без передачи данных на стенд осуществляется по сети с помощью файлового сервера Samba через файлы обмена (рисунок 3.50) на общем SD диске Raspberry с определенной периодичностью на основе сравнения счетчиков циклов (приложение Д).

На Raspberry возложена задача визуализации дорожной обстановки на основе ее видеозаписи, прием данных положений педалей газа, тормоза, рулевого колеса и передачи КПП через АЦП_ЦАП и передача информации в виде бинарных файлов File1, File2 (приложение Ж). File1 формируется программой Raspberry, а File2 – S-Function Builder Simulink. В качестве маршрутизатора использовался телефонный HUAWEI HG8245H (1 Гбит/с).

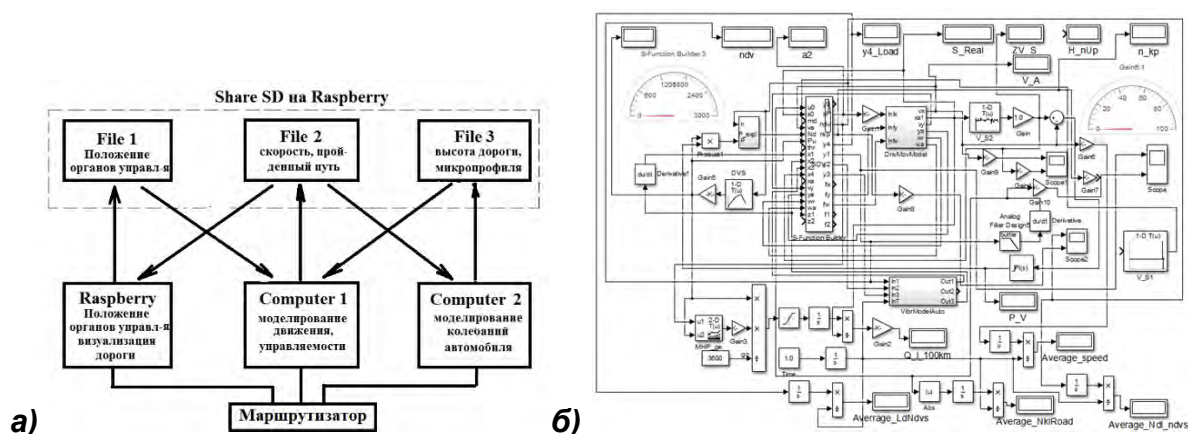


Рисунок 3.50. – Схема обмена информацией (а) и блок-схема имитационного моделирования движения, колебаний и управляемости автомобиля (б)

Для реализации имитационное моделирование необходимо вначале установить программное обеспечение на Raspberry Pi 3/4, включающего:

- установка ОС Raspbian (с ядром Linux) с помощью NOOBS, версии 4.0;
- установка файлового сервера Samba;
- установка компилятора C/C++ GC-10.0 и новее;
- установка библиотеки OpenCV 4.0.1;
- установка библиотеки bsm2835-1.55 (обеспечивающих работу АЦП, ЦАП).

И далее создание на их основе в Raspberry специальной программы по управлению моделированием и визуализацией с помощью компилятора C/C++.

Все эти этапы подробно описаны в приложениях Д, Е, Г.

3.13.5. Реализация обмена информации на Raspberry

Для обмена информации при моделировании без задействования стенов использовался файловый сервер Samba, обеспечивающий время обмена информацией 0,2 мс. Он создает общую папку share folder **share**, доступную, как компьютеру PC, так и Raspberry. В эту папку в виде двух бинарных файлов 8 и 16 байт записывалась и считывалась информация с Simulink и Raspberry с периодичностью 0,2 с (5 Гц), для чего использовались функции fread, fwrite. Эти параметры вполне достаточны для управления и не тормозят процесс моделирования в Simulink. Особенностью программы визуализации [34, 100] является использование функции cap.read(frame) для прокрутки кадров вперед для ускорения просмотра видео и задержка вывода видео установкой параметра stp=0, останавливающего процесс. Параметр

waitKey(frshv) используется только для возможной корректуры скорости просмотра. Синхронизация изображения и пройденного пути осуществляется на основе передаваемой информации из Simulink в файле pda.bt и сравнения их счетчиков кадров. Число кадров, соответствующих пройденному пути $chr=K \cdot 30$, где $K=30 \cdot t/l=1,6346$, t и l – соответственно пересчетный коэффициент, время прохождения/моделирования участка в с, длиной l , м, 30 – частота записи video.

В приложении Г приведена основная часть исходного текста разработанной программы shv.cpp на C/C++ реализации Video на Raspberry с управляемой частотой просмотра, предназначенного для использования в симуляторе автомобиля. В дальнейшем при использовании основного компьютера с программой на MS Visual Studio 2022 применен другой алгоритм.

3.13.5 Передача данных с Simulink на стенды по CAN BUS

3.13.5.1 Для полноценной реализации системы имитационного моделирования необходимо еще передача данных с Simulink на объекты управления (стенды) [101].

Последнее является довольно сложным ввиду необходимости обеспечения одновременного моделирования автомобиля в реальном масштабе времени на Matlab/Simulink и передача с него данных на стенды. И если еще для передачи данных с органов управления достаточно воспользоваться сетью Samba при низкой частоте передачи (5 Гц), то для передачи данных с Simulink на стенды требуется частота обмена > 100 Гц. При файловом обмене с такой частоте не обеспечивается корректная передача данных из-за несинхронизации процессов записи и чтения.

Для передачи данных целесообразно использовать протокол CAN BUS (1024 кб/с), который сейчас широко применяется в роботах, авиа и автомобилестроении в системах управления для обмена данными между микроконтроллерами, благодаря своей простоте, надежности и скорости. В нем используется витая пара в качестве кабеля (рисунок 3.51).

Передача данных осуществляется с помощью кадра (frame), содержащего как служебную, так и передаваемую информацию. Поле данных составляет 8 символьных байт.



Рисунок 3.51. – Предлагаемая общая схема передачи данных с Simulink на стенд

Для передачи данных по CAN BUS необходимо наличие аппаратных средств – адаптеров, которые связывают компьютерную технику с устройствами управления либо наличие их в устройствах управления. Сейчас большинство современных электроактуаторов и гидропульсаторов оборудованы интерфейсами CAN BUS и производители предлагают инструкции и средства создания программ на языке C/C++.

Помимо этого, требуется специальное программное обеспечение (ПО) для адаптеров, которое отличается у разных производителей ввиду различной реализации самого адаптера. Производители, как правило, предлагают ПО в виде CAN_API.dll, созданного в MicroSoft Visual Studio (MS VS) либо Linux без предоставления исходных текстов.

Тестированием установлено, что ПО CAN_API.dll, откомпилированное в MS VS 2017-2019, не работает в TDM-GCC-64 Matlab/Simulink из-за разного подхода в именах функций dll по стандарту C++11. Чтобы устранить эту проблему требуется перекомпилирование dll в среде TDM-GCC-64 под Windows, которое может выполнить только разработчик dll.

Сейчас в версии Matlab/Simulink R2019b появились компоненты, позволяющие создать коммуникацию по CAN BUS через Virtual Channel, Vehicle Network Toolbox. Они требуют отдельной покупки и использования недешевых адаптеров определенных фирм, таких как, Vector, Kvaser, PEAK-System и других и их программного обеспечения. К сожалению, нет публикаций по практической реализации данного решения для стендов с большим количеством цилиндров и оценок насколько это эффективно.

3.13.6. Выбор адаптера и средств разработки

Определяющим фактором для выбора адаптера является возможность реализации в нем частоты 100 Гц на каждый цилиндр из 6-ти, используемых в платформе 6-DOF. В этом случае можно обеспечить погрешность 1 % на верхней частоте. Исходя из этого был выбран адаптер фирмы Titan TITAN ELECTRONICS INC., SN #T16820100, Тайвань, являющийся наиболее оптимальным по соотношению цена-качество [102]. Кроме того, эта фирма пошла мне навстречу, бесплатно предоставила два адаптера и перекомпилировала ПО под TDM-GC-64 и помогла на начальном этапе с освоением CAN BUS. Примечание. Тогда еще не было MS VS R2022.

Для работы CAN BUS вначале необходимо установить драйвер USB-CAN CDM21228_Setup.exe. Определить какой подключен COM PORT (например, COM3). Затем запустить в командной строке CAN_BAUDRATE_SET.exe COM3. Должно быть получено сообщение

Searching...

Find COM PORT: COM3

Setting every baud rate to 3Mbit...

Set baud rate Success

свидетельствующее об успешной установке. После чего требуется отключить USB кабель от компьютера и через 5 с вновь его подсоединить либо выключить и включить компьютер.

Примечание. В ряде случаев чтобы установить CAN_BAUDRATE_SET.exe на Windows 10 64 bit необходимо задать Свойства этого файла в режиме Совместимость, с отмеченными галочками, как показано на рисунке 3.52. После чего выполнить вышеприведенную команду.

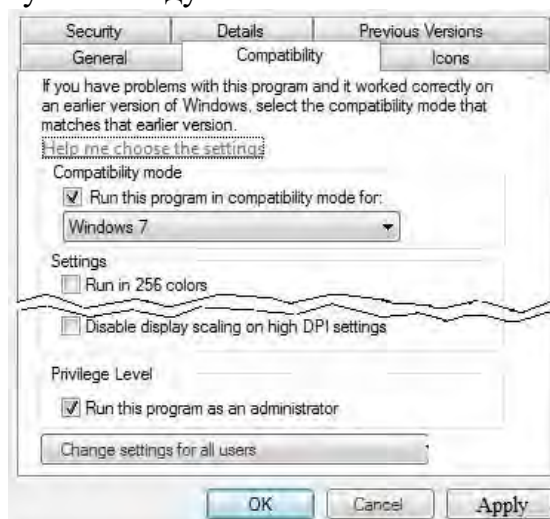


Рисунок 3.52. – Установка режима совместимости для CAN_BAUDRATE_SET.exe

Для моделирования в различные периоды использовались пакеты Matlab/Simulink R2015–R2024a (ode5 Dormand-Prince, $t=0,001$ с).

Сейчас для разработки программы в S-Function Builder Matlab/Simulink R2024a необходимо использовать компилятор MS Visual C/C++ 2022, v27.12, ранее применялся TDM-GC-64 (tdm64-gcc-10.3.0.exe). Установка и настройка TDM-GCC-64 описана в работах [14, 17]. Для него в Windows 10 требуется еще установка пути на C:\TDM-GCC-64\bin через кнопку Set Path на панели Matlab.

Моделирование проводилось с использованием реального возмущения дороги [17, 100, 103] в виде массива объемом 96000 записей типа float в электронной памяти и выполнением логических операций по переключению КПП, расчету тяговой динамики автомобиля в программе на C в S-Function Builder с использованием совершенных моделей, обеспечивающих погрешность 10–15 % с учетом спектра [15, 28, 30, 31]. Кроме этого в отдельных блоках Simulink моделировались вибрации, управляемость автомобиля. Визуализация на этом этапе осуществлялась с помощью мини-компьютера Raspberry Pi 3B+. Выбор такого подхода обуславливался необходимостью обеспечения максимальной скорости моделирования в реальном масштабе времени.

3.13.7 Реализация обмена информации

Реализация моделирования с передачей данных по CAN BUS осуществлялась с помощью структурной схемы и блок-схемы (рисунки 3.53, 3.54), в которой задействовано 18 интеграторов [100].

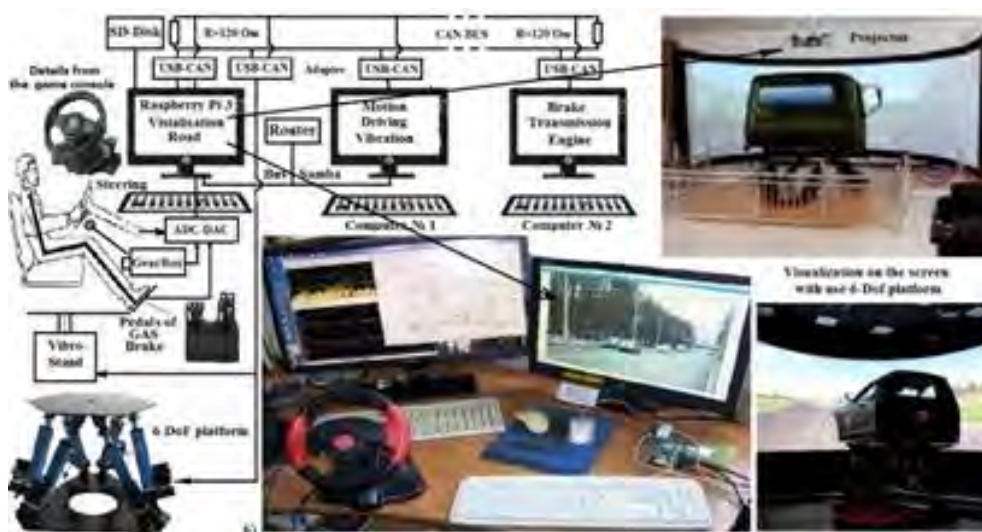


Рисунок 3.53. – Схема имитационного моделирования с CAN BUS с реализацией на 6-DoF вибростенде и макетном настольном образце

Для моделирования первоначально использовался компьютер (AMD Risen 5 2600, 6-ти ядерный 3,4/4 ГГц, RAM 16 Гб DDR-4 2666 МГц, накопитель M2. SD 2500 Мб/с, MB ASUS PRIME B450 PLUS, $t=0,0001$ с) с передачей данных в асинхронном режиме, а также детали от игровой приставки.



Рисунок 3.54. – Процесс и блок-схема моделирования движения, колебаний и управляемости автомобиля

Прием сообщений осуществлялся на двухядерном компьютере AMD 2,5 ГГц, RAM = 8 Гб, DDR-2 400 МГц, MB ASUS M2-VM, который имитировал исполнительное приемное устройство (стенда). Для моделирования автомобиля и передачи данных с Simulink на стенды разработана программа на языке C для S-Function Builder, общий объем которой составил 1372 строк исходного текста. Из них 850 строк служебные.

Первоначальная отладка CAN BUS проводилась на упрощенной программе **wrdc.c**, приведенной в приложении 3, которая поможет Вам понять механизм работы CAN BUS и создать свою программу. Ее функция `void srd(struct ds *ps, struct dr *pr)` использована в общей программе для S-Function Builder.

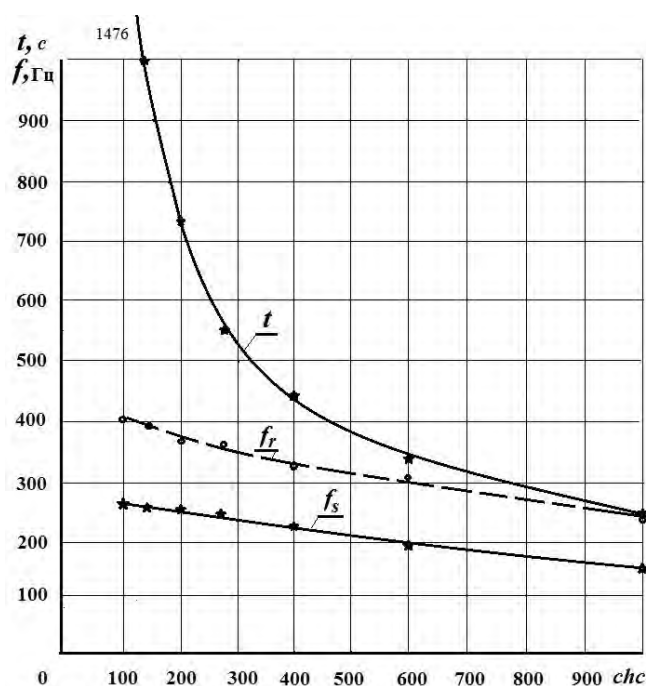
Используемые в программе функции и настройка адаптера также подробно описаны и в [102]. В каталог с программой **wrdc.c** должны быть установлены файлы **CAN_API.dll** и **CAN_API.h**. Компиляция программы необходимо производить в MS VS 2022 v.17.7 и новее. Также можно откомпилировать программу в TDM-GCC-64 введя команду

gcc wrdc.c

В результате получится исполняемый файл **a.exe**, который использовался для приема сообщений по CAN BUS. Особенностью примененного подхода является то, что вначале информация для 6-ти актуаторов заносится в структуру данных `struct ds sf`. Далее в функции `void srd()` реализовано сжатие данных: занесение в поле 8 байт `SendMSG.Data[0]` двух значений `float` данных побайтно вместо одного (фрагмент выделен желтым цветом). Фактически это передача данных блоками на два актуатора сразу, имеющих одинаковый идентификатор. А в программе контроллера актуатора производится программное их разделение. Это позволяет повысить производительность обмена данными в два раза (тем самым до 6 Мб/с).

3.13.8. Результаты моделирования по обмену информацией

Передача данных осуществлялась с периодичностью по счетчику циклов **chc**, который обнулялся при достижении максимальной величины. Счетчик циклов **chc** определяет количество тактов моделирования, после которого происходит передача данных на 6 электроактуаторов по CAN BUS. Это позволяет сократить затраты времени на передачу информации. Квантование в 100 Гц вполне достаточно для воспроизведения стендом верхних частот 10 Гц с погрешностью 1 %. Результаты моделирования приведены на рисунке 3.55.



t – время моделирования, f_s – частота посылки сообщений, f_r – частота приема сообщений

Рисунок 3.55. – Влияние параметра квантования на частоту посылки и приема сообщений и время моделирования

Базовое время моделирования без CAN BUS на 6-ти ядерном компьютере составляет 105 с при реальном процессе 470 с. Как видно из графика использование CAN BUS значительно увеличивает время моделирования (в 4,5 раза при $chc = 350$) т.е. замедляет его. Характер изменения частоты посылки и приема сообщений имеет вид небольшой нелинейности.

При уменьшении chc резко увеличивается время моделирования t и оно определяет выбор chc . При параметре квантования $chc=350$ достигается режим реального времени моделирования, частота передачи данных при этом составляет 230 Гц.

Режим реального времени может также достигнут путем подбора шага интегрирования, частоты процессора в UEFI BIOS материнской платы, подключением других подсистем, функций, отказом от сжатия данных; использованием более сложной пространственной модели автомобиля, применением синхронного режима передачи.

При увеличении параметра квантования с 350 до 1000 частота передачи данных уменьшается с 230 до 150 Гц. Но этого вполне достаточно для управления стендами (>100 Гц). Имеющийся запас в дальнейшем позволяет реализовать моделирование при усложнении модели автомобиля.

Следует отметить, что скорость моделирования на старом 2-х ядерном компьютере 2,5 ГГц с CAN BUS получилась 8,7 раза меньше чем на современном (3,4 ГГц) и свидетельствует о необходимости использования быстросействующих компьютеров.

Частота чтения данных без Simulink получается в 1,5 раза быстрее чем их запись за счет исключения процесса моделирования.

Результаты моделирования свидетельствуют о возможности разнесение вычислений подсистем между компьютерами при использовании CAN BUS для решения сложных задач и целесообразности полного перехода на CAN BUS.

К сожалению, автопроизводители СНГ (РФ, РБ) пока не проявляют интереса к этому из-за сложной экономической ситуации и своей недалёковидности. Из-за чего не удалось реализовать вариант вибростенда кабины и широкого экрана с поворачивающимся проектором, как показано на рисунок 7.21 внизу слева. А ведь при развитии методов имитационного моделирования в сочетании со стендовыми испытаниями автомобиля на беговых барабанах (по предложенной мной оригинальной схеме) можно исследовать каждый элемент конструкции и оптимизировать его в стендовых условиях. Важную роль здесь играет использование CAN BUS для обмена информацией. Но в СНГ предпочитают работать по старинке, не проводя

таких работ, как и не читаются курсы CAN BUS студентам. Все это приводит к созданию неконкурентной продукции.

Выводы по главе 3

1. Созданы колебательные модели автомобиля и автопоезда с учетом основной подвески, включая пневмогидравлическую, подрессоривания кабины, сиденья, биомеханической модели водителя, водителя на спальном месте, основных типов гидравлических амортизаторов: двухтрубного и газонаполненного, учитывающие нормальный и аномальный режим амортизатора [44, 49], на основе перетекания жидкостей в его полостях, описываемая дифференциальными уравнениями и объясняющая ухудшение виброзащитных свойств сиденья с газонаполненным амортизатором, дизель-генераторной установки, блок-схемы их реализации, показавшие хорошую сходимость в октавных полосах частот 1, 2, 4, 8, 16 Гц (расхождение с экспериментом 5–12 %), что подтверждает корректность моделей.

2. Предложено рассматривать характеристику упругого элемента как сумму характеристик упругого элемента и трения, описываемого интегралом по относительной скорости, обеспечивающее существенно лучшее совпадение с экспериментом (5–12% по ускорениям в октавных полосах частот).

3. Разработаны математические модели устойчивости, опрокидывания, дано их аналитическое описание, а также управляемости автомобиля с учетом новой модели увода шины, которую следует рассматривать как зависимость от скорости угла в виде интеграла от скорости, а не самого угла, как делают большинство исследователей.

4. Разработан метод совместного моделирования движения и колебаний автомобиля с учетом переменного характера движения, характеристик двигателя, топливной экономичности на основе использования полученного экспериментально объединенного массива макро и микропрофиля дороги (96000 точек), заносимого в оперативную память компьютера через модуль S-Function Builder Simulink и реализацию на его основе колебаний и движения автомобиля, логики переключения КПП на основе баланса сил, ускорения ТС, точек снижения скорости, учета дорожных факторов, передачу звука, информации на стенды, что позволяет проводить моделирование в реальном масштабе времени на больших участках дорог, оценку тягово-скоростных свойств, экономичности, плавности хода машины, расчеты нагрузок на узлы, долговечности и осуществлять выбор оптимальных параметров на стадии проектирования, чего невозможно сделать средствами Matlab;

ГЛАВА 4

ЭКСПЕРИМЕНТАЛЬНЫЕ ИССЛЕДОВАНИЯ

4.1 Применение микропроцессорных и мини-компьютерных систем

4.1.1 Доводка любой конструкции (транспортного средства, самолета, машины, станка) требует проведение экспериментальных исследований и испытаний. Для регистрации параметров при исследованиях и испытаний обычно используются тензоусилительные измерительные станции (например, ZET 017-T8), информация с которых может записываться на флешпамять или ПЭВМ. Последние могут работать, как автономно, так и в паре с ПЭВМ. К ним можно подключить датчики ускорений, тензодатчики и индуктивные датчики перемещений и др. Это профессиональная промышленная аппаратура высокой надежности и точности (погрешность 0,1%). Число каналов регистрации обычно 8–24. Стоимость станции на 8 каналов без датчиков 234 тыс. рос. руб. (в ценах 2018 г.). Однако для обработки данных необходимо приобретать дополнительные устройства, например, анализатор спектра ZET 017-U8 стоимостью 420 тыс. рос. руб. В условиях экономического кризиса многие организации не могут позволить себе приобрести их. В результате мы не проводим необходимых испытаний и имеем неконкурентную продукцию.

В этой ситуации более целесообразным является использование дешевых микроконтроллеров Arduino и мини-компьютеров Raspberry Pi 3/4 [14–72], которые во многих областях начали широко применяться. Стоимость их системных плат составляет соответственно 30 и 36 \$. К ним имеется около полусотни датчиков с погрешностью 0,2–1 % и модулей сопряжения. На базе их можно создать измерительно-регистрирующую аппаратуру, вполне подходящую для проведения многих исследований и испытаний.

Arduino – является однозадачным 1–4-х ядерным микроконтроллером на основе 8-ми битных команд с относительно небольшим объемом оперативной памяти (оперативной памяти SRAM 96 Кбайт, объем флеш-памяти программ - 512 КБ), быстродействием 16–84 МГц, отсутствием мультимедийных и сетевых возможностей. Он характеризуется низким энергопотреблением и достаточно высокой скоростью реакции на управление объектами. Однако он не являющийся полноценным компьютером. Для создания программ Arduino требуется компьютер, или ноутбук с USB-портом, что существенно увеличивает стартовый бюджет. Для программирования Arduino используется С-подобный язык. Arduino предназначен для управления несложными роботами, БЛА, квадрокоптерами и различными автоматизированными системами. Но управлять роботом через интернет и обрабатывать маршрут Arduino не может.

Raspberry Pi 3/4 – представляет более полноценный 4-ядерный одноплатный компьютер на основе 32-ти битной операционной системы Rasbian (на ядре Linux) с 1–4 Гб оперативной памяти с быстродействием (600–1400 МГц) и возможностью подключения через USB внешних накопителей, обладающий продвинутыми мультимедийными (Open GL, HD-Video) и коммуникационными (WiFi, Bluetooth, Ethernet) возможностями. Параметры Raspberry Pi и аналогичных ему постоянно развиваются. Сейчас уже появились платы ASUS с ОП 4 Гб и быстродействием 1800 МГц. За рубежом Raspberry Pi широко используется для начального обучения программированию в школах и студенческой среде. К нему можно подключить обычный монитор с разрешением до 1920x1080 через VGA/HDMI-разъем, USB-клавиатуру и мышь либо использовать специальную мини клавиатуру с мышью (рисунок 4.1), а также радиомодуль или модуль GPS. В разработанной конструкции использована литиевая батарея 3800 мА·ч, позволяющая работать автономно 6 часов. Все это помещено в корпус Z-16/J (114x69x63 мм). В варианте Looptop использована батарея 10000 мА·ч ввиду большого потребления цветного дисплея и Raspberry Pi 3 B+.

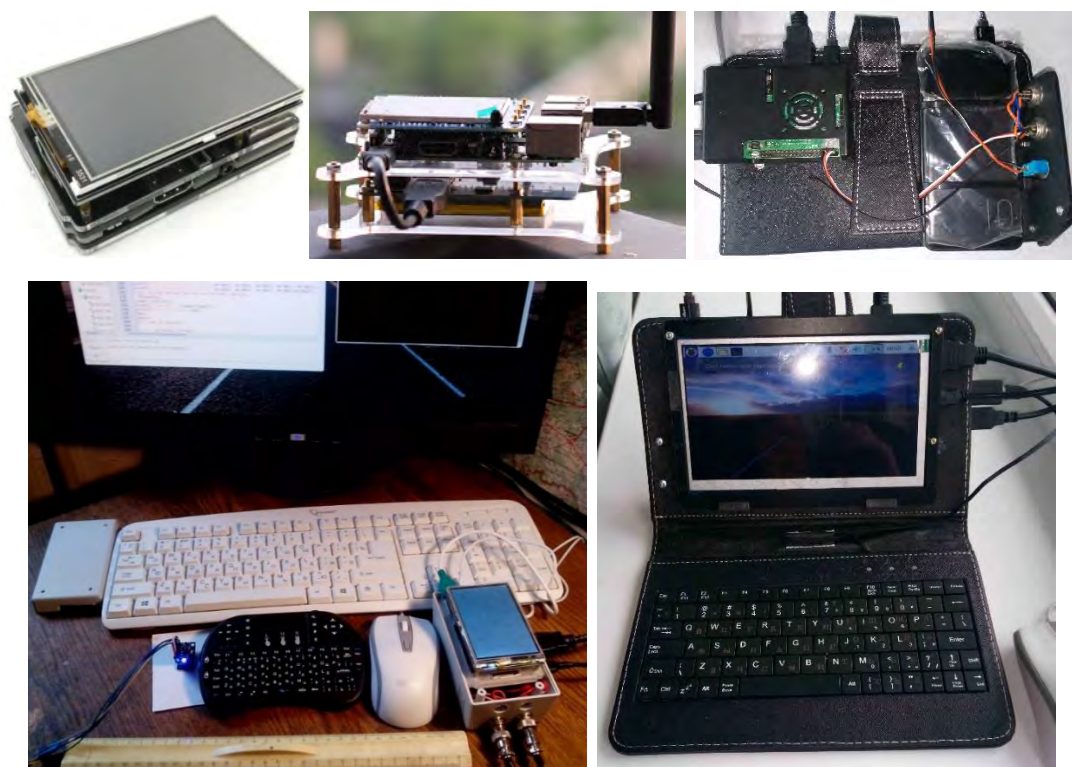


Рисунок 4.1– Общий вид собранных автором мини-компьютеров на базе Raspberry Pi 3

Raspberry Pi 3 может быть укомплектован LCD дисплеями 3,2; 3,5; 4; 5; 7; 10“, требующие установки специального драйвера. Стоимость показанного

на рисунке 4.1 такого компьютера на базе Raspberry Pi 3 с дисплеем 3,5”, батареей и мини-клавиатурой с мышью – 70 у.е. + 20 у.е. – корпус, разъемы, кабели, 2 датчика MPU-6050. На него мной установлены компилятор GCC-10.3 C/C++ стандарта 11, библиотеки OpenCV для C/C++, АЦП_ЦАП bcm2835. Raspberry Pi 3 располагает интерфейсом GPIO для управления всеми датчиками, изначально предназначенных для Arduino и может интегрироваться с последним для управления через него. Комплект из 37-ми датчиков стоит 1057 рос. руб.

Имеется плата расширения **High-Precision AD/DA Board** для мини-компьютера Raspberry Pi 3 на основе прецизионных АЦП и ЦАП стоимостью 30 \$. В основе платы прецизионный 8-канальный 24-битный АЦП ADS1256 с частотой дискретизации 30 ksps и прецизионный 2-канальный 16-битный ЦАП DAC8532. На плате установлены разъемы интерфейса аналоговых сигналов, выводы которых совместимы с интерфейсами сенсорных оценочных плат Waveshare. Таким образом, к плате можно легко подключить различные модули аналоговых датчиков. Входы/выходы для подключения к источникам аналоговых/цифровых сигналов реализованы на основе винтовых клеммников. Имеется более дешевый 4-х канальный ADS1115 (5 \$).

Особенностью подхода, примененного в Arduino и Raspberry Pi 3, является разделение функций между датчиком и самой системной платой. Так, например, датчик MPU-6050 содержит в себе регистратор ускорений, углов, тензоусилитель, фильтр и АЦП. А на платы Arduino и Raspberry Pi 3 возлагается функция приема цифровых данных с датчиков, их первичная обработка и регистрация на SD-карте либо выдачи сигнала на управление объектом. Такой же подход используется и в ультразвуковых или ИК-сонаров, которые могут применяться в качестве датчиков перемещений вместо индуктивных датчиков.

4.1.2 Реализации измерительной системы

Датчик MPU-6050 является ключевым элементом робототехнических систем. Проблемным местом является подключение двух датчиков MPU-6050 к Raspberry Pi 3 и одновременная работа с ними. На рисунке 4.2 показано использованные схемы подключения двух датчиков MPU-6050 и АЦП ADS1115.

В мини-компьютерах Raspberry Pi реализована унифицированная система управления датчиками на основе команд библиотеки **wiringPi/wiringPiI2c**, которая предусматривает следующие этапы:

1 этап – идентификация устройств и инициализация датчиков с помощью команды.

```
int wiringPiI2CSetup (int devId) ;
```

Например, fd = wiringPiI2CSetup (0x68);

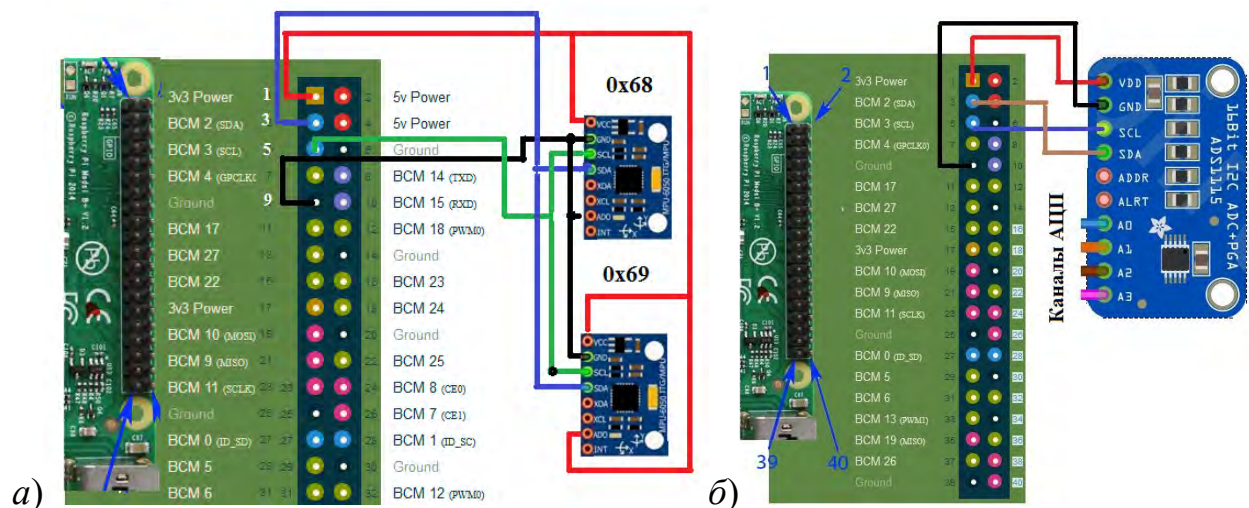


Рисунок 4.2 – Схема подключения двух датчиков MPU-6050 (а) и ADS1115 (б)

2 этап – активизация датчика (перевод его в активное состояние) путем записи в соответствующий регистр необходимого значения байта.

```
int wiringPiI2CWriteReg8 (int fd, int reg, int data);
```

У датчика MPU-6050 за это отвечает регистр 107 (0x6B).

```
wiringPiI2CWriteReg8 (fd, 0x6B, 0x00);//disable sleep mode.
```

И эта же команда используется для перевода в спящее состояние

```
wiringPiI2CWriteReg8 (fd, 0x6B, 0x06); //set sleep mode.
```

3 этап – чтение данных с регистров датчика, осуществляемое командой

```
int wiringPiI2CReadReg8 (int fd, int reg);
```

```
val = wiringPiI2CReadReg8(fd, addr);
```

и последующей операции сдвига полученных данных и математическое преобразование для получения углов.

Датчик MPU-6050 является программируемым для установки чувствительности и полосы фильтрации сигнала. Для этого используется команда

```
int wiringPiI2CWriteReg8 (int fd, int reg, int data);
```

Многие пользователи опасаются операционной системы Raspbian (на ядре Linux) и использование техники на ее основе, считая ее сложной.

Чтобы помочь в освоении этой системы и ее средств собрана, обобщена и проанализирована наиболее необходимая информация по датчику MPU-6050 с сайтов и форумах Arduino и Raspberry, текстов библиотек, приведенная ниже [14].

В таблице 4.1 приведена выжимка параметров регистров 26, 27, 28, 107, влияющих на чувствительность датчика MPU-6050 и полосу частот.

Таблица 4.1 – Параметры основных определяющих регистров

Addr (Hex)	Addr (Dec.)	Register Name	Serial I/F	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
1A	26	CONFIG		R/W		EXT_SYNC_SET[2:0]			DLPF_CFG[2:0]		
1B	27	GYRO_CONFIG	R/W	XG_ST	YG_ST	ZG_ST	FS_SEL[1:0]				
1C	28	ACCEL_CONFIG	R/W	XA_ST	YA_ST	ZA_ST	AFS_SEL[1:0]				
6B	107	PWR_MGMT_1	R/W	DEVICE _RESET	SLEEP	DEVICE _RESET CYCLE		TEMP_DIS	CLKSEL[2:0]		

В таблице 4.2 приведены значения байта регистра 27 (1B), 28 (1C), определяющие полосу фильтра низких частот для ускорений и углов.

Таблица 4.2 – Значения байта регистра 26 (1A), определяющие полосу фильтра низких частот

Полоса, Гц	256	188	98	42	20	10	5
Значение байта	0x00	0x01	0x02	0x03	0x04	0x05	0x06

В таблице 4.3 приведены значения байта регистра 27 (1B), 28 (1C) определяющие чувствительность датчика MPU-6050 (**int data**).

Следует отметить, что в Raspberry и Arduino предусмотрено только два адреса для датчика MPU-6050, которые устанавливаются путем соединения контакта ADO либо с VCC либо с Ground. В первом случае один датчик получает адрес 0x68, а второй 0x69. И таким образом разрешается проблема адресов. Для увеличения количества подключаемых датчиков MPU-6050 требуется подключение специальной платы расширения, но это приводит к снижению частоты их опроса.

Таблица 4.3 – Значения байта регистра 27 (1B), определяющие чувствительность датчика акселерометра MPU-6050 и 28 (1C) гироскопа 27 (1B)

Акселерометр, регистр 28 (1C)			Гироскоп, регистр 27 (1B)		
Диапазон ускорений	Обозначение параметра	Значение байта	Чувствительность гироскопа	Обозначение параметра	Значение байта
± 2 g	ACCEL_FS_2	0x00	± 250 °/s	GYRO_FS_250	0x00
± 4 g	ACCEL_FS_4	0x01	± 500 °/s	GYRO_FS_500	0x01
± 8 g	ACCEL_FS_8	0x02	± 1000 °/s	GYRO_FS_1000	0x02
± 16 g	ACCEL_FS_16	0x03	± 2000 °/s	GYRO_FS_2000	0x03

Особенностью работы с двумя датчиками MPU-6050 является необходимость программного отключения одного датчика, чтобы можно считывать данные с другого. Это осуществляется с помощью функции **wiringPiI2CWriteReg8(fd,0x6B,0x06)**; путем записи в регистр 107 (0x6B) значения 0x06 переводящий датчик в спящий режим. Активизации датчика осуществляется этой же функцией путем внесения в этот же регистр значения **0x00**. Необходимость поочередного выключения и включения датчиков существенно снижает быстродействие системы. Считывание данных осуществляется побайтно двойным запуском функции **wiringPiI2CReadReg8(fd,0x3B)** и последующим математическим преобразованием. Датчик MPU-6050 выдает 7 параметров: 3 параметра ускорений по осям *Z*, *X*, *Y* и 3 параметра поворота вокруг них, а также температуры. Следует отметить, что угол поворота вокруг оси *Z* сильно подвержен дрейфу и нестабилен. Поэтому в большинстве случаев можно отказаться от этого параметра и температуры, так как уменьшение количества считываемых параметров влияет на быстродействие системы. При необходимости регистрации поворота вокруг оси *Z* требуется использование датчика MPU-9150/9250 с магнитным компасом.

4.1.3 Оценка производительности Raspberry Pi 3 с датчиками MPU-6050

Для оценки производительности проводилось на программе для записи ускорений и углов наклона, приведенной в приложении Б.

Эта программа считывает данные с регистров, определяющих ускорения по осям Z , X , Y и углы поворота вокруг них в начале в структуру, а затем записывает на SD-карту.

Результаты исследования влияния количества задействованных датчиков, считываемых параметров, частоты шины I2C сведены в таблицу 4.4 [14].

Таблица 4.4 – Влияние количества задействованных датчиков, считываемых параметров, частоты шины I2C на быстродействие системы

Кол-во датчиков	Кол-во считываем. параметров	Частота шины, kps/s	Время цикла, мс при 1000 записях	Частота датчика, Гц
1	2	3	4	5
1	9	100000	24952	400,8
1	6	200000	37895	263
1		400000		
2	14	200000	107448	93
2	12	200000	100104	99,8
2	12	200000	93605	107
2	10	200000	79080	126,5
2	8	200000	65459	152,3
2	12	200000	9404	94
2	10	200000	7313	139
2	8	200000	5929	168
2	6	200000	5239	198
2	4	200000	3842	260
2	2	200000	3842	260
2	1	200000	2458	408
1	6	200000	4195	238
1	5	200000	3493	279
1	4	200000	2813	357
1	3	200000	2102	475
1	2	200000	1402	710
1	1	200000	716	1394

Как видно и таблицы существенное влияние на быстродействие оказывают количество задействованных датчиков и фиксируемых параметров. Увеличение частоты шины свыше 200000 kpsb/s приводит к остановке процесса. Уменьшение частоты шины до 100000, 50000 kpsb/s приводит к пропорциональному снижению частоты квантования датчиков MPU-6050. Оптимальная частота шины 200000 kpsb/s. Максимальная достигаемая частота квантования на одном датчике и одном параметре – 1394 Гц, а при двух датчиках и одном параметре – 408 Гц.

Исследование показало, что изменение частоты процессора с 700 до 1000 МГц не влияет на частоту считывания информации с датчиков MPU-6050 и нет необходимости ее увеличения так как последнее приводит к дополнительному энергопотреблению и требует дополнительного охлаждения. Следует также отметить, что часов в платах Raspberry и Arduino нет. Можно задействовать отдельный модуль часов I2C RTC DS1307. Но он использует адрес 0x68 и часто из-за этого приходится от него отказываться. Для определения времени процесса необходимо брать разность millis().

Сравнение с ранее полученными результатами на Arduino Due (2 датчика, 6 параметров, 153 Гц) показало, что Raspberry Pi 3 обеспечивает более высокую частоту квантования 198 Гц и на нем нет провалов в записываемом блоке данных. Последнее на Arduino Due связано с низкой скоростью записи на SD-карту из-за использования неэффективной функции mF.write(). Если не записывать данные, то быстродействия Arduino Due (84 МГц) вполне достаточно для считывания данных с двух датчиков и управления несложными робототехническими устройствами (мобильными роботами, квадрокоптерами).

При необходимости более сложной математической обработки и регистрации более предпочтительным является использование Raspberry Pi 3/4, тем более что его 32-тибитная операционная система позволяет хранить в памяти гораздо больший объем чем 8-ми битная. Например, задание сложного маршрута движения робота, БЛА, квадрокоптера, перевод на автономный режим управления при потере радиоуправления либо осуществлении спектрального анализа. Следует также отметить, что скорость записи на SD-карту с помощью функции fwrite() достаточно высока и не тормозит работу системы, на нее тратится менее 0,1–0,2 мс.

Преимуществом разработанной программы и системы Raspberry Pi 3 является то, что ее можно менять и откомпилировать в полевых условиях. Например, когда требуется обеспечить замеры с большей частотой квантования можно отключить (закомментировать) определенные параметры

и за счет этого повысить быстродействие. При выявлении ошибок можно оперативно ввести корректуру. Положительным фактором является возможность контролировать процесс регистрации с помощью дисплея, перезаписывать на SD-накопитель результаты исследования и их фрагменты.

Для тензометрирования нагрузок и напряжений в Raspberry Pi 3 может использоваться модуль HX711, позволяющий работать с тензодатчиками [14] (рисунок 4.3).

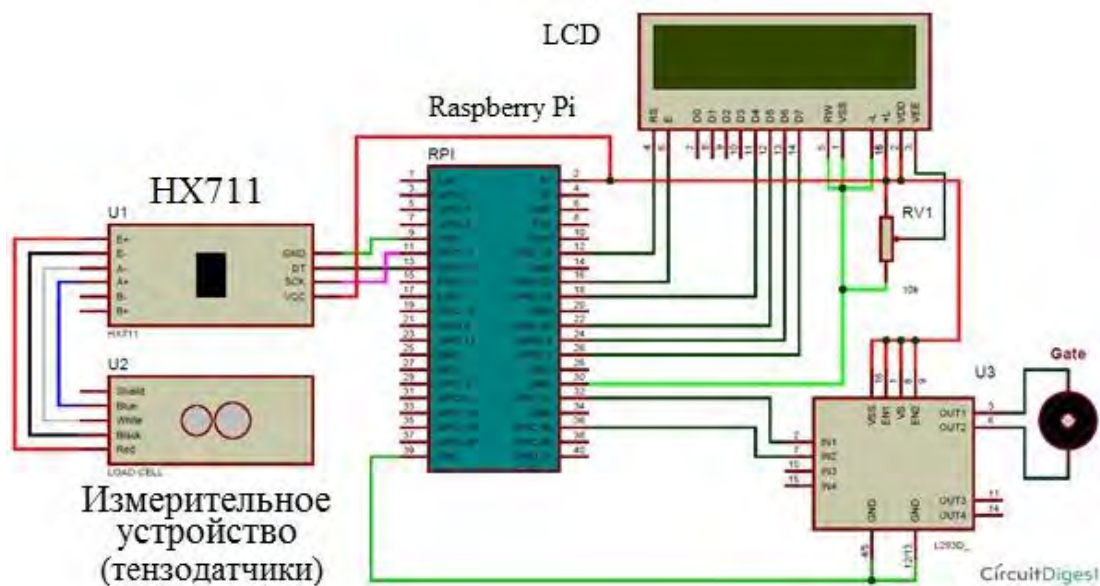


Рисунок 4.3 – Схема подключения модуля HX711 к Raspberry Pi 3

Следует отметить, что на погрешность измерений сказывается влияние многих факторов. Например, положение автомобиля захватившего на площадку взвешивания, что снижает реальную точность или вибрации дорожного полотна, порывы воздуха от движущегося рядом транспорта. Или фактор неперпендикулярности перемещения объекта исследования при использовании индуктивных датчиков и датчиков ускорений. Погрешность измерительной системы во многом определяется датчиками. Так реальная погрешность наклеянных тензодатчиков – 2–3 %. Поэтому реальная погрешность в итоге на различных системах и типах датчиков с разной точностью не будет сильно отличаться. Исходя из чего в большинстве практических случаях при исследовании мехатронных систем можно использовать более дешевые датчики и системные платы Arduino и Raspberry Pi 3.

Разработанная система и программа может найти применение в качестве регистрирующей аппаратуры фиксации вибраций, углов наклона.

Данный подход может использоваться при:

- оценке вибронагруженности рабочих мест операторов, вибраций различных машин, нагрузок с помощью модуля HX711 [14], позволяющего работать с тензодатчиками;

- вибрационной диагностики технических устройств, приборов, машин, мостов, железнодорожных путей и др;

- различных исследованиях вибронагруженности, нагруженности узлов, в дорожных и стендовых условиях:

- управления испытательными стендами

- для обеспечения дорожной безопасности автомобилей за счет применения ультразвуковых или ИК-сонаров, при парковке, слепых зон, контроля дистанций, информация с которых поступает на микроконтроллер и информирует водителя в виде звуковых сигналов или изображения;

- управлении позиционированием (подъема и горизонтирования) платформ машин и их модулей;

- оценке качества дорог (микропрофиля).

Использование Raspberry Pi 3/4 с датчиками от Arduino позволяет создать дешевую измерительную систему с приемлемой погрешностью для исследований мехатронных систем.

4.2 Результаты исследования дорог через ускорения поддрессоренных и неподдрессоренных масс по п.2.2

Исследования проводились на следующих дорогах возле Минска;

- участок 1 асфальтированного шоссе Вишневка – Вишневка – Новинки, характеризуемого субъективно как дорога очень хорошего качества, 4 полосы движения с небольшими уклонами, протяженностью 10 км;

- участок 2 бетонного шоссе МКАД-2, между Острошицким городком – перекресток М, недавно введенная в эксплуатацию, 4 полосы движения с небольшими уклонами;

- участок 3 местной дороги Белробот – Приморье – Семков городок – Чучаны – МКАД2 длиной 10 км. Данный маршрут характерен значительными перепадами высот, разным качеством покрытия (асфальт, бетонные плиты), проездом через 3 населенных пункта, наличием двух поворотов, две полосы движения;

- участок 4 местной дороги Саломоречье – Вишневка, дорога из бетонных плит, значительные перепады высот, 12 км;

- участок 5 местной дороги возле г.п. Салюты, грунтовое разбитое шоссе, 5 км.

В качестве основных критериев качества дорог оценивалась возможность использования среднеквадратических значений высоты, скорости и ускорения неровности дороги в третьоктавных, октавных и широких полосах частот (1,25–25 Гц) для микропрофиля и среднеквадратические значения скорости изменения высоты профиля дороги в третьоктавных полосах 0,03–0,63 Гц, характеризующее уклоны. Следует отметить, что значение скорости изменения высоты профиля дороги соответствует значению tg наклона дороги в процентах.

Сравнение скорости изменения наклона дороги, получаемого различными методами [55, 56], приведенные на рисунке 4.4 показали, что характер и значения спектра при различных методах близки и лежат в пределах погрешности этих методов. Поэтому исходя из близости кривых (расхождение от среднего 10–25 %) на практике в зависимости от особенностей дорог может использоваться один из этих методов (по h или по углу). Также могут использоваться данные и по радарным замерам.

Результаты исследования характеристик дорог, полученных по углу наклона для продольного профиля приведены в таблице 4.5, а микропрофиля в таблице 4.6 и показаны на рисунках 4.5–4.8. Для построения графиков использовался метод интерполяции Catmull-Rom, реализованный в S-Function Builder в MatLab/Simulink.

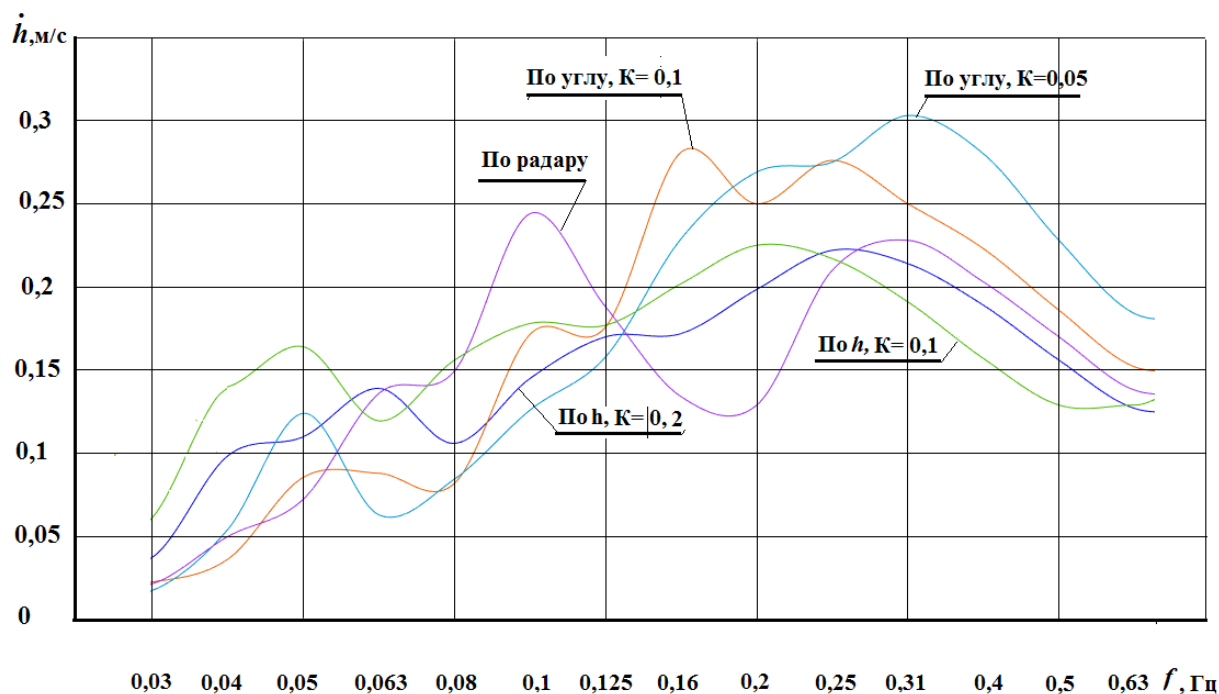


Рисунок 4.4 – Сравнение спектра скорости изменения высоты при различных методах получения продольного профиля дороги

Таблица 4.5. – Результаты замеров наклонов дорог

1/3 октавная полоса частот, Гц	Изменение среднеквадратических значений скорости изменения высоты продольного профиля от типа дороги в треть октавных полосах 0,03–0,63 Гц, м/с				
	Тип дороги				
	1	2	3	4	5
0,03	0,0116	0,0091	0,0227	0,0202	0,0361
0,04	0,0239	0,01	0,0364	0,0505	0,039
0,05	0,0363	0,0269	0,0856	0,065	0,0762
0,063	0,0619	0,0259	0,088	0,1784	0,0602
0,08	0,0726	0,0667	0,082	0,146	0,163
0,1	0,106	0,086	0,1715	0,24	0,174
0,125	0,137	0,105	0,176	0,275	0,214
0,16	0,187	0,138	0,281	0,406	0,261
0,2	0,216	0,136	0,25	0,397	0,311
0,25	0,235	0,132	0,276	0,38	0,37
0,31	0,247	0,125	0,25	0,378	0,3906
0,4	0,221	0,115	0,223	0,345	0,354
0,5	0,18	0,096	0,186	0,287	0,292
0,63	0,145	0,0786	0,152	0,234	0,235
Значения в октавных полосах частот					
0,05	0,08	0,0417	0,131	0,202	0,11
0,1	0,206	0,161	0,266	0,424	0,355
0,2	0,496	0,301	0,645	0,89	0,728
0,4	0,61	0,317	0,625	0,956	0,975
σ	0,817	0,468	0,9	1,388	1,273
max	3,49	3,675	5,58	6,624	4,73

Примечание. 1 – асфальтовое шоссе Вишневка–Новинки, 2 – бетонное шоссе МКАД-2, 3 – шоссе Белробот–Семков городок– перекресток МКАД-2, 4 – перекресток МКАД-2 – Вишневка, 5 – гравийная разбитая дорога

Таблица 4.6. – Значения среднеквадратических величин перемещений, скорости, ускорений микропрофиля дорог

Час- тота, Гц	Значения среднеквадратических величин перемещения, скорости, ускорений микропрофиля дорог														
	По перемещению, мм					По скорости см/с					По ускорению, м/с ²				
	Номер дороги														
	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5
1,25	0,316	0,206	1,247	1,187	1,793	0,251	0,168	0,987	0,941	1,421	0,036	0,045	0,097	0,094	0,134
1,6	0,259	0,24	1,123	1,348	1,928	0,276	0,252	1,139	1,367	1,953	0,057	0,074	0,148	0,169	0,232
2	0,297	0,29	1,159	1,203	1,835	0,383	0,383	1,472	1,537	2,344	0,091	0,120	0,238	0,245	0,354
2,5	0,277	0,242	0,976	1,241	1,638	0,453	0,417	1,576	1,995	2,658	0,147	0,184	0,342	0,393	0,522
3,15	0,246	0,28	1,028	1,31	1,894	0,522	0,636	2,11	2,67	3,896	0,2	0,303	0,564	0,652	0,927
4	0,24	0,419	0,958	1,31	2,149	0,675	1,147	2,592	3,45	5,691	0,359	0,534	0,907	1,08	1,678
5	0,232	0,297	1,072	1,343	2,472	0,877	1,134	3,705	4,49	8,23	0,473	0,793	1,553	1,74	2,96,
6,3	0,267	0,312	1,344	1,556	3,21	1,308	1,621	5,86	6,59	13,22	0,967	1,32	2,88	3,1	5,69
8	0,301	0,371	1,657	1,771	3,08	1,957	2,532	8,93	9,35	15	1,674	2,3	5,19	5,311	7,643
10	0,373	0,449	1,627	1,67	2,125	2,97	3,85	10,11	10,2	11,3	2,815	3,88	6,82	6,715	6,71
12,6	0,417,	0,556,	1,271	1,283	1,594	3,974	5,6	8,8	8,54	9,07	4,26	6,13	6,96	6,534	6,354
16	0,409	0,581	0,982	0,988	1,288	4,365	6,35	7,19	6,85	7,41	5,095	7,35	6,36	5,85	5,68
20	0,339	0,48	0,805	0,817	1,113	3,71	5,33	5,8	5,55	6,156	4,45	6,26	5,206	4,79	4,69
25	0,272	0,381	0,681	0,7	0,983	2,944	4,2	4,73	4,56	5,215	3,535	4,942	4,17	3,85	3,824

Окончание таблицы 4.6

Значения в октавных и широких полосах частот															
	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5
2	0,488	0,448	1,913	2,216	3,188	0,694	0,68	2,534	2,96	4,24	0,266	0,364	0,689	0,63	0,954
4	0,442	0,6	1,911	2,408	4,027	1,497	2,055	5,84	7	12,12	1,093	1,537	2,6	2,802	4,418
8	0,75	0,919	3,43	3,71	6,13	5,478	7,22	19,04	19,72	28,22	5,206	7,25	12,17	12,12	14,64
16	1,088	1,5	2,99	3,03	3,95	11,13	15,93	20,89	20,18	22	12,72	18,18	17,4	16,2	15,85
2-16	1,476	1,91	5,29	5,8	8,92	12,52	17,62	28,97	29,2	38,03	13,79	19,63	21,4	20,43	22,04
8-16	1,321	1,76	4,55	4,79	7,3	12,4	17,5	28,27	28,2	36,8	13,75	19,57	21,2	20,23	21,57
max	706,4	437,9	5023	627,1	784,3	146,9	101,4	630,4	579	715,5	63,8	46	68,7	56,8	77,83

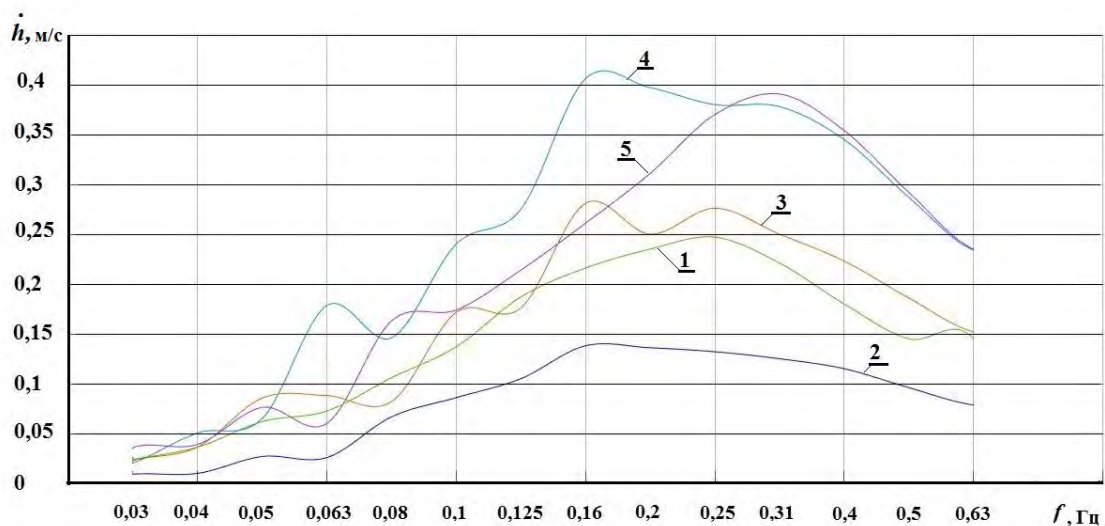


Рисунок 4.5. – Изменение среднеквадратических значений скорости изменения высоты продольного профиля (наклона дороги) от типа дороги в третьоктавных полосах 0,03–0,63 Гц

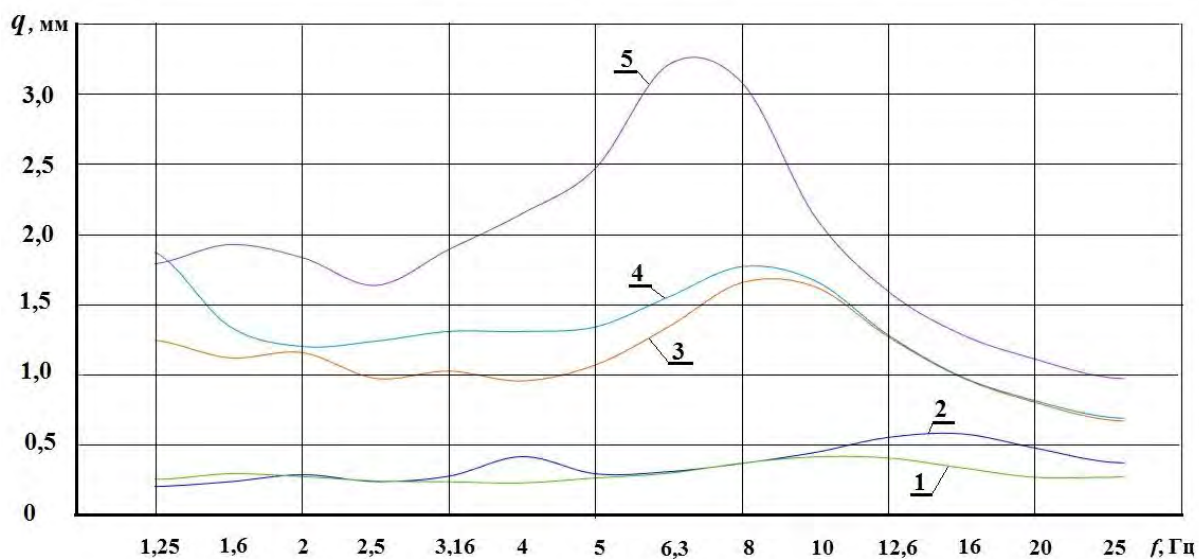


Рисунок 4.6. – Изменение среднеквадратических значений высоты микропрофиля от типа дороги в третьоктавной полосе 1,2–25 Гц

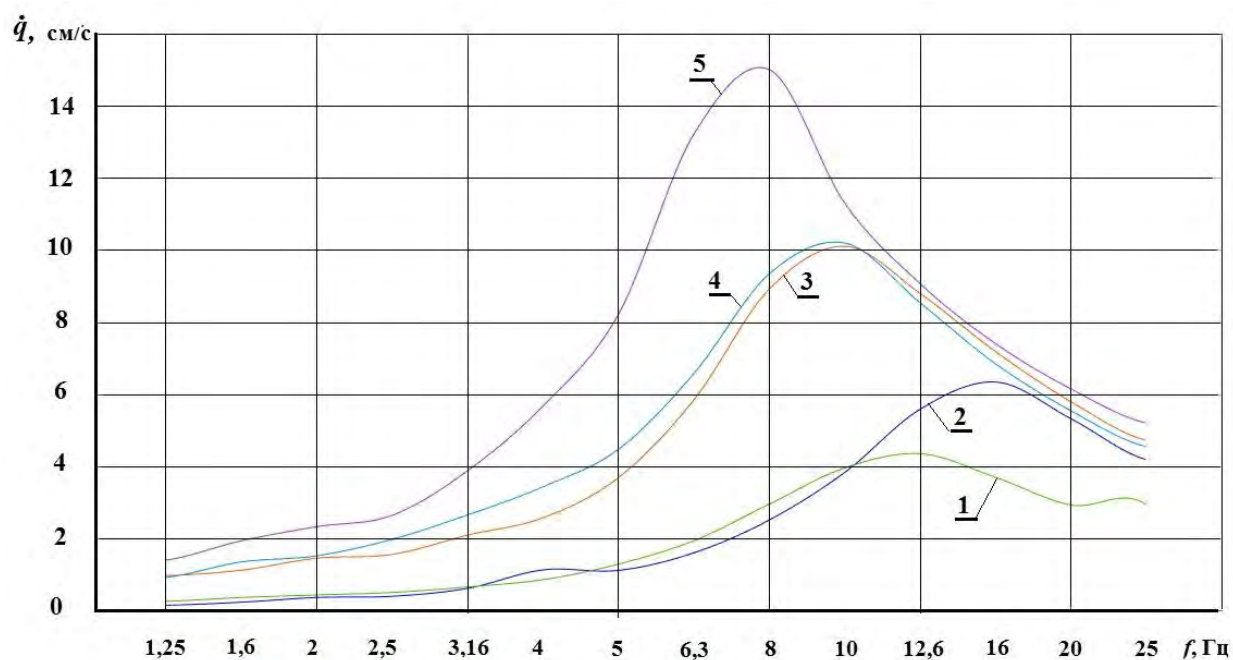


Рисунок 4.7. – Изменение среднеквадратической скорости микропрофиля дороги в третьоктавных полосах 1,2–25 Гц от типа дороги

Из рисунка 4.9 видно, что основной спектр скорости изменения высоты продольного профиля приходится на третьоктавы 8, 10, 12,6 Гц. Уровень среднеквадратических значений составляет 0,035 м/с при 10 Гц у дороги 1, 0,055 у дороги 1, 0,035 м/с при 10 Гц у дороги 2, 0,094 м/с при 12,6 Гц у дороги 3, 0,101 м/с при 10 Гц у дороги 4, 0,169 м/с при 16 Гц у дороги 5. И такой же характер изменения в целом по общей полосе 0,03–0,63 Гц. Приведенные значения в полосе 0,16–0,4 Гц могут характеризовать состояние продольного профиля дороги по уклонам: до 0,15 м/с – дороги с малыми уклонами, до 0,25 м/с – небольшими уклонами, до 0,3 м/с – со средними, свыше 0,4 – большими.

Из рисунков 4.7–4.10 следует, что для оценки микропрофиля дороги лучше подходит использование значений высоты и скорости ее изменения. Но в первом случае необходимо гарантировать фильтрацию более низких частот (менее 1 Гц), относящихся к продольному профилю и имеющих большие значения. Использование значений ускорений микропрофиля дороги неоднозначно в области больше 10 Гц. Из рисунков 4.11 видно, что дороги 3,4 по ускорению микронеровности практически совпадают, дороги 1, 2 имеют близкие параметры до частоты 11 Гц. У грунтовой дороги 5 спектр смещен влево, в диапазоне 10–25 Гц ее спектр совпадает с дорогой 4.

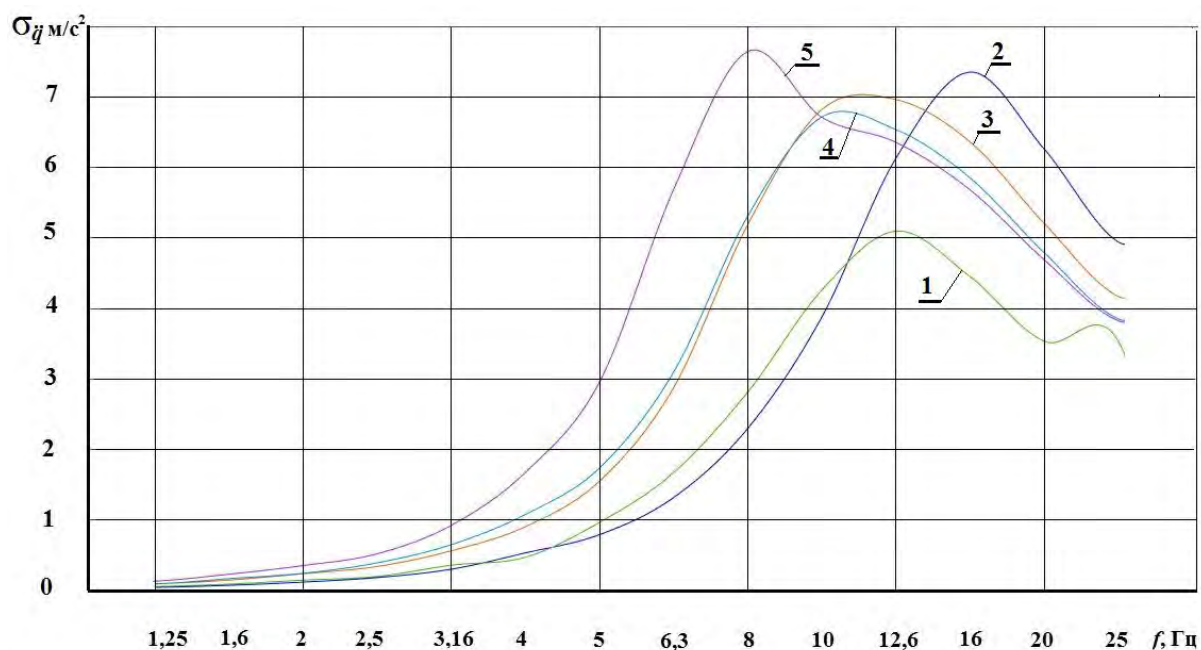


Рисунок 4.8. – Изменение среднеквадратических ускорений микропрофиля дороги от частоты и типа дороги

Учитывая необходимость оценки больших участков дорог, все же лучше использование значений скорости изменения высоты.

При обработке данных по методу IRI величина неровностей дорог 1,2,3,4, 5 составила соответственно 1,07; 0,9; 6,93; 8,96; 9,5 мм/м против 1,476; 1,91; 5,29; 5,8; 8,92 мм среднеквадратических значений высоты в полосе 2–22 Гц и 12,5; 17,6; 29; 29,2; 38 м/с при оценке по скорости изменения высоты в полосе 6,3–22 Гц. По методу IRI получается больший диапазон изменений значений ровности исследованных дорог (9 против 6-ти и 3). В большинстве работ по плавности хода в основном приводят данные по среднеквадратическим величинам высоты неровностей, но это данные замеров небольших (200–400 м) ровных участков дорог.

Приведенные на рисунке 4.5 графики могут характеризовать дороги по продольным уклонам, основной спектр которых приходится на третьоктавы 0,16–0,315 Гц. Наиболее низкие значения у дороги 2 (0,138 м/с при 0,16 Гц, 2-е место занимает дорога 1 (0,25 м/с при 0,16 Гц), 3-е место у дороги 3 (0,2 м/с при 0,16 Гц), 4-е место у дороги 5 (0,39 м/с при 0,315 Гц), 5-е место у дороги 4 (0,406 м/с при 0,16 Гц). На основании этих данных предлагается оценивать дороги по уклонам по скорости изменения высоты уклонов в полосе 0,16–0,4 Гц до 0,15 м/с дороги с малыми уклонами, до 0,25 м/с небольшими, до 0,3 м/с средние, свыше 0,4 м/с большими.

Качество микропрофиля дорог исходя из вышеизложенного целесообразно оценивать по спектру среднеквадратической скорости

микропрофиля в третьоктавных полосах 6,3–12,6 Гц: $\dot{q}$ до 4 см/с – дороги отличного качества, до 6 см/с – дороги хорошего качества, до 8 см/с – дороги удовлетворительного хорошего качества, свыше 10 см/с – плохие либо по более широкой полосе 6,3–22,4 Гц с распределением значений градаций 12,5; 18; 30; 38 см/с для этих же дорог.

4.3 Исследование движения на местных дорогах с помощью GPS

4.3.1 Ниже представлены графики изменения профилей и режима движения на местных дорогах (рисунок 4.9). На рисунке показано движение на участке круговая развязка Северного кладбища – н.п. Семково – МКАД-2.

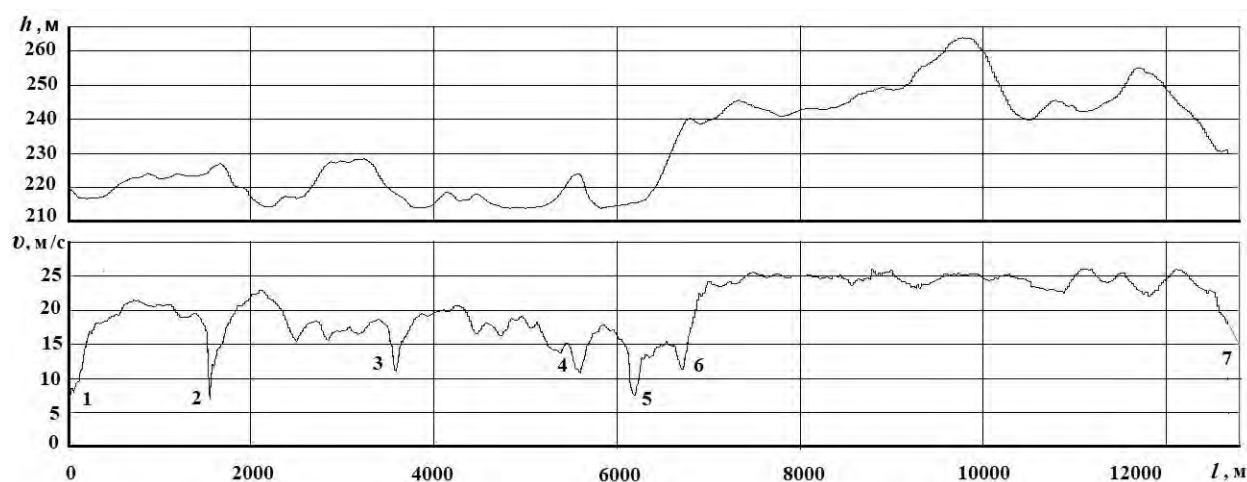


Рисунок 4.9. – Движение на участке развязка Северного кладбища – н.п.Семково – МКАД-2

Параметры прохождения участка следующие: $l = 12500$ м, $v_{ср}=53,85$ км/ч, $v_{max}= 85,9$ км/ч, $\angle_{ср}= 2,08$ % $\angle_{max}=5,6$ %.

В таблице 4.7 приведено описание прохождения точек изменения скорости.

Таблица 4.7. – Характеристика точек снижения скорости

Точка	Скорость	Наклон		Описание
1	7			Начало движения
2	8			Переезд препятствия
3	12			Поворот
4	11,5			Торможение перед н/п
5	8			Торможение из-за идущего медленно впереди автобуса
6	12			Торможение перед мостом Чучаны
7	0			Остановка

На участке 500 м водитель разгоняется до 17,5 м/с. После чего следует дальнейший разгон до $l=1200$ м. Далее следует спуск с разгоном до $v = 22,5$ м/с, после чего снижение скорости до $v = 17,5$ м/с, вызванное ограничением скорости 60 км/ч. Перед поворотом дороги на 120° (точка 3) водитель снижает скорость, начинает торможение. Далее следует разгон до $v = 17,5$ м/с с последующим снижением (волнообразным) до $v = 12$ м/с перед населенным пунктом (н/п) ($S=5000$ м). Участок 4, 5 проезд н/п. Далее разгон до $v = 17,5$ м/с с последующим снижением/торможение перед мостом, имеющим некоторый горб. Точка 6 – проезд н/п (ограничение скорости). После этого крутой подъем и потом разгон до скорости до $v = 25$ м/с. На участке 7000–12500 м водитель удерживает скорость порядка 90 км/ч, несмотря на имеющийся подъем 22 м.

Скорость движения на участке 0–1500 м $v = 20$ м/с, при спуске разгоняется до $v = 25$ м/с. Скорость движения на участке 2500–3500 м $v = 17$ м/с связана с ограничением скорости в населенном пункте.

4.3.2 Движение на участке МКАД-2–Саламоречье–Касынь–Вишневка

Результаты приведены на рисунках 4.10 – 4.12 и в таблице 4.8.

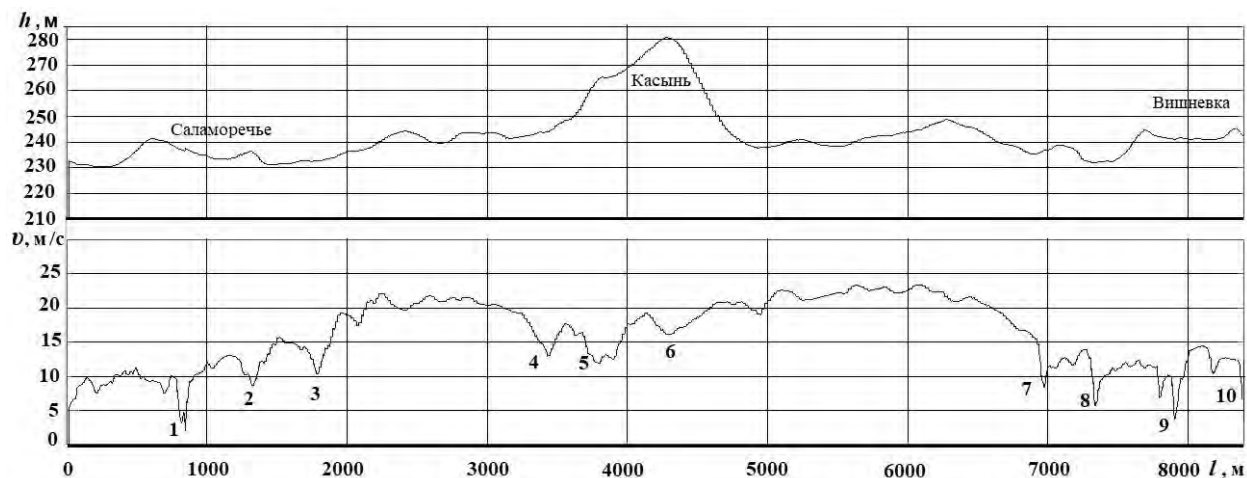


Рисунок 4.10. – Движение на участке МКАД-2–Саламоречье–Касынь–Вишневка

Таблица 4.8. – Характерные точки на маршруте

Точка	Скорость	Наклон		Описание
1	2	3	4	5
0	5			Начало движения по разбитой грунтовой 1 км
1	10			Движение по разбитой грунтовой дороге $v = 10$ м/с при сильной вибрации

Окончание таблицы 4.8

1	2	3	4	5
2	9–12			Переезд препятствия при въезде в н/п
3	9			Поворот дороги в Н/П на 90
4	13			Торможение перед поворотом
5	8			Торможение перед поворотом и подъемом.
6	12			Достижение верхней части подъема с дальнейшим спуском
7	8			Поворот дороги на 90
8				Поворот дороги на 90 в Вишневке
9	4			
10				Остановка

Движение по Р-58 на участке Вишневка– развязка Северного кладбища приведено на рисунке 4.11.

Данный участок характеризуется небольшим перепадом изменений высот профиля дороги: на участке 0–3000 м – 14 м, на участке 3000–6500 м – 25 м. Практически для всего участка дороги характерна постоянная скорость в районе $v = 25$ м/с за исключением начальной и конечной части. И лишь после 7500 м водитель начинает снижать скорость до $v = 20$ м/с. И далее до $v = 10$ м/с перед развязкой и последующей остановкой.

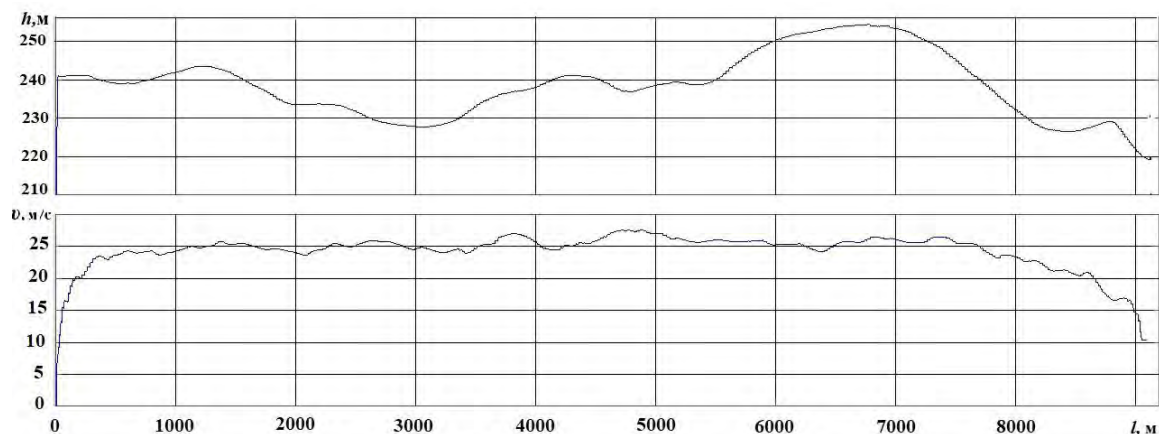


Рисунок 4.11. – Движение по Р-58 Вишневка– развязка Северного кладбища

4.4 Исследование движения в городе и магистральных трассах

Результаты испытаний приведены на рисунках 4.12–4.15.

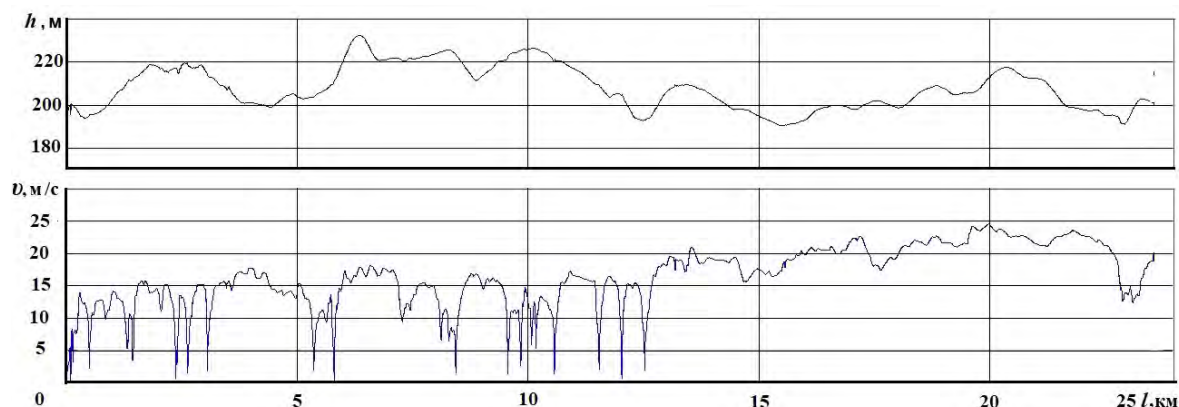


Рисунок 4.12. – Движение в городском режиме Минска: Комаровка – Запорожская площадь – пр-т Партизанский – н.п. Абчак

На рисунке 4.12 представлено движение в городском режиме Минска: Комаровка – Запорожская площадь – пр-т Партизанский – Абчак, для которого получены следующие данные для полного участка: $l = 23320$ м, $v_{\text{средн}} = 59.14$ км/ч, $v_{\text{max}} = 88,4$ км/ч (24.56 м/с), $\mu = 2.22\%$, $\mu_{\text{max}} = 3.98\%$, где l – длина участка, $v_{\text{средн}}$, v_{max} – средняя и максимальная скорость, μ , μ_{max} – среднее и максимальное значение наклона дороги в %.

На участке Комаровка – Запорожская площадь – до выезда на пр-т Партизанский наблюдаются 16 точек значительного снижения скорости, связанные с переездом искусственных препятствий, светофорами, движением по двух полосной дороге. Показатели движения на нем следующие: $l = 0 - 12500$ м, $v_{\text{средн}} = 47.7$ км/ч, $v_{\text{max}} = 65,06$ км/ч (18.07 м/с), $\mu = 3.161\%$, $\mu_{\text{max}} = 3.98\%$ в экстремальном режиме.

Для участка пр-т Партизанский – выезд из города – н.п. Абчак показатели движения на нем следующие: $l = 12,5-23,3$ км, $v_{\text{средн}} = 72.36$ км/ч, $v_{\text{max}} = 88,4$ км/ч (24.56 м/с), $\mu = 2.928\%$, $\mu_{\text{max}} = 3.88\%$.

Большое значение скорости на этой части участка связано с наличием шести полос и разрешенной максимальной скоростью $v_{\text{max}} = 70-80$ км/ч и тем обстоятельством, что имеющиеся на нем четыре светофора были включены на зеленый свет. В противном случае необходимо добавить еще 3 точки.

Как видно из рисунка 1 на участке $l = 0-12$ км водитель стремится поддерживать уровень скорости порядка 15 м/с за исключением участков с падением скорости, связанное с переездом искусственных препятствий, светофорами и последующим разгоном. А на участке $l = 12-23$ км водитель увеличивает скорость с 18 до 23 м/с.

Для дороги, представленной на рисунке 4.13, показатели движения следующие: $l = 107915$ м, $v_{\text{средн}} = 81.1$ км/ч, $v_{\text{max}} = 109$ км/ч (30.4 м/с), $\mu = 1.46\%$, $\mu_{\text{max}} = 3.81\%$.

Имеющиеся семь точек значительного падения скорости связаны с двумя светофорами, развязками дорог, разворотом в обратную сторону.

Для участка 0–42 км показатели движения следующие: $l = 42000$ м, $v_{\text{средн}} = 81.06$ км/ч, $v_{\text{max}} = 109$ км/ч (30.4 м/с), $\mu = 1.479\%$, $\mu_{\text{max}} = 3.81\%$.

Для участка 42–107,9 км показатели движения следующие: $l = 42-107,9$ км, $v_{\text{средн}} = 81.10$ км/ч, $v_{\text{max}} = 96,7$ км/ч (26.87 м/с), $\mu = 1.777\%$, $\mu_{\text{max}} = 4.39\%$.

То есть на трассе водитель стремится поддерживать постоянную скорость 23–25 км/ч несмотря на перепады высот.

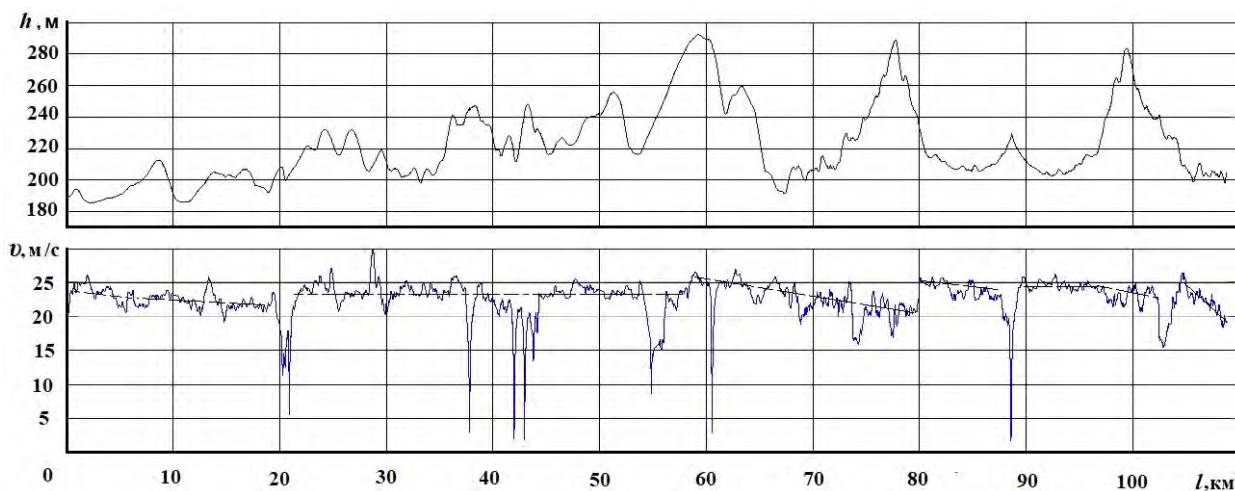


Рисунок 4.13. – Движение Абчак – Плещеницы – поворот от автозаправочной на Логойской трассе

На рисунке 4.14 представлены записи движения от поворота на трассе Р-53 от автозаправочной, далее – Логойск – Смиловичи.

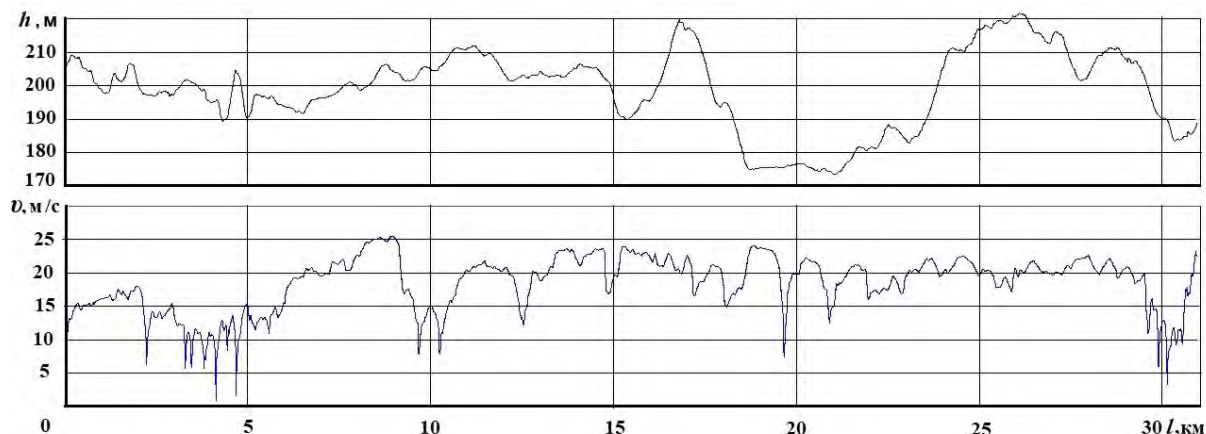


Рисунок 4.14. – Движение от поворота от автозаправочной – Логойск – Смиловичи

Это дорога местного значения: две полосы движения, много искусственных препятствий, пешеходный тротуар в населенных пунктах близко расположен к дороге, резкие повороты, проезд через три деревни, круговая развязка, что обуславливает низкую скорость движения на участке 0 – 6 км.

Для полного участка получены следующие данные: $l = 0\text{--}31,4$ км, $v_{\text{средн}} = 66,50$ км/ч, $v_{\text{max}} = 91,6$ км/ч (25,45 м/с), $\mu = 2,034\%$, $\mu_{\text{max}} = 4,09\%$.

Для участка 0–6 км, включая проезд н.п. Логойск имеем: $l = 0\text{--}6$ км, $v_{\text{средн}} = 47,99$ км/ч, $v_{\text{max}} = 64,6$ км/ч (17,94 м/с), $\mu = 3,130\%$, $\mu_{\text{max}} = 4,09\%$.

На нем видно 8 точек остановки или значительного снижения скорости, связанные с искусственными препятствиями, двумя светофорами, что характерно небольших населенных пунктов.

На участке 6–31,4 км наблюдаем 6 точек значительного снижения скорости из-за проезда 2-х населенных пунктов, участка кругового движения: $l = 6\text{--}31,4$ км, $v_{\text{max}} = 91,6$ км/ч (25,45 м/с), $\mu = 2,079$, $\mu_{\text{max}} = 3,87\%$.

После 11 км скорость стабилизируется в районе $v_{\text{средн}} = 70,87$ км/ч. Падение скорости после 30 км связано с выездом на трассу Р-59.

На рисунке 4.15 представлены записи движения от поворота с трассы Р-3 на Р-59 в направлении Минска и движение по городу. Этот участок имеет 4- полосы движения, характерные для автострад.

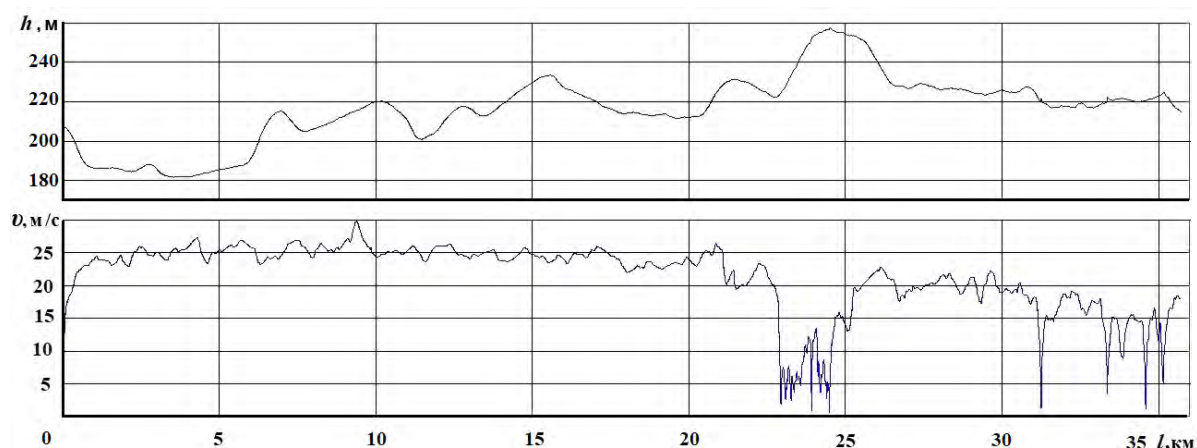


Рисунок 4.15. – Движение от Смиловичи – Курган Славы – 9-км – Минск

Для полного участка получены следующие данные: $l = 35380$ м, $v_{\text{средн}} = 76,94$ км/ч, $v_{\text{max}} = 107,2$ км/ч (29,8 м/с), $\mu = 1,539$, $\mu_{\text{max}} = 4,12\%$.

Для участка 0–23 км характерна высокая скорость движения и практически линейная зависимость до 22 км: $l = 0\text{--}23$ км, $v_{\text{средн}} = 86,5$ км/ч, $v_{\text{max}} = 107,2$ км/ч (29,8 м/с), $\mu = 1,855\%$, $\mu_{\text{max}} = 4,12\%$.

После 22 км наблюдается резкое падение скорости, связанное с ремонтом дороги. На участке 23–35,5 км наблюдается линейное падение скорости, связанное с городским движением, остановками, светофорами и поворотами: $l = 23\text{--}35,5$ км, $v_{\text{средн}} = 59.17$ км/ч, $v_{\text{max}} = 82$ км/ч (22.8 м/с), $\mu = 2.816\%$, $\mu_{\text{max}} = 4.65\%$.

Но в целом, этот участок показывает более высокую скорость в городе из-за 4-х полос движения и движущегося потока легковых автомобилей, возвращающихся из пригорода после выходных.

Как видно из графиков движение автомобиля представляет собой череду движения с переменной скоростью, торможений и разгона. В тоже время прослеживается стремление водителя поддерживать определенную среднюю скорость, зависящей от числа полос, категории дороги: на I категории $v_{\text{средн}} = 24$ км/ч, на II-ой $v_{\text{средн}} = 20$ км/ч, а городском режиме $v_{\text{средн}} = 14\text{--}15$ км/ч при наличии препятствий и до 17–18 км/ч без помех.

При значительном отличии удельной мощности предлагается использовать следующий подход. На основе выше полученных графиков задается режим движения, проводится моделирование движения. Полученные в процессе моделирования новые данные по режиму движения используются уже в качестве исходных для решения задач движения и нагруженности автомобиля (см. гл. 6).

4.5 Исследования элементов систем поддрессирования

Необходимость таких исследований обусловлена отсутствием экспериментальных данных по этим элементам, что приводило к использованию упрощенных моделей, неправильным выводам и дискредитировало моделирование. Ниже приведены экспериментальные данные, выполненные автором на МАЗе, приведенные в его кандидатской диссертации [47] и монографии [14].

4.5.1 Подушка и подвеска сиденья

Подушка и подвеска сиденья является одними из элементов, влияющих на виброзащитные свойства поддрессированного сиденья транспортного средства. Их свойства мало изучены. Подушка, как правило, не используется при расчетах [20, 21, 54, 56]. В качестве упругого и демпфирующего элемента подушки используется латексная губчатая резина 1-й и 2-й категории твердости. При использовании неподдрессированных сидений имеет место значительно усиление (в 3–4 раза) колебания водителя в области 3,5–7 Гц [14, 47, 76]. Последние совпадают с собственными частотами внутренних органов

человека [14, 41, 47] и их резонанс вызывает неприятные ощущения. Все это обуславливает необходимость проведения исследований характеристик подушек и подвески сиденья.

4.5.2 Методика стендовых исследований и использованная аппаратура

Исследование подушек [14, 76, 78] включало в себя:

- а) определение статических упругих характеристик;
- б) определение виброзащитных характеристик подушек и подвески сиденья;
- в) определение динамических, упругих и демпфирующих характеристик.

Определение статической упругой характеристики подушки и подвески сиденья производилось на стенде Минского автомобильного завода при медленном нагружении и разгрузке указанных элементов сиденья.

В качестве опорной поверхности при исследовании статической характеристики подушки использовался, согласно ГОСТ 16526, стальной диск диаметром 300 мм и толщиной 4 мм. Измерение усилий, приложенных к указанным элементам сиденья, осуществлялось с помощью образцового динамометра сжатия ДОС-0,5-200. Прогиб регистрировался через каждые 5 мм с помощью миллиметровой линейки. Статическая жесткость подушки и подвески сиденья определялась по тангенсу угла наклона касательной в точке, соответствующей $P = 600 \text{ Н}$.

Виброзащитные характеристики подушек и подвески сиденья определялись на электродинамических вибростендах ВЭДС-200А и ВЭДС-400А (рисунок 4.12) при максимальных амплитудах синусоидальных вертикальных ускорений основания, на которое крепилась подушка (подвеска сиденья), равных 1; 1,5; 2 м/с², в диапазоне частот 1,2...20 Гц. Виброзащитные характеристики подушек оценивались по коэффициенту передачи K_{Π} , которой определялся по отношению эффективной амплитуды вертикальных ускорений груза на подушке к величине эффективных ускорений основания:

$$K_{\Pi} = \frac{\ddot{z}_{\text{гр}}}{\ddot{z}_{\text{осн}}}. \quad (4.1)$$

Применяемый при испытаниях подушек груз состоял из трех плоских, жестко скрепленных между собой дисков диаметром 300 мм, имеющих массу 60 кг.

Измерение ускорений груза и основания производилось с помощью пьезоэлектрических датчиков ускорений КД-3,5а, виброизмерителя SM-231 и

октавно-полосового фильтра OF-201 (аппаратура фирмы “RFT”, суммарная погрешность 2–3% с учетом погрешности считывания показаний стрелочного прибора).

Для повышения точности при воспроизведении на стенде заданных величин ускорений учитывался небольшой завал частотной характеристики прибора SM-231 вместе с датчиком в низкочастотной области и вводилась соответствующая коррекция измерения.

Определение динамических характеристик подушек сидений заключалось в получении и регистрации двухкоординатной зависимости усилия, приложенного к сиденью, от относительного прогиба подвески при синусоидальном возбуждении основания сиденья на частоте 5 Гц.

Усилие, приложенное к подушке, определялось косвенным путем – через величину вертикальных ускорений поддрессоренной массы сиденья ($m_{гр} = 60$ кг) по формуле,

$$F = m_{гр} \cdot \ddot{Z}_{гр} \quad (4.2)$$

Вертикальное ускорение груза и относительная деформация подушки измерялись с помощью датчика ускорений BWH-201, индуктивного датчика перемещения JWT-302 с пределом измерений 100 мм и тензоусилителя UM-131 (аппаратура фирмы “PFT”, каждое устройство которой имеет погрешность $<1\%$). Регистрация результатов испытаний производилась с помощью электронного двухкоординатного осциллографа “Universal Indicator В-00” и кинокамеры (аппаратура фирмы “Disa Electronic” Дания, погрешность, которых $<1\%$).

Схема замеров и регистрации динамических характеристик подушки представлена на рисунок 4.16.



Рисунок 4.16. – Схема замеров и регистрации динамических характеристик подушки (а) и подвески сиденья (б)

Примененный при определении динамических характеристик подушек груз состоял из трех плоских, жестко скрепленных между собой дисков, имеющих массу 60 кг. Обработка полученных фотоснимков характеристик производилась в фотолаборатории с помощью фотоувеличителя (восьмикратное увеличение).

Жесткость подушки сиденья определялась по тангенсу угла наклона касательной к средней линии динамической характеристики при абсциссе относительных перемещений, равной нулю.

С целью упрощения модели подушки демпфирование в ней было представлено в виде линейного амортизатора с эквивалентным коэффициентом затухания $k_{\text{зат}}$, который рассчитывался по формуле

$$k_{\text{зат}} = \frac{v_{\text{рез}} * m_{\text{гр}}}{\sqrt{K_{\Pi}^2 - 1}}, \quad (4.3)$$

где ω_0 – резонансная частота груза на подушке, 1/сек;

$f_{\text{рез}} = 6,283 \cdot v_p$ – резонансная частота груза на подушке, Гц;

$f_{\text{рез}} = \frac{1}{6,283} \sqrt{\frac{c}{m_{\text{гр}}}}$ – резонансная частота груза на подушке, Гц;

K_{Π} – усредненный коэффициент передачи на резонансной частоте колебаний груза на подушке;

$m_{\text{гр}} = 60$ кг – масса груза.

Указанная расчетная формула для $k_{\text{зат}}$ получена путем упрощения нижеприведенной общеизвестной формулы [47], учитывая, что при резонансе $v \approx \omega_0$,

$$K_{\Pi} = \frac{\ddot{z}_{\text{гр}}}{\ddot{z}_{\text{осн}}} = \sqrt{\frac{\omega_0^4 + \left(\frac{k_{\text{зат}}}{m_{\text{гр}}}\right)^2 v^2}{(v^2 - \omega_0^2)^2 + \left(\frac{k_{\text{зат}}}{m_{\text{гр}}}\right)^2 v^2}}; \quad (4.4)$$

где v и ω_0 – соответственно частота вынужденных и собственных колебаний груза на подушке;

$$\omega_0 = \sqrt{\frac{c}{m_{\text{гр}}}} \quad (4.5)$$

Помимо коэффициента затухания, определялся также относительный коэффициент затухания колебаний груза на подушке ψ по общеизвестной формуле

$$\Psi = \frac{k_{\text{зат}}}{2\sqrt{C \cdot m_{\text{гр}}}}, \quad (4.6)$$

где C и $k_{\text{зат}}$ – соответственно динамическая жесткость и коэффициент затухания подушки; $m_{\text{гр}} = 60$ кг - масса груза на подушке.

4.5.3 Результаты стендовых исследований

Результаты исследований подушек сидений приведены на рисунках 4.17–4.20 [47].

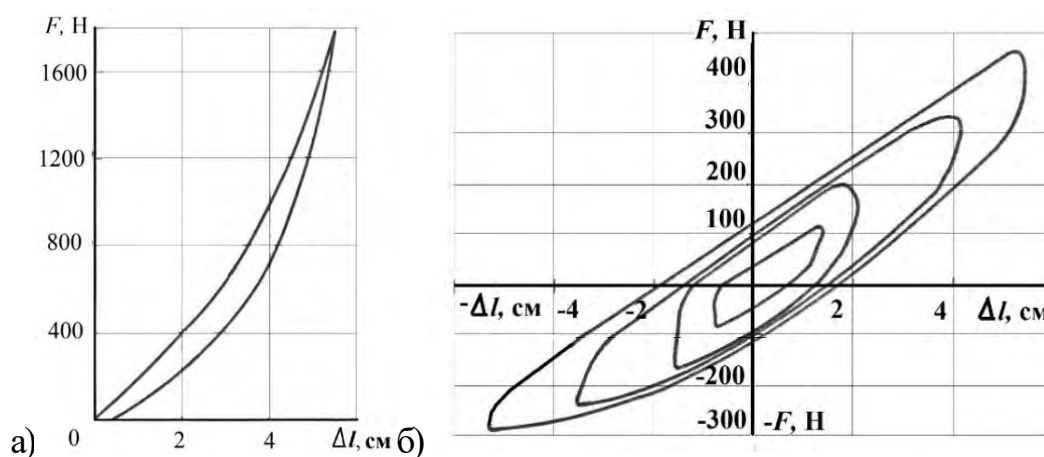


Рисунок 4.17 – Статическая (а) и динамическая (б) характеристика подушки сиденья

Из приведенных на рисунках 4.17–4.18 зависимостей видно, что с увеличением толщины прокладки с 50 до 120 мм уменьшаются статические и динамические жесткости подушек, а также снижается частота резонанса груза на подушке. При указанном изменении толщины прокладки статическая жесткость подушки из резины I-й категории твердости снижается с 39,1 до 18,2 кН/м, а динамическая – с 90,5 до 32 кН/м.

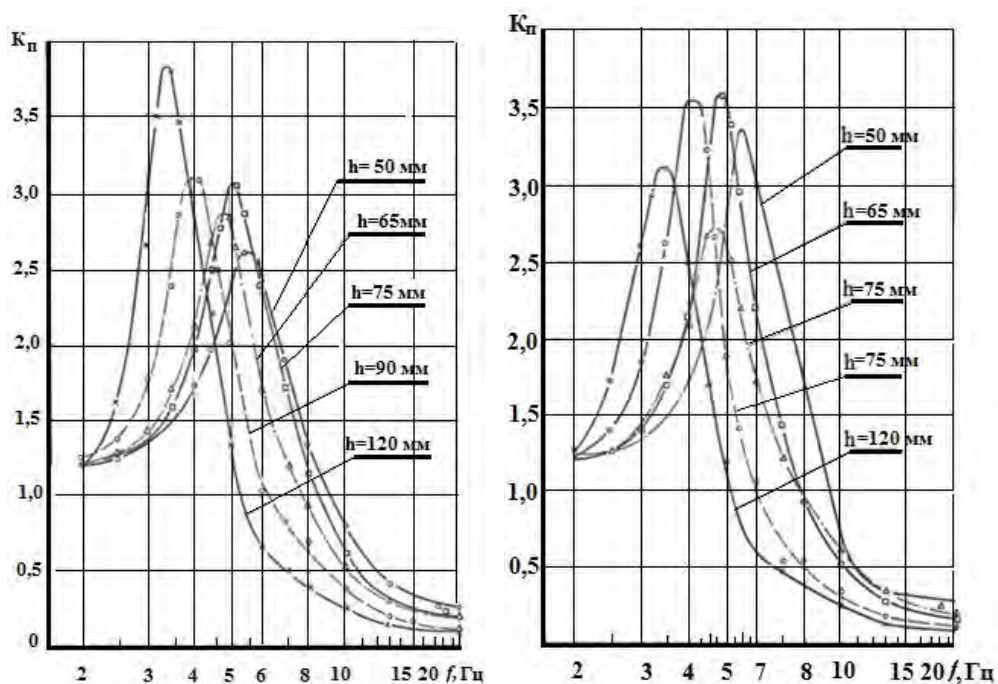


Рисунок 4.18. – Амплитудно-частотные характеристики колебаний груза на подушках сидений в зависимости от толщины подушки и категории твердости резины (слева 1-й категории, справа 2-й)

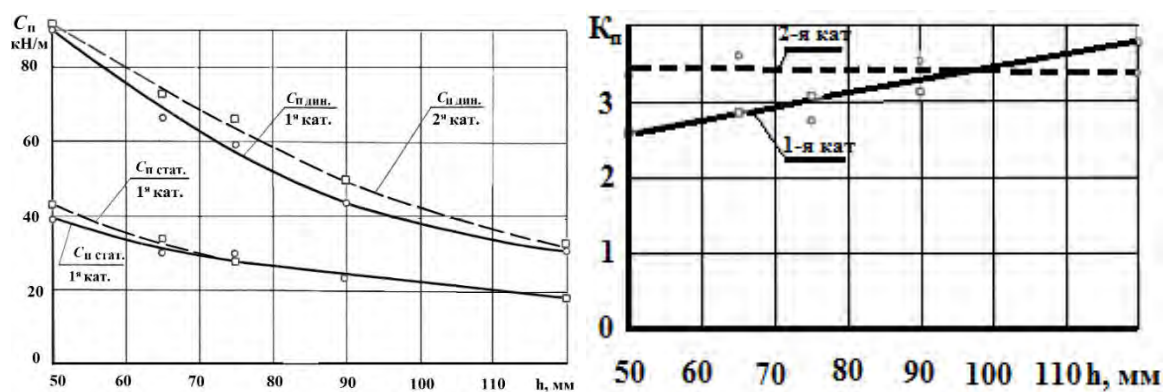


Рисунок 4.19. – Изменение статической и динамической жесткости подушки и коэффициента передачи вибраций в зависимости от категории резины и толщины прокладки

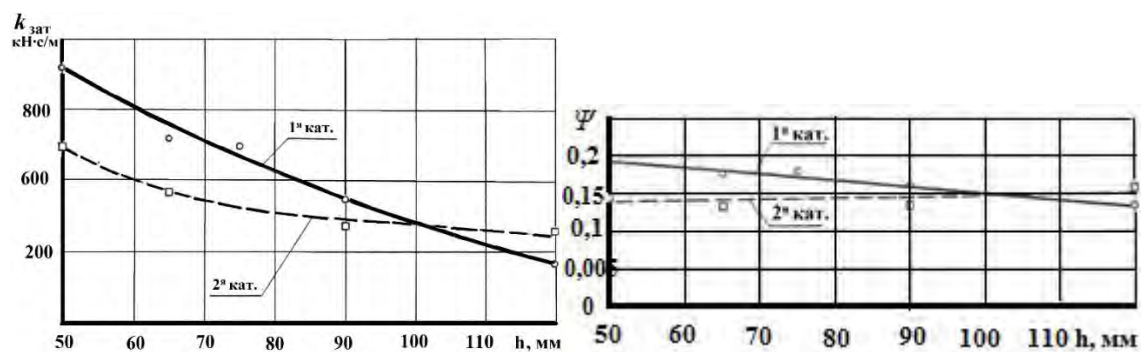


Рисунок 4.20. – Изменение коэффициентов затухания в зависимости от категории резины и ее толщины

Частота резонанса груза на подушке при этом уменьшается с 5,6 до 3,48 Гц. Для подушек из резины 2-й категории твердости наблюдается аналогичный характер изменения указанных параметров.

Анализ, приведенных на рисунках, данных показывал, что:

- при динамическом нагружении жесткость подушки в 1,8–2,4 раза больше, чем в статике;
- относительный коэффициент затуханий колебаний груза на подушке составляет 0,13–0,19 и 0,145–0,16 соответственно для подушек из резины I-й и 2-й категории твердости;
- подушки из резины I-й категории твердости при толщине прокладки $h = 50\text{--}100$ мм обеспечивают несколько лучшее затухание колебаний ($k_{\text{зат}} = 912\text{--}285$ Н·с/м), чем подушки из резины 2-й категории твердости ($k_{\text{зат}} = 687\text{--}285$ Н·с/м).

Проверка полученных значений динамической жесткости и коэффициентов затуханий по формуле (5.2) (на линейной модели) показала, что обеспечивается хорошее совпадение расчетной амплитудно-частотной характеристики (АЧХ) с экспериментальной и приемлемая для расчетных исследований точность аппроксимации виброзащитных свойств подушки в диапазоне частот 2–13 Гц, где расхождение расчетных и экспериментальных коэффициентов передачи не превышает 10–15%. На частоте 20 Гц расхождение увеличивается и достигает 30–40 %. Учитывая, что уровень вибрации в октавной полосе 16 Гц (11,2–22,4 Гц) небольшой и не является лимитирующим для большинства грузовых автомобилей. Это дает основание для использования линейной модели подушки при расчетном исследовании сиденья, когда не требуется высокая точность в этой области.

В тех случаях, когда требуется более высокая точность (погрешность <10 %) в октавной полосе 16 Гц, целесообразно использовать следующую модель (рисунок 4.21), в виде 2-х упругих элементов (соответствующих статической и динамической жесткостям. Последняя ограничена величиной “сухого” трения и двух демпфирующих элементов (один с чисто линейной, второй также с линейной характеристикой, но ограниченной максимальной величиной).

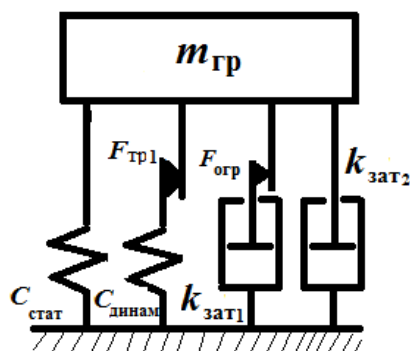


Рисунок 4.21. – Модель подушки, обеспечивающая более корректное совпадение в октавной полосе 16 Гц

Усилия упругих и демпфирующих элементов можно описать следующими уравнениями [14, 76], базируясь на анализе работ [14, 47],

$$F_{\text{тр1}} = \begin{cases} C_{\text{дин}} \int_0^t (\dot{z}_{\text{осн}} - \dot{z}_{\text{сид}}) dt & F_{\text{тр1max}} > C_{\text{тр}} \int_0^t (\dot{z}_{\text{осн}} - \dot{z}_{\text{сид}}) dt > -F_{\text{тр1z}} \\ F_{\text{тр1}} & \text{если } C_{\text{дин}} \int_0^t (\dot{z}_{\text{осн}} - \dot{z}_{\text{сид}}) dt > F_{\text{тр1}} \\ -F_{\text{тр1}} & C_{\text{дин}} \int_0^t (\dot{z}_{\text{осн}} - \dot{z}_{\text{сид}}) dt < -F_{\text{тр1z}} \end{cases}$$

$$F_{\text{огр}} = \begin{cases} k_{\text{зат1}} (\dot{z}_{\text{осн}} - \dot{z}_{\text{сид}}) & F_{\text{тр1}} > k_{\text{зат1}} (\dot{z}_{\text{осн}} - \dot{z}_{\text{сид}}) > -F_{\text{огр}} \\ F_{\text{тр1}} & \text{если } k_{\text{зат1}} (\dot{z}_{\text{осн}} - \dot{z}_{\text{сид}}) > F_{\text{огр}} \\ -F_{\text{тр1}} & k_{\text{зат1}} (\dot{z}_{\text{осн}} - \dot{z}_{\text{сид}}) < -F_{\text{огр}} \end{cases}$$

$$F_{\text{зат2}} = k_{\text{зат2}} (\dot{z}_{\text{осн}} - \dot{z}_{\text{сид}}); F_{\text{стат}} = C_{\text{стат}} (Z_{\text{осн}} - Z_{\text{сид}});$$

где $C_{\text{стат}}$, $C_{\text{дин}}$ – статическая и динамическая жесткость подушки; $k_{\text{зат1}}$, $k_{\text{зат2}}$ – коэффициенты затуханий в подушке; $F_{\text{тр1}}$, $F_{\text{огр}}$ – максимальные значения трения, ограничивающего усилия от динамической жесткости и демпфирующего элемента; $\dot{z}_{\text{осн}}$, $\dot{z}_{\text{угр}}$ – текущее значение виброскорости основания и груза на подушке; $Z_{\text{стат}}$, $Z_{\text{гр}}$ – текущее значение перемещений основания и груза на подушке.

Параметры этих элементов в зависимости от толщины прокладки приведены на рисунок 4.22. Значения величин: $F_{\text{огр}} = 0,005 \text{ кН}\cdot\text{с/м}$, $F_{\text{тр1}} = 0,0029 \text{ кН}\cdot\text{с/м}$.

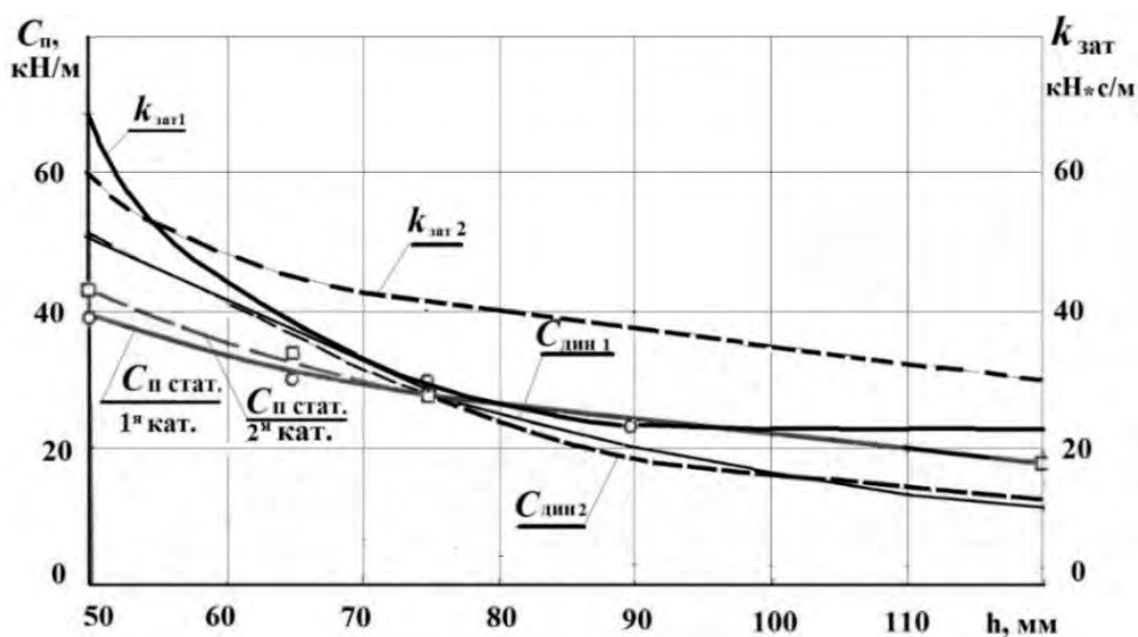
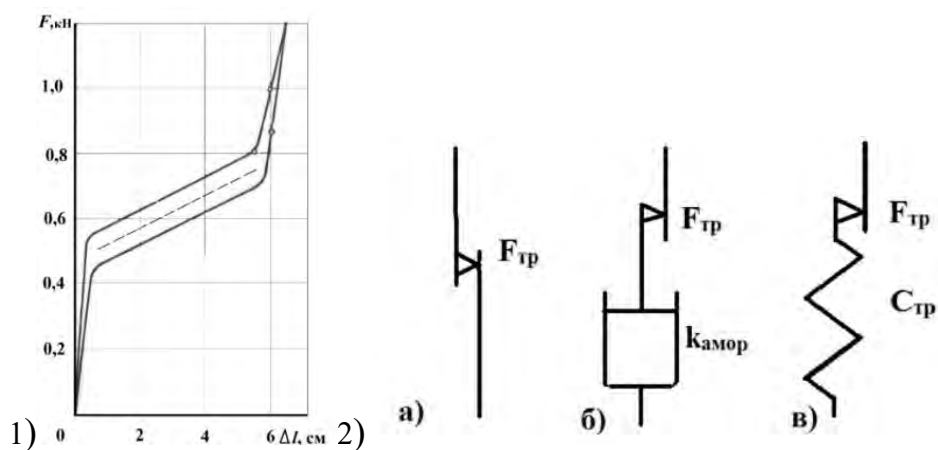


Рисунок 4.22. – Изменение параметров усовершенствованной колебательной модели подушки в зависимости от категории резины и толщины прокладки

4.6 Характеристики подвески сиденья

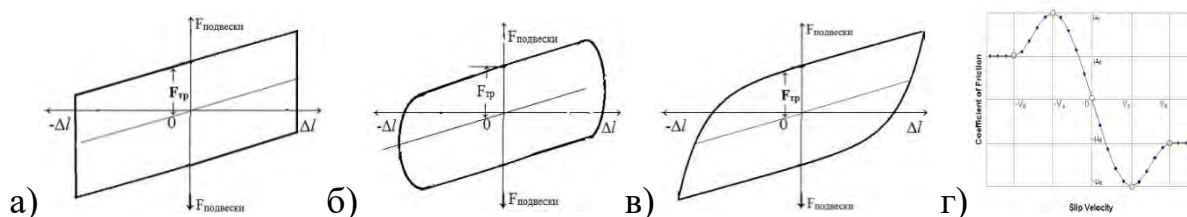
В качестве характеристики подвески сиденья обычно используется зависимость $F=f(\Delta l)$ с учетом гистерезисной петли трения (рисунок 4.23).

На сходимость колебаний сильное влияние оказывает модель трения [15, 16]. Их схемы и зависимости приведены на рисунках 4.24.



а – Кулонового трения, б – «вязкого» трения, в – «жесткостного» трения Кольцова

Рисунок 4.23. – Статическая характеристика подвески сиденья (1), схемы различных моделей трения (2)



а – Кулонового трения, б – «вязкого» трения, в – «жесткостного» трения Кольцова
г – зависимость $F_{тр}$ от скорости, используемая в пакете MSC.ADAMS

Рисунок 4.24. – Вид упругих характеристик при различных моделях трения

В основном при колебаниях используют модель "вязкого" трения в виде зависимости от скорости с ограничением ее максимальной величины. Модель "кулонового" трения практически уже не используется, так как она приводит к неустойчивости системы. Как показывают работы [43, 45, 47] модель «вязкого» трения обеспечивает приемлемую сходимость (10–15%) только при моделировании переезда автомобилем препятствий высотой 10 см и моделировании движения по дорогам со среднеквадратическими значениями неровностей > 2 см в общей полосе частот 0–22,4 Гц или моделировании сиденья при постоянной амплитуде возмущений (> 2 см), но не отражает характер колебаний на реальной дороге с меньшими амплитудами возмущений.

В результате проведенных исследований установлено, что трение в подвеске сиденья носит "жесткостной" характер с наличием трения покоя со всплеском усилия при смене направления скорости и установившимся затем состоянием движения [14, 47, 48] (рисунок 4.25).

На сиденьи с грузом (без подушки и амортизатора) наблюдается второй небольшой пик резонансных колебаний (10 Гц) при ускорениях на стенде, равных 1 м/с^2 , 2 м/с^2 (рисунок 4.26).

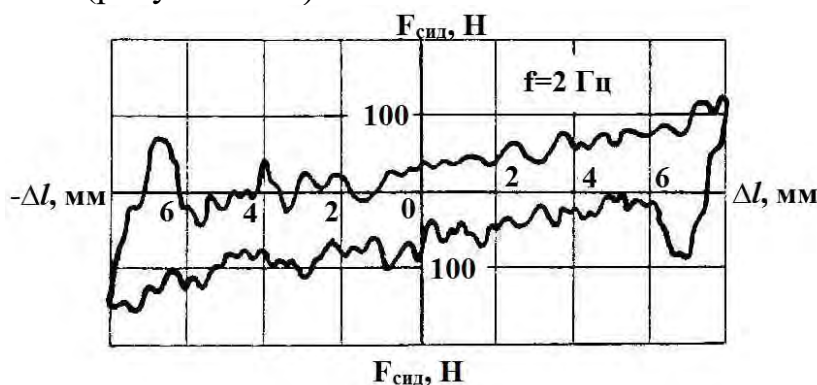


Рисунок 4.25 – Динамическая характеристика поддресоренного сиденья

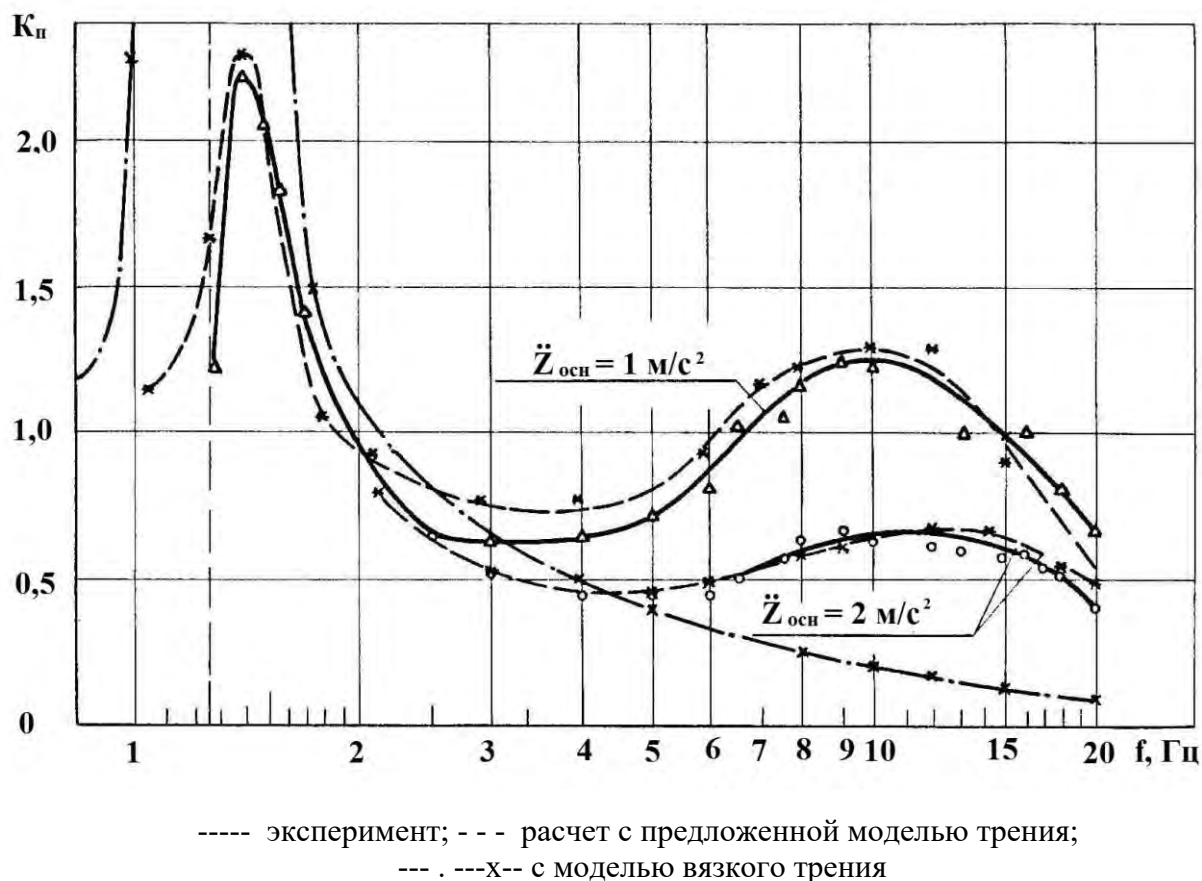


Рисунок 4.26 - Амплитудно-частотные характеристики подрессоренного сиденья без подушки и амортизатора

При анализе амплитудно-частотной характеристики (АЧХ) было установлено, что при малых возмущениях из-за блокировки подвески левая ветвь резонансного пика смещается вправо, происходит сужение резонансной области. А в области высоких частот из-за значительного уменьшения амплитуд перемещений основания и большой величины трения покоя также происходит блокировка подвески. Из-за чего возникают колебания на податливостях рычагов и других элементах, чем и объясняется наличие второго пика АЧХ.

В то же время при постоянных амплитудах возмущения ($Z=10 \text{ мм}$, 20 мм) смещение левой ветви резонансной характеристики и второго пика не наблюдалось. Исследования показали, что модель “вязкого” трения при малых амплитудах возмущения не отражает реальную амплитудно-частотную характеристику сиденья и низкая эффективность подрессоренных сидений во многом обусловлена трением в подвеске.

На основании этих исследований предложена следующая математическая аппроксимация трения и ее реализация на аналоговых машинах.

$$\begin{aligned}
F_{\text{тр1}} = \begin{cases} C_{\text{тр}} \int_0^t (\dot{z}_{\text{осн}} - \dot{z}_{\text{сидн}}) dt & F_{\text{тр1max}} > C_{\text{тр}} \int_0^t (\dot{z}_{\text{осн}} - \dot{z}_{\text{сидн}}) dt > -F_{\text{тр1max}} \\ F_{\text{тр1max}} & \text{если } C_{\text{тр}} \int_0^t (\dot{z}_{\text{осн}} - \dot{z}_{\text{сидн}}) dt > F_{\text{тр1max}} \\ -F_{\text{тр1max}} & C_{\text{тр}} \int_0^t (\dot{z}_{\text{осн}} - \dot{z}_{\text{сидн}}) dt < -F_{\text{тр1max}} \\ \left[a \frac{dF_{\text{тр1}}}{dt} - b \frac{d^2 F_{\text{тр1}}}{dt^2} \right] & F_{\text{тр2max}} > \left(a \frac{dF_{\text{тр1}}}{dt} - b \frac{d^2 F_{\text{тр1}}}{dt^2} \right) < -F_{\text{тр1max}} \end{cases} \\
F_{\text{тр2}} = \begin{cases} +F_{\text{тр2max}} & \text{если } \left(a \frac{dF_{\text{тр1}}}{dt} - b \frac{d^2 F_{\text{тр1}}}{dt^2} \right) > -F_{\text{тр2max}} \\ -F_{\text{тр2max}} & \left(a \frac{dF_{\text{тр1}}}{dt} - b \frac{d^2 F_{\text{тр1}}}{dt^2} \right) < -F_{\text{тр2max}} \end{cases}
\end{aligned}$$

где - $C_{\text{тр}}$ суммарная жесткость податливостей рычагов подвески, $F_{\text{тр2max}}$ – величина втулок при блокировке подвески, $F_{\text{тр1max}}$ – величина трения движения, трения покоя; $F_{\text{тр1max}}$, $F_{\text{тр2max}}$ – максимальное значение трения покоя и скольжения;

$\dot{Z}_{\text{осн}}$, $\dot{Z}_{\text{сид}}$ – текущее значение виброскорости основания и сиденья.

Отличие предложенной модели от модели Кольцова заключается в реализации двух состояний "жесткостного" трения: покоя и движения с всплеском амплитуды покоя в 2,5 раза больше чем при движении. Данная модель использовалась на аналоговых машинах и обеспечивала хорошую сходимость по среднеквадратичным ускорениям в октавных полосах частот (2–8%) при моделировании колебаний сиденья с водителем в реальном масштабе времени при использовании магнитозаписи ускорений пола кабины при движении по асфальтовому и булыжному шоссе, а также по АЧХ на стенде при ускорениях, равных 1 м/с², 2 м/с² и постоянных амплитудах перемещений 10 мм, 20 мм.

Сейчас вместо аналоговых машин используется пакет Matlab/Simulink на персональных компьютерах. В связи с чем для него предложена и апробирована следующая блок-схема одномассовой колебательной системы сиденья (рисунок 4.27).

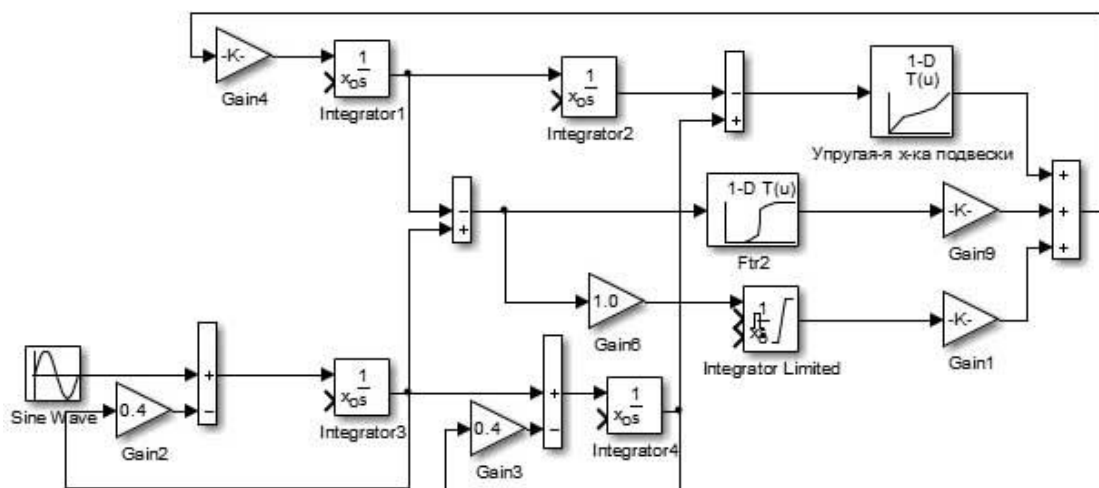


Рисунок 4.27 – Блок-схема реализации одномассовой колебательной системы сиденья с предложенной моделью «жесткого» и «вязкого» трения

Трение в сиденье было представлено как совокупность «жесткого» ($F_{\text{тр1}}$) и «вязкого» трения ($F_{\text{тр2}}$). «Вязкое» трение моделировалось как зависимость усилий сопротивления от относительной скорости с ограничением величины. Такое решение было вызвано сложностью реализации на Matlab/Simulink блока дифференциатора со сглаживающим элементом (на АВМ – это последовательное соединение сопротивления с емкостью во входной цепи), который был необходим для устойчивости процесса дифференцирования. Из-за чего дифференциатор на Simulink исключен и вместо него был включен узел «вязкого» трения ($F_{\text{тр2}}$).

Данная модель описывается следующими уравнениями,

$$F_{\text{тр1}} = \begin{cases} C_{\text{дин}} \int_0^t (\dot{z}_{\text{осн}} - \dot{z}_{\text{сид}}) dt & F_{\text{тр1max}} > C_{\text{тр}} \int_0^t (\dot{z}_{\text{осн}} - \dot{z}_{\text{сид}}) dt > -F_{\text{тр1max}} \\ F_{\text{тр1}} & \text{если } C_{\text{дин}} \int_0^t (\dot{z}_{\text{осн}} - \dot{z}_{\text{сид}}) dt > F_{\text{тр1max}} \\ -F_{\text{тр1}} & C_{\text{дин}} \int_0^t (\dot{z}_{\text{осн}} - \dot{z}_{\text{сид}}) dt < -F_{\text{тр1max}} \end{cases}$$

$$F_{\text{тр2}} = \begin{cases} [k_{\text{ам}} (\dot{Z}_{\text{осн}} - \dot{Z}_{\text{сид}})] & F_{\text{тр1}} > k_{\text{ам1}} (\dot{Z}_{\text{осн}} - \dot{Z}_{\text{сид}}) > -F_{\text{амmax}} \\ F_{\text{тр1max}} & \text{если } k_{\text{ам1}} (\dot{Z}_{\text{осн}} - \dot{Z}_{\text{сид}}) > F_{\text{амmax}} \\ -F_{\text{тр1max}} & k_{\text{ам1}} (\dot{Z}_{\text{осн}} - \dot{Z}_{\text{сид}}) < -F_{\text{амmax}} \end{cases}$$

$$F_{\text{стат}} = C_{\text{стат}} (Z_{\text{осн}} - Z_{\text{сид}});$$

где $C_{\text{тр}}$ – суммарная жесткость податливостей рычагов подвески, втулок;
 $F_{\text{тр1max}}$, – максимальное значение трения покоя;
 $f_{\text{тр2}} (\dot{Z}_{\text{осн}} - \dot{Z}_{\text{сид}})$ – функция вязкого трения от относительной скорости с ограничением ее максимальной величины;

$\dot{Z}_{\text{осн}}$, $\dot{Z}_{\text{сид}}$ – текущее значение виброскорости основания и сиденья.

Были подобраны следующие параметры модели “вязкого” трения, приведенные в таблице 4.9, обеспечивающие приемлемую сходимость результатов моделирования.

Таблица 4.9 – Параметры «вязкого» трения

Параметры	Значения величины трения и скорости «вязкого» трения								
$f_{\text{тр2}}, \text{Н}$	-40	-40	-35	-25	0	25	35	40	40
$v, \text{м/с}$	-0,25	-0,12	-0,08	-0,03	0	0,03	0,08	12	25

Моделирование проводилось при следующих параметрах: $m_{\text{груза}} = 80 \text{ кг}$, $C_{\text{подвески}} = 54 \text{ Н/см}$, величина ограничения интегратора “жесткого” трения – $0,002 \text{ м}$, $C_{\text{тр}} = 20000 \text{ Н/см}$. Виброзащитные свойства сиденья оценивались по коэффициенту передачи $K_n = \dot{Z}_{\text{сиденье}} / \dot{Z}_{\text{основание}}$.

Выполненное с помощью приведенной блок-схемы моделирование амплитудно-частотной характеристики поддрессоренного сиденья при ускорениях 1 м/с^2 , 2 м/с^2 (таблица 4.10) показало хорошую сходимость с ранее выполненными стендовыми испытаниями [47] в полосе частот 2–20 Гц (расхождение 5–15 %).

Таблица 4.10 – Результаты моделирования амплитудно-частотной характеристики поддрессоренного сиденья

Амплитуда ускорений, м/с^2	Частота, Гц					
	Коэффициент передачи вибраций					
	2 Гц	3 Гц	8 Гц	10	15	20
1 (эксперим.)	0,9	0,65	1,18	1,2	1,0	0,65
1 (расчет)	1,0	0,65	1,0	1,05	1,05	0,6
2 (эксперим.)	0,9	0,53	0,68	0,71	0,58	0,4
2 (расчет)	0,95	0,575	0,72	0,68	0,57	0,45

4.7 Экспериментальное исследование влияния подвески сиденья

4.7.1. Влияние жесткости подвески сиденья

Исследование влияния жесткости подвески на виброзащитные свойства сиденья, оцениваемые по K_n , проводились на автомобиле МАЗ-5336, (оборудованном передними рессорами $l = 1630$ мм, $C_p = 225$ кН/м) при движении его по асфальтированному и булыжному шоссе со скоростью 70 и 60 км/ч с различными типами амортизаторов: газонаполненным (5320-6804672) и опытным с двумя выходами штока, обладающими вязкостной зависимостью сопротивления при жесткости подвески сиденья 504В-6800010-02 4,3 и 6,5 кН/м [47].

Результаты испытаний показали, что уменьшение жесткости подвески в 1,5 раза приводит к улучшению виброзащитных свойств подрессоренного сиденья 504В-6800010-02. Для сиденья с амортизаторами КамАЗ выигрыш по K_n в октавах 2; 4 Гц составляет соответственно в среднем 10,9; 8,9% на асфальтированном шоссе и 5,5; 8,8% на булыжном шоссе, а для сиденья с амортизатором с двумя выходами штока при регулировке А-2, соответствующей минимальным значениям K_n , и отсутствию пробоев подвески соответственно 22,4; 7,3% и 14; 4,95%.

При увеличении сопротивления амортизатора с двумя выходами штока эффект от снижения жесткости уменьшается. Так, уже при регулировке А3 улучшение коэффициента виброизоляции сиденья K_v составляет в октавах 2; 4 Гц соответственно 13,7; 5,2% на асфальтированном шоссе и 11,6; 11% на булыжном шоссе. При регулировке амортизатора А7 имеет место незначительное улучшение виброзащитных свойств сиденья в октаве 2 Гц, которое составляет 5,8; 8,3% соответственно на асфальтированном и булыжном шоссе.

Полученные результаты исследования показывают целесообразность уменьшения жесткости подвески сиденья до величины $C_{сид} = 4,3$ кН/м при любом типе амортизатора и хорошо подтверждают выводы расчетного исследования.

4.7.2. Влияние типа и характеристики амортизатора на виброзащитные свойства подрессоренного сиденья

Сравнительные испытания сиденья 504В-6800010-02 с различными амортизаторами проводились в два этапа: на первом этапе были испытаны амортизаторы КамАЗ и ВАЗ, исследование которых было проведено на автопоезде МАЗ-5432-9397. Примененное при испытаниях подрессоренное

сиденье 504В-6800010-01 имело параметры: $C_{\text{сид}} = 5,4$ кН/м; $F_{\text{тр}} = 45...50$ Н и было оборудовано подушкой из губчатой резины толщиной $h = 75$ мм. Характеристики упругих и демпфирующих элементов сиденья, испытанного на автопоезде МАЗ-5432-9397, приведены на рисунках 4.25, 4.26.

Результаты испытаний сиденья 504В-6800010-02 с амортизатором ВАЗ и КамАЗ, приведенные на рисунок 4.28, 4.29 показывают, что применение амортизатора ВАЗ вместо амортизатора КамАЗ снижает коэффициенты передачи вибраций K_v сиденья 504В-6800010-02 на асфальтированном шоссе в среднем на 9,2; 28,7; 6,7% соответственно в октавах 2; 4; 16 Гц. Исключение составляет октава 8 Гц, где коэффициент виброизоляции сиденья с амортизатором ВАЗ на 8,1% выше, чем с амортизатором КамАЗ.

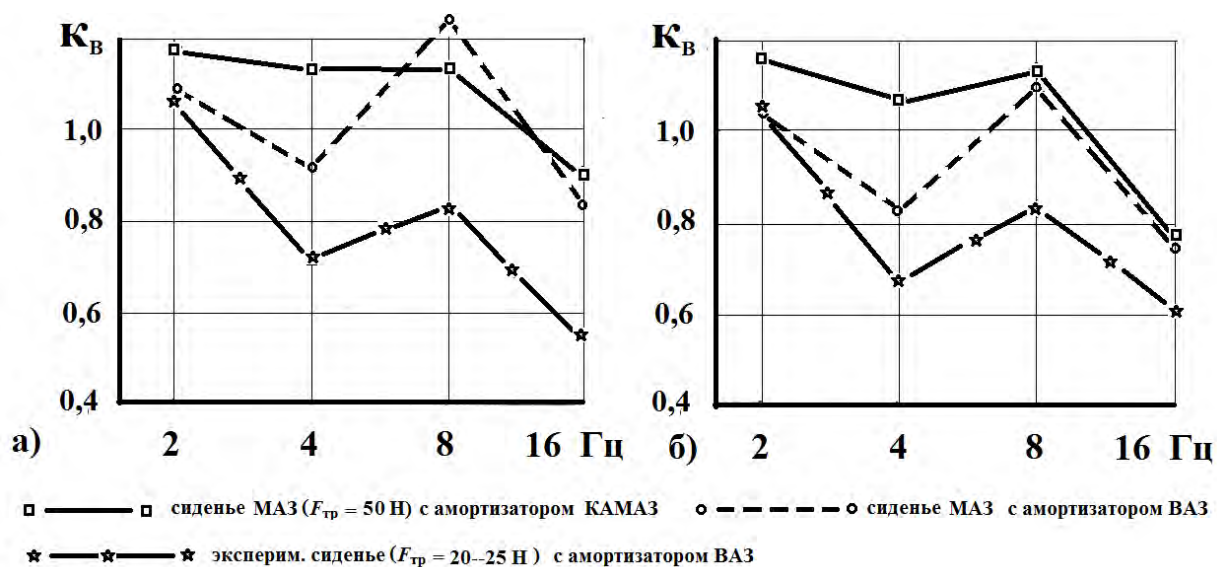


Рисунок 4.28 – Влияние типа и величины “сухого” трения на виброзащитные свойства сиденья при движении автопоезда МАЗ-5432-9397 по асфальтированному шоссе со скоростью $V_a = 50$ км/ч (а) и $V_a = 70$ км/ч (б)

При движении автопоезда по булыжному шоссе выигрыш от применения на сиденье амортизатора ВАЗ по сравнению с сиденьем с амортизатором КамАЗ несколько меньше, чем при движении автопоезда по асфальтированному шоссе, а именно 5,1% в октаве 2 Гц, 20,3% в октаве 4 Гц, 5,3% в октаве 16 Гц. В октаве 8 Гц наблюдается некоторое ухудшение виброзащитных свойств сиденья с амортизатором ВАЗ (на 7,7%) по сравнению с сиденьем с амортизатором КамАЗ. По субъективной оценке водителя сиденье с амортизатором КамАЗ значительно жестче, чем сиденье с амортизатором ВАЗ, однако в последнем случае имеют место отдельные пробои подвески.

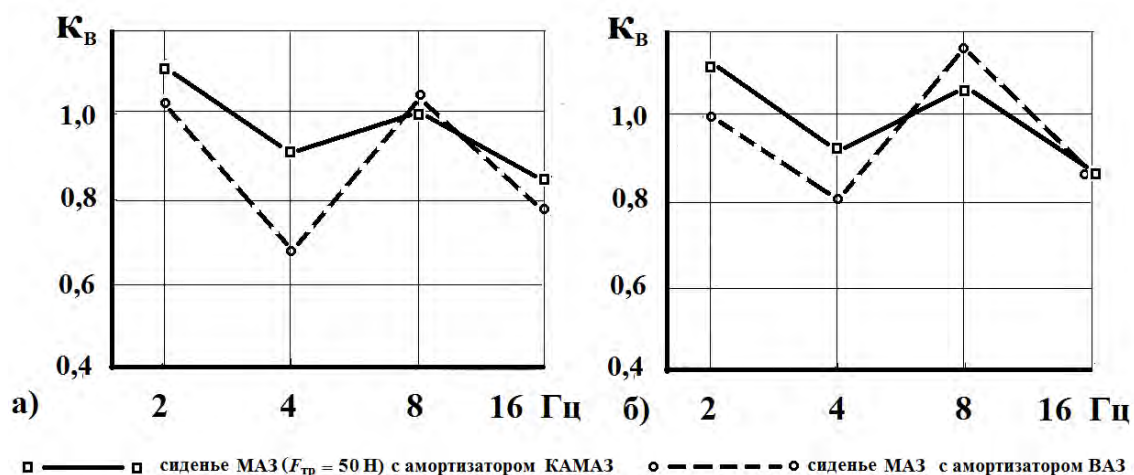


Рисунок 4.29 – Влияние типа и величины “сухого” трения на виброзащитные свойства сиденья при движении автопоезда МАЗ-5432-9397 по булыжному шоссе со скоростью $V_a = 30$ км/ч (а) и $V_a = 50$ км/ч (б)

В целом результаты проведенных исследований показывают следующее:

- использование газонаполненного амортизатора приводит к увеличению вибронагруженности рабочего места водителя, что делает малопригодным его применение в подвеске сиденья;

- двухтрубный амортизатор из-за наличия пробоев не в полной мере отвечает требованиям, предъявляемым к демпфирующим устройствам сиденья, хотя и обладает определенными преимуществами перед газонаполненными. Его использование оправдано лишь при небольших масштабах выпуска сидений, когда можно добиться необходимого для исключения пробоев подвески повышения сопротивления сжатия путем тщательной регулировки клапанных систем.

Выявленные в процессе исследования недостатки существующих типов амортизаторов потребовали создания специального типа амортизатора для подрессоренного сиденья, обладающего вязкостной симметричной характеристикой.

Для реализации указанной характеристики был создан экспериментальный макетный образец регулируемого амортизатора с двумя выходами штока и проведены исследования по выбору его оптимальных параметров на автомобиле МАЗ-5336. Наряду с этим амортизатором для сравнения были проведены аналогичные испытания двух амортизаторов КамАЗ при различных регулировках сопротивления. Указанные испытания представляли собой второй этап исследований по влиянию амортизатора на виброзащитные свойства подрессоренного сиденья.

Примененное при испытаниях подрессоренное сиденье 504В-6800010-02 было укомплектовано подушкой из губчатой латексной резины 2-й категории

твердости, имеющей толщину $h = 75$ мм. Характеристики подушки и подвески сиденья представлены на рисунках 4.20, 4.21. Замеры ускорений на сиденье производились на стальном диске в соответствии с ГОСТ 16526-70.

Результаты экспериментального исследования показаны на рисунке 4.30.

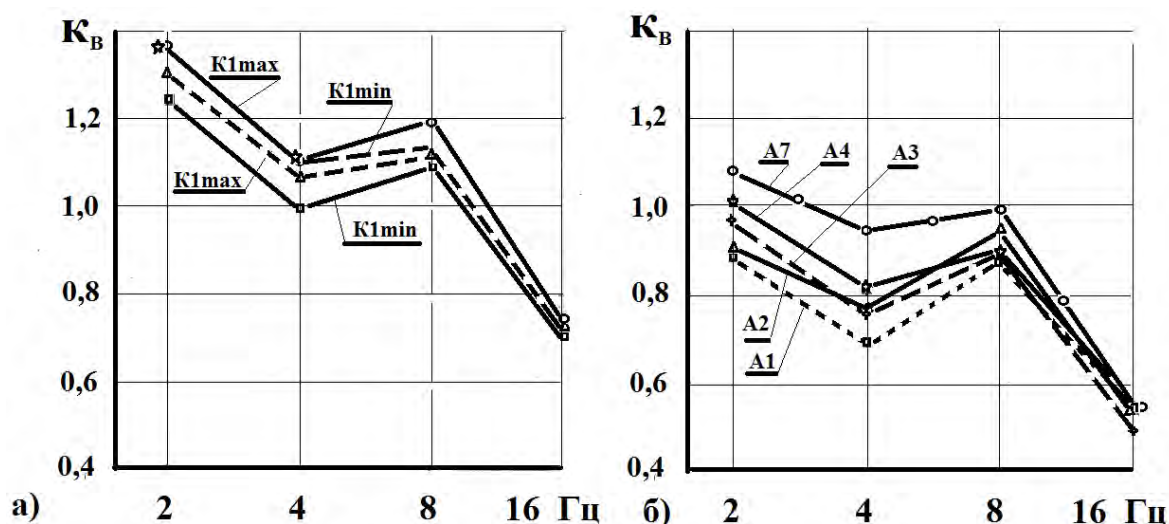


Рисунок 4.30 – Влияние типа амортизатора: газонаполненного КАМАЗ (а) и с двумя выходами штока (б) и их регулировок на виброзащитные свойства сиденья при движении автопоезда МАЗ-5432-9397 по асфальтированному шоссе со скоростью $V_a = 70$ км/ч

Как видно из полученных данных, изменение усилий сжатия в пределах регулировки амортизаторов КамАЗ слабо на изменение коэффициентов виброизоляции сиденья в октавных полосах частот. Коэффициенты виброизоляции сиденья с этими амортизаторами, полученные для широких полос частот 0...5,6 Гц; 0...22,4 Гц, более наглядно подтверждает эти выводы.

Численные значения коэффициентов виброизоляции сиденья 504В-6800010-02 $C_{сид} = 4,3$ кН/м с амортизатором КамАЗ (регулировка $K4max$) на асфальтированном шоссе составляют 132,5; 113; 113,5; 77% соответственно в октавах 2; 4; 16 Гц, что свидетельствует об усилении вибраций в октавных полосах частот 2; 4; 8 Гц.

Анализ полученных экспериментальных данных с амортизатором с двумя выходами штока показал, что уменьшение сопротивления этого амортизатора при регулировке с А7 до А2 приводит к значительному улучшению виброзащитных свойств поддрессоренного сиденья 504В-6800010-02. Пробои подвески при этом отсутствовали.

Оптимальное сопротивление амортизатора с двумя выходами штока при $C_{сид} = 4,3$ кН/м соответствует регулировке А2. Дальнейшее уменьшение сопротивления амортизатора приводит к появлению пробоев подвески, вследствие чего значения в высокочастотных октавах не уменьшаются.

Степень передачи вибраций сиденья K_p с опытным амортизатором с двумя выходами штока при оптимальной регулировке A_2 и жесткости подвески $C_{сид} = 4,3$ кН/м составляет на асфальтированном шоссе 110,9; 95,7; 99,4; 75% соответственно в октавах 2; 4; 8; 16 Гц, что на 18; 15; 12; 11% лучше, чем с амортизатором КамАЗ.

Результаты экспериментального исследования полностью подтверждают выводы расчетного исследования о преимуществах амортизатора с вязкостной зависимостью сопротивления перед другими типами амортизаторов и перспективности дальнейшей работы в направлении создания более технологичной конструкции однотрубного амортизатора (рисунок 4.31), в которой за счет большого соотношения диаметра поршня к диаметру штока и низкого давления ($p = 1$ Па) компенсационной камеры достигается вышеуказанная характеристика.

Примечание. Амортизатор устанавливается компенсационной камерой вверх.



Рисунок 4.31. – Предлагаемая конструкция регулируемого амортизатора для сиденья

4.7.3. Влияния величины "сухого" трения в подвеске

С целью исследования влияния величины "сухого" трения в подвеске на виброзащитные свойства сиденья на автопоезде МАЗ-5432-9397 были проведены сравнительные испытания серийного сиденья 504В-68000І0-02 ($C_{сид} = 5,4$ кН/м; $F_{тр} = 45...50$ Н) и экспериментального сиденья с пружинной подвеской $C_{сид} = 5,4$ кН/м; $F_{тр} = 20...25$ Н, на которые поочередно устанавливались одни и те же амортизатор ВАЗ и подушка, изготовленная из поропласта толщиной 75 мм. Результаты испытаний приведены на рисунках 4.29, 4.30. Как видно из рисунков, уменьшение величины "сухого" трения с $F_{тр} = 45...50$ Н до $F_{тр} = 20...25$ Н приводит к значительному улучшению виброзащитных свойств поддрессоренного сиденья (в среднем на 21,5; 44,6; 37,3% соответственно в октавах 4; 8; 16 Гц).

В октаве 2 Гц влияние изменения величины "сухого" трения на коэффициент виброизоляции сиденья практически незаметно.

Анализ полученных результатов испытаний поддрессоренных сидений на автопоезде МАЗ-5432+9397 (рисунок 4.30) также показывает, что уменьшение "сухого" трения в подвеске сиденья до 20...25 Н позволяет обеспечить выполнение требований ГОСТ-12.1.012 по ограничению уровня вибраций на рабочем месте водителя в октавах 4; 8; 16 Гц на асфальтированном шоссе общего пользования.

4.7.4 Влияние подушки на виброзащитные свойства поддрессоренного сиденья

Экспериментальное исследование влияния подушки на виброзащитные свойства сиденья проводилось на автопоездах МАЗ-5432-9397 (асфальт, $v_a = 70$ км/ч) и МАЗ-54331-9571 (асфальт, $v_a = 70$ км/ч, булыжник $v_a = 50$ км/ч).

Исследованию на автопоезде МАЗ-5432-9397 подвергалось поддрессоренное сиденье 504В-6800010-02 с газонаполненным амортизатором КамАЗ, на которое поочередно устанавливались следующие подушки:

- подушка № 1 (материал - поропласт толщиной 75 мм);
- подушка № 2 (материал - губчатая латексная резина 1-й категории твердости толщиной 55 мм);
- подушка № 3 (материал - губчатая латексная резина 1-й категории твердости толщиной 75 мм);
- подушка № 4 (материал - губчатая латексная резина 2-й категории твердости толщиной 75 мм);
- подушка № 5 (материал - поропласт толщиной 90 мм).

Замеры ускорений на сиденье проводились на стальном диске в соответствии с ГОСТ 16526-70.

Результаты испытаний, приведенные на рисунке 4.31, показывают, что в октавах 2; 4 Гц коэффициенты виброизоляции сидений с испытанными подушками практически одинаковы.

Влияние подушки сказывается в октавах 8; 16 Гц, где лучшими виброзащитными свойствами обладает сиденье с подушкой № 3, худшими – сиденье с подушкой № 2. Различие в коэффициентах виброизоляции указанных сидений в октавах 8; 16 Гц составляет соответственно 36,8 и 49,1%.

Сиденье с подушкой № 4 по виброзащитным свойствам занимает промежуточное положение по сравнению с сиденьями с подушками № 2 и 3.

По сравнению с подушкой № 4 подушка № 3 обеспечивает сиденью в октавах 8; 16 Гц в 1,4; 1,38 раза соответственно меньшие коэффициенты виброизоляции.

Сиденье с подушкой № 5 (поропласт $h = 90$ мм) имеет несколько лучшие виброзащитные свойства, чем сиденье с подушками № 1, 2, 4. Однако в целом

полученные данные свидетельствуют, что поропласт при одинаковой толщине прокладки обеспечивает сиденью худшие виброзащитные свойства, чем губчатая латексная резина 1-й категории твердости, которая является более оптимальной для подрессоренного сиденья.

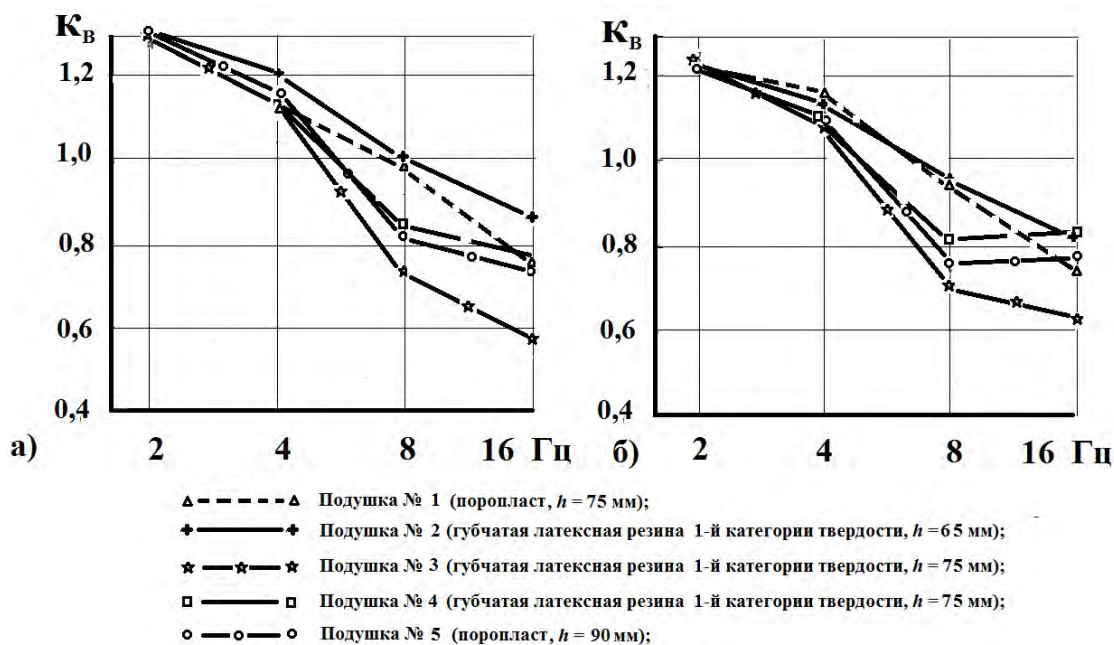


Рисунок. 4.31 – Влияние подушки на виброзащитные свойства сиденья в октавных полосах частот при движении автопоезда МАЗ-5432-9397 по асфальтированному шоссе со скоростью $v_a = 50$ км/ч (а) и $v_a = 70$ км/ч (б)

Исходя из этого, дальнейшие исследования проводились только с подушками, имеющими прокладки из губчатой латексной резины 1-й категории твердости. Дальнейшим испытаниям на автопоезде МАЗ-54331-9571 подвергались 4 подушки, имеющие прокладки толщиной 50; 65; 75; 90 мм, которые поочередно устанавливались на сиденье 504В-6800010-02 ($C_{сид} = 4,3$ кН/м, $F_{тр} = 50$ Н) с опытным амортизатором с двумя выходами штока (регулировка А2).

Анализ результатов испытаний, приведенных на рисунке 4.32, показал следующее:

- в октавах 2; 4 Гц изменение толщины прокладки подушки с 50 до 90 мм практически мало сказывается на изменении уровня вибраций на сиденье;

- в октаве 8 Гц при увеличении толщины прокладки с 50 до 90 мм происходит небольшое уменьшение уровня вибрации на сиденье;

- в октаве 16 Гц увеличение толщины прокладки с 50 до 90 мм приводит к более значительному снижению вертикальных ускорений на сиденье (на булыжном шоссе с 1,36 до 1,05 м/с²).

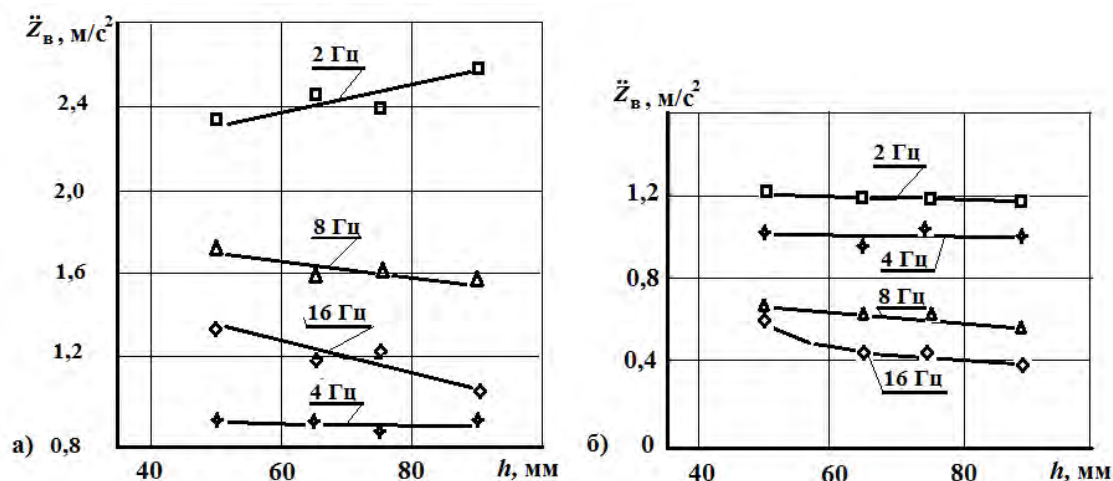


Рисунок 4.32 – Изменение вертикальных среднеквадратических ускорений на сиденье в зависимости от толщины прокладки подушки при движении по булыжному (а) $v_a = 50$ км/ч и асфальтированному шоссе (б) $v_a = 70$ км/ч

Полученные результаты испытаний свидетельствуют, что подушка с толщиной прокладки $h = 90$ мм обеспечивает наименьший уровень ускорений в октавах 8; 16 Гц по сравнению с другими испытанными подушками, что подтверждает выводы расчетного исследования. Кроме того, она субъективно оценивалась водителем, участвовавшим в испытаниях, как наиболее комфортабельная.

4.7.5. Оценка виброзащитных свойств модернизированного сиденья и вибронегруженности рабочего места водителя автомобиля МА3-5336

Для оценки эффективности предложенных в разделе 3.4 мероприятий были проведены дополнительные испытания модернизированного сиденья с торсионно-пружинной подвеской на автомобиле МА3-5336 (4×2), оборудованного передними рессорами 5432-2902012 уменьшенной жесткости ($C_p = 192$ кН/м). Указанное сиденье имело следующие параметры: $C_{сид} = 5,4$ кН/м; $F_{тр} = 22$ Н; $P_{сж} = 760$ Н; $P_{отб} = 960$ Н при $\dot{\Delta} = 0,4$ м/с; $C_{сид} = 44$ кН/м; $k_{зат} = 537$ Н·с/м, которые, за исключением $C_{сид}$, соответствовали рекомендуемым расчетным. Автомобиль МА3-5336 испытывался на тех же участках дорог, что и его предыдущий вариант. Результаты испытаний приведены на рисунке 4.33. Как видно из приведенных графиков, при движении автомобиля МА3-5336 по асфальтированному шоссе $v_a = 40...70$ км/ч уровень вертикальных ускорений на указанном сиденье не превышает предельно-допустимых значений ГОСТ 12.1.012. Только на булыжном шоссе при $v_a > 30$ км/ч превышаются требования ГОСТ: на 37,5%

в октаве 2 Гц при $v_a = 50$ км/ч и на 50,3; 23,5% в октавах 2; 8 Гц при $v_a = 60$ км/ч. Однако при этом указанный уровень ускорений на сиденье не превышает нормативных значений норм ИСО-2631 для четырехчасового воздействия вибрации. Продольные ускорения на сиденье на всех испытанных скоростях движения по булыжному шоссе ($v_a = 20\text{--}60$ км/ч) не превышают предельных значений по ГОСТ 12.1.012.

Анализ приведенных на рисунке 4.28 данных показывает, что модернизированное сиденье при движении по асфальтированному шоссе ($v_a = 70$ км/ч) обеспечивает ослабление вибраций в октавах 4; 8; 16; 31,5; 63 Гц соответственно в 1,765; 2,122; 2,73; 6,85; 20,04 раза. Лишь только в октавах 1; 2 Гц наблюдается небольшое усиление вибраций соответственно в 1,2; 1,18 раза, что связано с неоптимальным значением жесткости подвески у испытанного образца сиденья ($C_{\text{сид}} = 5,4$ кН/м вместо рекомендуемого значения $C_{\text{сид}} = 4$ кН/м).

В целом полученные данные свидетельствуют о высокой эффективности указанного подрессоренного сиденья, правильности выбора его параметров, а также некоторой возможности дальнейшего его улучшения за счет уменьшения жесткости подвески до 4 кН/м и подтверждают теоретические выводы.

На рисунке 4.34 представлено сравнение результатов моделирования и эксперимента с газонаполненным амортизатором.

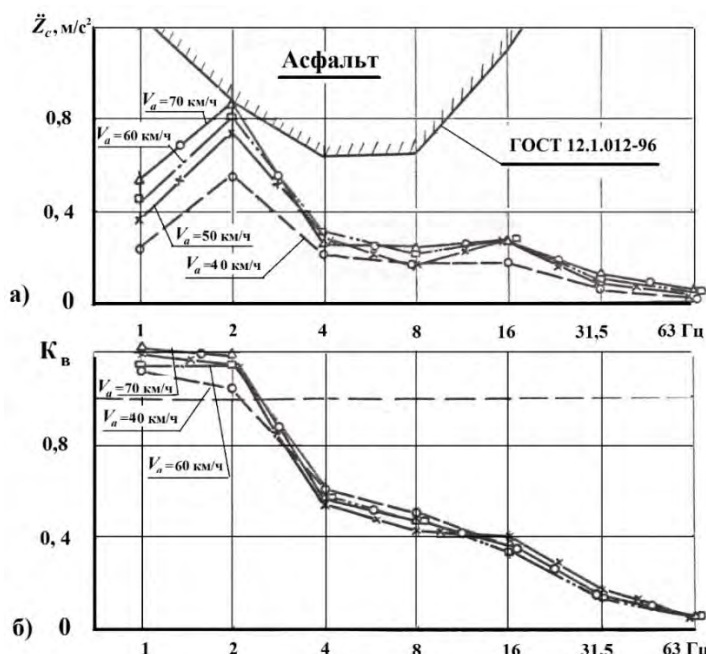


Рисунок 4.33. – Уровень вибраций и эффективность сиденья K_v на автомобиле МАЗ-5336 с предлагаемыми решениями и параметрами сиденья

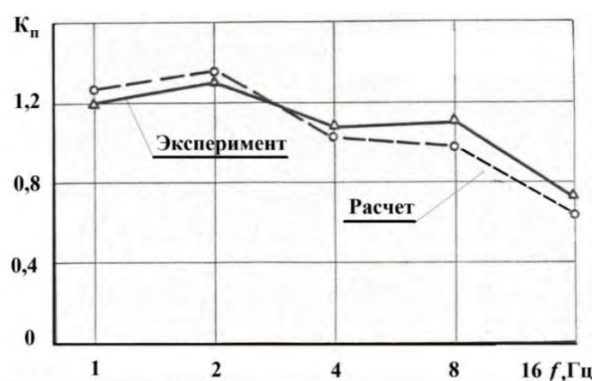


Рисунок 4.34. –Сравнение результатов моделирования и эксперимента с газонаполненным амортизатором

4.8 Исследования поперечной устойчивости штабной машины

Подтверждением корректности результатов моделирования и расчетов показателей поперечной устойчивости являются результаты экспериментального исследования штабной машины МШ 6317-ПО-КШ, проведенные в Испытательном центре ГУ «Белорусская МИС» (рисунок 4.35а) и НИИЦ АТ 3 ЦНИИ МИНОБОРОНЫ РФ (рисунок 4.35б).

Штабная машина МШ 6317-ПО-КШ имеет близкие параметры с рассчитанными машинами: на шасси МАЗ-6317 аналогичные параметры подвески и близкие параметры масс и моментов инерции кузова-контейнера.

Испытания на поперечную статическую устойчивость проводились в соответствии с п. 2.2.4 ГОСТ 12.2.000. Масса автомобиля составляла 19016 кг;

Результаты испытаний приведены в таблице 4.11. Там же для сравнения даны расчетные значения, полученные в пакете MS.ADAMS (модель описана в п.3.6, рисунки 3.20, 3.21) и с использованием формулы 3.12 п.3.6.2.

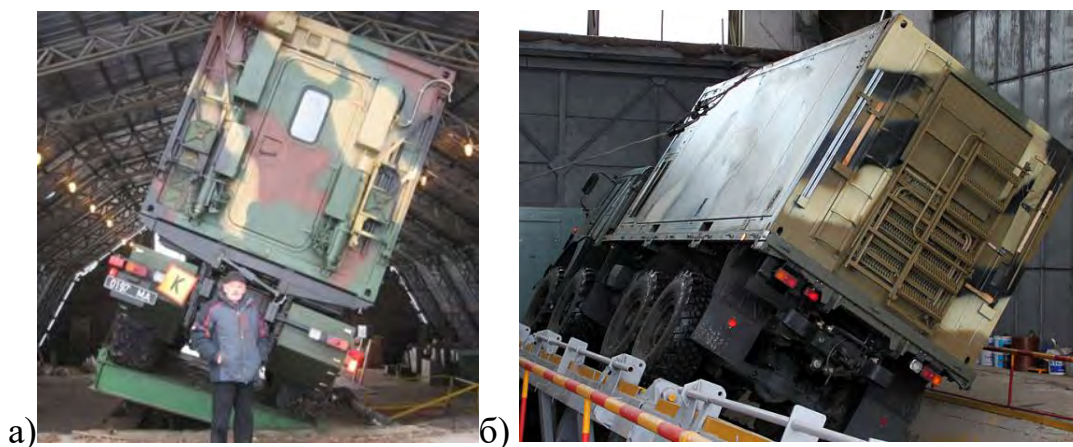


Рисунок 4.35. – Испытания штабной машины на статическую поперечную устойчивость

Таблица 4.11 – Сравнение результатов испытаний и расчетов

Предельный угол статической устойчивости				
Испытания	Расчет в ADAMS	Расхождение	Расчет по формуле (2.14)	Расхождение
28°	32,7°	14%	29,6°	6,7%

Как видно из таблицы в случае применения пакета MSC.ADAMS расхождение полученных экспериментальных и расчетных (п. 2.3.2) данных составляет 14 %, что можно считать приемлемым. При использовании предложенной формулы (2.50) предельный угол статической устойчивости при массе контейнера и части рамы равной 6500 кг имеет расхождение с экспериментом 6,7 %.

Это подтверждает корректность разработанных моделей и получаемых по ним расчетных результатов.

4.9 О лабораторном варианте имитационного моделирования

Основным фактором, препятствующим широкому внедрению имитационного моделирования, является высокая стоимость симулятора (≈ 20 мил. евро) из-за использования сложной перемещаемой на рельсах платформы с пульсаторами.

Как показал проведенный анализ можно значительно удешевить систему, если отказаться от перемещения по рельсам платформы и использовать предлагаемые на рынке 6-DoF платформы с электроактуаторами с реализацией схем на рисунке 4.36.



Рисунок. 4.36. – Предлагаемые на рынке конструкции вибростендов на базе электросимуляторов

Стоимость установки из 6-ти китайских электроактуаторов составляет \$7,800. Плюс экран и проекторы 2–3 тыс. \$. Однако их скорость перемещения всего 0,5–0,7 м/с и меньше чем у электроактуаторов Parker и MooG (1,5 м/с).

ВЫВОДЫ ПО ГЛАВЕ 4

1. Рассмотрено применение микропроцессорных и мини-компьютерных систем для экспериментальных исследований, разработан и создан программно-технический комплекс на базе мини-компьютера Raspberry Pi 3, позволяющий вести регистрацию вибронагруженности автомобиля и записывать макро и микропрофиль дороги.

2. Предложены новые методы получения массивов и характеристик дорог с помощью микропроцессоров Arduino и Raspberry на основе регистрируемых данных ускорений и наклонов на кузове и над задней осью автомобиля и их математического преобразования в пакете Simulink в высоту неровности, приведены их программы. Дана оценка методов интерполяции дорог по точкам.

Показано, что для получения промежуточных точек дороги более оптимальным является использование метода интерполяции Catmull-Rom.

3. Получены и исследованы характеристики пяти типов автомобильных дорог и дана оценка ровности и качества этих дорог. Предложено характеризовать состояние продольного профиля дороги по уклонам: до 0,15 м/с – дороги с малыми уклонами, до 0,25 м/с – небольшими уклонами, до 0,3 м/с – со средними, свыше 0,4 – большими, а качество микропрофиля дорог оценивать по спектру среднеквадратической скорости микропрофиля в третьоктавных полосах 6,3–12,6 Гц: $\dot{q}$ до 4 см/с – дороги отличного качества, до 6 см/с – дороги хорошего качества, до 8 см/с – дороги удовлетворительного хорошего качества, свыше 10 см/с – плохие либо по более широкой полосе 6,3–22,4 Гц с распределением значений градаций 12,5; 18; 30; 38 см/с для этих же дорог.

4. На основании исследованных характеристик подушек сидений, трения в подвеске и полученных их зависимостей от толщины прокладки подушки выявлено, что для сиденья более оптимальным является использование губчатой резины 1-й категории твердости толщиной 75–90 мм.

5. Выявлено влияние “сухого” трения на появление второго резонансного пика на амплитудно-частотной характеристики, объясняющего ухудшение виброзащитных свойств сиденья. Предложена математическая модель трения, корректно отражающая процесс колебаний водителя на сиденье.

6. На основе проведенных экспериментальных исследований основных элементов подвески: рессор, амортизаторов, сиденья, выявлены их зависимости и предложены более корректные их динамические модели, включая тип амортизатора, “сухого” трения, шины и автомобиля в целом,

которые обеспечивают хорошую сходимость с экспериментом с учетом спектра (погрешность 10–15 %).

6. Проведено всестороннее исследование вибронагруженности, устойчивости, устойчивости, дана оценка эффективности систем поддрессирования автомобиля (основной подвески, поддрессирования кабины и сиденья водителя, влияние типа и характеристики амортизатора).

7. Выявлено влияние типа амортизатора, показавшее необходимость использования для сиденья специального амортизатора с чисто вязкостной зависимостью сопротивления от скорости. Газонаполненный амортизатор КАМАЗ работает как дополнительный упругий элемент, увеличивая колебания и его применение нежелательно.

8. Рассмотрены практические реализации систем имитационного моделирования на симуляторах на основе 6-DoF платформы с электроактуаторами, обеспечивающие значительное удешевление системы путем отказа от перемещения по рельсам платформы и использования, имеющихся на рынке 6-DoF платформ, с электроактуаторами для создания вибростенда кабины.

ГЛАВА 5

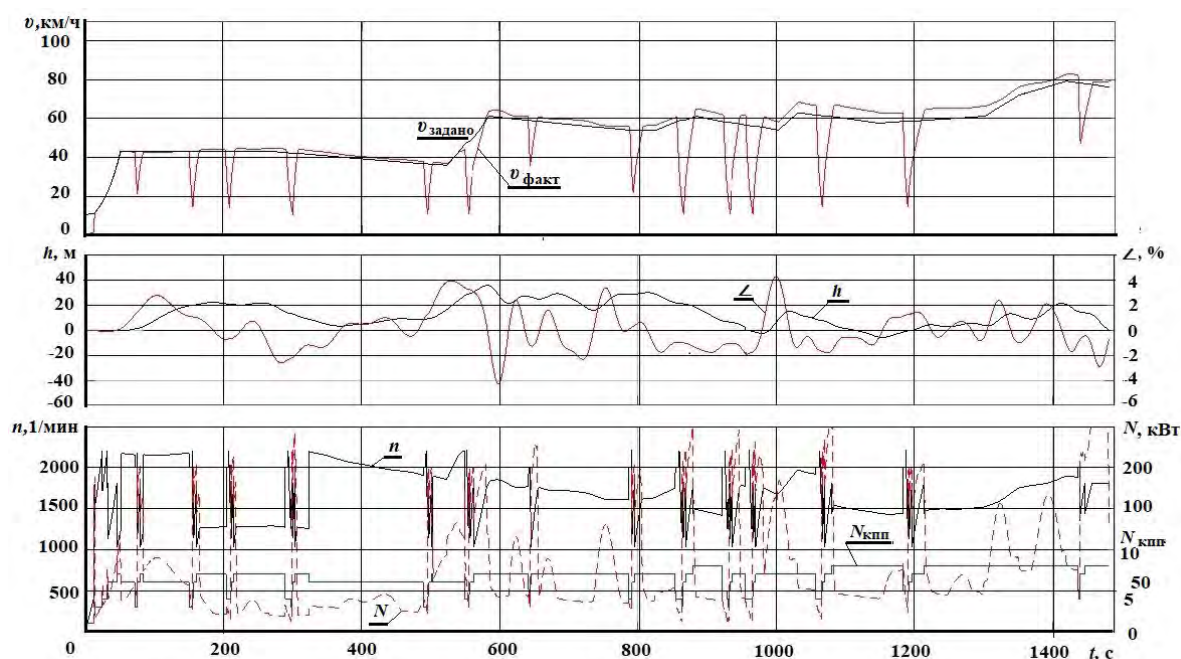
МОДЕЛИРОВАНИЕ ДВИЖЕНИЯ, КОЛЕБАНИЙ И ДОЛГОВЕЧНОСТИ КОЛЕСНЫХ МАШИН

В данной главе рассматриваются результаты моделирования движения и колебаний автомобиля на основе полученных данных с помощью GPS макропрофилей дорог, режимов движения и способа учета торможений перед препятствиями, а также влияние микропрофилей дорог на скорость движения и его виброн нагруженность [65, 66]].

5.1 Моделирование движения колесной машины

5.1.1 Моделирование движения колесной машины в городских условиях

На рисунке 5.1. показано движение колесной машины при $m_a = 20000$ кг в городских условиях.



а – скорость движения; б – высота, наклон дороги; в – обороты двигателя, номер передачи

**Рисунок 5.1. – Движение колесной машины при $m_a = 20000$ кг
в городских условиях (2 полосы)**

На участке длиной 22090 м средняя скорость составляет $v_a = 51.93$ км/ч, расход $Q=41,17$ л/100 км, средний наклон дороги – 1,24 %, средняя мощность

и число оборотов двигателя соответственно 80,28 кВт и 1674 1/мин. Имеет место 11 торможений, возможны еще 4 торможения на перекрестках. Максимальный угол наклона достигает 4 %. Обороты двигателя варьируются от 1000 до 2200 1/мин. При движении используются передачи КПП от 3-ей до 8-й. Каждый переезд препятствий вынуждает переходить на более низшую передачу. Максимальная скорость на выезде из города достигла 82 км/ч. Как видно из рисунка ограничивают скорость препятствия, остановки движения.

На рисунке 5.2. представлено движение колесной машины при $m_a = 25000$ кг в городских условиях.

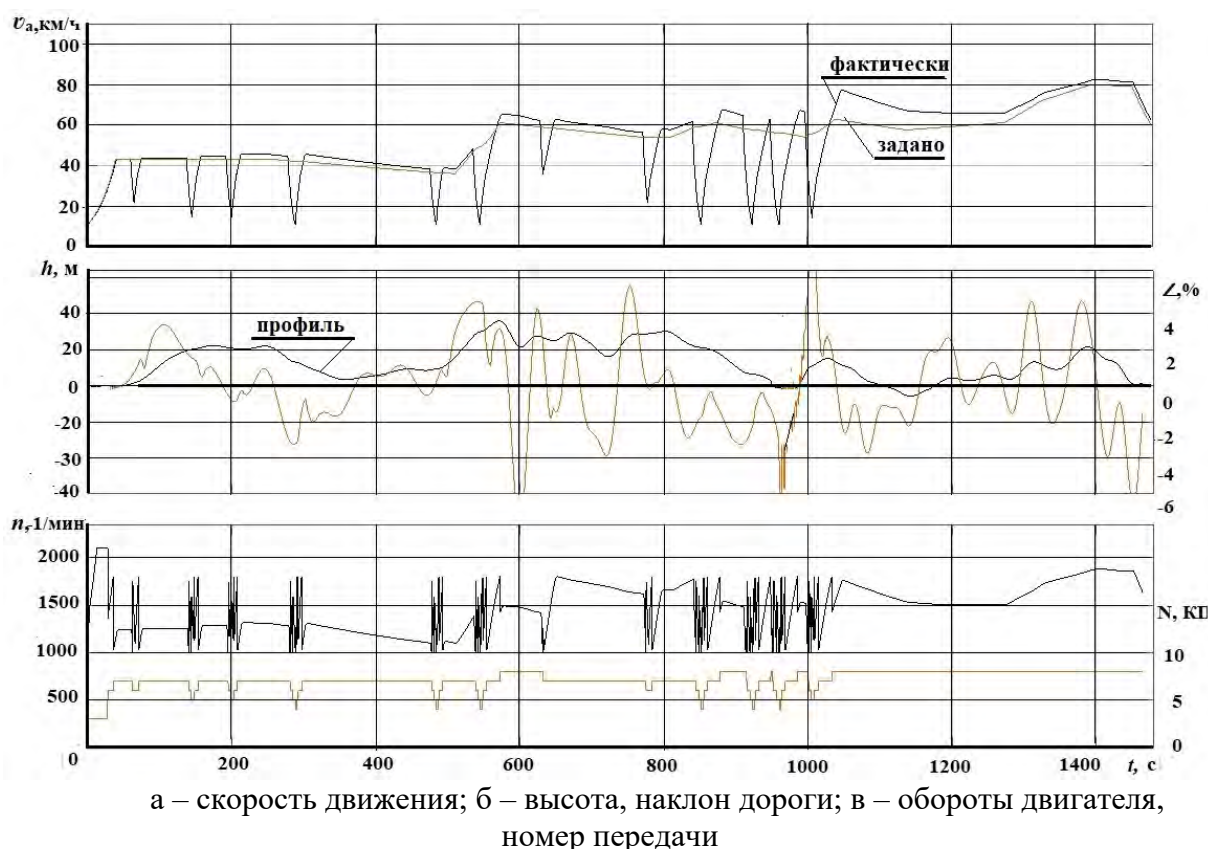
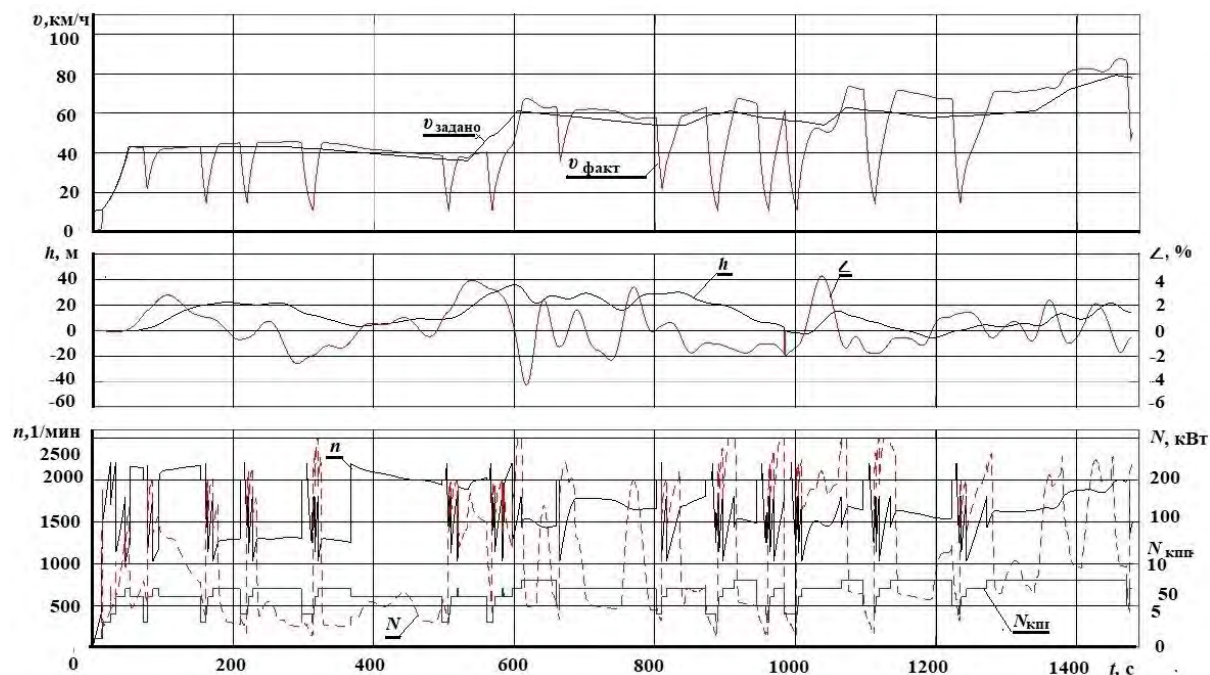


Рисунок 5.2. – Движение колесной машины при $m_a = 25000$ кг в городских условиях (2 полосы)

Средняя скорость при $m_a = 25000$ кг составляет $v_a = 53,2$ км/ч, расход $Q=76,8$ л/100 км, средний наклон дороги – 1,84 %, средняя мощность и число оборотов двигателя соответственно 0,275 кВт и 1472 1/мин. Имеет место 11 торможений, возможны еще 4 торможения на перекрестках. Максимальный угол наклона достигает 6–7 %. Обороты двигателя варьируются от 1000 до 1850 1/мин. При движении используются передачи КПП от 4 до 7-й. При каждом переезде препятствий происходит переход на более низшую передачу. Максимальная скорость достигает 80,5 км/ч. Как видно из рисунка 5.2 ограничивают скорость препятствия, остановки движения.

На рисунке 5.3. представлено движение колесной машины при $m_a = 32000$ кг в городских условиях.



а – скорость движения; б – высота, наклон дороги; в – обороты двигателя, номер передачи

Рисунок 5.3. – Движение колесной машины при $m_a = 32000$ кг в городских условиях (2 полосы)

На участке длиной м при $m_a = 32000$ кг средняя скорость составляет $v_a = 50.03$ км/ч, расход $Q=54,3$ л/100 км, средний наклон дороги – 1,24 %, средняя мощность и число оборотов двигателя соответственно 108,3 кВт и 1660 1/мин. Имеет место 11 торможений, также возможны еще 4 торможения на перекрестках. Максимальный угол наклона дороги 4 %. Обороты двигателя варьируются от 1000 до 2000 1/мин. При движении используются передачи КПП от 3 до 8-й. Каждый переезд препятствий вынуждает переходить на более низшую передачу. Максимальная скорость при выезде из города достигает 82 км/ч. Как видно из полученных данных и рисунка ограничивают скорость препятствия, остановки движения. При увеличении массы до 32000 кг увеличивается потребляемая средняя мощность двигателя в 1,35 раза.

5.1.2 Моделирование движения по магистральной и загородной трассах

На рисунке 5.4 представлено движение по магистральной трассе.

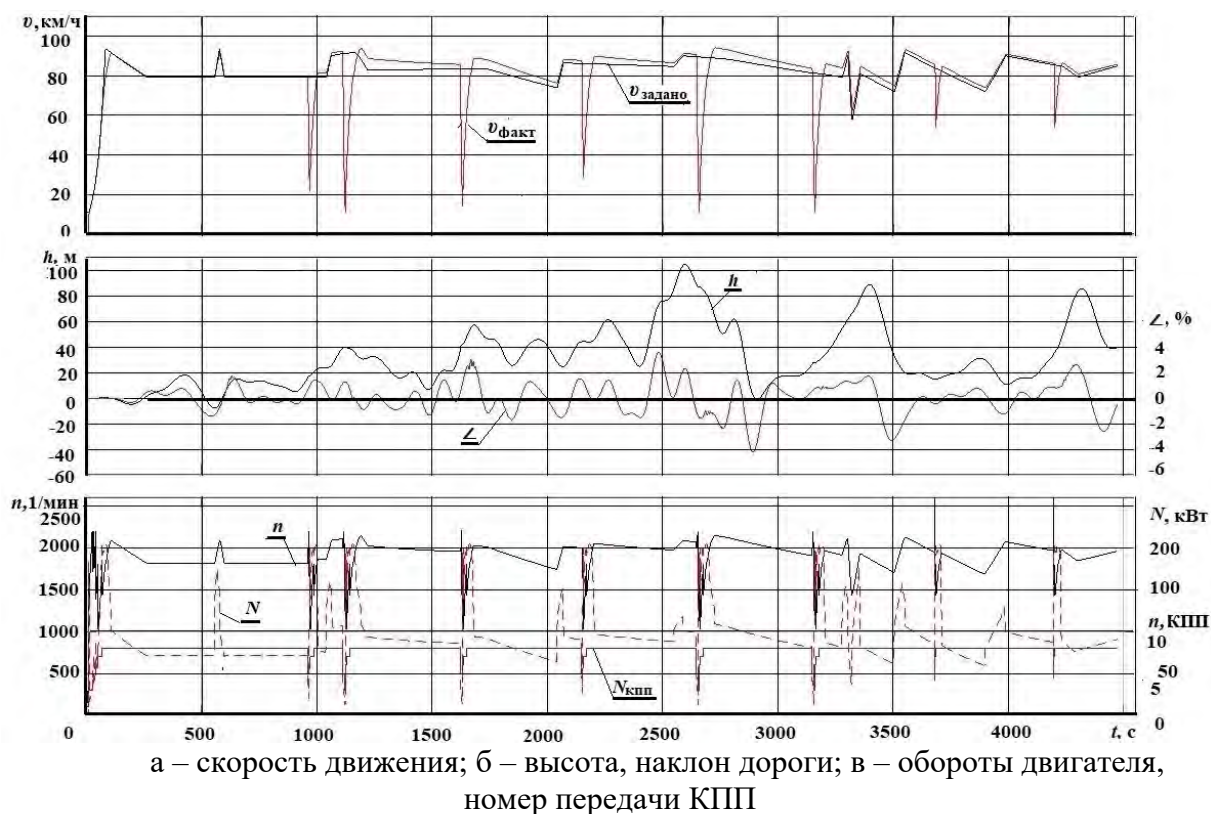
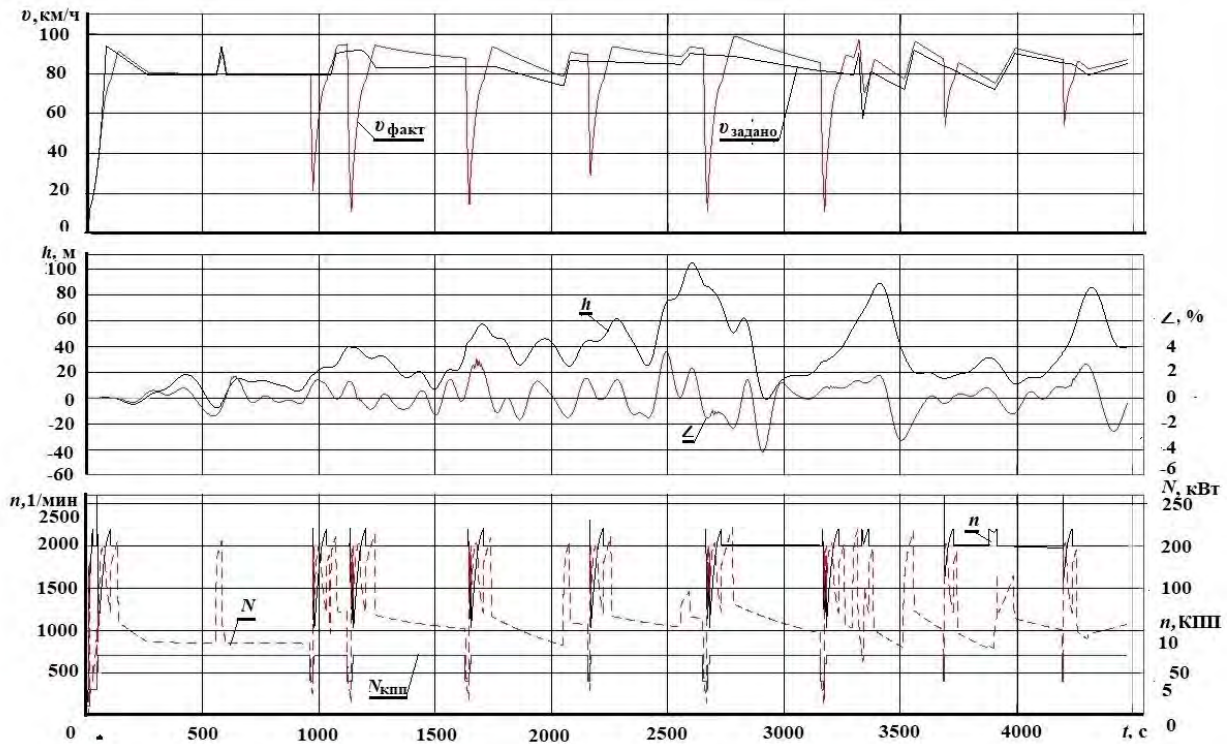


Рисунок 5.4. – Движение колесной машины при $m_a = 20000$ кг по магистральной трассе (2, 4 полосы)

Данный участок длиной 103 км примерно до половины пути характеризуется небольшим подъемом с изменением высоты до 60 м. Для второй половины характерны более существенные подъемы, спуски, на которых перепад высот достигает 100 м, максимальный угол наклона 4 %. На всем участке при $m_a = 20000$ кг средняя скорость составляет $v_a = 81.8$ км/ч, расход $Q=30,6$ л/100 км, средний наклон дороги – 0,88 %, средняя мощность и число оборотов двигателя соответственно 96,2 кВт и 1902 1/мин. Максимальная скорость достигает 95 км/ч. Обороты двигателя варьируются от 1000 до 2150 1/мин. В основном двигатель работает в диапазоне 1700 – 1900 1/мин. На трассе 103 км имеют место 3 торможения. При движении используются передачи КПП от 4 по 8-й с преобладанием 8-й. При переезде препятствий происходит переход на более низшую передачу. Как видно из данных и рисунка 5.4 препятствия, остановки движения практически не ограничивают скорость движения. При увеличении массы до 32000 кг увеличивается средняя мощность двигателя в 1,26 раза.



а – скорость движения; б – высота, наклон дороги; в – обороты двигателя, номер передачи КПП

Рисунок 5.5. – Движение колесной машины при $m_a = 32000$ кг по магистральной трассе (2, 4 полосы)

Данный участок длиной 103 км примерно до половины пути характеризуется небольшим подъемом с изменением высоты до 60 м. Для второй половины характерны более существенные подъемы, спуски, на которых перепад высот достигает 100 м, максимальный угол наклона 4 %. На всем участке при $m_a = 20000$ кг средняя скорость составляет $v_a = 81.8$ км/ч, расход $Q=36,6$ л/100 км, средний наклон дороги – 0,88 %, средняя мощность и число оборотов двигателя соответственно 116 кВт и 1972 1/мин. Максимальная скорость достигает 95 км/ч. Обороты двигателя варьируются от 1000 до 2150 1/мин. В основном двигатель работает в диапазоне 1700 – 1900 1/мин. На трассе 103 км имеют место 3 торможения. При движении используются передачи КПП от 4 по 8-й с преобладанием 8-й. При переезде препятствий происходит переход на более низшую передачу. Как видно из данных и рисунка 5.5 препятствия, остановки движения практически не ограничивают скорость движения. При увеличении массы до 32000 кг увеличивается средняя мощность двигателя в 1,26 раза.

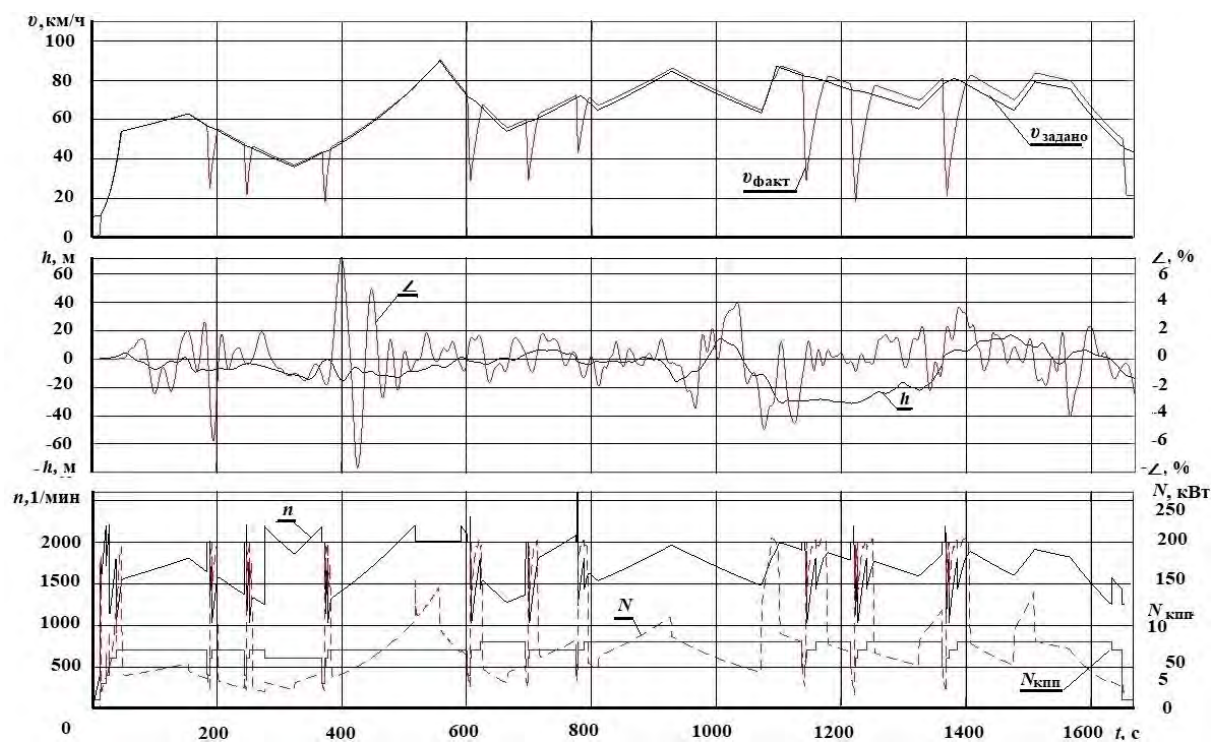


Рисунок 5.6. – Движение колесной машины при $m_a = 20000$ кг по местной 2-х полосной дороге Логойск–Смолевичи

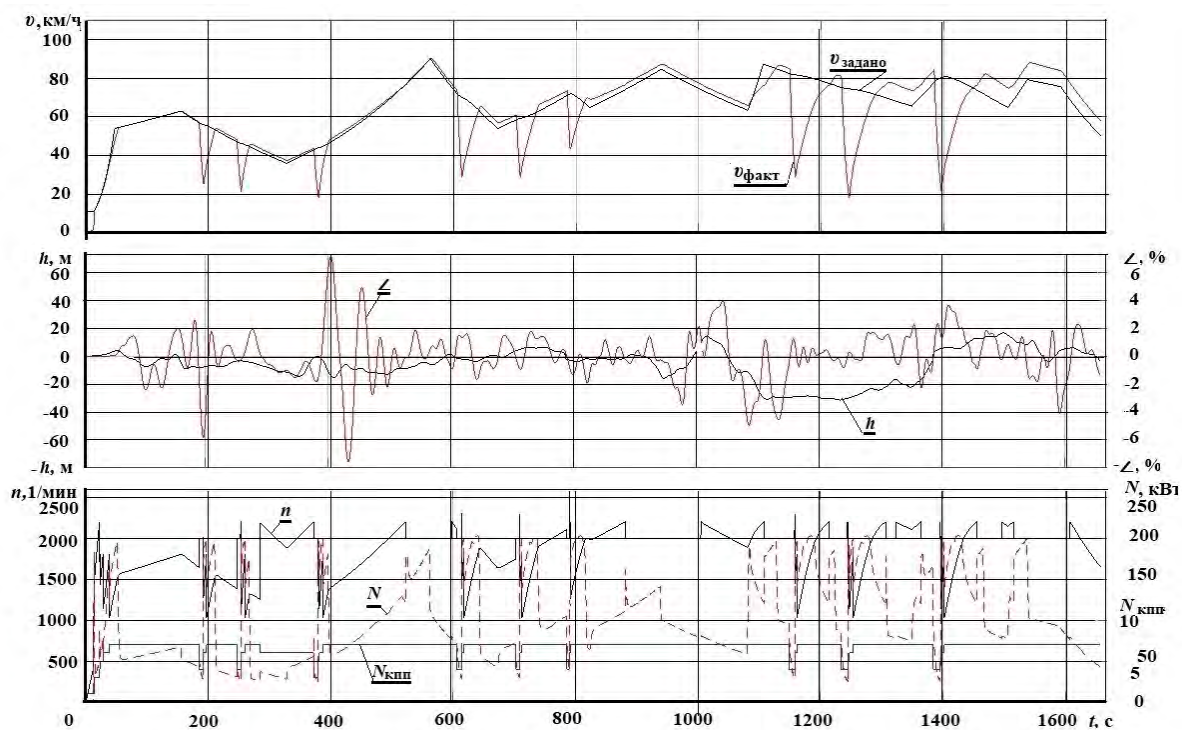


Рисунок 5.7. – Движение колесной машины при $m_a = 32000$ кг по местной 2-х полосной дороге Логойск–Смолевичи

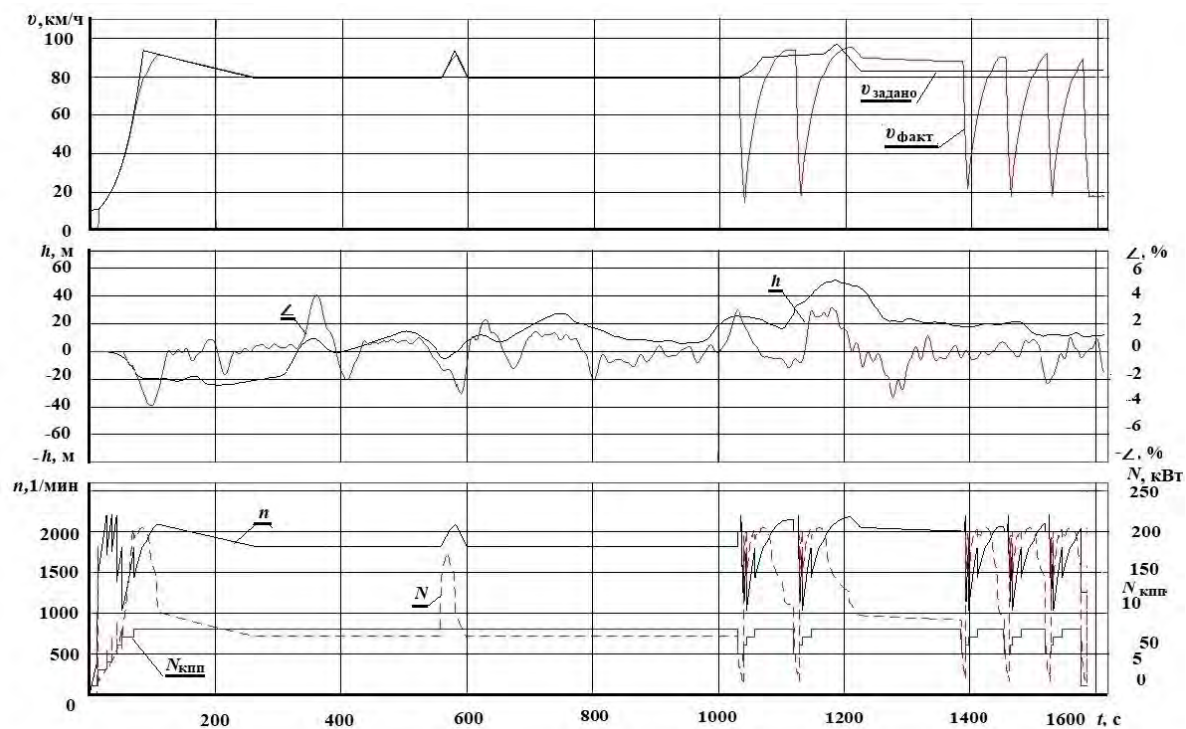


Рисунок 5.8. – Движение колесной машины при $m_a = 20000$ кг
по 4-х полосной дороге Смоленичи – Минск

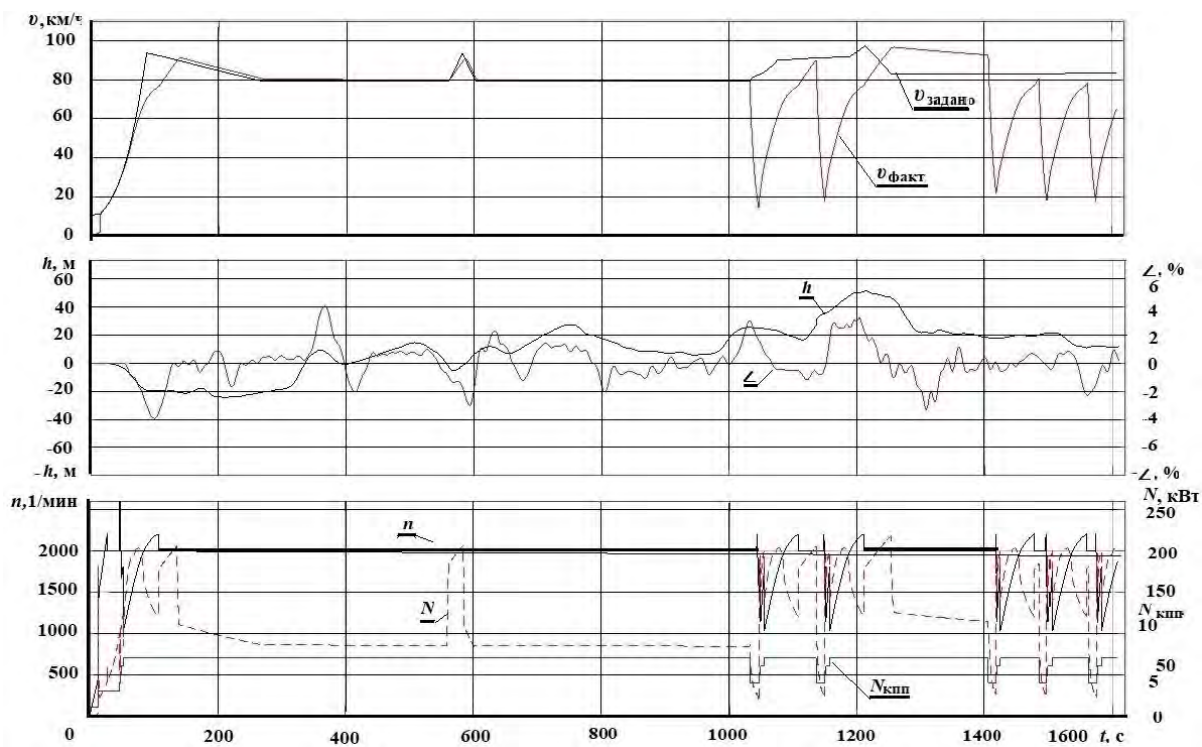


Рисунок 5.9. – Движение колесной машины при $m_a = 32000$ кг
по 4-х полосной дороге Смоленичи – Минск

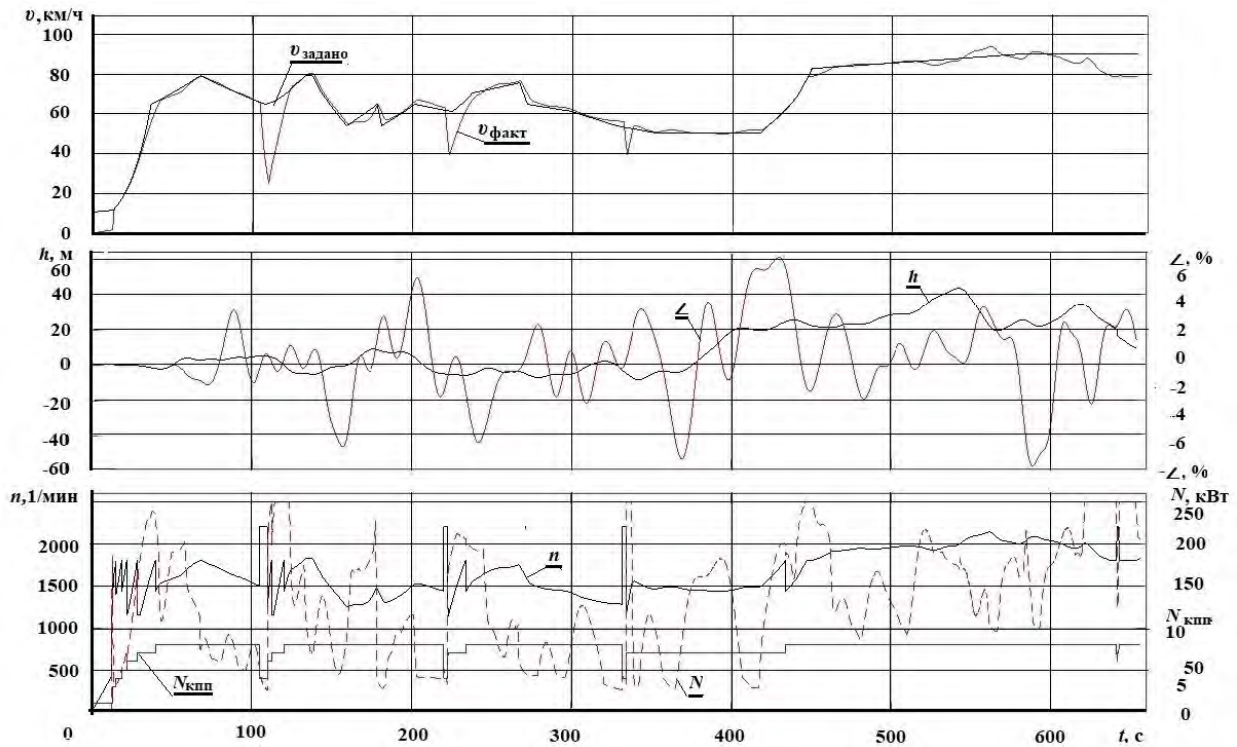


Рисунок 5.10. – Движение колесной машины при $m_a = 20000$ кг по местной 2-х полосной дороге Сев. кладбище – Семков городок – МКАД-2

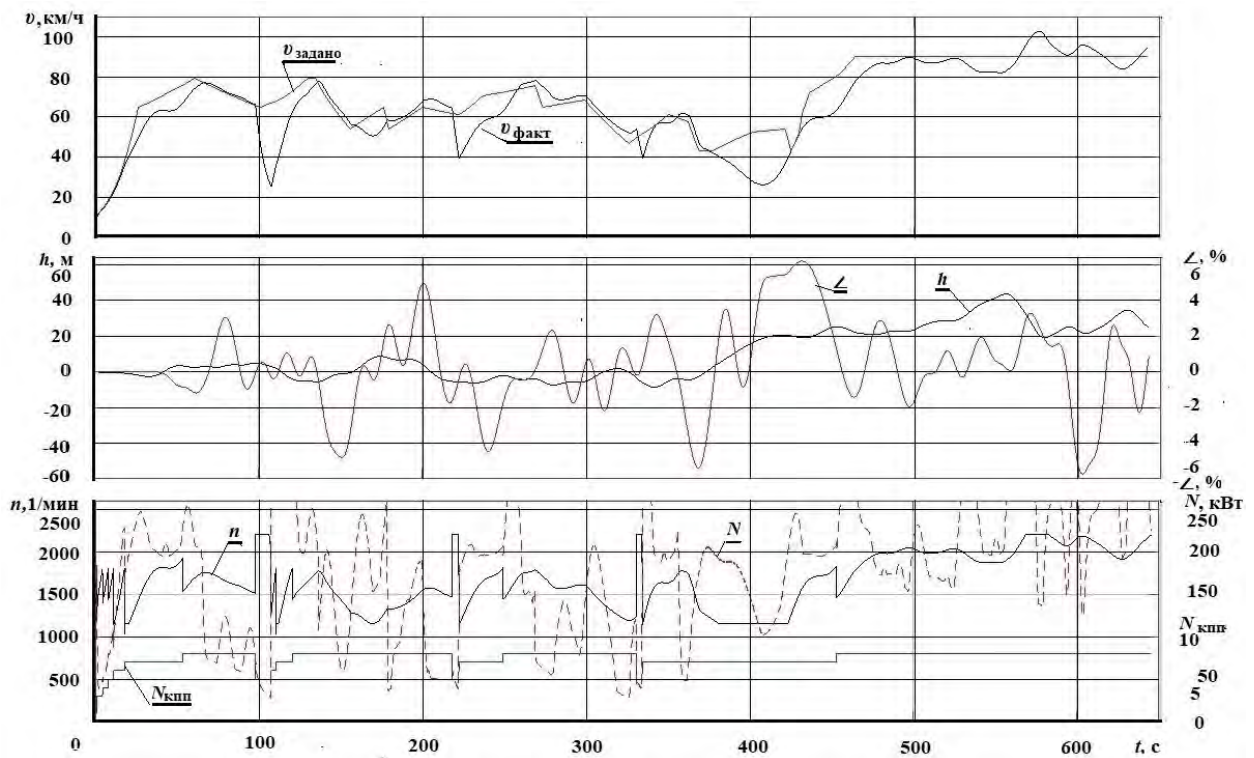


Рисунок 5.11. – Движение колесной машины при $m_a = 32000$ кг по местной 2-х полосной дороге Сев. кладбище – Семков городок – МКАД-2

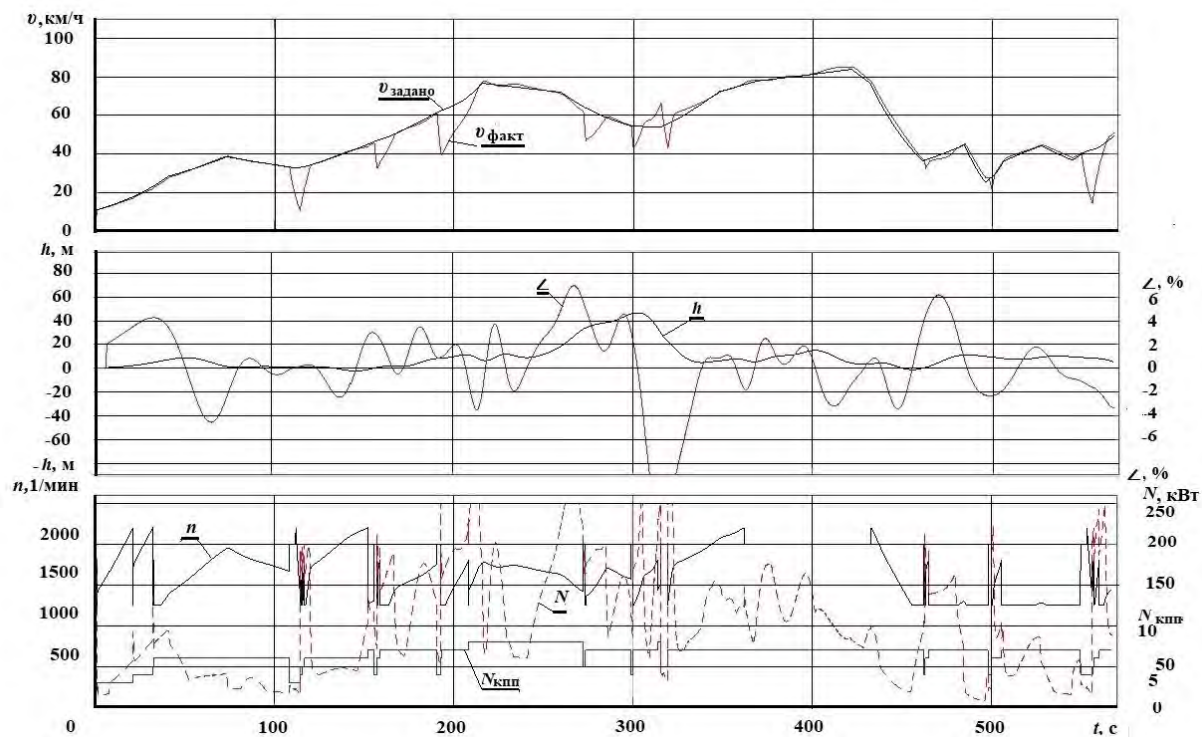


Рисунок 5.12. – Движение колесной машины при $m_a = 20000$ кг по местной 2-х полосной дороге МКАД-2 – Вишневка

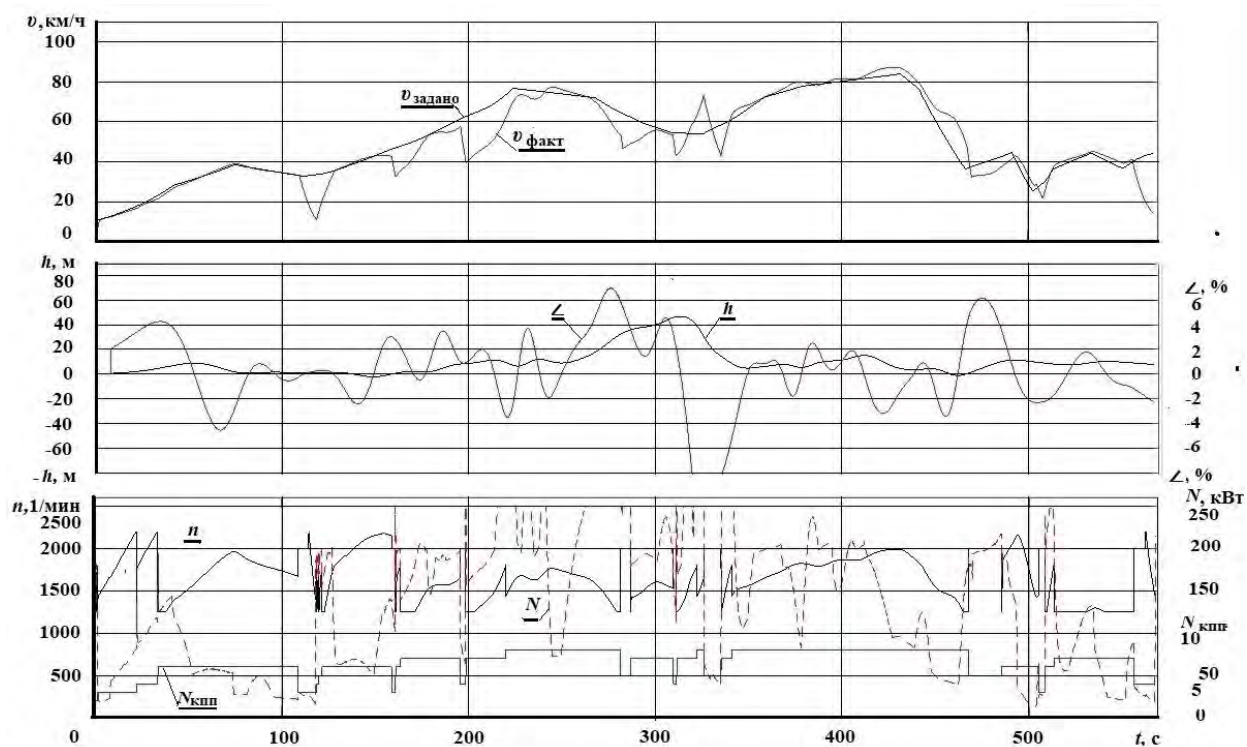


Рисунок 5.13. – Движение колесной машины при $m_a = 32000$ кг по 2-х полосной дороге МКАД-2 – Вишневка

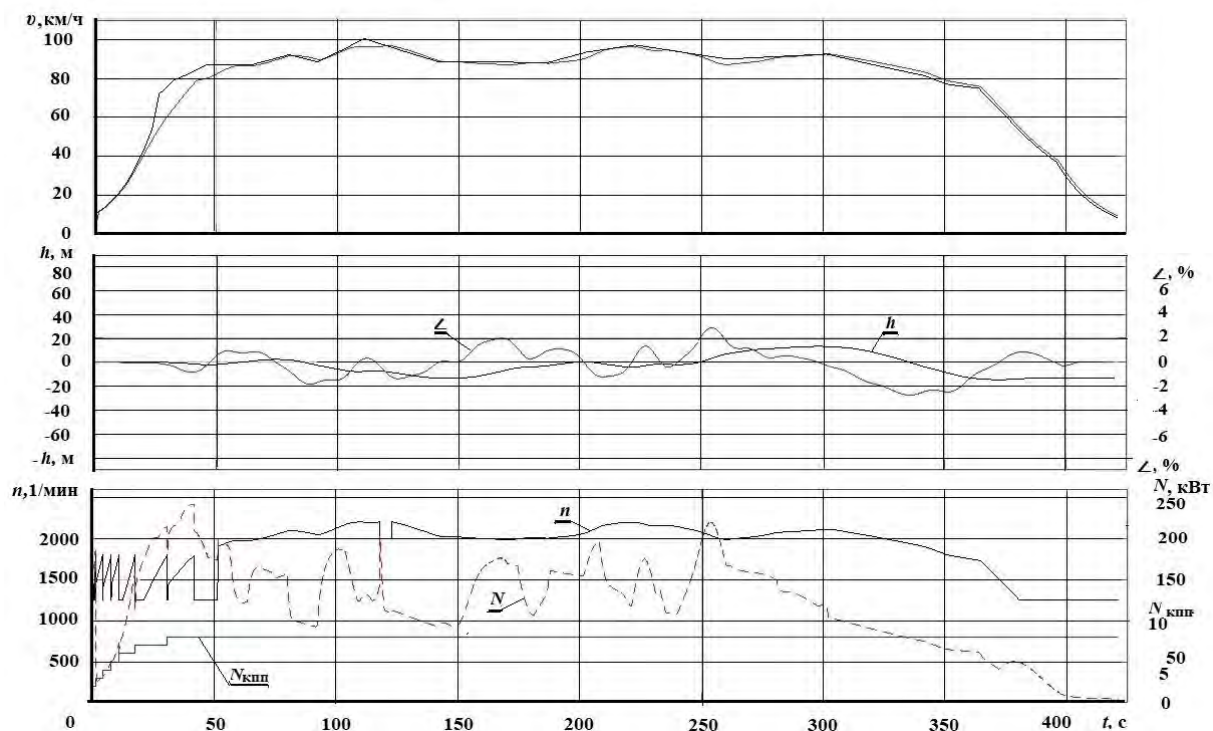


Рисунок 5.14. – Движение колесной машины при $m_a = 20000$ кг по 4-х полосной дороге Вишневка – Минск

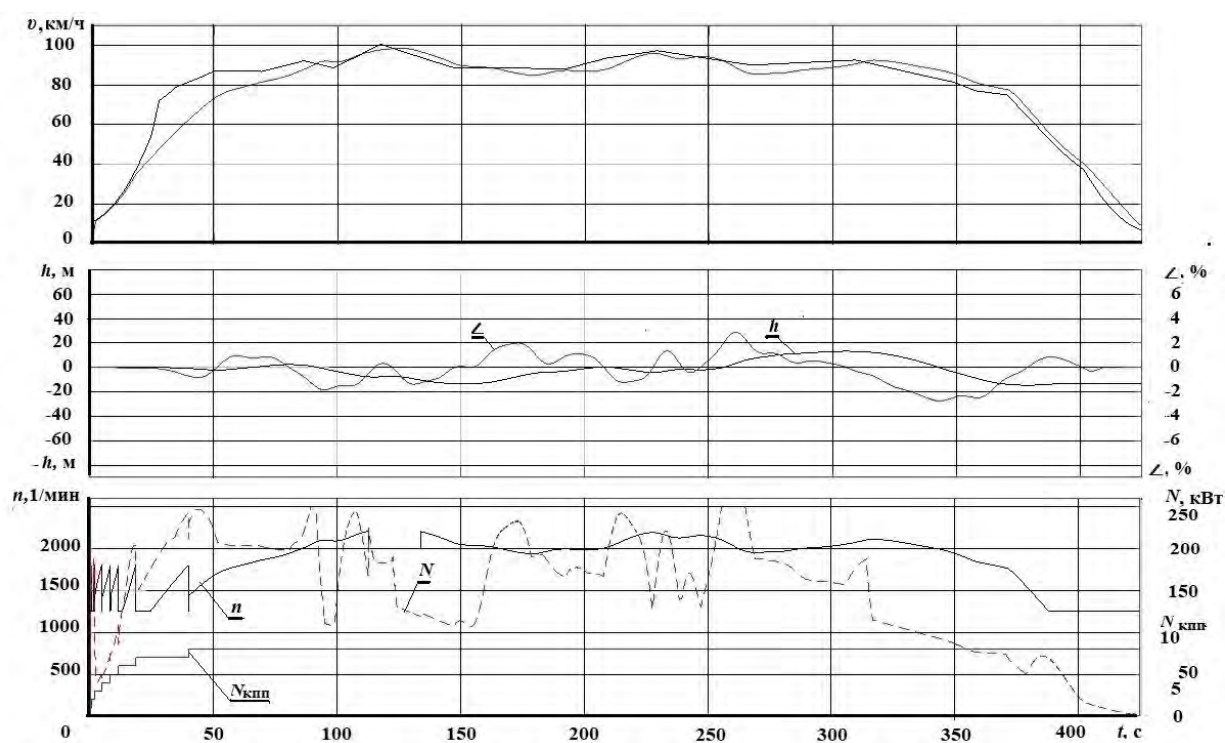


Рисунок 5.15. – Движение колесной машины при $m_a = 32000$ кг по 4-х полосной дороге Вишневка – Минск

В таблицу 5.1 сведены данные по результатам моделирования движения в городе (1), трассе (2), местной дороге (3), смешанной трасса – город (4).

Таблица 5.1. – Результаты моделирования движения в городе (1), трассе (2), местной дороге (3), смешанной трассе – город (4)

Дорога	m	$\mu\angle$ %	Max $\angle$ %	$Q_{л/100км}$	V	μN	n, 1/мин	S, м
1	20000	1,24	4,2	41,17	51.93	80,28	1674	21490
с/торм.	25000	1,25		46,64	51.27	92,68	1662	21220
	32000	1,24		54,3	50.03	108,27	1660	20710
1	20000	1,22		32,51	53.6	58,44	1822	22180
б/торм.	25000	1,22		35,87	53.54	67,05	1831	22160
	32000	1,22		40,85	53.43	79,48	1851	22120
2	20000	0,877	6	30,59	81.8	96,	1902	103400
с/торм.	25000	0,877		32,65	81.79	102,9	1908	103400
	32000	0,877		36,56	81.77	116	1972	103300
2	20000	0,877	6	27,07	81.8	82,97	1878	103300
б/торм.	25000	0,877		28,46	81.79	88,45	1879	103400
	32000	0,876		31,89	81.79	97,8	2001	103400
3	20000	1,294		32,25	63.87	77,69	1696	29630
с/торм.	25000	1,285		35.2	63.8	86,95	1689	29480
	32000	1,285		41,9	64.48	86,95	1689	29480
3	20000	1,288	7	29,16	64.48	63,0	1914	29790
б/торм.	25000	1,288		31,02	64.47	69,14	1914	29770
	32000	1,289		33,63	64.45	77,65	1913	29760
4	20000	0,903	5	33,66	75.08	97,78	1817	33790
с/торм.	25000	0,904		35,67	74.9	104,7	1815	33700
	32000	0,907		39,52	74.0	113,5	1928	33300
4	20000	0,863	5	27,46	79.81	82,2	1857	35920
б/торм.	25000	0,863		28,89	79.79	87,68	1857	35910
	32000	0,863		32,54	79.77	97,67	1976	35900

В таблицу 5.2 сведены данные по результатам моделирования движения в Минске (город 1), трассе (2), местной дороге (3), смешанной трасса–город (4).

Таблица 5.2 – Результаты моделирования движения на местных дорогах (1), трассе (2), местной дороге (3), смешанной трасса – город (4)

	m	$\mu \angle \%$	Max $\angle \%$	$Q, \text{л/100км}$	V	μN	n	S, м
1	20000	1,77	6	46,95	65.7	129.9	1642	12310
Brake	25000	1,78		55,48	65.7	152.0	1658	12040
	32000	1,8		63,2	64.15	170.1	1678	11760
1	20000	176		44,58	67.62	122,7	1632	12400
	25000	179		52,64	66.4	144,4	1696	12170
	32000	180		59,6	65.22	161,7	1721	11960
2	20000	2,3	7	49,61	50.79	99,22	1709	8042
Brake	25000	229		57,21	50.43	115,2	1709	7986
	32000	229		64,39	50.08	131,8	1680	7922
2	20000	227		41,96	51.8	85,92	1574	8202
Brake	25000	227		47,57	51.3	98,46	1575	8202
	32000	234		56,96	51.02	119,6	1583	8078
3	20000	0,873	3	38,75	77.57	118,4	1869	9157
Без	25000	0,873		42,42	77.55	131,6	1877	9155
Brake	32000	0874		47,51	77.57	150,4	1864	9157

Примечание. Дорога 1 – Сев. Кладбище – МКАД-2. Дорога 2 – МКАД-2 – н/п Вишневка, Дорога 3 – н/п Вишневка – Сев. Кладбище.

5.2 Влияние вибраций на скорость движения

Вибрации являются одним из ограничивающим скорость фактором, особенно на плохих дорогах: гравийных и грунтовых. К сожалению, работ по этой тематике, особенно по совместному моделированию движения и колебаний автомобиля, крайне мало. Одной из наиболее серьезных работ является исследование движения армейских автомобилей на грунтовых дорогах Гришкевича А.И. [57], где рассматривалось влияние вибраций на скорость движения. Исследовалось с какой скоростью двигаются армейские автомобили по гравийным/грунтовым дорогам на полигоне Бронницы МО РФ. Также моделировались колебания и движение на аналоговых машинах и проводилась оценка их параметров. Это было первая работа по совместному моделированию движения и колебаний автомобиля. Недостатком работы

является использование упрощенной 2-х массовой линейной модели колебаний, приводящей к большой погрешности, а также задание возмущения макро и микропрофиля дороги через передаточную функцию с помощью генератора белого шума, параметры фильтров, которых не менялись в зависимости от скорости. Использование передаточной функции является нонсенсом на нынешнем этапе развития компьютерной техники и современных пакетов. Но тогда это соответствовало уровню развития теории и возможностям вычислительной техники. Хотя в интернете до сих пор много работ на основе использования передаточной функции. Наиболее ценным в работе [57] является экспериментальные данные по скоростям движения армейских автомобилей на грунтовых дорогах, представляющие интерес для сравнения.

Сейчас изменился парк автомобилей, их возможности. Это обуславливается необходимостью проведения подобных исследований на качественно новом уровне. Для этого проведено исследование движения и колебаний автомобиля МАЗ-6317 с кузовом-контейнером (общая масса варьировалась 20000, 25000, 32000 кг) с использованием реального макро и микропрофиля дороги и усовершенствованных динамических моделей.

В связи отсутствием массивов данных совместного возмущения на грунтовых дорогах использовался способ формирования его на основе продольного профиля дороги, полученного с помощью GPS [55, 56] и микропрофиля гравийной дороги возле н/п Салюты под Минском [44] (путем их суммирования), замеренного автором еще в 2018 г.

Учитывался режим движения, полученный с помощью GPS, что является важным для такого моделирования.

Скорость движения варьировалась от 16 до 40 км/ч. Это тот диапазон по субъективным ощущениям вибраций, которого можно двигаться на автомобиле по грунтовой дороге.

Следует указать, что эта дорога плохого качества, разбитая грунтозацепами тракторов и сельхозтехники, на которой давно не проходил грейдер. Но она соответствует условиям, которым придется столкнуться специальным машинам на гравийных дорогах в обычной и боевой обстановке.

Дискутируемым вопросом является как проводить оценку вибраций. По ГОСТ/ISO [16] рекомендуется проводить оценку по приведенным вибрациям с учетом частоты и суммирования воздействия в трех направлениях.

$$a_v = \left[(k_x \cdot A_{wx})^2 + (k_y \cdot A_{wy})^2 + (k_z \cdot A_{wz})^2 \right]^{0.5},$$

где k_x, k_y, k_z – коэффициенты чувствительности в определенной полосе частот (в третьоктавах);

B_x, B_y, B_z – среднеквадратичные значения вибраций в плоскостях x, y, z.

На их основе согласно ГОСТ/ИСО предлагается оценивать по индексу комфорта (Comfort Index, CI):

если CI является меньше, чем 0.315 m/s^2 , дискомфорт не ощущается, оценивается как **«весьма удобно»**;

если $CI = 0.315 \text{ m/s}^2 - 0.565 \text{ m/s}^2$, легкое ощущение дискомфорта это **«мало неудобно»/“not uncomfortable”**;

если $CI = 0.565 \text{ m/s}^2 - 0.9 \text{ m/s}^2$, приемлемое ощущение дискомфорта **«довольно неудобно»/“little uncomfortable”**;

если $CI = 0.565 \text{ m/s}^2 - 0.9 \text{ m/s}^2$, это **«довольно неудобно»/“uncomfortable”**;

если $CI = 1.425 \text{ m/s}^2 - 2.25 \text{ m/s}^2$, это **«очень неудобно»/ “very uncomfortable”**

если CI больше, чем 2.25 m/s^2 , крайняя степень дискомфорта это **«чрезвычайно неудобно»/“extremely uncomfortable”**.

В связи с чем необходимо проводить оценку вибраций не только в вертикальном направлении, но и в продольном и поперечном направлении.

Из-за сложности оценки воздействия вибраций на водителя в новом ISO 2631-5: 2018 предлагается дополнительный подход: в зависимости от потери контакта и превышения пиковых ускорений более $9,8 \text{ m/s}^2$ оценивать по давлению на позвоночник и времени воздействия, при значениях меньше по скорректированным вибрациям в вертикальной плоскости с учетом частоты (рисунок 5.15).

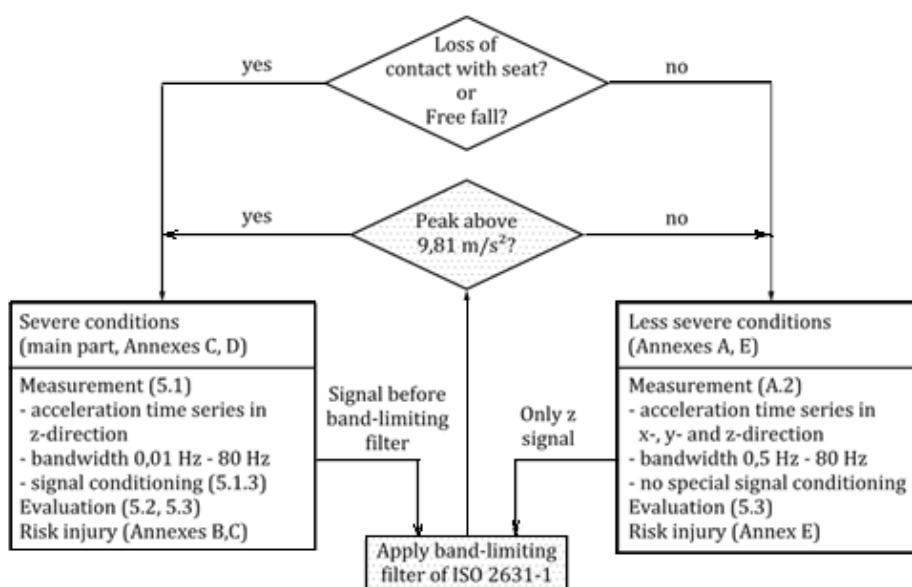


Рисунок 5.15. – Схема применения моделей

Результаты моделирования колебаний автомобиля с оценкой по приведенным среднеквадратичным значениям вертикальных вибраций показаны на рисунках 5.16–5.18 и в таблице 5.3.

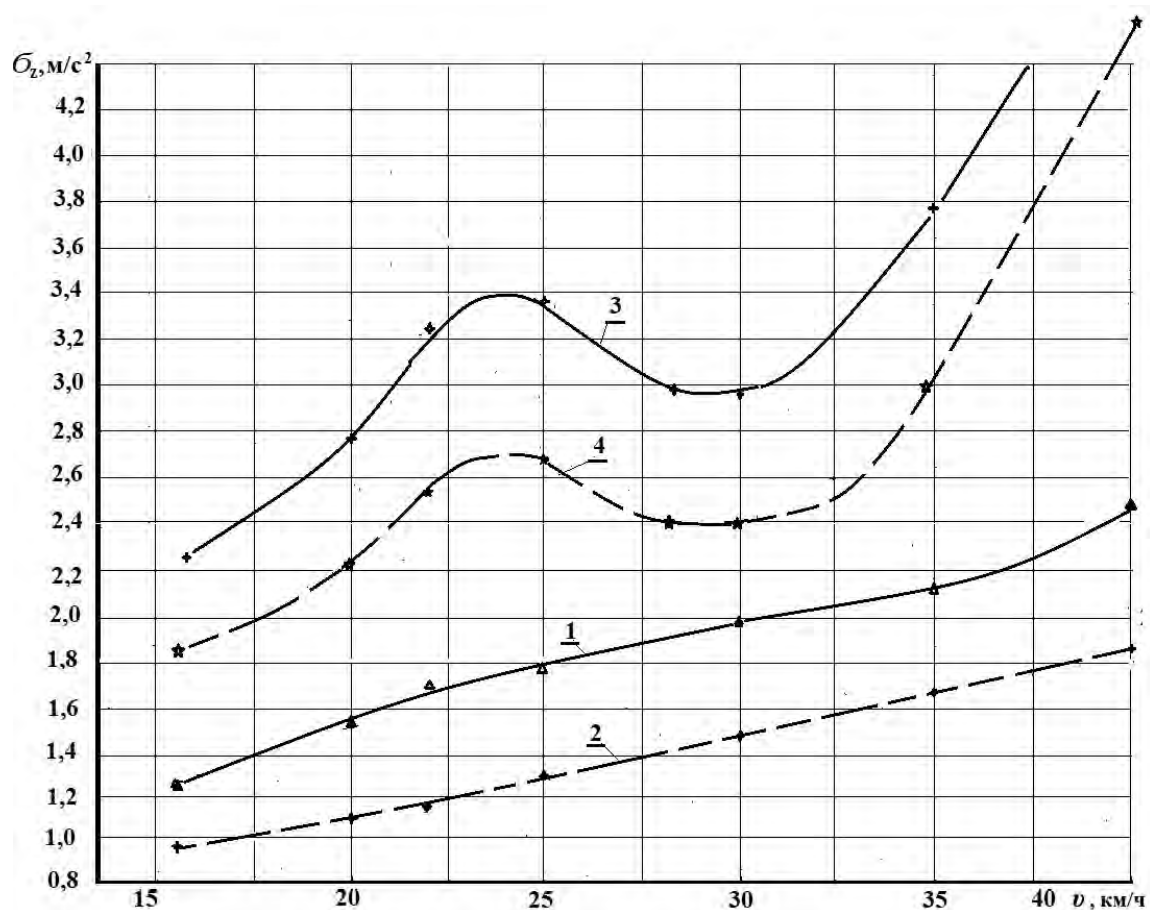


Рисунок 5.16. – Изменение непосредственных (сплошная) и скорректированных (прерывистая) вертикальных вибраций в полосе 1–25 Гц на сиденье (1, 2) и над задней подвеской (3, 4) в зависимости от скорости автомобиля

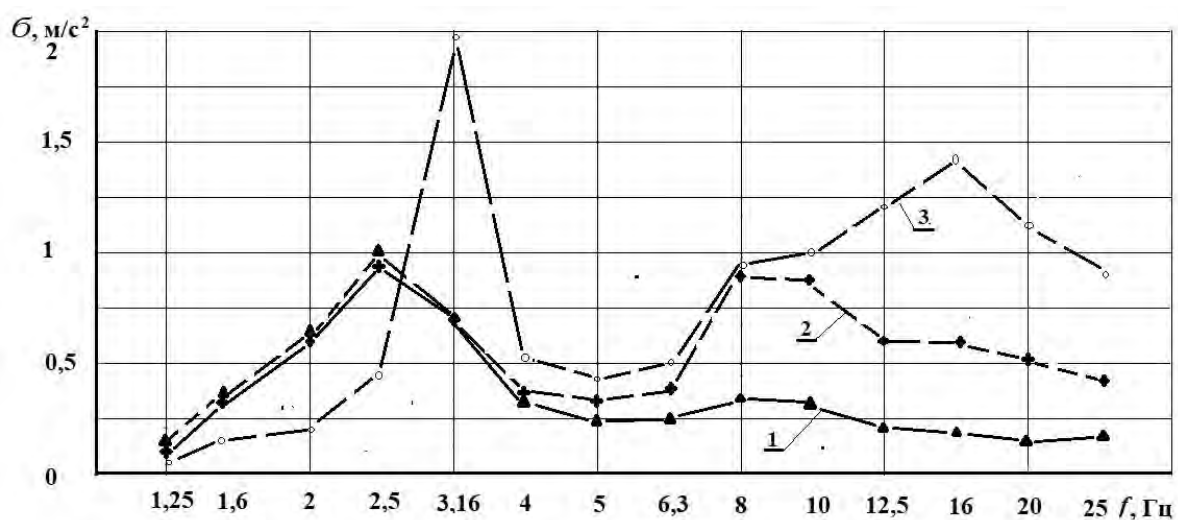


Рисунок 5.17. – Изменение уровней вибраций в третьоктавных полосах частот на сиденье (1), на раме над передним мостом (2), на раме над осью балансира (3) задней подвески при скорости движения $v=22$ км/ч

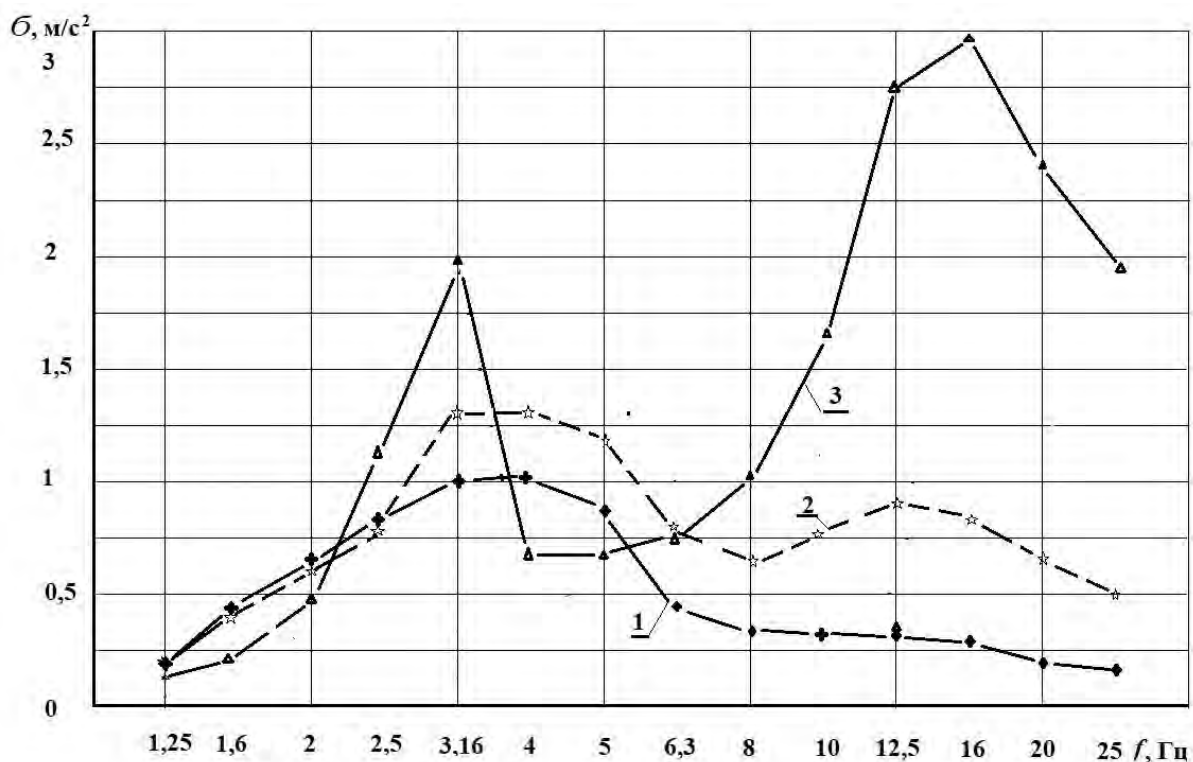


Рисунок 5.18. – Изменение уровней вибраций в третьоктавных полосах частот на сиденье (1), на раме над передним мостом (2), на раме над осью балансира (3) задней подвески при скорости движения $v=40$ км/ч

Наличие горба при скорости 23–25 км/ч обусловлено резонансными колебаниями из-за совпадения частот воздействия дороги (сформированными сельхозтехникой, ее скоростью передвижения), отсутствием амортизаторов в задней подвеске.

Для сравнения укажем, что уровни вибраций в октавных полосах частот (2, 4, 8, 16 Гц) на автомобиле МАЗ-6303 (ближайший аналог, образца 1978 г.) при движении по булыжному шоссе ($v=40$ км/ч) [14, 42, 75] составляют соответственно 1,06; 0,84; 1,127; 1,52 м/с² при общем уровне 2,4 м/с², что близко к полученным данным. Уровень продольных ускорений в октавах 2, 4, 8, 16 Гц составляет соответственно 0,125, 0,59, 2,10, 1,39 м/с². Необходимо учесть, что у шасси МАЗ-63017 более жесткая задняя подвеска т.к. он рассчитан на нагрузку 25 т, а МАЗ-6303 (1978 г.) на 14 т. А также тот факт, что масса кузова-контейнера всего 9,3 т, что сказывается на приведенных подрессоренных массах в сторону их уменьшения. Этим и состоянием дороги, разбитой грунтозацепами сельхозтехники, можно объяснить повышенный уровень вибраций.

При моделировании движения по асфальтовым дорогам уровень приведенных вертикальных ускорений на сиденье составляет соответственно 1,07 и 1,2 м/с² на дорогах хорошего и среднего качества при средних скоростях 78 и 66 км/ч. Это соответствует ощущение дискомфорта.

На рисунке 5.19 представлено изменение уровней вибраций на асфальтовой дороге в третьоктавных полосах частот на сиденье (1), на раме над передним мостом (2), на раме над осью балансира (3) задней подвески при средней скорости движения $v=66$ км/ч.

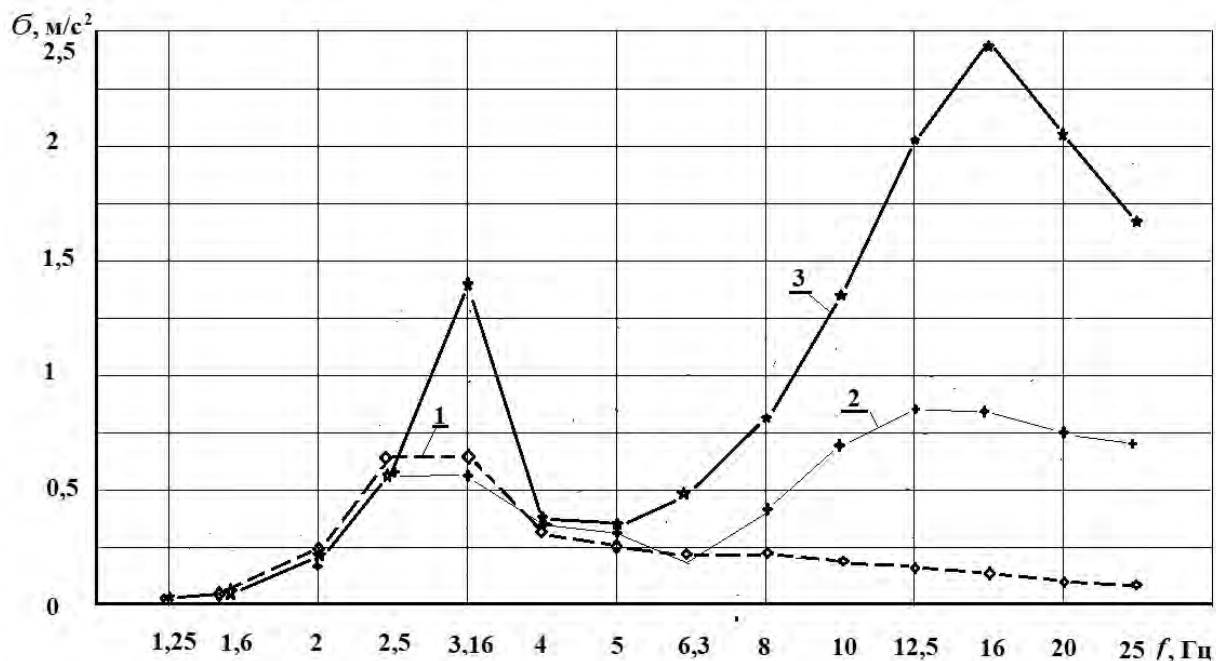


Рисунок 5.19. – Изменение уровней вибраций в третьоктавных полосах частот на сиденье (1), на раме над передним мостом (2), на раме над осью балансира (3) задней подвески при средней скорости движения $v=66$ км/ч

Анализ характеристики показывает, что спектр на сиденье приходится на третьоктавы 2,5; 3,16 Гц и свидетельствует о жесткой передней подвеске. На раме над задними мостами основной уровень вибраций приходится на третьоктавы 12,5 – 20 Гц и достигает среднеквадратического значений $2,45$ м/с².

По данным работы Гришкевича А.И. [57], предельно допустимым уровнем вибраций для водителя является $B = 2,5$ м/с² в общей полосе частот при максимальных значениях $7,5$ м/с². Подтверждением этого является то, что при проведении исследований движения армейских автомобилей на гравийных дорогах [57] водители двигались со средней скоростью 26 км/ч в условиях 8-ми часового дня, что близко к полученным выше результатам моделирования. Следует отметить, на том этапе (1970 – 1980 гг.) еще не проводились оценки вибраций в октавных и третьоктавных полосах частот и не учитывался фактор влияния частот.

Следует отметить высокий уровень общей вибраций ($B > 2,8$ м/с²) полученный на раме над задним мостом, что делает невозможным перевозку

персонала в кузове-контейнере на гравийных дорогах. Необходимо уточнить, что по инструкции перевозка персонала в кузове-контейнере не предусмотрена. Обычно на учениях для этого используются автобусы. Однако в боевых условиях это исключать нельзя.

Ниже на рисунках 5.20–5.21 приведены кривые зависимости ГОСТ/ISO чувствительности вибраций от частот и допустимого времени воздействия.

Следует отметить, что даже у легковых автомобилей по данным работы [100] наблюдается высокий уровень вибраций. Так, уровень вибраций зарубежных легковых автомобилей при средней эксплуатационной скорости 50 км/ч при поездках составляет 1–1,4 м/с² и имеет тенденцию увеличения с увеличением скорости. Это соответствует по ISO-2631 1–2 часа. И хотя он в 1,5 раза ниже чем у МАЗ-6317 он все же велик.

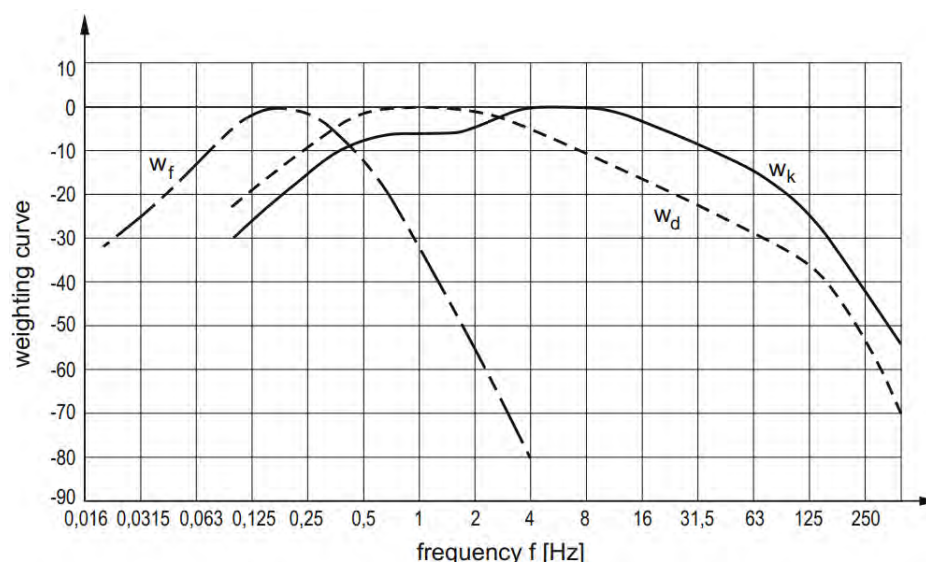


Рисунок 5.20. – Изменение коэффициентов чувствительности от частоты воздействия

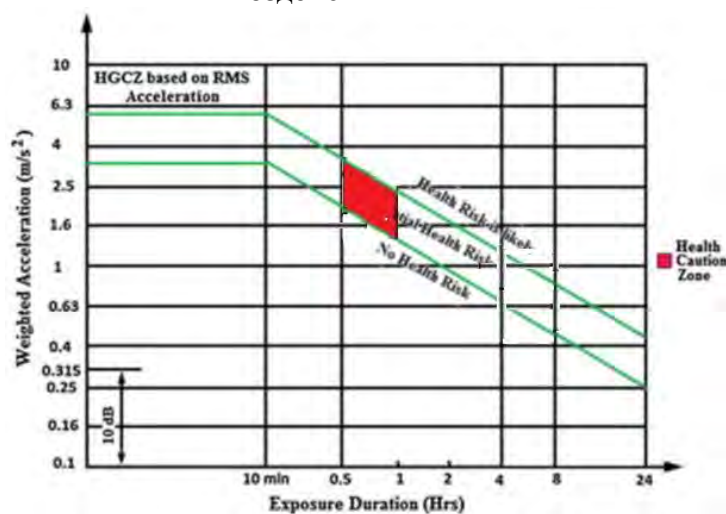


Рисунок 5.21. – Соответствие времени воздействия и уровня вибраций согласно ISO 2631

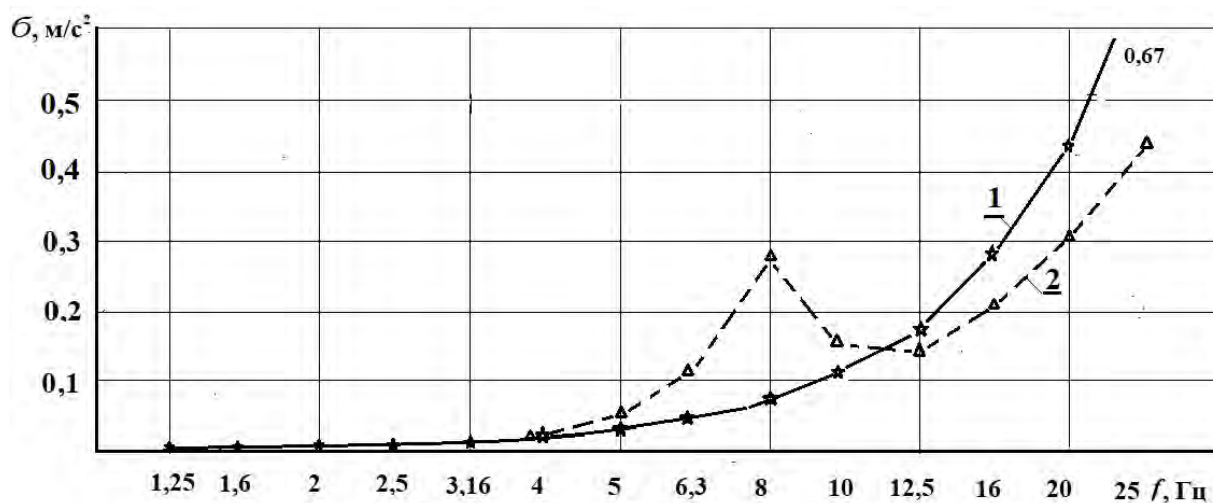
Анализ работ [75, 77, 78] по исследованию систем поддрессирования на булыжном шоссе показывает, что для снижения уровня вибраций колесных машин наиболее эффективным является применение малолистовых рессор. При этом потребуется установить амортизаторы в заднюю подвеску ввиду уменьшения трения в них.

5.3 Продольные и поперечные ускорения и результаты моделирования

Полученные результаты моделирования продольных колебаний автомобиля при средней скорости движения на асфальтированном шоссе $v = 57$ км/ч приведены на рисунке 5.22.

Под индексом 1 приведен график без учета колебательной системы трансмиссии, а под 2 с использованием двух массовой модели трансмиссии в виде: момента двигателя со сцеплением $0,25$ кг·м² и автомобиля 5770 кг·м².

Как видно из рисунка 5.22 основной спектр без модели трансмиссии приходится на полосу 8–28 Гц и достигает максимума в третьоктавной полосе 25 Гц ($0,67$ м/с²).



1 – без модели трансмиссии, 2 – с моделью трансмиссии

Рисунок 5.22. – Изменение уровней продольных вибраций МАЗ-6317 в третьоктавных полосах частот при средней скорости движения $v=57$ км/ч по асфальтированному шоссе

В случае варианта с моделью трансмиссии вибрации в третьоктаве 8 Гц составляют $0,285$ м/с², а в третьоктаве 25 Гц $0,44$ м/с². Уровень среднеквадратических ускорений в октавах 8, 16 Гц составляет соответственно $0,14$ м/с² и $0,36$ м/с², а в 16 Гц $0,54$ м/с² и $0,41$ м/с². Для сравнения

экспериментальный уровень вибраций в октавах 8, 16 Гц составляет 0,36; 0,41 м/с² [81].

Сравнение данных моделирования при воздействии трансмиссии (рисунок 3.48) на продольные колебания показывает хорошее совпадение с экспериментальными данными [81] и с характером изменения спектра и объясняет происхождение продольных вибраций автомобиля.

Аналогично дело обстоит с поперечными вибрациями. Источником их возмущения являются те же пульсации крутящих моментов двигателя. Уровень поперечных вибраций такой же как у продольных, что подтверждается данными экспериментальных исследований [81].

Проведенные исследования по воздействию микропрофиля дороги на продольные ускорения по схеме рисунка 3.1в показали, что их уровни вибраций получаются незначительными (несколько процентов от вертикальных).

5.4 Продольные ускорения от перемещений мостов

В данном подразделе будем рассматривать только динамические колебания моментов, вызываемые воздействием вертикальных сил на мост (его поворот) без учета колебательных процессов в трансмиссии согласно схемы на рисунке 3.43 и модели, описанной в п.3.13.4.

При моделировании рассматривались колебания правой и левой стороны машины при воздействии соответствующих микропрофилей дороги. Более идеальным было бы использование пространственной модели [11, 50]. Но это очень сложная и трудоемкая для Simulink задача с учетом оценки спектра и реальных нелинейностей [43]. Поэтому выбран подход рассматривать автомобиль разделенный на две части в поперечной плоскости при разном микропрофиле.

В процессе моделирования усилие записывалось в файл (через модуль To File). После чего эти усилия воспроизводятся через (через модуль From File) и обрабатывались по спектру.

Полученные результаты моделирования вертикальных и продольных колебаний автомобиля при средних скоростях движения $v = 40, 22$ км/ч приведены в таблице 5.3.

Сопоставление данных моделирования показывает хорошее совпадение по характеру изменения спектра и с экспериментальными данными.

Если учитывать продольные ускорения, то общие увеличиваются на 16 % по сравнению с вертикальными при скорости $v = 22$ км/ч и составляют 1,97 м/с². Это соответствует 0,5–1 часа допустимой работы.

Отметим, что на асфальтированном шоссе такого воздействия не наблюдается.

Таблица 5.3. – Значения продольных ускорений автомобиля при скоростях $v = 40, 22$ км/ч

Полоса в Гц	Скорость движения, моделирование		Экспериментальные при $v = 40$ км/ч	
	Гравийное шоссе		Булыжное шоссе	
	$v = 40$ км/ч	$v = 22$ км/ч	Продольные	Поперечные
0–25 Гц без коррекции	2,14	0,894	1,97	1,87
0–25 Гц коррек.	1,65	0,7		
2 Гц	0,125	0,03	0,125	0,25
4 Гц	0,263	0,103	0,58	0,456
8 Гц	0,94	0,44	1,27	1,44
16 Гц	1,745	0,704	1,39	1,07
1,25	0,112	0,0083		
1,6	0,0234	0,012		
2	0,048	0,019		
2,5	0,112	0,21		
3,15	0,157	0,053		
4	0,118	0,043		
5	0,174	0,077		
6,3	0,314	0,163		
8	0,514	0,251		
10	0,717	0,32		
12,5	0,98	0,39		
16	1,1	0,441		
20	0,944	0,383		
25	0,77	0,311		

5.5 Определение долговечности узлов автомобиля на стадии проектирования

Долговечность является основным параметром надежности.

Для определения долговечности узлов автомобиля на стадии проектирования требуется:

- иметь более совершенные модели движения и колебаний автомобиля;
- располагать данными по макро и микропрофилю дорог (заданными по точкам) и распределению движения по различным дорогам;
- располагать техническими средствами осуществления имитационного моделирования движения с учетом дорожной обстановки с переменной скоростью;
- создать 3D-модели исследуемого узла и иметь усталостные характеристики материалов, располагать и уметь пользоваться пакетами оценки напряженно-деформированного состояния;

– уметь провести систематизацию полученных данных по нагрузкам узла и переводу их в напряжения и расчету долговечности.

К сожалению, опубликованных работ по этому направлению на основе моделирования в реальных дорожных условиях как единого целого в СНГ пока нет. Хотя и имеется много работ по решению отдельных задач на базе экспериментальных данных [32, 33], использованию MSC.ADAMS, для оценки действующих максимальных нагрузок и прочностного напряженного состояния в MSC.NASTRAN узлов автомобиля при переезде препятствий. Но это оценка прочности, а не долговечности. По косвенным признакам из информации в интернете по ведущим автомобилестроительным фирмам [1, 2] можно сделать вывод, что задача оценки долговечности у них решается. Но она является их **Know how** и информация по ней недоступна, что требует собственных исследований.

5.5.1. Необходимые компоненты

А. Используемые модели

В процессе решения данной задачи необходимо осуществить совместное моделирование движения и колебаний автомобиля, движущегося по реальной дороге с переменной скоростью, и обеспечивающее приемлемую точность вычислений нагрузок (10–15 %) с учетом спектра. Этим требованиям отвечают модели главы 3 и работы [14]. Важным моментом для моделирования является использование пакета Matlab/Simulink, который позволяет осуществлять моделирование в реальном масштабе времени при реальном воздействии дороги и дорожной обстановки, а также организовать взаимодействие с другими техническими средствами.

Б. Реализация воздействия дороги и дорожной обстановки

Для решения этой комплексной задачи требуется осуществление имитационного моделирования [14]. Для обеспечения хорошей сходимости с экспериментом необходимо использование реального макро и микропрофиля дороги. Их получение и использование подробно описаны в работах [55, 56]. Использование спектральной плотности характеристик дорог, статистических методов и линеаризации, предлагаемых в пакете MSC.ADAMS, нежелательно, так как это приведет к большой погрешности расчета колебаний. Кроме того, в этом пакете невозможно осуществить моделирование с использованием больших массивов данных дороги, отсутствуют инструменты взаимодействия с другими средствами.

В. Технические средства осуществления имитационного моделирования

Чтобы получить данные о реальной нагруженности автомобиля целесообразно применение имитационного моделирования в реальном масштабе времени [14], и через колебания выйти на нагрузки на мост. Для этого необходимо использование быстродействующих компьютеров (CPU ~ 3,6–4 ГГц с соответствующей памяти (ОП > 16 ГБ, твердотельных накопителей .M2), мини-компьютеров с разнесением задач на них и организации их взаимодействия по сети CAN BUS. Наиболее идеальным вариантом является реализация схемного решения с вибрационным стендом с проекторами (рисунок 7.20 п. 7.8.7.3), подробно рассмотренная в работе [14], создающая иллюзию движения по реальной дороге.

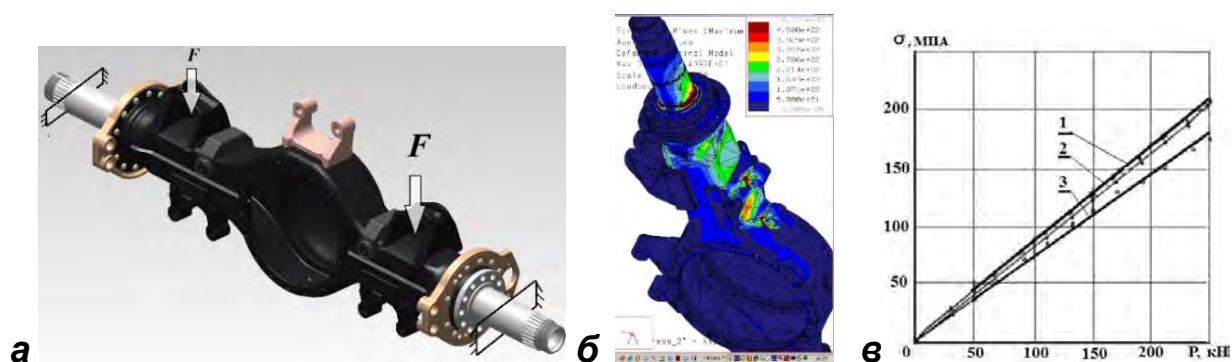
Как показали исследования на макетном настольном образце без вибрационного воздействия все же нет полного ощущения реальной дороги. Ввиду отсутствия стендов принято решение проводить исследования с помощью совместного моделирования движения и колебаний автомобиля на тех же моделях [14].

Г. 3D-модель исследуемого узла и усталостные характеристики материала

Использование 3D-модели исследуемого узла (рисунок 5.23 а) необходимо для выявления в статике наиболее опасных зон нагружения, по которым определяется НДС и переводной коэффициент от нагрузки на узел к напряжениям, на основе которых далее рассчитывается долговечность.

По данным работы [101], напряженное состояние картера без узлов трансмиссии незначительно отличается от моста в сборе (рисунок 5.11 в), что позволяет упростить модель. Для проведения расчетов необходимо также располагать усталостными характеристиками примененного материала в конструкции.

При отсутствии 3D-модели требуется проведение тензометрирования картера моста на стенде либо на машине путем его поддомкрачивания. Для этого может использоваться простейшая аппаратура на базе мини-компьютера Raspberry [14], либо использовать данные ранее проведенных исследований [101], что было применено в работе.



1 - в сборе с колесами; 2 - в сборе с тормозами и ступицами;
3 - картер в сборе без узлов трансмиссии

Рисунок 5.23. – 3D-модель моста (а), его напряженно-деформированное состояние (НДС) в PASTRAN/NASTRAN (б), зависимость напряжений от нагрузки для различных вариантов комплектности заднего моста (в)

Д. Систематизация полученных данных по нагруженности узла и пересчет ее в его долговечность

В результате совместного моделирования движения и колебаний автомобиля получаем данные по нагрузкам. Это массивы усилий, моментов, действующие на мост. Для расчета долговечности требуется перевести их в напряжения. Далее, чтобы определить долговечность узла необходимо систематизировать их и на основе кривой усталости определить долговечность узла. Это довольно сложная задача, которая подробно будет рассмотрена ниже.

5.5.2. Оценка долговечности узла

А. Оценку долговечности узла рассмотрим на примере картера заднего моста (рисунок 5.12).

Для систематизации данных более предпочтительным представляется использование метода “падающего дождя”, значительно сокращающего число циклов и режимов нагружений [102, 103] при оценке долговечности. Данный метод дает расчетные оценки долговечности, более сопоставимые с экспериментальными данными (см. рисунок 5.24 в, взятый из работы [104]).

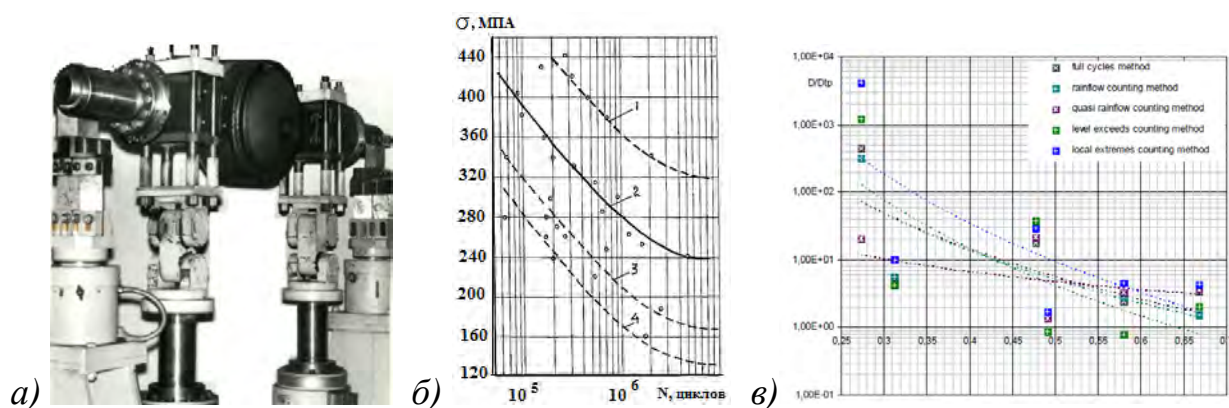


Рисунок 5.24. – Общий вид стендовых испытаний заднего моста (а), характеристики усталости материала (б) и сравнение методов оценки систематизации данных (в)

Б. Выбор метода систематизации данных

Выделение экстремальных значений случайного процесса нагружения

При проведении схематизации по любому из методов [81] вначале производится выделение экстремальных значений из процесса нагружения. Процесс выделения экстремумов подробно описан в работах [19, 81].

Метод “падающего дождя” позволяет проводить схематизацию в режиме реального времени, при этом запоминается лишь траектория потоков дождя. Обработка данных сложная и требует использование специальных программ. Для реализации метода “падающего дождя” в интернете предлагается ряд программ на языке C/C++. Одна из них [82], использовалась в данной работе.

5.5.3 Расчет долговечности

При расчетах долговечности используются методы, основанные на линейной, скорректированной линейной и нелинейной гипотезах суммирования повреждений, а также на учете снижения предела выносливости по мере наработки детали, рассмотренные в работе [101].

Одним из наиболее простых и распространенных условий суммирования усталостных повреждений является линейная гипотеза Велера [101]

$$\sum_{i=1}^n \frac{n_i}{N_i} = 1, \quad (5.1)$$

где N_i – число циклов по кривой усталости при амплитуде (σ_{ai}), соответствующей вероятности разрушения 50 %; n_i – число циклов повторения амплитуды σ_{ai} на этом режиме за срок службы.

Наибольшее распространение получила следующая формула, корректирующая линейную гипотезу усталостных повреждений:

$$\begin{aligned}\sigma_{-1\text{пов}} &= \sigma_{-1\text{пов},i-1} \left[1 - \frac{n_i}{N_i} K \left(\frac{\sigma_{ai}}{\sigma_{-1\text{пов},i=1}} - 1 \right) \right] \\ \sum_{\sigma_{ai} > \sigma_{-1D}} \frac{n_i}{N_i} &= a_p \\ a_p &= \frac{\varepsilon \sigma_{ai} - 0,5 \sigma_{-1D}}{\sigma_{amax} - 0,5 \sigma_{-1D}} \\ \varepsilon &= \sum_{\sigma_{ai} > 0,5 \sigma_{-1D}} \frac{\sigma_{ai}}{\sigma_{max}} \frac{n_{i6}}{n_6} = a_p, \quad (5.2)\end{aligned}$$

где σ_{ai} – амплитуда i – го уровня в блоке нагружения, состоящем из n ступеней ($i = 1, \dots, n$); $\sigma_{amax} = \sigma_a$ – наибольшая амплитуда в блоке; σ_{-1D} – предел выносливости детали; n_{i6} – число циклов в блоке, соответствующее амплитуде a_p ; n_6 – общее число циклов в блоке после исключения из него амплитуд $\sigma_{ai} < 0,5 \sigma_{-1D}$.

По данным работы [101], использование предложенных зависимостей для расчетов долговечности образцов приводит в некоторых случаях к существенным ошибкам в сторону занижения долговечности.

В работе [101] рассмотрен метод суммирования усталостных повреждений при нерегулярном нагружении, основанный на учете постепенного снижения предела выносливости вследствие предварительного динамического нагружения. В основу метода положено уравнение

$$\sigma_{-1\text{пов}} = \sigma_{-1\text{пов},i-1} \left[1 - \frac{n_i}{N_i} K \left(\frac{\sigma_{ai}}{\sigma_{-1\text{пов},i=1}} - 1 \right) \right], \quad (5.3)$$

где $\sigma_{-1\text{пов},i-1}$ – величина предела выносливости поврежденного металла после приложения $(i-1)$ -й ступени повреждения;

N_i – число циклов по кривой усталости поврежденного металла после приложения $(i-1)$ -й ступени при амплитуде σ_{ai} ; n_i – число реальных циклов σ_{ai} в диапазоне квантования; K – параметр уравнения, характеризующий интенсивность снижения предела выносливости от предварительного нагружения.

Анализ работы [101] показывает, что относительные долговечности $\frac{n_i}{N_i}$, найденные по скорректированной линейной гипотезе (6.4) и по методу (6.3), весьма близки между собой, поэтому можно воспользоваться более простой формулой (5.1).

А. Определение долговечности картера моста автомобиля

Для получения оценки долговечности картера моста автомобиля вначале необходимо определить коэффициент перевода нагрузки в напряжения. Для чего воспользуемся 3D-моделью картера моста, загруженной в NASTRAN или ANSYS, где проведем заземление и нагружение моста (см. рисунок 5.11) либо экспериментальными данными. Это необходимо для выявления наиболее нагруженной точки и определение коэффициента связи нагрузки и напряжений, составившего для исследованной конструкции 0,8 МПа/кН.

Далее для оценки нагруженности проводилось моделирование движения с переменной скоростью и колебаний автомобиля 6×6 на трех (10-километровых) участках дороги: асфальт очень хорошего качества, среднего и на разбитом грунтовом шоссе, описанные соответственно, как 1, 2, 5 в работах [14, 55, 56] с распределением пробега 40, 50 и 10 %.

Оценивая нагрузку на мост через ускорения неподрессоренных масс и статическую нагрузку на мост как $F = m_m \cdot \ddot{z}_m + F_{ст}$ в результате моделирования через программу на C/C++ в модуле S-Function Builder Simulink через переводной коэффициент получим массив напряжений на основе усилий (в формате txt, dm.txt), действующих в мосте (~ 40000 записей на 10 км пути) для каждой дороги. Далее суммируем файлы массивов командой Windows

copy 2dm.txt+2dm.txt+2dm.txt+2dm.txt+2dm.txt+3dm.txt+3dm.txt+3dm.txt+3dm.txt+3dm.txt+3dm.txt+1dm.txt /b dmsum.txt

соответственно по 4, 5, 1 блоков массивов, полученных для этих дорог, равных суммарному пробегу 100 км (около 400000 записей).

Для систематизации данных по методу дождя использовалась модернизированная автором программа RainFlow.exe [104] (приложение И), в которую вводился файл указанного массива (**dmsum.txt**). Указывался выходной файл и значение напряжений ниже предела усталости. Результаты их пересчета с помощью разработанной программы на C/C++ приведены в таблицах 5.4–5.6.

Таблица 5.4 – Результаты обработки НДС моста при движении по грунтовой, асфальтированным дорогам с распределением пробега 10, 40, 50 % (при средней скорости $v = 26,5$ км/ч на грунтовой дороге) программой RainFlow.exe

Сред амп, МПа	Мах ампл, МПа	Сред Знач, МПа	Min Val, МПа	Мах Peak, МПа	R	Кол-во циклов	$\sigma_{\text{ЭКВИВ}}$, МПа	n_i/N_i
1	2	3	4	5	6	7	8	9
224,20	233,60	40,64	-193,70	273,40	-0,829	3,00	215,39	0,0000183
197,30	200,60	58,87	-151,40	274,20	-0,755	4,50	178,14	0,0000121
172,40	181,40	51,07	-142,70	246,30	-0,787	9,00	163,73	0,0000167
148,10	160,60	48,46	-126,50	219,20	-0,788	28,50	145,03	0,0000313
125,80	138,70	54,73	-98,68	211,20	-0,711	78,00	120,43	0,0000383
					Ресурс	1275065	км	0,0000784

Примечание. 1. Цветом выделены данные, получаемые в RainFlow.exe, остальные в Resource_M.exe.

2. Данные по дорогам 2, 3 ввиду малости значений не приводятся.

3. Для сокращения места удалены строки с $\sigma_{\text{ЭКВИВ}} < 140$ МПа.

Таблица 5.5 – Результаты обработки НДС моста при движении по грунтовой дороге (при средней скорости $v = 26,5$ км/ч на грунтовой дороге) программой RainFlow.exe

Сред амп, МПа	Мах амп, МПа	Сред знач, МПа	Min Val, МПа	Мах Peak, МПа	R	Кол- во цикло в	$\sigma_{\text{ЭКВИВ}}$, МПа	n_i/N_i
1	2	3	4	5	6	7	8	9
224,70	235,00	38,10	-195,70	274,20	-0,833	4,00	217,06	0,0000038
197,30	200,60	58,87	-151,40	274,20	-0,755	4,50	178,14	0,0000042
173,90	181,40	53,73	-140,40	246,30	-0,774	8,00	162,68	0,0000075
149,60	163,70	56,43	-114,30	219,20	-0,698	18,50	141,15	0,0000175
					Ресурс	302857	км	0,0000330

Таблица 5.6 – Результаты обработки НДС моста при движении по грунтовой дороге (при средней скорости $v = 38,5$ км/ч на грунтовой дороге) программой RainFlow.exe

Сред амп, МПа	Мах ампл, МПа	Сред знач, МПа	Min Val, МПа	Мах Peak, МПа	R	Кол-во циклов	$\sigma_{\text{ЭКВИВ}}$, МПа	n_i/N_i
1	2	3	4	5	5	7	8	9
352,50	361,20	3,60	-348,90	373,50	-0,966	1,00	355,59	0,0000009
297,70	305,80	33,11	-285,30	373,50	-0,933	2,00	296,44	0,0000019

Окончание таблицы 5.6

1	2	3	4	5	5	7	8	9
273,60	281,00	2,87	-272,9	289,10	-0,971	1,50	277,30	0,0000014
232,60	252,60	32,35	-216,3	289,10	-0,856	9,50	236,03	0,0000090
194,80	209,40	31,29	-194,7	249,70	-0,930	18,50	202,69	0,0000175
160,40	180,10	35,59	-172,3	241,80	-0,957	50,00	176,54	0,0000472
					Ресурс	53266	км	0,0001877

Примечание. 1. Цветом выделены данные, получаемые в RainFlow.exe, остальные в Resource_M.exe.

2. Данные по дорогам 2, 3 ввиду малости значений не приводятся.

3. Для сокращения места удалены строки с $\sigma_{\text{эквив}} < 140$ МПа.

Полученные данные в RainFlow.exe необходимо привести к симметричным напряжениям эквивалентным по повреждению для чего воспользуемся, приведенными в работе [101] формулами

$$\sigma = \sigma_{\max} - (a b - 1)\sigma_{-1}, \quad (5.4)$$

$$\text{где } a = \frac{2}{2-(1-\varphi)(1+R)}; \quad b = \frac{v_2}{v_R} = \frac{1}{\frac{v_1}{v} + (\frac{v_1}{v_2} - 1)R},$$

σ_{-1} – предел выносливости при симметричном цикле;

v_2 – угол наклона кривой усталости при симметричном цикле нагружения;

v_1 – угол наклона кривой усталости при отнулевом цикле нагружения.

Заданным значениям пределов выносливости σ_R соответствуют значения характеристик угла наклона кривых усталости v , определяющихся из уравнения

$$v = v_0 \frac{\sigma_R}{\sigma_R + v_0}, \quad (5.5)$$

где v_0 – параметр уравнения кривой усталости при симметричном цикле, МПа.

v – показатель угла наклона кривой усталости при симметричном цикле.

В работе [101] на основе преобразований предлагается использование более упрощенного варианта

$$\sigma_{\text{экв } i} = \frac{\sigma_{\max i}}{a} \quad (5.6)$$

где $a = \frac{2}{2-(1-\varphi)(1+R)}$ – параметр, характеризующий изменение величины предела выносливости при переходе от симметричного цикла напряжений к циклу с коэффициентом асимметрии R ;

φ – коэффициент чувствительности к асимметрии цикла, определяемый на основе кривой усталости. Для стали 10ХСНД φ равен 0,087, $\sigma_0 = 140$ МПа [18].

$R = \frac{\sigma_{\min i}}{\sigma_{\max i}}$ – коэффициент асимметрии напряжений цикла в блоке.

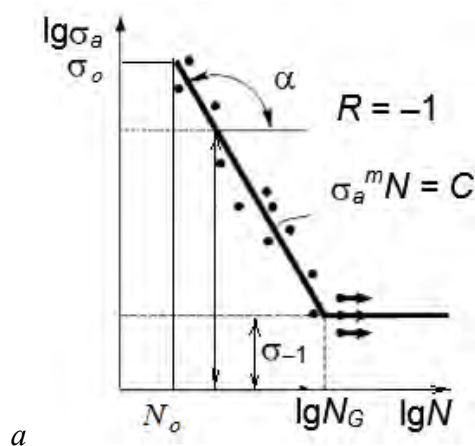
На основе этих формул и полученных данных производился расчет эквивалентных напряжений, данные которых дополнительно внесены в таблицы 5.2, 5.3, 5.4 к данным RainFlow. Далее на основе эквивалентных напряжений и кривой усталости определялись соответствующие им количества циклов N_i , которое может выдержать узел на этом режиме.

Если разделить 1 на отношение $\frac{n_{\text{действ}}}{N_{\text{устал}}}$, то получим коэффициент, во сколько раз больше узел выдержит циклов при заданном напряжении и пробеге по сравнению с кривой усталости, а при его умножении на длину участка получим пробег автомобиля.

Расчет количества циклов на основе кривой выносливости (рисунок 5.25) определялось по следующей формуле:

$$N = N_{\sigma} \left(\frac{\sigma_{-1}}{\sigma_a} \right)^m,$$

где $\operatorname{tg} \alpha = m = \frac{\lg N_{\sigma} - \lg N_{\sigma_0}}{\lg \sigma_a - \lg \sigma_{-1}}.$



б

Рисунок 5.25. – Кривая усталости (а) и разработанная программа пересчета данных (б)

Для автоматизации расчетов по вышеперечисленным зависимостям разработана программа Resource_M.exe ввиду трудоемкости ручного пересчета из-за того, что получаемый текстовый файл результатов в программе RainFlow.exe содержит коды табуляции и переменное число пробелов между данными, затрудняющие считывание по позиционированию. Resource_M.exe учитывает все это при считывании данных из этого файла и выполняет необходимый расчет, готовый (в формате txt) для вставки в таблицу

MS.Word. Программа может также использоваться для обработки экспериментальных данных.

Для определения долговечности на разных дорогах с разной длиной участков ($L_{\text{баз}}, L_i$) и различными долями распределения ($d_{\text{баз}}, d_i$) может применяться и другой подход:

- выполняется расчет напряжений отдельно для этих дорог в программе RainFlow.

- за базис берется дорога с наибольшей нагруженностью (например, грунтовая), а в в строках файлах других участков корректируются значения циклов ($N_{\text{Cycle Counts}}$) в пропорции их длин участков и долями (d_i) распределения пробегов по сравнению с базовым ($d_{\text{баз}}$);

$$N_{\text{Cycle Counts_new}} = \left(\frac{L_i}{L_{\text{баз}}} \right) \left(\frac{d_i}{d_{\text{баз}}} \right) N_{i(\text{Cycle Counts_old})}; \quad (5.7)$$

- во всех их текстовых файлах выделяются строки с напряжениями выше предела выносливости, которые копируются в новый файл (itog.txt) редактором FAR.EXE (не вносящим управляющих символов);

- далее производится расчет по этому файлу в программе Resource_M.exe с указанием базовой длины участка первой дороги.

5.5.4. Анализ полученных данных

Результаты моделирования свидетельствуют о существенном влиянии скорости движения на нагруженность картера моста. Так, ресурс на грунтовой дороге при средней скорости движения $v = 26,5$ км/ч составляет 302857 км, а при $v = 38$ км/ч снижается до 53266 км. Выявлено, что при движении грузового автомобиля по асфальтированным дорогам напряжения на штамповарном картере ведущего моста могут быть исключены из рассмотрения как неповреждающие, что позволяет сократить время ускоренных испытаний на стенде. В общем пробеге определяющим является движение по грунтовой дороге: при ее доле 10 % в пробеге общий ресурс (при $v = 26,5$ км/ч) получается равным 1275065 км, т. е. при эксплуатации автомобиля картер моста практически не повреждается.

Сложными моментами в совместном моделировании является задание скоростного режима на различных участках дороги и распределение по дорогам. Здесь могло бы помочь использование имитационного моделирования с применением стенда (см. рисунок 7.20), позволяющее учесть влияние водителя на выбор скорости и воздействия на него вибраций, визуальной обстановки на дороге. Это дало бы возможность получить данные

о реальном режиме движения, а на основе их рассчитать нагруженность и долговечность узлов автомобиля на стадии проектирования.

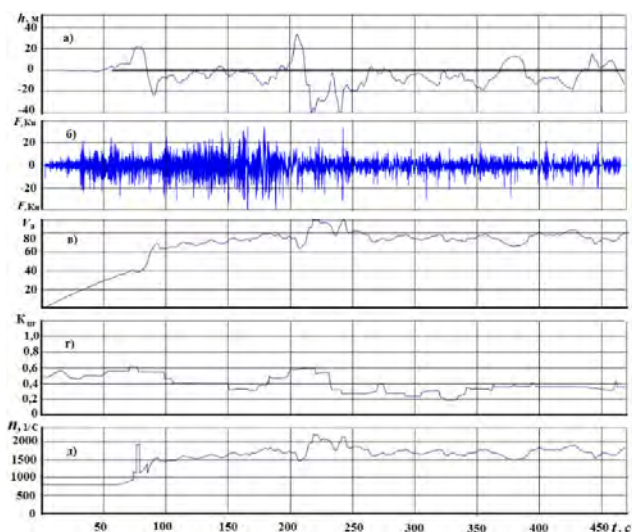
5.6 Результаты имитационного моделирования автомобиля на макетном образце с участием водителя в режиме ручной КПП и АКПП

С помощью блок-схемы (рисунок 3.38) проведено сравнительное моделирование в режиме ручной КПП и АКПП при задействовании человека.

Учитывалось переменное сопротивление качения, связанное с высотой микропрофиля дороги, взятое из п.2.11.

В режиме ручного управления КПП электрический сигнал с галетного переключателя, соответствующий номеру передачи КПП, подавался через АЦП в Raspberry Pi 3 и далее через сеть Samba в модуль S-Function Builder в Simulink.

Результаты моделирования при восьмискоростной АКПП приведены на рисунке 5.26, а при ручном переключении на рисунке 5.27.



а – профиль дороги, б – вертикальные усилия, действующие на передний мост, в – скорость движения, г – степень нажатия на педаль газа, д – обороты двигателя

Рисунок 5.26 – Результаты имитационного моделирования движения и нагруженности автомобиля при автоматическом переключении передач КПП

Средняя скорость при использовании режима автоматического переключения КПП составила 67,5 км/ч, расход топлива 47,2 л/100 км. Уровень среднеквадратических ускорений на раме над передней осью в октавных полосах частот 2, 4, 8, 16 Гц составил соответственно 0,84; 1,0; 0,68; 0,72 м/с².

Максимальное вертикальное усилие от положения статического равновесия, действующее на передний мост составило 40 кН, при среднеквадратическом значении 4,83 кН. Максимальный крутящий момент в трансмиссии, подведенный к колесам получился равным 44,3 кН·м при среднеквадратическом значении 6,66 кН·м.



Рисунок 5.27 – Результаты имитационного моделирования движения автомобиля при ручном переключении передач КПП

При использовании режима ручного переключения КПП (рисунок 5.27) при четырех моделирований средняя скорость составила 63,4/69,3/63,5/65,9 км/ч, расход топлива соответственно 43,9/38,2/45,1/42 л/100 км. Как видно из этих результатов в режиме ручного переключения КПП получается более низкая средняя скорость (до 3 %) при более низком расходе топлива (на 12 %).

Причиной этого является использование испытуемым более низких оборотов двигателя при переключении передач, что подтверждается результатами работ [32, 33].

Отмечено, что на макетном настольном образце все же нет полного ощущения автомобиля и не чувствуется работа двигателя, которая обычно оценивается водителем по звуку двигателя и динамике машины. Это свидетельствует о необходимости введения в систему имитации звука двигателя и создания полноценного стенда.

Использование предложенных систем моделирования позволяет на стадии проектирования предварительно определить режимы движения и нагруженности автомобиля, отработать его эргономику, оценить его параметры, влияние параметров трансмиссии на динамику, экономичность и использовать полученные данные при испытаниях на стенде.

Выводы по главе 5

1. Проведены комплексные исследования на основе предложенного метода моделирования движения автомобиля, базирующегося на данных GPS массивов макропрофилей, режимов движения и способа учета торможений перед препятствиями.

2. Путем моделирования движения в городских условиях на основе вышеперечисленного установлено, что в городских условиях на участке длиной 22090 м средняя скорость составляет $v_a = 51.93$ км/ч, расход $Q=41,17$ л/100 км, средний наклон дороги – 1,24 %, средняя мощность и число оборотов двигателя соответственно 80,28 кВт и 1674 1/мин. Имеет место 11 торможений, возможны еще 4 торможения на перекрестках. Максимальный угол наклона достигает 4 %. Обороты двигателя варьируются от 1000 до 2200 1/мин. При движении используются передачи КПП от 3 до 8-й. Каждый переезд препятствий вынуждает переходить на более низшую передачу. Максимальная скорость достигает 82 км/ч (на выезде из города). Ограничивают скорость движения препятствия, остановки.

3. При моделировании движения по магистральной трассе длиной 103 км, характеризующейся небольшими подъемами с изменением высоты до 60 м на первой половине и перепадом высот 100 м на второй, с максимальным углом наклона 4 % установлено, что при $m_a = 20000$ кг средняя скорость составляет $v_a = 81.8$ км/ч, расход $Q=30,59$ л/100 км, средний наклон дороги – 0,88 %, средняя мощность и число оборотов двигателя соответственно 96,2 кВт и 1902 1/мин. Максимальная скорость достигает 95 км/ч. Обороты двигателя варьируются от 1000 до 2150 1/мин. В основном двигатель работает в диапазоне 1700 – 1900 1/мин.

На трассе 103 км имеют место 3 торможения. При движении используются передачи КПП от 4 по 8-й с преобладанием 8-й. При переезде препятствий происходит переход на более низшую передачу. Как видно из данных и рисунка препятствия, остановки движения практически не ограничивают среднюю скорость движения. При увеличении массы до 32000 кг увеличивается средняя мощность двигателя в 1,26 раза.

4. При моделировании движения по загородной трассе установлено, что при $m_a = 20000$ кг средняя скорость составляет $v_a = 81.8$ км/ч, расход $Q=30,59$ л/100 км, средний наклон дороги – 0,877 %, средняя мощность и число оборотов двигателя соответственно 96,2 и 1902 1/мин. Максимальная скорость достигает 95 км/ч. Обороты двигателя варьируются от 1000 до 2150 1/мин. В основном двигатель работает в диапазоне 1700 – 1900 1/мин.

5. Выявлено влияние вибраций на скорость движения с помощью предложенного способа формирования совместного макро и микропрофиля

дороги путем суммирования продольного профиля дороги, полученного с помощью GPS [55, 56] и микропрофиля гравийной дороги возле н/п Салюты под Минском, записанных на основе ускорений поддрессоренной и неподдрессоренной части автомобиля [14, 55, 56].

6. Получены результаты моделирования колебаний автомобиля по ИСО-2631 с использованием подхода обработки колебаний по третьоктавным полосам частот (1–25 Гц) с пересчетом их к приведенным вибрациям.

8. Моделированием движения по асфальтовым дорогам установлено, что уровень приведенных вертикальных ускорений на сиденье составляет соответственно 1,07 и 1,2 м/с² на дорогах хорошего и среднего качества при средних скоростях 78 и 66 км/ч и соответствуют ощущению дискомфорта.

9. Выяснено, что продольные и поперечные ускорения автомобиля вызываются работой двигателя, а также поперечно-угловыми поворотами моста под действием вертикальных вибраций относительно карданных валов, передающих моменты, необходимых для движения автомобиля. На основе этого предложены схемы и математические модели вертикальных и продольных и поперечных колебаний, вертикальных сил на мост с учетом колебательных процессов в трансмиссии.

10. На основании предложенной схемы и модели проведено моделирование продольных и поперечных ускорений. Установлено, что при учете продольных ускорений общие увеличиваются на 16 % по сравнению с вертикальными при скорости $v = 22$ км/ч и составляют 1,97 м/с² при допустимом времени работы 0,5–1 часа, достаточной для смены позиции в условиях гравийного дорожного покрытия или бездорожья.

11. Установлено, что при использовании подходов на основе воздействия $\tan \alpha$ наклона микропрофиля дороги и вертикальных перемещений мостов уровни продольных и поперечных вибраций получаются незначительными (несколько процентов от вертикальных).

ГЛАВА 6

ПРОГНОЗИРОВАНИЕ ДВИЖЕНИЯ КОЛЕСНЫХ МАШИН

6.1. Задание маршрута и режима движения с помощью карт и интернета

Рассмотрим, как с помощью карт и интернета можно задать маршрут (точки 1–20–1) и режим движения автомобиля, необходимые для компьютерного моделирования (без стенда) и оценки его свойств. Для этого воспользуемся топографическими картами и сайтами с картами высот [72, 73], на основе которых создан совмещенный рисунок 6.1.



Рисунок 6.1 – Задание маршрута и режима движения с помощью карт и интернета

Наметим маршрут, определим высоты в определенных точках. Для получения продольного профиля дороги и карты используем сайты [88, 89] и топографическую карту масштаба 1:500000, на которой намечаем маршрут и точки замеров. На ней с помощью курвиметра определяем расстояние от базовой или характерных точек. Определяем примерный масштаб задаваемых расстояний между точками (шаг) и начинаем указывать мышью шаги на электронной карте. Получаем значения высот, которые фиксируются в журнале, после чего вводятся в виде массива в текстовый файл (каждое значение в своей строке). Их данные будут в программе внесены в бинарный массив памяти высот (float, 4 байта), формирующий продольный профиль дороги. Но последний требует фильтрации. Для чего используем следующую блок-схему Simulink для фильтрации профиля дороги (рисунок 6.2).

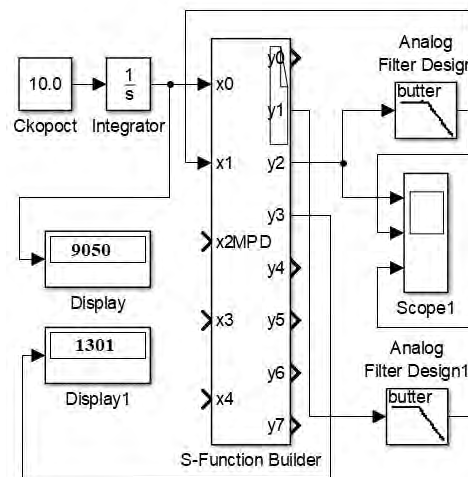


Рисунок. 6.2 – Блок-схема фильтрации, формирующая продольный профиль дороги

Выбор такого решения обуславливается необходимостью интерполяции и фильтрации исходных данных из-за их ступенчатого характера и нивелирует погрешности указаний точек и их замеров.

Основная часть исходного текста программы на языке C для модуля S-Function Builder приведена ниже.

```
//-----
// Заголовочная часть программы модуля S-Function Builder, сформированная им
float rd[25000];
long ch=0; //счетчик первоначальной загрузки массива высот
long chf=0; //счетчик (триггер), фиксирующий задержку
long i=0; //счетчик числа занесенных строк высот

//-----
static void mdlOutputs(SimStruct *S, int_T tid)
{
```

```

const real_T *x0 = (const real_T*) ssGetInputPortSignal(S,0);
const real_T *x1 = (const real_T*) ssGetInputPortSignal(S,1);
const real_T *x2 = (const real_T*) ssGetInputPortSignal(S,2);
const real_T *x3 = (const real_T*) ssGetInputPortSignal(S,3);
const real_T *x4 = (const real_T*) ssGetInputPortSignal(S,4);
real_T *y0 = (real_T *)ssGetOutputPortRealSignal(S,0);
real_T *y1 = (real_T *)ssGetOutputPortRealSignal(S,1);
real_T *y2 = (real_T *)ssGetOutputPortRealSignal(S,2);
real_T *y3 = (real_T *)ssGetOutputPortRealSignal(S,3);
real_T *y4 = (real_T *)ssGetOutputPortRealSignal(S,4);
real_T *y5 = (real_T *)ssGetOutputPortRealSignal(S,5);
real_T *y6 = (real_T *)ssGetOutputPortRealSignal(S,6);
real_T *y7 = (real_T *)ssGetOutputPortRealSignal(S,7);

double x, ts, h, t, t2, t3, x_1, y_1, P0, P1, P2, P3, u, mp, hr;
float mp, hr, hm;
long n, dl;

FILE *fp1,*fp2;
if (ch==0)
{
fp1=fopen("phv.txt","rt"); // Открываем сформированный текстовый файл массивов высот
while(fgets(hm, 198, fp7)!=0)
{
dl=strlen(hm);
hm[dl-1]='\0'; // удаление последнего невидимого символа перевода в строке высот
rd[i]=hm-240.8; //Заносим в массив со смещением высот точки
// (берется по первой точке замеров)
i++;
} // конец while
ch=1; // включение триггера завершения загрузки высот
fclose(fp1); // закрытие файла после загрузки информации
}
ts=100.0; // шаг замеров на карте
x = x0[0];
if (x<0.0)
x=1.0;
x_1 = x1[0];

// Определяем значение высоты для точки x1, используя метод интерполяции
//CatMull-Rom [10] по 4-м точкам при фиксированном шаге
n=u=x/ts;
t=u-n;
//t=fabs(u- n);
t2=t*t;
t3=t*t*t;
// sx =20* x1[0];

if (n>2)
{
P0=rd[n-1];
P1=rd[n];

```



```

P2=rd[n+1];
P3=rd[n+2];

mp= 0.5 *((2*P1) +(-P0 + P2) * t +(2*P0 - 5*P1 + 4*P2 - P3) *t2 +(-P0 + 3*P1- 3*P2 + P3) *
t3);
*y2=mp;
hr=x1[0];
// Определяем задержку пути после фильтра (Display1)
if (chf==0&& x>100.0&&(x_1>0.2||x_1<-0.2))
{
fx=x;
*y3=fx;
chf=1;
}
// Запись макропрофиля с задержкой
if (x>1301)
{
fp2=fopen("RD.bt","ab");
    fwrite(&hr, 4, 1, fp2);
    fclose(fp2);
}
} //if (n>2)
}

```

Учитывая, что в работе ориентируемся на комплексные исследования, для чего используем совместный файл макро и микропрофиля, для которого устанавливаем шаг 0,125 м. Для получения его варьируем шагом моделирования Simulink ($t=0.04\text{--}0.075$ с), подбирая экспериментально его значение, чтобы получить необходимую длину участка и шаг ($S=N_{\text{байт}} \cdot 0,125/4$).

На основе реализованных программой графиков (рисунок 6.3) и карт маршрута, используя программы mspaint и Photoshop, строим совмещенную карту маршрута и высот (см. рисунки 6.1, 6.3), которые позволяют всесторонне провести анализ.

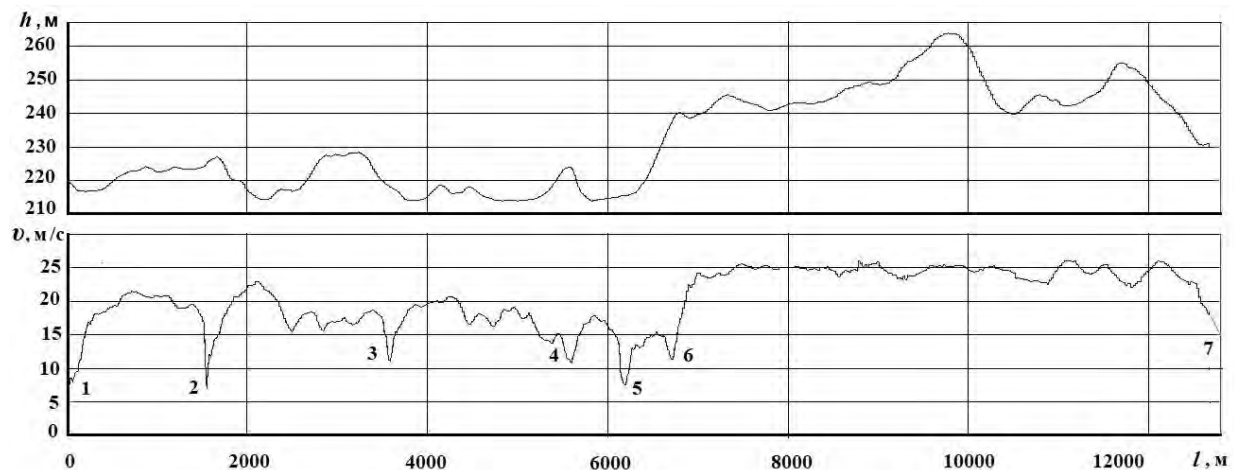


Рисунок 6.3. – Изменение высоты макропрофиля дороги и скорости движения

Анализ экспериментальных данных показывает, что при разгоне для достижения скорости 17 м/с требуется 250 м, до 22 м/с – 500 м. В дальнейшем водитель несколько снижает скорость до 20 м/с, особенно если, он видит впереди возможную помеху: перекресток, скопление техники возле торговых центров. Перед перекрестками за 100 м начинает снижать скорость до 15–18 м/с, а проезд препятствий (лежащий полицейский, мост) осуществляет со скоростью 7–8 м/с. На подъемах, где ограничена видимость (волновой характер высот) придерживается скорости 17 м/с. Перед населенными пунктами за 100–200 м снижает скорость до ограниченной там скорости (40/50/60 км/ч).

При значительных поворотах дороги 60-90° за 100–200 м снижает скорость до 12 м/с, здесь чаще применяется торможение.

На автострадах (I категории) при отсутствии помех и ограничений водители придерживается скорости 22–25 м/с.

Анализ экспериментальных данных показывает, что водитель на местных дорогах при наличии помех ограничивается скоростью 20–23 м/с,

На участках местных дорог, где нет помех и позволяет дорожная обстановка, движется со скоростью 25 м/с (90 км/ч допустимой правилами дорожного движения).

Скорость прохождения поворота с учетом влияния жесткостей упругих элементов подвески и шин можно определить по формуле 3.12.

$$v_{кр} = \sqrt{\frac{gBR}{2(h_g + \frac{h_{кр}^2 m_a}{2c_p B_p^2} + \frac{h_{ш}^2 m_a}{2c_{ш} B^2})}},$$

где $h_{кр}$, $h_{ш}$ – плечо крена подрессоренной массы при деформации подвески и шин соответственно; m_a – подрессоренная масса колесной машины; $g = 9,81 \text{ м/с}^2$;

c_p , B_p – жесткость подвески и расстояние между площадками рессор соответственно;

$c_{ш}$, B – жесткость шин и колея соответственно; R – радиус поворота.

На основе этого и снимков местности в интернете проводим анализ маршрута. Создаем вначале укрупненную общую таблицу характеристики маршрута (таблица 6.1).

Таблица 6.1 – Характеристика и особенности маршрута

№ точки	Характеристика	Особенность	L от базовой точки	v , м/с	h точки, м
1	2	3	4	5	6
1	Начало движения		0	0	208
2	Нерегулируемый перекресток	На что, указывает характер к нему дороги.	1050	20	225
3	н/п Военный городок, остановка автобуса	Наличие препятствия “лежачий полицейский”	1550	7	227
4	Дачи, н/п Приморье	Ограничение скорости 60 км/ч	2400	18	208
5	Поворот	Необходимость снижения скорости	3540	11	205
6	Населенный пункт Семков городок	Ограничение скорости 40 км/ч	4500	14	206
7	Конец н/п	Снятие ограничения скорости	4800	18	207
8	н/п Чучаны, мост	Ограничение скорости 60 км/ч	5520	16	205
9	Развязка дорог	Целесообразно снизить скорость	5850	14	240
10	Поворот на н/п	Возможно ограничение скорости	7000	17	242
11	Остановка		12500	0	230
12	Въезд в н/п	Саламоречье	20500	4	235
13	Поворот в н/п		15600	8	232
14	Поворот дороги		14300	11	235
15	Перекресток, подъем дороги	Перед Касынью	16200	13	242
15	Поворот после подъема	Ограничение скорости 50 км/ч из-за ухудшения видимости	16450	12	265
16	Крутой спуск	Ограничение скорости исходя из безопасности	17000	15	280
17	Конец спуска	Начинает разгоняться до 80 км/ч	17250	20	238
18	Въезд в н/п	Ограничение скорости 40 км/ч	19450	11	237

Окончание таблицы 6.1

1	2	3	4	5	6
18	Поворот в н/п	Снижение скорости до 30–40 км/ч	20300	11	242
20	Выезд на Р-58		20600	12	243

Далее на основе всего этого проводим более детальный анализ изменения высот и дорожной обстановки с помощью карты и снимков в интернете. Создаем таблицу изменения режима движения автомобиля (таблица 6.2) в виде 2-х параметрической зависимости скорости от пути.

Таблица 6.2. – Изменение режима движения автомобиля по маршруту

Путь, м	Скорость v , м/с	Путь, м	Скорость v , м/с
0	3	5800	12
200	18	6000	12
800	22	6100	12
1600	22	6150	13
2000	20	6200	14.6
2100	15	6300	15
2200	18	6450	12
2500	15	6600	14
2800	18	6450	17
2850	17	6600	20
3200	19.6	6700	14
3600	21	6730	17
3800	18	6800	20
4400	19	7100	22,5
4500	13	7300	25
5000	17	10400	25
5450	16	12200	25
		12500	22.5

Эта информация заносится в компонент V_S2 (n-D Lookup Table). Отметим, что при моделировании в работе применено комплексное задание режима движения с помощью компонента V_S2 (рисунок 6.4) в Simulink в программе модуля S-Function Builder на языке C в виде двухмерной таблицы long tb для реализации торможения (см. приложение К), что является новым подходом.

```
long tb[10][2]={ {1550,7},{3540,11},{5520,11},{12500,25},{12500,25}};
```

Это позволяет реализовать торможение перед препятствиями (“лежащий полицейский”, перекресток, мост и др.) с помощью задаваемых значений в таблице, где первым параметром задается длина пути, на котором происходит торможение, а вторым величина скорости переезда препятствия.

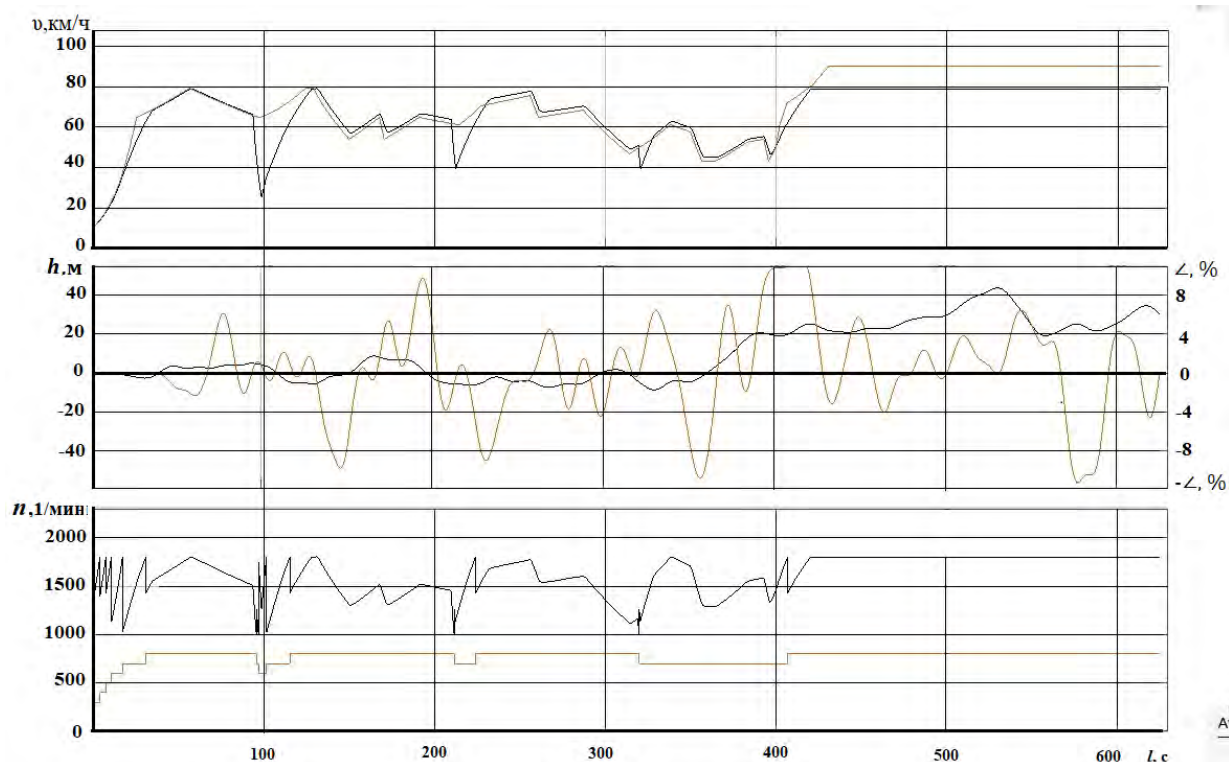


Рисунок 6.4. – Использование полученных данных при моделировании автомобиля

Такой подход позволяет учесть многие факторы, влияющие на динамику автомобиля, оценить работу двигателя и КПП по многим параметрам, а также влияние вибраций

6.2 Оценка эксплуатационных характеристик колесной машины

Для оценки эксплуатационных характеристик колесной машины в таблицу 6.3 сведены данные по результатам моделирования на семи участках дорог.

Таблица 6.3. – Данные результатов моделирования колесной машины

	Минск	Обчак-Догойск	Логойск , Смилов	Смилов ищи – Минск	Сев.кл- МКАД- 2	МКАД- 2- Вишнев	Вишнев ка- Минск
1	2	3	4	5	6	7	8
Длина участка пути, м	23380	103000	31200	35440	12655	8400	9050
Время в пути, с экспер.	1839	5065	1894/35	1989/47	679	621	393

Окончание таблицы 6.3

Средн. скорость, км/ч	45,8	73,2	61,2	65,5	67	48,7	82
Перепад высот, м	20	107	46	76	60	50	27
Средний наклон дороги, $\angle$ %	2,442	2,255	2,51	1,62	3,311	4,29	1,684
Максим. наклон дороги, $\angle$ %	8	8–12	8–13	8	10	6 13	2,6
Кол-во торможений	16	7	13	5, 6	5	9	Нет
Средн. скорость, км/ч экспериментальная	45,8	73,2	61,2	65,5	67	48,7	82
Средняя скорость v , км/ч расчетная	52,05	76,52	63,32	70,25	66,69	50,1	78,0
Средняя скорость v , м/с расчетная	14,46	21,25	17,6	19,51	18,5	13,9	21,7
Максимальная скорость v , км/ч	82,5	91	91	105	80,0	83	97
Расход топлива	75,62	60,35	65,46	58,36	65,7	77,09	47,25
Средн. обороты двигателя, 1/мин	1450	1777	1652	1803	1618	1666	1769
Максим. обороты двигателя, 1/мин	1850	1770	2200	1900	1800	2200	1800
Коэф-т загрузки двигателя	0,2554	0,914	0,3837	0,3371	0,5171	0,210	0,433
Исполь-е. передачи КПП	4-8	4–8	5–8	5–8	6–8	3-8-5	8 6 5

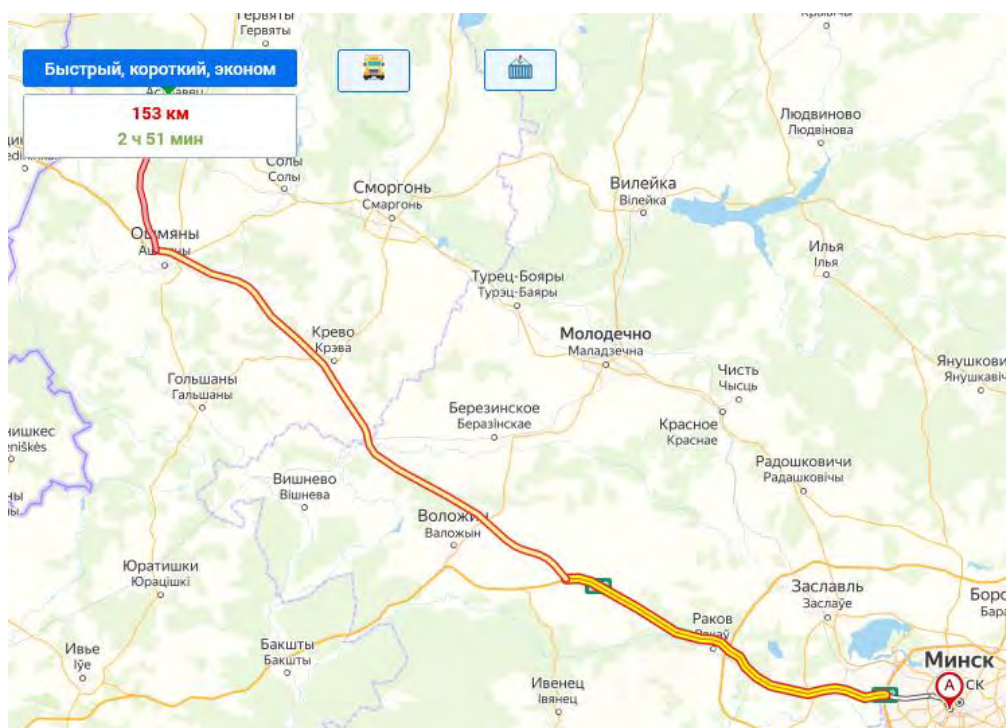


Рисунок 6.5. – Маршрута Минск–Островец

Для маршрута Минск-Новополоцк (рисунок 6.6) время составит $23380/14,46 + 176180/21,25 + 35440/19,51$. В итоге получим $1616,87 + 8291 + 1816,5 = 11724$ с или 3,25 ч, $v = 72,3$ против 54,3 км/ч по сравнению с [92].

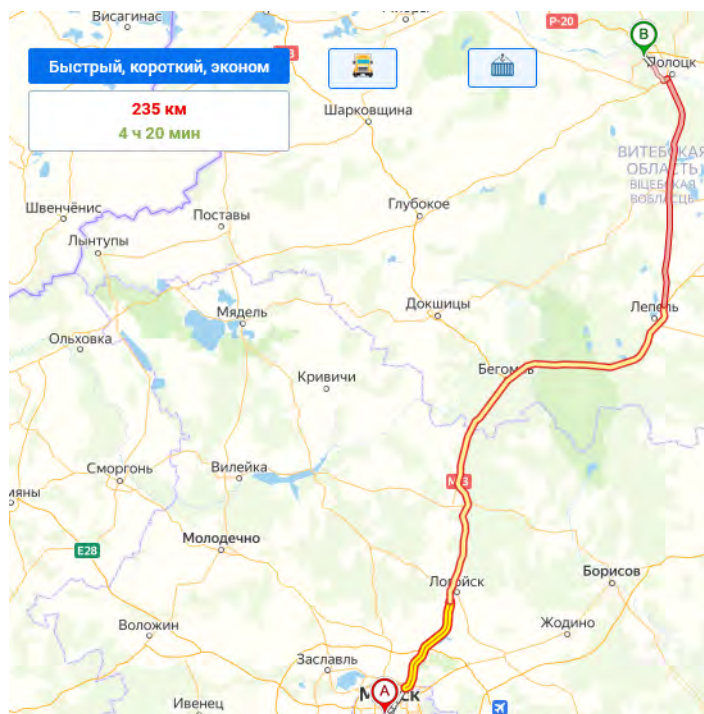


Рисунок 6.6. – Маршрут Минск-Новополоцк

Для маршрута Минск – Гродно ((рисунок 6.7) время составит $23380/14,46 + 219180/21,25 + 35440/19,51$. Получим $1616,87 + 10314,4 + 1816,5 = 13747$ с или 3,81 ч, $v = 72,96$ км/ч против 57,8 км/ч по сравнению с [92].

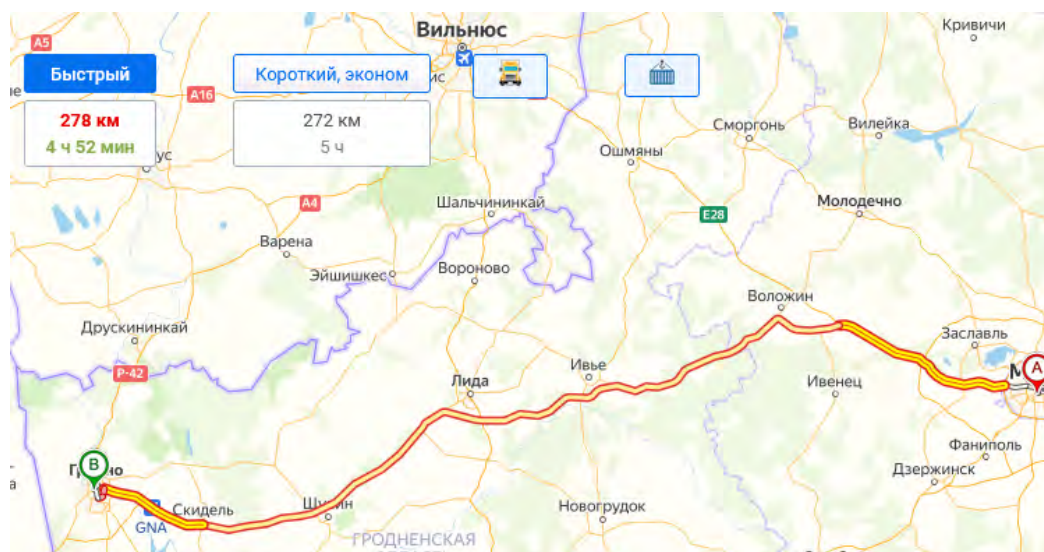


Рисунок 6.7. – Маршрут Минск-Гродно

Для маршрута Минск – Бровары (рисунок 6.8) время составит $23380/14,46 + 514180/21,25 + 35440/19,51$. Получим $1616,87 + 24196,7 + 1816,5 = 27629,7$ с или 7,68 ч, $v=78,5$ против 54,8 км/ч по сравнению с [92].

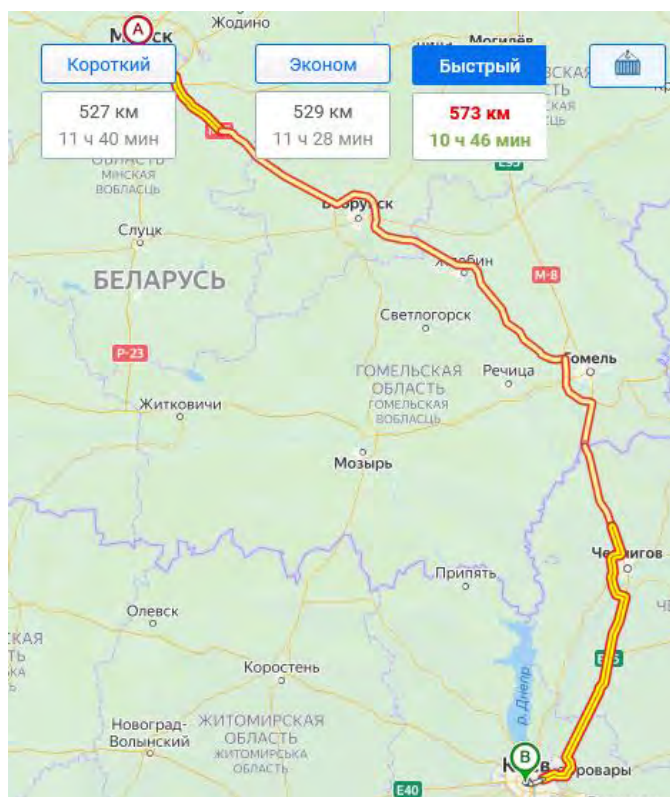


Рисунок 6.8. – Маршрут Минск – Бровары

Для маршрута Минск – Брянск (рисунок 6.9) время составит $23380/14,46 + 514180/21,25 + 35440/19,51$. Получим $1616,87 + 24196,7 + 1816,5 = 27629,7$ с или 7,68 ч, $v=78,5$ против 54,8 км/ч по сравнению с [92].

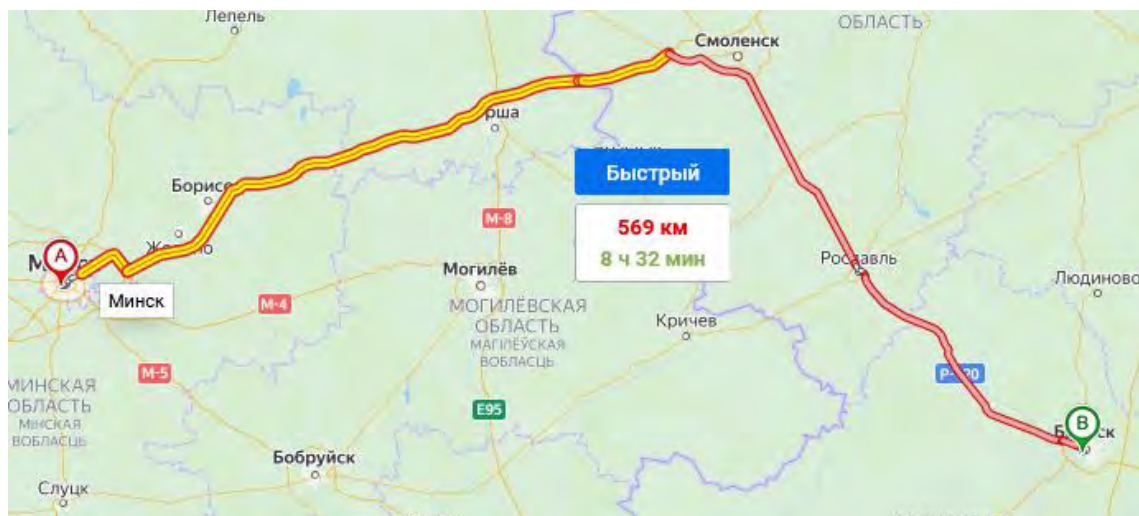


Рисунок 6.9. – Маршрут Минск – Брянск

Для маршрута Минск – Москва (рисунок 6.10) составит) $23380/14,46 + 659180/21,25 + 35440/19,51$ получим $1616,87 + 31020 + 1816,5 = 34453$ с или 9,57 ч 2 ч 8 мин. $v = 75,02$ против 69,3 км/ч по сравнению с [92].

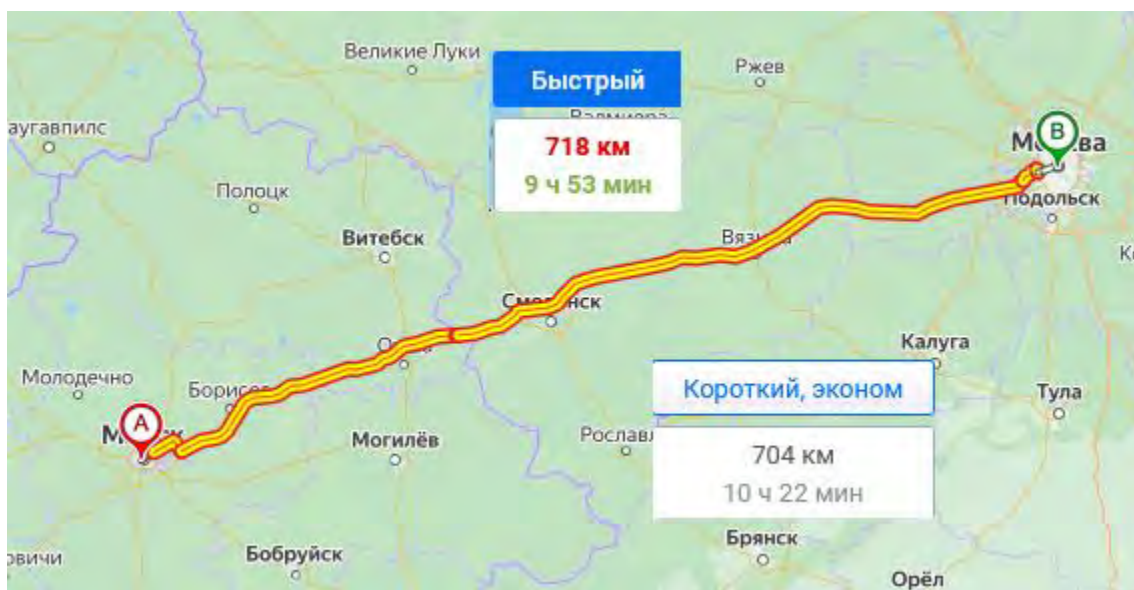


Рисунок 6.10. – Маршрут Минск – Москва

Для маршрут Минск – Воронеж (рисунок 6.11), задавая длину маршрута Минск–Воронеж через параметры скорости участков 2, 3, 4 и изменяя значения длин соответственно $23380/14,46 + 978180/21,25 + 35440/19,51$ получим $1616,87 + 460290,8 + 1816,5 = 49465$ с или 13,74 ч, $v = 75,5$ против 60,3 км/ч по [92].

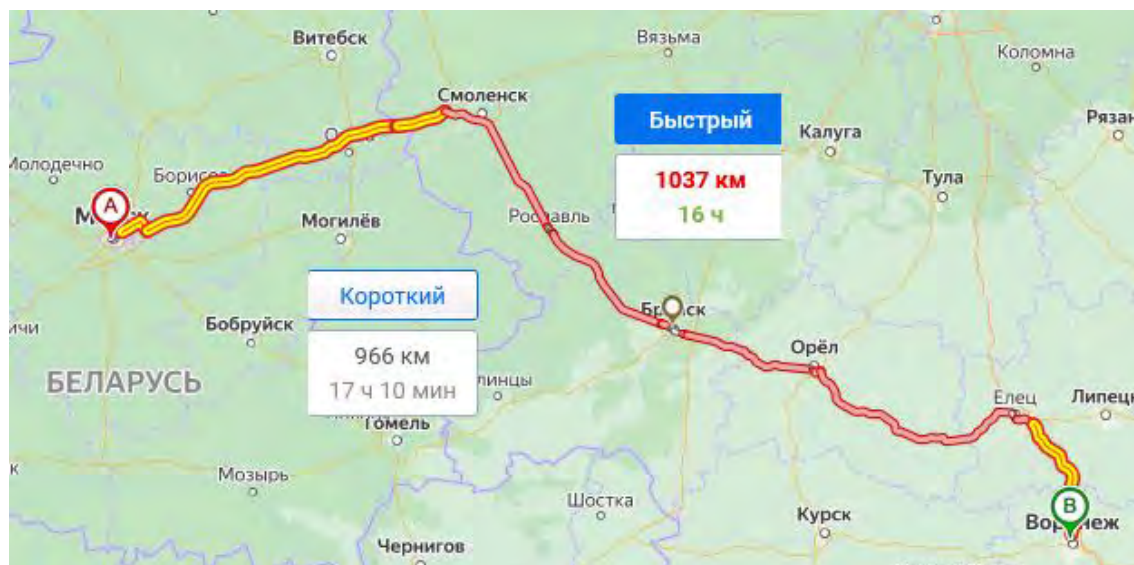


Рисунок 6.11. – Маршрут Минск – Воронеж

Полученные результаты моделирования переброски колесных машин между населенными пунктами сведены в таблицу 6.4.

Таблица 6.4. – Результаты моделирования переброски колесных машин между населенными пунктами.

Параметры	Номер маршрута							
	1	2	3	4	5	6	7	8
Длина пути, км	153	192	235	278	569	573	718	1037
Время в пути расчетное, ч	2,13	2,79	3,81	7,68		7,68	9,57	13,74
Время в пути транспортных компаний, ч	2,85	3,27	4,33	4,87	8,53	10,76	9,88	16
Расчетная скорость колесных машин, км/ч	71,8	59,0	72,3	73,0	78,5	78,5	75	75,5
Скорость у транспортных компаний, км/ч	53,7		54,3	57,8	54,8	54,8	72,7	60

Примечание. Имеются сомнения в корректности данных времени транспортных компаний по маршрутам 5, 7 на фоне других маршрутов. В первом случае оно завышено (что может объяснено прохождением границы), во втором занижено (слишком большая средняя скорость 72,7 км/ч).

На основании ее данных построен график зависимости времени в пути колесных машин от расстояния (рисунок 6.13).

Учитывая, емкость топливных баков 550 л и расход на трассе 60 л/100 км (при $m=20000$ кг) его хватит на 916 км пути, т.е. к полученной зависимости необходимо прибавлять 0,5 ч на заправку после 458 км (половина) пути.

На основании этого графика можно просто рассчитывать необходимое время в пути при перебросе колесных машин специального назначения.

Полученные данные могут также служить основой для планирования расположения точек заправок колесных машин.

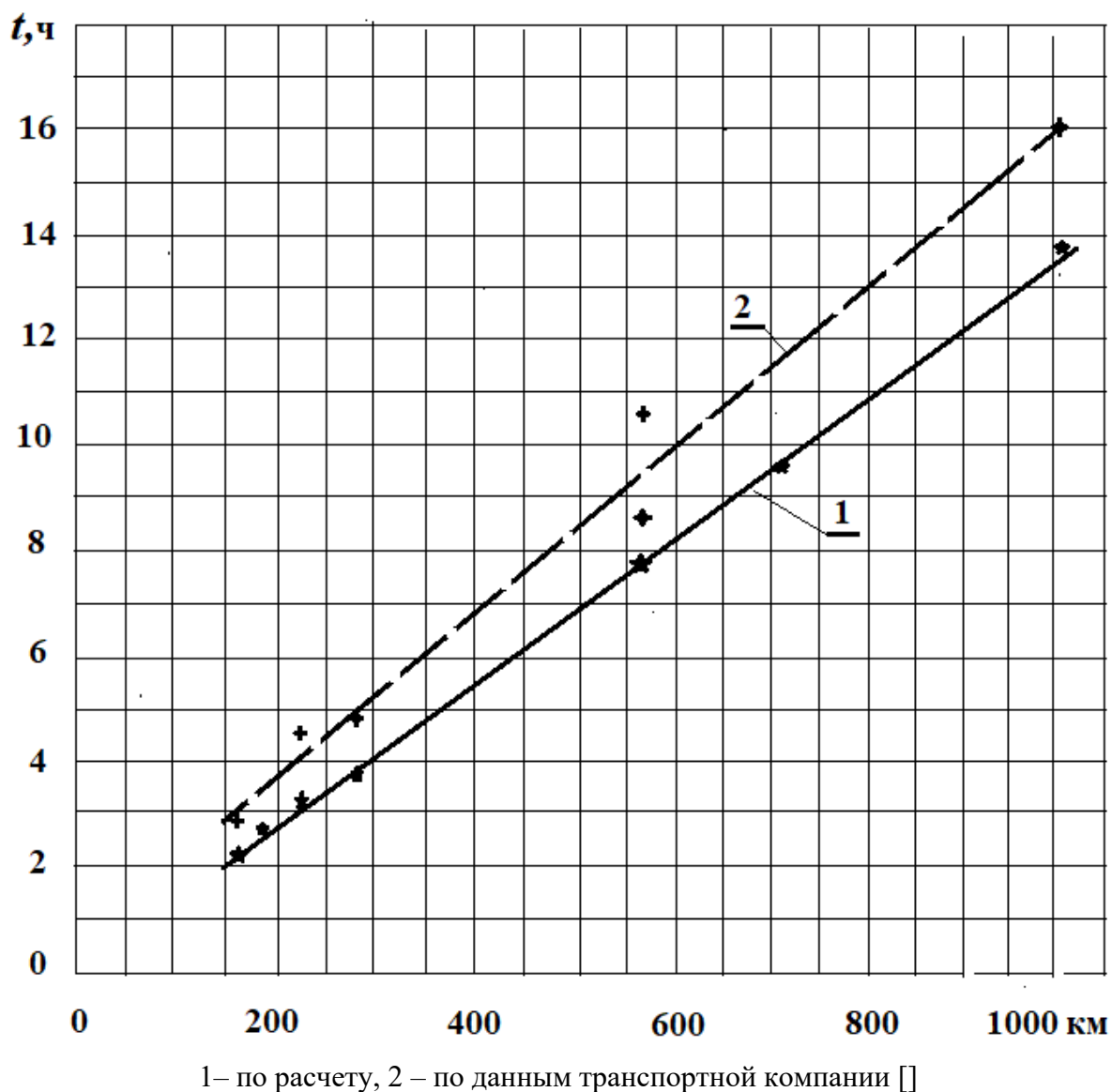


Рисунок 6.13. – Время в пути колесных машин в зависимости от расстояния

Выводы по главе 6

1. Предложен метод прогнозирования движения колесных и специальных машин на основе моделирования движения, радарных и топографических карт, снимков местности в интернете для прокладки незнакомых маршрутов движения и оценки ее параметров, обеспечивающий хорошее совпадение с экспериментом (до 5 %).

2. На основе анализа различных маршрутов движения построен линейный график времени движения от расстояния, свидетельствующий об возможности более меньших затрат времени (высокой скорости передвижения) по сравнению с данными транспортных компаний и обеспечивающий простой способ прогнозирования времени передвижения колесных машин в экстремальных ситуациях.

ГЛАВА 7

МОДЕЛИРОВАНИЕ АВТОМОБИЛЯ СО СМЕШАННОЙ РЕАЛЬНОСТЬЮ С СИМУЛЯТОРОМ

Моделирование автомобиля со смешанной реальностью с симулятором базируется на имитационном моделировании с задействованием испытательного оборудования с водителем со шлемом VR. Реализация MR достигается путем использования части автомобиля или его кабины либо имитатора рабочего места водителя: сиденья, руля, органов управления, установленного на вибростенд для создания ощущения движения у водителя. В первых двух случаях для визуализации могут использоваться проекторы либо мониторы на кабине. В третьем варианте используется шлем виртуальной реальности. Эти решения позволяют охватить в комплексе основные задачи: движения, колебаний, управляемости и эргономики автомобиля (рисунок 7.1) и минимизировать затраты на их проведение.

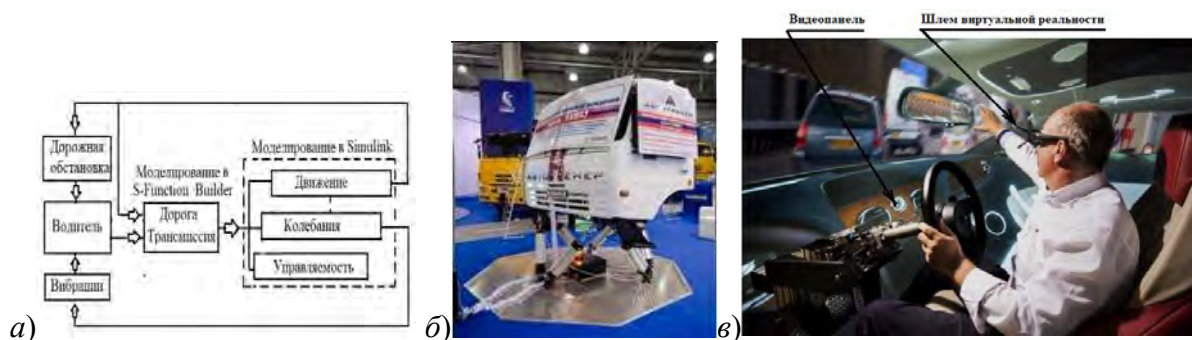


Рисунок 7.1. – Общая схема моделирования основных систем автомобиля (а), использование смешанной реальности при испытаниях (б) и в эргономике Бентли (в)

Для реализации задач имитационного моделирования со смешанной реальностью применяются следующие методы:

- имитационное моделирование на вибростенде или без него с задействованием/без водителя в качестве управляющего звена движения;
- виртуальной/дополненной/смешанной реальности с участием человека в сочетании с имитационным моделированием со стендом или без него.

При компьютерном моделировании из этой схемы исключаются связи водителя по дорожной обстановке и воздействию на него вибраций. Хотя можно вводить ограничения скорости движения от уровня вибраций и помех на дороге, задавая их в Simulink и программе S-Function Builder.

Особенностью имитационного моделирования является задействование водителя для управления на основе визуализации дорожной обстановки на экране компьютера/видеопанели/проектора и вибростенда. Имитационное

моделирование требует использования макетного образца кабины для более полного воссоздания реальной обстановки с органами управления (рулем, педалями газа, тормоза, ручкой переключения КПП) и воздействия вибростенда (рисунок 7.1б). Однако при этом варианте полному ощущению вождения водителя может мешать просматриваемая лабораторная обстановка.

Сейчас для снижения затрат начали шире использоваться методы виртуальной **VR**/дополненной **AR**/смешанной **MR** реальности [112–115, 120, 121, 124], которые представляют собой компьютерную технологию погружения человека в искусственный/моделируемый мир, чтобы усилить атмосферу погружения в рабочую обстановку с одновременной реализацией имитационного моделирования, чтобы создать иллюзию у водителя, что он находится в машине во время движения и управляет ею, осуществляет обслуживание, ремонт машины, прорабатывает ее эргономику [124]. Это достигается путем использования специального шлема с очками (мониторами), ПО, звукового воздействия, “искусственной” руки, вибростенда и моделирования автомобиля. Использование смешанной реальности, как наиболее продвинутой, обусловлено необходимостью учета человеческого фактора, существенно влияющего на режим движения автомобиля, нагруженность его узлов и снижения затрат на симуляторы и исследования автомобиля [10, 11]. Она позволяет заменить многие физические модели фотореалистической смешанной действительностью. В шлеме при MR выводится дорожная обстановка и внутренняя часть кабины машины с органами управления, транслируемая с основного компьютера (рисунок 7.1 б).

Процесс реализации может проводится поэтапно (рисунок 7.2).



Рисунок 7.2. – Этапы эволюционного развития моделирования

Благодаря чему можно путем моделирования основных систем более точно оценить режим движения автомобиля, нагрузки на узлы и далее

провести их ресурсные испытания в стендовых условиях, а также оценить управляемость автомобиля, отработать эргономику рабочего места в движении, не подвергая риску водителя.

Ниже в подразделах приведены результаты имитационного моделирования автомобиля, смешанной реальности и рассмотрены их аспекты.

7.1 Имитационное моделирование автомобиля и рулевого управления

Имитационное моделирование автомобиля и рулевого управления проводилось с использованием блок-схема автомобиля, показанной на рисунке 7.5, где представлены результаты моделирования.

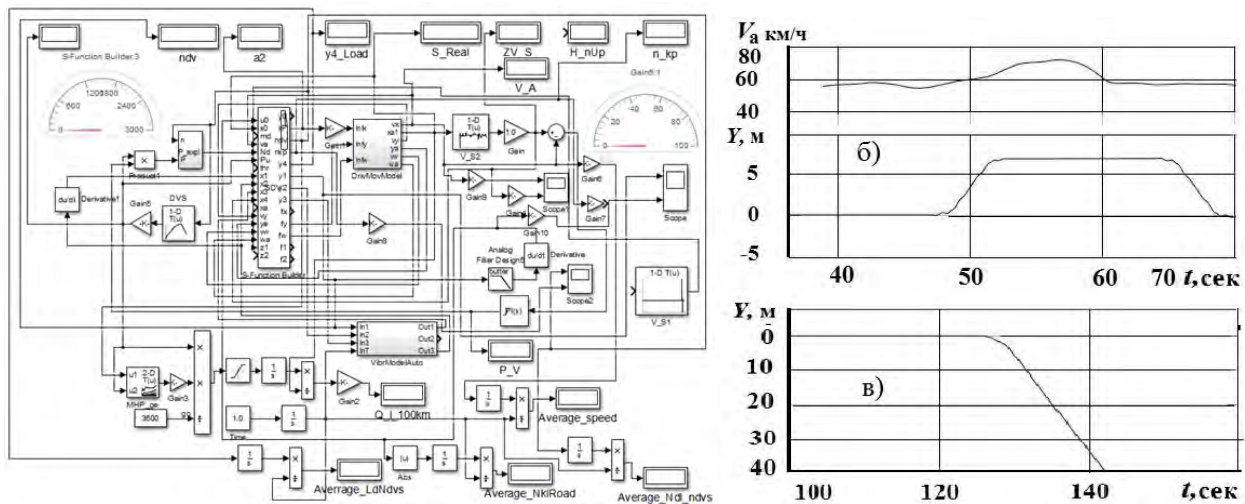


Рисунок 7.5. – Блок-схема реализации в Simulink (а) и результаты моделирования управляемости при выполнении маневров “переставка”(б) и “поворот”(в)

Полная модель включала в себя модели движения, колебаний и управляемости автомобиля с использованием модуля S-Function Builder. В последнем реализована логика управления, задание макро и микропрофиля дороги, связь по сети с АЦП Raspberry, где осуществлено управление автомобилем и визуализация дорожной обстановки. Из-за того, что рулевое управление требует 5 интеграторов пришлось отказаться от модуля моделирования колебаний для обеспечения моделирование в реальном масштабе времени. Эта блок-схема позволяет рассчитать переходной процесс в системе при выполнении маневров автомобиля (“переставка”, “поворот”) при переменной скорости, оценить влияние производительности насоса,

момента сопротивления повороту, номинального давления и других параметров системы.

Проверка корректности модели проведена путем сравнения с результатами поворота колес “на месте”, приведенными в работе [39], которые подтверждены экспериментально. Получено хорошее совпадение (расхождение с последними менее 5 %).

7.2 Выбор реализации имитационного моделирования автомобиля

Реализация имитационного моделирования автомобиля зависит от выбранной схемы и возможности организации. Наиболее простым решением является использование макетного настольного образца (рисунок 7.6а) либо простой конструкции стенда с Aliexpress.com с использованием двух компьютеров (основного PC и мини-компьютера Raspberry Pi 3) и сети Samba [76].

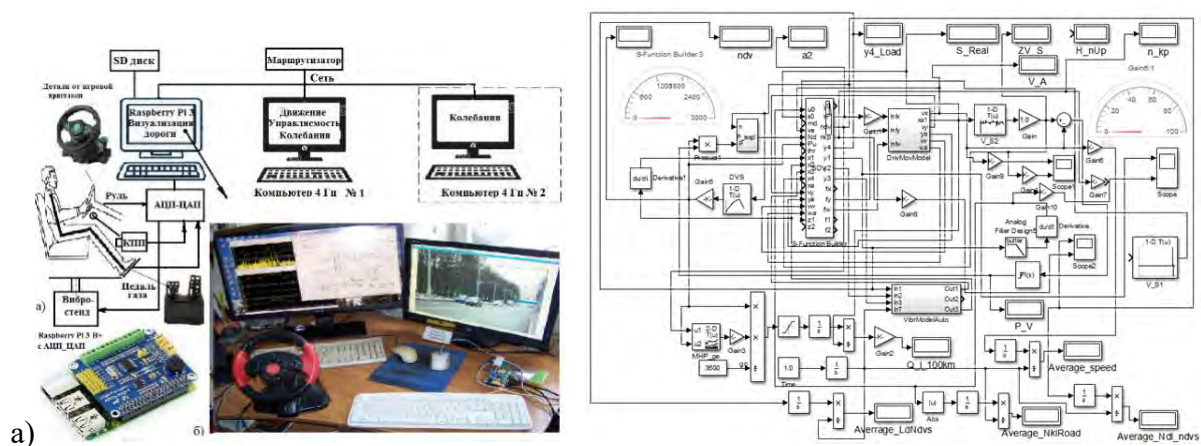


Рисунок 7.6. – Общая схема (а) с использованием Raspberry, игровой приставки и блок-схема на Simulink (б) реализации имитационного моделирования автомобиля

Мини-компьютер Raspberry Pi 3 используется для получения управляющих сигналов от педали газа, КПП и рулевого колеса и визуализации дорожной обстановки (воспроизведения видеозаписи дороги) пропорциональной скорости движения. Визуализация дорожной обстановки в Raspberry Pi 3 осуществлялась с помощью разработанной программы на языке C/C++ (компилятор GCC 10.3) и библиотеки OpenCv версии 4.0.1. Обмен информации между компьютерами при данной схеме осуществлялся по сети с использованием файлового сервера Samba через три бинарных файла на общем (share) SD диске Raspberry с определенной периодичностью (5 Гц) на основе счетчиков циклов, чтобы не снижать быстродействие системы. Новизной предложенного подхода является использование разработанной

программы на C/C++ в модуле S-Function Builder для задания и воспроизведения массива продольного профиля и микропрофиля дороги (96000 точек), поворотов, дорожных условий, реализации многопараметрических зависимостей двигателя, учета передаточных чисел трансмиссии, нелинейных характеристик двигателя, логики переключения передач и обмена информацией, а также использование созданного ПО в Raspberry для просмотра реальной видеозаписи дорожной ситуации с переменной частотой просмотра, а не анимации [1, 2], требующей больших затрат при ее создании и соответствующего программного обеспечения.

Для настольного макетного образца симулятора использовались руль и педаль газа от игровой приставки типа xbox 360 PS2, имеющие потенциометрические датчики (5, 6 кОм), которые запитаны напряжением 5 В. Сигналы напряжений с датчиков подаются на Raspberry Pi 3 через АЦП платы расширения High-Precision AD/DA Board либо ADS1115.

Дальнейшее создание имитационного моделирования требует использования более сложных схем. Наиболее идеальным вариантом было бы использование схемы на рисунке 7.7, справа. Но она требует больших затрат (до 20 мил. евро только для головной части симулятора). Из-за чего для лабораторных исследований более целесообразным является использование первых двух схем (слева) на базе 6-DoF платформы с электроактуаторами Parker или Moog, стоимость которых оценивается в 20...30 тыс.\$.



Рисунок 7.7– Схемы реализации имитационного моделирования автомобиля

Следует отметить сложность в осуществлении моделирования автомобиля и рулевого управления, которая связана с отсутствием реальных характеристик шин и рулевого управления и их математической аппроксимации.

Эту проблему целесообразно решать методом полунатурных испытаний. Для чего необходимо использование стенда с беговыми

барабанами, скорость вращения которых управляется компьютером. Вместо бегового барабана предлагается использовать задний мост с такими же шинами. Это позволяет максимально имитировать процесс реального движения и нагружения автомобиля (рисунок 7.8). В результате моделирования в реальном масштабе времени движения и колебаний автомобиля [14], показа синхронного видео дорожной обстановки, задается режим движения водителем исходя из тягово-скоростных характеристик машины, передаваемых через ЦАП на стенд скорости вращения колес, а также нагружение рамы над передним мостом. Водитель управляет рулем, осуществляет повороты колес. Получаемые с датчиков сигналы усилий и моментов с тензоусилителя подаются обратно через АЦП в модель, где осуществляется реагирование на них. Такой подход позволяет получить данные о реальных характеристиках шины, рулевого управления и реагирование рулевого управления на воздействие водителя.



Рисунок 7.8. – Полунатурные испытания автомобиля и рулевого управления

В этом случае нет необходимости в математическом задании характеристик: они фиксируются аппаратурой на стенде и передаются через цепочку АЦП–Raspberry–сеть в модель Simulink. Необходимость использования такого стенда с беговыми барабанами обуславливается тем, что при вращении характеристики шин могут значительно отличаться от статических и сложно математически описать динамическую работу усилителя руля. Такой подход позволяет значительно упростить проведение имитационных испытаний рулевого управления и является новым не только к

последнему, но и к всему автомобилю в целом так как здесь задействована еще и заземленная рама сзади.

Предлагаемый метод полунатурных испытаний позволяет не только смоделировать движение автомобиля, но и получить данные о нагруженности его основных систем и узлов: подвески, трансмиссии, рулевого управления, рамы и кабины. Фактически с помощью двух гидропульсаторов можно воспроизвести то, что реализуется шестью пульсаторами в классическом варианте симулятора. И стоимость такого стенда значительно дешевле, чем на симуляторе с рельсами. И такая реализация испытаний по средствам нашим заводам, имеющим такие стенды (МАЗ, МЗКТ, БелАЗ, КАМАЗ, ВАЗ). Используя предложенный метод можно сократить значительное отставание от зарубежных автопроизводителей.

7.3 Использование 6-DOF платформы для моделирования в Simulink и испытаний ТС

7.3.1 Для проведения полунатурных испытаний автомобиля целесообразно использование 6-DOF платформы с электроактуаторами, как более дешевого варианта по сравнению с гидропульсаторами, что подтверждается зарубежным опытом (рисунок 7.9).



Рисунок 7.9. – Конструкция платформы, симуляторы самолета, автомобиля, танка

В основе платформы лежит прямая кинематика механизма Stewart [107], которая определяет положение и ориентацию верхней части платформы по значениям перемещений 6-ти цилиндров.

Прямой анализ кинематики платформы Stewart является чрезвычайно сложным из-за 6-ти степеней свободы, его структурной связанности, которая обеспечивает ее механическую жесткость. Прямая кинематика механизма Stewart была изучена в ряде научных исследованиях, тем не менее, до конца приемлемое решение не найдено [108–110 78–80]. Из-за чего многие

прибегают к ее упрощению – используют кинематику механизмов 3-х степеней свободы, где уравнение получаются более простыми. Производители выпускаемых платформ не раскрывают алгоритма реализации движения, являющегося их Know how.

Для выбора платформы необходимо располагать данными об усилиях, перемещениях испытуемого объекта. Их можно получить путем моделирования движения платформы и на их основе подобрать подходящую платформу.

7.3.2 Кинематика 6-DOF платформы

7.3.2.1 Система координат

В работе принята следующая схема, соответствующая осям (X - Y - Z), установленным в центре опорной плиты в плоскости с осью Y , направленной вертикально вверх. Схема 6-Dof платформы Stewart представлена на рисунке 7.10. Оси X - Z установлены согласно правилу правой руки.

Математическая модель должна обеспечивать желаемую траекторию путем перемещения мобильной платформы в заданное положение с ориентированием за счет действие сил актуаторов (рисунок 7.11).

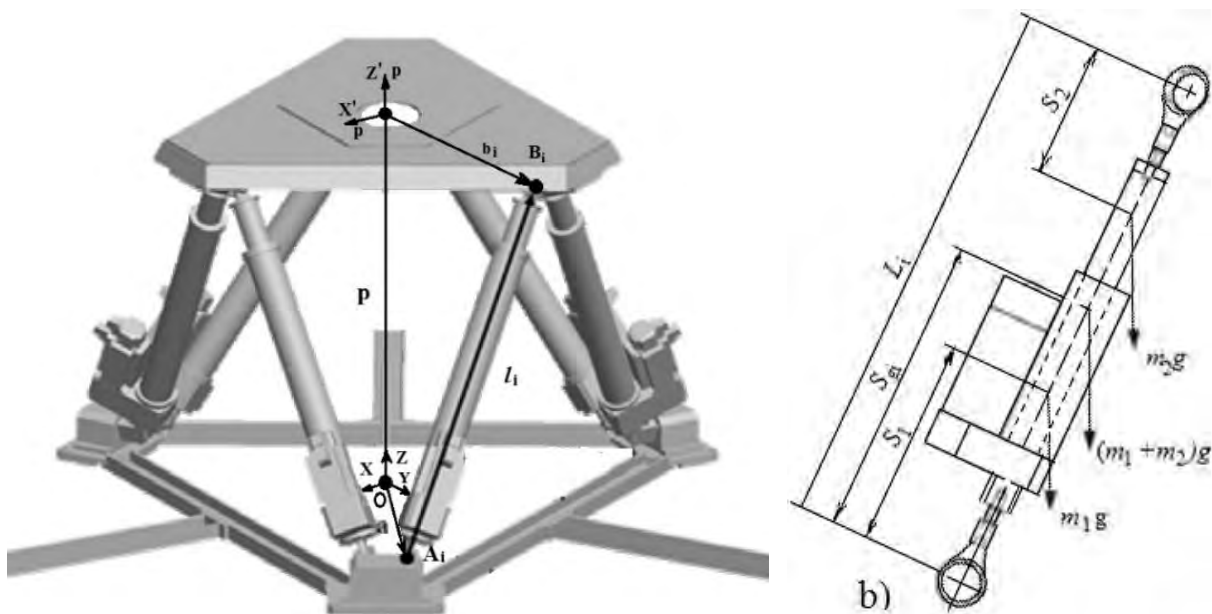
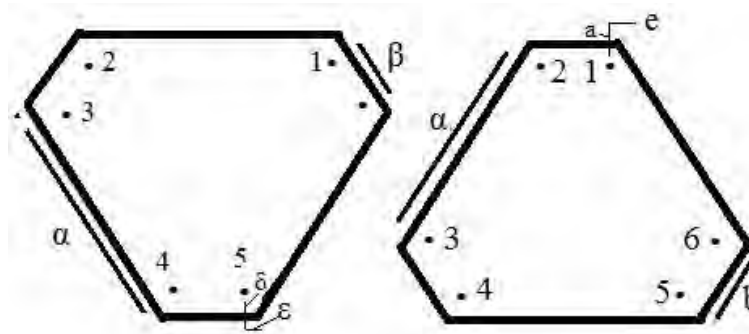


Рисунок 7.10. –Схемы 6-Dof платформы Stewart (a) и актуатора (b)



а) Нижнее неподвижное основание, б) движущая часть платформы

Рисунок 7.11. – Геометрическая модель платформы - контрольные точки приводов.

7.3.2.2 Математическое описание

В общем виде математическую модель можно описать следующим выражением

$$X_i = f(L_i) \quad (7.1)$$

где $L_i = (L_1 \ L_2 \ \dots \ L_6)$ объединенное линейное положение от актуаторов, $X_i = (X \ Y \ Z \ \psi \ \theta \ \varphi)$ вектор ориентации положения точки платформы.

Тогда матрица преобразования для вращений может быть представлена как

$$T(\psi, \theta, \varphi) = \text{rot}(x, \varphi) \text{rot}(z, \theta) \text{rot}(z, \psi) = s\varphi \begin{bmatrix} c\varphi \ c\theta & -c\varphi s\theta & -c\varphi s\theta s\psi - s\varphi c\psi & c\varphi s\theta s\psi + s\varphi s\psi \\ c\varphi \ c\theta & -s\varphi s\theta s\psi + c\varphi c\psi & s\varphi s\theta c\psi - c\varphi s\psi & c\theta \ s\psi \\ -s\theta & & c\theta \ s\psi & c\theta \ c\psi \end{bmatrix} \quad (7.2)$$

$$\text{где } \theta = \text{ATAN2} \left[\frac{-n_z}{c\varphi n_z + s\varphi n_y} \right], \varphi = \text{ATAN2} \left[\frac{n_y}{n_z} \right], \psi = \text{ATAN2} \left[\frac{s\varphi a_z - c\varphi a_x}{s\varphi s_z + c\varphi s_y} \right],$$

а $n = [n_x \ n_y \ n_z]$, $s = [s_x \ s_y \ s_z]$, $a = [a_x \ a_y \ a_z]$ – ортонормальные векторы, которые описывают ориентацию платформы.

где $c\varphi$: $\cos \varphi$, $s\varphi$: $\sin \varphi$;

T является координатной матрицей преобразования от движущегося рамы $x' \ y' \ z'$ платформы до инерционной нижней рамы $X \ Y \ Z$.

Данная матрица преобразования позволяет изменять положение каждого привода для определения нового состояния модели кинематики, показанной на рисунке 7.10.

Точки, определяющие движение подвижной верхней части, расположены в концевых точках шести линейных приводов, зафиксированных на более низком остова платформы. При предположении, что приводы находятся в достигнутом их текущем положении и ориентированы, решение заключается в вычислении координат центра масс при исходной начальной

базе и углов ориентации RPY (крен/Roll, тангаж/Pitch, отклонением от курса/Yaw).

Текущие положения могут быть вычислены от относительного положения и ориентации (путем использования матрицы преобразования), выполняя расчеты для новых условий в рабочей области платформы.

Позиционный вектор для верхних/нижних частей актуатора p_i, p_s , определяется относительно фиксированной ссылочной системы центра масс нижней части, как приведено в (7.3).

Параметры $\alpha, \beta, \delta, \varepsilon, a, b, d, e$ показанные на рисунках 7.10, 7.11, h представляет положение их центра массы верхнего основания в ее начальной конфигурации и каждой линии, P_i, P_s представленной ниже ($A_1 \dots A_6$), и выше ($B_1 \dots B_6$) скоординированных оконечностей приводов.

$$P_i = \begin{bmatrix} A_i + \varepsilon & D_i - \delta & 0 \\ -A_i + \varepsilon & D_i - \delta & 0 \\ -A_i + \varepsilon + C_i & D_i - \delta + C_i & 0 \\ -B_i + \varepsilon & D_i + \delta & 0 \\ B_i + \varepsilon & -D_i + \delta & 0 \\ A_i - \varepsilon - C_i & -D_i - \delta + C_i & 0 \end{bmatrix} \quad P_s = \begin{bmatrix} A_s + e & D_s - d & h \\ -A_s + e & D_s - d & h \\ -A_s + e + C_s & -D_s - d + C_s & h \\ -B_s + e & D_s + d & h \\ B_s + e & -D_s + d & h \\ A_s - e - C_s & D_s - d + C_s & h \end{bmatrix} \quad (7.3)$$

где $A_i = 0,5a$, $A_s = 0,5b$, $B_i = 0,5\beta$, $B_s = 0,5a$, $C_i = 2(\varepsilon - B_i)\cos(t)$, $C_s = 2(e - B_s)\cos(t)$, $D_i = (A_i + B_i)\cos(t)$, $D_s = (A_s + B_s)\cos(t)$,

Каждый актуатор ассоциирован с позиционирующим вектором $\underline{X}_i$ его нижнего конца и значением его растяжения, связанного с i -ым актуатором. В матрице преобразования, $\underline{X}_i^T$ представляет новую ассоциированную позицию вектора i -го актуатора для каждого верхнего i -го положения, приведенного в (7.4)

$$\underline{X}_i = T(\psi \theta \varphi) \underline{X}_i^T \quad (7.4)$$

Координаты конечных точек верхней плиты при известном положении могут быть вычислены с помощью предыдущих уравнений. Это приводит к новым данным, которые соответствует новому размеру привода. Если X_0 является контрольной точкой, то различие между текущими размерами и целевыми является растяжение, которое должно быть соответствовать каждому актуатору для достижения его нового положения, как представлено в (7.5)

$$\Delta L = \left| \underline{X_i^T} - \underline{X_0} \right| \left| \underline{X_i} - \underline{X_0} \right|. \quad (7.5)$$

Таким образом расстояние между конечными точками осей вычисляется с помощью матрицы преобразования и известных координат. Кинематическая модель платформы получает передающую информацию в векторной форме и вращении от матрицы с углами RPY.

Данный анализ позволяет вычислить длины каждой оси так, чтобы платформа переместилась в заданное положение, как требуется, чтобы каждый k линейный актуатор, подключенный к верхней мобильной плите до и после того, имел перемещение, описанное в выражении 7.6 и 7.7.

$$L = \sqrt{\sum_{j=1}^3 (P_s^{kj} - P_i^{kj})^2} \quad \text{при } k=1 \dots 6 \quad (7.6)$$

$$L + \Delta L = \sqrt{\sum_{j=1}^3 (T_j^{-1}(\psi, \theta, \varphi) P_s^{kj} - P_i^{kj})^2} \quad \text{при } k=1 \dots 6 \quad (7.7)$$

Связи платформы определяются:

$$A_i = \left| r_p \cos(a_i), r_p \sin(a_i), 0 \right|^\tau \left| A_{ix}, A_{iy}, A_{iz} \right|^\tau, a_i = \frac{i\pi - a_p}{2} \text{ for } i=1,3,5$$

$$a_i = a_{i-1} + a_p \text{ for } i=2,4,6 \quad (7.8)$$

а связи основания:

$$B_i = \left| r_b \cos(\beta_i), r_b \sin(\beta_i), 0 \right|^\tau = \left| B_{ix}, B_{iy}, B_{iz} \right|^\tau, \beta_i = \frac{i\pi - a_b}{2} \text{ for } i=1,3,5$$

$$\beta_i = \beta_{i-1} + a_b \text{ for } i=2,4,6 \quad (7.9)$$

где r_p – радиус платформы; r_b – радиус основания; a_p – угол платформы и a_b – угол основания.

7.4 Кинематическая модель платформы Stewart

Электроактуаторы платформы Stewart меняют его положение и ориентацию как функцию длины линейного актуатора. Рисунок 7.12 показывает соответствующую геометрическую модель, просматриваемую от вершины, где нижняя геометрия основания сформирована B_1 к точкам B_6 а для верхних A_1 к точкам A_6 .

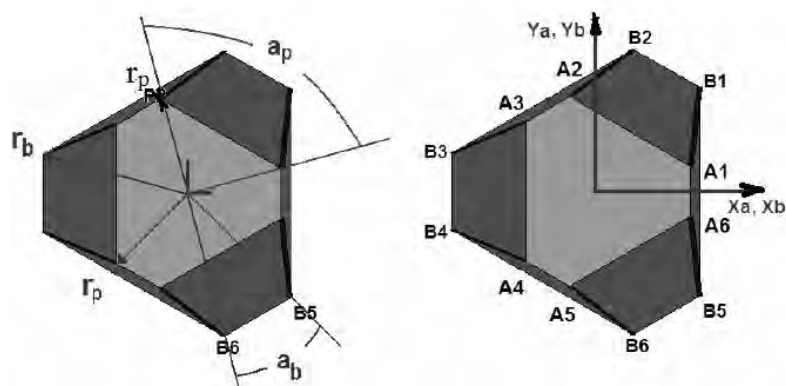


Рисунок 7.12. – Геометрическая модель платформы Stewart

7.5.1 Инверсная кинематика

Модель инверсной кинематики актуатора выражает объединенное линейное движение его как функцию положения и ориентирования при фиксированной системе координат для платформы, как представлено в выражении 7.10:

$$l=f(x), \quad (7.10)$$

где $l = (l_1, l_2, l_3, l_4, l_5, l_6)$ линейное положение соединений, $x = (X, Y, Z, \psi, \theta, \varphi)$ – позиционирующий вектор платформы, X, Y, Z – декартово положение, и ψ, θ, φ – представляет угловую ориентацию платформы. Ссылочные системы фиксированы к (u, v, w) и $B(x, y, z)$ в основании, как показано на рисунке 7.13.

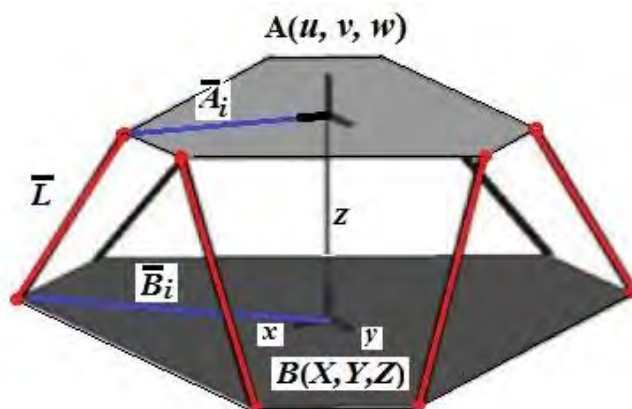


Рисунок 7.13. – Векторное представления манипулятора.

Преобразование для средней точки/центроида мобильной платформы к основанию, описывается позиционирующим вектором x и матрицей вращения ${}^B R_A$, где,

$${}^B R_A = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \quad (7.11)$$

Угловые движения выражены как угловые вращения Euler относительно осей X , Y , Z , т.е. креном/Roll, тангажем/Pitch, рысканием/Yaw в последовательности.

$${}^B R_A = \begin{bmatrix} c\varphi c\phi & c\psi s\phi s\theta - s\psi c\theta & c\psi s\phi c\theta + s\psi s\phi c\theta \\ c\varphi s\phi & s\psi s\phi s\theta + c\psi c\theta & s\psi s\phi c\theta - c\psi s\theta \\ -s\phi & c\phi c\theta & c\phi s\theta \end{bmatrix} \quad (7.12)$$

Уравнение векторного цикла для i -го манипулятора имеет следующий вид:

$$l_i = {}^A R_B + x - B_i \quad (7.13)$$

Путем замены условий для каждого актуатора выражение (7.14) будет описывать движение платформы относительно его нижней части.

$$l_i^2 = X^2 + Y^2 + Z^2 + r_p^2 + r_b^2 + 2(r_{11}A_{ix} + r_{12}A_{iy})(X - B_{ix}) + 2(r_{21}A_{ix} + r_{22}A_{iy})(Y - B_{iy}) + 2(r_{31}A_{ix} + r_{23}A_{iy})(Y - B_{iy}) + 2Z(r_{31}A_{ix} + r_{23}A_{iy}) - 2(XB_{ix} + YB_{iy}) \quad (7.14)$$

7.4.2 Уравнения динамики

При выводе уравнений динамики полагаем, что нет никакого вращения ни на каком звене его собственной оси из-за сферических соединений. Кинематику и динамику для каждого звена рассматриваем и вычисляем как приложенную силу к сферическому соединению на его вершине.

Динамическая модель платформы 6-DOF может быть вычислена методом Эйлера-Лагранжа, который выражает обобщенный крутящий момент. Динамическая модель, описывается рядом дифференциальных уравнений, названных уравнениями динамики движения как показано в (7.15).

$$\tau_i = J_i \ddot{L}_i + F_i \dot{L}_i + G_i \quad \text{для } i=1 \dots 6 \quad (7.15)$$

где $\tau_i(t)$ – является обобщенным вектором крутящего момента, $\dot{L}_i(t)$ – обобщенный вектор в сечении (линейного соединения), $J_i(t)$ – инерционная матрица, $F_i(t)$ – матрица нелинейных сил (например, центробежных), $G_i(t)$ – матрица гравитационной силы.

7.4.3 Модель актуатора

Каждое звено актуатора, состоящее из двигателя постоянного тока, системы передачи и управления, можно описать классическими уравнениями, представленными (7.16)

$$\begin{aligned} u(t) &= L_{mot} \frac{di(t)}{dt} + R_{mot}(t) + K_E \frac{d\theta_m(t)}{dt} \\ T_m(t) &= J_{eg} + \frac{d^2\theta_m(t)}{dt} + B_{eg} \frac{d\theta_m(t)}{dt} = K_T i(t) \end{aligned} \quad (7.16)$$

где $T_m(t)$ является крутящим моментом, θ_m – угловое положение моторной оси, $i(t)$ – ток, L_{mot} , R_{mot} – соответственно индуктивность, сопротивление, J_{eg} , B_{eg} – инерция, осевая нагрузка, рассчитанные со стороны двигателя.

7.5 Моделирование 6-DOF платформы в Simulink

7.5.1 Для моделирования 6-DOF платформы пакет MATLAB/Simulink предлагает готовое решение в виде ряда программ (.slx, .m) (рисунок 7.14).

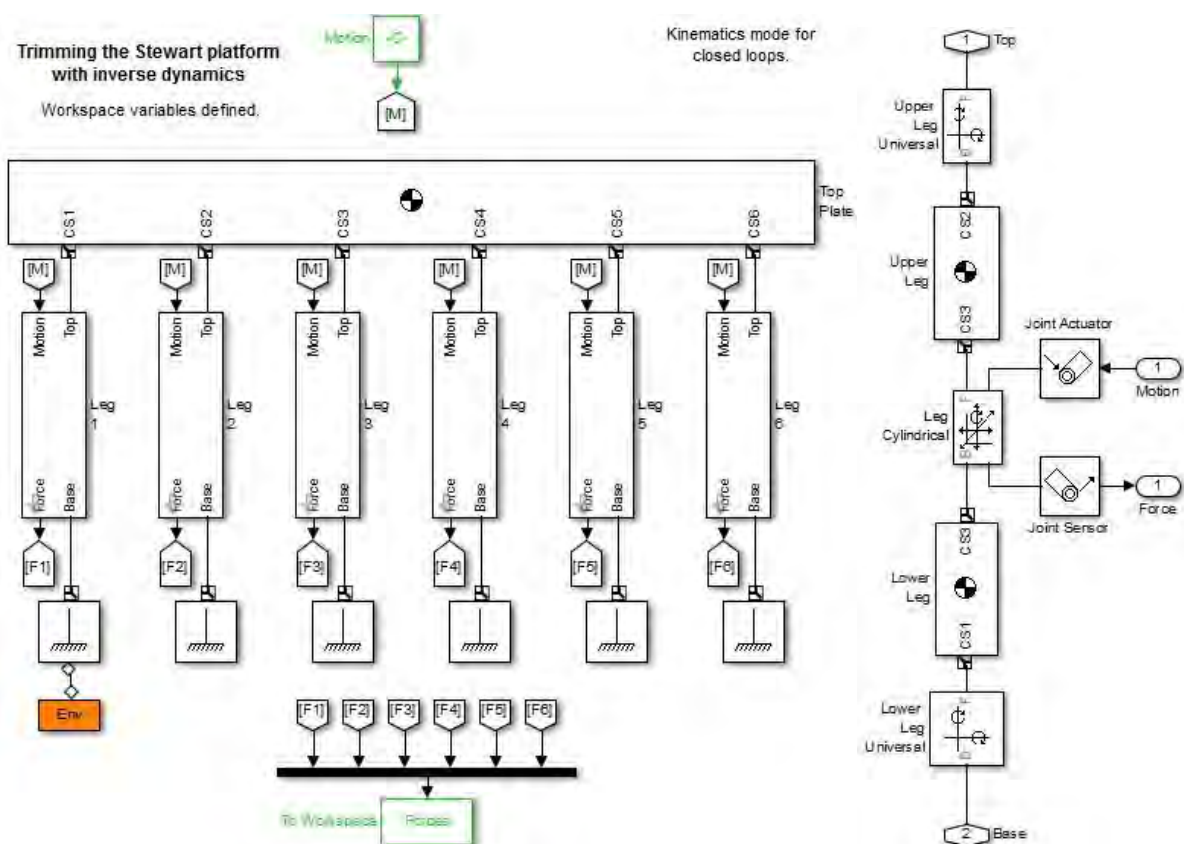


Рисунок 7.14. – Общая блок-схема моделирования движения платформы в Simulink и компоненты приводного звена

Они находятся в каталоге C:\ProgramFiles\MATLAB\R2015b\toolbox\physmod\mech\mechdemos. Эти программы позволяют определить усилия, которые должны развивать актуаторы при заданных перемещениях платформы по осям X , Y , Z и углам поворота вокруг них на основе кинематических, массовых параметров платформы и ее звеньев (рисунок 7.14). Кинематические и массовые параметры платформы задаются в файле `mech_stewart_studies_setup.m`, который должен предварительно запущен в Matlab, чтобы инициализировать переменные (приложение Л).

Однако в Simulink S-Function Builder они не работают. Для работы в нем с помощью сайта (<https://products.codeporting.ai/ru/convert/matlab-to-cpp>) осуществлена конвертация исходного кода из Matlab в C/C++ с использованием искусственного интеллекта ИИ А1.

```
#include <iostream>
#include <vector>
#include <cmath>
```

```
const double deg2rad = M_PI / 180.0;
const std::vector<double> x_axis = {1.0, 0.0, 0.0};
const std::vector<double> y_axis = {0.0, 1.0, 0.0};
const std::vector<double> z_axis = {0.0, 0.0, 1.0};
const std::vector<double> null_length = {0.0, 0.0, 0.0, 0.0, 0.0, 0.0};
```

```
// Connection points on base and top plate w.r.t. World CS at the center of the base plate
std::vector<std::vector<double>>> pos_base;
std::vector<std::vector<double>>> pos_top;
const double alpha_b = 2.5 * deg2rad; // +/- offset angle from 120 degree spacing on base
const double alpha_t = 10 * deg2rad; // +/- offset angle from 120 degree spacing on top
const double height = 2.0; // height in meters
const double radius_b = 3.0; // base radius in meters
const double radius_t = 1.0; // top radius in meters
```

```
for (int i = 1; i <= 3; ++i) {
    // base points
    double angle_m_b = (2 * M_PI / 3) * (i - 1) - alpha_b;
    double angle_p_b = (2 * M_PI / 3) * (i - 1) + alpha_b;
    pos_base.resize(2 * i);
    pos_base[2 * i - 1] = {radius_b * cos(angle_m_b), radius_b * sin(angle_m_b), 0.0};
    pos_base[2 * i] = {radius_b * cos(angle_p_b), radius_b * sin(angle_p_b), 0.0};

    // top points
    // Top points are 60 degrees offset
    double angle_m_t = (2 * M_PI / 3) * (i - 1) - alpha_t + 2 * M_PI / 6;
    double angle_p_t = (2 * M_PI / 3) * (i - 1) + alpha_t + 2 * M_PI / 6;
    pos_top.resize(2 * i);
    pos_top[2 * i - 1] = {radius_t * cos(angle_m_t), radius_t * sin(angle_m_t), height};
    pos_top[2 * i] = {radius_t * cos(angle_p_t), radius_t * sin(angle_p_t), height};
}
```

```

}

// Permute pos_top points so that legs are end points of base and top points
// 6th point on top connects to 1st on bottom
pos_top = {pos_top[5], pos_top[0], pos_top[1], pos_top[2], pos_top[3], pos_top[4]}; // 6th point on
top connects to 1st on bottom

// Compute points w.r.t. to the body frame in a 3x6 matrix
std::vector<std::vector<double>>> body_pts(3, std::vector<double>(6));
for (size_t j = 0; j < 6; ++j) {
    body_pts[0][j] = pos_top[j][0];
    body_pts[1][j] = pos_top[j][1];
    body_pts[2][j] = pos_top[j][2] - height;
}

// Leg vectors
std::vector<std::vector<double>>> legs(6, std::vector<double>(3));
std::vector<double> leg_length(6);
std::vector<std::vector<double>>> leg_vectors(6, std::vector<double>(3));

for (int i = 0; i < 6; ++i) {
    legs[i] = {pos_top[i][0] - pos_base[i][0], pos_top[i][1] - pos_base[i][1], pos_top[i][2] -
pos_base[i][2]};
    leg_length[i] = sqrt(pow(legs[i][0], 2) + pow(legs[i][1], 2) + pow(legs[i][2], 2));
    leg_vectors[i] = {legs[i][0] / leg_length[i], legs[i][1] / leg_length[i], legs[i][2] / leg_length[i]};
}

// Calculate revolute and cylindrical axes
std::vector<std::vector<double>>> rev1(6, std::vector<double>(3));
std::vector<std::vector<double>>> rev2(6, std::vector<double>(3));
std::vector<std::vector<double>>> cyl1(6, std::vector<double>(3));

for (int i = 0; i < 6; ++i) {
    rev1[i] = {leg_vectors[i][1] * z_axis[2] - leg_vectors[i][2] * z_axis[1],
        leg_vectors[i][2] * z_axis[0] - leg_vectors[i][0] * z_axis[2],
        leg_vectors[i][0] * z_axis[1] - leg_vectors[i][1] * z_axis[0]};
    double norm_rev1 = sqrt(pow(rev1[i][0], 2) + pow(rev1[i][1], 2) + pow(rev1[i][2], 2));
    rev1[i] = {rev1[i][0] / norm_rev1, rev1[i][1] / norm_rev1, rev1[i][2] / norm_rev1};

    rev2[i] = {-rev1[i][1] * leg_vectors[i][2] + rev1[i][2] * leg_vectors[i][1],
        rev1[i][2] * leg_vectors[i][0] - rev1[i][0] * leg_vectors[i][2],
        rev1[i][0] * leg_vectors[i][1] - rev1[i][1] * leg_vectors[i][0]};
    double norm_rev2 = sqrt(pow(rev2[i][0], 2) + pow(rev2[i][1], 2) + pow(rev2[i][2], 2));
    rev2[i] = {rev2[i][0] / norm_rev2, rev2[i][1] / norm_rev2, rev2[i][2] / norm_rev2};

    cyl1[i] = leg_vectors[i];
}

// Coordinate systems
struct Leg {
    std::vector<double> origin;

```

```

    std::vector<double> rotation;
    std::vector<double> end_point;
};

std::vector<Leg> lower_leg(6);
std::vector<Leg> upper_leg(6);

for (int i = 0; i < 6; ++i) {
    lower_leg[i].origin = {pos_base[i][0] + (3.0 / 8.0) * legs[i][0], pos_base[i][1] + (3.0 / 8.0) *
    legs[i][1], pos_base[i][2] + (3.0 / 8.0) * legs[i][2]};
    lower_leg[i].end_point = {pos_base[i][0] + (3.0 / 4.0) * legs[i][0], pos_base[i][1] + (3.0 / 4.0) *
    legs[i][1], pos_base[i][2] + (3.0 / 4.0) * legs[i][2]};
    lower_leg[i].rotation = {rev1[i][0], rev2[i][0], cyl1[i][0], rev1[i][1], rev2[i][1], cyl1[i][1], rev1[i][2],
    rev2[i][2], cyl1[i][2]};
    upper_leg[i].origin = {pos_base[i][0] + (1.0 - 3.0 / 8.0) * legs[i][0], pos_base[i][1] + (1.0 - 3.0 / 8.0)
    * legs[i][1], pos_base[i][2] + (1.0 - 3.0 / 8.0) * legs[i][2]};
    upper_leg[i].end_point = {pos_base[i][0] + (1.0 / 4.0) * legs[i][0], pos_base[i][1] + (1.0 / 4.0) *
    legs[i][1], pos_base[i][2] + (1.0 / 4.0) * legs[i][2]};
    upper_leg[i].rotation = {rev1[i][0], rev2[i][0], cyl1[i][0], rev1[i][1], rev2[i][1], cyl1[i][1], rev1[i][2],
    rev2[i][2], cyl1[i][2]};
}

// Inertias and masses
const double top_thickness = 0.05;
const double base_thickness = 0.05;
const double inner_radius = 0.03;
const double outer_radius = 0.05;
const double density = 76e3 / 9.81; // steel density in Kg/m^3

// Upper and lower leg inertia and mass
double lower_leg_mass, lower_leg_inertia;
double upper_leg_mass, upper_leg_inertia;
// Assume inertiaCylinder is a function that computes inertia and mass
std::tie(lower_leg_mass, lower_leg_inertia) = inertiaCylinder(density, 0.75 * leg_length[0],
    outer_radius, inner_radius);
std::tie(upper_leg_mass, upper_leg_inertia) = inertiaCylinder(density, 0.75 * leg_length[0],
    outer_radius, 0);

// Top and base plate mass and inertia
double top_mass, top_inertia;
double base_mass, base_inertia;
std::tie(top_mass, top_inertia) = inertiaCylinder(density, top_thickness, radius_t, 0);
std::tie(base_mass, base_inertia) = inertiaCylinder(density, base_thickness, radius_b, 0);

// Reference motion, dynamical, and control constants
const double freq = M_PI;
const double A = 100 * freq;
const double noise_freq = 200 * freq / (2 * M_PI);
const double force_act_max = 3e5;

const double Ki = 1e4;

```



```
const double Kp = 2e6;
const double Kd = 4.5e4;
```

```
const double initCondI = (upper_leg_mass + lower_leg_mass + (top_mass * 1.3 / 6)) * 9.81 / Ki;
```

Для проведения совместного моделирования переработанная модель движения платформы Matlab была интегрирована в общую модель автомобиля [17], движущегося по участку пересеченной местности по асфальтированному шоссе Белробот – Приморье – Семков городок – Чучаны – МКАД2 протяженностью 12 км. Данный маршрут характерен значительными перепадами высот профиля дороги, разным качеством покрытия (асфальт, разбитые бетонные плиты), проезд через 3 населенных пункта, наличием двух поворотов, две полосы движения.

Результаты моделирования приведены на рисунке 7.15.

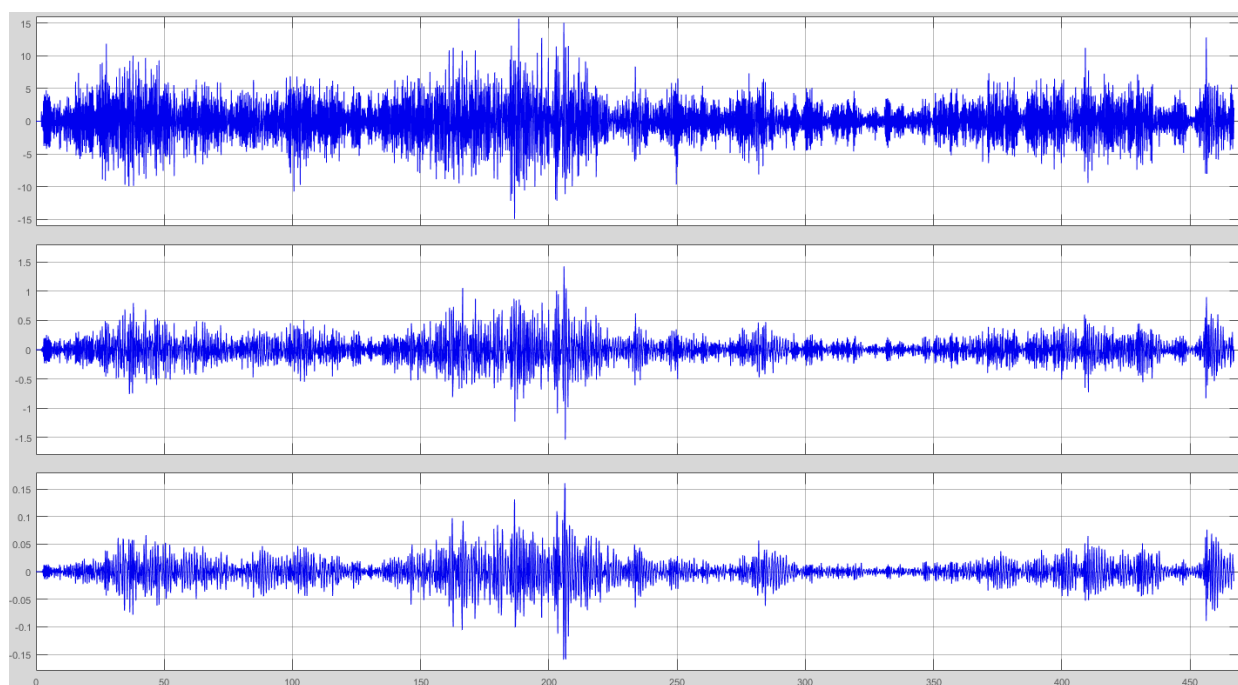


Рисунок 7.15. – Ускорения, скорость и перемещение рамы над передней осью

В таблице 7.1 приведены значения среднеквадратических и максимальных ускорений, скорости и перемещений кабины. Средняя скорость движения составила 64,9 км/ч.

Основные параметры электроактуаторов Parker приведены в таблице 7.2. На рисунке 7.16 приведена характеристика цилиндра актуатора Parker: усилие от скорости.

Таблица 7.1 – Значения среднеквадратических и максимальных ускорений, скорости и перемещений кабины

Параметр	В октавных полосах частот, Гц				В полосе	Макс. значение
	2	4	8	16	0–22,4 Гц	
Z , мм	3,84	2,16	1,0	0,85	4,5	160
$\dot{z}$, см/с	4,9	4,2	2,5	2,1	7,2	153
$\ddot{z}$, м/с ²	0,68	0,9	1,11	1,27	2,03	15,6
F , Н	884	1170	1443	1650	2640	20280

Таблица 7.2 – Основные параметры электроактуаторов Parker

Модель		ETH032				ETH050				ETH080	
Марка		M05		M05	M05	M10	M20	M05	M10	M32	
1	2	3	4	5	6	7	8	9	10	11	
Диам. червяка	мм	5	10	16	5	10	20	5	10	32	
Диам. гайки	мм		16			20			32		
Допуст. ход	мм	50-1000				50-1200			50-1600		
Макс. Скор. хода штока	мм/с										
50 - 400 мм		333	667	1067	333	667	1333	267	533	1707	
600 мм		286	540	855	333	666	1318	267	533	1707	
800 мм		196	373	592	238	462	917	267	533	1707	
1000 мм		146	277	440	177	345	684	264	501	1561	
1200 мм					139	270	536	207	394	1233	
1400 мм								168	320	1006	
1600 мм								140	267	841	
Макс. ускорение	м/с ²	4	8		12	8	15	4	8	15	
Осевое усилие	N	3600	3700	2400	930	7000	4400	17800	25100	10600	

Maximum Speed:

- 1600 mm stroke
- 1400 mm stroke
- 1200 mm stroke
- 1000 mm stroke
- Max actuator speed

Performance Zones:

Motor Codes	Continuous Operation					
	①	②	③	④	⑤	⑥
XPQ ⁽¹⁾	•	•	•			
XPR ⁽²⁾	•	•	•			
XPS ⁽¹⁾	•	•	•	•		
XPT ⁽²⁾	•	•	•	•		
XPU ⁽¹⁾	•	•	•	•		
XPV ⁽²⁾	•	•	•	•		
XPW ⁽¹⁾	•			•	•	•
XPX ⁽²⁾	•			•	•	•

⁽¹⁾ Without brake ⁽²⁾ With brake
 * In-line motor mount only

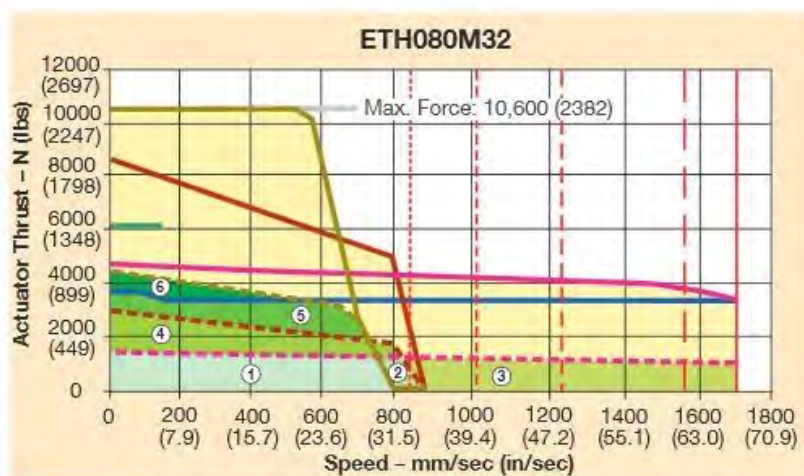


Рисунок 7.16. – Характеристика цилиндра актуатора (усилие от скорости)

Полученные при моделировании данные сравнивались с параметрами электроактуаторов Parker, MOOG (таблицы 7.34) [81, 82].

Как видно из рисунков и табл. 4.1 максимальное перемещение, скорость, ускорение рамы составляет соответственно 160 мм, 153 мм/с и $15,6 \text{ м/с}^2$. Они носят единичный характер. В основном же перемещения не превышают 50 мм. Основной спектр ускорений кабины над приходится на полосы 2–3, 8–10 Гц. Выявлено, что полученное максимальное значение усилия 20280 Н носит также единичный характер и при имитационном исследовании движения автомобиля на стенде можно ограничиться меньшими значениями, как и ограничивать значения перемещений электроактуаторов.

Таблица 7.3 – Основные параметры электроактуаторов MOOG

Вид				
1	2	3	4	5
Model	MB-E6DOF/24/ 1800KG	MB-EP-6DOF/24/ 2400KG	MB-EP-6DOFF40/ 10000KG	MB-EP-6DOF/60/ 28000KG
Макс. значения цилиндра DOF от нейтрал. Горизон. перемещение Боковое перемещение Вертик. перемещение Крен/Roll Тангаж/Pitch Рыскание/Yaw	-0.44 м/+056 м $\pm 0.46 \text{ м} (\pm 8 \text{ in})$ $\pm 0.38 \text{ м} (\pm 5.0 \text{ in})$ $\pm 22.0^\circ$ $-22.0 \text{ } ^\circ$ / $25.0 \text{ } ^\circ$ $\pm 24.0^\circ$	- 0.44м /+ 056 м $\pm 0.46 \text{ м} (\pm 8.0 \text{ in})$ $\pm 0.38 \text{ м} (\pm 5.0 \text{ in})$ $\pm 22.0^\circ$ $-22.0 \text{ } ^\circ$ / $25.0 \text{ } ^\circ$ $\pm 24.0^\circ$	-073м/+0.81 м $\pm 0.70 \text{ м}$ $(\pm 27.0 \text{ in})$ $\pm 0.68 \text{ м} (\pm 26.0 \text{ in})$ $\pm 22.0^\circ$ $\pm 21.0^\circ$ $\pm 22.0^\circ$	$\pm 0.34 \text{ м} (\pm 140 \text{ in})$ $\pm 0.32 \text{ м} (\pm 12.0 \text{ in})$ $\pm 0.36 \text{ м} (\pm 14.0 \text{ in})$ $\pm 10.0^\circ$ $-10.0 \text{ } ^\circ$ / $11.0 \text{ } ^\circ$ $\pm 9.0^\circ$
Максим. скорость при полной нагрузке Горизон. перемещение Боковое перемещение Вертик. перемещение Крен/Roll Тангаж/Pitch Рыскание/Yaw Surge	$\pm 0.70 \text{ м/с} (\pm 28.0 \text{ in/s})$ $\pm 0.70 \text{ м/с} (\pm 28.0 \text{ in/s})$ $\pm 0.55 \text{ м/с} (\pm 22.0 \text{ in/s})$ $\pm 33.0^\circ/\text{s}$ $\pm 34.0^\circ/\text{s}$ $\pm 35.0^\circ/\text{s}$	$\pm 0.70 \text{ м/с}$ $(\pm 28.0 \text{ in/s})$ $\pm 0.70 \text{ м/с} (\pm 28.0 \text{ in/s})$ $\pm 0.55 \text{ м/с} (\pm 22.0 \text{ in/s})$ $\pm 33.0^\circ/\text{s}$ $\pm 34.0^\circ/\text{s}$ $\pm 35.0^\circ/\text{s}$	$\pm 0.90 \text{ м/с} (\pm 36.0 \text{ in/s})$ $\pm 0.90 \text{ м/с} (\pm 36.0 \text{ in/s})$ $\pm 0.80 \text{ м/с} (\pm 31.0 \text{ in/s})$ $\pm 280^\circ/\text{s}$ $\pm 260^\circ/\text{s}$ $\pm 26.0^\circ/\text{s}$	$\pm 0.95 \text{ м/с} (\pm 380 \text{ in/s})$ $\pm 0.90 \text{ м/с} (\pm 36.0 \text{ in/s})$ $\pm 1.00 \text{ м/с} (\pm 40.0 \text{ in/s})$ $\pm 30.0^\circ/\text{s}$ $\pm 29.0^\circ/\text{s}$ $\pm 26.0^\circ/\text{s}$
Максим. ускорение при полной нагрузке Горизон. перемещение Боковое перемещение Вертик. перемещение Крен/Roll Тангаж/Pitch Рыскание/Yaw	$\pm 6.5 \text{ м/с}^2 (\pm 0.65 \text{ G})$ $\pm 6.5 \text{ м/с}^2 (\pm 0.65 \text{ G})$ $\pm 9.0 \text{ м/с}^2 (\pm 0.90 \text{ G})$ $\pm 220^\circ/\text{s}^2$ $\pm 220^\circ/\text{s}^2$ $\pm 360^\circ/\text{s}^2$	$\pm 6.5 \text{ м/с}^2 (\pm 0.65 \text{ G})$ $\pm 6.5 \text{ м/с}^2 (\pm 0.65 \text{ G})$ $\pm 9.0 \text{ м/с}^2 (\pm 0.90 \text{ G})$ $\pm 220^\circ/\text{s}^2$ $\pm 220^\circ/\text{s}^2$ $\pm 360^\circ/\text{s}^2$	$\pm 3.0 \text{ м/с}^2 (\pm 30 \text{ G})$ $\pm 2.0 \text{ м/с}^2 (\pm 20 \text{ G})$ $\pm 5.0 \text{ м/с}^2 (\pm 50 \text{ G})$ $\pm 400^\circ/\text{s}^2$ $\pm 400^\circ/\text{s}^2$ $\pm 450^\circ/\text{s}^2$	$\pm 8.0 \text{ м/с}^2 (\pm 0.80 \text{ G})$ $\pm 8.0 \text{ м/с}^2 (\pm 0.80 \text{ G})$ $\pm 2.0 \text{ м/с}^2 (\pm 20 \text{ G})$ $\pm 300^\circ/\text{s}^2$ $\pm 300^\circ/\text{s}^2$ $\pm 400^\circ/\text{s}^2$
Перемещаемая нагрузка (GML) up to	1.800 kg (4.000 lb)	2,800 kg (6,200 lb)	8.000 kg (17.600 lb)	28.000 kg (61.000 lb)
Полезная нагрузка	1.500 kg (3.300 lb)	2.500 kg (5.500 lb)	6.000 kg (13.200 lb)	24.000 kg (53.000 lb)
Частота воспроизведения	25 Hz	25 Hz	25 Hz	25 Hz

Из актуаторов MOOG подходит модель MB-EP-6DOF/24 (2400 кГ), а из Parker ETH080 M10 (25100 Н). Может также использоваться более дешевый китайский вариант (7800 \$), но его скоростные параметры в 2 раза хуже.

Стоимость лабораторного варианта симулятора для расчетного бюро оценивается в 30–40 тыс.\$, включая симулятор, шлем VR, экран, ПО.

Оптимальным вариантом для реализации системы имитационного моделирования является использование решения, показанного на рисунках В.1 (справа) либо 7.9 (третьего слева) с использованием экрана либо управляемых проекторов, которые смещают изображение дорожной обстановки на экране вправо-влево в соответствии поворотом руля и реагированием на это автомобиля.

Однако наши автопроизводители пока не проявляют интереса к этому и смешанной реальности из-за сложной экономической ситуации, отсутствия кадров и своей недалёковидности, как и не востребованы мои знания и опыт.

7.5.2 О реализации моделирования для пространственных моделей

Проведенное моделирование на плоских моделях натолкнуло на решение для пространственных и более сложных моделей, которые требуют больших затрат машинного времени. Из-за чего на них невозможно реализовать режим реального времени на компьютерах. Предлагаемое решение заключается в проведении вначале моделирования на плоских моделях с одновременной записи перемещений и скорости автомобиля по дороге в файл Simulink. А затем использование данных этих записей для задания режима движения для пространственных моделей, решаемых в замедленном режиме. Это позволяет проводить последующее моделирование нагруженности более сложных моделей (например, пространственных), реализуемых с замедлением вычислений, но эквивалентных режиму реального времени.

7.6 Реализация смешанной реальности

7.6.1 Смешанная (MR) реальность [112] применяется для комплексных исследований автомобиля и проработки его эргономики (рисунок 7.1). Для MR в основном используются мультимедийные, игровые пакеты, технические средства либо на основе создания специального ПО.

Более интенсивно работы в автомобилестроении по смешанной реальности, как средства визуализации, начались с 2016 г. с появлением шлемов виртуальной реальности 4К и пакетов Unity и Unreal Engine, как средств визуализации, предназначенных изначально для создания игр, мультимедиа, киноиндустрии. В основном ведущие автомобильные фирмы для целей MR и моделирования больше используют пакет Unreal Engine (UE)

[119,120]. Примерами его использования являются игра Euro Truck Simulator, рекламные материалы BMW [119] и других фирм [120–122] (рисунок 7.17). Volvo, БелАЗ для MR используют Unity [121]. Для создания VR/AR применяют Microsoft HoloLens и сочетание Creo, Vuforia, Unity [120]. Пакет Unreal Engine ориентирован на использование языка C/C++, а Unity на C#.

Основное внимание за рубежом в MR направлено на изучение поведения водителя в дороге, реакцию его на визуальные критические ситуации, проработку эргономики. Вибростенд при этом почему-то не используется, хотя это существенный фактор. Также из публикаций 2018–2022 гг. не видно, что они проводят исследования движения, колебаний, необходимые для определения долговечности и других параметров. А это ключевые задачи, которые следовало бы реализовывать при MR.

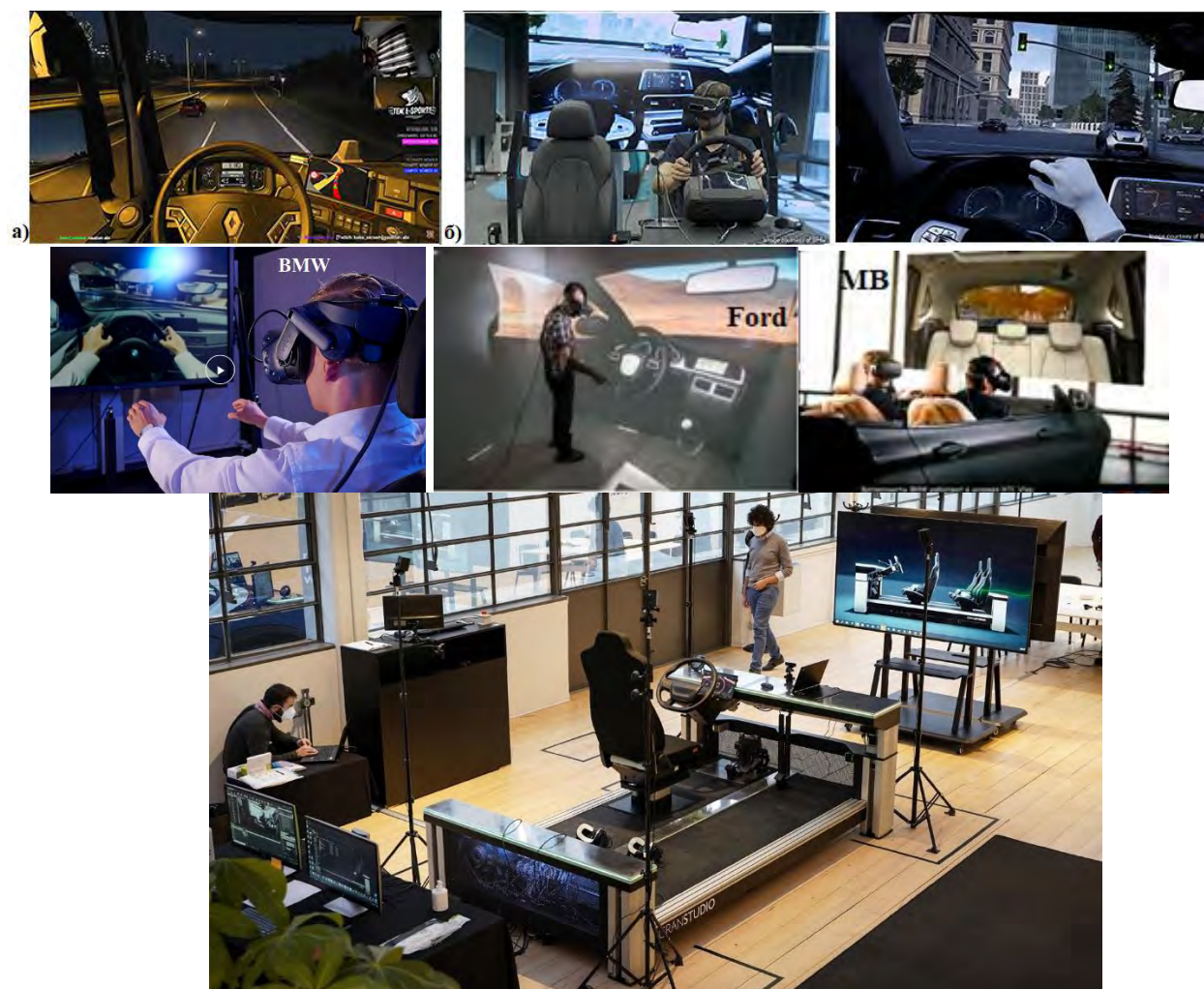


Рисунок 7.17. – Использование смешанной реальности в UnrealEngine в игре Euro Truck Simulator (а), фирмами BMW (б), Ford, Mercedes Benz (MB) и проработка эргономики автомобиля в Granstudio, Италия [112]

Ведущие фирмы сочетают использование различных пакетов и шлемов. Это зависит от решаемых задач. Так Audi для моделирования использует UE, а для проработки дизайна 3D max с Unity. Фирма Volvo первоначально ориентировалась на пакеты Unity, 3D max и шлем Meta Oculus Quest. Сейчас Volvo начала использовать шлемы финской фирмы Varjo XR-3/VR-3 и его программного обеспечения в сочетании с движком Unity [121].

Большинство автомобильных фирм ориентируется на шлемы HTC Vive Pro со станциями позиционирования и трекерами. У Meta Oculus Quest их нет. Mercedes Венц, Ford помимо HTC также используют Microsoft HoloLens 2, сотрудничают с фирмой Microsoft по вопросам MR. На рисунке 7.1 показано использование Microsoft HoloLens для MR, достигаемое с помощью видеокамеры шлема и видеопанели для показа части внутренней и внешней обстановки, то что предлагалось мной в работе [117], как один из подходов.

Сейчас задачи смешанной реальности значительно расширились. Они включают помимо рекламных, дилеровских задач, дизайна (на что больше сконцентрировано внимание) еще и моделирование движения, колебаний и управляемости автомобиля, проработку его эргономики, определение режима движения, нагрузок на его узлы, использование полученных данных при расчетах и стендовых испытаниях. Требуется комплексная реализация всего этого. Ключевым моментом является рассмотрение выбора моделей, программных и технических средств для создания симулятора с использованием смешанной реальности.

Необходимость такой работы обусловлена отсутствием серьезных публикаций по исследованиям в MR. В основном информация по MR в интернете носит общий, поверхностный характер, нет ни моделей, схем, решений указанных выше задач.

Рассмотренные ниже решения базируются на исследованиях автора по имитационному моделированию автомобиля, включая MR в Simulink в режиме реального времени [14, 17], тестировании новых программных продуктов и анализе решений и тенденций за рубежом.

Акцентируется внимание на трех ключевых моментах:

- визуализации дорожной обстановки, являющийся основой проведения исследований автомобиля путем моделирования в смешанной реальности;
- имитации звукового воздействия на водителя;
- проработки эргономики рабочего места водителя.

В качестве основного пакета для MR при проектировании и моделировании ситуаций ведущие автомобильные компании Mercedes Benz, Audi, GM, BMW, Volkswagen и другие в основном используют Unreal Engine. Возможности пакета Unity несколько слабее, он больше применяется в сфере

маркетинга (создание промороликов, VR- презентации), также могут применять и другие пакеты. UE и Unity очень сложны в усвоении и требуют работы в команде: проектировщиков, программистов и дизайнеров. UE позволяет использовать 3D модели основных CAD пакетов: Autodesk Inventor, Autodesk 3ds Max, Dassault Systèmes CATIA, PTC Creo, Siemens NX и др., но требует сложной конвертации.

Для применения в смешанной реальности целесообразно использование шлема HTC Vive Pro 2 со станциями позиционирования и трекерами ввиду более высокой точностью его системы позиционирования (SteamVR Lighthouse) и наличием у него специальной видеокамеры обзора, которых нет в Oculus (Meta) Quest 2.

Система позиционирования SteamVR Lighthouse состоит из двух базовых станций, оптических сенсоров и инерциальных измерительных блоков (IMU) в шлеме и контроллерах, подключенных к компьютеру (рисунок 7.18).

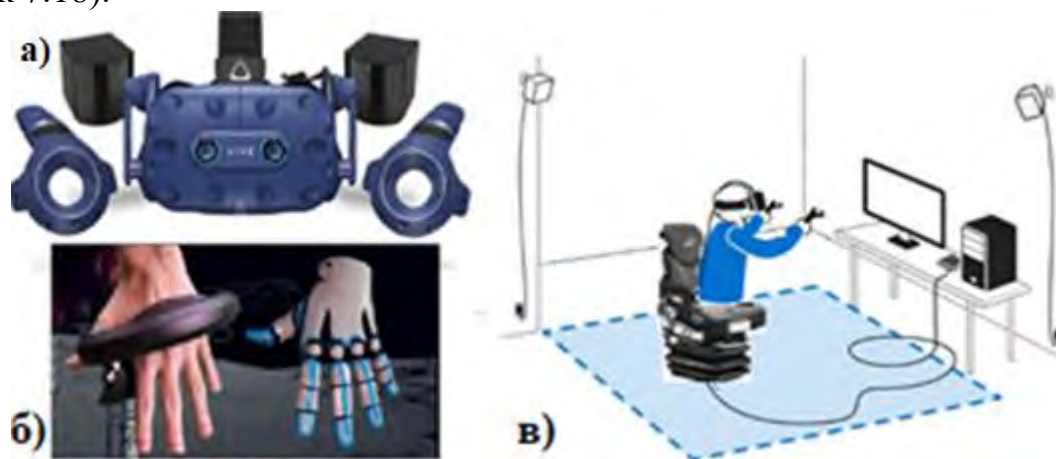


Рисунок 7.18. – Общий вид шлема HTC Vive Pro 2 с контроллерами и станцией (а, б), схема использования VR/MR (в)

Базовые станции состоят из двух вращающихся лазеров и массива инфракрасных светодиодов на шлеме и контроллерах (кастетах) руки. Один из лазеров вращается вертикально, второй – горизонтально. Таким образом лазеры по очереди «сканируют» окружающее пространство.

Сенсоры на контроллерах руки и шлеме фиксируют все оптические импульсы с базовых станций и измеряют время между ними. Так как частота вращения лазеров заранее известна (60/90 Гц), по времени между импульсами можно вычислить углы поворота каждого из лучей, что определяет 2D-координаты оптического сенсора. Зная взаимное расположение сенсоров на контроллере можно легко рассчитать 3D-положение контроллера в пространстве (задача PnP).

Проблемным местом до настоящего времени являлось точное определение положение руки испытуемого. Применяемые в HTC Vive Pro 2 контроллеры не давали необходимой точности позиционирования. Сейчас появилось интересное решение, предложенное небольшим стартапом Manus VR, которые интегрировали перчатки-контроллеры с платформой трекинга Valve Lighthouse со стандартными контроллерами SteamVR [123].

Это позволяет рассчитать точные движения локтя и руки (0,1 мм), используя информацию отслеживания от контроллера Vive (рисунок 7.19).

Вторым проблемным местом являлось отсутствие интерфейса для шлема и контроллеров руки. Сейчас компания HTC выпустила SDK (набор средств разработки на C++, C#) [123–126], который дает разработчикам возможность отслеживания перемещений рук в своих приложениях для Vive или Vive Pro. Для автономной гарнитуры Vive Focus этот SDK обеспечивает трекинг положения рук и распознавание жестов. Стоимость подписки на SDK на год 133 \$.



**Рисунок 7.19. – Общий вид перчаток Manus gloves (a),
с контроллером HTC Vive wand б)**

Шлем Oculus Quest использует иную технологию на основе распознавания рук в OpenCV и инерционных датчиков перемещений. Ему не требуется рабочие станции и поэтому он дешевле. Но точность позиционирования в нем хуже. На снимках AR/MR реальности BMW, Ford, Мерседес-Бенц видно, что они используют шлемы черного/синего цвета, характерные для шлемов HTC. Шлемы Oculus Quest выполнены в белом цвете.

7.6.2. Ключевые аспекты моделирования со смешанной реальностью

7.6.2.1 Использование реального макро и микропрофиля дороги

Более лучшим решением для смешанной реальности, как и для других видов моделирования является использование реального макро и

микропрофиля дороги в виде совместного массива данных, заданного по точкам с шагом 0,1–0,15 м, размещенного в оперативной памяти. Для реализации этого требуется разработка программы на C/C++ для модуля S-Function Builder в Simulink, что позволяет хранить 96000 точек пути (12500 м дороги) с возможностью дополнительной подкачки и вести моделирование с переменной скоростью движения. Данный подход является ключевым моментом для реализации режима реального времени [14]. Использование случайного возмущения на основе генератора белого шума не позволяет реализовать режим переменной скорости движения. Вопросы получения и использования макро и микропрофиля дороги требуют новых подходов [14], включая разработку способов записи реального макро и микропрофиля дороги на основе GPS [111].

7.6.2.2 Использование корректных характеристик элементов подвески, трансмиссии и их моделей

Важным для смешанной реальности является использование более корректных характеристик элементов подвески, трансмиссии и их моделей, рассмотренных в пп. 3.5, 3.9, 3.12, обеспечивающих погрешность не более 10–15 % с учетом спектра на реальной дороге, чтобы затраты оправдывали его.

Аналогичным способом должна описана характеристика увода шины.

7.6.2.3 Разнесение вычислений на компьютеры

Необходимость разнесения вычислений на компьютеры обусловлено тем, что существующие компьютеры сейчас достигли своего потолка производительности 4–5/6 ГГц и требуется компенсировать потери производительности из-за передачи информации на стенды по сети CAN BUS. Использование оперативной памяти DDR5 (>6000 МГц), жидкостного охлаждения процессоров с разгоном их до 5 ГГц позволяет повысить производительность новых компьютеров всего на 20–25 %.

Поэтому более лучшим решением является разнесения вычислений по подсистемам и реализации визуализации, звука на отдельные компьютеры, чтобы реализовать режим реального времени и охватить разные подсистемы, Для обмена данными между ними использовать сеть CAN BUS (как наиболее быстродействующую), позволяющую адресно направлять и обмениваться информацией, а для обмена с Raspberry использовать сеть Samba. Тем более, что уже имеются адаптеры с частотой до 10 Мбит/с, а также можно

использовать предложенный в этой работе и [14] метод сжатия информации, повышающий производительности CAN BUS в два раза.

Возможен вариант использования нескольких компьютеров с твердотельными накопителями .M2 (>6000 МГц), объединенных быстродействующей сетью (X4 DDR (20Gb/s) и обменом в ней бинарными файлами подсистем.

7.6.3 Визуализация дорожной обстановки, имитирование звука

7.6.3.1 Визуализация дорожной обстановки является важным компонентом моделирования. Для имитационного моделирования визуализация может быть реализована с помощью проекторов с большими экранами [114, 115] либо на мониторах. Данное решение хотя и может использоваться и для AR/MR реальности, но не создает реального эффекта присутствия в кабине из-за искусственной обстановкой вокруг, которую видит испытуемый водитель.

Поэтому за рубежом для визуализации в MR со шлемами основным используется компьютерная графика, реализуемая в игровых, мультимедийных пакетах Unity, Unreal Engine [123, 124]. Это довольно мощные пакеты с объемом до 21 Гб, но на их освоение требуется до 2-х лет.

Реализация имитационной обстановки в пакетах Unity и Unreal Engine, осуществляется на основе создания сцены с участием актеров, в данном случае водителя, пешеходов, дороги и других машин. В качестве сцены используется ландшафт изменяемой местности, где методом вытягивания участка (без задания физических значений) можно создать возвышенности, низины, горы через них проложить дорогу, вокруг их расставить деревья, здания и другие объекты тем самым реализовать обстановку. Либо используются готовые шаблоны типа предлагаемой кольцевой гоночной трассы или на основе городских условий США. Параллельно создается 3D модель кабины автомобиля, конвертируемая в формат .fbx Unreal Engine. Эта модель размещается на трассе, в ней располагается человек, камера показа в объекте и путем установки связей и настройки визуальных компонентов активизируется их действия управления автомобилем. Дополнительно помимо визуальных требуется программирование компонентов в MS Visual Studio на C/C++, включая снятие информации с АЦП рулевого колеса, педалей газа, тормоза, рычага переключения КПП. Благодаря моделируемой в UE визуальной обстановки создается иллюзия движения.

Отметим, что в пакетах Unreal Engine и Unity отсутствуют возможность задания продольного и микропрофиля дороги и синхронизации их с дорожной

обстановкой, что необходимо для комплексного моделирования автомобиля. Подчеркнем, что условия на дорогах в СНГ значительно отличаются: преобладают дороги с невысоким качеством, двухполосным движением, перепадами высот, что усложняет моделирование.

Необходима реализация моделирования в Simulink движения, колебаний, управляемости автомобиля по реальным дорогам и интеграция их с игровыми пакетами через создаваемые программы на C/C++.

На рисунке 7.20 приведены примеры использования MR в BMW, Volvo.

На рисунке 7.20а показано использование фирмой BMW смешанной реальности при проведении эргономических исследований автомобиля в городских условиях в пакете Unreal Engine [120] с применением его типового контента без учета микропрофиля дороги: на видео идеально ровная дорога. Из видео следует, что как такового осуществления колебаний на симуляторах BMW и Volvo вообще нет. Их симуляторы с MR не имеют вибростендов. Возможно это связано со сложностью реализации в UE, неактуальностью за рубежом вопроса вибрации на хороших дорогах. Хотя ранее на симуляторах с движущей платформой это реализовывалось [14].



Рисунок 7.20. – Использование фирмами BMW (а), Volvo (б) смешанной реальности

Из работы [120] следует, что они используют пакет Unreal Engine 4.1 в основном для проработки дизайна внутри автомобиля, демонстрируя возможность дотянуться рукой до какой-то кнопки. Следует указать, что в 2018 г. еще не было высокоточных трекеров (датчиков перемещений) и ПО для них: они появились лишь только в 2020 г. т.е. в тот период вряд ли они могли корректно оценивать перемещения руки.

В пакетах Unity и Unreal Engine упрощенно реализуется имитация перемещения через визуализацию обстановки. В лучшем случае можно задать внешнюю характеристику двигателя, массу автомобиля, однако при этом не учитывается трансмиссия поскольку они ориентированы на игры. Для подключения органов управления требуется написать программы на C# или C/C++ с задействованием датчиков, АЦП. Неясно, помимо этого, определяет

ли BMW на основе UE долговечность машины, которая связана со скоростью движения (режимом движения) и микропрофилем дороги.

Volvo в сотрудничестве с финской фирмой Varjo создала недвижущийся симулятор вождения (рисунок 7.20 б, the ultimate driving simulator) на базе игрового движка Unity, шлема дополненной/смешанной реальности Varjo XR-3 и британского тактильного костюма Teslasuit с обратной связью [121]. С помощью пакета Unity создается 3D-мир дорожной обстановки с наложением в очках Varjo и его ПО изображения внутреннего вида салона и внешней обстановки, что создает приближенное ощущение реального вождения машины. Это позволяет смоделировать различные дорожные ситуации: внезапного появления пешехода или животного на дороге, торможения впереди идущей машины, резкий объезд препятствий, обгоны, движение по дороге. Судя по видео, Varjo также использует совмещение записанного видео в дороге с изображением, смоделированным с помощью Unity без моделирования колебаний автомобиля. При проработке эргономики для более полного ощущения в пакете Varjo используется физический макет рулевого колеса, созданный с помощью 3D принтеров [122].

Похоже, что в основном за рубежом, да и у нас в РБ, в MR все делается только для проработки дизайна и эргономики.

БелАЗ, единственный из заводов СНГ, использует виртуальную, дополненную и смешанную реальность с помощью Unity, Meta Oculus Quest и 3D max в своих разработках. Имеет тренажер: стенд с мониторами для обучения водителей навыкам управления карьерными самосвалами, созданный с использованием смешанной реальности. Но он не подходит для исследовательских целей из-за недостаточного быстродействия имитатора перемещений и невозможности интегрировать его с Simulink.

По субъективной оценке автора, сидевшего на простых симуляторах (игровых с низкой скоростью перемещения 0,3–0,5 м/с), они не дают реального ощущения движения и колебаний. Такую же оценку можно сделать по симуляторам MR ведущих автомобильных фирм без вибростенда.

Следует отметить, что в основных упомянутых пакетах MR нет возможности задания продольного и микропрофиля дороги и синхронизации их с дорожной обстановкой, что необходимо для комплексного моделирования автомобиля.

Важным моментом является выбор одного языка программирования: C# либо C/C++. Лучше ориентироваться на C/C++ и Unreal Engine. Последний мощнее чем Unity. C/C++ используется в системе: в АЦП, микроконтроллерах, мини-компьютерах и в S-Function Builder, SiminTech. И по быстродействию программы на C/C++ в два раза быстрее чем C#, что важно для Real-Time,

учитывая, что компьютеры и пакеты и так работают на пределе своих возможностей.

Другим вариантом визуализации дорожной обстановки является предложенный автором способ записи видео дорожной обстановки при постоянной скорости движения автомобиля по дороге с последующим воспроизведением этого видео с помощью разработанных программ на компьютере и шлеме пропорционально скорости движения при моделировании в Simulink [14], рассмотренный подробно в пп. 7.8.4.1. Последний вариант гораздо проще и менее затратный.

Но все же в дальнейшем представляется более перспективным использование Unreal Engine для отработки дорожных ситуаций, эргономики.

7.6.3.2 О некоторых проблемах визуализации объекта в Vuforia

Пакет Vuforia [115, 116], интегрированный с 3D пакетом Creo, в основном используется для VR/AR. Информация о его применении в MR отсутствует, хотя в нем есть возможность показа видео.

В нем можно отобразить только один объект (в конкретном случае интерьер нижней, средней передней части кабины и управлять им (рисунок В.2 над Video projector). А верхнюю часть и боковые осуществить в Vuforia пока не получилось. Из-за чего приходится ограничиваться более простыми возможностями – проработкой только нижней частью кабины. Одним из решений этой проблемы может быть использование боковых видео панелей с соответствующим изображением или компьютерных декораций отдельных частей кабины по типу театральных и демонстрации видео.

7.6.3.3. Последовательность компьютерных операций по созданию модели в Creo и Vuforia для VR/AR

Для реализации этого необходимо выполнить процедуры с Creo и Vuforia, приведенные в видео инструкции [115], включающие 8 шагов для реализации VR/AR/MR:

- Step 1: Create 3D model in Creo Parametric;
- Step 2: Save model as *.pvz;
- Step 3: Open *.pvz in Creo Illustrate and set custom Figures and Sequences;
- Step 4: Publish ILLUstrate as *.pvz;
- Step 5: Import *.pvz into Vuforia Studio and prepare data for AR mobile view;
- Step 6: Print your ThingMark;
- Step 7: Run Vuforia View on your mobile and scan the ThingMark;

Step 8: Use your model in AR

>> Play sequence or simple investigate your model.

Примечание. Шаги и операции приведены на английском для облегчения работы с пакетами и их меню.

Частичное осуществление MR возможно только с задействованием видеокамеры, снимающей видео с экрана и накладывающей ее на изображение внутренней 3D обстановки кабины [115]. Данный способ менее подходит для MR. Более целесообразным все же является использование пакета Unreal Engine [119, 120] в сочетании со считыванием данных с датчиков положения рук испытуемого водителя и сравнивающее их данные с положением органов управления. Либо использовать подход, предложенный в п.7.8.7. Во всех случаях необходимо установить VIVEPORT SDK 1.7.20 [123–126] и разработать программное обеспечение на языке C++. Судя по ландшафту местности, похоже, именно Unreal Engine, использовался на Форде (рисунок 7.17).

7.6.3.4. О программной визуализации дорожной обстановки в MR

В шлемах HTC и Oculus имеется возможность установки режима воспроизведения видео, реализуемого в компьютере для трансляции на шлем путем следующих процедур [118].

В приложении Vive Home:

1. Нажмите кнопку «Меню» и выберите Объекты.
2. Выберите .
3. Выберите , а затем найдите желаемую видеозапись на компьютере.
4. Используйте элементы управления для воспроизведения или приостановки видео, регулировки громкости звука и т. д.
5. Выберите , чтобы зафиксировать видеопроигрыватель таким образом, чтобы он перемещался вместе с вашим взглядом.
6. Выберите , чтобы приглушить освещение во время просмотра видеозаписи.
7. Выберите , чтобы просматривать видеозапись с полным эффектом присутствия в виртуальной реальности.

Можно выбрать формат просмотра (моноскопический и несколько стереоскопических), который соответствует видеозаписи.

Предлагаемый подход визуализации позволяет осуществлять моделирование движения со смешанной реальностью с максимальной иллюзией участия водителя в движении и минимальными затратами на реализацию.

7.6.4 Разработка программы видео для смешанной реальности

На основе анализа компьютерного моделирования [14] пришли к выводу, что управление скоростью воспроизведения видео в отличие от ранее использованного подхода все же целесообразно осуществлять, используя функцию задержки `waitKey(zd)`. Особенностью этой функции является использование целого значения `zd`. Чтобы уменьшить погрешность при преобразовании `float` в `long`, принимающей меньшее значение, использовалась функция `fmod(vd,zd,1)`, чтобы скорректировать значение до ближайшего целого числа. Варьированием значений задержки можно регулировать скорость просмотра видео. Предложено формировать бинарный файл обмена информации (пройденный путь, скорость, величина задержки) в S-Function Builder, который считывается в программе видео и регулирует скорость воспроизведения видео.

Первоначально в работе [14] для просмотра видео ориентировались на Raspberry Pi 3b на основе программы, созданной на языке C++. Но он обеспечивал разрешение только 1280×720 и на его операционную систему нельзя установить ПО шлема, которое необходимо для трансляции видео с компьютера. Поэтому для функционирования шлема использован компьютер с Windows 10, обеспечивающий разрешение 1920×1080 и работу ПО шлема. Дополнительно потребовалось создать новую программу просмотра видео в Windows с использованием библиотеки OpenCV. Эта библиотека функционирует только на C++ или Python. Поэтому для разработки программы на C++ выбран пакет MS Visual Studio 2022 версии 17.12, для которого имеется готовая библиотека OpenCV 4.6.0 (`opencv-4.6.0-vc14_vc15.exe`) в виде бинарных файлов, что значительно упрощает ее установку и подключение к MS Visual Studio. В начале необходимо загрузить MS Visual Studio 2022 и библиотеку Open CV 4.6.0 из интернета и воспользоваться рекомендациями работ [127, 128] по их установке.

Далее использовать следующие процедуры, данные в инструкции на шлем HTC для приложения Vive Home для воспроизведения видео с компьютера, приведенные в пп. 7.8.4.1.

Ниже приведен исходный текст новой разработанной программы воспроизведения видео на C++ в MS Visual Studio 2022 v17.12.

```
// CASHv.cpp – program of implementation of video for modeling of vehicle
```

```

#define _CRT_SECURE_NO_WARNINGS
#include <opencv2\opencv.hpp>
#include <opencv2\highgui\highgui.hpp>
#include <opencv2\imgproc.hpp>

```

```

#include <iostream>
// #define _CRT_SECURE_NO_DEPRECATE
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include <share.h>
using namespace std;
using namespace cv;

struct d {
    long dl; // length of a way
    float v; // speed of vehicle
    float zd; // delay in mc
    float spe; // engine speed
    float pwe; // engine power
};

struct d vd;

long ch = 0;

int main()
{
    FILE* fp;
    long FPS, zd, zd0, chz, i;
    float sv;
    int a, zd5;
    long sumlv; // sum streng on speed of vehicle    delay in mc
    bool bSuccess;
    //char md[5] = "rb";
    int FramesSkipped = 1;
    ch = sumlv = 0;
    sv = 0.0;
    vd.zd = 20;
    zd = 15;
    a = 1;
    chz = 10;
    VideoCapture cap("road.mp4");    // Creating VideoCapture object and opening input
file of visualisation
    if (!cap.isOpened()) // //Checking of opening input file
    {
        cout << "Error opening video stream or file" << endl;
        return -1;
    }
    // cap.set(CAP_PROP_FPS, 50); // this function d'nt work

    Mat frame; // Creating Mat object

    a = 1;
    vd.zd = 40;
    while (a == 1)

```

```

{
    if (ch == 20000)
    {
        fp = _fsopen("ma.bt", "rb", _SH_DENYNO);
        fread(&vd, 20, 1, fp); // reading file of a time delay
        ch = 0;
        fclose(fp);

        zd = fmod(vd.zd, 1);
        if (zd > 80)
            zd = 80;
        if (zd <= 5)
            zd = 5;

    }
    ch++;
    chz = 8000*zd;
    for (i = 0; i < chz; i++)
        ;

    cap >> frame; // Capture frame-by-frame
    if (frame.empty()) // If the frame is empty, break immediately
        break;

    imshow("roadFrame", frame); // Display the frame
    char a = (char)waitKey(zd); //Press ESC on keyboard to exit/
    if (a == 27)
    {
        break;
    }
} // End while

return 0;
}

```

В данной программе путем считывания файла обмена, формируемого в модуле S-function Builder Simulink, определяются целые значения величины задержки `vd.zd` в мс, округленные до ближайшего целого, для обеспечения необходимой скорости воспроизведения видео. Рассчитываемая величина задержки корректируется на основе цикла `chz = 8000*zd;` `for (i = 0; i < chz; i++)` и функции `waitKey(vd.zd)`. Это связано с увеличенной скоростью моделирования в результате модернизации компьютера.

Для функционирования программы необходимо получить зависимость значения задержки в мс, которая обеспечивает необходимую скорость вывода видео от скорости движения. Для ее получения определяют время просмотра прохождения участка с последующим пересчетом в скорость автомобиля при заданном фиксированном значении задержки.

Считывание информации по задержке производится с периодичностью 0,3 с, чтобы не загружать процессор обращением к файлу. Для этого используется счетчик циклов ch , который после определенного числа обнуляется, считывается задержка и производится заново подсчет циклов.

Важным моментом является осуществление записи видео. Целесообразно проводить запись видео в разрешении 1920×1080 при 30 fps при постоянной скорости 20 м/с. Выбор этой скорости обусловлен возможностью достижения верхних значений 30–40 м/с при визуализации, который получается за счет времени задержки воспроизведения видео. Выяснено на это влияет еще объем файла (его кодирование). Из-за чего при первоначальном объеме файла 1,1 Гб, соответствующего участку длиной 12 км, на первом этапе не удалось обеспечить требуемый диапазон регулирования скорости ($v=30 \text{ м/с}$). Поэтому файл сжат в пакете Movavi Video Converter до 500 Мб.

На рисунке 7.21 представлена полученная зависимость значения задержки в мс, которая обеспечивает необходимую скорость вывода видео путем обработки времени просмотра видео участка пути 12 км.

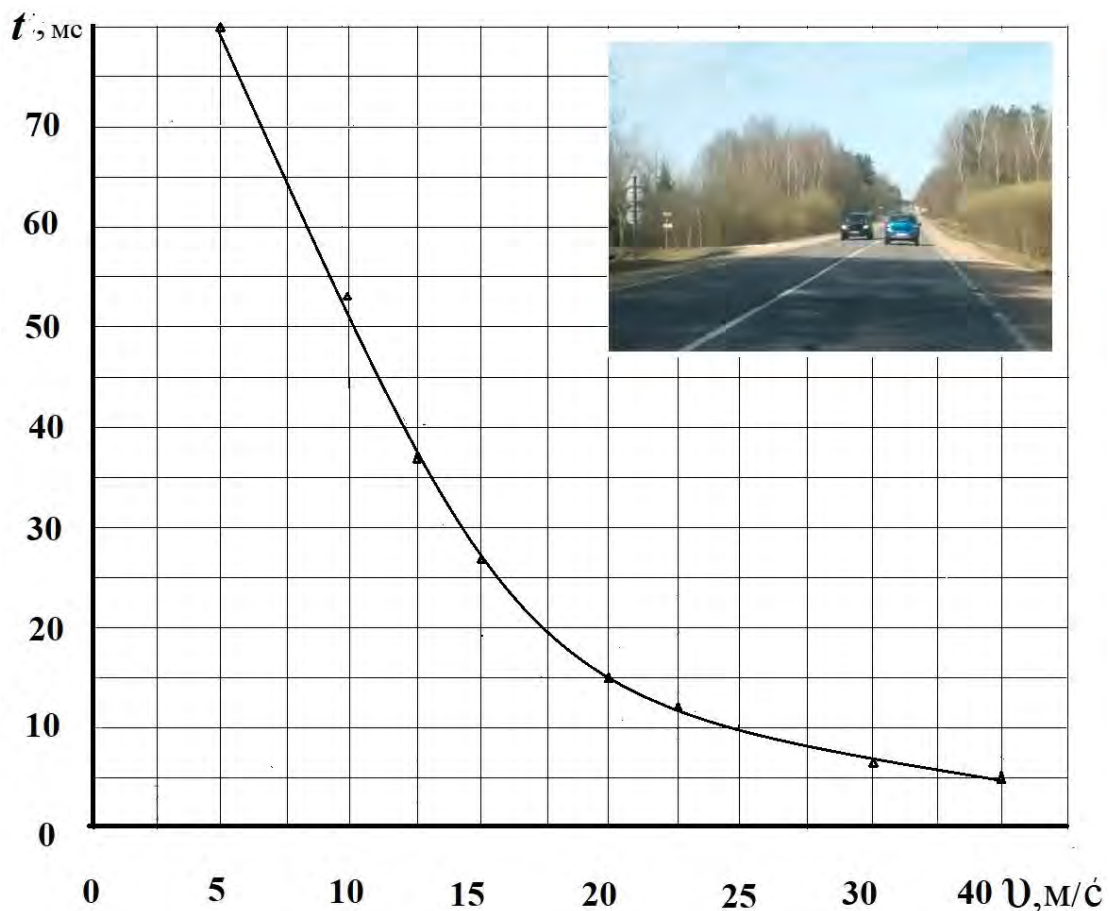


Рисунок 7.21. – График изменения времени задержки, необходимого для обеспечения соответствия просмотра видео скорости движения автомобиля

Моделирование движения, колебаний, управляемости, топливной экономичности автомобиля и видео осуществлялось в Simulink (R2024a) на одном компьютере (AMD Ryzen 5 5600X, 3,4 ГГц, ASUS B450 Plus, ОП 16 Гб, P5 Plus M.2 (6000 МГц), GeForce RTX 3050) с помощью блок-схемы (рисунок 7.22) на базе отработанных решений [14, 17].

С помощью модуля RelayMR осуществлялось преобразование скорости в величину задержки, а через S-Function Builder производилась запись данных пройденного пути и задержки в виде бинарного файла, который затем считывался в программе видео. И на основе этих данных в блок-схеме (рисунок 7.22) реализовалось управление скоростью просмотра видео.

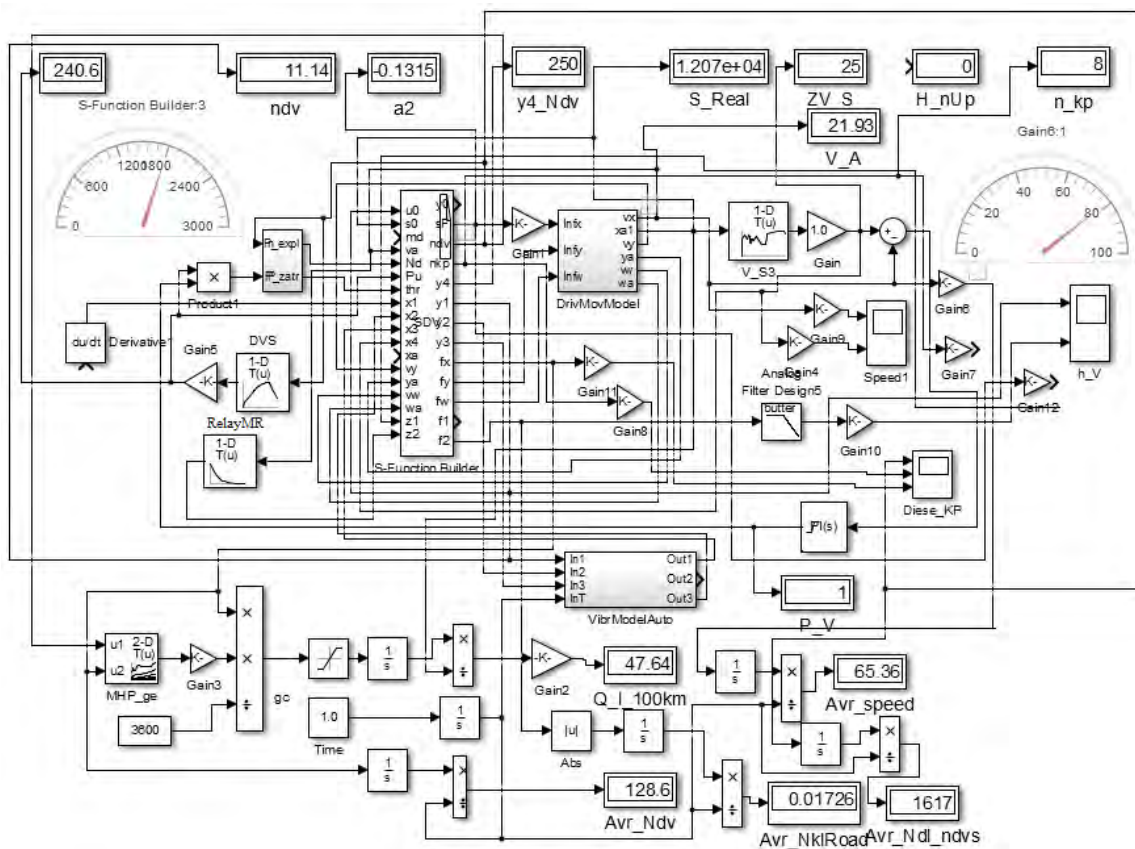


Рисунок 7.22. – Блок-схема реализации имитационного моделирования с AR/MR в Simulink

Для подтверждения предложенного метода сравнивалось совпадение времени прохождения конечной точки пути (визуального объекта на ней), показавшее расхождение по времени 3–4 % при средней скорости 65 км/ч, расходе топлива 47 л/100 км, $m_a = 20000$ кг, что вполне приемлемо для длины участка (12 км) и свидетельствует о корректности метода. Дополнительно проведено сравнение при уменьшении режима скоростей при $K=0,7$ ($v_a = 47,4$ км/ч, 42,7 л/100 км), показавшее увеличение времени в 1,38 раза

против 1,42 заданного, что также подтверждает корректность предложенного метода.

Данное решение предназначено в первую очередь для применения на симуляторах с вибростендом, где скорость задается нажатием на педаль газа, но может использоваться и на простых устройствах в составе шлема и имитатора рабочего места, используя игровой руль и педали. Также оно может использоваться и без шлема, используя конусный раструб из непрозрачной ткани или бумаги между экраном компьютера и лицом испытуемого, чтобы исключить стороннее воздействие комнатной обстановки. Такое решение применялось на заре виртуальной реальности, когда не было шлемов.

Последний вариант гораздо проще и дешевле, особенно, если не задействован вибростенд и не учитываются повороты, наклоны головы. На нем можно решать многие упомянутые задачи движения, колебаний и управляемости автомобиля.

Предлагаемое решение позволяет наглядно демонстрировать процесс движения и проводить оценку топливной экономичности, долговечности узлов проектируемого нового автомобиля на основе получаемых данных в процессе моделирования движения, задаваемого режима и вибронагруженности машины.

В полноценном варианте необходимо задействование рабочего места водителя с вибростендом с использованием датчиков педалей газа, управления КПП (селекторного переключателя), поворота руля на основе потенциометров, информация в виде электрического сигнала с которых снимается и подается через АЦП в Raspberry Pi 3/4, а из него записывается в файл обмена с Simulink и эти данные используются в процессе моделирования.

Для реализации схем на рисунках 7.21, 7.24 требуется использование трех компьютеров и одного мини-компьютера Raspberry Pi 3/4/400. На последний возлагается функция считывание данных с органов управления.

Для проведения имитационного моделирования движения, вибраций и управляемости автомобиля в режиме реального времени в целом достаточно производительности компьютера с частотой 3,8–4 Гц и памятью DDR4 (3200–3600 МГц) при использовании плоской модели автомобиля без передачи данных на стенд, а для пространственной модели может использоваться предложенный подход в работе [14]. Для AR/MR требуется использование более производительных компьютеров и видеокарт. На компьютер с приложением AR/MR помимо 3D моделирования возлагается функция наложения 3D модели внутреннего интерьера кабины на видеоизображение дорожной обстановки, фиксируемое камерой шлема водителя либо реализация визуализации дороги, описанная выше.

Важным условием помимо вибрационного является осуществление звукового воздействия, без которых нет ощущения реального движения. Реализовать это можно на базе моделирования работы двигателя и трансмиссии, рассмотренных в пп. 3.14.3, которое может являться источником звукового сигнала работы двигателя.

Как видно из сравнения с вариантом имитационного моделирования реализация смешанной реальности во многом повторяет ее как в схемном решении, так по использованию моделей. Но в части программного обеспечения смешанная реальность с учетом искусственной руки получается более сложной и более дорогой (стоимость в РФ только шлема и искусственной руки порядка 4 тыс.\$ и плюс затраты по интеграции ПО).

Для информации стоимость системы Varjo составляет 6500 евро и в списке стран, куда она поставляется России и Беларуси нет. Отсюда вытекает, что в определенных условиях и при отсутствии высококвалифицированных кадров в этой области ИТ лучше ориентироваться на имитационное моделирование, основные вопросы хорошо освещены в данной монографии и в работе [14] и являются базисом для смешанной реальности.

7.6.5 Реализация звукового воздействия

7.6.5.1 Важным моментом комплексного моделирования является осуществление звукового воздействия путем реализации звука работы двигателя и шума автомобиля. Без этого нет ощущения реального движения. На ведущих автомобильных фирмах для такого моделирования используются игровые пакеты UnrealEngine, Unity, где предусмотрена возможность некоторого имитирования звуков [119]. Но эти пакеты не позволяют реализовать процесс реального моделирования движения и вибраций автомобиля.

Как рассмотрено выше для целей имитационного моделирования на симуляторах необходимо использование пакета Matlab/Simulink, позволяющего решать эти задачи в комплексе в режиме реального времени.

Анализ информации в интернете показывает, что для Simulink задача реализации звука и шума не решена. Хотя и имеется ряд теоретических работ по звуку и шуму двигателя. Так в работе [129] рассматривается моделирование акустического излучения ДВС, базирующиеся на использовании простейшей физической модели, в которой возбуждающий фактор (процесс) посредством импульса воздействует на материальную среду (тело) и передает ей колебательную энергию. Затем она распространяется по среде и достигает ее наружной поверхности, контактирующей с окружающей газовой средой.

Спектр звука рассчитывается с помощью передаточных функций и его невозможно использовать в Simulink.

В другой работе [130] рассматриваются возможности моделирования структурного шума двигателей внутреннего сгорания на основе австрийского программного комплекса AVL EXCITE и созданной на его основе модели. Исследованы состав и структура акустического излучения двигателя внутреннего сгорания. Сопоставлены области их применения и решаемые задачи. Приведены примеры результатов расчетов структурного шума дизелей с использованием представленных методов. В данном комплексе невозможно получить необходимый сигнал для использования при имитации.

В работе [131] рассматривается методика расчёта связанных виброакустических задач методами конечных элементов (МКЭ) в пакете ANSYS в сочетании полуаналитическими по формулам статистической акустики, основанными на поршневой аналогии в полосе частот до 200 Гц. Предлагается определять перемещения узлов конечномерной модели совокупности поверхностей ограждения, затем, элементы последних, представляются акустическими поршнями – источниками звука, образующими поршневую группу. В каждой точке воздушного объёма находится звуковое давление, создаваемое элементарными источниками, а полное определяется формулами логарифмического суммирования. Приведенное сравнение с экспериментальными данными результатов расчёта уровней звукового давления на рабочем месте водителя автомобиля УАЗ-3160 различными методами, показало все же большое расхождение до 10-20 Дб.

Отметим, что данные работы ориентированы на стационарный режим работы двигателя. Хотя они и позволяют более или менее рассчитать спектральную характеристику, однако получить реальный сигнал звука для подачи на усилитель НЧ либо звуковую карту в них невозможно. Поэтому они не пригодны для реализации в Simulink. Кроме того, в них невозможно сформировать и реализовать сигнал звука в нестационарном переменном режиме работы двигателя при изменении нагрузки и мощности.

Целью данного подраздела работы является рассмотрение имитирования звука двигателя, шума с помощью ПО и пакета Simulink.

Для этого рассмотрим источники звука двигателя и шума при движении автомобиля, способы их реализации.

7.6.5.2 Источники звука двигателя и шума при движении автомобиля

7.6.5.2.1. Классификация источников звука и шума

Поршневой ДВС является сложным источником звука и шума, которые

формируются совокупностью акустического излучения ряда независимых источников, создающих следующие виды звука и шума (рисунок 7.23) [129].

О сложном характере звука свидетельствует спектральная характеристика дизельного двигателя, приведенная на рисунке 7.24, взятая из работы [132]. Даже при ее реализации узкополосными фильтрами не удастся сформировать реальный звук двигателя и шума автомобиля. Он определяется еще волнами, последовательностью их и сочетаниями гармоник [132]. На характер звука влияет еще резонатор и глушитель. Последние подавляют средне и высокочастотные (выше 600 Гц) составляющие шума аэродинамического происхождения. Смоделировать все это вместе довольно сложно.



Источник шума	Частотный диапазон Гц источника	Воздействующий фактор
Шум сгорания	500-8000	Давление в цилиндре
Удары поршня	2300-8000	Удары поршня
Шум клапанов	500-2000	Удары в седле клапана
Вентилятор	200-2000	Частота вращения
Система выпуска	50-5000	Обороты двигателя
Система впуска	2000	Тип насоса
Механизмы	4000	Число элементов
Дополнит.-ное. оборудование	3000	Передача от них звука

Рисунок 7.23. –Источники шума двигателя внутреннего сгорания

Поэтому при моделировании звука и шума в условиях стендовой имитации движения и колебаний автомобиля на симуляторах в основном используются упрощенные имитационные методы, рассматриваемые ниже.

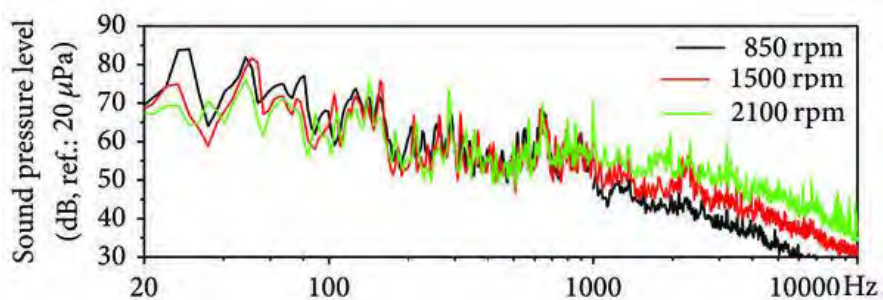


Рисунок 7.24. – Спектральная характеристика дизельного двигателя

7.6.5.2.2. Используемые методы реализации звука двигателя, шума автомобиля

7.6.5.2.2.1 Использование звука при создании компьютерных игр

Для этих целей используются:

– **записанный контент звука**, который воспроизводится в аудиодвижке с учетом задаваемых параметров (rpm, нагрузка, передача и т.д.). Данный метод использовался при создании игры World of Tanks в парке высоких технологий (ПВТ) при формировании звучания танков путем сочетания разных пакетов, включая Wwise [133].

В интернете имеется много готовых файлов звука работы двигателя различных моделей автомобиля в формате .mp3 [133, 134]. К сожалению, ни один из них не обеспечивает реальное звучание двигателя, особенно дизеля.

- **гранулярные сэмплов** в аудиодвижке. Например, Rev или Igniter.
- **встроенные в движки синтезаторы** или кастомных плагинов, создающих звук на основе физической модели.
- **гибридный метод** (всё вышеперечисленное).
- **электронные устройства** на транзисторах и микросхемах, генерирующие звук и шум двигателя [135–136].

В видеоматериалах доклада на конференции Game Industry Conference-2019 [137] (рисунок 7.25, слева) приводится опыт ПВТ осуществления анализа и воспроизведения звуков двигателя для создания правдоподобного звучания в игре Word of Tanks с помощью синтезаторов Serum от Xfer Records gentv путем смешивания волн и гармоник.



Рисунок 7.25. Спектр моделируемого звука в двигателя в синтезаторе в работе [13] (а) и использованная автором индикаторная диаграмма двигателя V8 (б)

Указывалось, что в ПВТ получилось создать прототип движка для звука, где есть все основные волны в синтезаторе, позволяющие переходить к моделированию слоёв двигателя в пакете PureData. Предложено далее конвертировать в плагин для аудиодвижка Wwise. Рекомендовано

использовать язык Faust для программирования звука, предназначенный для программирования DSP. Он отличается от подобных языков тем, что напрямую компилируется в код на C/C++/Java/Rust и его легко встроить в игру.

Однако, чтобы это реализовать, необходимо быть музыкантом и уметь генерировать звуки в синтезаторах и программировать на этих пакетах.

Используются и другие способы. Пакет MathWorks R2023a предлагает использование сочетания готовых модулей Simulink и программы на языке Matlab (.m) для создания звучания работы двигателя (рисунок 7.26) [138].

Для запуска программы в головном меню Matlab необходимо ввести команду

```
openExample('audio/ModelEngineNoiseExample')
```

Как показало ее тестирование звук получается далеким от правдоподобного звучания реального двигателя. Кроме того, после запуска этой программы невозможно выполнить программу в Simulink на языке C/C++ в S-Function Builder и процесс моделирования движения и колебаний. И от этого решения пришлось отказаться.

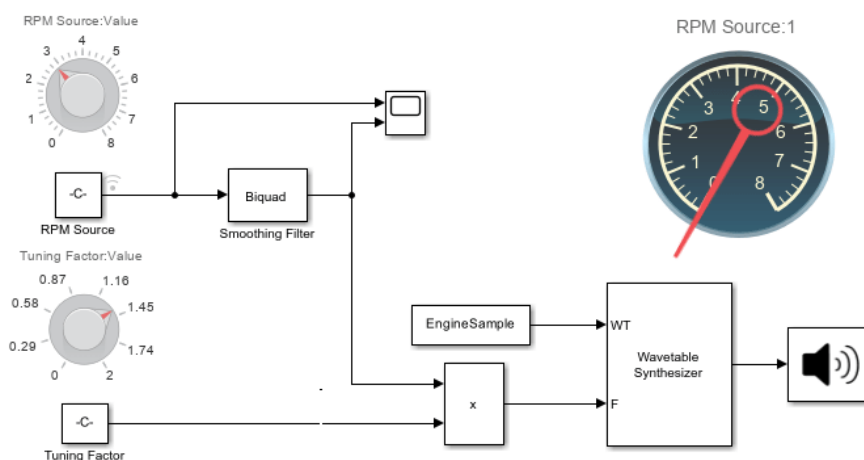


Рисунок 7.26. – Модель звука двигателя, предлагаемая MathWorks R2023a в Help Center

7.765.2.2 Предлагаемый метод реализации звука в Simulink

Для формирования звука двигателя первоначально в качестве исходных данных предлагалось использовать получаемые в процессе моделировании движения автомобиля данные (рисунок 7.27, слева) по оборотам и выдаваемой мощности применительно к дизельному двигателю V8

(ЯМЗ-7511, 240 кВт с развалом 90°, работающий по циклу с очередностью 1-5-4-2-6-3-7-8, интервал последовательности вспышек – 90 градусов).

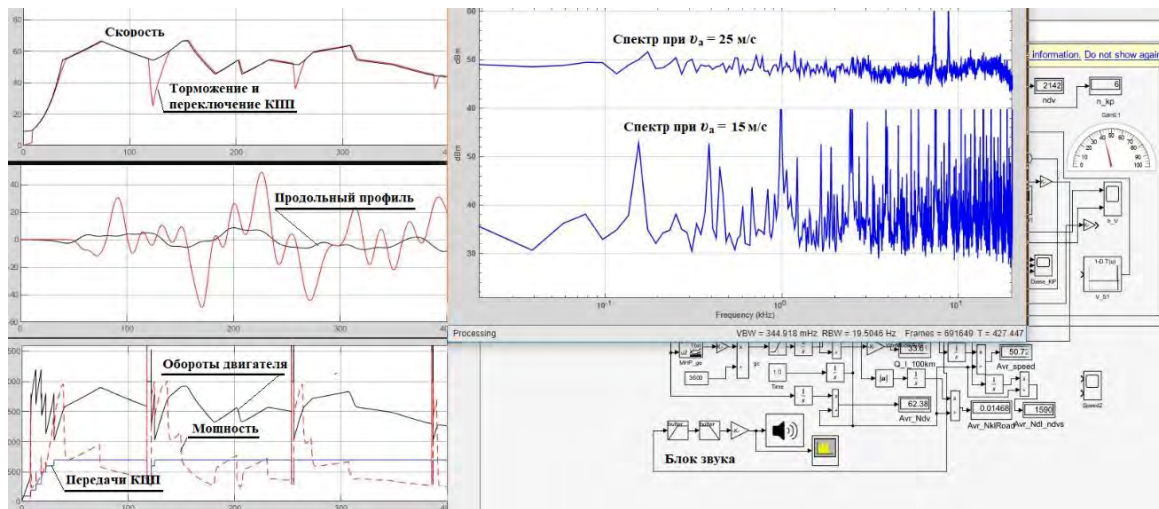


Рисунок 7.27. – Графики изменения скорости движения, профиля дороги, параметров работы двигателя, спектральной характеристики звука двигателя и фрагменты их реализации в Simulink

Предлагался следующий метод на основе аппроксимации индикаторной диаграммы двигателя синусоидой, для которой определяется мгновенное значение оборотов двигателя. По нему определяются необходимое количество точек синусоиды исходя из фиксированного шага моделирования в Simulink. Запускается цикл с учетом этого количества точек по реализации синусоиды по точкам. После достижения счетчиком этого значения определяется новое значение оборотов и цикл повторяется.

Значение синусоиды от угла поворота коленвала φ (уг) двигателя в реализуемой программе на языке C/C++ в S-Function Builder описано следующей зависимостью в программе

$$vs = (\sin(2 * \varphi + M_PI/2)).$$

Ниже приведен фрагмент программы на C/C++ в модуле S-Function Builder Simulink, реализующий это.

```
//-----Моделирование звучания двигателя
if (vt>=720)
{
vt=0;
nlc=86400/ndvs; //nlc=720.0*120/ndvs;
stc=720.0/nlc; //кол-во точек на цикл 90 град
vt=stc/tm;
}
```

```

if (vt<720.0) //Выполнение цикла
{
yg=(M_PI*vt)/180.0;
vs=(sin(4*yg+M_PI/2));
*y4=vs*Ndv; // Выходной сигнал с учетом мощности двигателя
}
vt++;
//-----

```

где vt – счетчик циклов коленвала; stc – шаг моделирования; nls – число колебаний в цикле, yg – угол поворота; $ndvs$ – обороты двигателя, vs – значение синусоиды в зависимости от угла yg , Ndv – текущее значение затрачиваемой мощности двигателя.

Учитывая, что в реальных условиях автомобиль движется с переменной скоростью и оборотами двигателя, последние необходимо перевести в углы φ . Текущие мгновенные обороты двигателя $ndvs$ можно определить на основе моделирования движения автомобиля [15] и их пересчитать в углы поворота коленвала φ по следующим формулам,

$$\omega = n/60,$$

$$\varphi = n/720.$$

Величину vs , являющуюся функцией φ , предложено моделировать в модуле S-Function Builder как повторяющийся цикл 720° (два оборота коленвала) с учетом циклов моделирования Simulink, его шага, на основе счетчика цикла 720° , вычисляемого по переменному значению шага

$$ch=(n/60) t_m,$$

где ch – текущее значение счетчика (целая величина), n – текущие обороты двигателя, t_m – шаг моделирования Simulink.

При шаге моделирования $t_m=0,001$ с процесс цикла 720° при оборотах $n=1200^{-1}$ разбивается на 40 значений (шагов), а при $n=1800^{-1}$ на ~ 27 . Этого достаточно для моделирования звучания. При достижении конечного значения ch счетчик обнуляется, вычисляется новое значение исходя из текущего значения оборотов n и процесс повторяется.

Моделируя таким образом получаем источник звукового воздействия двигателя на автомобиль и водителя. Умножая выходной сигнал на мощность ($W_{\text{сигнал}}=vs*Ndv$) получаем переменный уровень громкости звучания.

Полученный с S-Function Builder сигнал подается вначале на фильтр высоких частот 20 Дб, $f=90$ рад/с, а затем на фильтр низких частот (20 Дб, $f=2500$ рад/с) и далее на модуль звука (рисунок 7.27, справа внизу). Значения фильтров выбраны исходя из основного диапазона работы дизеля (850^{-1} – 2200^{-1}) и возможности реализовать спектр в Simulink при шаге

моделирования $t_m = 0,001$ только до 400 Гц. Однако этого оказалось недостаточно поскольку реальный спектр до 10000 Гц особенно для дизеля с электронным управлением впрыска топлива. Попытки использовать полосовые фильтры в спектре до 400 Гц не дают улучшения звучания и только замедляют процесс компилирования. Звук получается напоминающий тарыхтение старых моделей дизелей. И от него пришлось отказаться.

Для улучшения имитирования звучания предложен иной подход на основе использования записанных файлов звуков реального двигателя автомобиля, воспроизводимых в разработанной специальной программы на C/C++ в MS VS 2022, которая управляет их загрузкой и звучанием на основе данных, формируемых в модуле S-Function Builder и передающихся через бинарный файл. Она воспроизводит файлы, записанные на реальном грузовом автомобиле, которые соответствуют звучанию двигателя на определенных оборотах с учетом громкости/мощности двигателя. Если записать звуки в процессе движения, они еще более лучше отражают звук и шум в автомобиле.

Ниже приведен исходный текст на C/C++ для MS VS 2022 с использованием библиотеки winmm.lib, имеющейся в Windows 10/11. Ее необходимо связать с разработанной программой путем прописывания winmm.lib в строке “Дополнительные зависимости” (Проект->Свойства ->Свойства конфигурации ->Компоновщик->Вход->Дополнительные зависимости и произвести компиляцию этой программы.

// Plsound.cpp Исходный текст для воспроизведения звуков двигателя и шума автомобиля

```
#include <windows.h>
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <iostream>
#include <math.h>
#include <share.h>
#pragma comment(lib, "winmm.lib")
using namespace std;

struct d {
    long dl; // length of a way
    float v; // speed of vehicle
    float zd; // delay in mc
    float spe; // engine speed
    float pwe; // engine power
};
struct d vd;
int main()
{
```



```

FILE* fp;
long vls;
int a;
std::cout << "Sound of car playing";
a = 1;
vd.spe = 1450.0;
while (a == 1)
{
    fp = _fsopen("ma.bt", "rb", _SH_DENYNO);
    fread(&vd, 20, 1, fp); // reading file of a parameters: speed, power engine
    fclose(fp);

    vls = 250.0/vd.pwe; //относительный уровень громкости звука двигателя
    if (vls < 10) vls = 10; //Ограничиваем диапазон изменения
    if (vls > 1) vls = 1;

    // Устанавливаем уровень громкости звука 1/vls
    DWORD volume = 0x0000ffff / vls;
    waveOutSetVolume(NULL, volume);

//Использование звуковых файлов с помощью функции PlaySoundA в зависимости от
оборотов ДВС.
if (vd.spe >= 600.0 && vd.spe < 1000.0) PlaySoundA((LPCSTR)"sound1.wav", NULL, SND_SYNC);
if (vd.spe >= 1000.0 && vd.spe < 1200.0) PlaySoundA((LPCSTR)"sound2.wav", NULL, SND_SYNC);
if (vd.spe >= 1200.0 && vd.spe < 1400.0) PlaySoundA((LPCSTR)"sound3.wav", NULL, SND_SYNC);
if (vd.spe >= 1400.0 && vd.spe < 1500.0) PlaySoundA((LPCSTR)"sound4.wav", NULL, SND_SYNC);
if (vd.spe >= 1500.0 && vd.spe < 1600.0) PlaySoundA((LPCSTR)"sound5.wav", NULL, SND_SYNC);
if (vd.spe >= 1600.0 && vd.spe < 1800.0) PlaySoundA((LPCSTR)"sound6.wav", NULL, SND_SYNC);
if (vd.spe >= 1800.0 && vd.spe < 2000.0) PlaySoundA((LPCSTR)"sound7.wav", NULL, SND_SYNC);
if (vd.spe >= 2000.0 && vd.spe < 2200.0) PlaySoundA((LPCSTR)"sound8.wav", NULL, SND_SYNC);
if (vd.spe >= 2200.0 && vd.spe < 2400.0) PlaySoundA((LPCSTR)"sound9.wav", NULL, SND_SYNC);
if (vd.spe >= 2400.0 && vd.spe < 2600.0) PlaySoundA((LPCSTR)"sound10.wav", NULL, SND_SYNC);
if (vd.spe >= 2600.0 && vd.spe < 2800.0) PlaySoundA((LPCSTR)"sound11.wav", NULL, SND_SYNC);
}
return 0;
}

```

Запись звуковых файлов осуществлялась с помощью диктофона смартфона. Полученные файлы затем конвертировались в программе VLC media player из формата 3gpp в wav. Дополнительно они обрабатывались (обрезались на длительность звучания 5–7 с) в редакторе Audio файлов.

Для реализации звучания звуковые файлы, программа Plsound.exe должны быть помещены в каталог программы Simulink .slx, где также размещается откомпилированная программа визуализации ShowRD.exe, а в S-Function Builder блок на C, формирующий управляющий файл ma.bt.

Аналогичный подход по реализации звука через использования записей работы двигателя применяется в пакете Unreal Engine 5.

7.6.6 О схемной реализации смешанной реальности

Для полноценного моделирования MR необходимо применение вибростенда (рис. 7.28а) со шлемом VR. В качестве стенда целесообразно использование конструктивной схемы 6-DoF платформы на электроактуаторах без гидравлики, применяемой в симуляторах гоночных игр с организацией моделирования по схеме рисунка 7.28б.

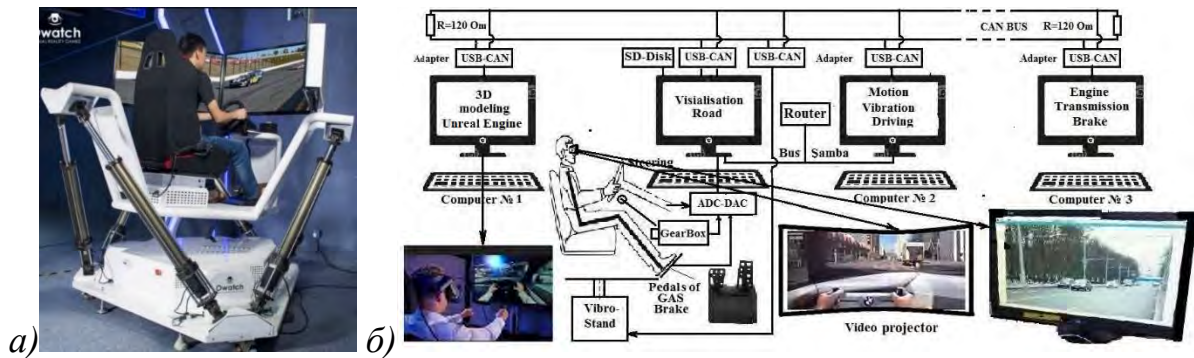


Рисунок 7.28. –Конструкция игрового японского стенда, подходящая для MR (а), схема реализации моделирования со смешанной реальностью (б)

Однако использовать игровые симуляторы для исследовательских целей нецелесообразно из-за невысокого их быстродействия (0,3–0,5 м/с) и невозможности изменить их ПО. Необходимо создание своего специального ПО и более быстродействующего стенда.

Проведенным выше моделированием движения и колебаний на 3-х компьютерах с имитацией обмена по сети CAN BUS по схеме рисунка 7.21б установлено, что электроактуаторы должны обеспечивать скорость перемещения 1,3–1,5 м/с [14]. Этим требованиям отвечают модели актуаторов MOOG MB-EP-6DOF/24 (24000 Н) или Parker ETH080 M10 (25100 Н). Скорость более дешевых китайских актуаторов и их стендов (0,3–0,7 м/с) недостаточна [14].

Для обмена информацией между компьютерами и передачи сигналов с Simulink на вибростенд целесообразно использование сети CAN BUS, как наиболее быстродействующую (1–10 Мб/с) и адресную [14]. Актуаторы MOOG, Parker имеют интерфейс для CAN BUS. Дополнительно необходимо использовать предложенный в работе [14] способ сжатия информации и повышения скорости обмена в сети CAN BUS в 2 раза за счет побайтного занесения двух значений float в поле данных с использованием одинаковых значений идентификаторов объектов управления для двух цилиндров и последующего их разделения в программе микроконтроллеров актуаторов.

Для создания более полного ощущения у водителя необходимо еще сиденье, руль, педали, селектор КПП. Возможно применение варианта без вибрационного воздействия, как это используется на стенде БелАЗ (рисунок 7.29). Готовые элементы для стенда можно приобрести на [Alliexpres.com](https://aliexpress.com).



Рисунок 7.29. – Стенд БелАЗ для дистанционного управления самосвалами в карьере

Важным моментом предлагаемого схемного решения моделирования в MR является необходимость разнесение вычислений на компьютеры, чтобы реализовать режим реального времени и охватить разные подсистемы. А с помощью мини-компьютера Raspberry и АЦП осуществлять снятие информации с органов управления и передачу ее по сети Samba на компьютеры (рисунок 7.28 б).

Необходимо осуществить распараллеливание вычисления подпрограмм по компьютерам в Simulink R2024 [122], рассмотренное в подразделе 5.5.

7.6.7 Выбор пакета 3D проектирования и средств программирования

Для 3D проектирования более целесообразным является использование пакета Cgeo в виду его параметризации: возможностью определять расстояние и изменять соотношения. Но преобразование его файла в формат .fbx требует сложной конвертации из-за текстур и подобное имеет место и в Unity. Обычно для задач виртуальной реальности, не требующих детализации и точных размеров применяется пакет 3D max в связке с Unity ввиду их простоты на основе типовых библиотек и решений. Но в данном случае это не подходит, т.к. нет возможности устанавливать точные размеры и определять расстояние

между точками. Кроме того, программирование в Unity осуществляется на C#, который в 2 раза медленнее чем C++, и он может не обеспечить режим реального времени, учитывая, что придется одновременно решать задачи движения, колебаний и управляемости. Это является определяющим фактором при выборе пакета и языка программирования. C# более предпочтителен для создания игр в сети и VR приложений.

7.6.8 Комплект технических средств

В связи необходимостью компенсировать снижение быстродействия из-за CAN BUS при передаче информации на стенды и появившимися на рынке новыми процессорами, материнскими платами и шлемами рекомендуются для AR/MR следующая конфигурация технических средств:

– процессор серии AMD Ryzen 9 9000 (9950X/9700X3D/9600X (5,7/4,7 ГГц), обеспечивающий более высокую производительность в играх чем у Intel Core i9-14700K (5,2/3,2 ГГц), материнская плата ASUS ROG Strix X670E-E Gaming WiFi либо ASRock X670E Taichi, 32/64 Гбайт памяти DDR5 G.Skill 6400 МГц, видеокарту с 12–16 Гб памяти (GeForce 5060 Ti (16 Гб)) для Unreal Engine, водоблок охлаждения EK-Quantum Velocity2 AM5.

Другим вариантом является использование процессоров Intel Core i9-14900K (5,8/4,4 ГГц ~ 60000 рос.руб) с соответствующему ему платами.

В качестве очков AR/MR реальности рекомендуется HTC Vivo Pro 2 (в РФ 149 990₽/2500 \$, на ebay.com – 1400 \$) с созданием приложений в UnrealEngine, VIVEPORT SDK 1.7.20 (C++), [115, 119, 120].

При использовании AR/MR на компьютерах в связи возросшими параметрами шлемов по video (> 5К на глаз) требуется использование более производительных видеокарт не ниже GIGABYTE NVIDIA GeForce RTX 5080/5070 TI 16 Гб либо Intel Arc B580 12 Гб.

7.6.9 Получение данных из смешанной реальности

Для осуществления получения данных из шлема при смешанной реальности необходимо установить VIVEPORT SDK 1.7.20 [117–119] и разработать программное обеспечение на языке C/C++ для считывания данных с датчиков положения рук испытуемого водителя и сравнивающее их данные с положением органов управления. Кроме того, необходимо разработать обеспечение самой смешанной реальности с использованием пакета Unreal Engine либо Unity [125, 126] или предложенного метода визуализации.

7.6.10 Моделирование движения и колебаний автомобиля

В качестве моделей движения и колебаний автомобиля при смешанной реальности целесообразно использовать, приведенные в данной работе модели и проработанные решения в главах 3, 5, базирующиеся на использовании:

- реального макро и микропрофиля дороги;
- более сложных моделей трения, подвески и движения автомобиля;
- отработанных блок-схем в Simulink;
- модуля S-Function Builder Simulink для реализации макро и микропрофиля дороги, заносимого в оперативную память компьютера, логики переключения передач, управления работой двигателя, что позволяет вести моделирование в режиме реального времени и обеспечивает хорошую сходимость с экспериментом (5-15%) по вибрациям с учетом спектра.

7.6.11 Необходимые устройства для реализации MR

Для реализации MR требуется:

- шлем виртуальной реальности HTC Vive Pro 2 с контроллерами;
- базовые станции 2.0;
- имитационный вибростенд на базе платформы 6-DoF;
- программное обеспечение VIVEPORT SDK 1.7.20 (C#, C++);
- пакеты Creo/NX, Unreal Engine [115];
- разработка программного обеспечения на C++ для получения данных из шлема и контроллеров руки.

7.7 Реализация визуализации в Simulink (VDBS), Blender, UnrealEngine, RoadRunner

7.7.1 Использование (VDBS) в Simulink

В пакете Simulink, помимо рассмотренного его использования, предусмотрены стандартные компоненты Vehicle Dynamic BlockSet, позволяющие реализовать ряд задач по движению, управляемости автомобиля (рисунок 7.30), а также некоторую визуализацию дорожной обстановки.

На рисунке 7.31 показаны блок-схемы движения и управляемости автомобиля параметры компонентов которых задаются в табличной форме. Но в процессе моделирования их нельзя менять. С помощью мыши можно менять угол поворота руля.

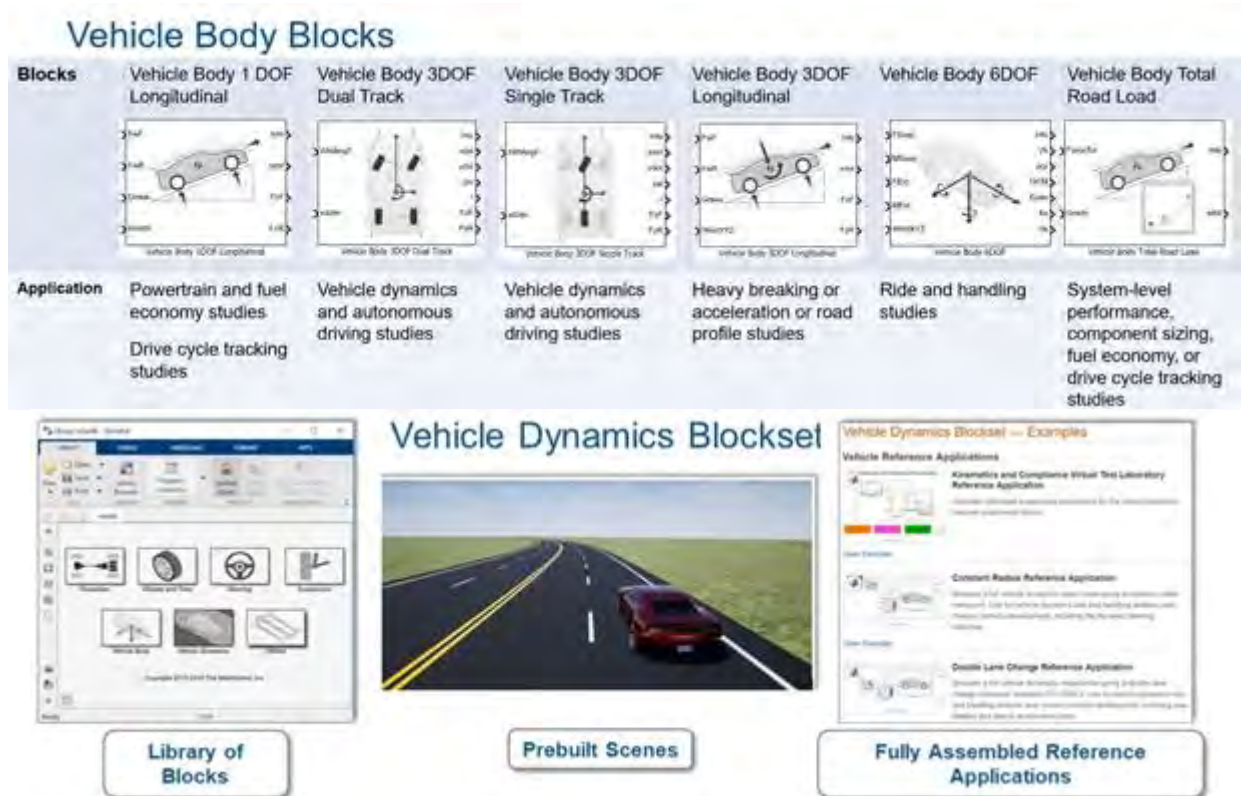


Рисунок 7.30. – Использование пакета Simulink для ряда задач по движению, управляемости автомобиля со стандартными компонентами

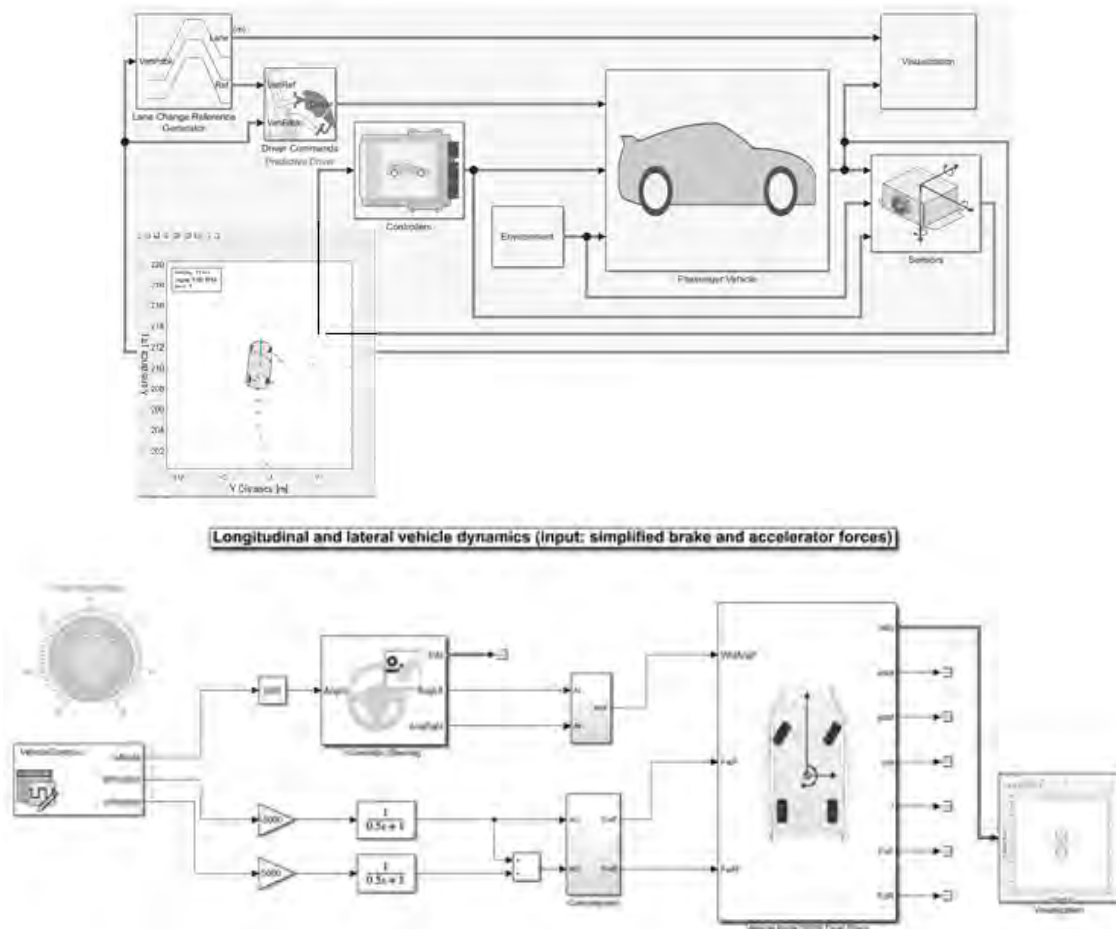


Рисунок 7.31. – Блок-схемы движения и управляемости автомобиля

В Simulink имеется некоторая возможность визуализации дорожной обстановки: создание испытательной дороги и движения по ней на основе готовых моделей UnrealEngine (5-ти типов автомобилей, вид сзади), позволяющих проводить маневры “переставка”, “поворот”.



Рисунок 7.32. – Реализация визуализации дорожной обстановки в Simulink

В целом данные компоненты больше подходят демонстрационных, учебных задач, предварительной оценки параметров машины, но не для исследовательских целей. Однако с каждой новой версии возможности Matlab/Simulink развиваются.

7.7.2 Использование пакета UnrealEngine

UnrealEngine [119, 120, 141] представляет собой пакет для создания игр, мультимедиа в киноиндустрии. Это довольно мощный и сложный пакет объемом 22 Гб. На его освоение в полном объеме с C/C++ требуется около 2-х лет. Он базируется на использовании объектно-ориентированного программирования, большого количества классов для создания различных игровых ситуаций с различными объектами в т.ч. автомобилей. Хотя в пакете в играх и реализуется некоторое имитирование движения и колебаний автомобиля, он не подходит для научных исследований в силу их значительного упрощения и сложен для интеграции с другими пакетами. Но данный пакет можно использовать для визуализации дорожной обстановки. Для решения задачи визуализации в Unreal Engine требуется:

- создание специальной анимационной 3D модели автомобиля;
- создание специальной 3D модели дороги;

– разработка системы управления автомобилем в пакете Unreal Engine.

Разработка 3D анимационной модели автомобиля осуществляется в пакетах Blender, 3Ds Max, Maya либо путем конвертации 3D моделей, созданных в Creo, Unigraphics NX, SolidWorks с доработкой в первых двух. Чтобы использовать модель, созданную в Creo, ее необходимо конвертировать вначале в SolidWorks, затем в 3Ds Max и в последнем преобразовать в FBX.

3D оболочную модель для моделирования в UnrealEngine можно создать на основе 3-плоскостных видов/рисунков/чертежей автомобиля в Blender [142] (рисунок 7.33).

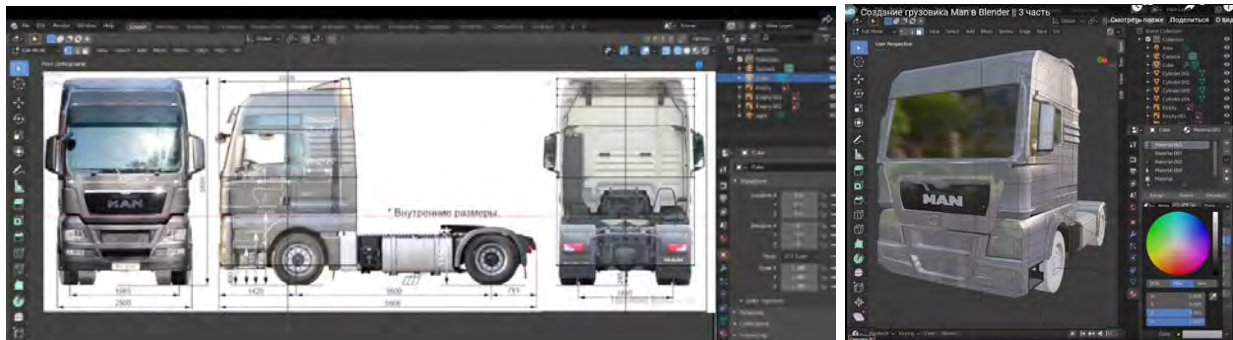


Рисунок 7.33. – Создание 3D модели на основе 3-плоскостных видов/рисунков/чертежей автомобиля

Созданная на их основе такая 3D модель может использоваться также и для рекламы, согласования решений по автомобилю с заказчиком. Так поступает БелаЗ, показывая заказчику через интернет будущий автомобиль со всех сторон и разных точек, еще невоплощенный в металле, как водитель взбирается в автомобиль выезжает из цеха, движется по дороге, в карьере выполняет технологические операции. На основании расчетов в других программах заказчику обосновывается выбор модели машины, ее грузоподъемности. Показывают заказчику обучение водителя на тренажере, реализованном с использованием смешанной реальности. Все это благоприятствует заключению контракта.

7.7.3. Создание специальной 3D модели дороги в UnrealEngine

Здесь возможны 3 варианта:

- создать ее на основе затирания рельефа сцены, создавая возвышенности, низины (рисунок 7.34), создать трассу, разместить на ней объекты (в т.ч. автомобиль), деревья, растительность, здания [10, 11];
- купить через магазин Eric games модель готовой дороги;

– создать сплайны (дороги).

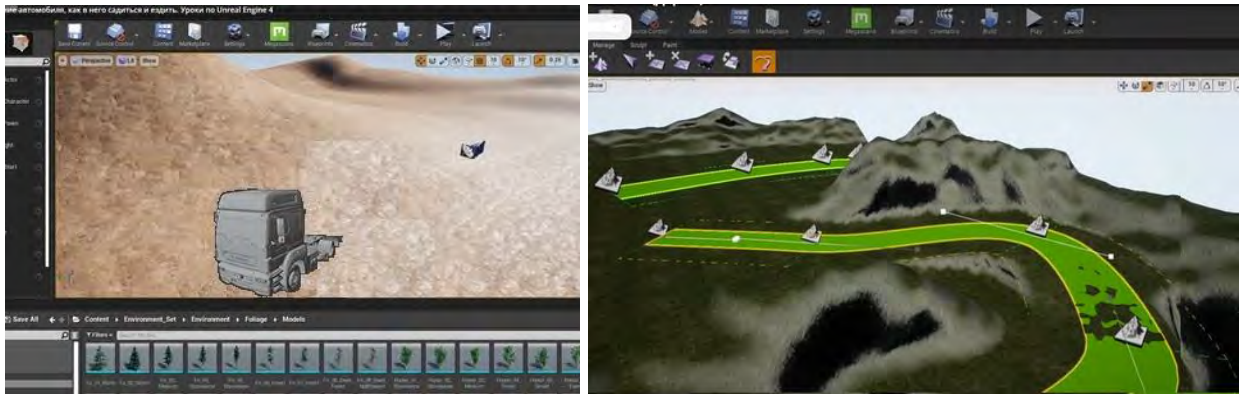


Рисунок 7.34. – Создание дороги в UnrealEngine и с помощью Spline

Недостатком таких способов является отсутствие физического значения высоты дороги по ее длине, что делает невозможным проведения на их основе моделирования движения автомобиля, которому требуется численное значение наклона дороги. Решить эту проблему можно использованием пакета RoadRunner.

Моделирование в UnrealEngine осуществляется двумя способами:

- на основе визуальных компонентов Blueprint;
- с помощью программирования на C/C++ [141].

UnrealEngine построен на основе строгой иерархии классов, их подклассов, компонентов с заданием их параметров и связей [119] (рисунок 7.35, приложения Н, П). В UE их нельзя удалять или изменять: только создавать новые на их основе. В первом случае не требуется программирование в C/C++. Оба способа базируются на одних тех же классах и ограничены в части изменению их функционала. Вариант с C/C++ быстрее.



Рисунок 7.35. – Создание связей компонентов для моделирования в UnrealEngine с помощью Blueprint

Моделирование автомобиля на базе Blueprint в UE 5.6 описано в приложении Н, анимации руля в приложении О, UE 4.27 в приложении П.

Собранный в них материал важен для понимания UE, его логики работы.

Моделирование на основе Blueprint целесообразно для тренажеров, решения вопросов эргономики, а для симулятора больше подходит C/C++.

Из-за того, что управление играми в Blueprint настроено через осуществление нажатия определенных клавиш клавиатуры (дискретное задание значений 1 или 0 к заданному фиксированному параметру в Set), невозможно плавно изменить, например, скорость движения, поворота рулевого колеса, величины торможения. В играх приходится воздействовать путем нажатия кнопок тормоза либо переключением передачи КПП.

Хотя в UnrealEngine задается масса автомобиля, кривая крутящего момента двигателя, передаточное число трансмиссии, радиус колеса этого все же недостаточно для полноценного моделирования движения автомобиля. Также невозможно оценить топливную экономичность. Все нацелено на внешние визуальные эффекты для игр, включая наклоны, опрокидывание автомобиля через параметры подвески. Внешняя визуализация для игр дороги решена хорошо, однако в режиме Blueprint невозможно плавно управлять автомобилем. Варьировать скоростью можно только при программировании в режиме C/C++ или в комбинации его с Blueprint.

Для этого можно использовать функцию 'SetPlayRate' объекта 'UAnimInstance' (приложение Р)

```
// Получаем ссылку на UAnimInstance
UAnimInstance* AnimInstance = GetCharacter()->GetMesh()-
>GetAnimInstance();
// Устанавливаем скорость воспроизведения анимации в Player по
отношению к базовой vb
AnimInstance->SetPlayRate(vr/vb);
а для управления движением (руль вправо_влево)
//Makes the character move to the right/left
void ASpriteCharacter::MoveHorizontal(float AxisValue)
{
    AddMovementInput(GetActorRightVector() * AxisValue);
}
```

Примеры реализации функций на C/C++ даны в приложении Р.

7.7.4 Использование пакета RoadRunner

RoadRunner является интерактивным редактором 3D окружения для создания сцен дорог, разметки маршрута, дорожных знаков, окружающей обстановки, растительности [143]. Он больше ориентирован на создание 3D сцен для симуляции и отладки систем автоматизированного вождения беспилотных автомобилей. Размещает в сценарии транспортные средства, определяет их пути движения и взаимодействия для отладки беспилотных

систем. Моделирование в нем осуществляется на фиксированных скоростях. Сценарий RoadRunner с обстановкой (сценой) может быть конвертирован в другие пакеты (UnrealEngine, Unity, MATLAB/Simulink) для совместного моделирования. Как и UnrealEngine, он позволяет с некоторым приближением сформировать дорогу по длине с заданным рельефом по высоте и ее окружением: остановки транспорта, перекрестки, светофоры, окружающая растительность (рисунок 7.36–7.38). Сцена RoadRunner напоминает театральную с ограниченным пространством и больше подходит создания полигонов. Сформированную таким образом дорогу далее необходимо конвертировать в UnrealEngine и в последнем использовать для визуализации движения.

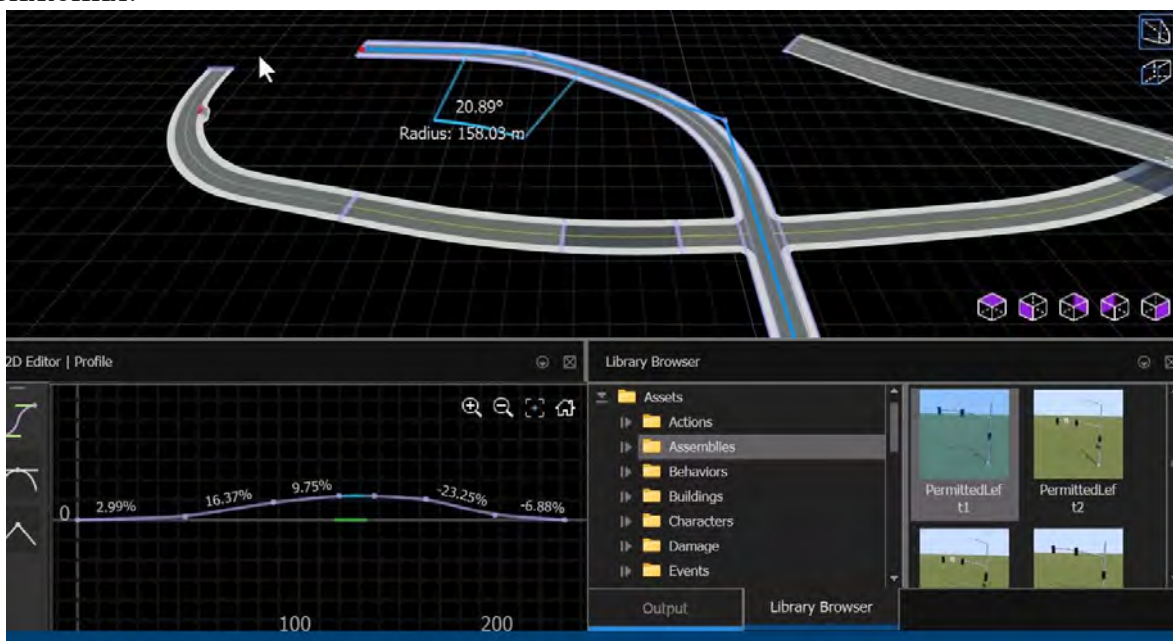


Рисунок 7.36. – Задание дороги в RoadRunner

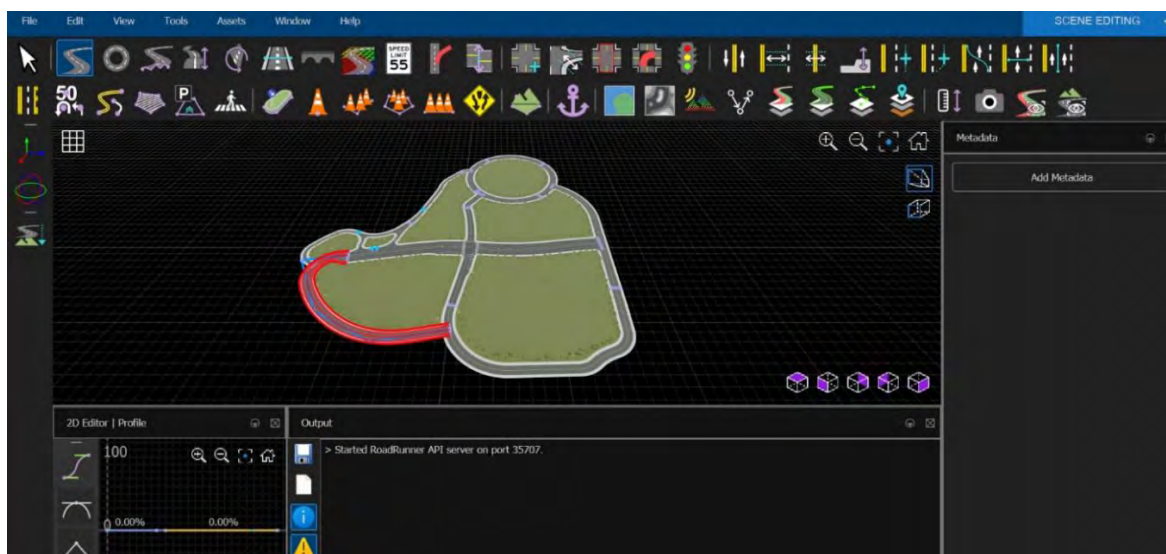


Рисунок 7.37. – Реализация испытательного полигона в RoadRunner



Рисунок 7.38 – Реализуемая обстановка на основе RoadRunner и UnrealEngine

7.7.5 Комбинированное использование Simulink и RoadRunner, UnrealEngine

Несмотря на отмеченные выше все же имеется возможность их совместного использования. Для этого необходимо:

- разработать в RoadRunner дорогу кусочными элементами с учетом ее траектории, высоты профиля, создать окружающую обстановку, конвертировать ее в UnrealEngine. Для создания дороги необходимо располагать массивом продольного профиля дороги. Наиболее простым способом является использование данных сканирования земли [9], далее создать 3D модель автомобиля и сконвертировать ее в UnrealEngine.

Далее возможно два варианта:

- осуществить в UE на основе комбинации Blueprint с компонентом на C/C++ имитирование движения автомобиля, скорость просмотра которого регулируется путем периодического считывания скорости моделирования в Simulink в компоненте на C/C++ путем установки Set.

- реализовать на основе Blueprint имитирование движения автомобиля с постоянной скоростью (20 м/с). Затем путем подбора передаточного числа трансмиссии, чтобы соответствовало этой скорости, воспроизвести это на мониторе, записать это видео (копию с экрана) специальными программами в виде файла. Затем использовать эту запись в варианте п. 7.7.4.

В первом случае требуется задействование параметров поворота рулевого управления на основе электрических сигналов в соответствии траектории движения аналогично играм, как это сделано похоже в BMW [6]. Для реализации требуется задействование Raspberry и двух стационарных компьютеров (6–8-ми ядерных, например, AMD 5 5600X с разгоном до 4–4,6 Гц с объемом оперативной памяти 32 Гб, которое резко повышает быстродействие компьютера). Это позволяет создать более полную иллюзию движения и просмотра обстановки водителем при повороте головы в шлеме

VR при задействовании станций позиционирования. Данный вариант дополнительно позволяет проводить проработку эргономики рабочего места водителя. Но для этого требуется использование полноценной 3D модели автомобиля, включая внутреннюю часть кабины. Обычно для этого применяют модель 3D max, которая по оценке БелаЗ не дает точной геометрии объекта.

Во втором варианте такой возможности нет, но он проще. В этом случае требуется задействование Raspberry и одного стационарного 6-8-ми ядерного компьютера 4-4,6 Гц с объемом оперативной памяти 32 Гб.

7.8 Обеспечение быстродействия моделирования

Проблемным местом одновременного совместного моделирования этих трех задач с задействованием вибростенда и водителя является недостаточное быстродействие существующих компьютеров на базе процессоров Intel и AMD (достигших своего потолка 5 ГГц). Увеличение частоты которых рассматривалось как основной способ повышения быстродействия моделирования. Из-за реализации сложных моделей в полосе 0–28 Гц вынуждены использовать шаг 0,0006–0,001 с, снижающий быстродействие. На снижении быстродействия в 1,5 раза [17] также сказывается передачи сигналов на 6-DoF вибростенд по сети CAN BUS. Для обеспечения быстродействия предложены следующие решения.

7.9.1 Повышение быстродействия моделирования автомобиля за счет перенесение визуализации и передачи данных по сети CAN BUS на отдельный компьютер (№3) согласно рисунка 7.39.

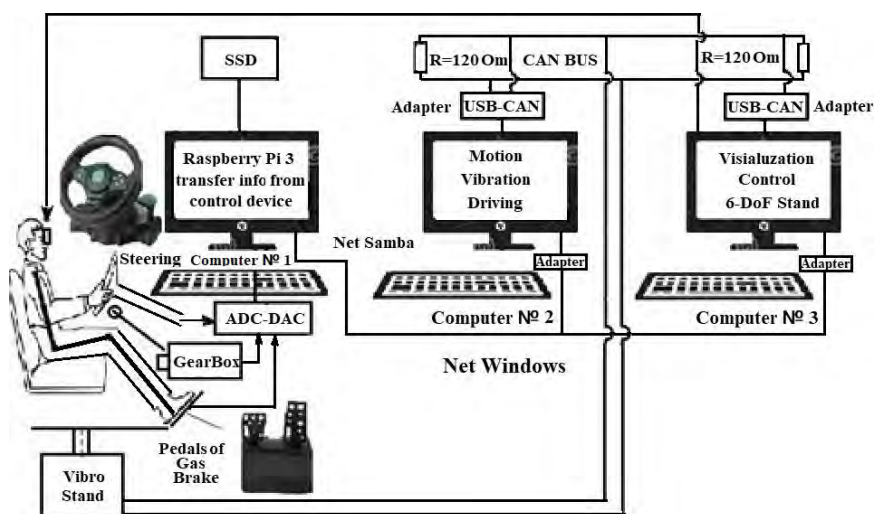


Рисунок 7.39. – Предлагаемая схема реализации визуализации и передачи данных по сети CAN BUS на отдельном компьютере

На первом (мини-компьютер Raspberry) осуществляется считывание данных органов управления автомобиля и передача их через сеть Samba на второй компьютер, где реализуется моделирование движения, управляемости автомобиля, колебаний. На третьем осуществляется визуализация дорожной обстановки, звуковое воздействие и передача информации на 6-DoF платформу вибростенда по сети CAN BUS. Синхронизация процессов на третьем компьютере осуществляется по значению скорости движения со второго компьютера, информация с которого передается в виде бинарного файла на третий.

Это дает выигрыш в быстродействии в 1,35 раза в виду некоторых потерь передачи бинарных данных по сети Windows со второго на третий компьютер.

7.9.2 Параллельное моделирование на двух Simulink на двух компьютерах (рисунки 7.40, 7.41), на втором (для движения, управляемости, колебаний в трансмиссии) и третьем компьютере (вибраций). Они имеют идентичную базу данных макро и микропрофиля дороги.

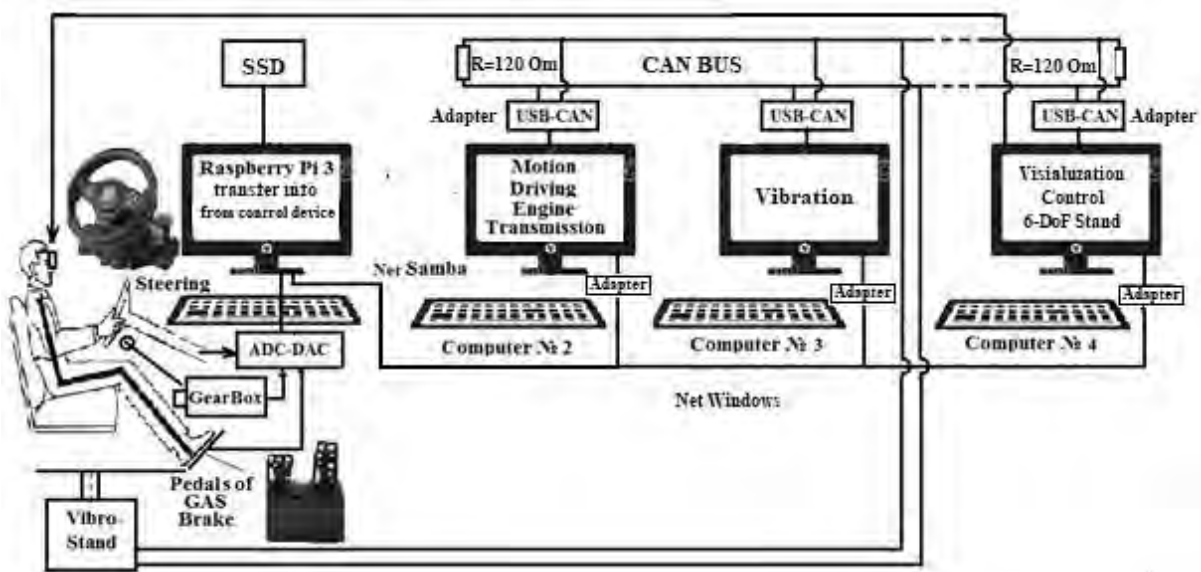


Рисунок 7.40. – Предлагаемая новая схема реализации моделирования автомобиля в режиме реального времени со смешанной реальностью и вибростендом

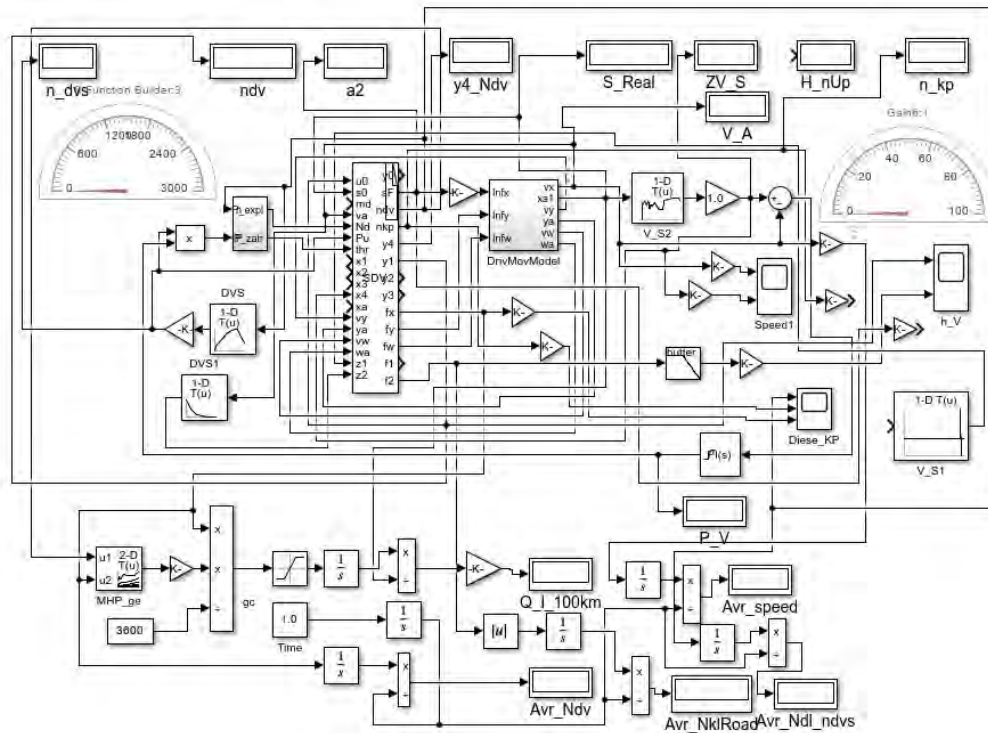


Рис. 7.41. – Блок-схема модифицированного варианта реализации моделирования движения, управляемости автомобиля в Simulink

Блок-схема моделирования вертикальных колебаний автомобиля показана на рисунке 7.42.

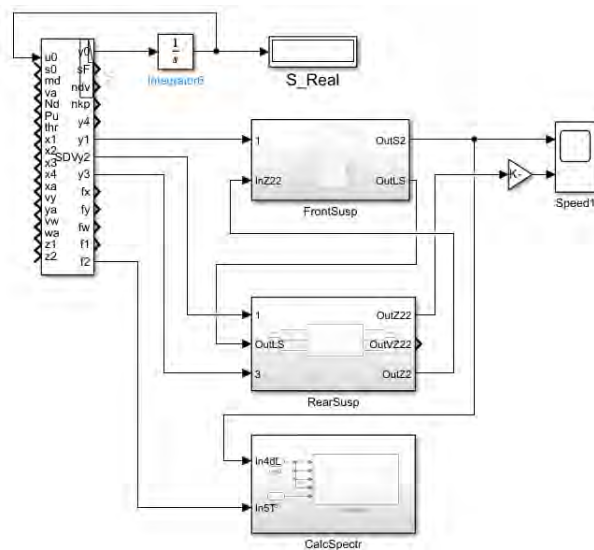


Рисунок 7.42 – Блок-схема моделирования вертикальных колебаний автомобиля, реализованная на отдельном компьютере № 3 с Simulink

На первом (мини-компьютер Raspberry) осуществляется считывание параметров с органов управления автомобиля и передача их через сеть Samba на второй основной более мощный компьютер, где реализуется

моделирование движения, управляемости автомобиля, колебаний в трансмиссии, на третьем компьютере вертикальных колебаний, на четвертом визуализация дорожной обстановки, звуковое воздействие, передача информации на 6-DoF платформу вибростенда по сети CAN BUS.

Синхронизация процессов в них осуществляется по значению скорости движения со второго компьютера в виде бинарного файла на третий. Исходя из того факта, что скорость движения не так мгновенно меняется, это позволяет квантовать не так часто (5 Гц). Для передачи информации требуется всего 0,2 мс. Недостатком этого метода является возможность пропуска цикла из-за загруженности сети, но это не так критично при таком распределении задач по компьютерам. Информация с №2, включая параметры скорости и звука передается на компьютеры № 3, 4 по общей компьютерной сети Windows в виде бинарного массива float файла с записью на накопитель .m2. Этот файл создается с помощью функции fwrite() в программе C/C++ модуля S-Function Builder Simulink при определенных значений счетчика циклов моделирования и не тормозит его процесс. Затем он считывается в компьютерах № 3, 4 с помощью функции fread() и преобразуется в сигналы перемещения (путем интегрирования), шлема VR и CAN BUS для 6-DoF вибростенда на электроактуаторах. Единственным условием является использование высокоскоростного сетевого адаптера на втором и третьем компьютере, а также накопителей .m2. В дальнейшем под другие задачи предполагается задействование 5-го компьютера с дополнительным Simulink. На данном этапе можно ограничиться использованием CAN BUS только для компьютера №4 и стенда. С развитием системы CAN BUS может понадобиться и другим.

Данное решение обеспечивает повышение быстродействия в 1.5 раза (с 840 до 560 с). Недостатком предложенного решения является также необходимость ручного запуска процессов на трех компьютерах, приводящий к возможной несогласованности запуска до 2 сек между процессами движения, колебаний и визуализации.

7.8.3 Повышение быстродействия за счет увеличения оперативной памяти компьютера и частоты процессора

Согласно рекомендациям MathWorks к компьютерам с R2024 предъявляются следующие требования:

Операционная система

Windows 11 (версии 22H2 или выше) либо Windows 10 (версии 22H2);

Процессор

Минимум: Любой Intel или процессор AMD x86-64 с двумя или больше ядрами

Рекомендуется: Любой Intel или процессор AMD x86-64 с четырьмя или больше ядрами и поддержкой системы команд AVX2

Примечание: Будущий выпуск MATLAB потребует процессора с поддержкой системы команд AVX2.

ОПЕРАТИВНАЯ ПАМЯТЬ

Минимум: 8 Гб.

Рекомендуется: 16 Гб.

Графика. Определенных требований в MatWorks R2024 по видеокарте пока не предъявляется (минимум 1 Гб памяти), наличие OpenGL 3.3, но в будущем потребуется мощная видеокарта т.к. планируется реализовать в нем визуализацию движения автомобиля. В случае задействования в процессе моделирования UnrealEngine, для него однозначно потребуется мощная видеокарта с объемом 12–16 МГб и частотой >2500 МГц.

Storage

4.2 GB for just MATLAB.

4-6 GB for a typical installation.

24 GB for an all products installation.

An SSD is strongly recommended.

Примечание. Неясно, как правильно трактовать перевод пункта **Storage** о 24 Гб, то ли это объем для размещения программ или все же желательный объем оперативной памяти на компьютере. Возможно MatWorks в рекомендациях ориентировался на несложные задачи, не требующие памяти и задействование параллельно других задач в т.ч. S-Function Builder. В моих исследованиях требования по 16 Гб памяти выполнялись.

При осуществлении модернизации компьютера (увеличение оперативной памяти до 32 Гб, использование более быстродействующего процессора AMD 5 5600X) было выявлено значительное повышение быстродействия. Поэтому в дальнейшем акцентировались на возможность реализации 3-х основных задач на одном модернизированном компьютере (рисунок 7.30, computer № 2) в отличие от работы [12].

Для получения более корректных сравнительных результатов, исключающих человеческий фактор, исследование проводилось без задействования мини-компьютера Raspberry (№1) путем задания режима движения в блоке V_S2 (рис.1) на Windows 10 64-битной версии 22H2, которая по информации в интернете на 20% пока быстрее Windows 11 с использованием блок-схемы (рисунок 7.38).

Имитирование усложнения модели (как пространственной) осуществлялось за счет добавления дополнительных блоков подпрограмм вибраций (рисунок 7.43).

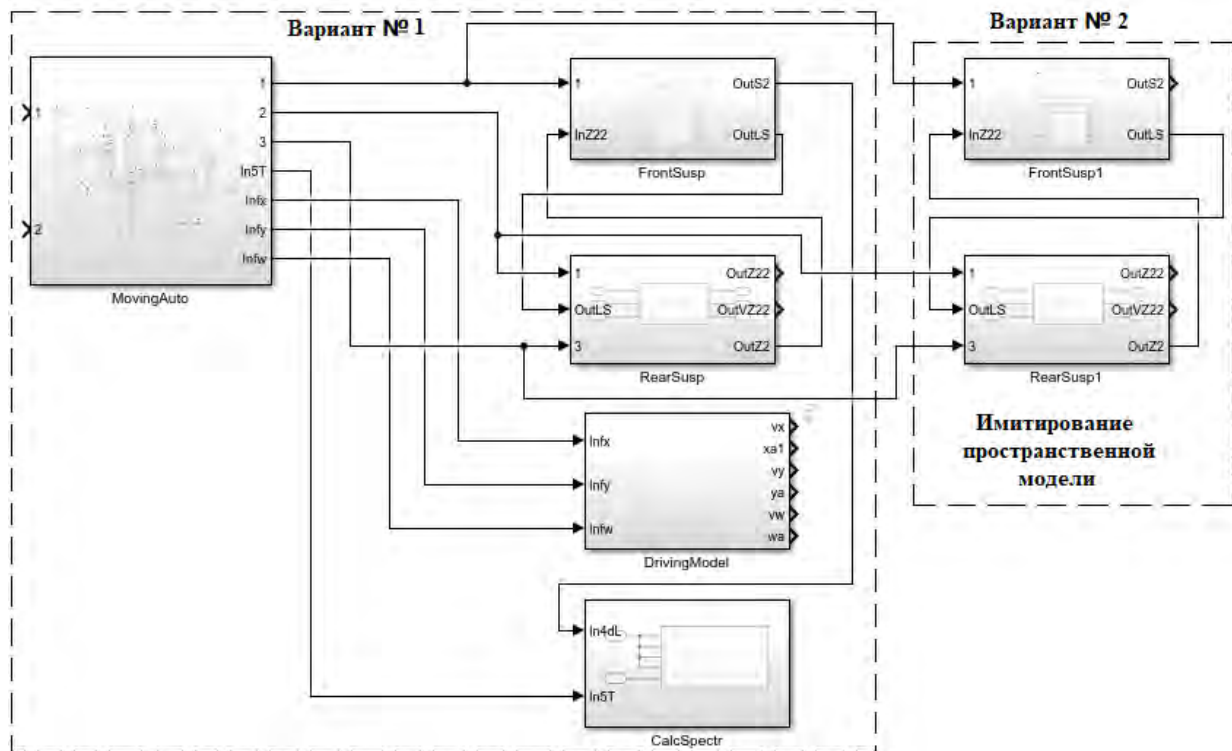


Рисунок 7.43. – Общая блок-схема реализации моделирования вибраций автомобиля в Simulink

Сравнительное исследование осуществлялось на трех компьютерах, имеющих следующие конфигурации:

– Вариант 1. Процессор Ryzen AMD 5 2600, 6-ти ядерный, 3,4-3,9 Гц, MB ASUS Prime B450 Plus, видео карта Geforce RTX 730, 2 Гб, 700 МГц, ОП Patriot 16 Гб (2x8), 2666 МГц.

– Вариант 2. Процессор Ryzen AMD 5 2600, 6-ти ядерный, 3,4-3,9 Гц, MB ASUS Prime B450 Plus, UEFI 4401, видео карта Geforce RTX 730, 2 Гб, 700 МГц, ОП Kingston Fury Beast 32 Гб (2x16), 3200 МГц.

– Вариант 3. Процессор Ryzen AMD 5 5600X, 6-ти ядерный, 3,7–4,6 Гц, MB ASUS Prime B450 Plus, UEFI 4202, видеокарта Geforce RTX 3050, 8 Гб, 1580–1800 МГц, ОП DDR-4 Kingston Fury Beast 32 Гб (2x16), 3200 МГц.

7.9.4 Примененные модели автомобиля и реализация

Проверка предложенного решения на первом этапе проводилось на одном компьютере на плоских моделях автомобиля, описанных в главе 2, которые позволяют вести совместное моделирование движения, колебаний и управляемости. Эти модели описываются 4–6-тью дифференциальными уравнениями второго порядка с учетом сложных моделей трения для

реализации моделирования в полосе 0–28 Гц, включая трансмиссию, при шаге 0,0004–0,001 с в режиме реального времени.

Имитирование усложнения модели (как пространственной) на втором этапе осуществлялось за счет добавления дополнительных блоков подпрограмм вибраций (включающих 16 интеграторов, вариант № 3 на рис. 3).

7.8.5 Результаты исследования

В процессе исследования рассматривались два варианта передачи информации на стенд:

- вариант № 1 с использованием передаваемых данных (9 параметров: 6 сигналов 6-DoF стенда, скорости движения, оборотов и мощности двигателя), записываемых в виде бинарного файла на накопитель .m2, 6 Мб/с (C:\\temp\\stnd.bt) для считывания их в последующем в компьютере №3 для формирования в нем сигналов CAN BUS;

- вариант № 2 с использованием компьютера № 2 для формирования сигналов для 6-DoF стенда, передаваемых с него по сети CAN BUS и обменом параметров визуализации и звука (скорости движения, оборотов и мощности двигателя) на основе бинарного файла.

На 1-м этапе использовался фиксированный шаг 0,0006 решателя ode5(DomandPrince).

В процессе моделирования по первому варианту подбирались значения счетчика, при котором происходит осуществление записи, чтобы обеспечить частоту 600 Гц (6×100 Гц). Исходили из того, что 10 значений квантования на частоте 10 Гц на электроактуаторе стенда достаточны для воспроизведения вибрационного воздействия на водителя [9]. После 12 Гц на электроактуаторах наблюдается завал характеристики и дальнейшее увеличение квантования нецелесообразно.

В варианте № 2 использовался фиксированный шаг 0,0004 с решателя ode5(DomandPrince) для приведения в соответствие времени моделирования, Real Time и визуализации дорожной обстановки, реализуемой с помощью новой разработанной программы ShowRD.exe, запускаемой на компьютере № 2 параллельно с программой звука.

Результаты исследований приведены в таблице 7.4.

Таблица 7.4. Затраты времени на моделирование при различных вариантах

Вариант моделирования	Время, с, вариант компьютера		
	Вариант № 1	Вариант № 2	Вариант № 3
1.Передача информации через бинарный файл без имитирования пространственных. моделей	547	540	300
2.1 Передача информации через бинарный файл с имитированием пространственных. моделей	600	545	328
2.2 Передача информации по CAN BUS с имитированием пространственных. моделей	898	653	599

Примечание. 1. В варианте 2.2 (Компьютер № 2) использован шаг 0,0004 с для реализации продольных и поперечных вибраций, вызываемых колебаниями в трансмиссии.

2. Реальное время моделирования составляет 470 с.

Как видно из таблицы использование модернизированного компьютера:

- увеличивает быстродействие в 1,82; 1,72; 1,5 раза, обеспечивая с запасом необходимое быстродействие в Real Time ($t=650$ с);

- позволяет реализовать три основные задачи на одном компьютере (№2), сокращает необходимое число компьютеров с четырех до трех, исключает возможные проблемы при передаче информации на другие компьютеры, а высвободившийся компьютер (№ 3) можно использовать для других задач;

- имеющийся запас по быстродействию позволяет загрузить в него (№ 2) некоторые задачи других подсистем автомобиля.

- передача инфо через бинарный файл с (без) имитирования пространственных моделей обеспечивает наименьшее время моделирования, но требует дополнительного компьютера (№ 3) для визуализации дорожной обстановки и звука;

- CAN BUS является с точки зрения минимизации числа компьютеров наиболее оптимальным вариантом для реализации 3-х основных задач.

Предложенное решение позволяет значительно снизить затраты на компьютерную технику по сравнению с более мощными и дорогими компьютерами, дает возможность большему числу исследователей на более слабой технике заниматься моделированием.

Даже в варианте № 1 при слабом компьютере по предлагаемому схемному решению возможна реализация Real Time при использовании передачи инфо через бинарный файл с имитированием пространственных моделей. Но этот вариант потребует использования еще одного компьютера (№ 3) для реализации визуализации дорожной обстановки, звука двигателя, шума движения автомобиля.

Полученные данные на основе предложенного решения по использованию измененной блок-схемы, увеличения объема памяти и производительности процессора позволяют более оптимальным способом с минимальными затратами (до 200 \$ на модернизацию) реализовать на одном компьютере (№2) три основные задачи моделирования, а высвободившийся компьютер (№ 3) использовать для других задач.

Предположительно в повышении быстродействия основную роль сыграло увеличение объема оперативной памяти и оптимальное сочетание соответствия параметров контроллера памяти в самом процессоре с оперативной памятью поскольку повышение быстродействия процессора обычно пропорционально увеличению его частоты. Это обусловлено особенностями архитектуры процессоров AMD Ryzen 5. При использовании памяти Patriot Viper 32 Гб 3600 МГц компьютер на данной материнской плате не стартует. Если исходить, что для ОС Windows 10 требуется 6 Гб памяти, а для MathWorks 16 Гб получается в сумме 22 Гб, это объясняет такое влияние 32 Гб ОП по сравнению с 16 Гб. Отсюда следует, что для повышения быстродействия моделирования в Simulink необходимо использовать на компьютере 32 Гб оперативной памяти с частотой >3200 МГц (6400 DDR-5).

Для реализации моделирования автомобиля со смешанной реальностью с симулятором предлагается следующая схема (рисунок 7.44) с задействованием испытательных гидропульсаторов, обладающая новизной данного решения с разделением вычислений по компьютерам.

На первом (мини-компьютер Raspberry) осуществляется считывание данных органов управления автомобиля и передача их через сеть Samba на второй компьютер, где реализуется моделирование движения, управляемости автомобиля. На третьем моделируются наиболее ресурсно затратные колебания. Синхронизация процессов на третьем компьютере осуществляется по значению скорости движения со второго компьютера, информация далее передается в виде бинарного файла на четвертый. На четвертом осуществляется визуализация дорожной обстановки, звуковое воздействие и передача информации на 6-DoF платформу вибростенда и стенды по сети. Данное решение позволяет обеспечить необходимое быстродействие и компенсировать потери при передаче информации по CAN BUS.

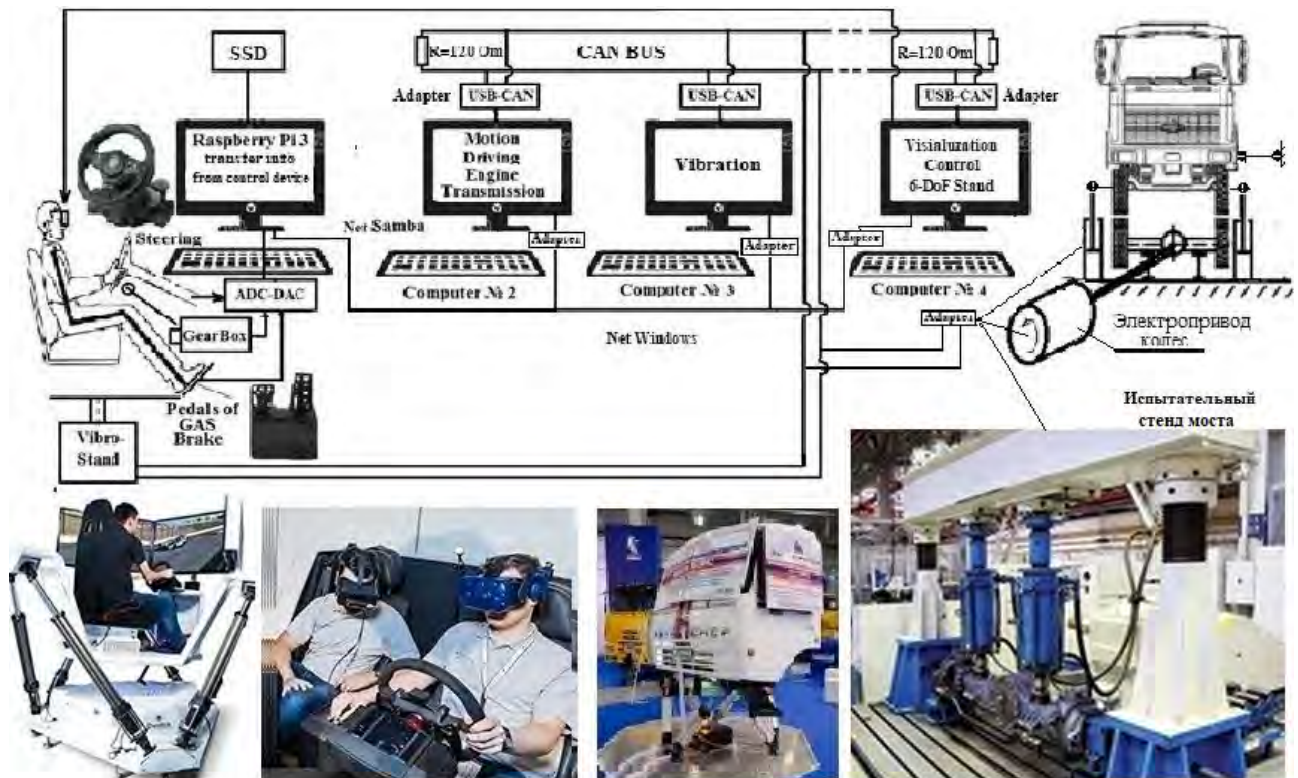


Рисунок 7.44. – Предлагаемая схема моделирования автомобиля со смешанной реальностью с симулятором в сочетании с испытательными стендами

7.9 Концепция использования смешанной реальности для эргономики

7.9.1 Обычно для проведения исследований автомобиля в дорожных или стендовых условиях необходим сам автомобиль, изготовление опытного образца которого обходится в 8–10 раз дороже серийного. Требуется еще установка измерительной аппаратуры, датчиков и проведение самих испытаний. Все это выливается в значительные затраты.

Одним из направлений снижения затрат является использование методов виртуальной (VR), смешанной реальности (MR) при отработке эргономики кабины, являющейся важной частью процесса проектирования нового автомобиля или его модернизации. Обычно это проводится на макетах (рисунок 7.45), требующих их разработки, изготовления, проведение эргономических замеров, что требует много времени и затрат.

При отработке эргономики кабины приходится решать много вопросов:

- удобство размещения водителя;
- правильное размещение органов управления и удобство ими пользования;
- обзорность с кабины и другое.

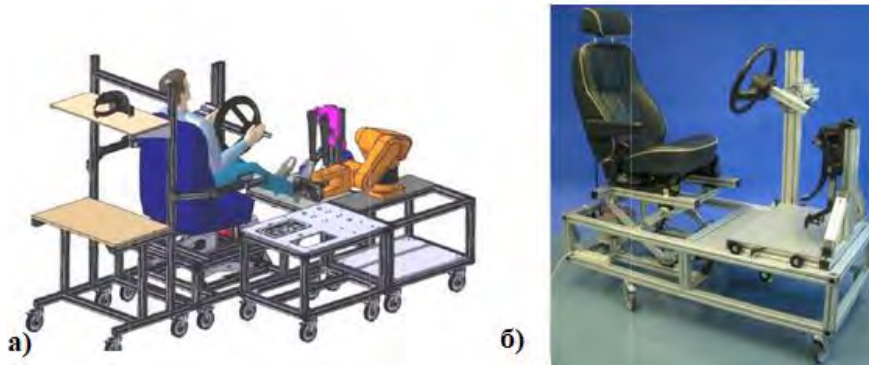


Рисунок 7.45. – Общий вид простых стендов для отработки эргономики рабочего места водителя

Обычно эта процедура проводится в статике, что не всегда соответствует ощущениям водителя в условиях движения.

Особенностью решения эргономики кабины в MR заключается в том, она должна соответствовать условиям движения под воздействием вибраций и дорожной обстановки, т.е. захватывать и эти задачи.

Сейчас с развитием компьютерных технологий виртуальной и смешанной реальности появилась возможность без изготовления макета кабины провести отработку эргономики с учетом безопасности движения на компьютерных моделях при задействовании человека (рисунок 7.46) [112–114]. Но процесс довольно сложный, поскольку требует разработки специального программного обеспечения с участием команды разработчиков.

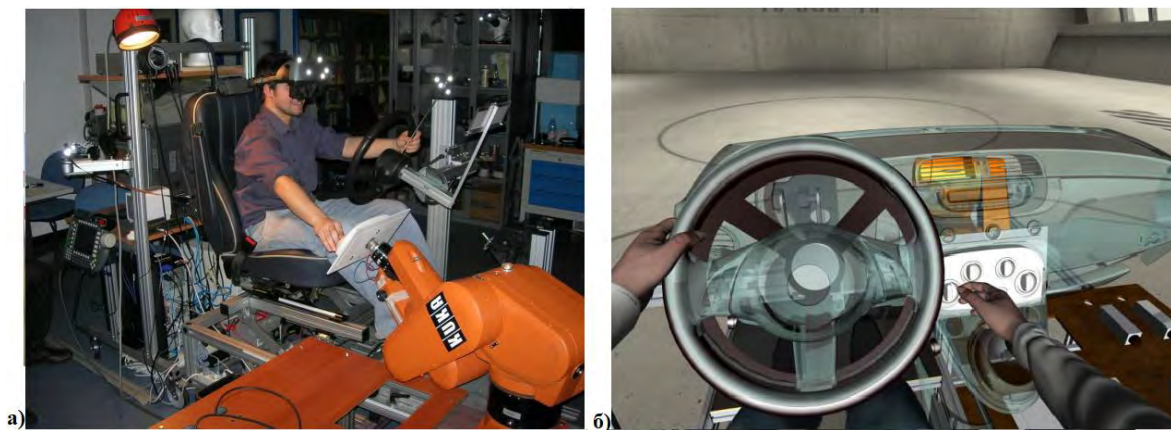


Рисунок 7.46. – Отработка эргономики рабочего места водителя с помощью MR

7.9.2. Концептуальный подход к решению задачи эргономики в смешанной реальности

Рассмотрим водителя в кабине и его имитирование на стенде (рисунок 7.47).

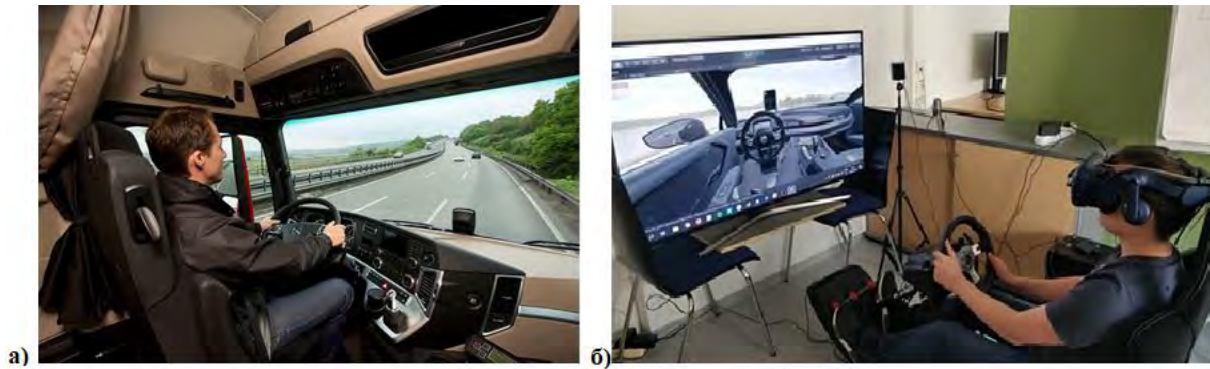


Рисунок 7.47. – Водитель в кабине (а), стенд университета Ессена ФРГ (б)

Для моделирования на стенде может быть применен следующий подход (алгоритм) в случае использования Creo:

- разрабатываются 3D модели водителя и кабины:
- выбирается базовая точка в 3D модели Creo Parametric [115] и от нее определяются координаты исследуемого органа управления (например, ручка КПП, клавиша чего-то), их значение x , y , z . И на основе 3D моделей определяется возможность дотянуться до того или иного органа управления. Далее эти значения закладываются в память программы определения положения руки;
- проводится моделирование движения и колебаний автомобиля совместно с моделированием MR. В процессе которых испытуемый пытается дотянуться визуальнo до органа управления (например, ручки КПП) и выполнить операцию с ней;
- считываются значения с датчика трекинга перчатки руки через SDK. И на основе полученных значений от перемещения трекинга перчатки руки сравниваются со значениями, заложенными в памяти;
- при их совпадении должен поступить сигнал на электромоторы кончиков пальцев либо звуковой.

Важно при этом контролировать управление водителем автомобиля т.е. одновременно должна решаться задача движения и управляемости. Для чего необходимо снятие информации с рулевого колеса, траектории движения и поворота головы с помощью SDK. Вторым способом при имитационном моделировании без шлема является использование поворачивающего управляемого проектора, связанного с рулевым колесом и траекторией движения и который смещает видео дорожной обстановки вправо, влево на экране в зависимости от поворотов рулевого колеса и траектории дороги.

И на основании всего этого судить о правильности созданной компоновки рабочего места водителя.

Выводы по главе 7

1. На основе предложенных моделей рулевого управления и совместного моделирования движения и колебаний автомобиля, блок-схем их реализации получены значения средней скорости 67,5/65,6 км/ч, расхода топлива 70,2/ 62,3 л/100 км при ручном и автоматизированном управлении КПП с участием водителя и задействованием педали газа, оценена траектория движения автомобиля при выполнении маневров “переставки” и “поворота”, показавшая хорошее совпадение (расхождение $< 5 \%$) в сравнении с экспериментальными результатами поворота колес на месте, приведенными в работе [39],

2. Предложены оригинальная общая схема и метод полунатурных испытаний автомобиля (включая рулевое управление) с использованием стенда с беговыми барабанами, выполненного на основе задних мостов с шинами, скорость вращения которых управляется компьютером.

3. На основе анализа кинематики 6-DoF платформы, созданной ее модели и проведенного моделирования перемещений кабины обосновано использование электроактуаторов в качестве исполнительных устройств для виброплатформы, определены необходимые усилия электроактуаторов (20280 Н) и другие параметры, позволившие подобрать подходящие на рынке платформы (MOOG MB-EP-6DOF/24, Parker ETN080 M10).

4. Предложен метод реализации имитационного моделирования пространственных и более сложных моделей (которые невозможно реализовать в режиме реального времени), заключающееся в проведении вначале моделирования на плоских моделях с одновременной записи перемещений и скорости автомобиля по дороге в файл Simulink. А затем использовать данные этих записей для задания режима движения для пространственных моделей, решаемых в режиме замедления. И через пространственную модель определять нагрузки и далее на основе их в пакете ANSYS проводить оценку напряженно-деформированного состояния рам, кабин, мостов и других узлов, испытывающих крутящие нагрузки.

5. Разработана программная реализация имитационного совместного моделирования движения, колебаний и управляемости автомобиля с участием водителя, распределением задач по компьютерам и обмена между ними по общей и сетями Samba и передачи информации с Simulink по сети CAN BUS на стенды, дана подробная инструкция по установке необходимого ПО.

6. Разработан способ сжатия информации и повышения скорости обмена в сети CAN BUS в 2 раза за счет побайтного занесения двух значений float в поле данных с использованием одинаковых значений идентификаторов

объектов управления для двух цилиндров и последующего их разделения в программе микроконтроллеров актуаторов, что является важным и новым.

7. Предложен программный метод реализации визуализации дорожной обстановки на основе записанного в дороге видео, подвергнутого обработке, которое воспроизводится путем трансляции в шлеме виртуальной реальности в соответствии скорости моделирования автомобиля в пакете Simulink с помощью разработанной программы управляемого видео на основе определенной функции значений задержки от скорости движения.

8. Рассмотренная ситуация с реализацией звука работы двигателя и шумов в автомобиле показала, что многие предлагаемые методы на основе синтезированного звука и технических средств не обеспечивает реального звучания и не могут быть применены в Mathlab/Simulink при имитационном моделировании на симуляторах в режиме реального времени при MR.

9. Рассмотрен метод получения звука двигателя на основе использования индикаторной диаграммы двигателя синусоидой с аппроксимированием ее по точкам, значение которых и количество определяется исходя из оборотов двигателя на основе счетчика. Звук получается напоминающий тарахтение старых моделей дизелей. Получить характерное звучание дизеля с электронным управлением впрыска топлива не удалось: недостаточен диапазон частот, реализуемых в Simulink, нет волн звуков. И от него пришлось отказаться.

10. Для улучшения звучания предложен новый подход на основе комбинации записей коротких файлов звука двигателя при фиксированных оборотах на реальном автомобиле, которые воспроизводятся в разработанной программе на C/C++ в зависимости оборотов и мощности двигателя. Информация для программы, реализующей звучания двигателя и шума, берется из бинарного файла обмена, формируемого программой S-function Builder Simulink. Данное решение более лучше отражает звук и шум в автомобиле.

11. Предложен алгоритм решения задач эргономики при смешанной реальности с использованием шлема и трекеров на руках.

12. Выбраны программные и технические средства реализации MR.

13. Предложены комбинированные способы использования Simulink для моделирования движения, колебаний, управляемости автомобиля и RoadRunner, UnrealEngine для реализации визуализации дорожной обстановки.

14. Протестированы различные варианты схемных решений, направленных на повышение быстродействия моделирования. Предложена блок-схема (рисунок 7.44) обладающая новизной с задействованием

испытательного оборудования с разделением вычислений по компьютерам, обеспечивающая необходимое быстродействие. Установлено, что увеличение оперативной памяти с 18 до 32 Гб и решения с разделением задач по подпрограммам:

- увеличивает быстродействие в 1,82; 1.72; 1,5 раза по сравнению с AMD Ryzen 5 2600, 6-ти ядерного, 3,4 Гц, ОП 16 Гб (2x8), 2666 МГц, обеспечивая с запасом необходимое быстродействие в Real Time ($t=650$ с);

- позволяет реализовать три основные задачи на одном компьютере (№2) для плоских моделей, сокращает необходимое число компьютеров с четырех до трех и исключает возможные проблемы при передаче информации на другие компьютеры, а высвободившийся компьютер (№ 3) можно использовать для других задач.

ЗАКЛЮЧЕНИЕ

Основные научные результаты работы

1. Разработаны научные основы и методология моделирования системы “дорога–автомобиль–водитель” на качественно новом уровне для решения задач движения, колебаний, управляемости, нагруженности, долговечности, топливной экономичности автомобиля с учетом воздействия вибраций дороги, ее обстановки на водителя на основе полунатурного моделирования со смешанной реальностью с симулятором и решения по их реализации.

2. Предложены, обоснованы и апробированы, обладающие новизной:

- новые методы получения и использования совместных массивов продольных профилей дорог на основе GPS, микропрофиля по ускорениям неподрессоренных и поддрессоренных масс и нагрузкам на мост;

- более совершенные модели (трения, амортизаторов, рессор, шин) и методика моделирования ТС на их основе с расхождением 5-15 %;

- использование модуля Simulink S-Function Builder и разработанной в нем программы на C/C++ для повышения быстродействия за счет занесения в ОП совместного массива дороги (96000 точек) и реализации через нее управления процессом моделирования, логики переключения КПП на основе баланса сил, ускорения ТС, точек снижения скорости, передачу звука, информации на стенды, что дало возможность осуществить режим Real Time, проводить оценку тягово-скоростных свойств, экономичности, плавности хода машины, расчеты нагрузок на узлы, долговечности и выбор оптимальных параметров на стадии проектирования, чего невозможно сделать в Matlab;

- метод оценки долговечности узлов колесных машин через цепочку этапов: моделирования движения и колебаний автомобиля, получения по ним нагрузок, напряжений в узлах через КЭМ модель, далее расчет долговечности по усталостным характеристикам материалов и систематизации данных напряжений методом дождя, позволившие оценить долговечность картера заднего моста МАЗ в 1275 тыс. км;

- предложенная оригинальная схема с распределением вычислений по компьютерам и метод полунатурных испытаний автомобиля (включая рулевое управление) с использованием стенда с беговыми барабанами на основе задних мостов с шинами, скорость вращения, которых управляется по результатам моделирования на компьютерах;

- метод реализации моделирования более сложных моделей в т.ч. пространственных (трудно реализуемых в режиме реального времени) на основе моделирования вначале на плоских моделях с последующим использованием полученных записей для пространственных моделей,

позволяющий проводить оценку напряженно-деформированного состояния рам, кабин, мостов и других узлов, испытывающих крутящие нагрузки;

- схема и модель работы дизеля V8 (сил сгорания топлива), ее реализация в Simulink как источника воздействия на трансмиссию, ТС, объясняющая его продольные и поперечные вибрации;

- более простой метод визуализации MR на основе видеосъемки дорожной обстановки при фиксированной скорости движения автомобиля ($v = 20$ м/с) с последующим ее воспроизведением в разработанной программе с управляемой частотой просмотра синхронно с процессом в Simulink;

- комбинированное использование пакетов: для моделирования движения, колебаний, управляемости автомобиля – Simulink, а для визуализации обстановки MR – RoadRunner, UnrealEngine и их интеграция;

- полученные результаты всесторонних исследований движения, колебаний, управляемости и экономичности колесной машин на основе предложенных моделей и подходов со смешанной реальности с симулятором;

- полученные графики прогнозирования передвижения колесных машин на различных дорогах в чрезвычайных ситуациях на основе моделировании движения в этих условиях.

3. На основе протестированных различных вариантов схемных решений, в т.ч. увеличение ОП до 32 Гб, направленных на повышение быстродействия моделирования до 2-х раз, предложена оригинальная блок-схема с задействованием испытательного оборудования с разделением вычислений по компьютерам, сетям, обеспечивающая необходимое быстродействие.

4. Путем анализа кинематики 6-DoF платформы, созданной ее модели определены необходимые усилия электроактуаторов (20280 Н) и подобраны подходящие на рынке платформы (MOOG/Parker).

5. Все это в совокупности позволяет реализовать моделирование со смешанной реальностью с симулятором в режиме реального времени, более точно оценивать параметры ТС еще на стадии проектирования, осуществлять проведения ускоренных стендовых испытаний узлов, деталей автомобиля и отладку его систем, значительно сократить время доводки КМ.

Рекомендации по практическому использованию результатов

Разработанные научные основы и методология оценки параметров автомобиля, ПО переданы БелАЗу и применяются в ООО «Мидивисана» при проектировании и производстве их машин, что позволило повысить их технический уровень и конкурентоспособность на рынке и рекомендуются к использованию на предприятиях автотракторостроения, в НИИ и ВУЗах.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. The Mercedes Benz Simulator Driving Center <https://www.mercedes-benz.com/en/mercedes-benz/innovation/the-mercedes-benz-driving-simulation-center>.
2. Mercedes Benz TecDay 2010 Driving Simulator 2 // https://www.youtube.com/watch?v=CxIo1l_YsK0 Video: Mercedes-Benz Driving Simulator Is Industry Leader.
3. Balaji Kamalakkannan Modelling and Simulation of Vehicle Kinematics and Dynamics /Halmstad University, January 13, 2017– May 29, 2017, P.76.
4. Emanuele Obialero, A. Refined, Vehicle Dynamics Model for Driving Simulator // Chalmers University of Technology/Göteborg, Sweden 2013. Master's thesis, P.120./ <https://publications.lib.chalmers.se/records/fulltext/179787/179787.pdf>
5. Michael Arnett, Modeling and Simulation of a Hybrid Electric Vehicle for the Challenge X Competition Submitted to: The Engineering Honors Committee 119 Hitchcock Hall College of Engineering The Ohio State University Columbus, OH 43210 Michael Arnett 4091 Millsboro Rd W Mansfield, OH 44903 Dr. Giorgio Rizzoni, Advisor May 20, 2005.
6. Generic and complete vehicle dynamic models for open-source platforms VADIRAJ PATIL/A thesis submitted in partial fulfillment for the Master of Science programme in Solid and Fluid Mechanics at the Mechanical, Maritime and Materials Engineering Department TU Delft Generic and complete vehicle dynamic models for open-source platforms.
7. An Electric Simulator of a Vehicle Transmission Chain Coupled to a Vehicle Dynamic Model By A. Chaibet, C. Larouci and M. Boukhniifer Published: January 1st 2010 DOI: 10.5772/7095.
8. Raffaele Di Martino, Modelling and simulation of the dynamic behaviour of the automobile / HAL Id: tel-00736040 // <https://tel.archives-ouvertes.fr/tel-00736040> Submitted on 27 Sep. 2012.
9. Andrew Hamish, Motion Cueing in Driving Simulators for Research Applications// Andrew Hamish, John Jamson, Submitted in accordance with the requirements for the degree of Doctor of Philosophy The University of Leeds Institute for Transport studies, November 2010, P.245.
10. Toshiji Kato, Requirements to Models of Automotive System Development for Future Model-Based Design // Toshiji Kato _ Kimitoshi Tsuji _ Satoshi Shimada _ Department of Electrical Engineering, Doshisha University, Kyotanabe, Kyoto, 610-0321, Japan, (e-mail: tkato@mail.doshisha.ac.jp) _ Advanced Vehicle Control System Engineering

Dev., TOYOTA MOTOR CORPORATION, 1200, Mishuku, Susono, Shizuoka, 410-1193. Japan, __ Automobile R&D Center, Honda R&D Co., Ltd. Haga, Tochigi, 321-3393 Japan.

11. Raffaele Di Martino, Modelling and simulation of the dynamic behaviour of the automobile. Automatic/ Université de Haute Alsace - Mulhouse, 2005. English. P.189.

12. G. E. Smid, Vehicle Dynamics Modelling for Multi-CPU Driving Simulation // G. E. Smid, Ka C. Cheok, Oakland University, USA Corresponding Author: G. E. Smid /Confirance Vienna 2003, P.10.

13. Georg Rill, Vehicle Dynamics, Short Course Prof. Dr. Georg Rill / Brasil, August 2007 P.190.

14. Михайлов, В.Г., Компьютерное и имитационное моделирование автомобиля/ В.Г. Михайлов, Минск : Белнаука, 2020, 316 стр.

15. BMW Group uses virtual reality engineering in its vehicle development BMW // <https://www.bmw.com/en/events/nextgen/global-collaboration.html>.

16. ГОСТ 31191.4-2006 (ИСО 2631-4 2001). Вибрация и удар. Измерение общей вибрации и оценка ее воздействия на человека / Москва, Стандартинформ, 2008.

17. Михайлов, В.Г. Моделирование системы “дорога–автомобиль–водитель”, симуляторы, смешанная реальность / монография, 390 стр. / URL:

<https://rep.bntu.by/handle/data/142369?show=full&ysclid=lx2qllkh1533830640>

18. Руктешель О.С., Определение динамических нагрузок в трансмиссии автомобиля / О.С. Руктешель / – Минск: БНТУ, 2010. – 50 с.

19. Колесникович А.Н., Моделирование движения автопоезда по заданной траектории / А.Н. Колесникович; А.Г. Выгонный /Актуальные вопросы машиноведения. 2016. Выпуск 5. ОИМ НАН Беларуси, г. Минск.

20. Инновационные разработки национальной академии наук Беларуси в области механики приводов / С.Н. Поддубко, В.Б. Альгин , Н.Н. Ишин, С.В. Шилько / Механика машин, механизмов и материалов. 2021. № 4(57).

21. Альгин, В.Б., Управление агрегатами автомобиля в параллельной системе моделирования MSC.ADAMS – MATLAB [Электронный ресурс] / В.Б. Альгин, Б.Н. Широков ; НИРУП «Белавтотракторостроение». – Режим доступа : http://www.mssoftware.ru/document/conf/Moscow_conf/conf_2005/stand/03_Belautotr_3.pdf. – Дата доступа : 26.08.2011.

22. Жилевич М.И., Анализ влияния растворимости газа на жесткость пневмогидравлической подвески карьерных самосвалов// М.И Жилевич, Заболоцкий Е.М., Заболоцкий М.М. // Вестник ГГТУ им. П.О. Сухого № 3-4 2002, с. 17–21.

23. Жилейкин М. М., Экспериментальное исследование нагрузочных характеристик двухкамерной пневмогидравлической рессоры подвески автомобильных платформ нового поколения средней и большой грузоподъемности // М. М. Жилейкин, Г. О. Котиев, Е.Б Сарач.// Наука и образование № 12, декабрь 2011.

24. Альгин, В.Б., Виртуальное моделирование как способ получения адекватных результатов поперечной и продольной устойчивости транспортных средств [Электронный ресурс] / В.Б. Альгин, С.В. Харитончик, А.Н. Колесникович ; НИРУП «Белавтотракторостроение». – Режим доступа : http://www.mscsoftware.ru/document/conf/Moscow_conf/conf_2005/stend/02_Belautotr_1.pdf. – Дата доступа : 26.08.2011.

25. Колесникович, А.Н., Виртуальные испытания транспортных средств на статическую устойчивость / А.Н. Колесникович, В.Б. Альгин, С.В. Харитончик // Повышение конкурентоспособности автотранспортных средств : сб. науч. тр. / НИРУП «Белавтотракторостроение» ; ред. кол.: М.С. Высоцкий [и др.]. – Минск, 2004. – С. 229–233.

26. Мишута, Д.В., Разработка штабных машин мобильных пунктов управления : дис. канд. техн. наук : 05.05.03. – Минск, 2013. – 191 л.

27. Мазлумян Г.С., Результаты математического моделирования движения подвижного агрегата ракетного комплекса с переменной скоростью с учетом условий движения, плавности хода и мастерства механика-водителя/ МАДИ, Электронный научный журнал Автомобиль Дорога-Инфраструктура № 2(2) декабрь 2014 С.20.

28. Davis, L., Development of a software model of a heavy vehicle suspension for research / L. Davis, J. Bunker. – Queensland : University of Technology, 2008. –37 p. – (QUT Digital Repository: <http://www.eprints.qut.edu.au>).

29 Blekhmana, I., Vertical-longitudinal dynamics of vehicle on road with unevenness / I. Blekhmana, E. Kremera// X International Conference on Structural Dynamics, EUROLYN, 2017.

30. IBRAHIM BAKIRCI The use of wheel torque modulation to mitigate road induced longitudinal vibrations / IBRAHIM BAKIRCI, NINO KATADZIC / Master's Thesis in Automotive Engineering, CHALMERS UNIVERSITY OF TECHNOLOGY, Göteborg, Sweden 2014, P. 51.

31. Лелиовский К.Я., Исследование вибраций трансмиссии при движении автомобиля по дорогам различного профиля / К.Я. Лелиовский, В.С. Макаров Труды Нижегородского государственного технического университета им. Р.Е. Алексеева № 4(101) Р. 6.

32. Выгонный, А.Г., Закон переключения передач, оптимальный по по тягово-скоростным и топливно-экономическим показателям магистрального автопоезда / А.Г. Выгонный, О.С. Руктешель, Р.Э. Шейбак // Актуальные вопросы машиностроения 2015 – Т. 4 – С.182–188.

33. Выгонный, А. Г., Сравнительный анализ многокритериальной и однокритериальной оптимизации параметров двигателя и трансмиссии магистрального автопоезда/ А. Г. Выгонный [и др.] // Механика машин, механизмов и материалов, 2014, № 4. С. 40-46.

34. Diana Yanakiev, Engine transmission modeling for Heavy-Duty Vehicle / Diana Yanakiev, Ioannis Kanellakopoulos / Universite Canifornia, May 1995, P.162.

35. A.Saulnie, Modeling and simulation of a vehicle with automatic transmission Guang Wu, Design, Modeling and Optimization of Hybridized Automated Manual Transmission for Electrified Vehicles / Guang Wu, B.Eng, Hunan University, University of Victoria China, 2017, P.168.

36 Maltseva N., Modeling the Load Mode of Transmission of a Heavy Truck in Case of Starting with a Clutch Release / 2018, <https://www.elibrary.ru/item.asp?id=38693344?>

37. N. Maltseva, V. Anchukov, A. Alyukov, Proceedings of the World Congress on Engineering and Computer Science 2018 Vol II WCECS 2018, October 23-25, 2018, San Francisco, USA.

38. Prince Lakhani, MODELING AND ANALYSIS FOR DRIVELINE JERK CONTROL Michigan Technological University, palakhan@mtu.edu 2018, P.54.

39. Aditya Modak, Modeling and Control of an Automated Manual Transmission for EcoCAR 3 Vehicle / The Ohio State University, THESIS Master's, 2017,/ P. 26.

40. A. J. D. Nanthakumar, Mathematical modeling and simulation of longitudinal road profiles / A. J. D. Nanthakumar, Karan Jariwala and Harshit Kumawat Site as: AIP Conference Proceedings 2277, 130002 (2020); <https://doi.org/10.1063/5.0025225> Published Online: 06 November 2020.

41. Ротенберг Р.В., Подвеска автомобиля/ Р.В. Ротенберг –М.: 1972.

42. Островерхов, Н.Л., Методика моделирования виброзащитных свойств сиденья водителя на аналоговых машинах / Н.Л. Островерхов, М.С. Высоцкий, В.Г. Михайлов. – Минск : Индмаш АН БССР, 1980. – 20 с.

43. Беленький, Ю.Ю. Влияние отдельных параметров на колебания автопоезда, состоящего из тяжелого тягача и полуприцепа (МАЗ) : дис. ... канд. техн. наук : 05.05.03 / Ю.Ю. Беленький. – Минск, 1967. – 171 л.

44 29. N. B. Kate N. B., Mathematical Modeling of an Automobile Damper / N. B. Kate, T. A. Jadhav // International Journal of Engineering Research (ISSN : 2319-6890), Nov 2013, Volume No 2, Issue No. 7, pp : 467-471.

45. Беленький, Ю.Ю., Совершенствование методики и механизация расчетов динамических систем автомобилей с применением ЭВМ : НИОКР / Ю.Ю. Беленький; А.М. Маринич, рук. Л.А. Янович. – Минск : Институт математики АН БССР, 1982. – 65 с.

46. Кольцов, В.И., Модель листовой рессоры / В.И. Кольцов, Ю.В. Пирковский, В.И. Ковицкий // Автомобильная промышленность. – 1970. – № 10. – С. 14–16.

47. Михайлов, В.Г., Исследования системы подрессоривания сиденья водителя грузового автомобиля : дис. ... канд. техн. наук : 05.05.03 / В.Г. Михайлов. – Минск : БПИ, 1982. – 231 л.

48. Михайлов, В.Г., Анализ моделей трения в подвесках транспортных средств / В.Г. Михайлов // Трение и износ – 2014, Т.2, № 35, с. 198–206.

49. Михайлов В.Г., Модель гидравлического амортизатора / В.Г. Михайлов // ТРУДЫ НАМИ – 2015, Вып.263. С.115-132.

50. СТО МАДИ 02066517.1–2006. Дороги автомобильные общего пользования. Диагностика. Определение продольного микропрофиля дорожной поверхности и международного показателя ровности IRI. Общие требования и порядок проведения.

51. Динамика системы «дорога–шина–автомобиль–водитель» / под ред. А.А. Хачатурова. – М. : Машиностроение, 1976. – 535 с.

52. Shaohua Li Dynamical Test and Modeling for Hydraulic Shock Absorber on Heavy Vehicle under Harmonic and Random Loadings / Li Shaohua, Lu Yongjie and Li Liyang // Research Journal of Applied Sciences, Engineering and Technology. 2012. № 4(13);, p. 1903–1910.

53. Davis, L, Development of a software model of a heavy vehicle suspension for research / L. Davis, J. Bunker. – Queensland : University of Technology, 2008. –37 p. – (QUT Digital Repository: <http://www.eprints.qut.edu.au>).

54. Pacejka Hans B., (2006). Tyre and vehicle dynamics, 3rd Edition. – Butterworth-Heinemann, Elsevier, 2012. – ISBN: 978-0-08-097016-5.

55. Михайлов, В.Г., Получение и использование единого массива продольного профиля и микропрофиля дороги для моделирования ТС // Журнал автомобильных инженеров 2018, № 2, с. 4–7.

56. Михайлов, В.Г., Метод замеров микропрофилей дорог на движущемся автомобиле или транспортном средстве // В.Г. Михайлов. Автомобильная промышленность – 2018– № 2: с.23–27.

57. Гришкевич А.И., Исследование динамики движения армейских автомобилей по дорогам с неровной поверхностью / А.И. Гришкевич/ – дис. докт. тех. наук: 05.05.03– Минск, 1973. – С. 353.

58. Merkator's projection / <https://wiki.openstreetmap.org/wiki/Mercator> / [Электронный ресурс /Electronic resource]. /– Режим доступа/Access mode: 22.12.2020.

59. Географическая система координат // http://wikichi.ru/wiki/Geographic_coordinate_system /Режим доступа/Access mode: 22.12.2020.

60. Wikipedia [Электронный ресурс] / https://en.wikipedia.org/wiki/Latitude#Length_of_a_degree_of_latitude // https://en.wikipedia.org/wiki/Longitude#Length_of_a_degree_of_longitude// Режим доступа: 24.02. 2021.

61. Глобальная навигационная спутниковая система. Система координат. Методы преобразований координат определяемых точек : ГОСТ 3245-2017.

62. Опыт использования навигационных спутниковых технологий ГЛОНАСС в горнодобывающей промышленности // <https://docplayer.ru/49298954-Opyt-ispolzovaniya-navigacionnyh-sputnikovyh-tehnologiy-ghonass-v-gornodobyvayushchey-promyshlennosti.html> . / [Электронный ресурс /Electronic resource] / – Режим доступа/Access mode: 12.12.2020.

63. Удаленный мониторинг техники / <http://tdbelaz.ru/service/service/programs-monitoring> // [Электронный ресурс /Electronic resource] /– Режим доступа/Access mode: 12.12.2020.

64. NMEA-0183. Standard For Interfacing Marine Electronic Devices / https://www.u-blox.com/sites/default/files/products/documents/u-blox8-8_ReceiverDescrProtSpec_%28UBX-13003221%29.pdf / [Электронный ресурс /Electronic resource]. /– Режим доступа/Access mode: 22.12.2020.

65. Мишута, Д., В. Выбор метода обработки данных и получения продольного профиля дороги, режима движения автомобиля с помощью GPS / Д. В. Мишута, В. Г. Михайлов, // Механика машин, механизмов и материалов. – 2022. – № 2. – С. 13–19.

66. Мишута Д.В., Использование GPS при исследовании дорог и моделировании автомобиля/колесных машин // Д.В. Мишута, В.Г. Михайлов, М.В. Сырай / “Системный анализ и прикладная информатика”, 2021, № 3, с. 25–32.

67. Ершов В.С. Исследование углов крена автомобиля при прохождении поворотов в зависимости от изменения его массы // В.С. Ершов А.А [и др.] / Электронный научный журнал МАДИ "Автомобиль. Дорога. Инфраструктура" 2020, № 4.

68. NMEA-GPS-master.zip // [Электронный ресурс /Electronic resource]. <https://github.com/craigpeacock> // NMEA-GPS//2020 26 Jun – Режим доступа/Access mode: 22.12.2020.

69. Волченко, Ю.М. Интерполяция функций в системе Mathematica и ее применение [Электронный ресурс] / <https://www.twirpx.com/file/760055> // Режим доступа: 24.02. 2021.

70. Catmull, Edwin; Rom, Raphael (1974). "A class of local interpolating splines". In Barnhill, Robert E.; Riesenfeld, Richard F. (eds.). *Computer Aided Geometric Design*. pp. 317–326. doi:10.1016/B978-0-12-079050-0.50020-5.

71. Marc Sol'e Gaset, *Software defined Radio for GPS* / Centre Tecnol'ogic de Telecomunicacions de Catalunya (CTTC) Escola T'ecnica Superior Enginyeria Superior de Telecomunicacions de Barcelona (ETSETB). Universitat Polit'ecnica de Catalunya (UPC). July 2009, 41802533.pdf. С. 220.

72. Михайлов, В.Г., Считывание, первичная обработка ускорений и углов наклона с MPU-6050 и запись бинарных данных на SD карту Arduino Due// В.Г. Михайлов /Системный анализ и прикладная информатика – 2016, № 4. С.53–60.

73. Uffelman, F., Protecting roads by reducing the dynamic wheel loads of trucks / F. Uffelman, W.D. Walte // European ADAMS user conference. – Frankfurt, Germany, 1994.

74. Михайлов, В.Г., О колебательной модели грузового автомобиля/ В. Г. Михайлов, Д.В. Мишута // Автомобильная промышленность. – 2016 – № 7 С.23–27.

75. Михайлов, В.Г., Оценка вибронагруженности штабной машины / В. Г. Михайлов Д.В. Мишута // Сборник научных трудов Военной академии Республики Беларусь – 2016. – № 1. С. 159–164.

76. Михайлов, В.Г., Исследование, аппроксимация характеристик подушек сидений и их влияние на вибронагруженность водителя ТС // Системный анализ и прикладная информатика – 2016– № 1. С.51–58.

77. Михайлов, В.Г., Оценка эффективности систем поддрессирования грузового автомобиля /В.Г. Михайлов, Д.В. Мишута /Автомобильная промышленность. – 2016–№5 – С.16–20.

78. Михайлов, В.Г., Расчетное исследование влияния параметров подвески сиденья на вибронагруженность рабочего места водителя грузового автомобиля/ В. Г. Михайлов, Д.В. Мишута // Сборник научных трудов НАМИ. – 2016, № 3 (266)– С.25–34.

79. Шишкин В.И., Динамические и эргономические исследования и оптимизация характеристик взаимодействия водителя и автомобиля: дис. ...канд. техн. наук: 05.05.03/ В.И. Шишкин – Горький, 1977–234 с.

80. Wendy Mevaluating the Impact of Whole-BodyVibration (WBV) on Fatigue and the Implications for Driver Safety / Wendy M. Troxel, Todd C. Helmus, Flavia Tsang, Carter C. Price // Research Report// Published by the RAND Corporation, Santa Monica, Calif., 2015, Available at: www.rand.org/t/rr1057 / (accessed 8 Jan. 2015).

81. Сводный отчет по экспериментальному и расчетному исследованию вибраций на рабочем месте водителя автомобилей семейства МАЗ-5432, МАЗ-6422 – Технический отчет МАЗ НИР-1.02-81. Минск, 1981–133 с.

82. Программное обеспечение ЭВМ Расчет плавности хода автопоезда на ЕС ЭВМ / Петрович А.И. [и др.] / Ин-т математики АН БССР–Минск, 1984, Вып. 49, 77 с.

83. Мишута, Д. В., Динамическая и математическая колебательная модель штабной машины с контейнером / Д. В. Мишута, В. Г. Михайлов // Актуальные вопросы машиноведения: сб. науч. тр. / Объедин. ин-т машиностроения НАН Беларуси; редкол.: А. А. Дюжев [и др.]. – 2013. – Вып. 2, С. 351–354.

84. Per Sahlholm, Piecewise Linear Road Grade Estimation /Per Sahlholm, Ather Gattami, Karl Henrik Johansson /Scania CV AB, 2ACCESS Linnaeus Centre Royal Institute of Technology, Sweden, 2011. KTH - School of Electrical Engineering SE-100 44 Stockholm, Sweden TRITA-EE 2011:007. P.16.

85. Per Sahlholm, Road Grade Estimation for Look-ahead Vehicle Control/Per Sahlholm _ Karl Henrik Johansson __ Scania CV AB, SE-151 87 Sodertalje, Sweden 2012/_ Royal Institute of Technology (KTH), SE-100 44, Stockholm. P.7.

86. Hong S Road Grade and Vehicle Parameter Estimation for Longitudinal Control Using GPS/ Hong S. Bae1, Jihan Ryu and J. Christian Gerdes. Road Grade Estimation for Look-ahead Vehicle Control Per Sahlholm _ Karl Henrik Johansson __ Scania CV AB, SE-151 87 Sodertalje, Sweden, Royal Institute of Technology (KTH), SE-100 44, Stockholm, 2012, P.7.

87. Skulic A., All Methods and devices for registration of road microprofiles / [Электронный ресурс]. / <http://scindeks-clanci.ceon.rs/data/pdf/0040-2176/2016/0040-21761605697S.pdf> – Режим доступа: 30.10.2017.

88. Шпенст В., Радиолокационные станции дистанционного зондирования Земли космического базирования //В. Шпенст /[Электронный ресурс]/ http://kit-e.ru/assets/files/pdf/2013_3_154.pdf – Режим доступа: 24.07.2017.

89. Карты высот / [Электронный ресурс] / <https://votetovid.ru/#53.593,28.982,6z,53.917281,27.729492i,trb> – Режим доступа: 24.07.2021.

90. Леонов Г.Н., Некоторые особенности моделирования динамики двигателей внутреннего сгорания / Г.Н. Леонов, А.В. Фролов, И.С. Шустов // [Электронный ресурс]. – 2012. – Режим доступа: http://elib.altstu.ru/elib/books/Files/pv2012_03_1/pdf/123leonov.pdf. – Дата доступа : 25.07.2012.

91. Чиркин С.Ю., Моделирование двигателя внутреннего сгорания как объекта управления на основе экспериментальных данных // Естественные и технические науки. –2010. №2.– с. 406–410.

92. Буй Н.Х. Моделирование судового дизеля как объекта управления на базе нечеткой логики / Буй Нгок Хай., С. В. Головкин, Ф. В. Вирепов // Вестник Астраханского государственного технического университета–2012–№ 1.

93. Михайлов, В.Г., Влияние подрессоривания дизель-генераторной установки на вибрации штабной машины / В. Г. Михайлов, Д.В. Мишута // Актуальные вопросы машиноведения: сб. науч. тр. / Объедин. ин-т машиностроения НАН Беларуси. 2015.– вып. 4.– с. 160–164.

94. Redroithu B.H., Tyre modelling for rolling resistance / B.H Redroithu Department of Applied Mechanics Division of Vehicle Engineering and Autonomous System, HALMERS UNIVERSITY OF TECHNOLOGY Göteborg, Sweden 2014 Master's thesis 2014:24.

95. Белецкий А.В., Моделирование профиля дорожного основания в задаче анализа динамики трансмиссии колесной машины (ГОУ ВПО «Липецкий государственный технический университет») [Электронный ресурс]. – 2013. – Режим доступа : <http://sdm.str-t.ru/insertfiles/5.pdf>. - Дата доступа : 22.05.2013.

96. Макаров В. С., Математическое моделирование трасс движения транспортных средств на примере дорог типа «stone-road» / В. С. Макаров, [и др.] // Известия Академии инженерных наук РФ им. акад. А. М. Прохорова. Юбилейный том, посвященный 20-летию Академии инженерных наук РФ; Под ред. Ю. В. Гуляева. – М.; Н. Новгород: НГТУ, 2011. – С. 129 – 134.

97. Мишута Д. В., О выборе режима нагружения штабной машины с кузовом-контейнером / Д. В. Мишута, В. Г. Михайлов, Утекало В.К. // Вестник военной академии Республики Беларусь – 2013. – № 4. – С.79–87.

98. Шлейхер А.А., Крылов В.А., Расчет ресурса и запасов прочности переднего моста автомобиля «УРАЛ - 5557» / А.А. Шлейхер, В.А. Крылов, [Электронный ресурс]. - 2012. - Режим доступа : <http://www.lab106.ru/projects/ural1/ur1.html>. - Дата доступа : 28.12.2012.

99. Михайлов, В.Г., Совместное моделирование движения и нагруженности автомобиля // В.Г. Михайлов, Д.В. Мишута/ Автомобильная промышленность – 2019, № 7. С.1–6.

100. Vincenzo Barone, Vibrational Comfort on Board the Vehicle: Influence of Speed Bumps and Comparison between Different Categories of Vehicle // Vincenzo Barone [and other] / Department of Civil Engineering, University of Calabria, 87036 Arcavacata di Rende, Italy / Volume 2016 |Article ID 2676021 / <https://doi.org/10.1155/2016/2676021> / , Advances in Acoustics and Vibration, vol. 2016, Article ID 2676021, 6 pages, 2016. <https://doi.org/10.1155/2016/2676021>.

101. Горбацевич М.И., Обработка конструкции и прогнозирование усталостной долговечности несущих деталей ходовой части транспортных средств : Дис. ... к.т.н., Минск: Ин-т проблем надежности АН БССР, 1984 – 176 с.

102. Михайлов, В.Г., Оценка долговечности узлов автомобиля на стадии проектирования_ / Электронный научный журнал МАДИ – 2020–№ 4 (26).

103. Wojciech Owczarek, Mirosław Rodzewicz, Investigations Into Glider Chassis Load Spectrum/ Owczarek Wojciech, Rodzewicz Mirosław / Warsaw University of Technology /January 2009/ [Электронный ресурс /Electronic resource]. https://www.researchgate.net/publication/271050396_Investigations_Into_Glider_Chassis_Load_Spectrum/– Режим доступа/Access mode: 22.05.2020.

104. Rainflow Fatigue Cycle Counting / 2012 October 31, / [Электронный ресурс /Electronic resource]. https://drive.google.com/file/d/0Bw1VlHh5X_tLTJ6d25SVkJwWEk/view/– Режим доступа/Access mode: 23.09.2020.

105. Сайт расстояний между городами / <https://flagma.by/rasstoyanie>.

106. ГОСТ 33101-2014 Дороги автомобильные общего пользования. Покрытия дорожные. Методы измерения ровности/М.:Стандартинформ, 2016.

107. D. Stewart, "A platform with six degrees of freedom", Proc. of institute for Mechanical Engineering, London, Vol. 180, pp. 371-386, 1965.

108. Shiwei Ni, Analysis of torsional vibration effect on the diesel engine block vibration // Harbin Engineering University, Harbin Heilongjiang, China Shipbuilding Power Engineering Institute Co., Ltd, Shanghai, China / https://www.mechanics-industry.org/articles/meca/full_html/2020/05/mi190333/mi190333.html#T2.

109. Neda Nickmehr, Methodology for modeling, parameter estimation, and validation of powertrain torsional vibration / Neda Nickmehr [other] // Dep. of Electrical Engineering, Linköping University, Sweden.

110. Enrico Armentani, Multibody Simulation for the Vibration Analysis of Turbocharged Diesel Engine /. Francesco Caputo, Luca Esposito, Roberto Citarella

/ Italy Peer-reviewed version available at Appl. Sci. 2018, 8, 1192; doi:10.3390/app8071192.

111. Мишута Д.В., Михайлов В.Г., Сырай М.В., Выбор метода обработки данных и получения продольного профиля дороги, режима движения автомобиля с помощью GPS (GNSS-приемника) / Д.В.Мишута, В.Г.Михайлов, М.В. Сырай /Механика машин, механизмов и материалов №2-2022. С.88-96.

112. DigiPHY is a mixed reality automotive design and communication tool that combines VR & physical support into an immersive design experience // <https://www.granstudio.com/digiphy>.

113. Björn Blissing, Driving in Virtual Reality Requirements for Automotive Research and Development / Dissertations No. 2085, Linköping Studies in Science and Technology, Division of Machine Design Department of Management and Engineering Linköping University, SE-581 83 Linköping, Sweden, 2020 // <https://www.diva-portal.org/smash/get/diva2:1459751/FULLTEXT01.pdf>.

114. Mark McGill Challenges in passenger use of mixed reality headsets in cars and other transportation //Julie Williamson, Alexander Ng, Frank Pollick, Stephen Brewster//<https://link.springer.com/article/10.1007/s10055-019-00420-x> / Published: 18 December 2019.

115. Augmented Reality (AR) in Creo // https://www.irisoft.ru/ptc_files/Datasheet-Augmented_Reality_In_Creo-en.pdf?ysclid=l4qo3kxpn926975499.

116. AR: from Creo Parametric to Vuforia view in 8 simple steps // <https://community.ptc.com/t5/Creo-Parametric-Tips/AR-from-Creo-Parametric-to-Vuforia-view-in-8-simple-steps/ta-p/820144>.

117. Михайлов В.Г., О подходах моделирования движения и колебаний автомобиля при смешанной реальности // Электронный научный журнал МАДИ “Автомобиль• дорога •инфраструктура”, 2022, № 4 декабрь. 15 с.

118. Show Video in application of Vive Home // <https://mcgrp.ru/files/viewer/296494/70?ysclid=lbahqxvd5s791512283>.

119. Unreal Engine 5.1 Documentation // <https://docs.unrealengine.com/5.1/en-US>.

120. BMW brings mixed reality to automotive design with Unreal Engine / <https://www.unrealengine.com/fr/spotlights/bmw-brings-mixed-reality-to-automotive-design-with-unreal-engine> / 24 april 2018.

121. Samuli Jääskeläinen, Senior Software Engineer, Varjo , Mixed reality driving: Real hands virtual wheel //<https://varjo.com/vr-lab/mixed-reality-driving-simulation/> 1 Feb. 2022.

122. Implement Task Parallelism in Simulink //<https://www.mathworks.com/help/simulink/ug/implement-task-parallelism-in-simulink.html> //1 Feb. 2023.

123. HTC выпускает ПО для отслеживания рук и пальцев для Vive, Vive Pro и Vive Focus 19 марта 2019 // <https://vr-j.ru/news/htc-vypuskaet-po-dlya-otslezhivaniya-ruk-i-paltsev-dlya-vive-vive-pro-i-vive-focus/?ysclid=la0y2ok7iz946913446>.
124. Use Viveport, SDK // <https://developer.vive.com/resources/viveport/store-guide/store-submission-guide/english/submitting-your-mobile-vr-content/viveport-listing> / Posted July 21, 2020.
125. Documentation English VIVEPORT SDK // <https://developer.vive.com/resources/viveport/sdk/documentation/english/viveport-sdk/>.
126. SERLatBTH / StarterGuide-VivePortVR//<https://github.com/SERLatBTH/StarterGuide-VivePortVR?ysclid=ll64oxtwj2545630529>.
127. Install and configure OpenCV-4.2.0 in Windows 10, VC++ // <https://towardsdatascience.com/install-and-configure-opencv-4-2-0-in-windows-10-vc-d132c52063a1>.
128. How to Install Visual Studio 2022 <https://www.c-sharpcorner.com/article/how-to-install-visual-studio-2022/>.
129. Шатров М.Г., Шум автомобильных двигателей внутреннего сгорания / М.Г. Шатров и [др.] //учеб. пособие / М.: МАДИ, 2014. – 68 с.
130. Яковенко А.Л., Анализ возможностей моделирования структурного шума для исследования и доводки двигателей внутреннего сгорания/ А.Л. Яковенко и [др.] //<https://aco.bmstu.press/preprints/2779> / Дата публикации: 23 мая 2022, 02:14.
131. Моделирование структурного шума внутри салона автомобиля УАЗ-3160 в связанной постановке // БГТУ, "Военмех" //<https://pandia.ru/text/77/482/10940.php?ysclid=lk0r8thgrn356708852>.
132. Spectral characteristics of (a) gasoline and (b) diesel engine //https://www.researchgate.net/figure/Spectral-characteristics-of-a-gasoline-and-b-diesel-engine-sounds-measured-in-the_fig2_324433298
133. Звук двигателя //<https://zvukogram.com/category/zvuk-dvigatelya/?ysclid=lk3khcqjqb392947851>
134. Звук двигателя, 193 композиции //<https://web.ligaudio.ru/mp3/%D0%B7%D0%B2%D1%83%D0%BA%20%D0%B4%D0%B2%D0%B8%D0%B3%D0%B0%D1%82%D0%B5%D0%BB%D1%8F/5>.
135. Имитатор звука мотора и гудка автомашины. Система имитации звука двигателя //<https://qzoreteam.ru/imitator-zvuka-motora-i-gudka-avtomashiny-sistema-imitacii-zvuka/>.
136. Система имитации звука двигателя. Радиосхемы схемы электрические принципиальные //<https://halzen.ru/multimedia/sistema-imitacii-zvuka-dvigatelya-radioshemy-shemy-elektricheskie.html>

137. Хилько А., Синтез в игровом саунд-дизайне // <https://www.youtube.com/watch?v=5ceSAprojq8&t=1916s>
138. Model Engine Noise // <https://www.mathworks.com/help/audio/ug/model-engine-noise.html>.
139. Loof, J. Modeling and control of a truck steering-system for active driver support Eindhoven: Technische Universiteit Eindhoven Published: 19/02/2018.
140. B. Shyrokau et al. Effect of steering model fidelity on subjective evaluation of truck steering feel. In: Driving Simulation Conference and Exhibition, (Sept. 2015).
141. C++ Game Programming in Unreal Engine // <https://romeroblueprints.blogspot.com/p/table-of-contents-c.html>.
142. Создание грузовика MAN Blender 2.82 Часть 1// <https://rutube.ru/video/bc1774d4565fda1880362005b8ba2773/?ysclid=m1644yso7146514434>
143. RoadRunner Design 3D scenes for automated driving simulation <https://nl.mathworks.com/products/roadrunner.html>.

ПРИЛОЖЕНИЯ

ПРИЛОЖЕНИЕ А

**Фрагмент программы, реализующий загрузку данных
макропрофиля и микропрофиля дороги в оперативную память и
подкачку нового блока дороги**

```
// File: SPD.c После
#include "simstruc.h"
// вставляем свои библиотеки и переменные и Блок своих библиотек и
переменных
#include <stdio.h>
#include <math.h>
float rd[384000];
long ch=0;
long nbl=0;
//S-function methods
//После
static void mdlOutputs(SimStruct *S, int_T tid)
{ // вставляется свой текст программ
double x, x_2, x_3, sx, y_0, y_1, y_2, ts, N, h;
double yn, kr1, Va, ma, Ndv, ndvs, Fk, Fv, Fy, jm, a2, kz, m_d, p_u, a, b, c,
P0, P1, P2, P3, u, t, t2, t3;
FILE *fp1;
long n, i;
float an;
// Переменные const real_T *x0.. real_T *y0 = (real_T
*)ssGetOutputPortRealSignal(S,0); //это переменные генерируемые самим
модулем S-Function Builder.
const real_T *x0 = (const real_T*) ssGetInputPortSignal(S,0);
const real_T *x1 = (const real_T*) ssGetInputPortSignal(S,1);
real_T *y0 = (real_T *)ssGetOutputPortRealSignal(S,0);
real_T *y1 = (real_T *)ssGetOutputPortRealSignal(S,1);
real_T *y2 = (real_T *)ssGetOutputPortRealSignal(S,2);
real_T *y3 = (real_T *)ssGetOutputPortRealSignal(S,3);
real_T *y4 = (real_T *)ssGetOutputPortRealSignal(S,4);
// Закачка микропрофиля в массив
if (ch==0)
{
```

```

sx=384000*nbl;
fp1=fopen("rd.bt","rb");
fseek(fp1, sx, SEEK_SET);
for (i=0; i<96000; i++)
{
    fread(&an, 4, 1, fp1);
    rd[i]=an;
} //Конец for
ch=1;
fclose(fp1);
}
ts=0.125;
x = x0[0]-(11892*nbl); //определение значения для точки X1 правая колея
переднее колесо
if (x<0.0)
    x=0.0;
n=u=x/ts;
t=u-n;
//t=fabs(u- n);
t2=t*t;
t3=t*t*t;
// Интерполирование значений промежуточных точек методом Catmull_Rom
P0=rd[n-1];
P1=rd[n];
P2=rd[n+1];
P3=rd[n+2];
*y0 =0.5* ((2*P1) +(-P0 + P2) * t +(2*P0 - 5*P1 + 4*P2 - P3) *t2 +(-P0 + 3*P1-
3*P2 + P3) *t3);
//-----
h=(x-11992.0)*(nbl+1.0);
if (h>=0.0)
{
    ch=0;
    //j=0;
    nbl++;
}
if (nbl==10)
ch=1;
} // Конец программы загрузки

```

```
//                                Function:                                mdlTerminate
=====
static void mdlTerminate(SimStruct *S) {}
#ifdef MATLAB_MEX_FILE  /* Is this file being compiled as a MEX-file? */
#include "simulink.c"    /* MEX-file interface mechanism */
#else
#include "cg_sfun.h"     /* Code generation registration function */
#endif
```

ПРИЛОЖЕНИЕ Б

**Исходный текст программы на C/C++ Raspberry P 3B+ для
получения данных продольных профилей дорог с GPS**

```
// 21-03-10 13:10
#include <stdio.h>
#include <string.h>
#include <unistd.h>
#include <fcntl.h>
#include <errno.h>
#include <termios.h>
#include <time.h>
#include <stdlib.h>
#include <math.h>
#define M_PI acos(-1.0)

// Program 12.03.20211
//Для компилирования ввести команду g++ gps7.c -o gps7

#define DEG2RAD(a) ((a) / (180 / M_PI))
#define RAD2DEG(a) ((a) * (180 / M_PI))
#define EARTH_RADIUS 6378137
void trdt(void);
void trgps(void);

//-----struct transmion hms (hhmmss) into long
struct spdt {
char hms[30];
char hh[4];
char mm[4];
char ss[4];
long lhh;
long lmm;
long lss;
};
struct spdt sd;
//-----
struct s5 {
```

```

char sx[30]; //Position x (lat) into mm
char sy[30]; //Position y (lon) into mm
float x; //Position x (lat) into mm
float y; //Position y (lon) into mm
float h; // hight into mm
float v; // Speed
float pr; //Passable way
long dt; // Paramer of time hhmmss
long stl; // Count of satelites
};
struct s5 sr;
//-----
double lat_d, lat_m, lat_m0,lat_mb,lt,lt2;
double lon_d, lon_m, lon_m0, lon_mb,ln, ln_m0,ln2,ln_m, lr,lr0, lrn;
long lhh, lmm, lss;
char slat[40], slon[40], slat_m[40], slon_m[40],slrn[20],hms[24], hh[4], mm[4],
ss[4], *endptr, *yk;
long i, ch, che; //??

int hex2int(char *c);
int checksum_valid(char *string);
int parse_comma_delimited_str(char *string, char **fields, int max_fields);
int debug_print_fields(int numfields, char **fields);
int OpenGPSPort(const char *devname);
int SetTime(char *date, char *time);
int hex2int(char *c);
int checksum_valid(char *string);
int parse_comma_delimited_str(char *string, char **fields, int max_fields);
int debug_print_fields(int numfields, char **fields);
int OpenGPSPort(const char *devname);
int SetTime(char *date, char *time);

int main(int argc, char **argv)
{
//
    int fd;
    char buffer[255];
    int nbytes;
    char *field[20];
    FILE *fp1;

```

```

ch=0;
che=atol(argv[1]);
fp1=fopen("gps.bt","wb");
fclose(fp1);
    if ((fd = OpenGPSPort("/dev/ttyS0")) < 0)
    {
        printf("Cannot open GPS port\r\n.");
        return 0;
    }

do {
    if ((nbytes = read(fd, &buffer, sizeof(buffer))) < 0) {
        perror("Read");
        return 1;
    } else {
        if (nbytes == 0) {
            printf("No communication from GPS module\r\n");
            sleep(1);
        } else {
            buffer[nbytes - 1] = '\0';
            //printf("[%s]\r\n",buffer);
            if (checksum_valid(buffer)) {
                if ((strcmp(buffer, "$GP", 3) == 0) |
                    (strcmp(buffer, "$GN", 3) == 0)) {

                    if (strcmp(&buffer[3], "GGA", 3) == 0) {
                        i = parse_comma_delimited_str(buffer, field, 20);
                        //debug_print_fields(i,field);
                        printf("UTC Time :%s\r\n",field[1]);
                        printf("Latitude :%s\r\n",field[2]);
                        printf("Longitude :%s\r\n",field[4]);
/*
                        sprintf(sd.hms, "%s",field[1]);
                        sprintf(sr.sx, "%s",field[4]);
                        sprintf(sr.sy, "%s",field[2]);

                        sprintf(sl原因, "%s",field[4]);
                        sprintf(slat, "%s",field[2]);

*/
                        strcpy(sd.hms, field[1]);
                        strcpy(sr.sx, field[4]);

```

```

strcpy(sr.sy, field[2]);

strcpy(slou, field[4]);
strcpy(slat, field[2]);

printf("hight :%s\r\n",field[9]);
printf("Satel :%s\r\n",field[7]);

sr.stl=atol(field[7]);
sr.h=atof(field[9]);
if (ch>=7)
{
if (strlen(sr.sx)>0)
{
trgps();
trdt();

fp1=fopen("gps.bt","ab");
fwrite(&sr,88,1,fp1);
fclose(fp1);
}
}
ch++;
printf("ch=%ld\r\n",ch);

}
if (strncmp(&buffer[3], "RMC", 3) == 0) {
i = parse_comma_delimited_str(buffer, field, 20);
//debug_print_fields(i,field);
printf("Speed :%s\r\n",field[7]);
sr.v=atof(field[7]);
//printf("UTC Time :%s\r\n",field[1]);
//printf("Date :%s\r\n",field[9]);

SetTime(field[9],field[1]);
}
}
}
}
}
}
}
}

```



```

    } while(ch<che);

    if (close(fd) < 0) {
        perror("Close");
        return 1;
    }

    return (0);
}

int debug_print_fields(int numfields, char **fields)
{
    printf("Parsed %d fields\r\n",numfields);

    for (int i = 0; i <= numfields; i++) {
        printf("Field %02d: [%s]\r\n",i,fields[i]);
    }
}

int hexchar2int(char c)
{
    if (c >= '0' && c <= '9')
        return c - '0';
    if (c >= 'A' && c <= 'F')
        return c - 'A' + 10;
    if (c >= 'a' && c <= 'f')
        return c - 'a' + 10;
    return -1;
}

int hex2int(char *c)
{
    int value;

    value = hexchar2int(c[0]);
    value = value << 4;
    value += hexchar2int(c[1]);

    return value;
}

```

```

int checksum_valid(char *string)
{
    char *checksum_str;
    int checksum;
    unsigned char calculated_checksum = 0;

    // Checksum is postcede by *
    checksum_str = strchr(string, '*');
    if (checksum_str != NULL) {
        // Remove checksum from string
        *checksum_str = '\0';
        // Calculate checksum, starting after $ (i = 1)
        for (int i = 1; i < strlen(string); i++) {
            calculated_checksum = calculated_checksum ^ string[i];
        }
        checksum = hex2int((char *)checksum_str+1);
        //printf("Checksum Str [%s], Checksum %02X, Calculated Checksum %02X\r\n", (char *)checksum_str+1, checksum, calculated_checksum);
        if (checksum == calculated_checksum) {
            //printf("Checksum OK");
            return 1;
        }
    } else {
        //printf("Error: Checksum missing or NULL NMEA message\r\n");
        return 0;
    }
    return 0;
}

int parse_comma_delimited_str(char *string, char **fields, int max_fields)
{
    int i = 0;
    fields[i++] = string;

    while ((i < max_fields) && NULL != (string = strchr(string, ','))) {
        *string = '\0';
        fields[i++] = ++string;
    }
}

```

```

        return --i;
    }

int SetTime(char *date, char *time)
{
    struct timespec ts;
    struct tm gpstime;
    time_t secs;
    char tempbuf[2];
    int ret;

    printf("GPS   UTC_Date %s, UTC_Time %s\r\n",date, time);
    // GPS date has format of ddmmyy
    // GPS time has format of hhmmss.ss

    if ((strlen(date) != 6) | (strlen(time) != 9)) {
        printf("No date or time fix. Exiting\r\n");
        return 1;
    }

    // Parse day:
    strncpy(tempbuf, (char *)date, 2);
    tempbuf[2] = '\0';
    gpstime.tm_mday = atoi(tempbuf);

    // Parse month:
    strncpy(tempbuf, (char *)date+2, 2);
    tempbuf[2] = '\0';
    gpstime.tm_mon = atoi(tempbuf) - 1;

    // Parse year:
    strncpy(tempbuf, (char *)date+4, 2);
    tempbuf[2] = '\0';
    gpstime.tm_year = atoi(tempbuf) + 100;

    // Parse hour:
    strncpy(tempbuf, (char *)time, 2);
    tempbuf[2] = '\0';
    gpstime.tm_hour = atoi(tempbuf);

```

```

// Parse minutes:
strncpy(tempbuf, (char *)time+2, 2);
tempbuf[2] = '\0';
gpstime.tm_min = atoi(tempbuf);

// Parse seconds:
strncpy(tempbuf, (char *)time+4, 2);
tempbuf[2] = '\0';
gpstime.tm_sec = atoi(tempbuf);

printf("Converted UTC_Date %02d%02d%02d, UTC_Time
%02d%02d%02d.00\r\n",gpstime.tm_mday,(gpstime.tm_mon)+1,(gpstime.tm_yea
r)%100, gpstime.tm_hour, gpstime.tm_min, gpstime.tm_sec);

ts.tv_sec = mktime(&gpstime);
// Apply GMT offset to correct for timezone
ts.tv_sec += gpstime.tm_gmtoff;

printf("Number of seconds since Epoch %ld\r\n",ts.tv_sec);

ts.tv_nsec = 0;
ret = clock_settime(CLOCK_REALTIME, &ts);
if (ret)
    perror("Set Clock");
printf("\r\n");
}

int OpenGPSPort(const char *devname)
{
    int fd;
    struct termios options;

    if ((fd = open(devname, O_RDWR | O_NOCTTY | O_NDELAY)) < 0) {
        perror("Open");
        return 1;
    }

    // Set to blocking
    fcntl(fd, F_SETFL, 0);

```

```

// Get port attributes
tcgetattr(fd, &options);

// Set input and output baud rates
cfsetispeed(&options, B9600);
cfsetospeed(&options, B9600);

// Set input modes
options.c_iflag |= ICRNL;

// Set 8 bits, no parity, 1 stop bit
options.c_cflag &= ~PARENB;
options.c_cflag &= ~CSTOPB;
options.c_cflag &= ~CSIZE;
options.c_cflag |= CS8;

options.c_lflag &= ~ECHO;
options.c_lflag |= ICANON;

// Set port attributes
tcsetattr(fd, TCSAFLUSH, &options);

return(fd);
}
//-----

// Data transformation GPS
void trgps(void)
{
char *endptr, *yk;
char gr[4], mm[10], ss[4],ms[6];
double grd,mmd, ssd, msd;

// Transmision slon to lon_m
yk=slon;
strncpy(gr, yk,3);
gr[3]='\0';
grd=atol(gr);

yk=slon+3;

```

```

strncpy(mm,yk,8);
mm[8]='\0';
mmd=(atof(mm))/60.0;
lon_d=grd+mmd;
//-----

// Transmission slat to lat_m

yk=slat;
strncpy(gr, yk,2);
gr[2]='\0';
grd=atol(gr);

yk=slat+2;
strncpy(mm,yk,8);
mm[8]='\0';
mmd=(atof(mm))/60.0;
lat_d=grd+mmd;

lon_m=111320.0*lon_d;
lat_m=111320.0*lat_d;

if (ch==7)
{
lat_mb=lat_m;
lon_mb=lon_m;
lrn=lr0=0.0; ///?
}

lat_m=lat_m-lat_mb;
lon_m=lon_m-lon_mb;
//-----
lt=lat_m-lat_m0; //Difference of values
ln=lon_m-lon_m0; //Difference of values
//-----                lt2=lt*lt;
lr=sr.v;
lrn=lrn+lr; ///?

lat_m0=lat_m;
lon_m0=lon_m;

```

```

//-----
// Write Transmion data into struct
sr.x=lon_m;
sr.y=lat_m;
sr.pr=lrn; //Passable way
//-----
}
//Function transmiion time hhmmss into long
void trdt(void)
{
char *yk;
strncpy(sd.hh,sd.hms,2);
sd.hh[2]='\0';

yk=sd.hms+2;
strncpy(sd.mm,yk,2);
sd.mm[2]='\0';
yk=sd.hms+4;
strncpy(sd.ss,yk,2);
sd.ss[2]='\0';
sd.lhh=atol(sd.hh);
sd.lhh=10000*sd.lhh;
sd.lmm=atol(sd.mm);
sd.lmm=100*sd.lmm;
sd.lss=atol(sd.ss);
sd.lhh=sr.dt=sd.lhh+sd.lmm+sd.lss;

} //-----

```


**Реализация фильтрации и интерполяции данных
продольных профилей дорог, полученных с GPS**

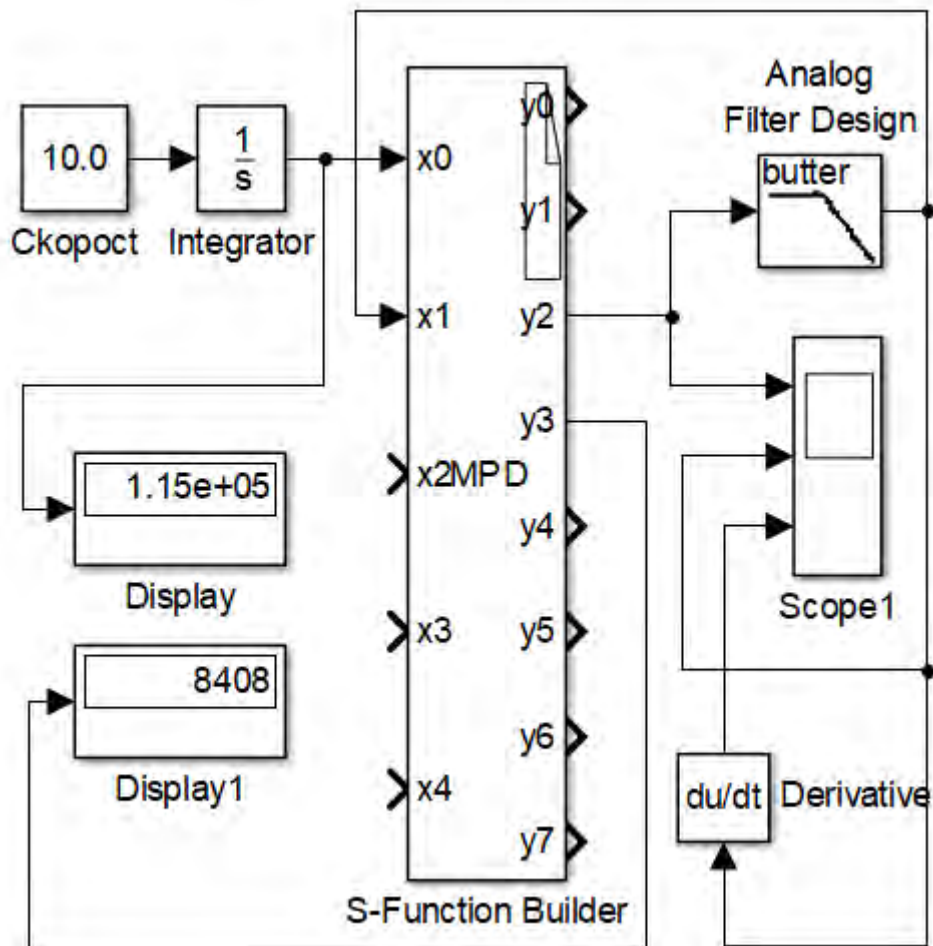


**Рисунок В.1. – Блок-схема реализация фильтрации и
интерполяции данных продольных профилей дорог полученных с GPS**

**Программа на C/C++ для реализации фильтрации и интерполяции
данных продольных профилей дорог, полученных с GPS**

```
// MPD.C
#include "simstruc.h"
#include "simstruc.h"
#include <stdio.h>
#include <math.h>
```

```

extern void MPD_Outputs_wrapper(const real_T *x0,
                                const real_T *x1,
                                const real_T *x2,
                                const real_T *x3,
                                const real_T *x4,
                                real_T *y0,
                                real_T *y1,
                                real_T *y2,
                                real_T *y3,
                                real_T *y4,
                                real_T *y5,
                                real_T *y6,
                                real_T *y7);

//-----

long ch=0;
long chf=0;
unsigned long j;
long nbl=0;
double fx;
float rd[25000];

FILE *fp1,*fp2;
static void mdlOutputs(SimStruct *S, int_T tid)
{
    const real_T *x0 = (const real_T*) ssGetInputPortSignal(S,0);
    const real_T *x1 = (const real_T*) ssGetInputPortSignal(S,1);
    const real_T *x2 = (const real_T*) ssGetInputPortSignal(S,2);
    const real_T *x3 = (const real_T*) ssGetInputPortSignal(S,3);
    const real_T *x4 = (const real_T*) ssGetInputPortSignal(S,4);
    real_T *y0 = (real_T *)ssGetOutputPortRealSignal(S,0);
    real_T *y1 = (real_T *)ssGetOutputPortRealSignal(S,1);
    real_T *y2 = (real_T *)ssGetOutputPortRealSignal(S,2);
    real_T *y3 = (real_T *)ssGetOutputPortRealSignal(S,3);
    real_T *y4 = (real_T *)ssGetOutputPortRealSignal(S,4);
    real_T *y5 = (real_T *)ssGetOutputPortRealSignal(S,5);
    real_T *y6 = (real_T *)ssGetOutputPortRealSignal(S,6);
    real_T *y7 = (real_T *)ssGetOutputPortRealSignal(S,7);
    double x, x_1, u, xsm, sx, y_0, y2_, ts, N, h,t, t2,t3,x_70, P0,P1,P2,P3;

```

```

float mp,ygl1, hr, dh;
long n,i, p;
j=0;

//
if (ch==0)
{
fp1=fopen("h.bt","rb");
//fseek(fp1, 480, SEEK_SET);
for (i=0; i<5050; i++)
{
fread(&dh, 4, 1, fp1);
rd[i]=dh-190.5;
} //for
ch=1;
fclose(fp1);
} // if (ch==0)
//-----

ts=20.0; //f=130 Hz
x = x0[0];
if (x<0.0)
x=1.0;
x_1 = x1[0];

//      определяю значение для точки X1
//mp= x1[0];
n=u=x/ts;
t=u-n;
//t=fabs(u- n);
t2=t*t;
t3=t*t*t;
// sx =20* x1[0];
if (n>1)
{
P0=rd[n-1];
P1=rd[n];
P2=rd[n+1];
P3=rd[n+2];

```

```

mp= 0.5 *((2*P1) +(-P0 + P2) * t +(2*P0 - 5*P1 + 4*P2 - P3) *t2 +(-P0 + 3*P1-
3*P2 + P3) * t3);
*y2=mp;
hr=x1[0];

```

```

//Определение задержки

```

```

if (chf==0&& x>1000.0&&(x_1>0.2||x_1<-0.2))

```

```

{

```

```

fx=x;

```

```

*y3=fx;

```

```

chf=1;

```

```

}

```

```

//Задержка

```

```

if (x>8400)

```

```

{

```

```

//-----

```

```

fp2=fopen("RD.bt","ab");

```

```

    fwrite(&mp, 4, 1, fp2);

```

```

    fclose(fp2);

```

```

}

```

```

} //if (n>1)

```

```

//-----

```

```

//  MPD_Outputs_wrapper(x0, x1, x2, x3, x4, y0, y1, y2, y3, y4, y5, y6, y7);

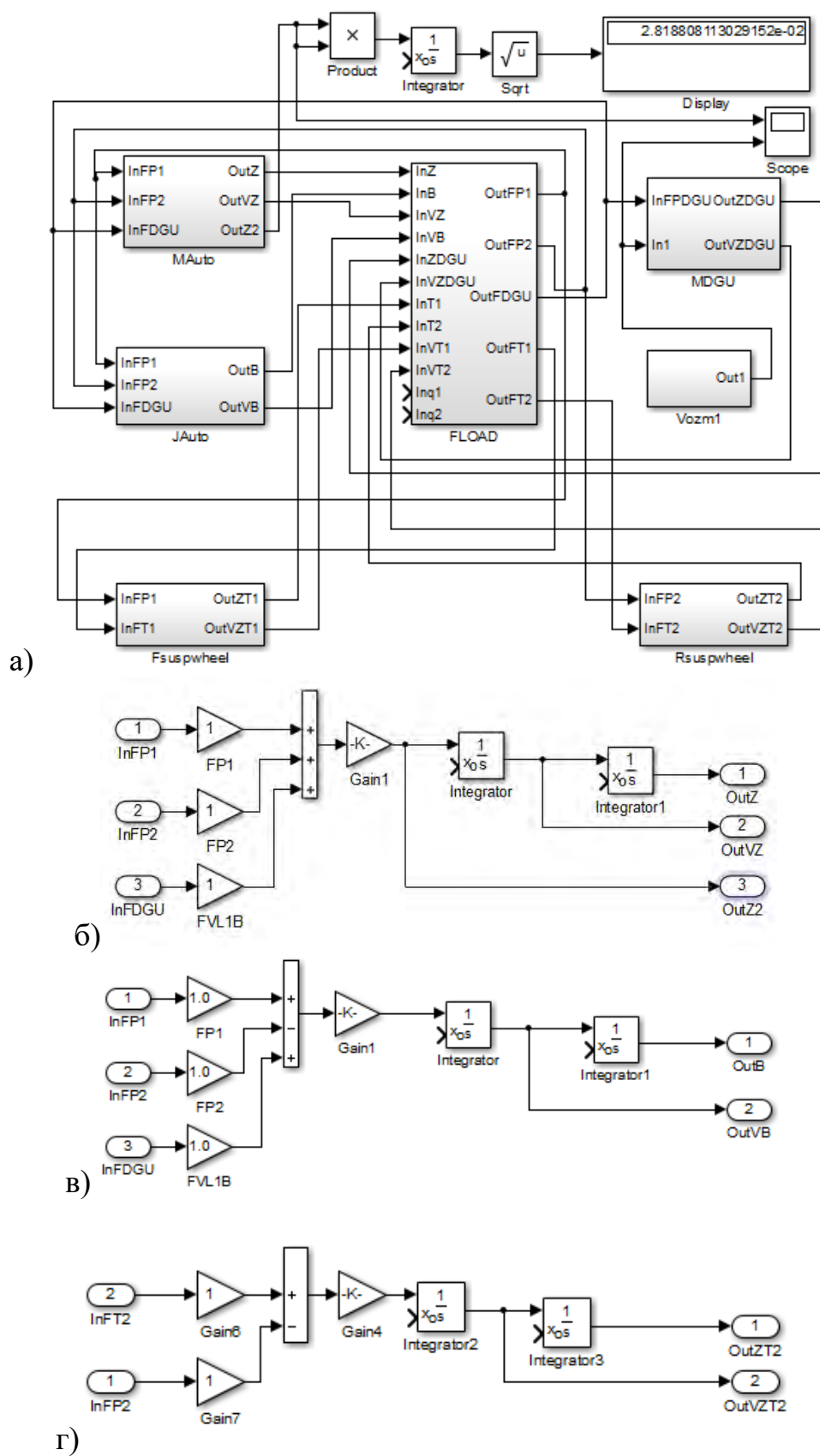
```

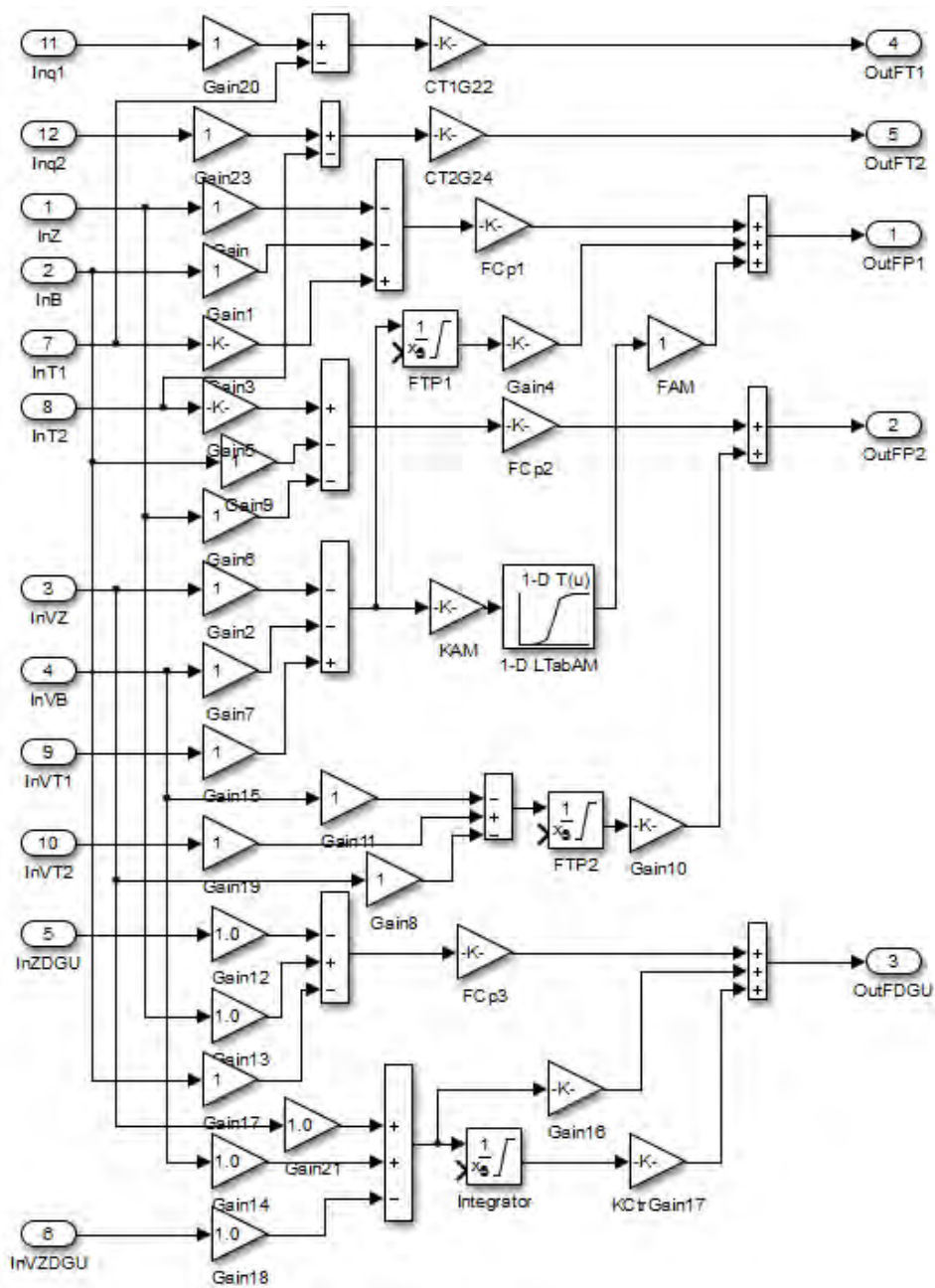
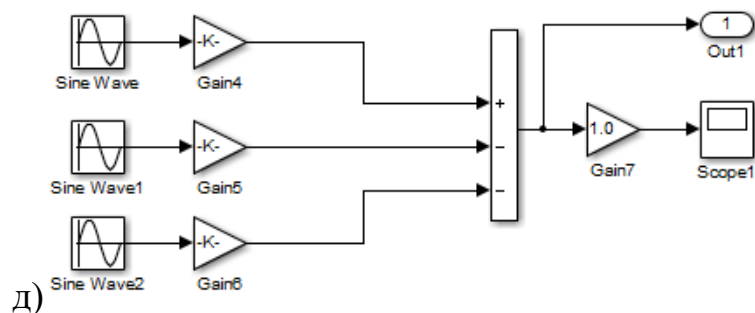
```

}

```

Блок-схема реализации модели ДГУ на автомобиле





а – Общая блок-схема подсистем; б – блок-схема передней (задней) подвески; в – блок-схема ДГУ; г – блок-схема неподрессоренных масс; д – блок-схема реализации возмущений ДГУ, е – блок-схема формирования усилий в подвеске

Рисунок Г.1 – Блок-схема реализации данной модели при компьютерном моделировании

ПРИЛОЖЕНИЕ Д

Установка программного обеспечения на Raspberry Pi 3

Д.1 Установка ОС Raspbian с помощью NOOBS

Установка ОС Raspbian с помощью NOOBS подробно описана в интернете.

Д.2 Установка даты и времени, настройка сети.

Для установки даты и времени введите

sudo date -d "Sat Feb 07 19:26:30 MSK 2019"

Примечание. В Raspberry Pi 3 отсутствуют встроенные часы реального времени и при выключении компьютера время сбрасывается. Дата и время поддерживается только через интернет или требуют использования модуля DS3231/DS1307 с памятью, питаемой от батарейки, управляемой по шине i2c. Для синхронизации времени на АЦП_ЦАП (bcm2835) необходимо обязательное использование модуля DS3231/DS1307 или сети интернета, откуда берется время.

Д.3 Настройка сети.

Д.3.1 Для организации сети необходимо обязательно прописать статический IP-адрес для Raspberry. Для этого вначале узнаем IP-адрес маршрутизатора в своей локальной сети, выполнив консольную команду на стационарном компьютере:

netstat -r -n

Обычно он выглядит как 192.168.1.1 или 192.168.xxx.1.

Узнав IP-адрес, отредактируем файл конфигурации /etc/dhcpd.conf на Raspberry Pi:

sudo nano /etc/dhcpd.conf

Допишем в конце строки после

Example static IP configuration:

interface eth0

static ip_address=192.168.1.4

static routers=192.168.1.1

Необходимо установить на Raspberry IP-адрес отличный от PC (например 192.168.1.4).

Д.3.2 Как показало тестирование наиболее оптимальным решением для организации обмена информацией является использование файлового сервера Samba. Для его установки вначале необходимо обновить систему:

sudo apt-get update

sudo apt-get dist-upgrade

Устанавливаем Samba командой: **sudo apt-get install samba samba-common-bin.**

После установки Samba необходимо создать папку **/share** в корневой директории на MicroSD диске Raspberry Pi с правами доступа

sudo mkdir -m 1777 /share

Затем Samba нужно настроить. Для этого открываем файл конфигурации:

sudo leafpad /etc/samba/smb.conf

и добавляем в конец файла следующие настройки:

[share]

Comment = Pi shared folder

Path = /share

Browseable = yes

Writeable = yes

only guest = no

create mask = 0777

directory mask = 0777

Public = yes

Перезагружаем Samba, чтобы изменения вступили в силу:

sudo /etc/init.d/samba restart

Raspberry Pi 3 появится в локальной сети в проводнике Windows 7, 8, 10 и он станет доступным пользователям сети.

Далее создаем сетевой диск Z в PC (Z://RASPBERRY/share).

Д.4 Установка компилятора C/C++ GCC 8.2.0

Д.4.1 Для разработки и компиляции программы управления, считывания датчиков органов управления и визуализации на основе OpenCV необходим GCC 8.2.0 или более новая версии 10, 12.

Использование языка Python, входящего по умолчанию в NOOBS, являющегося интерпретатором нецелесообразно из-за его низкого быстродействия. Процесс загрузки и установки компилятора C/C++ GCC 8.2.0 аналогичен его более ранним версиям.

Д.5 Установка OpenCV 4 на Raspberry Pi

Д.5.1 Установка OpenCV 4 зависимостей на Raspberry Pi

Вначале обновим систему

```
sudo apt-get update
```

```
sudo apt-get -y dist-upgrade
```

Корректируем командой `sudo nano /boot/config.txt` значение

```
gpu_mem= 128 и сохраняем файл.
```

Д.5.2 Увеличиваем SWAP на Raspberry Pi

Прежде начать компилировать необходимо увеличить swap пространство. Для этого откроем `/etc/dphys-swapfile` file командой:

```
$ sudo nano /etc/dphys-swapfile
```

...и отредактируем `CONF_SWAPSIZE` переменную с 100 до 2048. И сохраним файл.

Далее перегрузим swap service:

```
$ sudo /etc/init.d/dphys-swapfile stop
```

```
$ sudo /etc/init.d/dphys-swapfile start
```

Далее устанавливаем инструменты разработки, включающие CMake:

```
$ sudo apt-get install build-essential cmake unzip pkg-config
```

Д.5.3 Следующим шагом устанавливаем **image** and **video** библиотеки, которые критичны для работы

```
sudo apt-get install build-essential cmake pkg-config
```

```
sudo apt-get install libjpeg-dev libtiff-dev libjasper-dev libpng12-dev
```

```
sudo apt-get install libavcodec-dev libavformat-dev libswscale-dev
```

```
libv4l-dev
```

```
sudo apt-get install libxvidcore-dev libx264-dev
```

```
sudo apt-get install libgtk3.0-dev libgtk-3-dev
```

```
sudo apt-get install libatlas-base-dev gfortran
```

Далее устанавливаем **libcanberra** – это позволяет уменьшить реакцию GTK сообщений:

```
$ sudo apt-get install libcanberra-gtk*
```

Д.5.4 Выгружаем и распаковываем unzip OpenCV 4.0.1 `opencv_contrib-4.01.zip`

```
wget -O opencv.zip https://github.com/opencv/opencv/archive/4.0.1.zip
```

```
wget -O opencv_contrib.zip
```

```
https://github.com/opencv/opencv_contrib/archive/4.0.1.zip
```

```
unzip opencv.zip
```

```
unzip opencv_contrib.zip
```

Pi

Д.5.5 Создание CMake и компилирование OpenCV 4 для Raspberry

Вначале создадим каталог build, выполнив, команды

```
$ cd ~/opencv-4.0.1
$ mkdir build
$ cd build
```

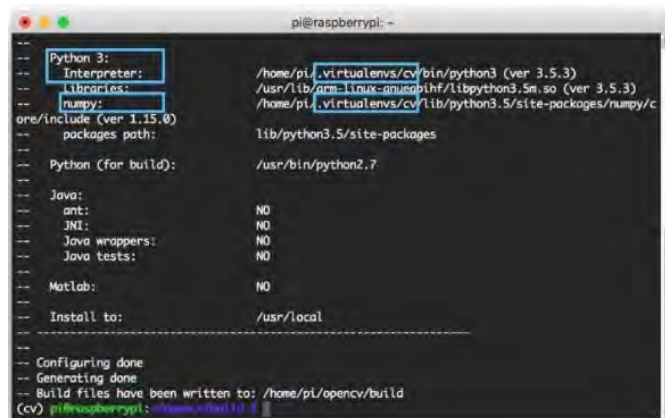
Д.5.6 Запуск CMake для OpenCV 4

Формируем в терминале CMake, чтобы конфигурировать OpenCV 4:

```
$ cmake -D_CMAKE_BUILD_TYPE=RELEASE \
-D OPENCV_GENERATE_PKGCONFIG=YES \
-D_CMAKE_INSTALL_PREFIX=/usr/local \
-D ENABLE_NEON=ON \
-D ENABLE_VFPV3=ON \
-D BUILD_TESTS=OFF \
-D INSTALL_PERF=OFF \
-D WITH_LIBV4L=ON \
-D BUILD_opencv_python2=OFF \
-D BUILD_opencv_python3=OFF \
-D BUILD_opencv_ts=OFF \
-D BUILD_PERF_TESTS=OFF \
-D OPENCV_EXTRA_MODULES_PATH=~/opencv_contrib-
4.0.0/modules \
-D INSTALL_C_EXAMPLES=ON \
-DBUILD_DOCS=OFF ..
```

Примечание. -D OPENCV_GENERATE_PKGCONFIG=YES \ обеспечивает автоматическое формирование важного файла конфигурации opencv.pc.

После чего запускаем cmake. В результате через 8–10 мин должно появиться следующее сообщение



```
pi@raspberrypi: ~
-- Python 3:
-- Interpreter:      /home/pi/.virtualenvs/cv/bin/python3 (ver 3.5.3)
-- Libraries:       /usr/lib/arm-linux-gnueabihf/libpython3.5m.so (ver 3.5.3)
-- numpy:          /home/pi/.virtualenvs/cv/lib/python3.5/site-packages/numpy/c
one/include (ver 1.15.0)
-- packages path:   lib/python3.5/site-packages
-- Python (for build): /usr/bin/python2.7
--
-- Java:
-- ant:            NO
-- JNI:            NO
-- Java wrappers:  NO
-- Java tests:     NO
--
-- Matlab:        NO
--
-- Install to:      /usr/local
--
-- Configuring done
-- Generating done
-- Build files have been written to: /home/pi/opencv/build
(cv) pi@raspberrypi: ~/opencv/build$
```

Д.6 Компилирование OpenCV 4

Для этого запустим команду

\$ make -j4

Ее выполнение займет 45–50 мин вместо 2 часов для Python.

Признаком завершения является достижение 100% и появление такого сообщения

```

pi@raspberrypi:~$ make -j4
[100%] Building CXX object modules/python2/MakeFiles/opencv_python2.dir/__/src2/cv2.cpp.o
[100%] Building CXX object modules/ximgproc/MakeFiles/opencv_perf_ximgproc.dir/perf/perf_rollin
g_guidance_filter.cpp.o
[100%] Building CXX object modules/optflow/MakeFiles/opencv_perf_optflow.dir/perf/perf_variatio
nal_refinement.cpp.o
Scanning dependencies of target opencv_python3
[100%] Building CXX object modules/python3/MakeFiles/opencv_python3.dir/__/src2/cv2.cpp.o
[100%] Building CXX object modules/ximgproc/MakeFiles/opencv_perf_ximgproc.dir/perf/perf_weight
ed_median_filter.cpp.o
[100%] Linking CXX executable ../../bin/opencv_perf_optflow
[100%] Built target opencv_perf_optflow
[100%] Building CXX object modules/ximgproc/MakeFiles/opencv_perf_ximgproc.dir/perf/perf_joint_
bilateral_filter.cpp.o
[100%] Linking CXX executable ../../bin/opencv_perf_ximgproc
[100%] Built target opencv_perf_ximgproc
[100%] Linking CXX shared module ../../lib/cv2.so
[100%] Built target opencv_python2
[100%] Linking CXX shared module ../../lib/python3/cv2.cpython-35m-arm-linux-gnueabi.so
[100%] Built target opencv_python3
(cv) pi@raspberrypi:~$ sudo make install
[ 0%] Built target gen-pkgconfig
[ 80%] Built target libwebp
[ 100%] Built target libjasper
[ 150%] Built target libImf
[ 200%] Built target libprotobuf
  
```

Далее выполняем команды

\$ sudo make install

\$ sudo ldconfig

Возвращаемся в /etc/dphys-swapfile file и изменяем его:

Reset CONF_SWAPSIZE to 100MB

Restart the swap service

Д.6 Выполняем настройку OpenCV

Д.6.1 Вносим в файл sudo nano /etc/ld.so.conf.d/opencv.conf строку
/usr/local/lib

Нажимаем Control + X сохранить и выполняем команду

sudo ldconfig

В файле sudo nano /etc/bash.bashrc необходимо внести команды

PKG_CONFIG

PATH=\$PKG_CONFIG_PATH:/usr/local/lib/pkgconfig

export PKG_CONFIG_PATH

и сохранить файл.

Проверяем наличие файла /usr/local/lib/pkgconfig/opencv.pc. Если он имеет расширение .pc4, переименовываем его в opencv.pc. При его отсутствии создаем его командой sudo nano /usr/lib/pkgconfig/opencv.pc со следующим содержанием

Package Information for pkg-config

```

prefix=/usr/local
exec_prefix=${prefix}
libdir=${exec_prefix}/lib
includedir_old=${prefix}/include/opencv4/opencv
includedir_new=${prefix}/include/opencv4
Name: OpenCV
Description: Open Source Computer Vision Library
Version: 4.0.1
Libs: -L${exec_prefix}/lib -lopencv_dnn -lopencv_gapi -lopencv_ml -
lopencv_objdetect -lopencv_photo -lopencv_stitching -lopencv_video -
lopencv_calib3d -lopencv_features2d -lopencv_highgui -lopencv_flann -
lopencv_videoio -lopencv_imgcodecs -lopencv_imgproc -lopencv_core
Libs.private: -ldl -lm -lpthread -lrt
Cflags: -I${includedir_old} -I${includedir_new}

```

Д.7 Перегружаем Raspberry, все должно работать. Для проверки OpenCV создаем простую программу вывода изображения рисунка `shl.cpp`.

```

#include <opencv2/opencv.hpp>
#include <iostream>
int main() {
    std::cout << "Hello, OpenCV version " << CV_VERSION <<
    '\n';

    cv::Mat input = cv::imread("lake.jpg", 1);
    if(!input.data) {
        std::cout << "Error! Unable to find the image file!\n";
        return -1;
    }
    cv::Mat gray;
    cvtColor(input, gray, cv::COLOR_BGR2GRAY);
    cv::imwrite("lake_gray.jpg", gray);
}

```

Вместо `lake.jpg` можно использовать другой рисунок, заменив имя. Создаем в `/home/pi` каталог **prg** командой `mkdir prg`. В него помещаем программу и файл рисунка. Переходим в этот каталог `cd prg`, набираем следующие команды для компиляции

```

g++ $(pkg-config --cflags opencv) shl.cpp $(pkg-config --libs opencv) -o
shl

```

Компилируем, затем запускаем программу `./shl`.

Д.9 Установка библиотеки `bcm2835`

Для установки библиотеки bcm2835 (для платы АЦП_ЦАП) необходимо выполнить следующие команды

```
wget https://www.airspayce.com/mikem/bcm2835/bcm2835-1.55.tar.gz
tar xvfz bcm2835-1.55.tar.gz
cd bcm2835-1.55
./configure
make check
sudo make install
```

Для проверки используем /home/pi/bcm2835-1.55/spi.c (gcc -o spi -l rt -spi.c -l bcm2835).

Примечание. Не использовать версию **bcm2835-1.6/1.9.tar.gz**, которая не совместима с OpenCV 4.

ПРИЛОЖЕНИЕ Е

Программа визуализации дорожной обстановки в Raspberry Pi 3

```

#include "opencv2/opencv.hpp"
#include <iostream>
#include <stdio.h>
#include <stdlib.h>
using namespace std;
using namespace cv;
int frshv; //параметр, определяющий частоту кадров просмотра 1000
мс/f=33

int main(){
struct pd {
float v; //скорость автомобиля
long lr; //пройденный путь
};
struct pd prd;
int i;
FILE *fp;
long chr,chl, chk,sm,stp;
chr=chl=chk=sm=0;
stp=1; //параметр позволяющий выводить video
// Создание VideoCapture object и открытие файла видеозаписи дороги
road.mp4
VideoCapture cap("road.3gp");
frshv=33;

// Проверка открытия файла, если успешно
// If(!cap.isOpened()){
// cout << "Error opening video stream or file" << endl; //в противном
выдается сообщение
// return -1;
// }
chk=1;
Mat frame;
cap >> frame;
imshow( "Frame", frame );

while(1)
{
//считывание пройденного пути автомобилем с периодичностью
chl==10000 (0,2 с)

```



```

if (chl==0)
{
    fp = fopen("pda.bt", "rb");
    fread(&prd, 8,1, fp);
    fclose(fp);
    chr=1.6346*prd.lr;
}
chl++;
if (chl==10000)
chl=0;
//-----
if ((chr-chk)<=2)
{
    sm=chr-chk;
    for (i=0;i<sm;i++)
    {
        cap.read(frame);
        cap >> frame;
        imshow( "Frame", frame );
    }
    chk=chk+sm;
    stp=1;
}

//-----
if (chr<(chk+2))
{
    sm=chk-chr;
    for (i=0;i<sm;i++)
    chk--;
    stp=0;
}
//-----
if (chr-chk<2||chk-chr<2)
stp=1;

//-----

if (stp==1)
{
    // Capture frame-by-frame
    cap >> frame;
    // Если frame переполнено, выполняется прерывание
    if (frame.empty())

```

```

        break;
        // Осуществление вывода видео на экран
        imshow( "Frame", frame );
    }
//-----
// Для выхода из просмотра нажмите кнопку ESC c==27
char c=(char)waitKey(frshv);
if (c==27)
    break;
    chk++;

} //-----End While

// Когда все выполнено реализуется video capture object
cap.release();

// Закрываются все frames
destroyAllWindows();

return 0;
}

```

Для компиляции программы введите

```
g++ $(pkg-config --cflags opencv) shs.cpp $(pkg-config --libs opencv) -o
shs
```

Запуск программы командой ./shs.

ПРИЛОЖЕНИЕ Ж

Схема и программа считывания данных с органов управления: педалей, КПП, рулевого управления на основе АЦП ADS1115 и визуализации на Raspberry Pi 3

Схема подключения АЦП ADS1115 к Raspberry Pi 3 приведена на рисунке Ж.1.



Рисунок Ж.1 – Электрическая схема управления стендом

Исходный текст программы на C++ считывания данных с органов управления: педалей, КПП, рулевого управления на основе АЦП ADS1115 и визуализации на Raspberry Pi 3 приведен ниже.

```
#include <stdio.h>
#include <stdlib.h>
#include "opencv2/opencv.hpp"
#include <iostream>
#include <sys/types.h> // open
#include <sys/stat.h> // open
#include <fcntl.h> // open
#include <unistd.h> // read/write usleep
#include <inttypes.h> // uint8_t, etc
#include <linux/i2c-dev.h> // I2C bus definitions
#include <sys/ioctl.h>
#include <math.h>
```

```
using namespace std;
using namespace cv;
```

```

int fd,i;
// defaults to 0x48!
int ads_address = 0x48;
int16_t val;

uint8_t writeBuf[3];
uint8_t readBuf[2];

float fval;

const float VPS = 4.096 / 32768.0; //volts per step
//-----
struct sn {
    float pg;
    float nkp;
    float brake;
    float steering;
};
struct sn snr;
//-----
struct pd {
    float v;    // скорость автомобиля
    long lr;    // пройденный путь
};
struct pd prd;

//nsigned long time1, time2;
int frshv; // частота кадров видео 1000 /f
void get_ads(void);
long t,chr;
//-----
int main(){
//void get_ads(void);
FILE *fp;
    long chl, chk,sm,stp, br;

    chr=chl=chk=sm=t=br=0;
    stp=1;
    // make VideoCapture object and open file show road road.mp4
    VideoCapture cap("road.3gp");

```

```

    frshv=33; // параметр, определяющий частоту просмотра кадров 1000 мс/ф
// Проверка открытия файла успешно
// If (!cap.isOpened()){
// cout << "Error opening video stream or file" << endl; //в противном выдается
сообщение
// return -1;
// }
    chk=1;
    Mat frame;
    cap >> frame;
//-----
// open device on /dev/i2c-1
// the default on Raspberry Pi B
if ((fd = open("/dev/i2c-1", O_RDWR)) < 0) {
    printf("Error: Couldn't open device! %d\n", fd);
    exit (1);
}

// connect to ads1115 as i2c slave
if (ioctl(fd, I2C_SLAVE, ads_address) < 0) {
    printf("Error: Couldn't find device on address!\n");
    exit (1);
}
//-----

while (br!=1)
{
    if (t<27)
    {

//-----

if ((chr-chk)<=2)
{
sm=chr-chk;
for (i=0;i<sm;i++)
{
cap.read(frame);
    cap >> frame;
    imshow( "Frame", frame );

```

```

}

chk=chk+sm;
stp=1;
}

//-----
if (chr<(chk+2))
{
sm=chk-chr;
for (i=0;i<sm;i++)
chk--;
stp=0;
}
//-----
if (chr-chk<2||chk-chr<2)
stp=1;

//-----

if (stp==1)
{
// Capture frame-by-frame
cap >> frame;
// Если frame переполнено выполняется прерывание
if (frame.empty())
break;
// Осуществление вывода видео на экран
imshow( "Frame", frame );
}
//-----
chk++;

} //End (if(t<27)

//-----
/* if (t>27)
get_ads();
t++;
if (t==32)

```

```

t=0;
*/
/* if (t==5)
    get_ads();
if (t==10)
    get_ads();
if (t==15)
    get_ads();
if (t==20)
    get_ads();
if (t==25)
    get_ads();
*/
if (t==30)
{
    get_ads();
t=0;

}
t++;

// Для выхода из просмотра нажмите кнопку ESC c==27
char c=(char)waitKey(frshv);
if (c==27)
    br=1;
//    break;
} //-----End While 1
close(fd);

// Когда все сделано реализуем video capture object
cap.release();

// Закрываем все frames
destroyAllWindows();

return 0;
}

//-----

```



```

void get_ads(void) {
FILE *fp1, *fp2;

for (i=0;i<4; i++)
{
writeBuf[0] = 1; // config register is 1 4.096V
if (i==0)
writeBuf[1] = 0b11000011; // ANC0 0b11010011; // bit 15-8 0xD3
if (i==1)
writeBuf[1] = 0b11010011; // ANC1 bit 15-8 0xD3
if (i==2)
writeBuf[1] = 0b11100011; // ANC2
if (i==3)
writeBuf[1] = 0b11110011; // ANC3
// set config register and start conversion
// ANC1 and GND, 4.096v, 128s/s
// bit 15 flag bit for single shot
// Bits 14-12 input selection:
// 100 ANC0 0b11000011; 101 ANC1 0b11010011; 110 ANC2 0b11100011; 111
ANC3 0b11110011
// Bits 11-9 Amp gain. Default to 010 here 001 P19
// Bit 8 Operational mode of the ADS1115.
// 0 : Continuous conversion mode
// 1 : Power-down single-shot mode (default)

writeBuf[2] = 0b10000101;//0x00A0;
//0b10000101; // bits 7-0 0x85
// Bits 7-5 data rate default to 100 for 128SPS
// Bits 4-0 comparator functions see spec sheet.

// begin conversion
if (write(fd, writeBuf, 3) != 3) {
perror("Write to register 1");
exit(-1);
}
usleep(20);
// wait for conversion complete
// checking bit 15
do {
if (read(fd, writeBuf, 2) != 2) {

```

```

    perror("Read conversion");
    exit(-1);
}
}
while ((writeBuf[0] & 0x80) == 0);

// read conversion register
// write register pointer first
readBuf[0] = 0; // conversion register is 0
if (write(fd, readBuf, 1) != 1) {
    perror("Write register select");
    exit(-1);
}

// read 2 bytes
if (read(fd, readBuf, 2) != 2) {
    perror("Read conversion");
    exit(-1);
}

// convert display results
val = readBuf[0] << 8 | readBuf[1];

if (val < 0) val = 0;

fval = val * VPS; // convert to voltage
switch (i) {
case 0: {snr.pg=900.0*fval+900.0;

fp1 = fopen("pca.bt", "wb");
fwrite(&snr,16,1, fp1);
fclose(fp1);

fp2 = fopen("pda.bt", "rb");
fread(&prd, 8,1, fp2);
fclose(fp2);

chr=1.6246*prd.lr;
fp1=fopen("lsd.txt","at");

```

```

fprintf(fp1,"t=%ld lr=%ld ch0=%.03f ch1=%.03f ch2=%.03f ch3=%.03f\n", t, chr,
snr.pg,snr.nkp,snr.brake,snr.steering);
fclose(fp1);
}
break;
case 1: snr.nkp=fval; break;
case 2: snr.brake=fval; break;
case 3: snr.steering=fval; break;
// default; break;
}
}

```

Оценка применения CAN BUS на Raspberry Pi 3

Для реализации CAN BUS на Raspberry Pi 3 задействован микроконтроллер MCP2551/MCP2515. На его плате выполнен разрыв цепи питания +5V согласно рисунка Ж.2, чтобы обеспечить раздельное питание +3,3 В и +5В, подпаяв отдельный провод. Подключение микроконтроллера к Raspberry Pi 3 проведено согласно рисунка Ж.3.

Далее скачены утилиты Linux-CAN / SocketCAN user space applications с сайта <https://github.com/linux-can/can-utils>, включая их библиотеки и откомпилированы программы. Из-за большого объема они здесь не приводятся. Выяснено, что CAN BUS значительно загружает Raspberry Pi 3, из-за чего сильно тормозится (до 50%) визуализация дороги на нем. Это делает нецелесообразным использование CAN BUS на Raspberry Pi 3 при совместном считывании данных с АЦП и визуализации дороги. Более целесообразным все же является использование сети Samba для обмена информации с основным компьютером.

Рисунок Ж.2 – Схема микроконтроллера МСР2551/2515

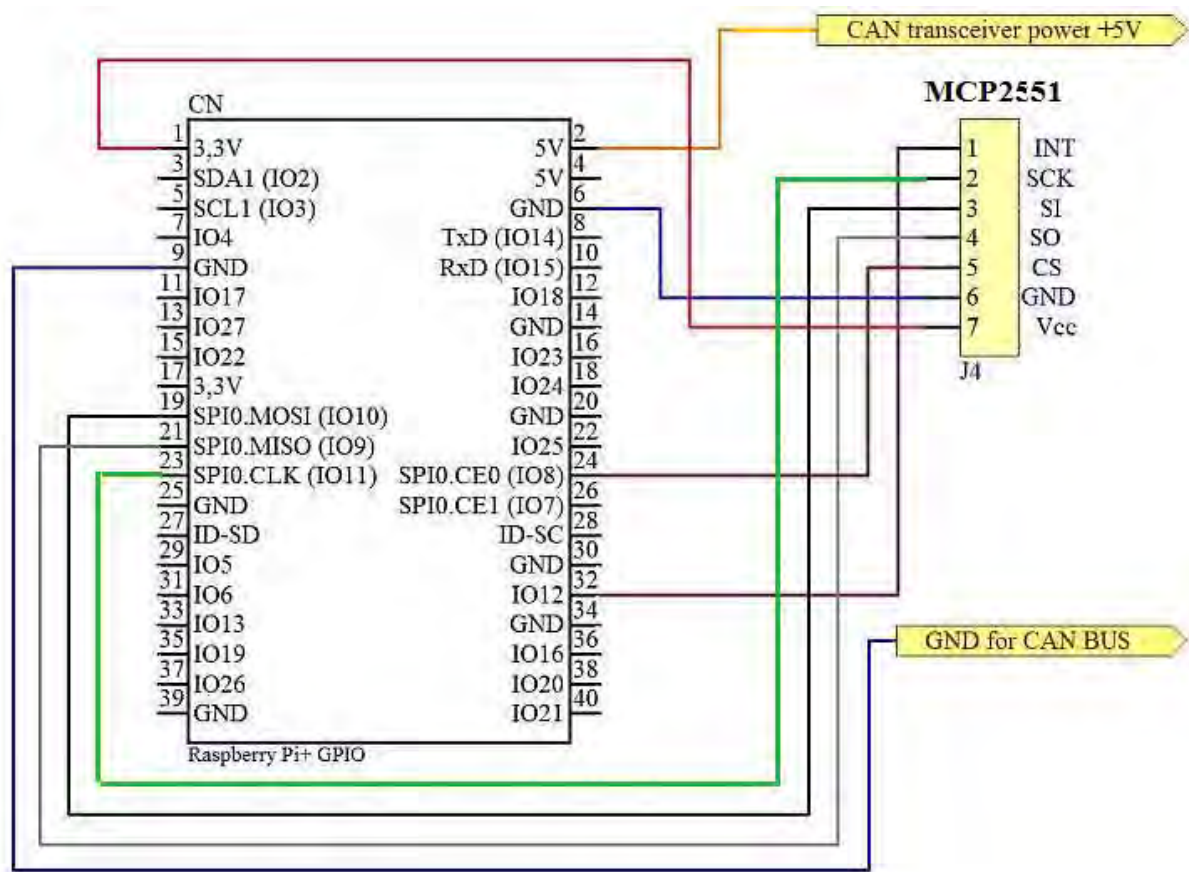


Рисунок Ж.3 – Схема подключения MCP2551/MCP2515 к Raspberry Pi 3

ПРИЛОЖЕНИЕ 3

Программа обмена данными по CAN BUS

```

#include <stdio.h>
#include <stdlib.h>
#include "CAN_API.h"
#include <windows.h>

//-----struct send data CAN BUS
struct ds {
float a1;
float b1;
float a2;
float b2;
float a3;
float b3;
};
struct ds sf; // struct send data into CAN BUS

struct dr { // struct read data from CAN BUS
float a1;
float b1;
float a2;
float b2;
float a3;
float b3;
};
struct dr rf; // struct read data from CAN BUS

double stm=469.997;
int ch;
typedef TCAN_HANDLE (*FNPTR1)(CHAR *,CHAR *,CHAR *,CHAR
*,void *,DWORD);
typedef TCAN_HANDLE (*FNPTR2)(TCAN_HANDLE);
typedef TCAN_HANDLE (*FNPTR3)(TCAN_HANDLE,CAN_MSG*);
typedef TCAN_HANDLE (*FNPTR4)(TCAN_HANDLE,CHAR *);

HMODULE hMod;
TCAN_HANDLE Handle;

```

```

TCAN_STATUS Status;
//-----Function Exchange data on CAN BUS
void srd(void)
{
byte *yf, *yf1, *yf2;
int q, nbz;
long j;
FILE *fp;
    CHAR *ComPort = "COM4";
    CHAR *szBitrate = "1000";
// "9216000";
    CHAR *acceptance_code = "1FFFFFFF";
    CHAR *acceptance_mask = "00000000";
    VOID *flags = CAN_TIMESTAMP_OFF;
    DWORD Mode = Normal; //LoopBack;
    FNPTR1 CAN_Open;
    FNPTR2 CAN_Close;
    FNPTR2 CAN_Flush;
    FNPTR3 CAN_Write;
    FNPTR3 CAN_Read;
    FNPTR4 CAN_Version;
    FNPTR2 CAN_Status;
    char version[10];
    CAN_MSG SendMSG;
    CAN_MSG RecvMSG;
//  Handle = -1;
    Status = 0;
    nbz=0;
    j=0;
//-----
    CAN_Open = (FNPTR1)GetProcAddress(hMod, "CAN_Open");
    CAN_Close = (FNPTR2)GetProcAddress(hMod, "CAN_Close");
    CAN_Flush = (FNPTR2)GetProcAddress(hMod, "CAN_Flush");
    CAN_Write = (FNPTR3)GetProcAddress(hMod, "CAN_Write");
    CAN_Read = (FNPTR3)GetProcAddress(hMod, "CAN_Read");
    CAN_Version = (FNPTR4)GetProcAddress(hMod, "CAN_Version");
    CAN_Status = (FNPTR2)GetProcAddress(hMod, "CAN_Status");
    RecvMSG.Flags = CAN_FLAGS_STANDARD; //EXTENDED;
    RecvMSG.Size = 8;
//-----

```

```

    if (ch==0)
        Handle = CAN_Open ( ComPort, szBtrate, acceptance_code,
acceptance_mask, flags, Mode );
    //-- Example read data from RecvMSG.Data
    RecvMSG.Id = 0x001;
    memset ( version, 0, sizeof ( char ) * 10 );
    Status = CAN_Flush ( Handle );
    Status = CAN_Version ( Handle, version );
    for (j=0;j<60000;j++) {
        Status = CAN_Read ( Handle, &RecvMSG );
        if ( Status == CAN_ERR_OK ) {
            switch (RecvMSG.Id) {
                case 1:      yf=(byte *)&rf.a1;          for (q=0;q<8; q++)
*yf++=RecvMSG.Data[q]; break;
                case 2:      yf=(byte *)&rf.a1+8;        for (q=0;q<8; q++)
*yf++=RecvMSG.Data[q]; break;
                case 3:      yf=(byte *)&rf.a1+16;       for (q=0;q<8; q++)
*yf++=RecvMSG.Data[q]; break;
                default:
                    break;
            }
        }
    }
    // Контроль приема данных
    fp=fopen("tc.txt","at");
    fprintf(fp,"Handle=%d ID=%X Status= %d %.4f %.4f %.4f %.4f %.4f
%.4f\n", Handle, RecvMSG.Id, Status,rf.a1,rf.b1,rf.a2,rf.b2,rf.a3,rf.b3);
    fclose(fp);
}
} //-- End Example read data from RecvMSG.Data
/*
//----- Example send data into SendMSG.Data
    for (nbz=0;nbz<3;nbz++) {
        yf=(byte *)&sf.a1+8*nb;
        for (q=0;q<8; q++)
            SendMSG.Data[q] = *yf++;
    }
//-----
    switch (nbz) {
        case 0: SendMSG.Id = 0x001; break;
        case 1: SendMSG.Id = 0x002; break;
        case 2: SendMSG.Id = 0x003; break;
        default:

```



```

    break;
}
memset ( version, 0, sizeof ( char ) * 10 );
Status = CAN_Flush ( Handle );
Status = CAN_Version ( Handle, version );
Status = CAN_Write ( Handle, &SendMSG );
if (nbz==2) { // Контроль передачи данных
    fp=fopen("tcb.txt","at");
    fprintf(fp,"Handle=%d ID=%X Status= %d %.4f %.4f %.4f %.4f %.4f
%.4f\n", Handle, RecvMSG.Id, Status,rf.a1,rf.b1,rf.a2,rf.b2,rf.a3,rf.b3);
    fclose(fp);
}
/-- End Example send data into SendMSG.Data
*/
if (ch==3999)
    Status = CAN_Close ( Handle );
}
//-----End function sd()
int main(void)
{
    float f1, f2;
    int r;
    long j;
    hMod = LoadLibrary ("CAN_API.DLL");
    if (hMod==NULL) {
        printf ("LoadLibrary failed\n");
    }
    for (ch=0;ch<4000;ch++)
        srd();
    FreeLibrary(hMod);
    printf ( "Test finish_FreeLibrary\n" );
    return 0;
} //End program main

```

Исходный текст программы RainFollow.cpp

```
#include <cstdlib>
#include <iostream>
#include <vector>
#include <math.h>
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <conio.h>
#include <time.h>
#include <sys/types.h>
#include <sys/timeb.h>

#define MAX 51000000
using namespace std;
class Rainflow
{
private:
    float C[100];
    float AverageMean[100];
    float MaxPeak[100];
    float MinValley[100];
    float MaxAmp[100];
    float AverageAmp[100];
    double L[20];
    double ylast;
    float sum;
    float ymax;
    float mina,maxa;
    float t;
    float scale;
    float X,Y;
    long ijk;
    long kv;
    long hold;
    long i,j,k,n;
    long num;
```

```

    long nkV;
    int ic, iscale;
    long N, NP;
    long last_a;
    FILE *pFile[6];
    char filename[6][FILENAME_MAX];
    vector<float> y;
    vector<vector<float> > B; // need space > >

public:
    void read_data(void);
    void rainflow_engine(void);
    void damage_index(void);
    void print_data(void);
};
// *****

int main()
{
    Rainflow rf;

    printf("\n ");
    printf("\n rainflow.cpp ver 2.8 June 24, 2014 \n");
    printf("\n by Tom Irvine Email: tom@vibrationdata.com \n");
    printf("\n ASTM E 1049-85 (2005) Rainflow Counting Method \n");

    time_t start = time(0);
    rf.read_data();
    rf.rainflow_engine();
    rf.print_data();
    int icd;
    printf("\n\n Calculate relative damage index D? 1=yes 2=no \n");
    scanf("%d",&icd);
    if(icd==1)
    {
        rf.damage_index();
    }
    time_t end = time(0);
    double time = difftime(end, start);
    printf("\n Elapsed Time = %8.4g sec \n",time);

```

```

        printf("\n Press any key to exit.\n");
        getch();
        exit(1);
    }
    void Rainflow::print_data()
    {
        printf("\n Amplitude = (peak-valley)/2 \n");

        fprintf(pFile[1],"\n Amplitude = (peak-valley)/2 \n");

        //*****
        *****

        printf("\n ");printf("\n      Range      Cycle   Ave   Max   Ave
Min      Max");
        printf("\n      (units)      Counts   Amp   Amp   Mean   Valley   Peak
\n");
        for(i=13;i>=1;i--)
        {
            printf(" %8.2lf to %8.2lf\t%8.1lf\t%6.4g\t%6.4g\t%6.4g\t%6.4g\t
%6.4g\n",
L[i],L[i+1],C[i],AverageAmp[i],MaxAmp[i],AverageMean[i],MinValley[i],MaxPe
ak[i]);
        }
        fprintf(pFile[1],"\n ");
        fprintf(pFile[1],"\n      Range      Cycle   Ave   Max   Ave   Min
Max");
        fprintf(pFile[1],"\n      (units)      Counts   Amp   Amp   Mean   Valley
Peak \n");

        for(i=13;i>=1;i--)
        {
            fprintf(pFile[1], " %8.4lf to %8.4lf\t%8.1lf\t%6.4g\t%6.4g\t%6.4g\t
%6.4g\t %6.4g\n",
L[i],L[i+1],C[i],AverageAmp[i],MaxAmp[i],AverageMean[i],MinValley[i],MaxPe
ak[i]);
            fprintf(pFile[2], " %ld \t %g \n",i,C[i]);
        }
        fclose(pFile[0]);
        fclose(pFile[1]);
        fclose(pFile[2]);

```

```

fclose(pFile[3]);
fclose(pFile[4]);

printf("\n\n Total Cycles = %g hold=%ld NP=%ld
ymax=%g\n",sum,hold,NP,ymax);
fprintf(pFile[1],"\n\n Total Cycles = %g hold=%ld NP=%ld
ymax=%g\n",sum,hold,NP,ymax);

printf("\n\n The output files are: \n");

printf(" %s \n",filename[1]);
printf(" %s \n",filename[2]);
printf(" %s \n",filename[3]);
printf(" %s \n",filename[4]);
}
void Rainflow::read_data(void)
{
float aa;

printf("\n The input file must be a time history. \n");
printf("\n Select format: ");
printf("\n 1=amplitude ");
printf("\n 2=time & amplitude \n");

scanf("%d",&ic);

if(ic==1)
{
printf( "\n\n The base input file must contain one column: unit
\n");
}
else
{
printf( "\n\n The base input file must contain two columns: \n");
printf( " time & unit \n");
}

printf( "\n Input filename \n");
scanf("%s",filename[0]);

```

```

pFile[0] = fopen(filename[0], "rb");

while(pFile[0] == NULL )
{

    printf("\n Failed to open file: %s \n", filename[0]);

    printf("\n Please enter the input filename: \n");

    scanf("%s",filename[0]);

    pFile[0] = fopen(filename[0], "rb");

}
printf("\n File: %s opened. \n", filename[0]);

printf( "\n\n Enter the output table filename: \n");
scanf("%s",filename[1]);

pFile[1]=fopen(filename[1], "w");

strcpy(filename[2],"rainflow_graph.out");
pFile[2]=fopen(filename[2], "w");

strcpy(filename[3],"range_cycles.out");
pFile[3]=fopen(filename[3], "w");

strcpy(filename[4],"amp_cycles.out");
pFile[4]=fopen(filename[4], "w");

//  strcpy(filename[5],"points.out");
//  pFile[5]=fopen(filename[5], "w");

i=0;

if(ic==1)
{
    while( fscanf(pFile[0],"%f",&aa)>0)

```

```

        {
            y.push_back(aa);

            i++;

            if(i==MAX)
            {
                printf("\n Warning: input data limit reached \n.");
                break;
            }
        }
    }
else
{
    while( fscanf(pFile[0],"%f %f",&t,&aa)>0)
    {
        y.push_back(aa);

        i++;

        if(i==MAX)
        {
            printf("\n Warning: input data limit reached \n.");
            break;
        }
    }
}
ylast=y[i-1];

NP=i+1;

// printf("\n ref 1: last_a = %ld \n",last_a);

printf("\n ");
printf("\n Multiply data by scale factor?");
printf("\n 1=yes 2=no \n");
scanf("%d",&iscale);

if(iscale==1)

```

```

    {
        printf("\n Enter scale factor \n");
        scanf("%f",&scale);

        for(i=0;i<NP;i++)
        {
            y[i]*=scale;
        }
    }
}

void Rainflow::damage_index(void)
{
    double b;
    double D=0;
    double Y;

    printf("\n\n Enter fatigue exponent: ");
    scanf("%lf",&b);

    for(long i=0;i<=kv;i++)
    {
        Y=B[i][0];
        D+=B[i][1]*pow((Y/2.),b);
    }
    printf("\n D=%8.4g \n",D);
}

void Rainflow::rainflow_engine(void)
{
    vector<float> a;

    double slope1;
    double slope2;

    ymax=0.;

    nk=0;

    k=0;

```



```

//      a[k]=y[k];
a.push_back(y[k]);

      k=1;

for(i=1;i<(NP-1);i++)
{

      slope1=( y[i]-y[i-1]);
      slope2=(y[i+1]-y[i]);

      if((slope1*slope2)<=0. && fabs(slope1)>0.)
      {
              a.push_back(y[i]);
              k++;
      }

}
a.push_back(ylast);
k++;

      last_a=k-1;

hold=last_a;

long WIDTH=4;
long HEIGHT=0;
B.resize(HEIGHT);

for(i=0; i < HEIGHT; i++)
{
      B[i].resize(WIDTH);
}

// printf(" H=%ld W=%ld ",HEIGHT,WIDTH);
// getch();

      mina=100000;

```

```

maxa=-mina;

for(i=0;i<=last_a;i++)
{
    if(a[i]<mina)
    {
        mina=a[i];
    }
    if(a[i]>maxa)
    {
        maxa=a[i];
    }

//      fprintf(pFile[5]," %8.4g \n",a[i]);
}

num=long(maxa-mina)+1;

n=0;
i=0;
j=1;

sum=0;

kv=0;

long LLL=last_a;

std::vector<float> row(4);

printf("\n percent completed \n");

while(1)
{
Y=(fabs(a[i]-a[i+1]));
X=(fabs(a[j]-a[j+1]));

if(X>=Y && Y>0 && Y<1.0e+20)
{
    if(Y>ymax)

```

```

        {ymax=Y;}

    if(i==0)
        {
            n=0;
            sum+=0.5;

            row[3]=a[i+1];
            row[2]=a[i];
            row[1]=0.5;
            row[0]=Y;

            B.push_back(row);

            //      printf("1 %8.4g %8.4g %8.4g %8.4g
\n",B[kv][0],B[kv][1],B[kv][2],B[kv][3]);

            kv++;

            a.erase (a.begin());

            last_a--;

            i=0;
            j=1;
        }
    else
        {
            sum+=1;

            row[3]=a[i+1];
            row[2]=a[i];
            row[1]=1.;
            row[0]=Y;

            B.push_back(row);

            //      printf("2 %8.4g %8.4g %8.4g %8.4g
\n",B[kv][0],B[kv][1],B[kv][2],B[kv][3]);

```

```

kv++;

n=0;

a.erase (a.begin()+(i+1));
a.erase (a.begin()+i);

last_a-=2;

i=0;
j=1;
}

nkvv++;

if(nkv==3000)
{
double ratio = fabs((last_a)/double(LL));

printf(" %3.1lf\n", (1-ratio)*100.);
nkvv=0;
}
}
else
{
i++;
j++;
}

if((j+1)>last_a)
{
break;
}

}

for(i=0;i<(last_a);i++)

```

```

    {
        Y=(fabs(a[i]-a[i+1]));
        if(Y>0. && Y<1.0e+20)
            {
                sum+=0.5;

                row[3]=a[i+1];
                row[2]=a[i];
                row[1]=0.5;
                row[0]=Y;
                B.push_back(row);
                //      printf("3  %8.4g %8.4g %8.4g %8.4g
\n",B[kv][0],B[kv][1],B[kv][2],B[kv][3]);

                kv++;

                if(Y>ymax)
                    {ymax=Y;}

            }

    }

L[1]=0;
L[2]=2.5;
L[3]=5;
L[4]=10;
L[5]=15;
L[6]=20;
L[7]=30;
L[8]=40;
L[9]=50;
L[10]=60;
L[11]=70;
L[12]=80;
L[13]=90;
L[14]=100;

for(ijk=1;ijk<=14;ijk++)

```

```

{
    L[ijk]*=ymax/100.;

    C[ijk]=0.;
    AverageMean[ijk]=0.;
    MaxPeak[ijk]=-1.0e+20;
    MinValley[ijk]= 1.0e+20;

    MaxAmp[ijk]=-1.0e+20;
    AverageAmp[ijk]= 1.0e+20;

//    printf(" L[%ld]=%g ymax=%8.4g\n",ijk,L[ijk],ymax);
}

//  getch();

kv--;

for(ijk=13;ijk>=0;ijk--)
{
    AverageAmp[ijk]=0.;
}
For (i=0;i<=kv;i++)
{
    Y=B[i][0];
//    printf(" %ld %ld %10.4e \t %3.1f\n",i,kv,Y,B[i][1]);

    fprintf(pFile[3]," %10.4e \t %3.1f\n",Y,B[i][1]);
    fprintf(pFile[4]," %10.4e \t %3.1f\n",Y/2.,B[i][1]);

//    printf("i=%d Y=%g \n",i,Y);

    for(ijk=13;ijk>=0;ijk--)
    {
//        printf(" %8.4g %8.4g %8.4g \n",Y,L[ijk],L[ijk+1]);

        if(Y>=L[ijk] && Y<=L[ijk+1])
        {
            C[ijk]=C[ijk]+B[i][1];

```

```

AverageMean[ijk]+=B[i][1]*(B[i][3]+B[i][2])*0.5; // weighted
average

    if(B[i][3]>MaxPeak[ijk])
    {
        MaxPeak[ijk]=B[i][3];
    }
    if(B[i][2]>MaxPeak[ijk])
    {
        MaxPeak[ijk]=B[i][2];
    }

    if(B[i][3]<MinValley[ijk])
    {
        MinValley[ijk]=B[i][3];
    }
    if(B[i][2]<MinValley[ijk])
    {
        MinValley[ijk]=B[i][2];
    }

    if(Y>MaxAmp[ijk])
    {
        MaxAmp[ijk]=Y;
    }

    AverageAmp[ijk]+=B[i][1]*Y*0.5;

    break;
}
}
}

for(ijk=1;ijk<=14;ijk++)
{
    if(C[ijk]>0)
    {
        AverageMean[ijk]/=C[ijk];
        AverageAmp[ijk]/=C[ijk];
    }
}

```

```

MaxAmp[ijk]/=2.;

if(C[ijk]<0.5)
{
    AverageAmp[ijk]=0.;
    MaxAmp[ijk]=0.;
    AverageMean[ijk]=0.;
    MinValley[ijk]=0.;
    MaxPeak[ijk]=0.;
}
//      printf(" %8.4g %8.4g %8.4g %8.4g %8.4g
\n",AverageAmp[ijk],MaxAmp[ijk],AverageMean[ijk],MinValley[ijk],MaxPeak[ij
k]);
}

```



```
//----- File: SDV.c  
// --- THIS FILE GENERATED BY S-FUNCTION BUILDER: 3.0 ---  
/* This file is an S-function produced by the S-Function  
 * Builder which only recognizes certain fields. Changes made  
 * outside these fields will be lost the next time the block is  
 * used to load, edit, and resave this file. This file will be overwritten  
 * by the S-function Builder block. If you want to edit this file by hand,  
 * you must change it only in the area defined as:  
 *     %%%-SFUNWIZ_defines_Changes_BEGIN  
 *     #define NAME 'replacement text'  
 *     %%% SFUNWIZ_defines_Changes_END  
 * DO NOT change NAME--Change the 'replacement text' only.  
 * For better compatibility with the Simulink Coder, the  
 * "wrapper" S-function technique is used. This is discussed  
 * in the Simulink Coder's Manual in the Chapter titled,  
 * "Wrapper S-functions".  
 * | See matlabroot/simulink/src/sfuntmpl_doc.c for a more detailed template |  
* Created: Mon Jan 7 14:43:02 2019  
*/  
  
#define S_FUNCTION_LEVEL 2  
#define S_FUNCTION_NAME SDV  
  
/*<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<  
<<<<<<<*/  
  
/* %%%-SFUNWIZ_defines_Changes_BEGIN --- EDIT HERE TO _END */  
#define NUM_INPUTS      18  
#define IN_PORT_0_NAME   u0  
#define INPUT_0_WIDTH    1  
#define INPUT_DIMS_0_COL  1  
#define INPUT_0_DTYPE    real_T  
#define INPUT_0_COMPLEX  COMPLEX_NO  
#define IN_0_FRAME_BASED FRAME_NO  
#define IN_0_BUS_BASED   0  
#define IN_0_BUS_NAME  
#define IN_0_DIMS        1-D  
#define INPUT_0_FEEDTHROUGH 1
```

```

#define IN_0_ISSIGNED      0
#define IN_0_WORDLENGTH    8
#define IN_0_FIXPOINTSCALING 1
#define IN_0_FRACTIONLENGTH 9
#define IN_0_BIAS          0
#define IN_0_SLOPE         0.125
/* Input Port 1 */
#define IN_PORT_1_NAME     s0
#define INPUT_1_WIDTH      1
#define INPUT_DIMS_1_COL   1
#define INPUT_1_DTYPE      real_T
#define INPUT_1_COMPLEX    COMPLEX_NO
#define IN_1_FRAME_BASED   FRAME_NO
#define IN_1_BUS_BASED     0
#define IN_1_BUS_NAME
#define IN_1_DIMS          1-D
#define INPUT_1_FEEDTHROUGH 1
#define IN_1_ISSIGNED      0
#define IN_1_WORDLENGTH    8
#define IN_1_FIXPOINTSCALING 1
#define IN_1_FRACTIONLENGTH 9
#define IN_1_BIAS          0
#define IN_1_SLOPE         0.125
// АналогичноSimulink создает для Input Port
/* Input Port 2 */
//-----
/* Input Port 17 */
#define IN_PORT_17_NAME    z2
#define INPUT_17_WIDTH     1
#define INPUT_DIMS_17_COL  1
#define INPUT_17_DTYPE     real_T
#define INPUT_17_COMPLEX   COMPLEX_NO
#define IN_17_FRAME_BASED  FRAME_NO
#define IN_17_BUS_BASED    0
#define IN_17_BUS_NAME
#define IN_17_DIMS         1-D
#define INPUT_17_FEEDTHROUGH 1
#define IN_17_ISSIGNED     0
#define IN_17_WORDLENGTH   8
#define IN_17_FIXPOINTSCALING 1

```

```

#define IN_17_FRACTIONLENGTH 9
#define IN_17_BIAS          0
#define IN_17_SLOPE         0.125

#define NUM_OUTPUTS        13
/* Output Port 0 */
#define OUT_PORT_0_NAME    y0
#define OUTPUT_0_WIDTH     1
#define OUTPUT_DIMS_0_COL  1
#define OUTPUT_0_DTYPE     real_T
#define OUTPUT_0_COMPLEX   COMPLEX_NO
#define OUT_0_FRAME_BASED  FRAME_NO
#define OUT_0_BUS_BASED    0
#define OUT_0_BUS_NAME
#define OUT_0_DIMS         1-D
#define OUT_0_ISSIGNED     1
#define OUT_0_WORDLENGTH   8
#define OUT_0_FIXPOINTSCALING 1
#define OUT_0_FRACTIONLENGTH 3
#define OUT_0_BIAS         0
#define OUT_0_SLOPE        0.125
// Аналогично создает Simulink для Output Port
/* Output Port 1 */
//-----
/* Output Port 12 */
#define OUT_PORT_12_NAME    f2
#define OUTPUT_12_WIDTH     1
#define OUTPUT_DIMS_12_COL  1
#define OUTPUT_12_DTYPE     real_T
#define OUTPUT_12_COMPLEX   COMPLEX_NO
#define OUT_12_FRAME_BASED  FRAME_NO
#define OUT_12_BUS_BASED    0
#define OUT_12_BUS_NAME
#define OUT_12_DIMS         1-D
#define OUT_12_ISSIGNED     1
#define OUT_12_WORDLENGTH   8
#define OUT_12_FIXPOINTSCALING 1
#define OUT_12_FRACTIONLENGTH 3
#define OUT_12_BIAS         0
#define OUT_12_SLOPE        0.125

```

```
#define NPARAMS          0

#define SAMPLE_TIME_0    INHERITED_SAMPLE_TIME
#define NUM_DISC_STATES   0
#define DISC_STATES_IC   [0]
#define NUM_CONT_STATES   0
#define CONT_STATES_IC   [0]


#define SFUNWIZ_GENERATE_TLC 0
#define SOURCEFILES "__SFB__"
#define PANELINDEX        6
#define USE_SIMSTRUCT      0
#define SHOW_COMPILE_STEPS 0
#define CREATE_DEBUG_MEXFILE 0
#define SAVE_CODE_ONLY     1
#define SFUNWIZ_REVISION   3.0
/* %%%-SFUNWIZ_defines_Changes_END --- EDIT HERE TO _BEGIN */
/*<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
<<<<<<<* /

#include "simstruc.h"
//
#include <stdio.h>
#include <stdlib.h>
//Здесь указываем свои подключаемые библиотеки и переменные
#include <math.h>
float rd[384000];
long tb[10][2]={ {1550,7},{3540,11},{5520,11},{12500,25},{12500,25}};
//long tb[10][2]={ {23000,5},{31000,6},{31500,5},{33500,5}};
//long tb[10][2]={ {13000,5},{23000,5},{31000,6},{33500,5},{34000,5}};
//double
u_kp[19]={0.0,16.41,13.80,11.28,9.49,7.76,6.53,5.43,4.57,3.59,3.02,2.47,2.08,1.7
0,1.43,1.19,1.00}; //OMA
double u_kp[16]={0.0,8.73,6.13,4.51,3.50,2.78,1.75,1.00, 0.795}; //ЯМ3-202
double cyx=8000.0;
double ygk=0.0;
double f_1=0.0;
double f_2=0.0;
double l1=2.0;
double fmm,hx,mp,ma,Fy1,Fy2,ynd, gma;
```

```

double hx0=1;
double kpd=0.88;
long m=0;
long chm=0;
long jj, xl0, xtb, vv;
long j=0;
long chw;
int n_kp=1;
int n_kp0=1;
long ch=0;
long nbl=0;
long rt=0;
float fm, fm0, fm1,vm;
struct d {
    long dl; // length of a way
    float v; // speed of vehicle
    float zd; // delay in mc
    float spe; // engine speed
    float pwe; // engine power
};
struct d vd;
extern void SDV_Outputs_wrapper(const real_T *u0,
                                const real_T *s0,
                                const real_T *md,
                                const real_T *va,
                                const real_T *Nd,
                                const real_T *Pu,
                                const real_T *thr,
                                const real_T *x1,
                                const real_T *x2,
                                const real_T *x3,
                                const real_T *x4,
                                const real_T *xa,
                                const real_T *vy,
                                const real_T *ya,
                                const real_T *vw,
                                const real_T *wa,
                                const real_T *z1,
                                const real_T *z2,
                                real_T *y0,

```

```

real_T *sF,
real_T *ndv,
real_T *nkp,
real_T *y4,
real_T *y1,
real_T *y2,
real_T *y3,
real_T *fx,
real_T *fy,
real_T *fw,
real_T *fl,
real_T *f2);

```

```
//-----
```

```
#include <math.h>
```

```
/* параметры модели
```

```
KPD_tr=[0.863;0.863;0.863;0.863;0.863;0.863;0.863;0.887;0.878;0.878;0.878;0.878;0.878;0.878;0.878;0.902];
```

```
t_sh=0.5; //ремя переключения передачи
```

```
//U_most=3.45; //передаточное число ведущего моста
```

```
U_most=6.59*1.107; //передаточное число ведущего моста
```

```
//R0=0.522; радиус качения ведущего колеса, м
```

```
R0=0.675; радиус качения ведущего колеса МАЗ-6317, м
```

```
f=0.003902727; //коэффициент сопротивления качению
```

```
%%%Определение коэфф. учёта вращающихся масс%%%
```

```
Jen=4.51;
```

```
Jk=251.136;
```

```
delta_whells=1+(Jk)/(m_a*R0^2);
```

```
delta=delta_whells+Jen*KPD_tr.*(u_kp.*U_most/R0).^2/m_a;
```

```
F_T=30000; %суммарная максимальная тормозная сила, Н
```

```
*/
```

```
static void mdlInitializeSizes(SimStruct *S)
```

```
{
```

```
DECL_AND_INIT_DIMSINFO(inputDimsInfo);
```

```
DECL_AND_INIT_DIMSINFO(outputDimsInfo);
```

```
ssSetNumSFcnParams(S, NPARAMS);
```

```
if (ssGetNumSFcnParams(S) != ssGetSFcnParamsCount(S)) {
```

```
    return; /* Parameter mismatch will be reported by Simulink */
```

```
}
```

```

ssSetNumContStates(S, NUM_CONT_STATES);
ssSetNumDiscStates(S, NUM_DISC_STATES);
if (!ssSetNumInputPorts(S, NUM_INPUTS)) return;
/*Input Port 0 */
ssSetInputPortWidth(S, 0, INPUT_0_WIDTH); /* */
ssSetInputPortDataType(S, 0, SS_DOUBLE);
ssSetInputPortComplexSignal(S, 0, INPUT_0_COMPLEX);
ssSetInputPortDirectFeedThrough(S, 0, INPUT_0_FEEDTHROUGH);
ssSetInputPortRequiredContiguous(S, 0, 1); /*direct input signal access*/
/*Input Port 1 */
ssSetInputPortWidth(S, 1, INPUT_1_WIDTH); /* */
ssSetInputPortDataType(S, 1, SS_DOUBLE);
ssSetInputPortComplexSignal(S, 1, INPUT_1_COMPLEX);
ssSetInputPortDirectFeedThrough(S, 1, INPUT_1_FEEDTHROUGH);
ssSetInputPortRequiredContiguous(S, 1, 1); /*direct input signal access*/
// Здесь аналогично Simulink создает параметры для Input Port
/*Input Port 2 */
//-----
/*Input Port 17 */
ssSetInputPortWidth(S, 17, INPUT_17_WIDTH); /* */
ssSetInputPortDataType(S, 17, SS_DOUBLE);
ssSetInputPortComplexSignal(S, 17, INPUT_17_COMPLEX);
ssSetInputPortDirectFeedThrough(S, 17, INPUT_17_FEEDTHROUGH);
ssSetInputPortRequiredContiguous(S, 17, 1); /*direct input signal access*/
if (!ssSetNumOutputPorts(S, NUM_OUTPUTS)) return;
/* Output Port 0 */
ssSetOutputPortWidth(S, 0, OUTPUT_0_WIDTH);
ssSetOutputPortDataType(S, 0, SS_DOUBLE);
ssSetOutputPortComplexSignal(S, 0, OUTPUT_0_COMPLEX);
/* Output Port 1 */
// Здесь аналогично Simulink создает параметры для Output Port
// -----
/* Output Port 12 */
ssSetOutputPortWidth(S, 12, OUTPUT_12_WIDTH);
ssSetOutputPortDataType(S, 12, SS_DOUBLE);
ssSetOutputPortComplexSignal(S, 12, OUTPUT_12_COMPLEX);

ssSetNumSampleTimes(S, 1);
ssSetNumRWork(S, 0);
ssSetNumIWork(S, 0);

```

```

ssSetNumPWork(S, 0);
ssSetNumModes(S, 0);
ssSetNumNonsampledZCs(S, 0);

ssSetSimulinkVersionGeneratedIn(S, "8.6");

/* Take care when specifying exception free code - see sfuntmpl_doc.c */
ssSetOptions(S, (SS_OPTION_EXCEPTION_FREE_CODE |
                 SS_OPTION_WORKS_WITH_CODE_REUSE));
}
#define MDL_SET_INPUT_PORT_FRAME_DATA
static void mdlSetInputPortFrameData(SimStruct *S,
                                     int_T    port,
                                     Frame_T   frameData)
{
    ssSetInputPortFrameData(S, port, frameData);
}
/* Function: mdlInitializeSampleTimes
=====
* Abstract:   Specifiy the sample time.
*/
static void mdlInitializeSampleTimes(SimStruct *S)
{
    ssSetSampleTime(S, 0, SAMPLE_TIME_0);
    ssSetModelReferenceSampleTimeDefaultInheritance(S);
    ssSetOffsetTime(S, 0, 0.0);
}
#define MDL_SET_INPUT_PORT_DATA_TYPE
static void mdlSetInputPortDataType(SimStruct *S, int port, DTypeId dType)
{
    ssSetInputPortDataType( S, 0, dType);
}
#define MDL_SET_OUTPUT_PORT_DATA_TYPE
static void mdlSetOutputPortDataType(SimStruct *S, int port, DTypeId dType)
{
    ssSetOutputPortDataType(S, 0, dType);
}
#define MDL_SET_DEFAULT_PORT_DATA_TYPES
static void mdlSetDefaultPortDataTypes(SimStruct *S)
{

```



```

    ssSetInputPortDataType( S, 0, SS_DOUBLE);
    ssSetOutputPortDataType(S, 0, SS_DOUBLE);
}
/* Function: mdlOutputs */
static void mdlOutputs(SimStruct *S, int_T tid)
{
    double x, x_2, x_3, sx, y_0,y_1, y_2, ts, N, h;
    long n, i, nt, zl, xl,xc;
    float an,my;
    //float fm, vm;
    double yn, kr1, v_a, Ndv,ndvs,Fk, Fv,Fy,Fy_1,Fy_2,Fa2,jm,a2,kz, m_d,p_u,a,b,c,
    P0,P1,P2,P3,u,t, t2,t3,Nt,Neq;
    FILE *fp1,*fp2, *fp3;
    xl=0;
    //n_kp=0;
    const real_T *u0 = (const real_T*) ssGetInputPortSignal(S,0);
    const real_T *s0 = (const real_T*) ssGetInputPortSignal(S,1);
    const real_T *md = (const real_T*) ssGetInputPortSignal(S,2);
    const real_T *va = (const real_T*) ssGetInputPortSignal(S,3);
    const real_T *Nd = (const real_T*) ssGetInputPortSignal(S,4);
    const real_T *Pu = (const real_T*) ssGetInputPortSignal(S,5);
    const real_T *thr = (const real_T*) ssGetInputPortSignal(S,6);
    const real_T *x1 = (const real_T*) ssGetInputPortSignal(S,7);
    const real_T *x2 = (const real_T*) ssGetInputPortSignal(S,8);
    const real_T *x3 = (const real_T*) ssGetInputPortSignal(S,9);
    const real_T *x4 = (const real_T*) ssGetInputPortSignal(S,10);
    const real_T *xa = (const real_T*) ssGetInputPortSignal(S,11);
    const real_T *vy = (const real_T*) ssGetInputPortSignal(S,12);
    const real_T *ya = (const real_T*) ssGetInputPortSignal(S,13);
    const real_T *vw = (const real_T*) ssGetInputPortSignal(S,14);
    const real_T *wa = (const real_T*) ssGetInputPortSignal(S,15);
    const real_T *z1 = (const real_T*) ssGetInputPortSignal(S,16);
    const real_T *z2 = (const real_T*) ssGetInputPortSignal(S,17);
    real_T *y0 = (real_T *)ssGetOutputPortRealSignal(S,0);
    real_T *sF = (real_T *)ssGetOutputPortRealSignal(S,1);
    real_T *ndv = (real_T *)ssGetOutputPortRealSignal(S,2);
    real_T *nkp = (real_T *)ssGetOutputPortRealSignal(S,3);
    real_T *y4 = (real_T *)ssGetOutputPortRealSignal(S,4);
    real_T *y1 = (real_T *)ssGetOutputPortRealSignal(S,5);
    real_T *y2 = (real_T *)ssGetOutputPortRealSignal(S,6);

```

```

real_T    *y3 = (real_T *)ssGetOutputPortRealSignal(S,7);
real_T    *fx = (real_T *)ssGetOutputPortRealSignal(S,8);
real_T    *fy = (real_T *)ssGetOutputPortRealSignal(S,9);
real_T    *fw = (real_T *)ssGetOutputPortRealSignal(S,10);
real_T    *f1 = (real_T *)ssGetOutputPortRealSignal(S,11);
real_T    *f2 = (real_T *)ssGetOutputPortRealSignal(S,12);
//----- Расшифровка параметров S-Function Builder -----Входные параметры
//u0
//s0
//md
//v_a скорость авто
//Nd мощность ДВС
//Pk усилие на колесах
//Pu
//thr потери н оборудование
//x1 параметр загрузки ДВС/обороты для переключения КПП вверх
//x2 параметр загрузки ДВС/обороты для переключения КПП вниз
//x3
//x4
//-----Выходные параметры
//y0
//sF
//ndv обороты ДВС
//nkp передаточное отношение КПП
//y4 степень загрузки двигателя
//y1 высота неровности под 1 колесом
//y2 высота неровности под 2 колесом
//y3 высота неровности под 3 колесом
ts=0.125;
//-----инициализация переменных в цикле
v_a=va[0];
Ndv=Nd[0];
Neq=thr[0];
yn=u0[0];
m_d=md[0];
p_u=Pu[0];
ygk=z1[0];
vm=x1[0]; //Скорость изменения нагрузки
fmm =x2[0]; //Усилие на мост
fm=fmm;

```

```

//----- Занесение в массив высот неровности
if (ch==0)
{
sx=384000*nbl;
fp1=fopen("rd.bt","rb");
fseek(fp1, sx, SEEK_SET);
for (i=0; i<96000; i++)
{
fread(&an, 4, 1, fp1);
rd[i]=an;
} //for
fclose(fp1);
fm0=0.0;
fm1=2.0;
xtb=tb[0][0];
vv=tb[0][1];
ma=20000.0; //Масса автомобиля
// Fy=811.715+3639*yn; // 9.81*ma*(0.0039+0.0175 *yn);
Fw=9.81*ma*(f*cosa+sina);
// Сопротивление качения и наклона
gma=9.81*ma; //Множитель
Fy1=0.0039*gma; //Сопротивление качения
ch=1;
//-----
// определяю значение пути для точки X1
x = s0[0]-(11898*nbl); //расчетное положение на трассе для массива высот
при подкачке //инфо
x1 = s0[0]; // физическое положение на трассе
if (x<0.0)
x=0.0;
xc=x;
//mp= x1[0];
n=u/x/ts;
t=u-n;
//t=fabs(u- n);
t2=t*t;
t3=t*t*t;
// Микропрофиль правая колея переднее колесо
P0=rd[n-1];
P1=rd[n];

```

```

P2=rd[n+1];
P3=rd[n+2];
*y1 =mp= 0.5* ((2*P1) +(-P0 + P2) * t +(2*P0 - 5*P1 + 4*P2 - P3) *t2 +(-P0 +
3*P1- 3*P2 + P3) *t3);
ynd=yn=0.0;
//----- Определение величины уклона
if (n>200)
ynd=yn=(rd[n+200]-rd[n-200])/50.0;
*f2=ynd;
/*
if (ynd>0.04&&chm!=1)
{
  n_kp--;
  if (n_kp<1)
    n_kp=1;
  chm=1;
}
*/
//-----
if (x>=5.80)
{
  x_2=x-4.2;
  n=x_2/ts;
  n=u=x_2/ts;
  t=u-n;
  //t=fabs(u- n);
  t2=t*t;
  t3=t*t*t;
  // Микропрофиль правая колея с задержкой база 4.2
  P0=rd[n-1];
  P1=rd[n];
  P2=rd[n+1];
  P3=rd[n+2];
  *y2= 0.5*((2*P1) +(-P0 + P2) * t +(2*P0 - 5*P1 + 4*P2 - P3) *t2 +(-P0 + 3*P1-
3*P2 + P3) *t3);
  // Микропрофиль правая колея с задержкой база 5.6
  x_3= x - 5.6;
  n=u=x_3/ts;
  t=u-n;
  //t=fabs(u- n);

```

```

t2=t*t;
t3=t*t*t;
P0=rd[n-1];
P1=rd[n];
P2=rd[n+1];
P3=rd[n+2];
*y3 = 0.5*((2*P1) +(-P0 + P2) * t +(2*P0 - 5*P1 + 4*P2 - P3) *t2 +(-P0 + 3*P1-
3*P2 + P3) *t3);
}
else
*y1=*y2=*y3;
//-----Силы воздействующие на автомобиль
//float u_kp[16]={8.73,6.13,4.51,3.50,2.78,1.75,1.00, 0.795}; //КПП ЯМЗ-202
// Md=9549*Nd[0]/ndv; // v_a=20; //n_kp=7;
//-----
ndvs=103.26*v_a*u_kp[n_kp];//
ndvs=(30*va[0]*(u_kp*u_gp*u_rk))/(3.14*0.675);
if (ndvs>2200.0)
ndvs=2200.0;
if (ndvs<1000)
ndvs=950;
if (rt==0) // действует сила тяги с учетом КПД трансмиссии
{
Fk= (90827.07*Ndv*u_kp[n_kp])/ndvs; //9550*N*u_rk*u_kp*Um*kpd;//
//9550*N*1.107*6.59*u_kp[]/ndvs*rk // Fk= (M*u_kp*u_gp*kpd)/rk; Сила тяги
kz=Nd[0]/p_u;
*y4=kz;
}
else
{
Fk=-4000.0*v_a;
}
// Начало включения Режимы Торможения с учетом массива точек падения
скорости tb[j]
zl=xl-xtb; // разность положения и точки включения торможения
// условие включения торможение
if (zl>0) //
{
rt=1;
ndvs=1150.0;

```

```

Fk=-4000.0*v_a;
*y4=0.0;
n_kp=n_kp-3;
if (n_kp<1)
    n_kp=1;

// Fk=-20000.0; //заменяем Ft
vv=tb[j][1];
j++;
xtb=tb[j][0];
// vv=tb[j][1];
if (j>3)
    j=3;
}
//----- Отключение режима торможения, когда скорость упала меньше 5.0
if (rt==1&&v_a<vv)
{
    rt=0;
    vv=tb[j][1];
    Fk= (90827.07*Ndv*u_kp[n_kp])/ndvs;
    kz=Nd[0]/p_u;
    *y4=kz;
    if (j==8)
    {
rt=0;
    }
    }
if (xl>8500)
{
    rt=0;
    Fk= (90827.07*Ndv*u_kp[n_kp])/ndvs;
} //Конец отключения режима торможения
//----- Воздействие силы тяги, когда нет торможения
// Fv=3.997*v_a*v_a; //Fv=Kv*Aa*Va*Va; Сопротивление воздуха =
0.41202*9.7*v_a*v_a
Fy=Fy1+yn*gma;
jm=1.02599+0.01974*u_kp[n_kp]; //1+Jk/(ma*rk)^2+(Jen*kpd
//(u_kp*u_rk*u_gp)^2)/(ma*rk)^2 учет инерцион масс
//if (x<50)
a2=(Fk-Fv-Fy)/(ma*jm); // ускорение автомобиля

```

```

*sF=a2;
if (yn>0.0)
Fy_2=gma*yn*kpd;
else
Fy_2=0.0;
if (a2>0.0)
Fa2=ma*a2;
else
Fa2=0.0;
Nt=0.0010*(Fa2+Fy_2+Fy1+Fv)*v_a+Neq;
if (Nt>250.0)
    Nt=250.0;
*fx=Nt;
// *y4=Nt/p_u;
*y4=Nt;
/*
if (Fk<(Fy_2+Fy1+Fv))
    n_kp=n_kp-1;
    if (n_kp<1)
        n_kp=1;
*/
/*fx=((a2/(ma*jm)+Fv+Fy)*ndvs)/14148.15;
/*fx=a2;
//Nt=((a2*ma*jm +Fv+Fy)*ndvs)/14148.15; //rk/9550=14148.15
//Nt=((Fy+Fv)*ndvs)/14148.15;
//Nt=((a2*ma*jm)*ndvs)/14148.15;
// kz=Nd[0]/p_u;
// *y4=kz;
*y0=ch;
f_1=(Fk-Fv-Fy)*sin(ygk);
//f_1=0.4*(Fk-Fv-Fy)*sin(ygk)+0.6*(Fk-Fv-Fy);
f_2=2.0*cyx*ygk*cos(ygk);
if (ygk>=0.0)
{
*fy=0.4*f_1+f_2;
*fW= +f_2*11+0.4*f_1*1.3;
}
if (ygk<=0.0)
{
*fy=-0.4*f_1+f_2;

```

```

*fw= +f2*11-0.4*f1*1.3;
}
// -----Переключение КПП вверх
//if (a2>0.3&&ndvs>1400&&chm!=1)
if (a2>0.3&&ndvs>1800&&chm!=1)
{
nkp0=nkp;
nkp++;
if (nkp>8)
nkp=8;
else
ndvs=103.26*va*ukp[nkp0];
if (ndvs>2200.0)
ndvs=2200.0;
*nkp=nkp;
*ndv=ndvs;
chm=1;
m=0;
}
// -----Переключение КПП вниз
//if (Fk<(Fy2+Fy1+Fv))
if (a2<0.10&&ndvs<1250.0&&chm!=1)
{
nkp0=nkp;
nkp--;
if (nkp<1)
nkp=1;
else
ndvs=103.26*va*ukp[nkp0];
/*
if (ndvs>2200.0)
ndvs=2200.0;
if (ndvs<1150.0)
ndvs=1150.0;
*/
*nkp=nkp;
*ndv=ndvs;
chm=1;
m=0;
}

```



```

*nkp=n_kp;
*ndv=ndvs;
n_kp0=n_kp;
//-----Переход на нейтральную передачу
if (chm==1)
{
if (v_a>10.0)
m++;
Fv=3.997*v_a*v_a; //Fv=Kv*Aa*Va*Va; Сопротивление воздуха
0,41202*9,7 =3,997
Fy=Fy1+gma*yn;
// Fy=811.715+3639*yn; // 9.81*ma*(0.0039+0.0175 *yn);
Fw=9.81*ma*(f*cosa+sina); //Сопротивление качения и наклона
jm=1.02599; //1+Jk/(ma*rk)^2+(Jen*kpd (u_kp*u_rk*u_gp)^2)/(ma*rk)^2 учет
инерцион //масс
a2=1.0*(Fv+Fy)/(ma*jm); // ускорение автомобиля
*sF=a2;
// *nkp=n_kp;
ndvs=103.26*v_a*u_kp[n_kp0];
if (ndvs>2200.0)
ndvs=2200.0;
*ndv=ndvs;
if (a2<0.0)
{
chm=0;
m=0;
} //-----
if (m==500)
{
chm=0;
m=0;
}
} // -----Конец Переход на нейтральную передачу
/*
if(x>12400.0)
if(x<12600.0)
{
fp1=fopen("ldm.txt","at");
fprintf(fp1, "%.1f\n", x);
fclose(fp1);

```

```

}
*/
//-----Запись параметров движения для OpenCV
//if (chw==128000) //1063 записи
if (chw==10000) // записи
{
vd.dl=xl;
vd.v=v_a;
vd.zd=z2[0];
vd.spe=ndvs; // engine speed
vd.pwe=Ndv; // engine power
fp3=fopen("ma.bt","wb");
fwrite(&vd, 20,1, fp3);
fclose(fp3);
/*  fp3 = fopen("mar.txt", "at");
    fprintf(fp3, "dl=%ld v=%.1f zd=%.1f spe=%.1f Ndv=%.1f\n", vd.dl, vd.v,
vd.zd, vd.spe, vd.pwe); // reading file of a time delay
*/  fclose(fp3);
chw=0;
}
chw++;
//-----
h=(x-11898.0)*(nbl+1.0);
if (h>=0.0)
{
ch=0;
//j=0;
nbl++;
}
if (nbl==10)
ch=1;
//-----
//  SDV_Outputs_wrapper(u0, s0, md, va, Nd, Pu, thr, x1, x2, x3, x4, xa, vy, ya,
vw, wa, z1, z2, y0, sF, ndv, nkp, y4, y1, y2, y3, fx, fy, fw, fl, f2);
}
/* Function: mdlTerminate
 * Abstract: *   In this function, you should perform any actions that are necessary
 *   at the termination of a simulation. For example, if memory was *   allocated
in mdlStart, this is the place to free it.
*/

```

```
static void mdlTerminate(SimStruct *S)
{
}
#ifdef MATLAB_MEX_FILE /* Is this file being compiled as a MEX-file? */
#include "simulink.c" /* MEX-file interface mechanism */
#else
#include "cg_sfun.h" /* Code generation registration function */
#endif
```

ПРИЛОЖЕНИЕ Л

**Программа Matlab, генерирующая инициализационную
информацию по геометрии, динамике платформ Stewart**

```
% This MATLAB script generates the initial geometric, dynamic, and controller
% information for the platform in controls case study. The Stewart
% platform consists of a top plate, a bottom plate, and six legs connecting
% the top plate to the bottom plate. The overall system has six independent
% degrees of freedom. Each disconnected leg has six degrees of freedom and
% is composed of two bodies and one cylindrical and two universal joints.
%
% Copyright 2002-2005 The MathWorks, Inc.
```

```
% Angular conversion and CS axes
```

```
deg2rad = pi/180;
```

```
x_axis = [1 0 0];
```

```
y_axis = [0 1 0];
```

```
z_axis = [0 0 1];
```

```
null_length = [0 0 0 0 0 0];
```

```
% Connection points on base and top plate w.r.t. World CS at the center
% of the base plate
```

```
pos_base = [];
```

```
pos_top = [];
```

```
alpha_b = 2.5*deg2rad; % +/- offset angle from 120 degree spacing on base
```

```
alpha_t = 10*deg2rad; % +/- offset angle from 120 degree spacing on top
```

```
height = 2.0; % height in meters
```

```
radius_b = 3.0; % base radius in meters
```

```
radius_t = 1.0; % top radius in meters
```

```
for i = 1:3,
```

```
    % base points
```

```
    angle_m_b = (2*pi/3)* (i-1) - alpha_b;
```

```
    angle_p_b = (2*pi/3)* (i-1) + alpha_b;
```

```
    pos_base(2*i-1,:) = radius_b* [cos(angle_m_b), sin(angle_m_b), 0.0];
```

```
    pos_base(2*i,:) = radius_b* [cos(angle_p_b), sin(angle_p_b), 0.0];
```

```
% top points
```

```
% Top points are 60 degrees offset
```

```

angle_m_t = (2*pi/3)* (i-1) - alpha_t + 2*pi/6;
angle_p_t = (2*pi/3)* (i-1) + alpha_t + 2*pi/6;
pos_top(2*i-1,:) = radius_t* [cos(angle_m_t), sin(angle_m_t), height];
pos_top(2*i,:) = radius_t* [cos(angle_p_t), sin(angle_p_t), height];
end
% Permute pos_top points so that legs are end points of base and top points
% 6th point on top connects to 1st on bottom
pos_top = [pos_top(6,:); pos_top(1:5,:)]; %6th point on top connects to 1st on
bottom

% Compute points w.r.t. to the body frame in a 3x6 matrix
body_pts = pos_top' - height*[zeros(2,6);ones(1,6)];

% Leg vectors
legs = pos_top - pos_base;
leg_length = [ ];
leg_vectors = [ ];
for i = 1:6,
    leg_length(i) = norm(legs(i,:));
    leg_vectors(i,:) = legs(i,:) / leg_length(i);
end
% Calculate revolute and cylindrical axes
for i = 1:6,
    rev1(i,:) = cross(leg_vectors(i,:), z_axis);
    rev1(i,:) = rev1(i,:) / norm(rev1(i,:));
    rev2(i,:) = - cross(rev1(i,:), leg_vectors(i,:));
    rev2(i,:) = rev2(i,:) / norm(rev2(i,:));
    cyl1(i,:) = leg_vectors(i,:);
    rev3(i,:) = rev1(i,:);
    rev4(i,:) = rev2(i,:);
end

% Coordinate systems
lower_leg = struct('origin', [0 0 0], 'rotation', eye(3), 'end_point', [0 0 0]);
upper_leg = struct('origin', [0 0 0], 'rotation', eye(3), 'end_point', [0 0 0]);

for i = 1:6,
    lower_leg(i).origin = pos_base(i,:) + (3/8)*legs(i,:);
    lower_leg(i).end_point = pos_base(i,:) + (3/4)*legs(i,:);
    lower_leg(i).rotation = [rev1(i,:)', rev2(i,:)', cyl1(i,:)'];

```

```

upper_leg(i).origin = pos_base(i,:) + (1-3/8)*legs(i,:);
upper_leg(i).end_point = pos_base(i,:) + (1/4)*legs(i,:);
upper_leg(i).rotation = [rev1(i,:)', rev2(i,:)', cyl1(i,:)'];
end

```

```

% Inertias and masses

```

```

top_thickness = 0.05;
base_thickness = 0.05;
inner_radius = 0.03;
outer_radius = 0.05;
density = 76e3/9.81; % steel density in Kg/m^3

```

```

% Upper and lower leg inertia and mass

```

```

[lower_leg_mass, lower_leg_inertia] = inertiaCylinder(density, ...
    0.75*leg_length(1),outer_radius, inner_radius);
[upper_leg_mass, upper_leg_inertia] = inertiaCylinder(density, ...
    0.75*leg_length(1),outer_radius, 0);

```

```

% Top and base plate mass and inertia

```

```

[top_mass, top_inertia] = inertiaCylinder(density, ...
    top_thickness, radius_t, 0);
[base_mass, base_inertia] = inertiaCylinder(density, ...
    base_thickness, radius_b, 0);

```

```

% Reference motion, dynamical, and control constants

```

```

freq = pi; A = 100*freq; noise_freq = 200*freq/2/pi;
force_act_max = 3e5;

```

```

Ki = 1e4; Kp = 2e6; Kd = 4.5e4;

```

```

initCondI = (upper_leg_mass+lower_leg_mass+(top_mass*1.3/6))*9.81/Ki;

```

Блок-схемы моделирования движения и колебаний колесной машины

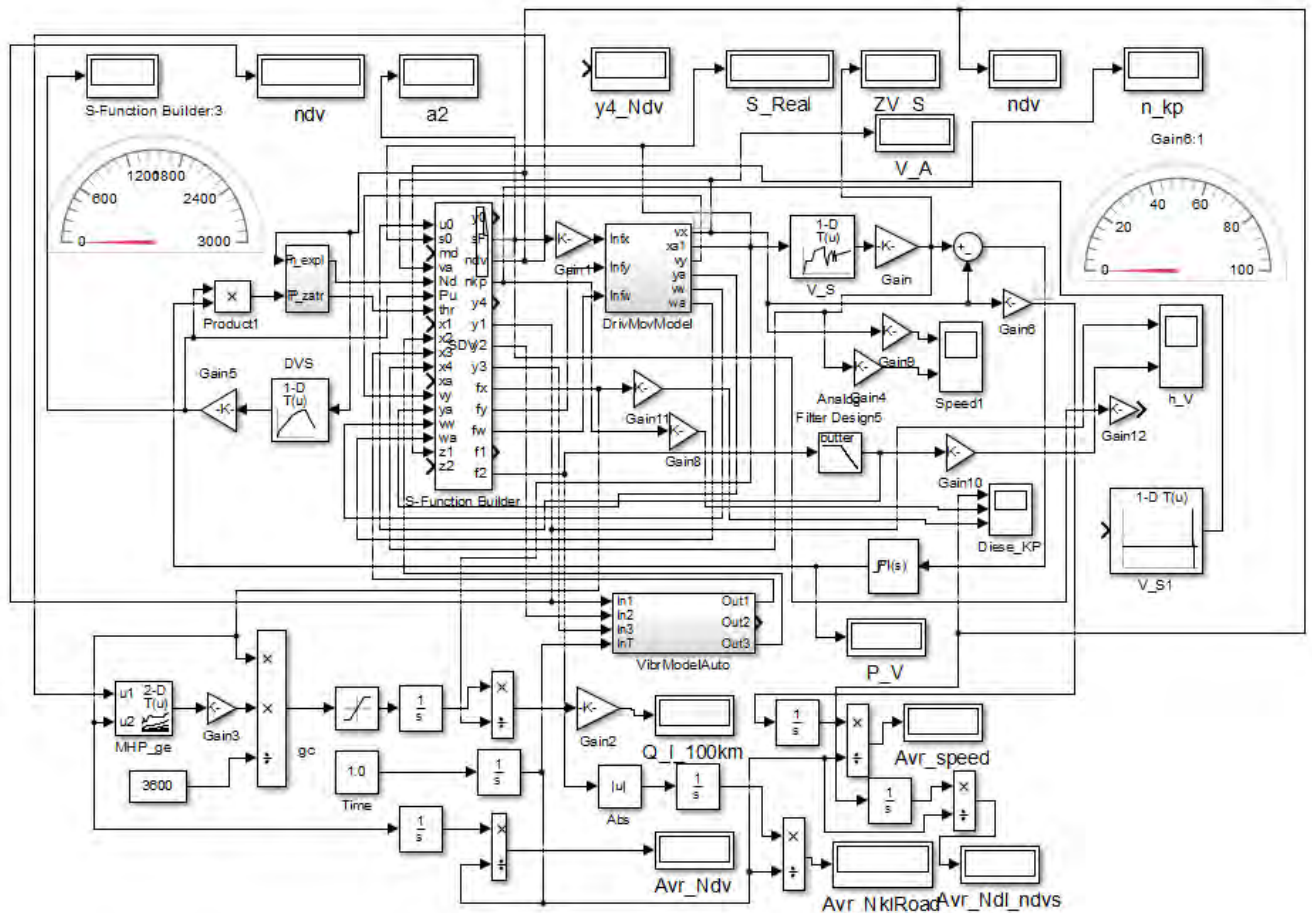


Рисунок М.1. – Общая блок-схема моделирования движения и колебаний колесной машины

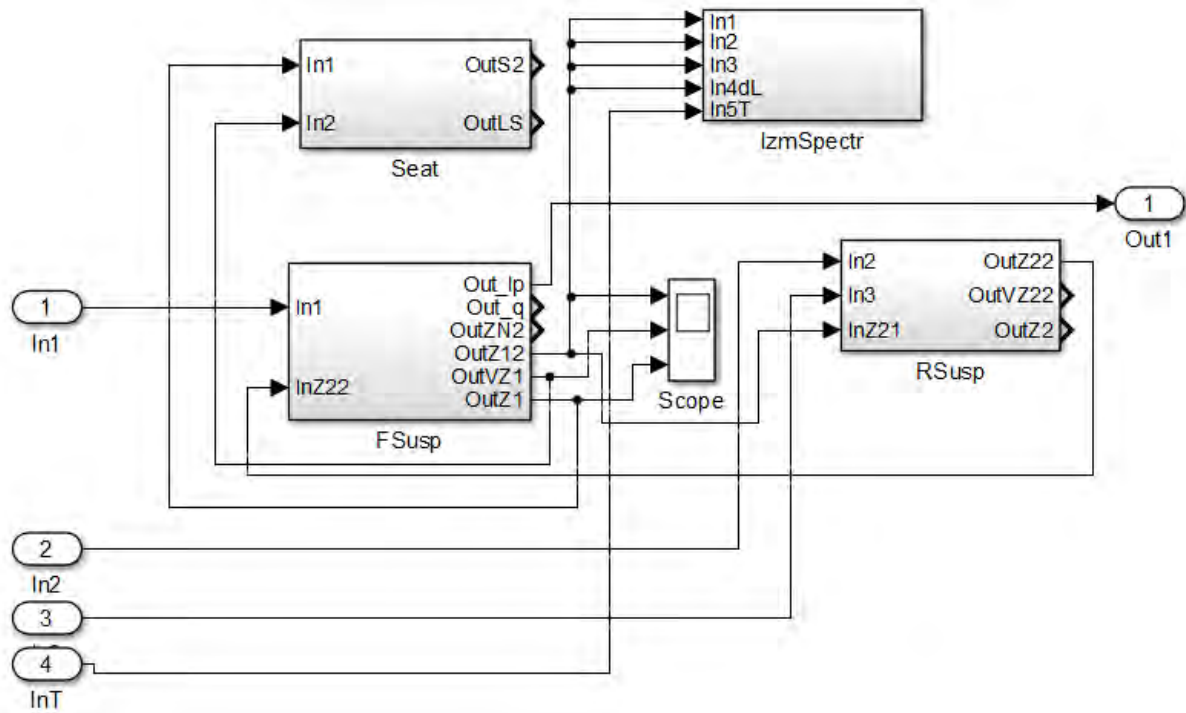


Рисунок М.2. – Общая блок-схема подрессоривания автомобиля

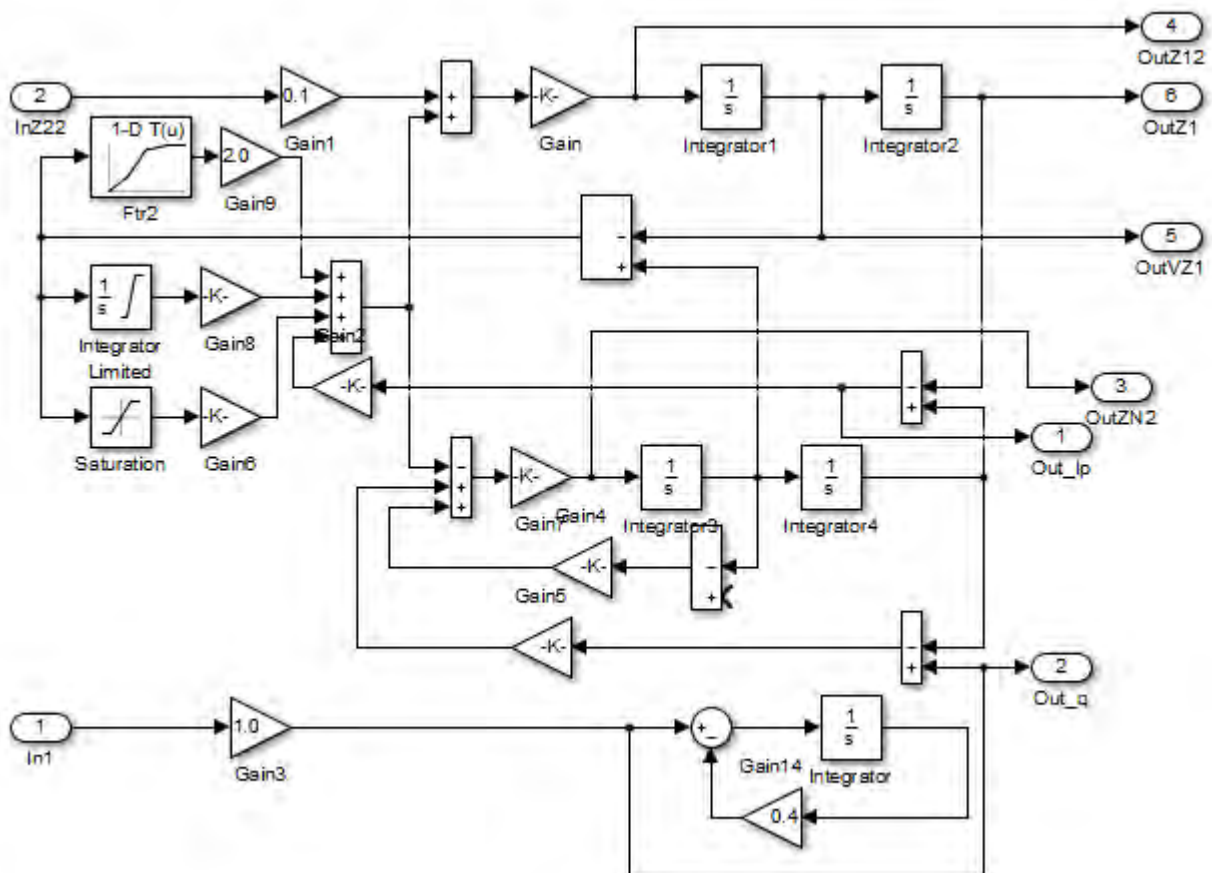


Рисунок М.3. – Блок-схема передней подвески автомобиля

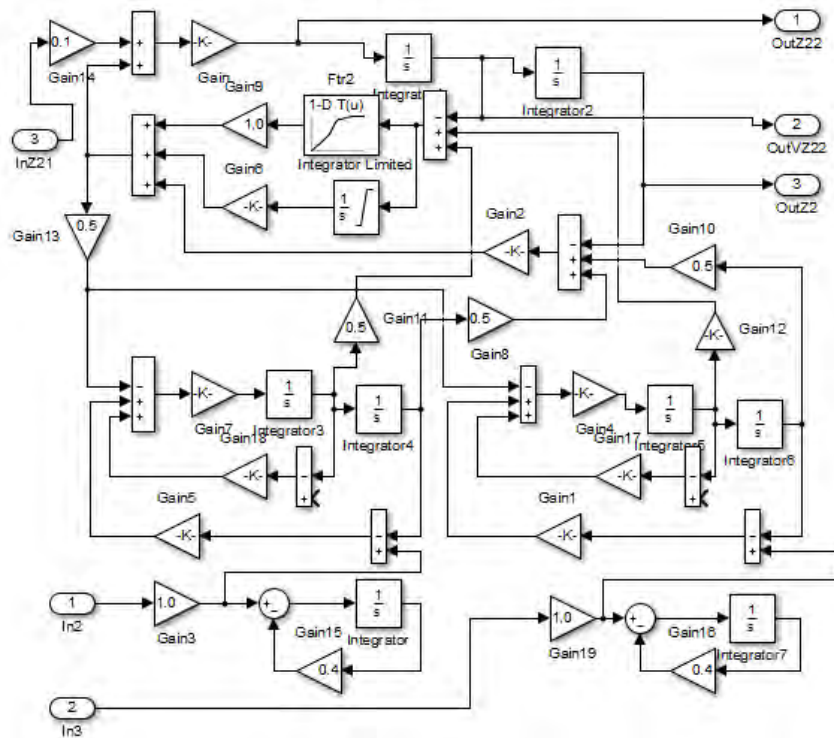


Рисунок М.4. – Блок-схема задней подвески автомобиля

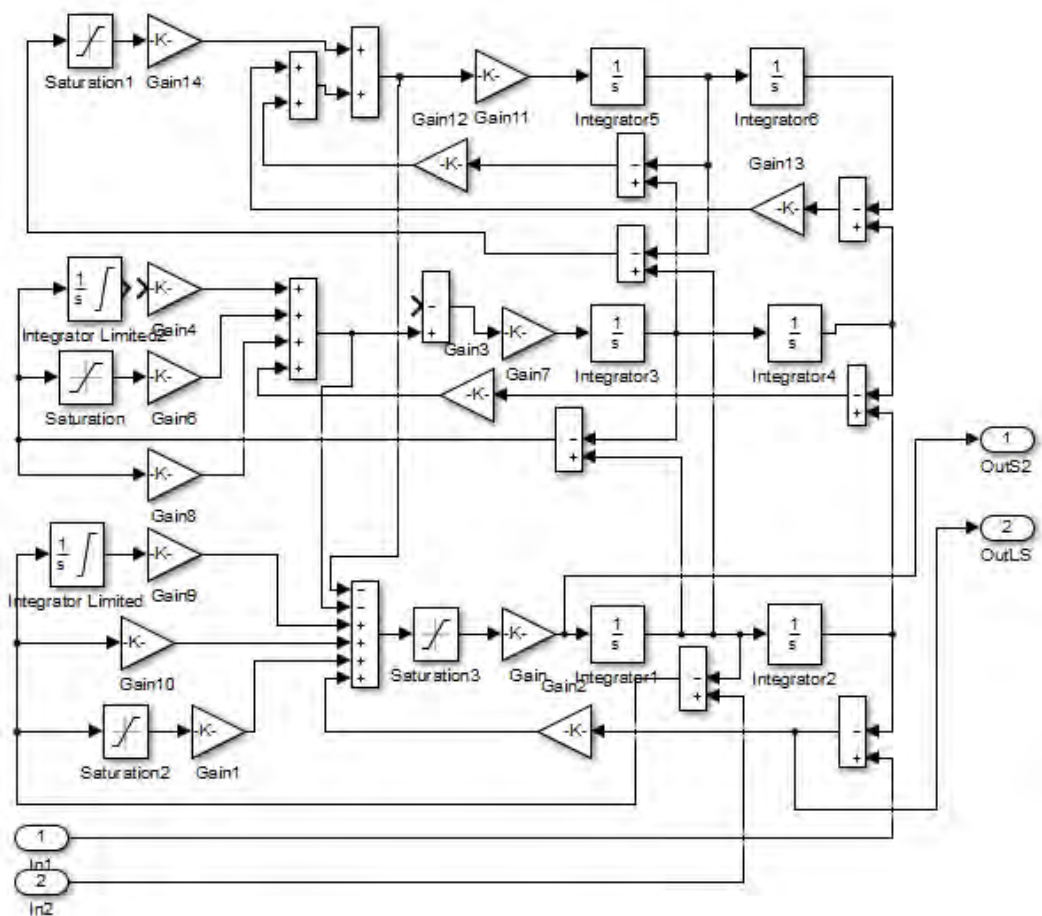


Рисунок М.5. – Блок-схема сиденья водителя

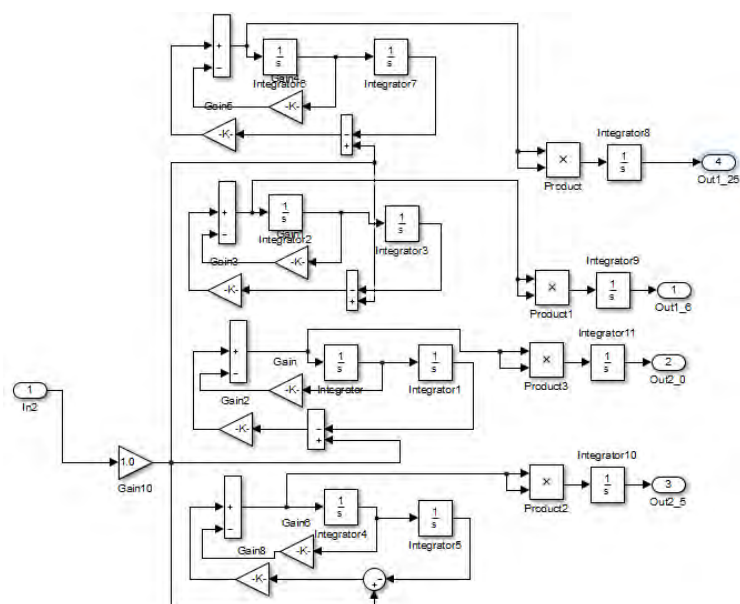


Рисунок М.7. – Блок-схема фильтра для третьоктавных полос частот

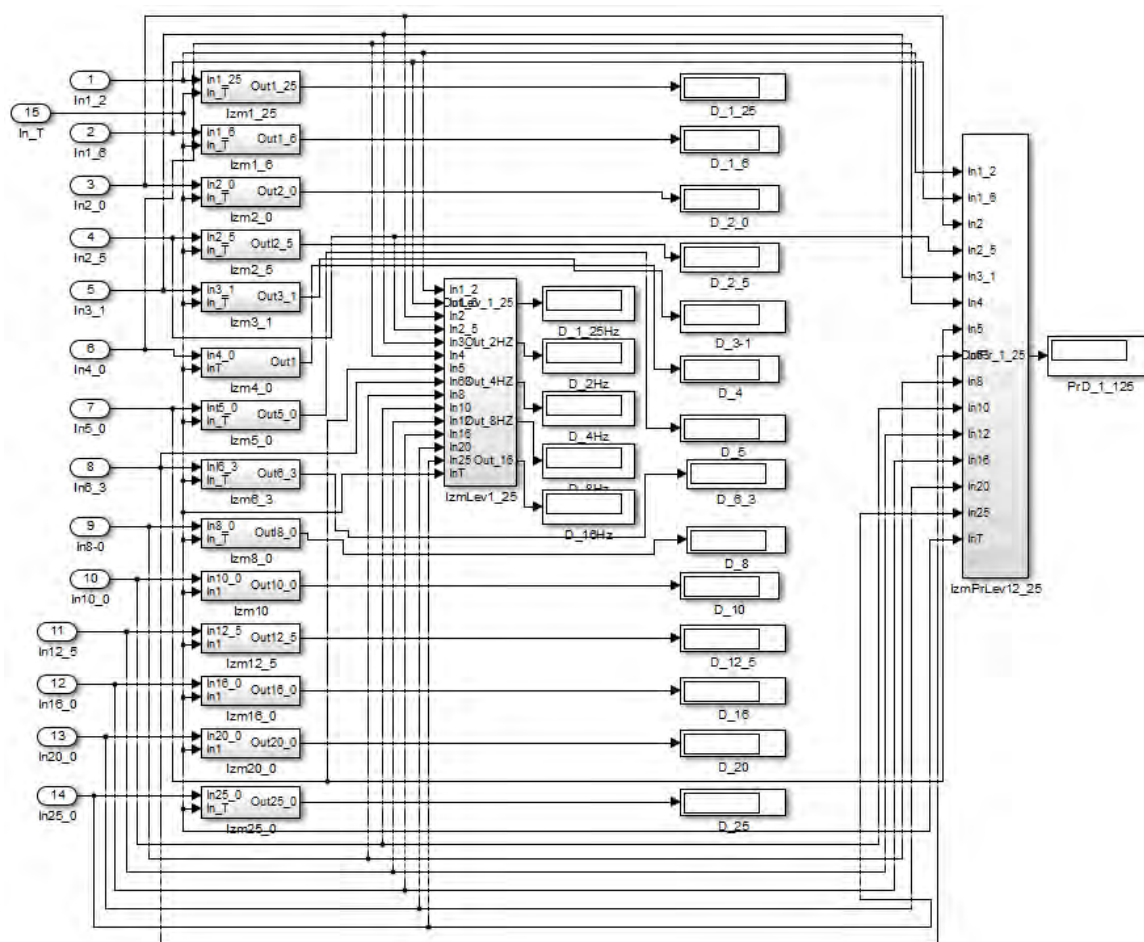


Рисунок М.8. – Блок-схема вычислений среднеквадратических значений в третьоктавных полос частот

Blueprint Class в Unreal Engine. Родительские и дочерние классы в UE5 (UE4)

Данный материал, рассматривающий базовые основы, понятия классов и компонентов в Unreal Engine, взят из <https://ueprosto.ru/blueprints/class.html?ysclid=mdtw812s3k617388792#sovettop>.

Blueprint Class в Unreal Engine

Blueprint Class в Unreal Engine или по «народному» просто блюпринты — это ассеты (внутренние файлы движка), позволяющие создавать новые классы Акторов или компонентов, используя визуальный скриптинг. Скрипты Blueprint в данном случае служат обёрткой для C++ классов, позволяя расширять их функциональность, настраивать поведение и логику, не требуя написания кода на C++.

Также Blueprint Class можно представить, как визуальный шаблон или чертеж, определяющий, как будет выглядеть и из каких компонентов состоять объект в игре, как он будет себя вести и взаимодействовать с окружающим миром и как будет отвечать на различные игровые события.

В игровом движке UnrealEngine заранее уже определены сотни готовых Blueprint Class'ов, со своими внутренними данными (составом, функционалом, свойствами и готовой логикой).

Все те блюпринт-классы, которые мы создаём, программируя игру в Unreal Engine, являются пользовательскими дочерними классами, которые позволяют легко добавлять новую функциональность, новые данные, а также логику, основанную на визуальных скриптах Blueprint к стандартным родительским классам движка. А после, позволяют на их основе создавать в игровом мире множественные объекты. Где каждый объект, содержащий в себе данные и Blueprint-логику, может при помощи этой логики изменять свои внутренние данные в ответ на игровые события.

Например, когда игрок нажимает на клавиатуре клавишу пробела, игровой объект, представляющий персонажа, может отреагировать на это игровое событие и изменить свои внутренние данные с помощью логики Blueprint. А именно, он может изменить данные анимации с покоя на прыжок, а также изменить свои данные координат в мире — переместиться вверх, а затем снова вниз, что симулирует прыжок.

Чтобы понять, что такое в целом объекты, а также родительские и дочерние классы, давайте немного углубимся в тему объектно-ориентированного программирования (ООП).

Объектно-ориентированное программирование (ООП) в Unreal Engine

Как и во многих популярных языках программирования, в игровом движке Unreal Engine применяются принципы ООП.

Объектно-ориентированное программирование в UnrealEngine — это подход к разработке игр, в котором элементы игрового мира представлены в виде объектов. Их данные, структура, свойства и поведение определяются классом, к которому они принадлежат.

То есть, все интерактивные объекты в игре, имеющие внутри себя определенную логику поведения, являются одним из экземпляров конкретного Blueprint Class.

Blueprint Class – это схема (чертёж) для создания объекта, имеющего в себе определенную логику. Схема содержит подробное описание своих будущих экземпляров в игровом мире. Определяет их поведение, структуру и состав данных.

Используя один класс, можно создать неограниченное количество экземпляров. По умолчанию они будут обладать одинаковыми данными, свойствами и методами, которые уже определены в их классе. Но значения этих свойств у каждого объекта будут разными.

То есть, любой Blueprint Class содержит набор обязательных элементов: данные, свойства и методы (функции), которые будут определять характеристики экземпляров этого класса и их поведение.

Например, возьмем класс «Транспорт». У него есть характеризующие их свойства (параметры):

тип;

класс;

способ передвижения;

цвет.

А также методы (её функции). То есть то, что она умеет делать:

ездить;

преодолевать препятствия;

передвигаться по воде.

летать.

Соответственно, имея один общий Blueprint Class «Транспорт», в Unreal Engine мы можем создать множество различных объектов машин. И поместить этих конкретные машины в игровой мир.

То же самое и с функциями. У всех экземпляров класса «Транспорт» будет функция «Ездить». Однако каждая конкретная машина будет ездить, то

есть исполнять свою функцию, по-разному: одна быстро, другая медленно, а третья, возможно, будет летать.

Принципы наследования

Все `Blueprint Class` в `UnrealEngine` поддерживают иерархию наследования.

Наследование — это один из самых мощных и важных инструментов в ООП. С его помощью можно создавать новый дочерний объект или класс на основе уже существующего родительского, наследуя все его данные, методы и свойства. При этом можно использовать как весь унаследованный функционал, так и частично, переопределив или добавив новый.

То есть, в дочерних классах можно:

- добавлять новые данные, свойства и методы;
- заменять алгоритмы методов родительского класса.

Но, важный момент, в дочерних `Blueprint Class` нельзя удалять родительские данные и функционал. Заменять можно, а удалять нет.

В итоге, наследование значительно упрощает разработку игр в `Unreal Engine`, так как позволяет избежать повторного написания одного и того же кода.

В качестве примера наследования рассмотрим транспортные средства: автомобиль и самолёт. И автомобиль, и самолёт могут перевозить людей и грузы. Однако способы их передвижения различаются. Таким образом, у них есть как общие, так и отличительные характеристики.

Соответственно, мы можем определить один общий родительский `Blueprint Class` «Транспорт», у которого будут общие свойства:

- цвет;
- количество пассажиров;
- максимальная грузоподъемность.

Далее, на основе родительского блюпринта «Транспорт» можно создать два дочерних: «Машина» и «Самолет». В этих классах мы можем описать методы, которые будут уникальными для каждого из них. Например, для класса «Машина» можно определить метод «Ездить», а для самолета — функцию «Летать». А вот свойства (цвет, количество пассажиров, максимальная грузоподъемность) уже определять не нужно. Всё это просто унаследуется от родительского класса «Транспорт». Единственное, значения у свойств будут разными. Для машины нужно задать одни значения свойств, для самолета другие.

Такой подход позволяет избегать дублирования кода в родительских и дочерних блюпринтах. А также даёт возможность легко дополнять новыми

общими данными, свойствами и методами все объекты, просто прописав код в родительском классе.

Принципы инкапсуляции в Blueprint Class

Также в Blueprint Class реализуется принцип инкапсуляции (Encapsulate — «поместить в капсулу, изолировать»).

Согласно этому принципу ООП, вся важная для существования объекта информация хранится внутри него самого. При этом весь внутренний функционал, свойства и логика скрыты от внешнего мира. И только лишь некоторые функции доступны для взаимодействия с другими объектами игры, в рамках специализированного инструмента Blueprint Interface.

Инкапсуляция — это механизм, который обеспечивает высокий уровень безопасности и снижает вероятность случайного повреждения данных внутри класса или объекта.

То есть, если в проекте на Unreal Engine имеются Blueprint Class «Машина» и «Игрок», то их объекты в игровом мире не должны знать друг о друге абсолютно ничего. Как игрок не должен знать, при помощи каких функций машина передвигается, как она открывает двери. Так и машина не должна знать, как устроены функции ходьбы у игрока и с помощью каких методов игрок садится в машину. А для того чтобы эти классы могли взаимодействовать друг с другом, когда игроку нужно сесть в машину, открыв её дверь, используется уже специальный инструмент Blueprint Interface.

Все принципы ООП характерны не только для Blueprint Class, но и для любых объектов в Unreal Engine.

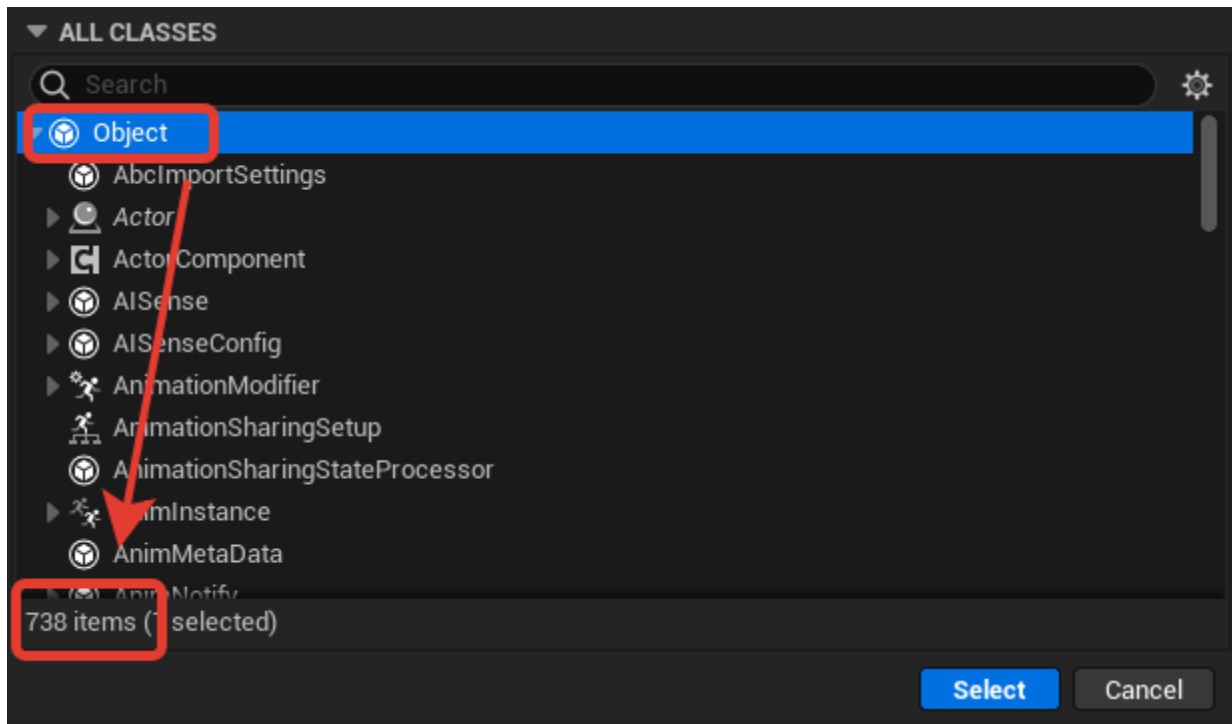
Например, принципам наследования соответствуют материалы в движке или компоненты в блюпринтах. А для функций характерна инкапсуляция. Так как для разработчика не важна внутренняя формула функции, а важно то, что она делает, важен результат ее работы.

Иначе говоря, ООП — это основа всего игрового движка.

Основные родительские Blueprint Class, встроенные в Unreal Engine

Для удобства разработчиков и экономии их времени, в Unreal Engine по умолчанию уже определены множество различных родительских классов с заранее подготовленной функциональностью. От которых создаваемый нами Blueprint Class наследует все основные характеристики и функции. Давайте рассмотрим самые основные из них.

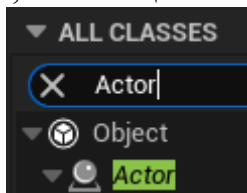
Object — самым базовым родительским классом в Unreal Engine (UE4, UE5) является класс Object.



Blueprint class Object. Именно от него стартует вся иерархия наследования в движке. Он является родительским для 737 других дочерних классов. Экземпляры Object невозможно физически разместить на игровом уровне. А вся функциональность Object сводится к инициализации и базовой подготовке наследуемого объекта:

- инициализация свойств;
- приведение типов;
- сетевая репликация;
- подсчет ссылок;
- сборка «мусора» и очистка памяти,
- и прочая подготовка.

Actor. Blueprint Class ActorActor (Актор) — класс, наследуемый от Object, являющийся основным строительным блоком в Unreal Engine.



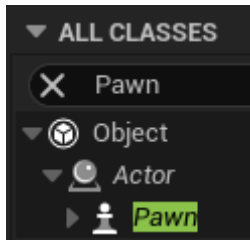
С объектами Actor можно взаимодействовать. Актор создаёт объект, который можно поместить в игровой мир и хранит в себе набор базовых свойств:

- Location — координаты в мире, Rotation — поворот, Scale — масштаб;
- наличие компонентов;
- функции репликации;

- методы жизненного цикла (Spawn, Tick, Destroy);
- наличие событий (Events) для реакции на внешний мир.

Класс Actor используется чаще остальных. Он подходит для создания неодушевленных объектов в игровом мире, которые наделены определенной логикой: открываемая дверь, зажигающаяся лампочка, подбираемый предмет и прочее.

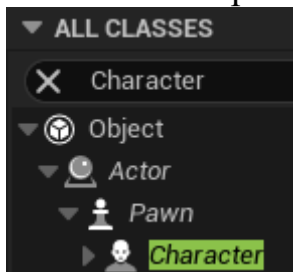
Pawn.



Класс Pawn (павн, пон, пешка) — класс, наследуемый от Actor. Объектами этого класса может «овладевать» (Possess) контроллер, получая входные данные (Input) и управляя движением Pawn'а. Управление достигается либо посредством искусственного интеллекта (AI Controller), либо же самим игроком (Player Controller). То есть, основное отличие от Actor — это возможность управления объектом Pawn через контроллер.

Данный Blueprint Class подходит для реализации управляемых объектов: машины, лодки, самолеты, животные, враги и так далее.

Character Blueprint Class

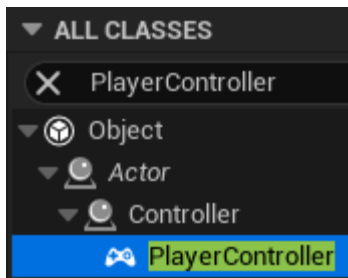


Иерархия наследования класса **Character** (персонаж) — класс, наследуемый от Pawn. Обычно, с помощью него создают главного героя, которым управляет сам игрок.

Класс Character содержит в себе специализированные компоненты (данные), облегчающие создание человеческих или других «двуногих» персонажей:

- Capsule Component (капсульная коллизия) — перекрытие и столкновение персонажа с другими объектами в мире;
- Skeletal Mesh Component (скелетный Меш) — 3D-модель персонажа и настройка правил его анимации;
- Character Movement — функциональность движения персонажа (ходьба, бег, прыжки, приседание, плавание, гравитация, сопротивление).

Player Controller



Иерархия класса `PlayerController` `Player Controller` (контроллер игрока) — `Blueprint Class`, наследуемый от класса `Controller`, который, в свою очередь, наследуется от родительского `Actor`.

Контроллер игрока управляет объектами `Pawn` или `Character`, передавая им от игрока (от клавиатуры, мышки, джойстика) входящие данные для движения (`Input Action`).

Иначе говоря, это своеобразный интерфейс между `Pawn` (`Character`) и игроком, управляющим этими объектами. По сути, представляет волю игрока.

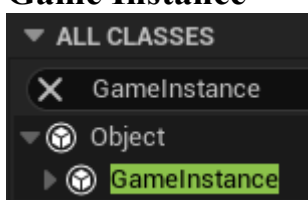
Game Mode

Иерархия класса `GameMode` `Game Mode Base` (`GameMode`, игровой режим) — это `Blueprint Class`, наследуемый от родительского класса `Info`, который в свою очередь, наследуется от `Actor`. Он может устанавливаться как отдельно для каждого уровня, так и для всего проекта `Unreal Engine` в целом.

Игровой режим — это набор правил и условий игры, по которым она проходит. Он может включать в себя:

- порядок присоединения игроков к игре;
- приостановку игры;
- переход на другой уровень;
- правила подсчёта очков;
- контрольные точки, которые определяют состояние игрового процесса;
- и другие правила игрового мира.

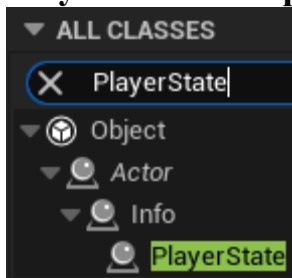
Game Instance



Blueprint class `GameInstanceGame Instance` (экземпляр игры) наследуется от родительского класса `Object` и содержит информацию о текущем сеансе игры. Экземпляр этого класса создается при запуске игры и существует до тех пор, пока игра не будет завершена.

`Game Instance` — это единственный `Blueprint Class`, который существует на протяжении всей игры, независимо от загрузки и выгрузки игровых уровней. Соответственно, в нем можно сохранять любые данные, которые нужно передать между разными игровыми уровнями.

Player State Blueprint Class



Иерархия класса `Player StatePlayer State` (состояние игрока) — класс, наследуемый от родительского класса `Info`, который в свою очередь, наследуется от `Actor`.

В сетевой версии игры экземпляры `Player State` реплицируются и содержат всю необходимую информацию об игроке, относящуюся к сетевой игре.

В целом может содержать любую информацию о состоянии игрока или бота:

- имя игрока;
- количество жизней;
- количество очков;
- состояние здоровья;
- информацию об инвентаре.

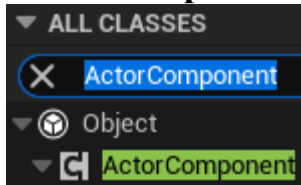
Game State

Иерархия блюпринт класса `Game StateGame State Base` (`GameState`, состояние игры) — `Blueprint Class`, наследуемый от родительского класса `Info`, который в свою очередь, наследуется от `Actor`.

В основном `Game State` используется в сетевых играх. Он содержит в себе информацию о подключенных игроках, их очках и других параметрах. И существует как на сервере, так и на каждом отдельном клиенте, постоянно

синхронизируясь между всеми участниками игры, чтобы все игроки имели актуальную информацию о текущей игре.

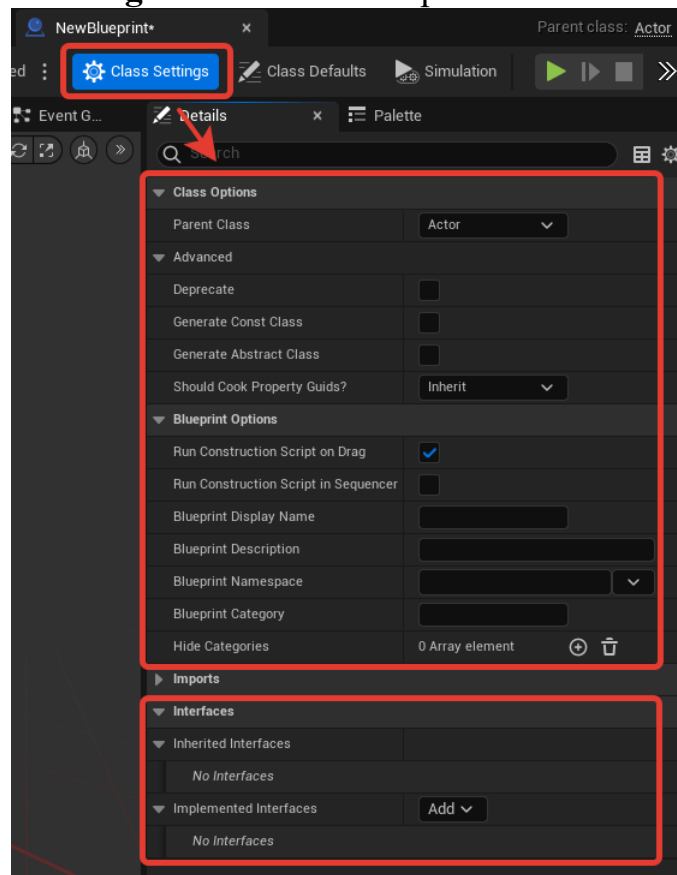
Actor Component



Unreal Engine Actor Component Actor Component (Актор компонент) — Blueprint Class, наследуемый от родительского класса Object. Является базовым классом для всех компонентов, которые могут быть присоединены к Actor. Он предоставляет фундаментальные функции и свойства, необходимые для работы компонентов в рамках движка UnrealEngine.

Actor Component используется для создания пользовательских компонентов и предназначен только для расчета логики без использования визуализации и трансформации. Применяется он исключительно в тех случаях, когда необходимо рассчитать и обработать какие-либо абстрактные действия, такие как перемещение, управление инвентарём, изменение свойств и атрибутов объектов, а также при реализации других нефизических концепций.

Blueprint Class Settings — базовые настройки класса



Меню Class Settings открывает базовые настройки блюпринт-классов во вкладке Details, где можно переназначить родительский класс, добавить блюпринт-интерфейс или изменить другие свойства, характерные для конкретного типа класса. Ниже мы разберем основные настройки, общие для всех типов блюпринтов.

Blueprint Class Settings

Blueprint Class Settings: основные настройки, общие для всех типов блюпринтов

Class Options:

Parent Class — по умолчанию указывает наименование родительского класса. В этом же пункте можно переназначить родительский класс, выбрав любой другой из списка.

Class Options / Advanced:

Deprecate (bool, по умолчанию False) — помечает текущий класс устаревшим. Deprecate позволяет предотвратить случайное размещение Blueprint и всех его дочерних объектов в игровом мире. Это удобно, если есть необходимость сохранить программный код в проекте для ознакомления или архивирования, но при этом нужно исключить возможность его случайного запуска в игровом процессе, что поможет избежать нежелательных ошибок.

Generate Const Class (bool, по умолчанию False) — помечает текущий класс константным.

Generate Abstract Class (bool, по умолчанию False) — изменяет класс на абстрактный. В результате чего текущий Blueprint становится непригодным для размещения в игровом мире, так как абстрактные классы предназначены исключительно для наследования логики.

Should Cook Property Guids? (по умолчанию Inherit) — включает свойства Guid в сборку созданного класса, которая будет конвертирована в формат платформы. Эта опция может привести к увеличению потребления памяти в преобразованной сборке, однако она позволит избежать необходимости добавления данных Core Redirect для классов Blueprint, хранящихся в архивах Save Game. Возможные варианты выбора:

- No (нет, не конвертировать);
- Yes (да, конвертировать);
- Inherit (наследовать от родительского класса, если родительского класса нет, то тогда возвращается значение No).

Blueprint Option:

Run Construction Script on Drag (bool, по умолчанию True) — перезапускать функцию Construction Script не только при перетаскивании объекта на сцену, но и при его перемещении в игровом мире. Если установлено значение False, Construction Script будет активироваться только после того, как Вы отпустите правую кнопку мыши.

Run Construction Script in Sequencer (bool, по умолчанию False) — перезапускать функцию Construction Script непрерывно в Sequencer.

Blueprint Display Name (по умолчанию пусто) — позволяет изменить имя, которое отображается для Blueprint в интерфейсе редактора. При этом название самого ассета в Content Drawer останется прежним, а во всех остальных меню и нодах будет отображаться новое переопределенное имя.

Blueprint Description (по умолчанию пусто) — позволяет задать пользовательское описание текущего ассета. Описание отображается в подсказке Content Drawer при наведении курсора мыши на файл блюпринта.

Blueprint Namespace (по умолчанию пусто) — позволяет задать пространство имен для текущего Blueprint class.

Blueprint Category (по умолчанию пусто) — позволяет задать категорию текущего класса.

Hide Categories — позволяет задать массив скрытых категорий.

Interfaces:

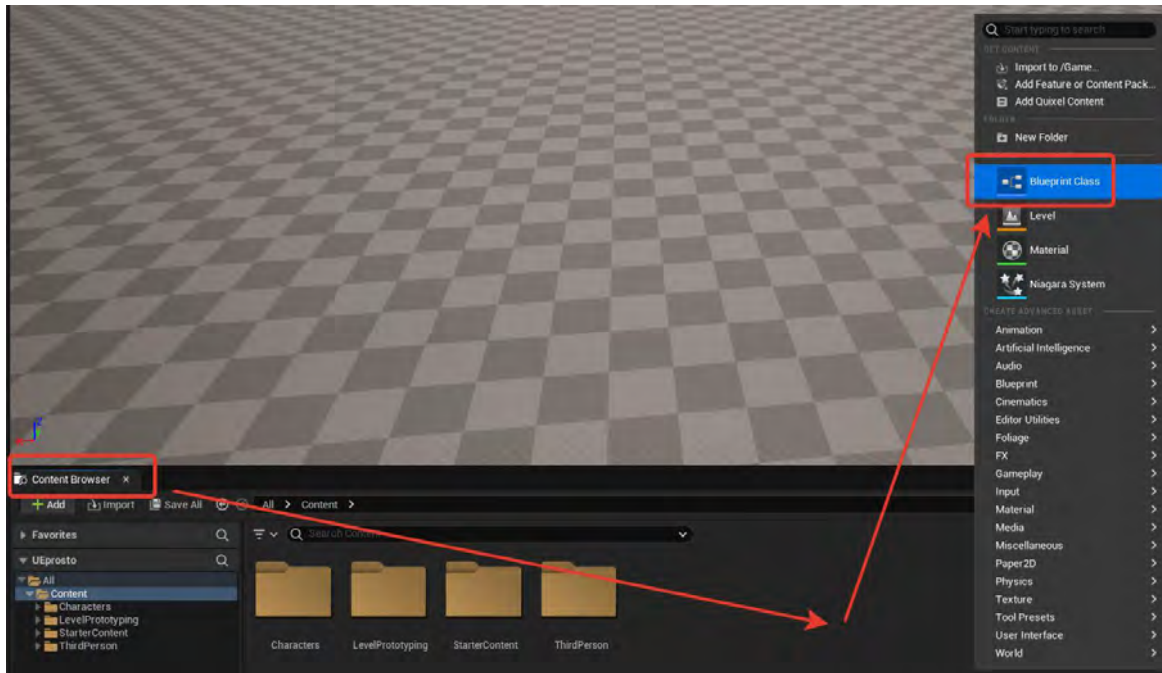
Inherited Interfaces — отображает список унаследованных интерфейсов от родительского класса.

Implemented Interfaces — добавляет Blueprint Interfaces к текущему классу.

Как создать Blueprint Class

Создать Blueprint Class можно несколькими способами: через контекстное меню в Content Browser и непосредственно во вьюпорте игрового мира. Рассмотрим эти варианты подробнее.

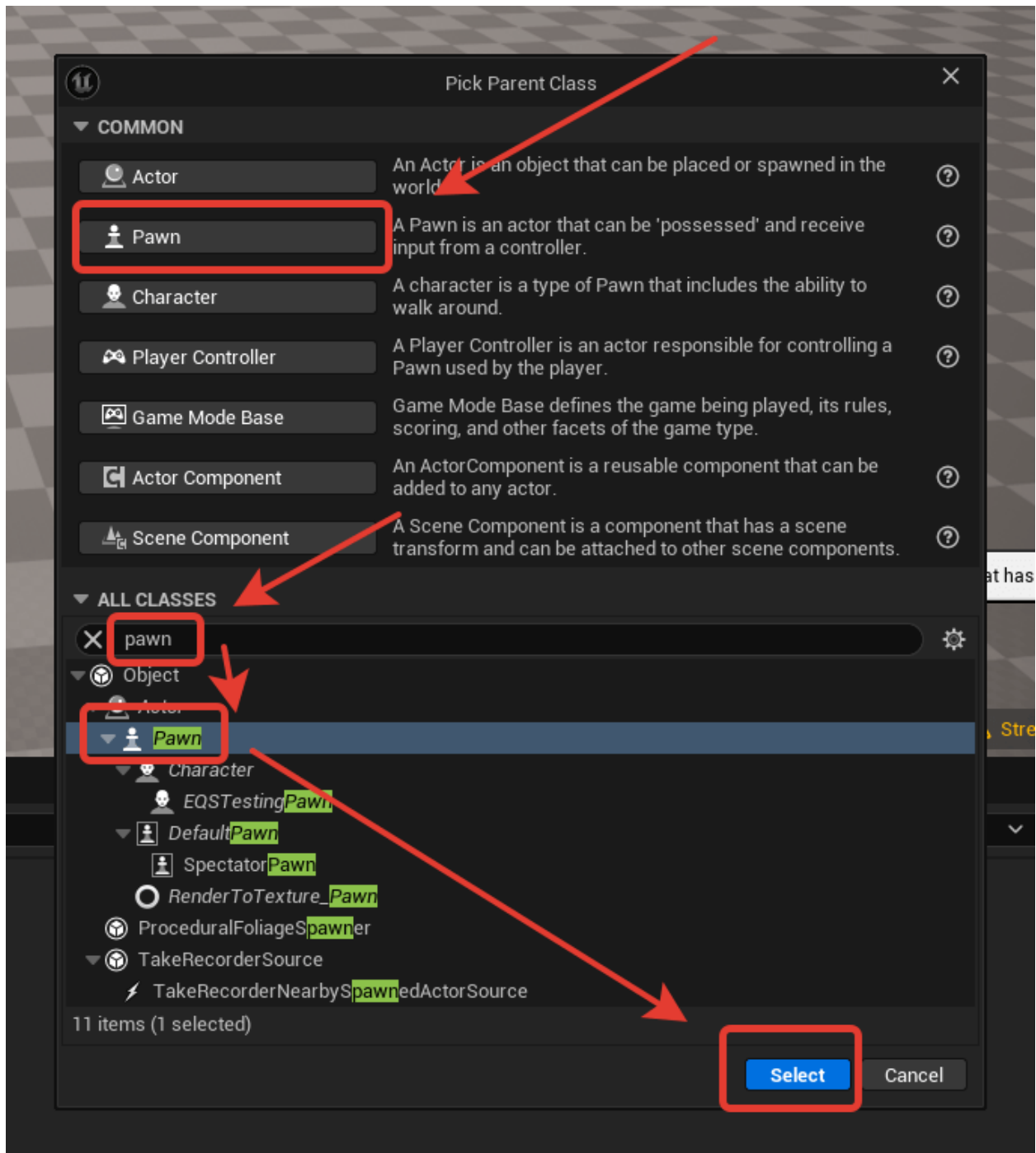
Content Browser



В текущей папке Content Browser щелкните правой кнопкой мыши по пустому полю. Откроется контекстное меню. В нем выберите подменю Blueprint Class.

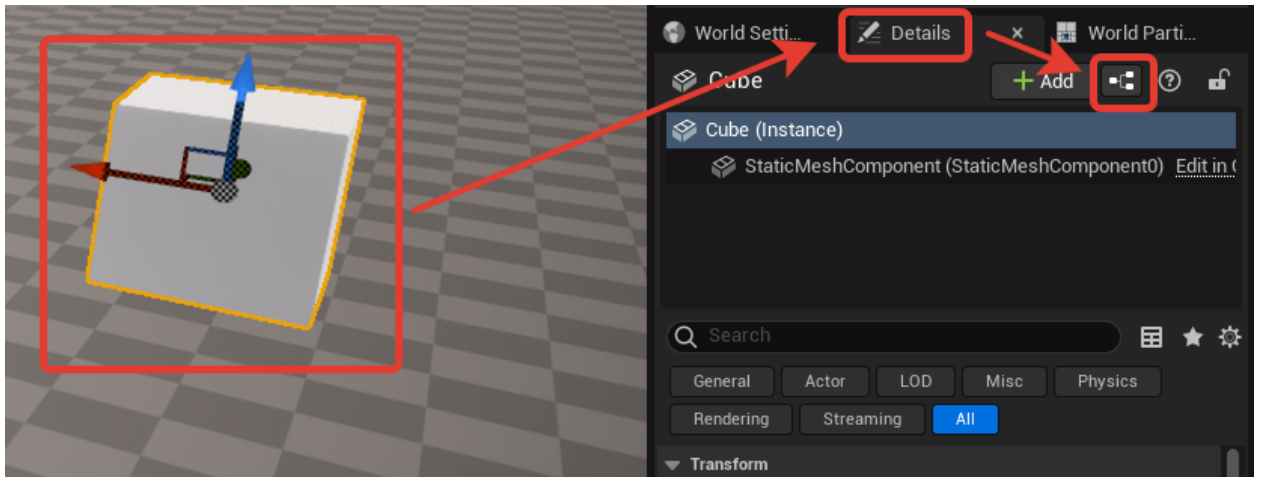
Контекстное меню создания Blueprint Class в Unreal Engine

Далее в открывшемся окне Pick Parent Class (выбор родительского класса) выберите нужный родительский блюпринт, на основе которого Вы хотите создать свой. Можно выбрать его из верхнего списка популярных классов или воспользоваться формой поиска. Далее нажмите кнопку Select (выбрать), после чего Blueprint Class будет создан в текущей папке Content Browser.



Выбор необходимого родительского блюпринт класса

Выбор необходимого родительского блюпринт класса
Viewport игрового мира

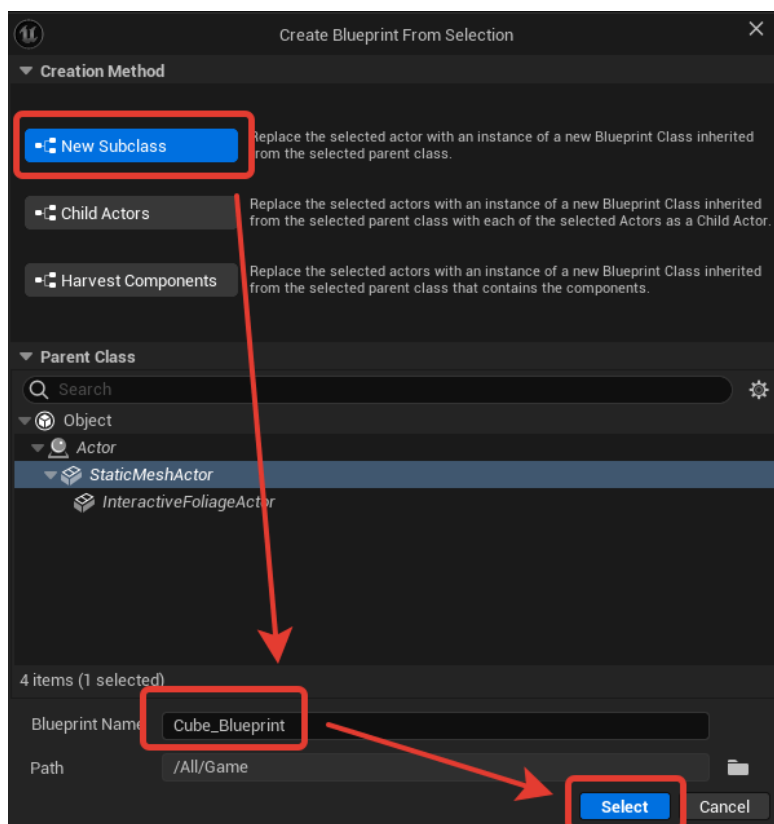


Если, находясь в игровом мире, Вам нужно добавить определённую логику для простого Actor, то данный объект необходимо изменить на Blueprint Class. Чтобы это сделать, выберите в игровом мире нужный объект. Затем в разделе Details справа нажмите на кнопку Convert Actor.

Выберите необходимый объект и нажмите на кнопку Convert Actor

Выберите необходимый объект и нажмите на кнопку Convert Actor

В открывшемся окне Create Blueprint From Selection (создать блюпринт из выделенного) выберите кнопку New Subclass (новый подкласс). Затем введите нужное название для Вашего блюпринта и нажмите на кнопку Select (выбрать).

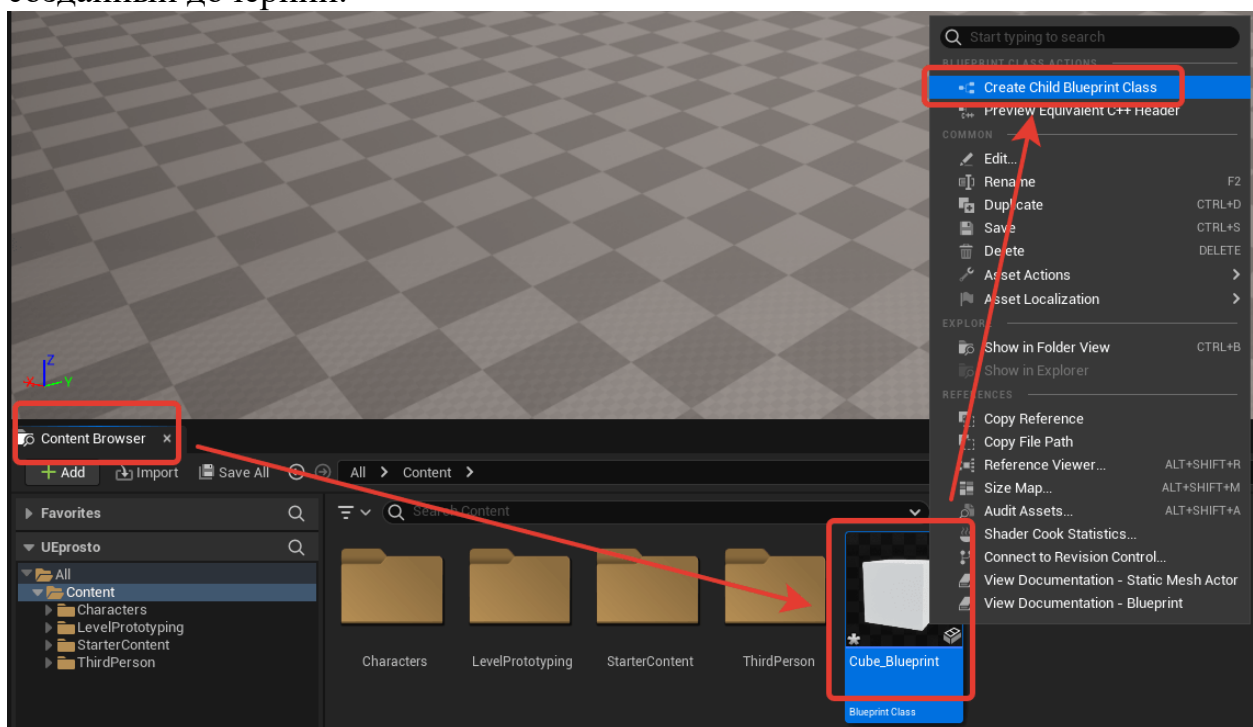


Окно Create Blueprint From Selection

После этого Actor на сцене будет заменен на Ваш созданный блюпринт. А сам файл ассета блюпринта будет находиться в текущей папке Content Browser.

Как в Unreal Engine создать дочерний Blueprint Class

Чтобы создать дочерний блюпринт на основе уже существующего, откройте Content Browser и выберите нужный родительский класс. Затем, нажав на нём правую кнопку мыши, вызовите контекстное меню и выберите пункт Create Child Blueprint Class (создать дочерний блюпринт класс). После этого в Content Browser рядом с родительским блюпринтом появится и вновь созданный дочерний.



Создать дочерний Blueprint Class в Unreal Engine

Меню «Создать дочерний Blueprint Class в Unreal Engine»

Руководства по настройке модуля транспортного средства Chaos Modular Vehicle System для UE 5.6

Ниже приведенный материал представляет перевод электронной документации epicgames.com с английского с некоторыми пояснениями. Источник <https://dev.epicgames.com/documentation/en-us/unreal-engine/chaos-modular-vehicles-quickstart>

. **Chaos Modular Vehicles** - новый (тестируемый) плагин в UE 5, где каждая часть транспортного средства (двигатель, кузов, трансмиссия, сцепление и т.д.) идут отдельным настраиваемым модулем.

Данное руководство поясняет как настроить модуль транспортного средства для моделирования в физическом решателе Chaos.

Отдельные Assets, составляющие модульное транспортное средство, следующие:

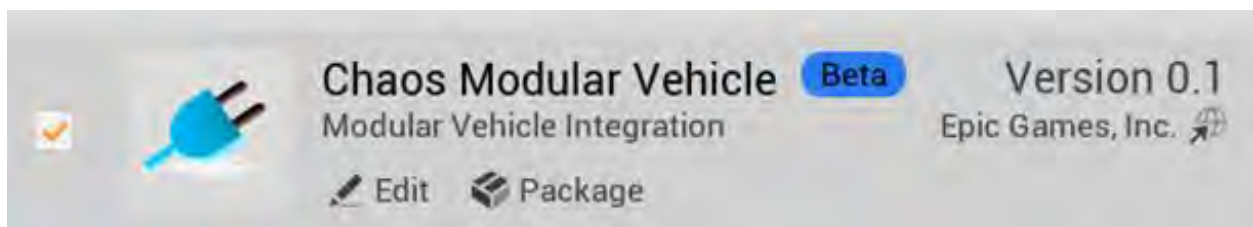
- Assets геометрической коллекции (для частей транспортного средства - произведенный от статических сеток / **Mesh**)
- A vehicle Blueprint / визуальный проект транспортного средства.

Assets – это компоненты, которые представляют собой графику, звуковое сопровождение или скрипты. Они прикрепляются к объектам и составляют важную часть игры.

Mesh - это сетка полигонов, из которых состоит любой 3D объект в компьютерной графике. Полигон, в свою очередь, состоит из вершин (вертексов), рёбер (линий, соединяющих две вершины между собой) и граней.

Подключение к Chaos Plugin модулю транспортного средства

Нажмите **Edit> Plugins** и найдите «Chaos modular vehicle/модуль транспортного средства». Проверьте Chaos на доступность версии Chaos Modular Vehicle plugin в вашем проекте

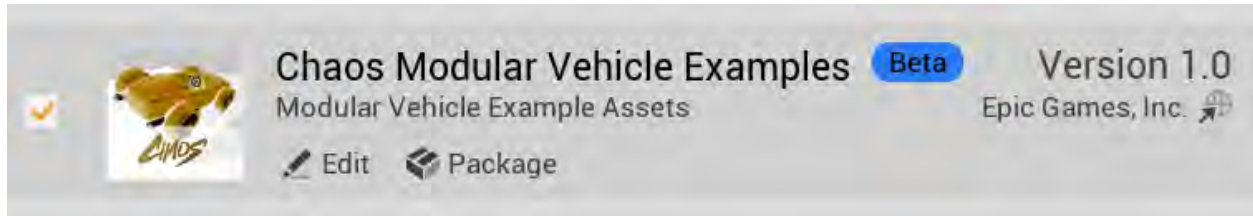


Это - тестируемый бета плагин транспортного средства Chaos. Вам не требуется оба.

Но все же дополнительно включите Chaos модульный плагин транспортного средства в качестве пр Assets

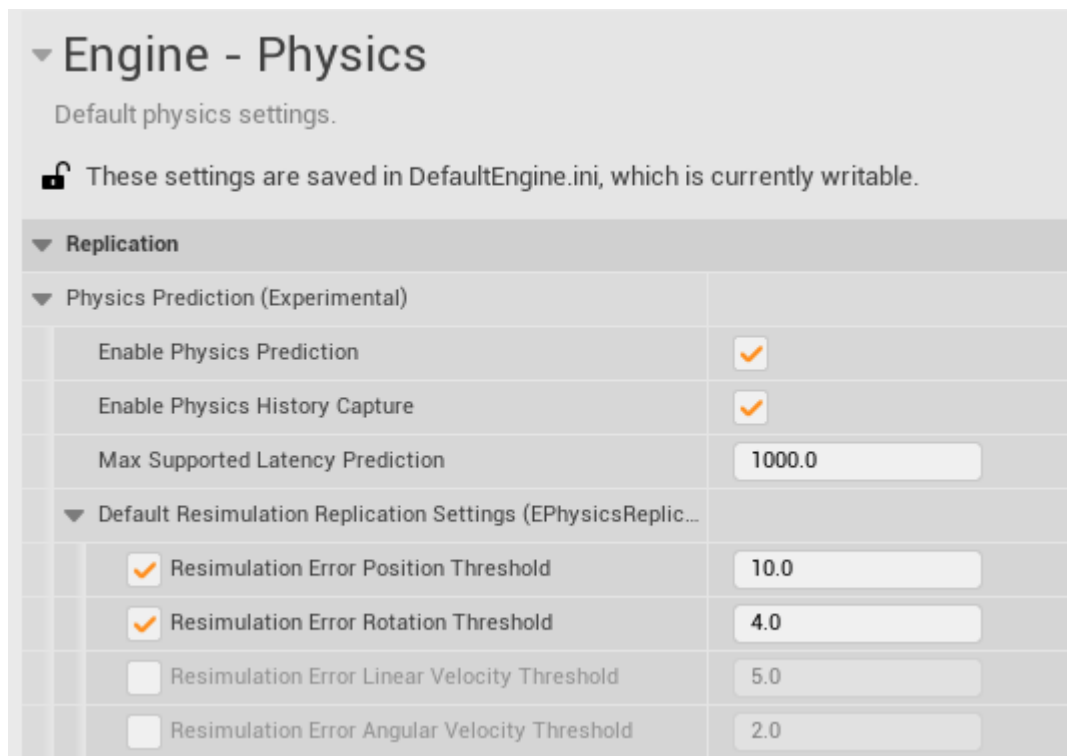
имера.

Щелкните по **Edit> Plugins** и найдите "chaos modular vehicle examples" / Chaos модульные примеры транспортного средства". Проверьте, что плагин Chaos Modular Vehicle Examples включен в ваш проект.



Подключение к сетевым настройкам

В Ваших настройках проекта под **Engine > Physics**, щелкните по **Enable Physics Prediction** и **Enable Physics History Capture**. Это обеспечивает преимущество в сети с сервером для авторизованного клиента с проведением возможности перемоделирования. Транспортное средство будет понимать, что им локально управляют без задержки от исходных данных управления.

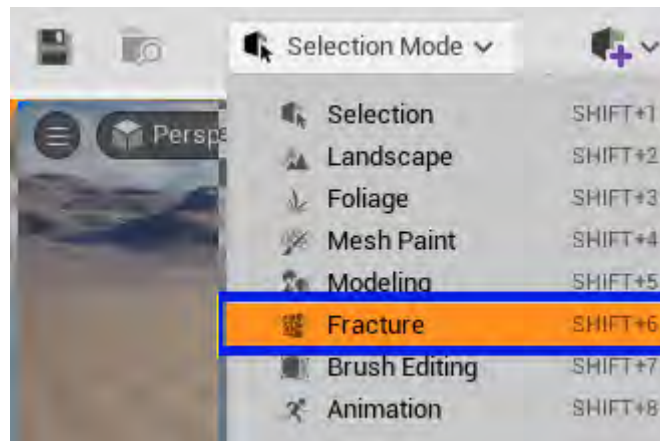


Создание геометрической коллекции Assets

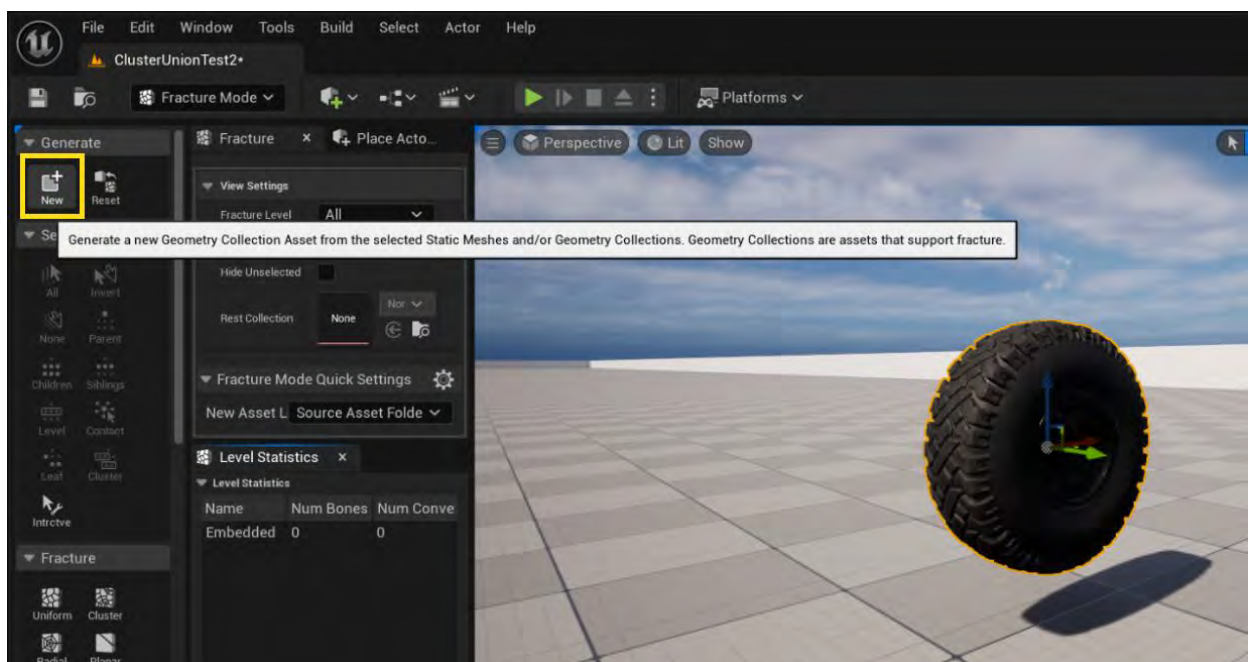
Модуль транспортное средство в настоящее время использует геометрическую коллекцию для частей транспортного средства. Тот же самый Assets используется при физическом разрушении. Инструменты разрушения могут использоваться, чтобы сгенерировать геометрическую коллекцию из статических сеток.

Преобразуйте каждую из отдельных частей транспортного средства от сетки статической **mesh** до геометрической коллекции, используя инструменты перелома, выполнив эти шаги для разрушаемых объектов in the Main Toolbar:

1. Тяните статическую сетку на уровень и измените в редакторе на главной панели инструментов на разрушаемых объектах **режим разрушения**.



2. На статической сетке Assets щелкните по **Generate > New**.



3. В Избранном окне **Пути** выберите местоположение, где будете хранить новый asset и дайте имя Assets.
4. Щелкните по **Generate Geometry Collection**.
5. В **Details** для Assets геометрической коллекции задайте массу каждой части и порог срабатывания/повреждения для того, чтобы часть легко прерывалась.



Повторите эти шаги для всех других частей транспортного средства. Вам нужны будут по крайней мере два Assets: Assets для шасси транспортного средства и Assets для колеса.

Создание кластерного узла Pawn Blueprint транспортного средства

Pawn обозначает аватар игрока. Классы Pawn получают данные от класса PlayerController и используют их, чтобы направлять аватара в игре.

1. В Браузере Контента/Content Browser щелкните по **Add (+)>Blueprint Class/ Класс Проекта**.
2. В окне Класса выбора Родителя /Pick Parent Class, под Всеми Классами /All Classes, найдите "ModularVehicle" и выберите **ModularVehicleClusterPawn**.
3. Щелкните по выбранному **Select**.
3. Click **Select**.



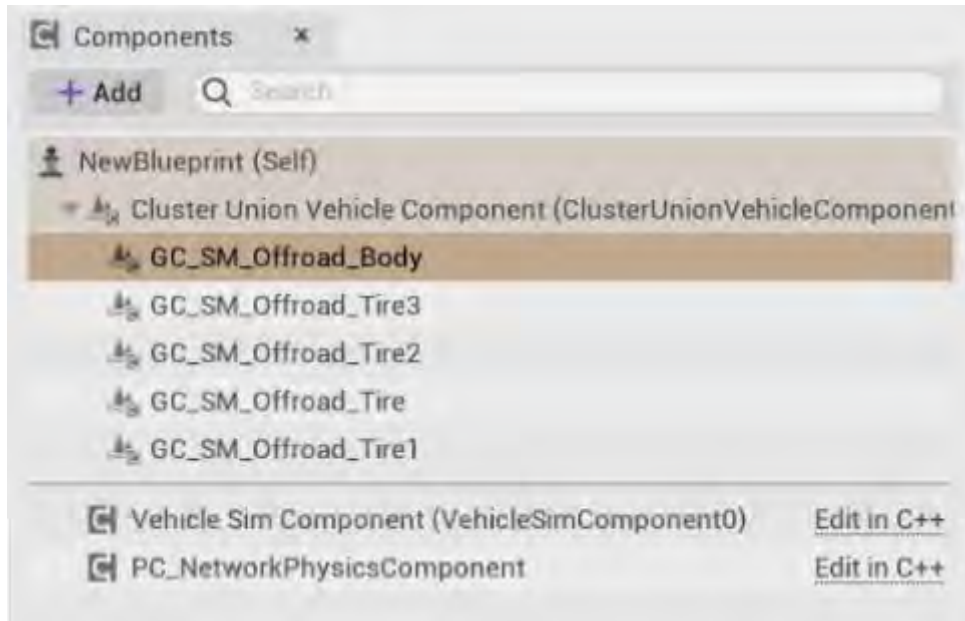
Новый Asset появится в браузере контента/Content Browser. Дайте ему узнаваемое имя так, чтобы можно было легко определить местонахождение его позже (например, "BP_ChaosFrontWheel").

Создание структуры транспортного средства в Blueprint

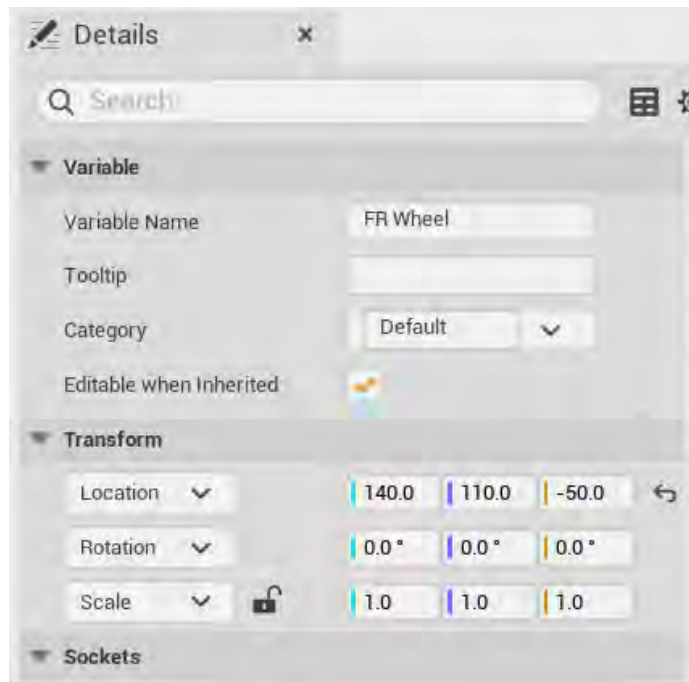
Теперь, когда Вы создали модульный проект транспортного средства / Modular Vehicle Blueprint, откройте его в редакторе проекта и начинаете добавлять компоненты (Geometry Collection Components). У кластерного Pawn узла транспортного средства есть некоторые компоненты значения по

умолчанию, уже назначенные: Компонент моделирования транспортного средства, ответственный за модульное моделирование транспортного средства, а также сетевой физический компонент.

1. Добавьте компоненты геометрической коллекции для каждой из частей транспортного средства, перетаскивая каждый Assets геометрической коллекции в панель **Components** под the **Cluster Union Vehicle**/Кластерным Компонентом транспортного средства узла.



2. Используйте **Viewport** / **Область просмотра**, чтобы поместить части транспортного средства, таким образом, чтобы они были расположены правильно относительно друг друга. Для точности может быть лучше войти в точное местоположение для частей в панели **Details** компонента геометрической коллекции. Это особенно важно для колес, как будто они не совершенно симметричны, моделирование не могло бы двигаться в прямой линии. Конвенция X – вперед, Y – право, Z – вверх.



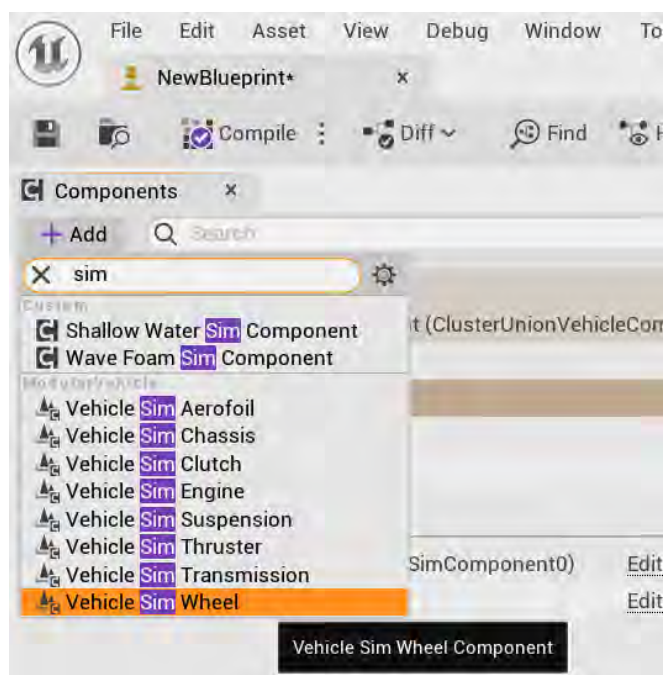
3. В панели **Details**, под **Chaos Physics > Clustering** доступно **Обновление Сил преобразования /Force Update Active Transforms**. Если Вы не сделаете этого, то части, которые слетают/падают с транспортного средства, будут следовать за главным актером вокруг, как только они останавливаются. Это происходит из-за оптимизации, которая прекращает обновление преобразования, когда объект не перемещается. Обычно прекрасно, если у геометрической коллекции есть свой собственный актер, однако здесь мы – в родительской связи с совместно используемому движущемуся actor.

4. Используя одну модель колеса для всего транспортного средства, Вы можете щелкнуть колесами на одной стороне транспортного средства, чтобы выглядеть правильными. Это включает добавление 180 backdegree вращений к оси вращения Z. Это щелкание осей также потребует изменения в следующем шаге этого руководства, когда Вы настроите компонентов моделирования, связанных с этими частями.

Добавление компонентов моделирования к геометрической коллекции

Компоненты моделирования добавляют возможности моделирования в геометрической коллекции. Например, если Вы хотите, чтобы геометрическая коллекция вела себя как колесо, Вы должны добавить компонент Wheel / Колесо.

В редакторе проекта щелкните по **Add (+)** и найдите "sim", чтобы получить список всех доступных опций моделирования.

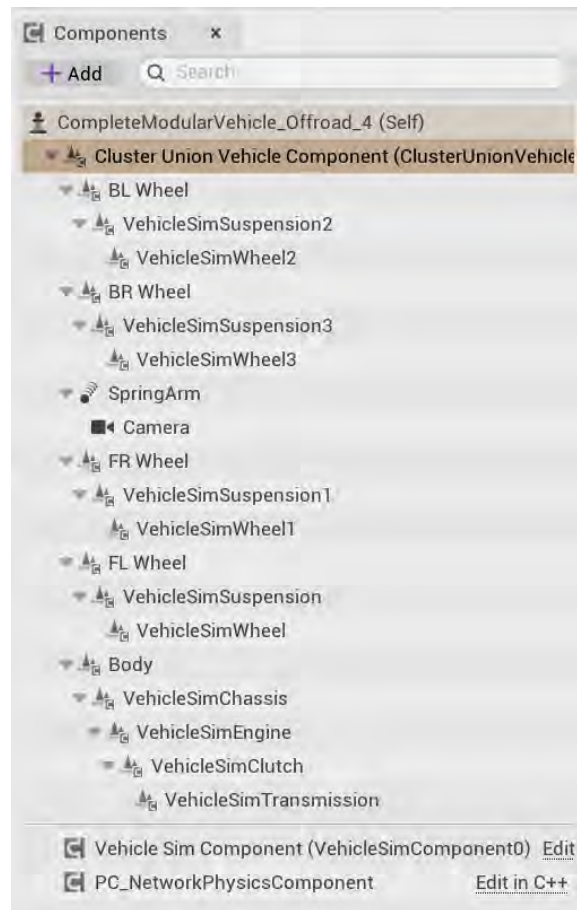


Иерархия компонентов моделирования очень важна. Компоненты моделирования должны быть наследуемыми/потомками своей родительской коллекции геометрии. Например, компонент `VehicleSimWheel` должен быть добавлен в соответствии с моделью колеса коллекции геометрии. Моделирование использует преобразование того компонента для того расположения, где силы должны быть применены от, и также выключить моделирование этой части, если это прервано. Отношения и потомков самих компонентов моделирования также важны. Отношения родителей и потомков двигателя, сцепления, КПП указывает на поток крутящего момента между модулями.

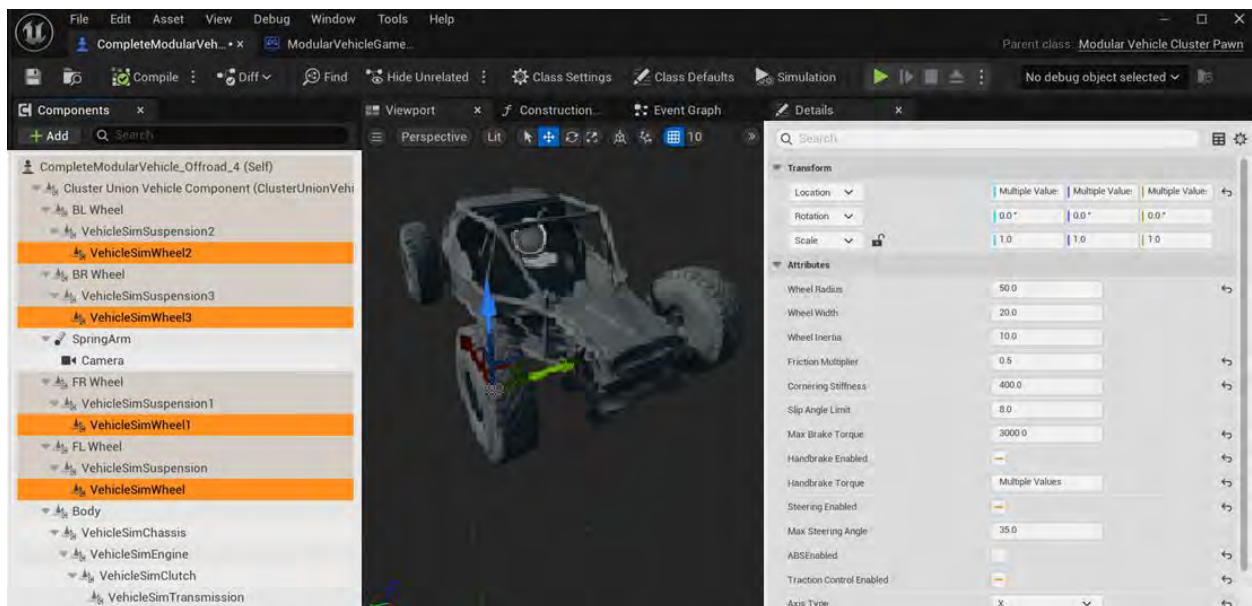
Для добавления компонентов моделирования к коллекциям геометрии выполните эти шаги:

1. Добавьте компонент `VehicleSimSuspension` под каждым из компонентов коллекции геометрии, представляющим колесо.
2. Добавьте компонент `VehicleSimWheel` под каждым компонентом `VehicleSimSuspension`.
3. Добавьте компонент `VehicleSimEngine` под компонентом коллекции геометрии шасси транспортного средства.
4. Добавьте компонент `VehicleSimClutch` под `VehicleSimEngine`.
5. Добавьте компонент `VehicleSimTransmission` под `VehicleSimClutch`.

Законченная установка должна быть похожей на следующее изображение.



Измените все параметры моделирования в панели деталей, подходящие для Вашего транспортного средства. Можно изменить параметры на нескольких компонентах колеса одновременно, удерживая **Ctrl** при выборе компоненты. Затем любые изменения, которые Вы вносите в панели деталей, будут относиться ко всем выбранным компонентам.

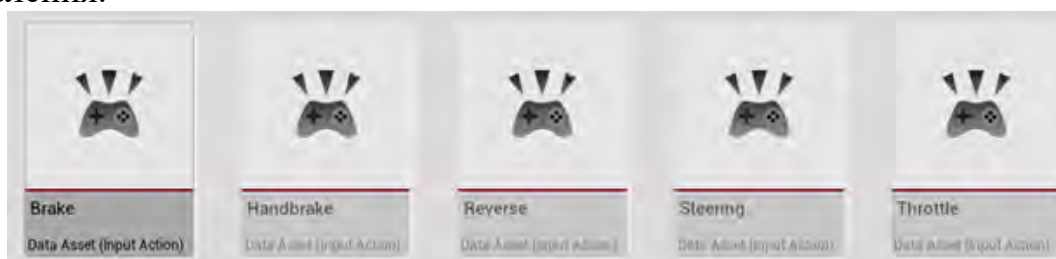


Чтобы заставить колесо моделировать движение назад (180 градусов относительно Z) при правильном условии вращения компонента колеса в редакторе, должно быть выбрано **Reverse Direction**.

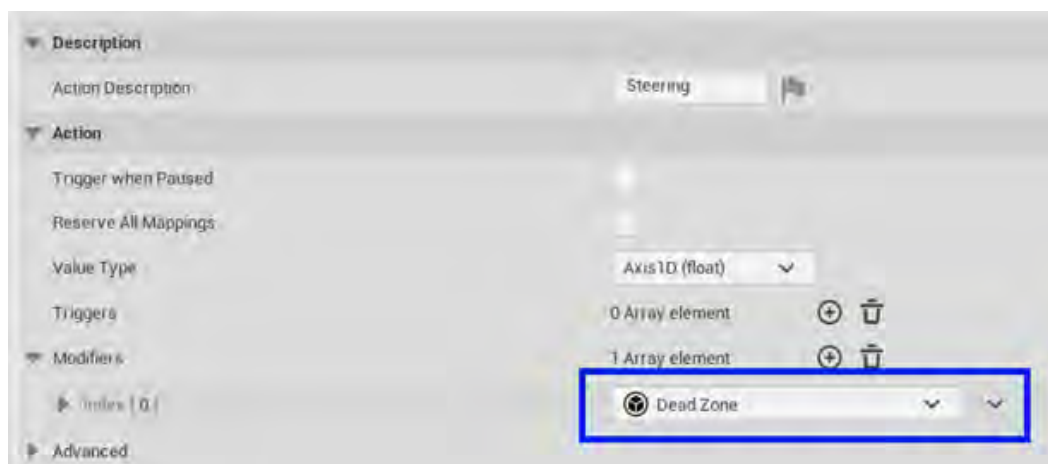


Определение входных данных транспортного средства

1. Создайте входные действия для каждого из ваших средств управления.



2. Для действия рулевого управления желательно добавить “мертвый зональный модификатор”, иначе регулирование может дрейфовать, когда оно должно двигаться прямо.



3. В **Content Browser** щелкните по **Add (+) > Input > Input Mapping Context**.

4. Откройте входной контекст отображения и средства **map controls** / управления картой к Вашему каждому из ваших входных действий, как показано на следующем рисунке/изображении.

▼ Mappings

▼ Mappings +

▼ Throttle

▼ Gamepad Right Trigger Axis

Triggers	0 Array element		
Modifiers	0 Array element		
Setting Behavior	Inherit Settings from Action ▼		
Player Mappable Key Settings	None ▼		

▶ W ▼

▼ Steering

▼ Gamepad Left Thumbstick X-Axis

Triggers	0 Array element		
Modifiers	0 Array element		
▶ Modifiers From Input Action	1 Array element		
Setting Behavior	Inherit Settings from Action ▼		
Player Mappable Key Settings	None ▼		

▼ A ▼

Triggers	0 Array element		
▼ Modifiers	1 Array element		
▶ Index [0]	Negate ▼ ▼		
▶ Modifiers From Input Action	1 Array element		
Setting Behavior	Inherit Settings from Action ▼		
Player Mappable Key Settings	None ▼		

▼ D ▼

Triggers	0 Array element		
Modifiers	0 Array element		
▶ Modifiers From Input Action	1 Array element		
Setting Behavior	Inherit Settings from Action ▼		
Player Mappable Key Settings	None ▼		

▼ Brake

▼ Gamepad Left Trigger Axis

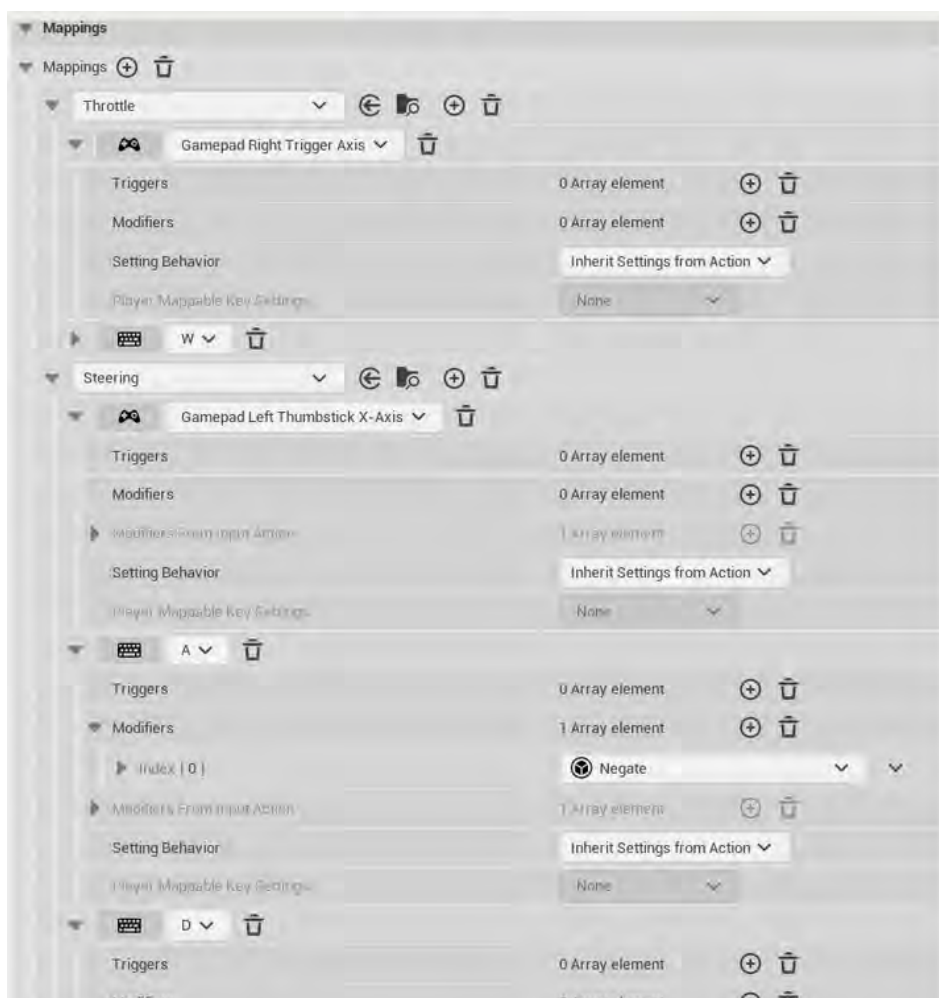
Triggers	0 Array element		
Modifiers	0 Array element		
Setting Behavior	Inherit Settings from Action ▼		
Player Mappable Key Settings	None ▼		

▼ Brake		
▼ Gamepad Left Trigger Axis		
Triggers	0 Array element	
Modifiers	0 Array element	
Setting Behavior	Inherit Settings from Action ▼	
Player Mappable Key Settings	None ▼	
▼ S ▼		
Triggers	0 Array element	
Modifiers	0 Array element	
Setting Behavior	Inherit Settings from Action ▼	
Player Mappable Key Settings	None ▼	
▼ Handbrake		
▼ Space Bar ▼		
Triggers	0 Array element	
Modifiers	0 Array element	
Setting Behavior	Inherit Settings from Action ▼	
Player Mappable Key Settings	None ▼	
▼ Gamepad Face Button Bottom ▼		
Triggers	0 Array element	
Modifiers	0 Array element	
Setting Behavior	Inherit Settings from Action ▼	
Player Mappable Key Settings	None ▼	
▼ Reverse		
▼ R ▼		
Triggers	0 Array element	
Modifiers	0 Array element	
▶ Triggers From Input Action	1 Array element	
Setting Behavior	Inherit Settings from Action ▼	
Player Mappable Key Settings	None ▼	
▼ Gamepad Face Button Right ▼		
Triggers	0 Array element	
Modifiers	0 Array element	
▶ Triggers From Input Action	1 Array element	
Setting Behavior	Inherit Settings from Action ▼	
Player Mappable Key Settings	None ▼	
▼ Description		

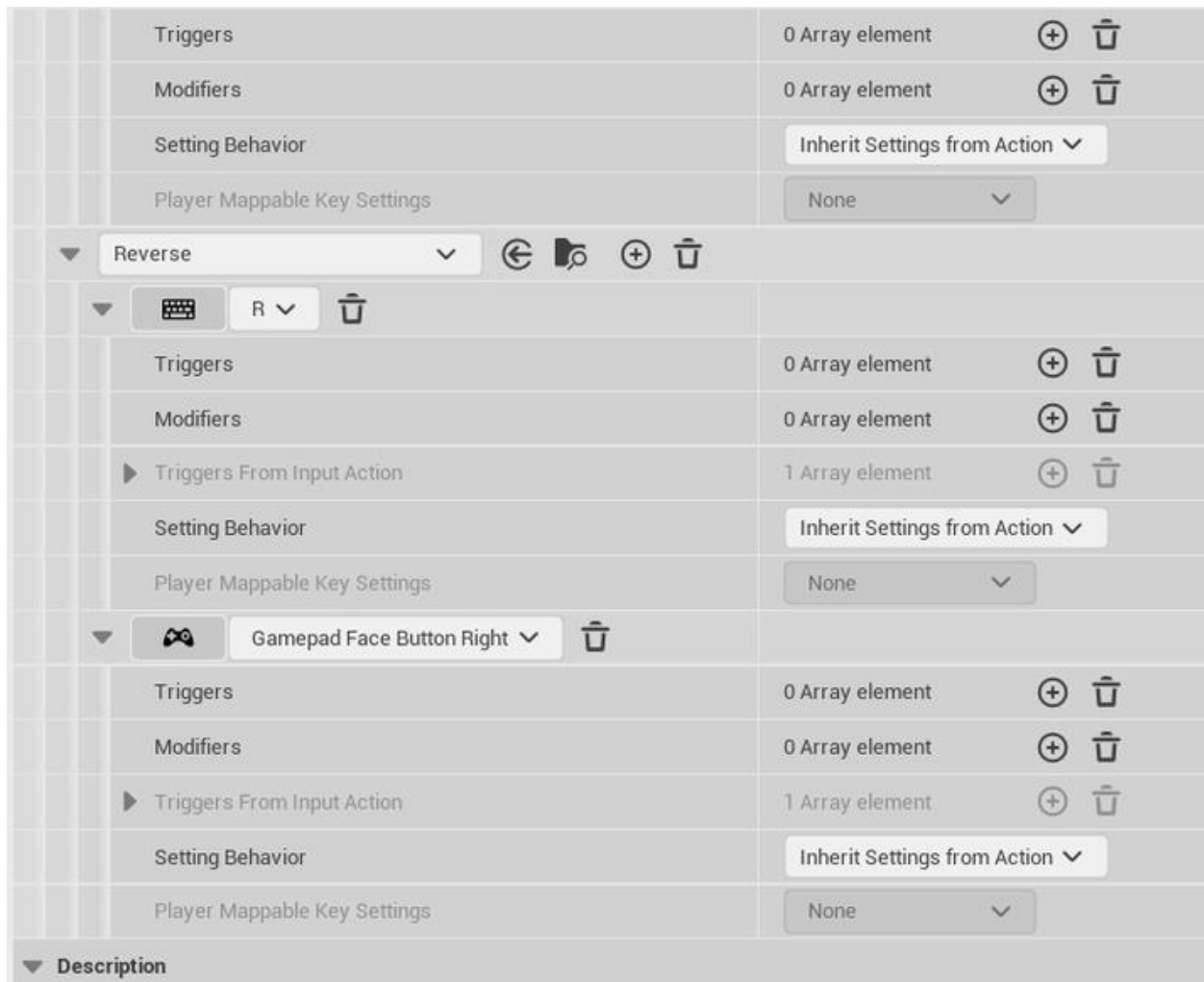
5. На Assets транспортного средства входные параметры определяются по имени и типу. Они должны соответствовать именам, существующих в коде модуля транспортного средства. Основные модули используют следующие входные данные:

Module	Input Name/ имя входа
Engine /Двигатель	Throttle/ Дроссель
Clutch/ Сцепление	Clutch/ Сцепление
Thruster/ Усилитель руля	Throttle/ Дроссель
Transmission/Передача КПП	ChangeUp, ChangeDown
Wheels/Колеса	Steering, Brake, Handbrake, рулевое управление, тормоз, ручной тормоз
Aerofoil/аэродинамический козырек	Roll, Pitch, Yaw

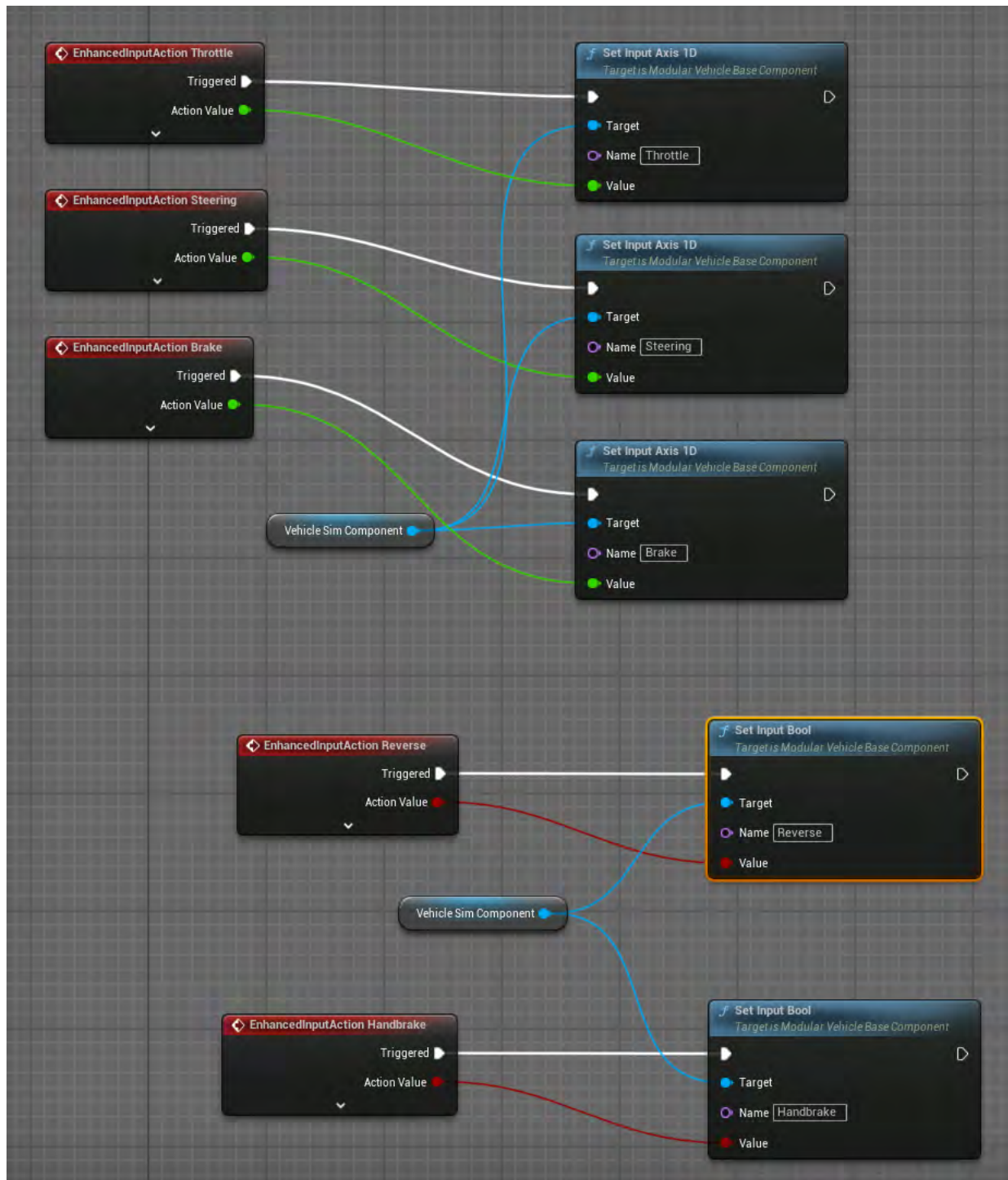
6. На компоненте **Vehicle Simulation/ транспортного средства** добавьте пять участников к разделу Vehicle Input, если они не существуют и заполните их в панели **Details** справа.







7. В Blueprint транспортного средства соедините события EnhancedInputAction с транспортным средством как показано на следующем изображении.



(Дополнительно) Настройте Камеру и пружинный рычаг.

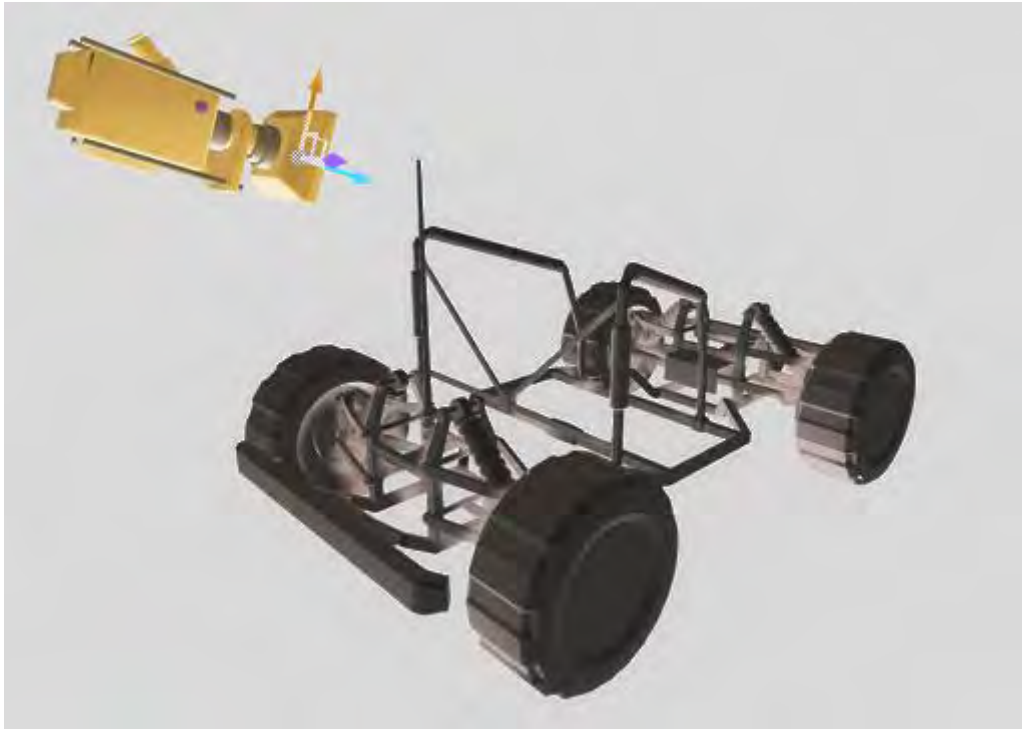
Spring Arm Component (пружинный рычаг, плечо) в UE — это компонент, который используется для удержания дочерних объектов (обычно камеры) на заданном расстоянии от родительского объекта с автоматическим управлением столкновениями и плавным движением.

Вы можете настроить камеру, которая следует за транспортным средством.

1. Откройте Blueprint транспортного средства.

2. В окне **Components** щелкните по **Add (+)>Camera / Камера**.

3. Используйте **Viewport/ Область просмотра** в редакторе Blueprint, чтобы разместить камеру. На следующем изображении камера помещена позади, немного поднята и наклонена вниз к транспортному средству.



1. С выбранным компонентом **камеры** проверьте, что использование **вращения управления узла /Use Pawn Control Rotation** отключено. Это гарантирует, что камера заблокирована на свое направление представления, а не на направление точки просмотра контроллера плеера/ Player Controller's view.

Управление скелетным Mesh из установки модуля

1. Щелкните правой кнопкой мыши в **браузере контента/ Content Browser** и нажмите **Animation> Animation Blueprint**.

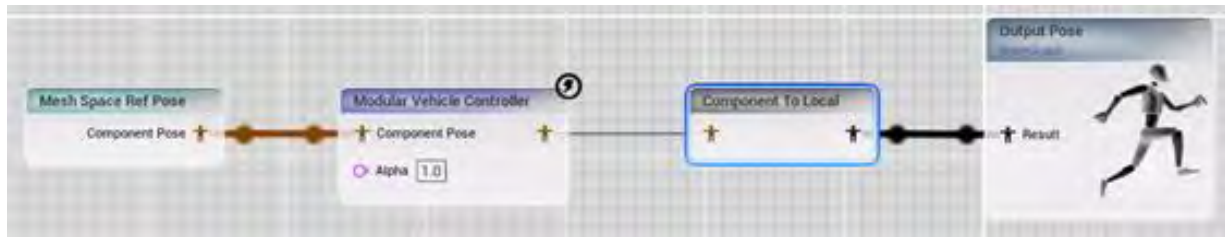
2. Щелкните по скелету, который Вы хотите использовать.

3. Во вкладке **Parent Class** нажмите **modularVehicleAnimationInstance**.

4. Нажмите **Create**.

5. В **Браузере контента** дайте имя своему проекту анимации.

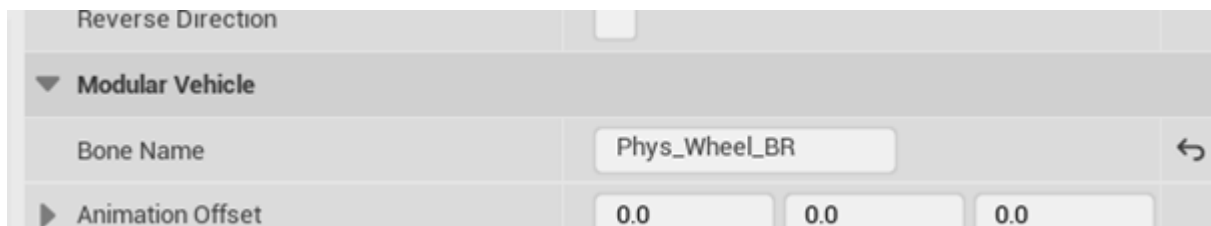
6. Откройте **Blueprint** анимации и создайте **Modular Vehicle Controller**/модульный узел контроллера транспортного средства и соедините его как показано на следующем изображении.



У выбранного компонента **Камеры** проверьте, что **Use Pawn Control Rotation/ Исползования вращения управления Pawn** отключено. Это гарантирует, что камера заблокирована к направлению представления, а не направлению Player Controller's view / представления контроллера плеера.

1. Добавьте vehicle skeletal mesh в vehicle Blueprint
2. На Вашем скелетном сетчатом компоненте убедитесь, что режим анимации установлен **Использовать Проект Анимации / Use Animation Blueprint** и установите **Класс Анимации/Animation Class** в Ваш недавно созданный проект анимации.
3. В каждом из модульных компонентов моделирования транспортного средства (колес и подвески), введите **Bone Name/Имя Кости**, которую Вы хотите оживить в скелетной сетке.

Bone (кость) — это элемент иерархии скелета (Skeleton), который используется для управления деформацией и анимацией Skeletal Mesh. Кости образуют структуру, напоминающую биологический скелет: каждая кость может быть родителем или потомком другой, образуя «скелетную» иерархию, по которой двигаются и деформируются вершины Mesh.



Создание ModularVehicleGameMode Blueprint

Если вы хотите создать транспортное средство вместо персонажа при запуске игры, вам нужно будет создать режим игры на транспортном средстве и назначить его в мировых настройках.

1. В **Content Browser/Браузере Контента** щелкните по **Add (+)>Blueprint Class / Класс Проекта**.
2. В окне **Pick Parent Class /Класса выбора родителя** щелкните по **Game Mode Base**.

3. В **Content Browser/Браузере Контента** дважды щелкните, чтобы открыть новый **Game Mode Blueprint / Режимы Игры в vehicle Blueprint** редакторе проекта.

4. В панели **Details** под **Классами**, установите набор **Класс Pawn** **Значения по умолчанию** к Вашему **vehicle Blueprint/ Проекту** транспортного средства.

5. Щелкните по **Save** и затем закройте окно.

6. В главном окне **Панели инструментов** щелкните по **Settings> World Settings**.

7. В панели **World Settings**, под **Game Mode/Режимом Игры**, устанавливается переопределение **GameMode / GameMode Override/** в Ваш проект режима игры.

Обзор модулей Обзор модулей симуляции

Следующие таблицы в оригинале на английском описывают модули и параметры установки каждого из модулей моделирования.

Module	Forces	Control Inputs Used	Animation
Wheel	Longitudinal and lateral grip forces can be driven or non-driven.	Uses the Steering input if they have steering enabled, the Brake input, and the Handbrake input. A normal vehicle will have steering enabled on the front wheels only however there is nothing stopping you from mixing this up. It is possible to reverse the steering input direction by defining a negative max steering angle, for rear wheel steering.	Wheels will automatically steer using the Steering input angle. Wheel rotation is also animated automatically.

Suspension	Suspending force generally on the Z axis. Ray traces are used to find the ground, which informs the suspension constraint of the hit point and suspension length.	N/A	Wheel travel along the suspension direction is animated automatically.
Engine	Provides torque for wheels. A torque curve can be defined here.	Uses the Throttle input.	N/A
Transmission	Connects to the engine or the clutch module. Gears and gear radios are defined here.	Uses the ChangeUp input and ChangeDown input	N/A
Clutch	Connects between the engine and the transmission modules.	Uses the Clutch input.	N/A
Thruster	A propelling force.	Uses the Throttle and Steering inputs.	Uses the Throttle input to scale force and can be steered using the Steering input
Aerofoil	Lift and drag forces are generated using realistic behavior. Can be used to create an entire flying model with wings, elevator and rudder.	Uses the Roll , Pitch , or Yaw inputs depending on whether it is defined as a wing aileron, an elevator, or a rudder.	The part will rotate depending on the control surface type. The animation can be disabled, reversed, or magnified relative to the actual simulated values.

Параметры установки модуля

Wheel Setup Parameters

Name	Description
Radius	The wheel radius in cm. If this is setup incorrectly, the wheels will look like they are hovering above the ground or embedded into the ground.
Width	Not currently used.
Inertia	How quickly the wheels spin up or down.
FrictionMultiplier	Multiplies the ground material friction to give less or more grip. A value of 1.0 is neutral.
CorneringStiffness	How much lateral forces the wheel generates when cornering.
SlipAngleLimit	The angle in degrees at which the tire starts losing grip.
Max Brake Torque	The maximum braking torque that will be used when the braking input is at its maximum value of 1.0.
Handbrake Enabled Handbrake Torque	The maximum handbrake torque that will be used when the braking input is at its maximum value of 1.0.
Steering Enabled	When enabled, the wheel will react to the Steering input
Max Steering Angle	The maximum steering lock in degrees. A negative value will cause the wheel to steer in the opposite direction. Can be used to employ 4-wheel steering, where the front wheels use positive values and the rear wheels use negative values.
ABS Enabled	When enabled, the wheel will not skid under heavy braking, no matter how high the Max Brake Torque is.
Traction Control Enabled	When enabled, the wheel will not slip or spin under acceleration, no matter how high the Max Brake Torque is.
LateralSlipGraph	The cornering stiffness graph. Not yet exposed to the component.
LateralSlipGraphMultiplier	Easy scaling of the LateralSlipGraph output. Not yet exposed to the component.
Max Rotational Velocity	Limits the wheels maximum spin speed. Not yet exposed to the component.
Axis	Defines whether the forces are acting on the model's X axis or Y axis. Matters if the original GC component has been rotated into position.
Reverse Direction	Lets you reverse the direction of the forces if they are being applied in the wrong direction. Matters if the original GC component has been rotated into position.

Suspension Setup Parameters

Name	Description
Axis	The direction of suspension travel.
Offset	Offsets the location of suspension forces.
Max Raise	The distance in cm that the suspension will compress above the initial position.
Max Drop	The distance in cm that the suspension will drop below the initial position.
Spring Rate	The standard spring rate.
Preload	The constant force applied when the suspension is active. Allows for lesser spring rates while still keeping the vehicle mass suspended.
Spring Preload	The constant force applied when the suspension is active. Allows for lesser spring rates while still keeping the vehicle mass suspended.
Damping	Stops the vehicle from oscillating around the neutral position. The higher the damping, the less the vehicle oscillates.

Engine Setup Parameters

Name	Description
Max Torque	The maximum Torque value
Torque Curve	The normalized torque curve, defining torque against RPM.
Max RPM	The maximum RPM (revolutions per minute) the engine can achieve.
Idle RPM	The idling RPM, when the vehicle is sitting with no throttle depressed.
Engine Braking Effect	How much the engine slows the vehicle down when the throttle is released.
Engine Inertia	How fast the engine revs up to speed.
EnableBoostCapability	When enabled, the engine can activate a boost.
BoostMultiplier	How much the engine torque is multiplied by when the boost is active.

Transmission Setup Parameters

Name	Description
Transmission Type	Whether gears are set manually or automatically. The default is automatic.

Final Drive Ratio	Multiplies the selected gear ratio to give final gear torque multiplying effect. Usually given alongside the regular gear ratios in vehicle specs.
Forward Gear Ratios	The ratios for gears 1 to n.
Reverse Gear Ratios	The ratios for gears -1 to -n.
Change Up RPM	The engine RPM at which the gears will change up in automatic transmission mode.
Change Down RPM	The engine RPM at which the gears will change down in automatic transmission mode.
Gear Change Time	The time it takes to change up or down through each of the gears.
Transmission Efficiency	A way of defining the frictional losses in the transmission system. Not exposed to the component.
Auto-Reverse	When enabled, the vehicle will reverse instead of braking when the velocity is near zero.

Clutch Setup Parameters

Name	Description
Clutch Strength	How abruptly the clutch matches the velocities on the two shafts.

Thruster Setup Parameters

Name	Description
Force Axis	The axis along which the force acts
Steering Axis	The axis around which the thruster rotates, if steering is enabled.
Force Offset	Offsets the location where the force is applied.
Max Thrust Force	The maximum thrust force.
Steering Enabled	When enabled, the thruster's direction can be controlled.
Max Steering Angle	The maximum angle through which the thruster can turn.
Steering Force Effect	Multiplies the steering component of the force so that the forward thrust can be tuned separately from the steering. A value between 0 to 1 is expected, where 0 applies no steering effect and 1 applies the full steering effect.
Boost Multiplier	How much the thrust force is increased when the boost input is active. Not used yet.
Max Speed	The maximum speed that should be achieved. Not used yet.

Aerofoil Setup Parameters

Name	Description
Force Axis	The axis along which the lift force is applied.
Offset	Offsets the location where the force is applied.
Control Rotation Axis	The axis around which the control surface rotates.
Surface Area	The surface area of the aerofoil. Doesn't need to be realistic or match the visual model. The larger the value, the more lift you will achieve.
Camber	The aerofoil shape. The larger the value, the more lift you will achieve.
Max Control Angle	The angle in degrees through which the control surface can turn.
Stall Angle	The stall angle of the aerofoil in degrees.
Type	Specifies whether this is a Wing, Elevator, or Rudder. Defines which control input is moving the control surface, where Wing uses the Roll input, Elevator uses the Pitch input, and Rudder uses the Yaw input.
Lift Multiplier	A way of cheating the calculated lift component of the aerofoil calculation, separate from the drag. The default value is 1.0.
Drag Multiplier	A way of cheating the calculated drag component of the aerofoil calculation, separate from the lift. The default value is 1.0.
Animation Magnitude Multiplier	Changes how much the part visually rotates independent of the control angle. The part will be animated to an angle equal to the control angle multiplied by this value.

Unreal: Как создать Chaos транспортного средства с анимацией рулевого колеса

Данный материал представляет сокращенный перевод “Unreal: How to create a Chaos Vehicle with animated Steering Wheel”, Purple Ions / January 19, 2024, взятый из

<https://neutroniogames.wordpress.com/2024/01/19/how-to-create-a-chaos-vehicle-with-animated-steering-wheel>.

Данное руководство позволяет не только анимировать руль, но также и другие части в транспортном средстве, например, двери или кнопки.

Создание 3D графики

Создайте 3D модель автомобиля с рулем, конвертируйте ее в UnrealEngine и расширьте структуру транспортного средства с bones /костями для руля. Создайте новую сетку руля, родитель это к арматуре транспортного средства, создайте и поместите кость, назначьте группы вершины.

Примите, во внимание, следующее:

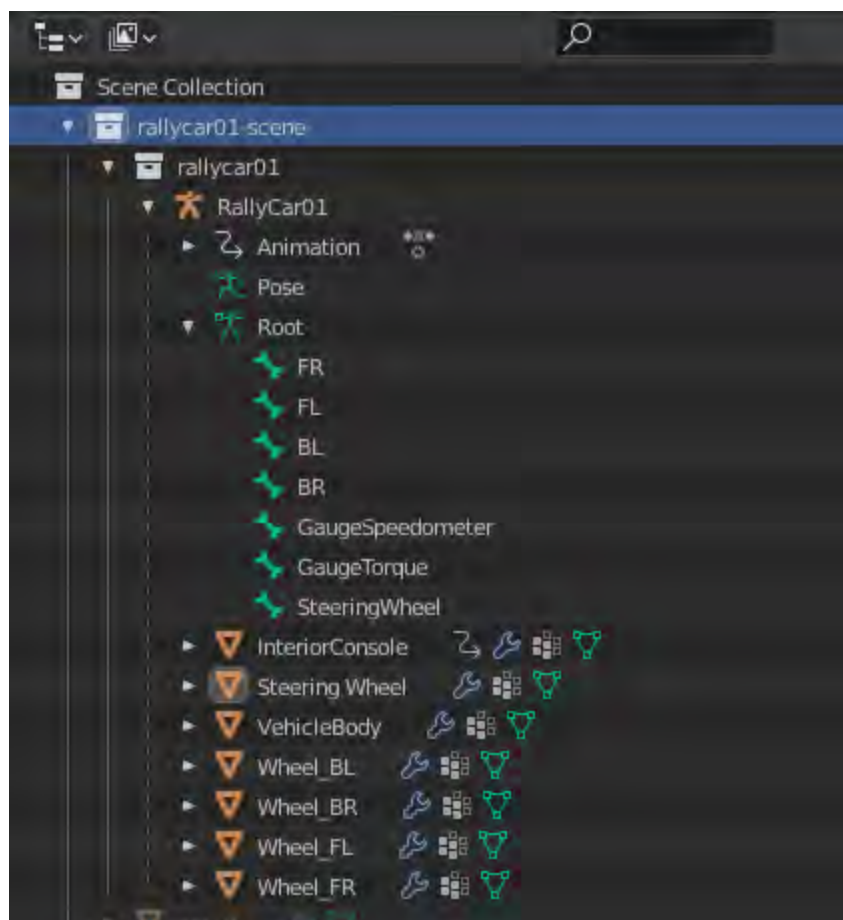
Руль должен быть подстроен, т.е. присоединен к Bone/кости в Вашем скелете транспортного средства в дополнение к Вашей стандартной установке колеса транспортного средства Chaos.

Руль должен быть *отдельной* сеткой, аналогично Chaos Vehicle Wheels.

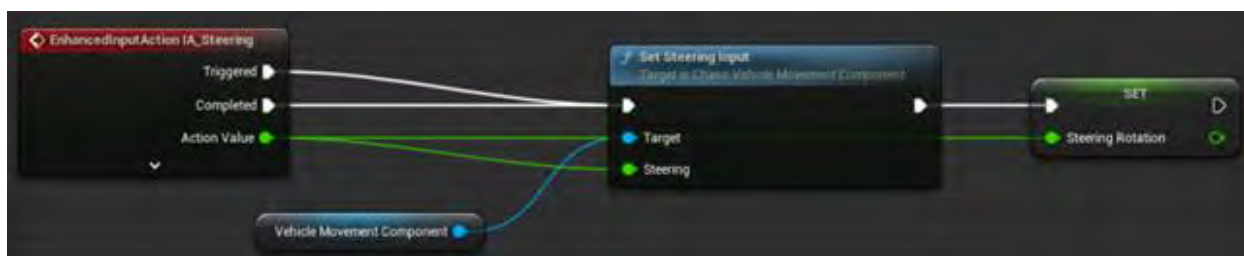
Соединение хвоста кости должен быть присоединено туда, где Вы хотите, чтобы руль вращался.



Конечный результат должен выглядеть примерно так:



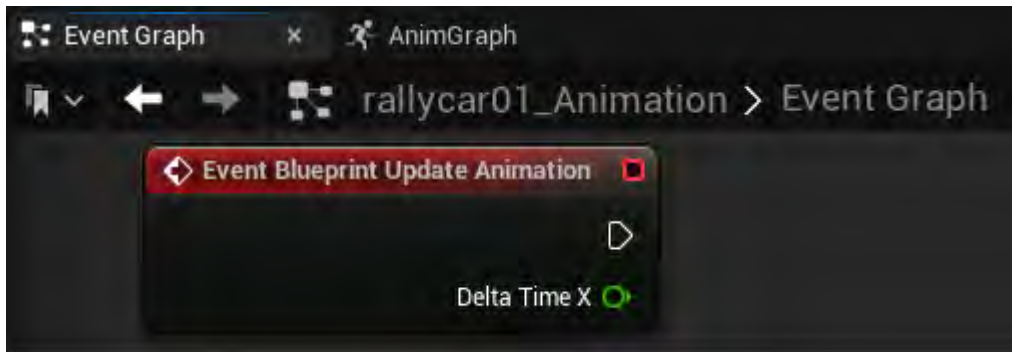
Добавим новое поле Float под названием *Steering Rotation*/ к нашему VehiclePawn. Когда входное событие задано, *IA_Steering*, удерживающий вращение, определен. Далее на стороне Актера необходимо настроить анимацию.



Настройка Анимации

Резюмируем: Если колеса крутятся влево или вправо, также хотим, чтобы наш руль вращался влево или вправо. Когда колеса возвращаются к нейтральному положению, наш руль делает то же.

Для настройки логики откройте проект Анимации своего транспортного средства и откройте График Событий.



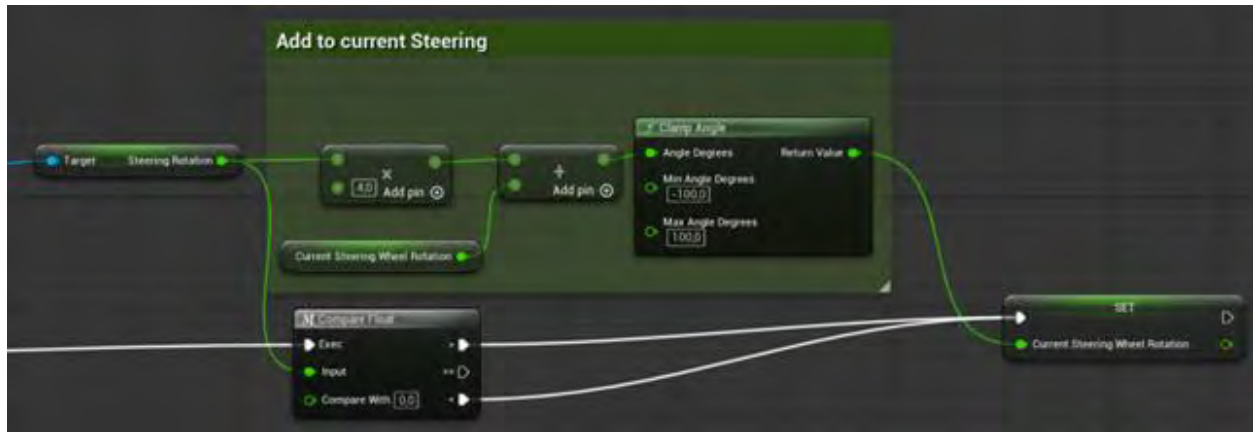
В проекте анимации создаем новое Поле “Текущее Вращение руля/ Current Steering Rotation” плавающего типа, которое представляет текущее вращение угла нашего руля.

Steering Rotation / вращение руля задается в диапазонах VehiclePawn от -1 до 1, где 0 определяет нейтральное положение. Необходим оператор ветвления, чтобы отличить эти случаи.

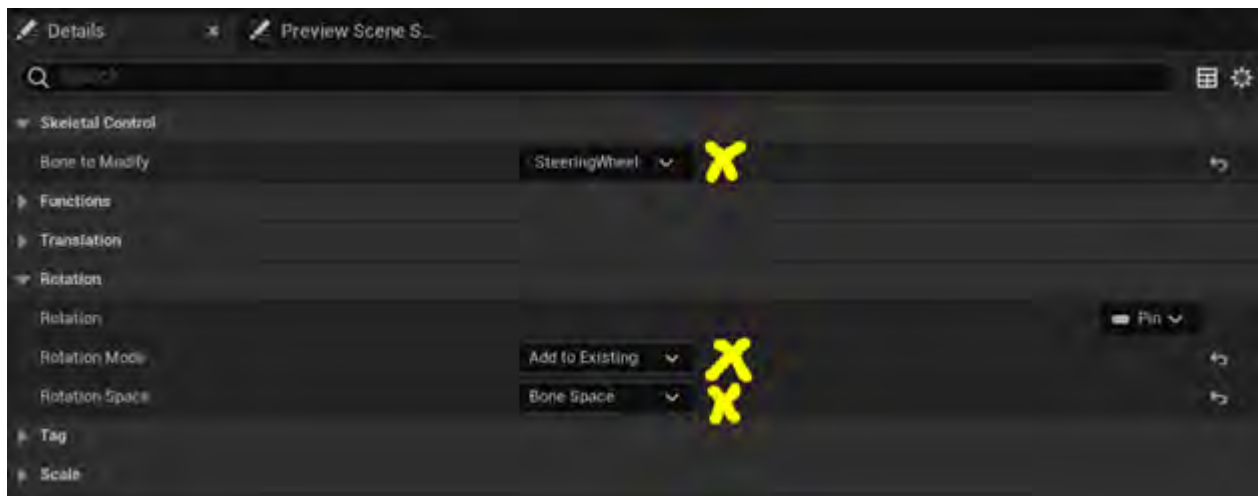


Для этого возьмем собственника VehiclePawn и переместим его в наш VehiclePawn. Таким образом, можем получить доступ к полю Steering Rotation, которое добавили ранее. Далее добавим Compare Float-node / сравнивающий узел плавающего типа, чтобы реагировать на различные направления руля.

Первый случай, с которым сталкиваемся, является нормальными событиями руля при повороте влево и вправо. Давайте возьмем в настоящее время сохраненное вращение руля в нашем VehiclePawn, умножим его на 4 (скорость анимации, скорректированная к скорости анимации колес). Добавим этот результат вместе с текущим Current Steering Wheel Rotation / вращением руля, ограничиваем его между -100 ° и 100 ° (сколько рулю позволяют прокрутить в каждую сторону). Тем самым задаем обновленное Current Steering Wheel Rotation / текущее вращение руля.



Еще должны дать команду кости вернуться к текущему вращению Руля. Чтобы сделать это, откройте График Anim. Чтобы вращать наш руль, мы должны добавить узел Кости (Вращающее устройство и Преобразование). Для чего откройте панель деталей “Преобразования (Изменение) узла *Кости* и внесите несколько корректировок. Определите кость, которую предполагается изменить в кости *SteeringWheel*. На вкладке *Rotation* измените Режим Вращения / *Rotation Mode* на добавление “Add to Existing / Существующему” и пространству вращения к “Bone Space/Пространству Кости”, которое гарантирует, чтобы мы применили вращение относительно местного места.



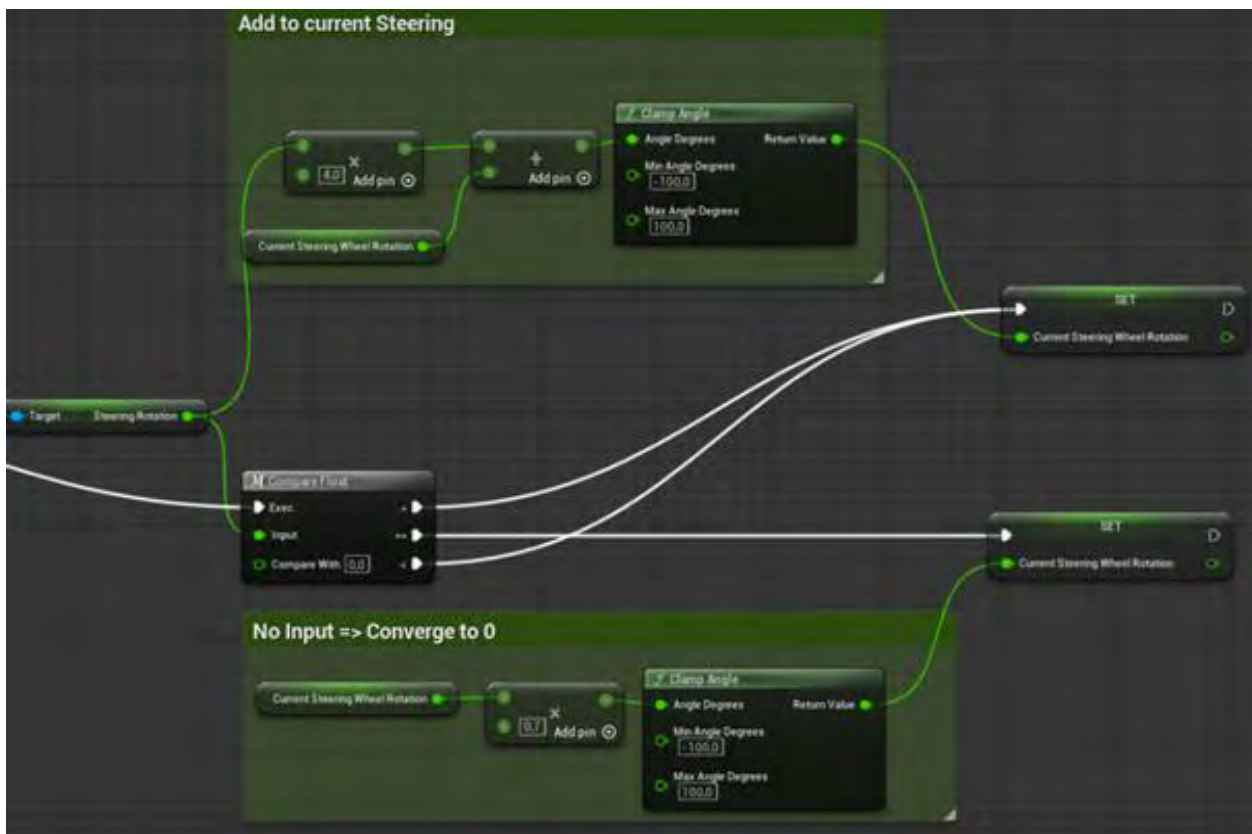
Воврашаемая назад на графику Anim, соединяем поле *Current Steering Wheel Rotation* с вращающим устройством на его соединение “X (Roll)”. Далее подключаем вращающее устройство *Rotator* к точке вращения *Transform (Modify) Bone node* /преобразования (модификации) узла *Bone/Кости*.

Конечный результат, включая *Chaos Wheel Controller* /контроллер колеса *Chaos* должен быть похожим на это.



Если теперь протестируете транспортное средство в редакторе, то увидите, что это действительно поворачивает свой руль. Но руль не возвращается назад, как он должен, как только Вы останавливаете входное воздействие руля. Необходимо решить вопрос с нейтральным положением колеса в нашем проекте анимации.

Когда нет никакого воздействия входа руля, поступающего из VehiclePawn, регулирование прекращается и колесо медленно вращается назад к его оригинальному вращению. Математически, значение, сохраненное в Текущем Вращении Руля / Current Steering Wheel Rotation, должно снизиться до нулевых значений. Чтобы достигнуть этого, умножаем это значение с фактором ниже, чем 1.0 (низкие значения для быстрой сходимости, высокие значения для медленной сходимости, подойдет что-либо между 0.7 и 0.99). Фактор схождения должен быть умножен на дельту времени.



The Unreal Engine 4 Blueprint Vehicle User Guide.

Ниже приведен материал создания автомобиля в Unreal Engine 4.27 Blueprint Vehicle, взятый в оригинале на английском из документации

https://dev.epicgames.com/documentation/en-us/unreal-engine/vehicle-user-guide?application_version=4.27,
который наиболее раскрывает понимание логики работы Unreal Engine.

At the end of this guide, you'll have a fully functional vehicle that you can use in your own game projects.

Goal

In this guide, we'll cover the creation of a vehicle using the **Wheeled Vehicle** Blueprint class in a new blank project. This guide will take you through the process of creating all the necessary in-editor items you need to have a functional vehicle but assumes you have your own **Skeletal Mesh** and a **Physics Asset** ready to go. At the end of this guide, you should have a fully functional vehicle similar to the one from the Vehicle Game or the Vehicle Template.

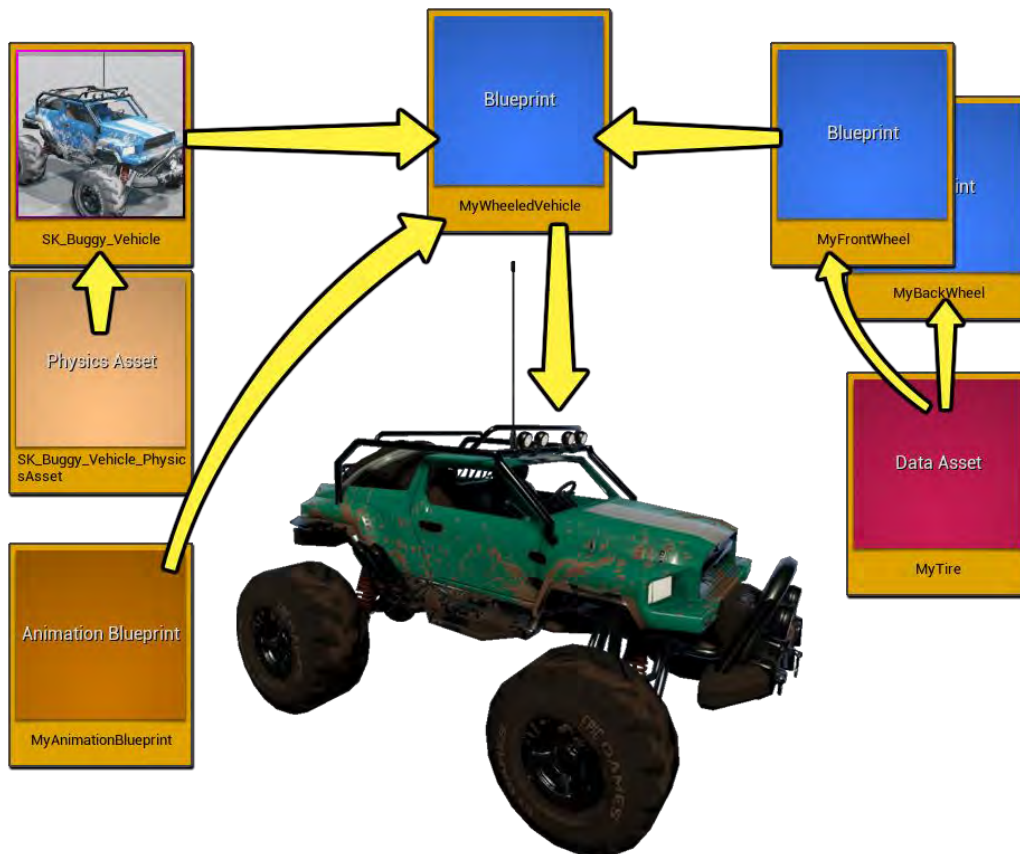
If you need information on getting a vehicle Skeletal Mesh and a Physics Asset into the engine, please see the Vehicle Content Guide. Alternatively, you can work out of the **Vehicle Template** or the **Vehicle Game** project to have access to a pre-built Skeletal Mesh and Physics Asset. The **Vehicle Template** is in the **New Project** menu of the **Unreal Project Browser**, and the **Vehicle Game** is on the **Learn** tab of the **Epic Games Launcher**.

Objective

During the course of this tutorial you will create the key components that make up a Vehicle in Unreal Engine 4 (UE4). Some of this requires assets that you create outside of UE4, while the majority of the components will be created right inside of the Editor!

This is the individual assets that make up the Vehicle we'll construct:

- A Skeletal Mesh
- A Physics Asset
- An Animation Blueprint
- A Vehicle Blueprint
- One or more Wheel Blueprints.
- A TireConfig Data Asset



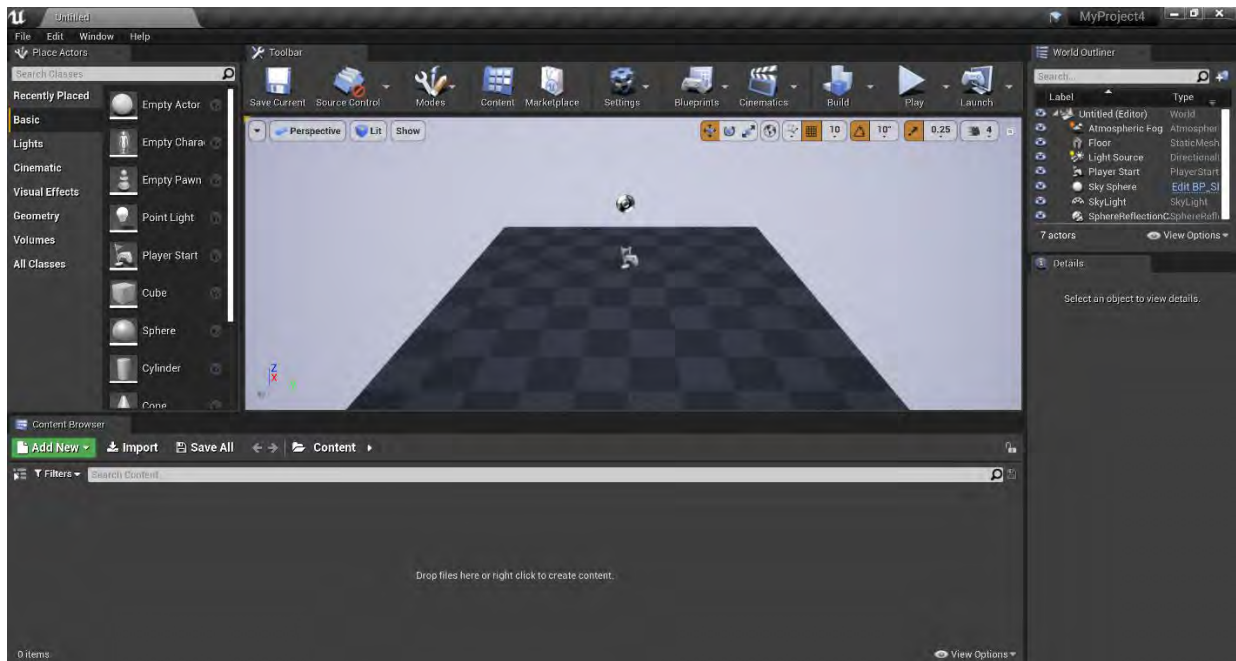
By the end of this guide you'll know how to setup each of these asset types and how they work together to bring your vehicles to life!

- How to configure a TireConfig Data Asset
- How to configure a Wheel Blueprint for your Front and Rear wheels.
- How to setup an Animation Blueprint specifically for a Vehicle.
- How to configure your Vehicle Blueprint with your Skeletal Mesh, Animation Blueprint, and Wheel Blueprint(s).
- How to configure your Axis Mappings and Bindings to control your Vehicle.
- How to create and assign a new Game Mode to spawn your vehicle.

1 - Required Project Setup

1. Open **Unreal Engine** from the Launcher. The Project Browser will appear.
2. Create a new Project from the **Games** category, with **C++** and **With Starter Content** enabled. We'll need to enter a project name, so we'll use "QuickStart". We can now click **Create Project** and get started.

The **Unreal Editor** will now open our new project. **Visual Studio** will also open and load the solution file that our project has created.



2 - Creating a TireConfig Data Asset and Wheel Blueprint

In this first step to creating our own vehicle, we have two parts that go together to make up the wheels for the vehicle; the TireConfig Data Asset and the Wheel Blueprint class. We'll start by first creating the TireConfig Data Asset and then move on to create the Wheel Blueprint that houses properties for our wheels, including where the TireConfig data is assigned to. Follow along with the steps below to get started creating your TireConfig data and then using that with a Wheel Blueprint.

Tire Config Data Asset

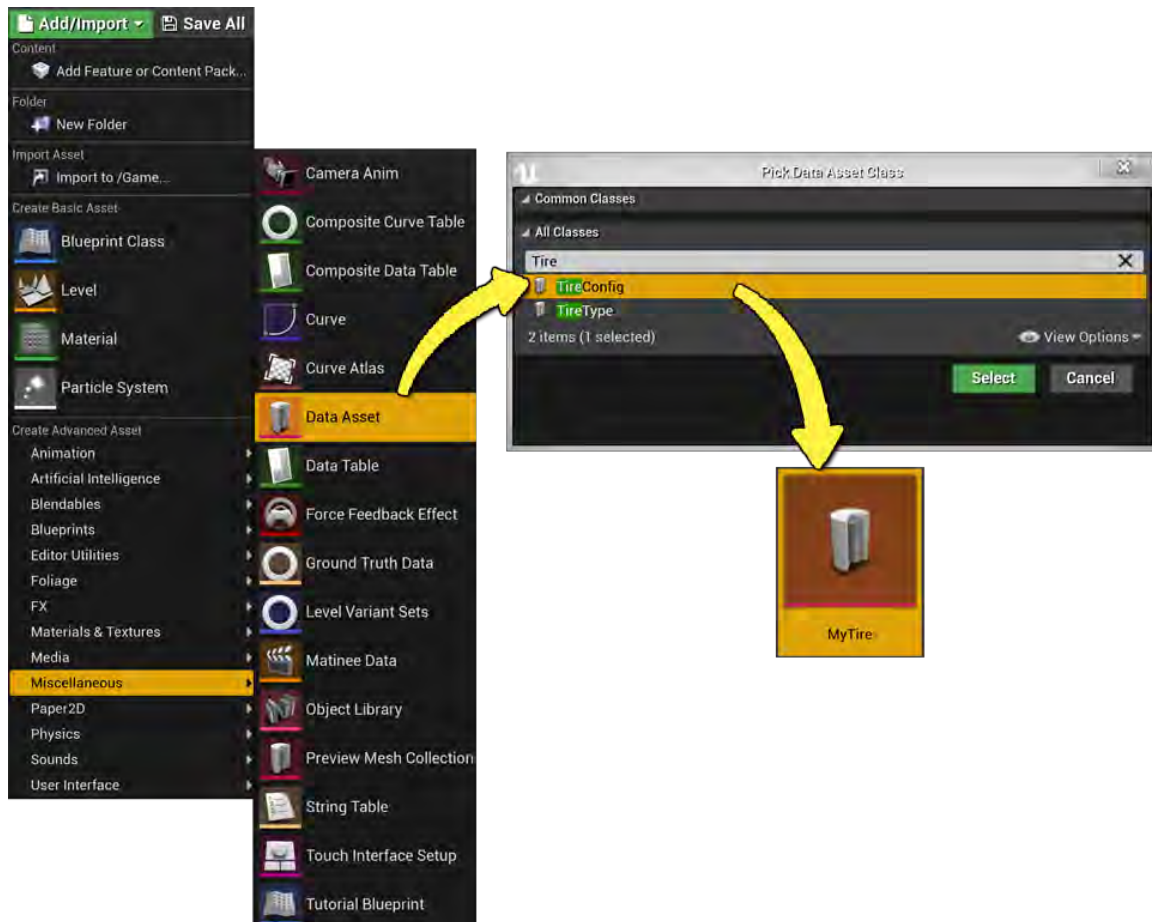
The **TireConfig Data Asset** is used to control **Friction Scale**. This value not only affects the raw friction of the wheel but also scales the values for how difficult (or easy) it is for the wheel to slide while in a hard turn. There is also the additional option of **Tire Friction Scales**, which enables you to specify a specific Friction Scale to different Physical Material types.



Create your TireConfig Data Asset

1. In the **Content Browser**, click the **Add New** Button, then hover over **Miscellaneous** and then select **Data Asset** from the Context Menu.
2. In the **Pick Data Asset Class** window, select **TireConfig** to create this type of Data Asset.

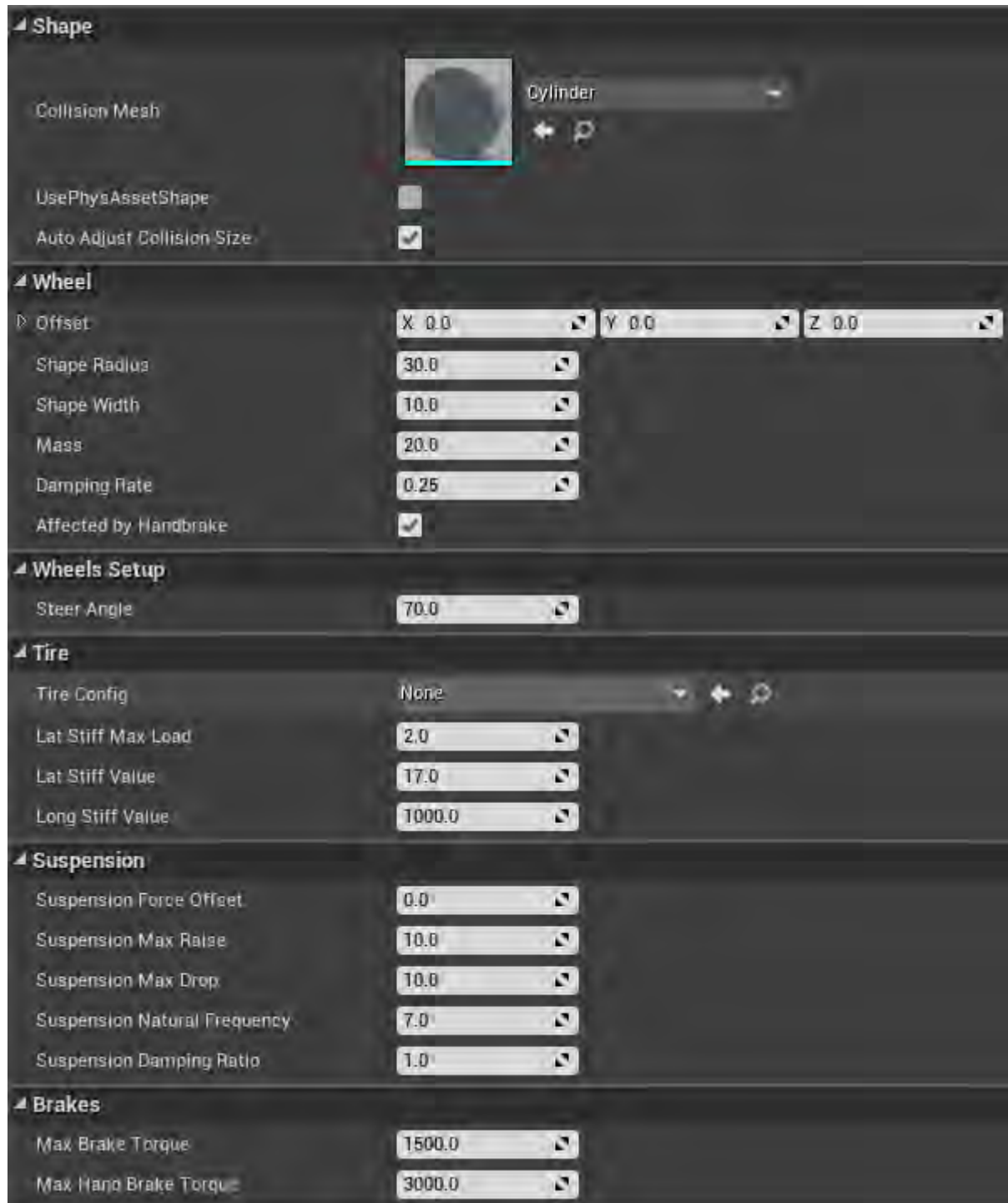
3. The new asset will be created in your **Content Browser**. Make sure to give it a recognizable name so you can easily locate it later.



In the Pick Data Asset Class window, you may notice that there is also a **TireType** Data Asset class. This is a deprecated function and is only used when upgrading older projects to a recent version of Unreal Engine. You should not have any reason to use this data asset type if using a recent version of Unreal Engine, 4.15 or higher.

Wheel Blueprints

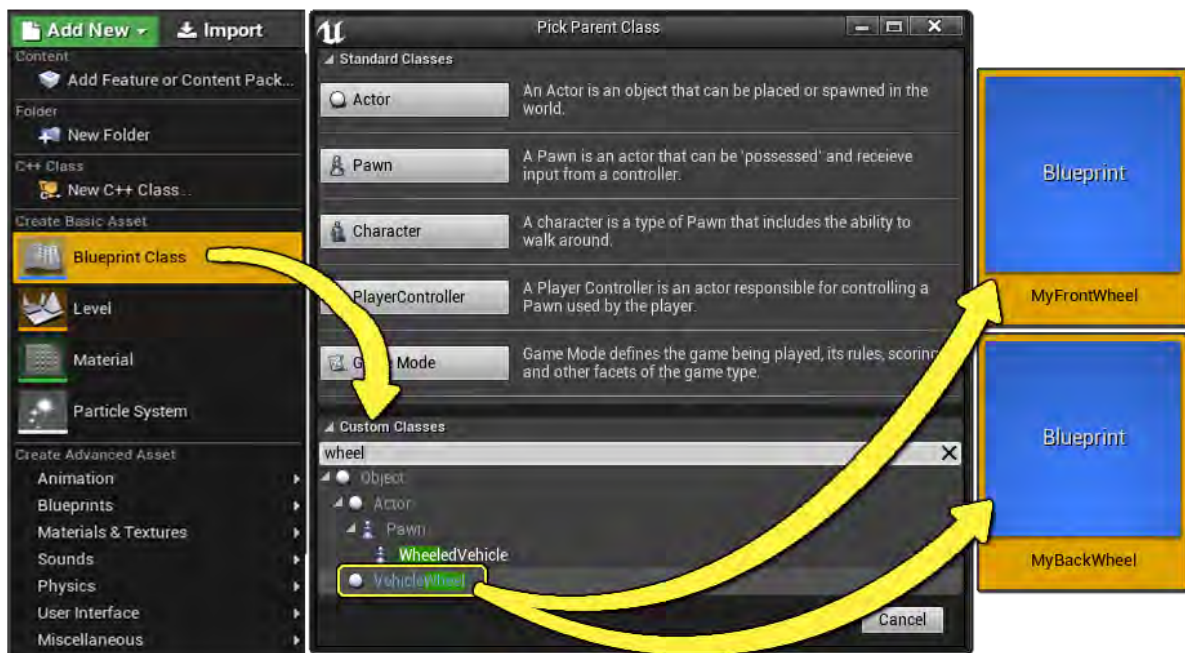
In most cases, you will have at least two wheel types; a wheel that is affected by steering and one that is not. Also, this may be the case for having differently sized wheels for the front or the back, in which case, you have full control over setting the differing radii, mass, width, handbrake effect, suspension, and many other properties to give your vehicle the handling you desire.



Click the image for full size view.

Create your Wheel Blueprint

1. In the **Content Browser**, click the **Add New** button, then select **Blueprint Class** from the menu.
2. In the **Pick Parent Class** window, under **All Classes**, search for "wheel" and then select **VehicleWheel**.
3. The new asset will be created in your **Content Browser**. Make sure to give it a recognizable name so that you can easily locate it later.
4. **Optional** - Repeat these steps again so that you have a **front** and **rear** wheel type.

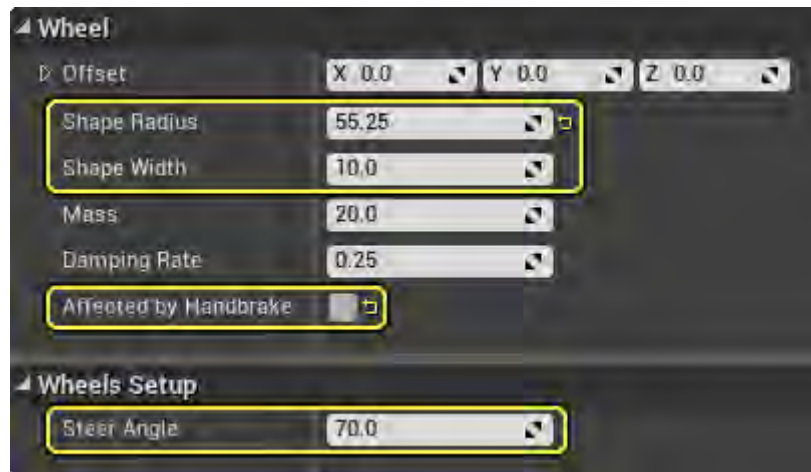


Editing the Wheel Blueprint

Now that you've got your Wheel Blueprint(s), open them in the Blueprint Editor, where you'll see all the available options to edit your wheels!

There are five properties that we need to initially change for each wheel, as the rest of them will change how the vehicle performs (and should be tweaked later) as you start to test out the vehicle in your own game.

- Shape Radius
- Shape Width
- Affected by Handbrake (usually restricted to the rear wheels)
- Steer Angle (usually only the front wheels)



The properties here are set to match the Buggy from Vehicle Game. If you're using your own Skeletal Mesh, you may need to use different values for the **Shape Radius** and **Shape Width**.

Now, under the **Tire** section, you can assign the Tire Config Data Asset that you initially created using the **Tire Config** selection box.



End Result

At this point, you've set up your Tire Config Data Asset, which is used to control how much the tires slide on any surface, having set up the Wheel Blueprint that can be used to specify all of the different properties for your wheel. In the next step, you'll continue creating the assets you need, which make up our vehicle ,by creating the Animation Blueprint that handles all of the necessary animation information for our vehicle (such as spinning tires, steering, suspension, etc).

3 - Creating a Vehicle Animation Blueprint

At this point, you've created the TireConfig Data Asset and your Wheel Blueprint(s), which you'll use later with the Vehicle Blueprint. In this step, you'll continue to ready your content by creating an Animation Blueprint that drives all of your animation needs for the wheels on the vehicle.

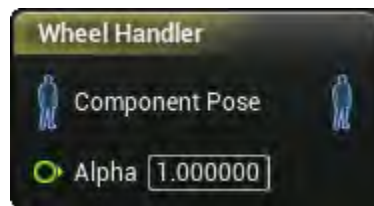
Animation Blueprints

The Animation Blueprint will be used to control the animation of our Vehicle Skeletal Mesh so that it has animations specific to a vehicle, like spinning tires, suspension, handbrakes, and steering. To offload a lot of the work in creating

animations for these types of things, you can use the **Wheel Handler** node to drive these types of animations.

Wheel Handler Node

Where the Animation Blueprint is used to get and control animations for the vehicle, more specifically, the **Wheel Handler** node makes controlling all of the animations for the vehicle fairly easy with little to no additional setup. It simply grabs the necessary information from the wheels (How fast is it spinning? Is it affected by the Handbrake? What are the suspension settings for this wheel?) and translates the query results to an animation on the bone that the wheel is associated with.



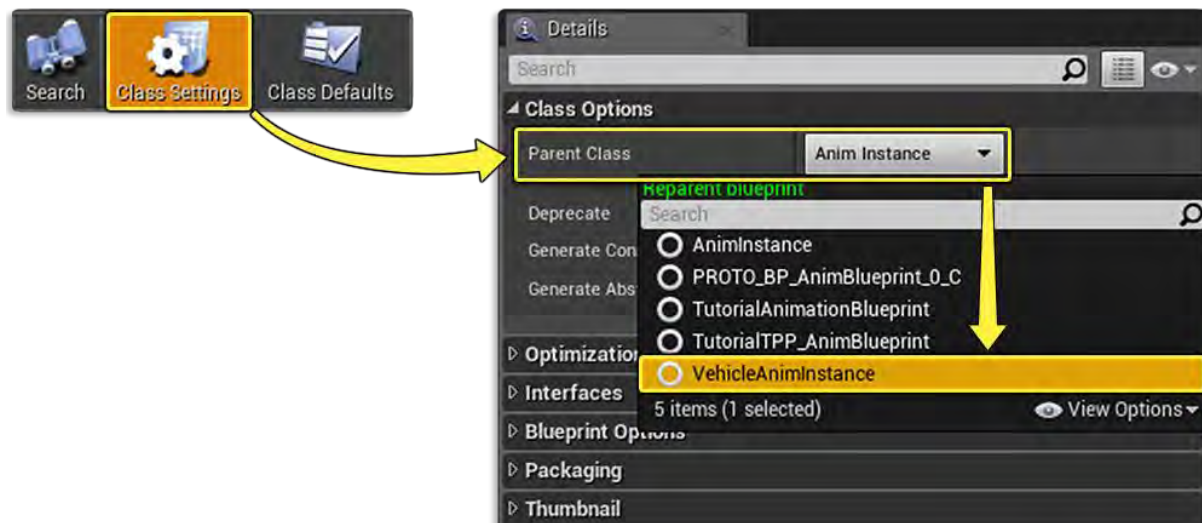
Creating the Animation Blueprint

1. In the **Content Browser**, click the **Add New** button, then hover over **Animation**, selecting **Animation Blueprint** from the menu list.
2. In the **Create Animation Blueprint** window, locate and select **your** Skeleton from your vehicle's list. Then click **OK**.

Alternatively, you can create an Animation Blueprint directly from your Skeletal Mesh by selecting it in the **Content Browser**, where you'll be able to select **Create > Anim Blueprint**.

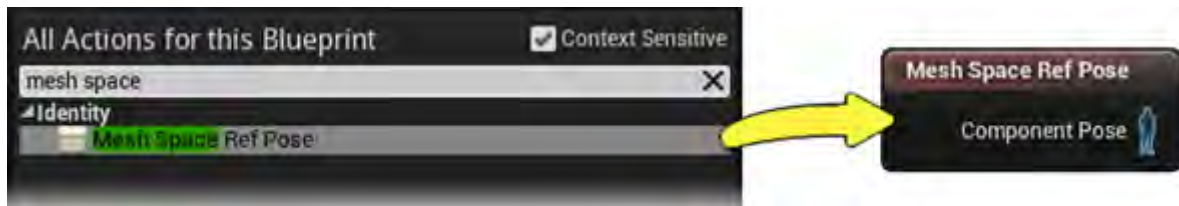


3. Now that you've got an Animation Blueprint to work from, you can open it from the **Content Browser** by selecting and double-clicking it. From the main toolbar select **Class Settings** and then in the **Details** panel set the **Parent Class** to **VehicleAnimInstance**.

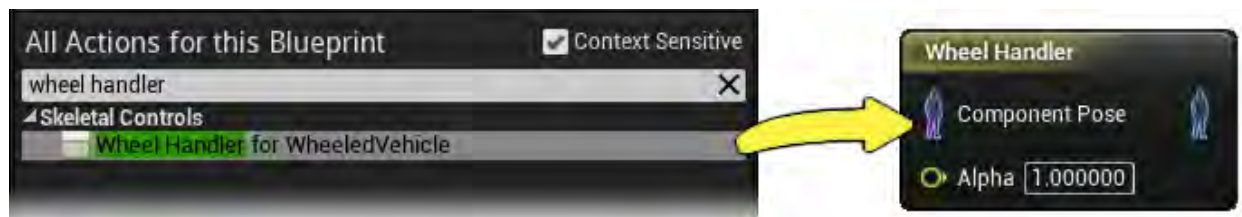


If this is not properly set to the right **Parent Class**, you will not have access to nodes needed later in this guide.

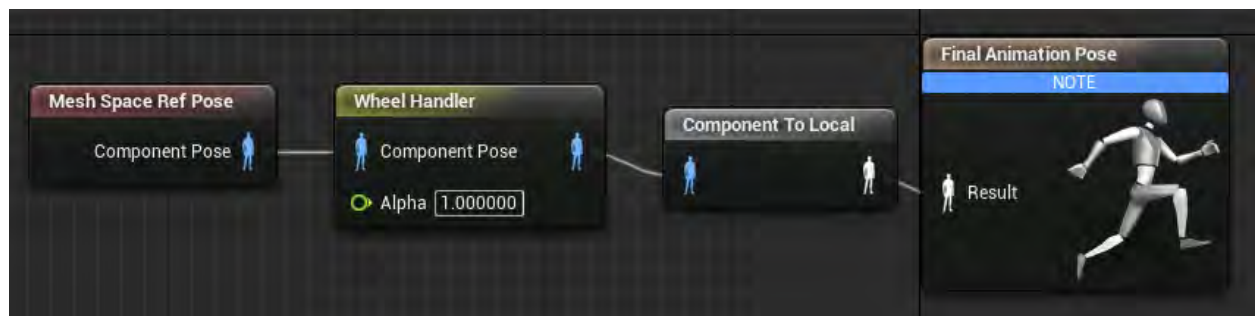
4. In the **Anim Graph**, you'll want to start by right-clicking to bring up the context menu and then searching for the **Mesh Space Ref Pose** node. Select it from the list to create it in place.



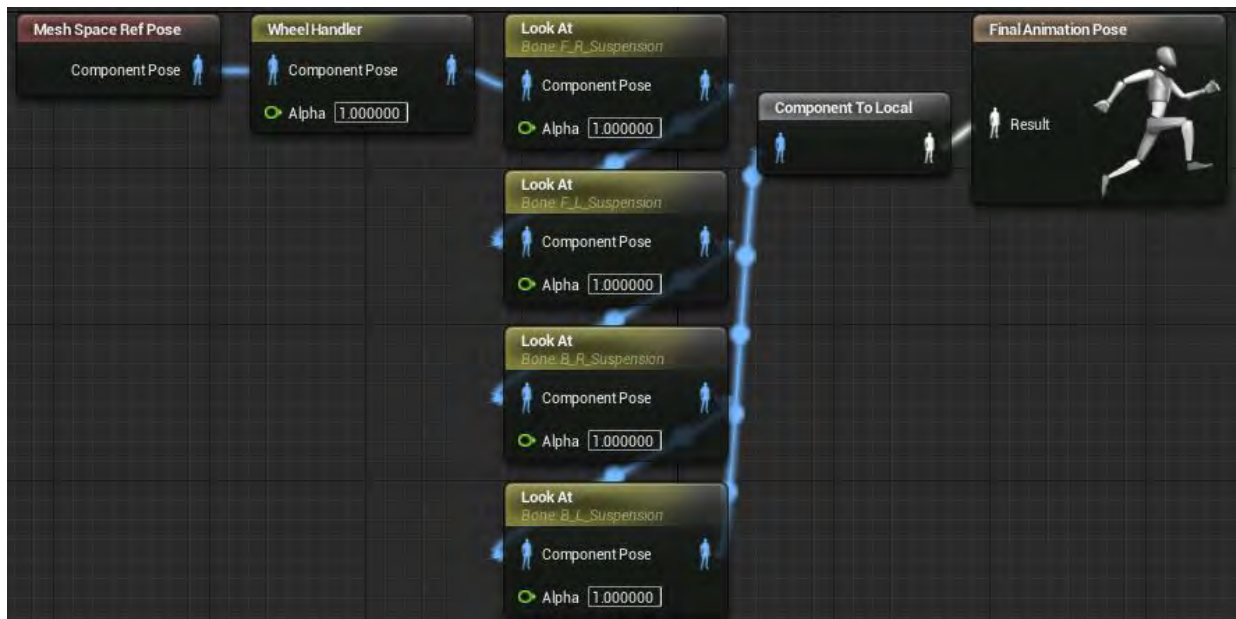
5. Next, right-click in the **Anim Graph** to bring up the context menu and then search for the **Wheel Handler**, selecting it from the list to create it.



6. Now, connect the nodes so that your **Anim Graph** looks like this one



7. Optionally, if you have additional struts or other suspension needs (like the Buggy from Vehicle Game) you will need additional nodes in your Animation Graph to handle the joints that affect those polygons. In the Buggy, from Vehicle Game, the extra joints are used to control the axle connections to the wheels. These are driven by a simple **Look At** node, which, when given the wheel joints, will be driven by the **Wheel Handler** node and the **Look At** nodes will make sure the suspension stays attached to the wheels.



End Result

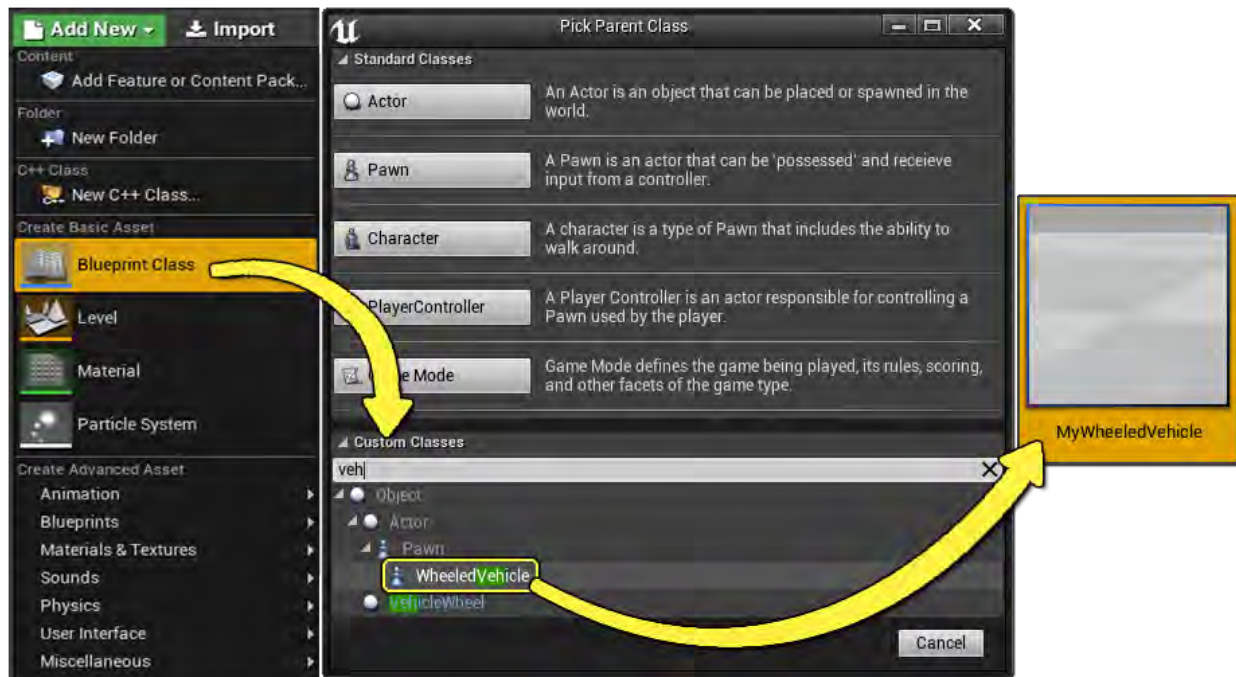
You've finished setting up the Animation Blueprint! The **Component to Local** conversion node will be created automatically and the Animation Blueprint will now drive the wheels of the vehicle. Continue on to the next step to learn how to create a Vehicle Blueprint that brings all of the previous steps together.

4 - Creating a Vehicle Blueprint

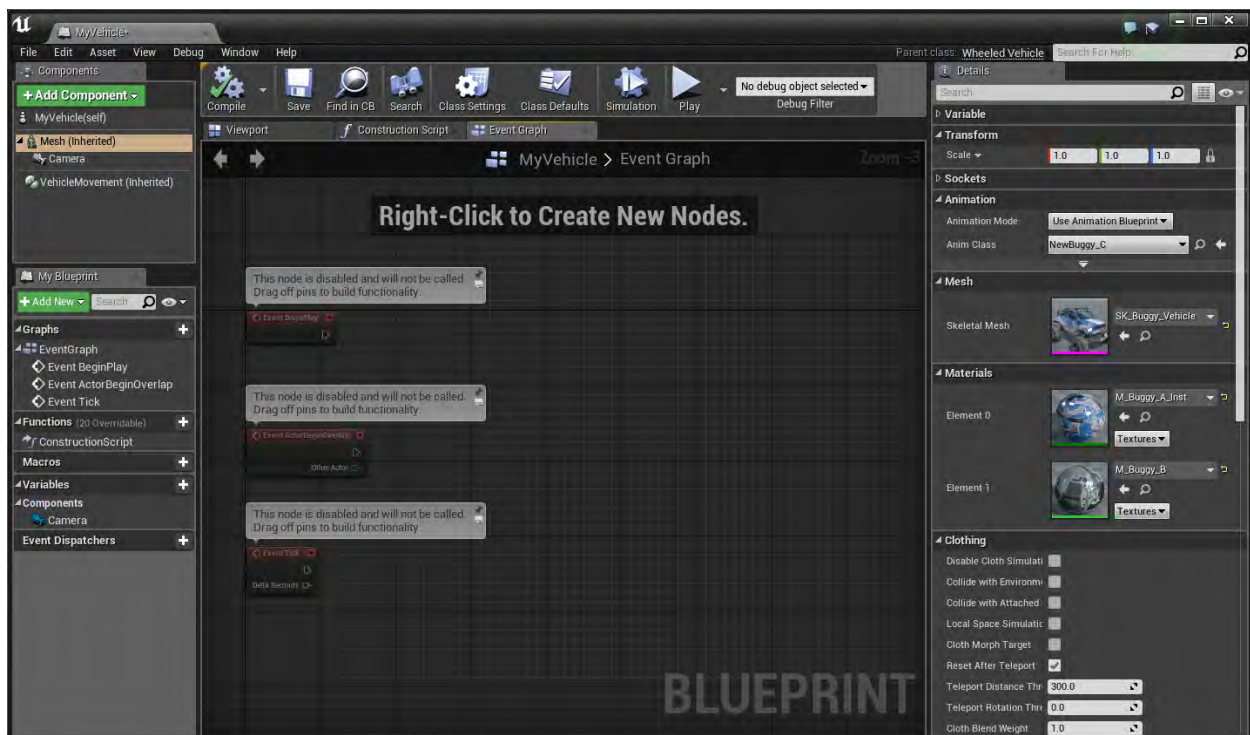
By now, you've created a few different assets (TireConfig Data Assets, Wheel Blueprints, and an Animation Blueprint), which you need to bring together to create a fully functional vehicle in Unreal Engine 4 (UE4). In this step, you'll create the Vehicle Blueprint, using those assets that were previously created and set up.

Creating and Editing the Vehicle Blueprint

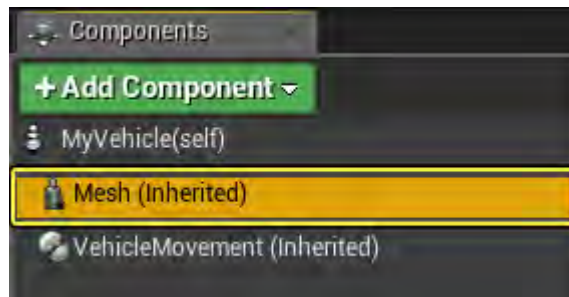
1. In the **Content Browser**, click the **Add New** button, then select **Blueprint Class** from the menu.
2. In the **Pick Parent Class** window, under **All Classes**, search for "wheel" and then select **WheeledVehicle**.
3. In your **Content Browser**, the new Blueprint asset will be created. Make sure to give it a recognizable name so that you can easily locate it later.



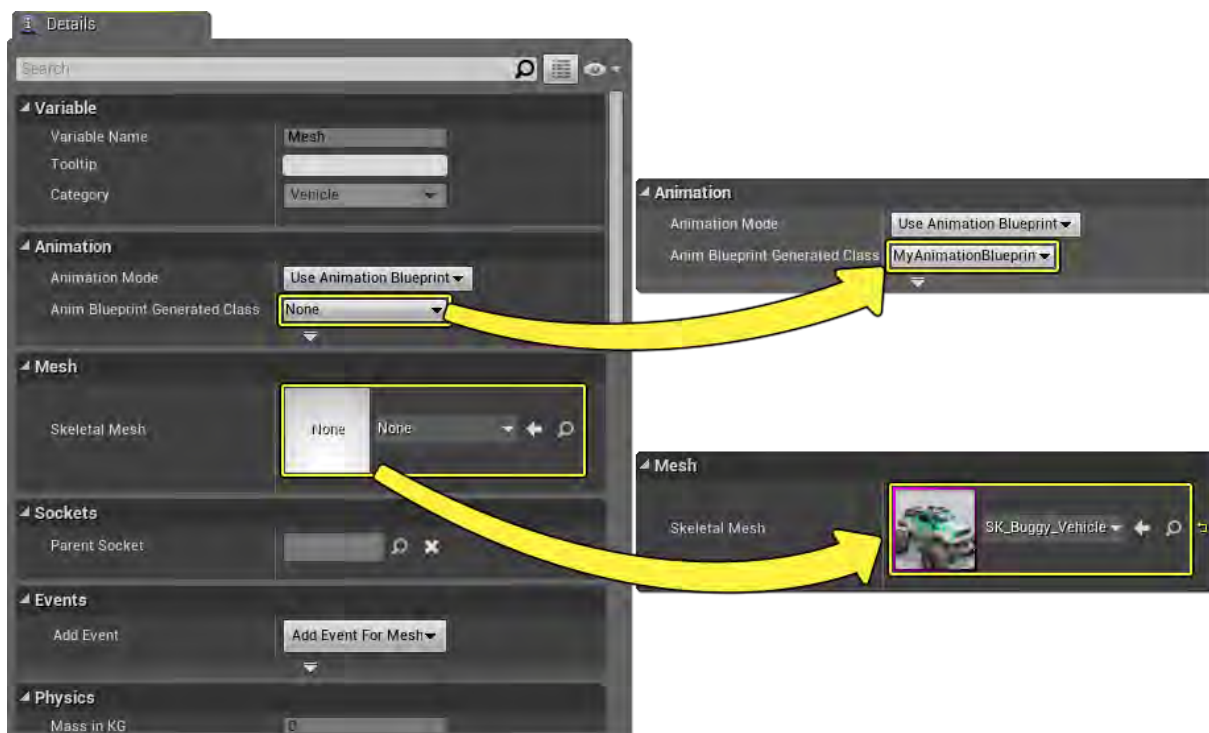
4. From the Content Browser, select and open your newly created Vehicle Blueprint.



5. Click on the **Skeletal Mesh Component** named **Mesh** from the **Components Window**. Then, in its **Details** panel, set the **Skeletal Mesh** selection box to your vehicle's own Skeletal Mesh asset.



6. Set the **Anim Blueprint Generated Class** to your vehicle's Animation Blueprint that we created in Step 2 of this guide.



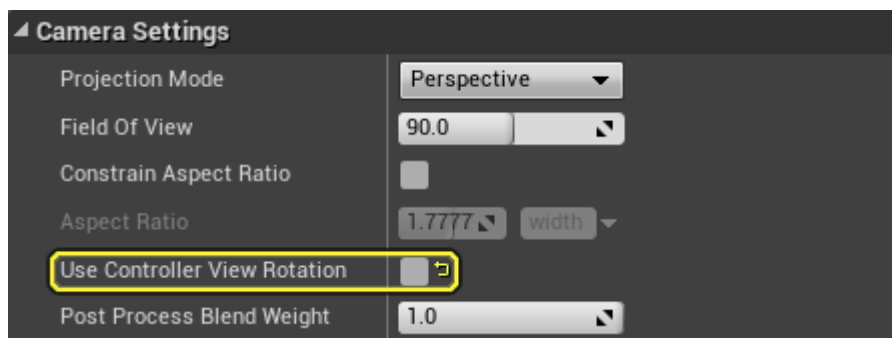
7. Go back to the **Components** window and select the **Add Component** button and select a **Camera Component**.



8. Use the Viewport in the Blueprint Editor to position the Camera where you would like it to be relative to your vehicle. In the image below, it is positioned behind, slightly raised, and tilted down towards the vehicle.



9. With the **Camera** component still selected (in the **Details** panel under **Camera Settings**), disable the option for **Use Pawn Control Rotation**. This will lock the camera to its view direction (rather than that of the Player Controller).

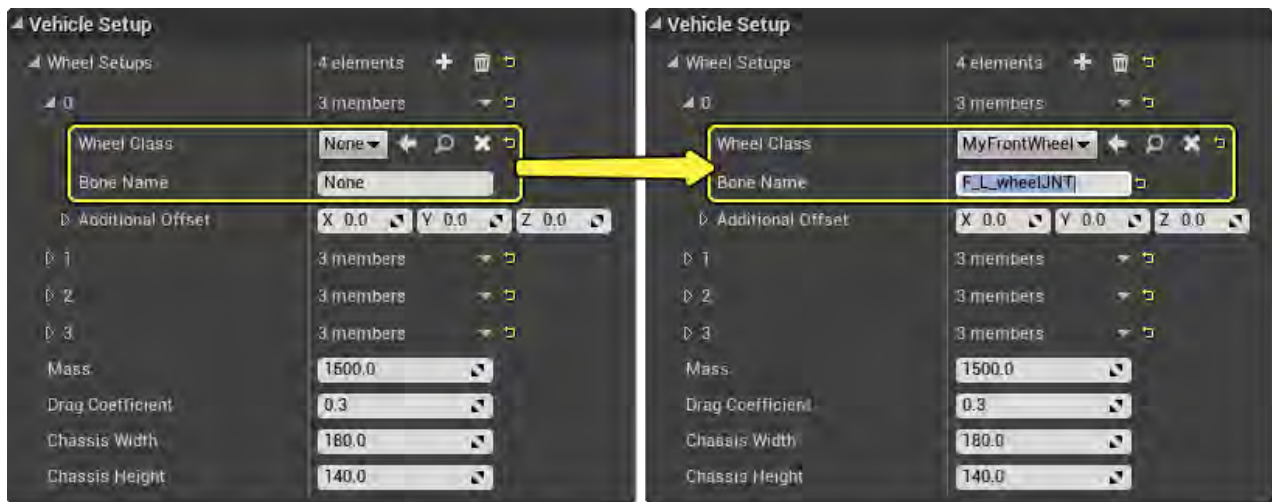


10. Next, select the **Vehicle Movement Component** in the Components window.



In the **Details** panel under the **Vehicle Setup** section, expand the arrow next to **Wheel Setups** and for each wheel (0-3) set the following:

- o Set the **Wheel Class** to the Wheel Blueprint(s) you created.
- o Set the **Bone Name** to the name of the joint that should be controlled by the wheel.



The **Bone Name** in the above image is specific to this skeletal mesh. If you're using the Vehicle from Vehicle Game, you can use these Bone Names.

The order you assign wheels has no bearing on if it is a front or a rear wheel, only the **Bone Name** and **Wheel Class** have any effect. For organizational purposes, it's best to keep the wheels (front and rear) together.

If you need more than 4 wheels, you can click the "+" icon next to the **Wheel Setups** property to add more.

End Result

At this point, assuming your vehicle is not substantially larger than a minivan, the default values will be a good starting point for you to move on to the next step in this guide, where you'll setup the inputs and controls to properly test the vehicle.

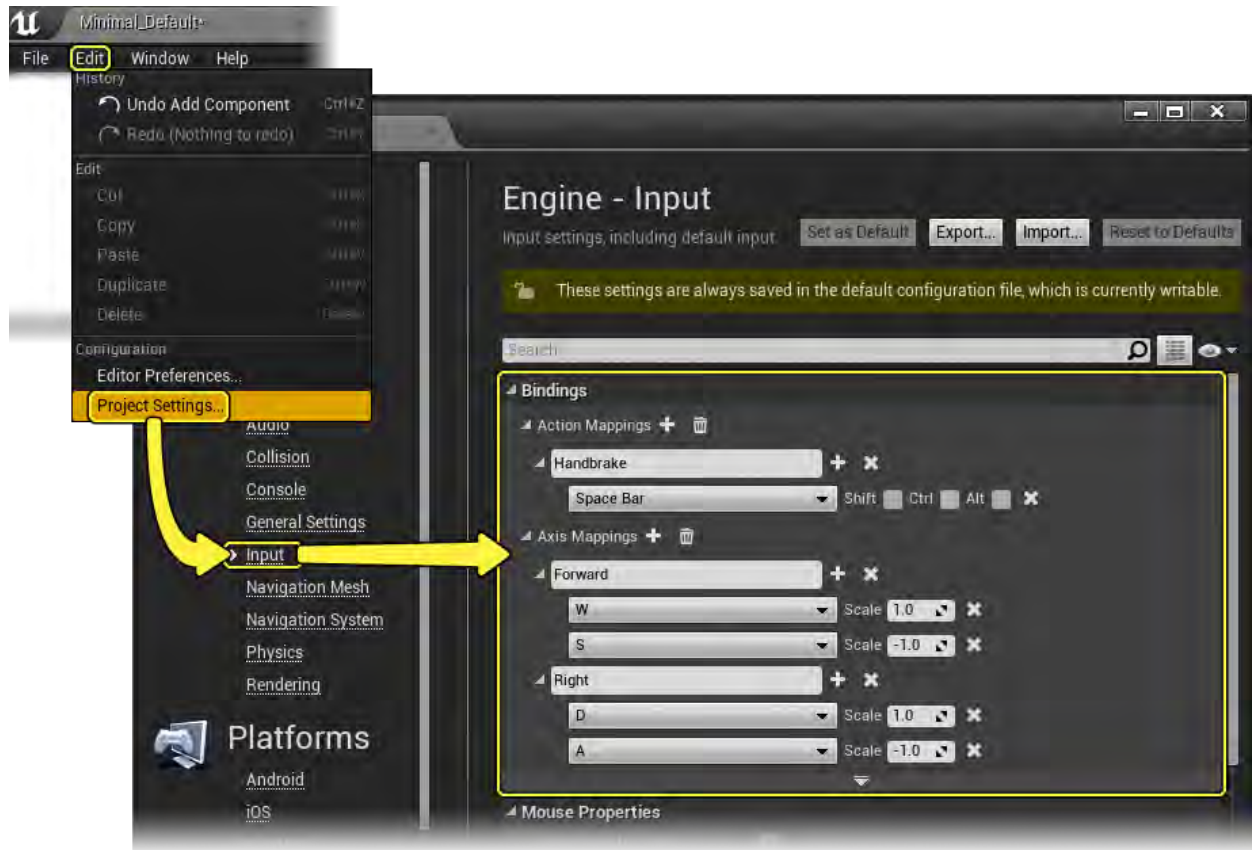
5 - Setting up and Testing Functionality

By now, you've done everything you need to do to have a fully functional vehicle by creating the Wheel, Animation, and Vehicle Blueprints in Unreal Engine 4, except to be able to control it in your game! In this step, we'll show you how to set up the necessary axis mappings and bindings so that you can drive the vehicle around and fully test its capabilities.

Because there are a few ways to go about setting up the vehicle for testing, and assuming that you started with a blank project template, all of the inputs will need to be added to control the vehicle. If you started with an existing template, some (or all) of these axis mappings and bindings may already be available for you in your project.

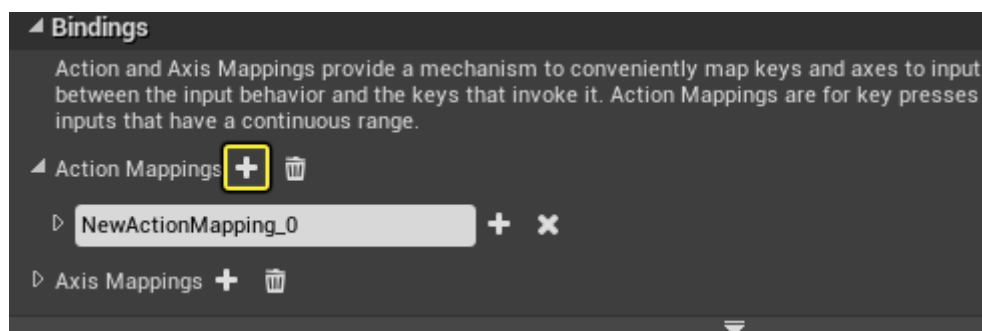
Setting Up The Controls

1. Go to the Main Menu and select **Edit > Project Settings** to open the Project Settings window. Then, under the **Engine** section in the side panel, select **Input**.

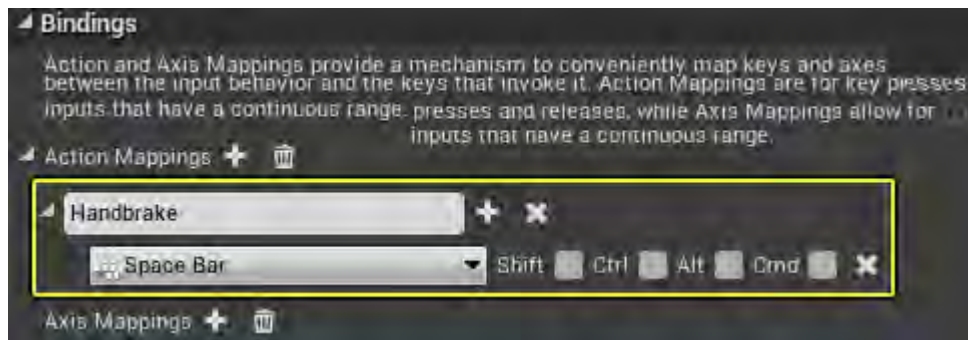


In the Bindings menu, set up the following controls (if you do not currently have them listed). If you do have them listed, it would be wise to double-check that they are set up like the ones listed here, so that you know your vehicle will work correctly.

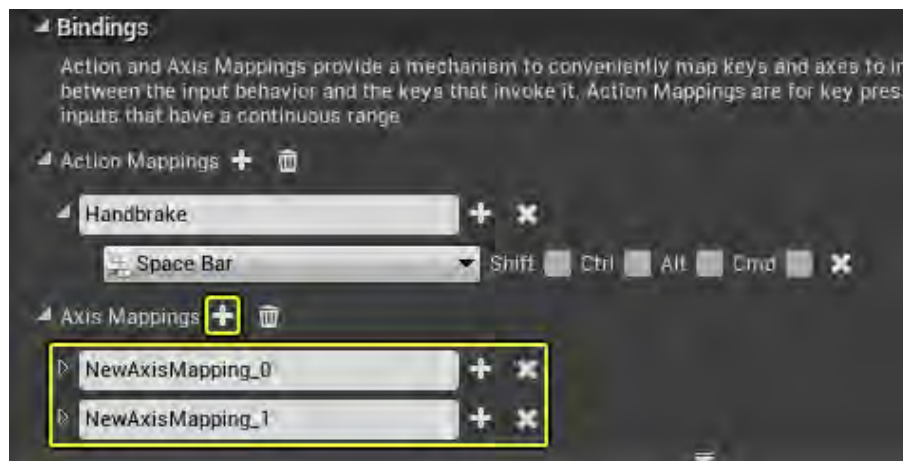
- a. First, we'll setup an **Action Mappings** by clicking the + sign next to the Action Mappings property.



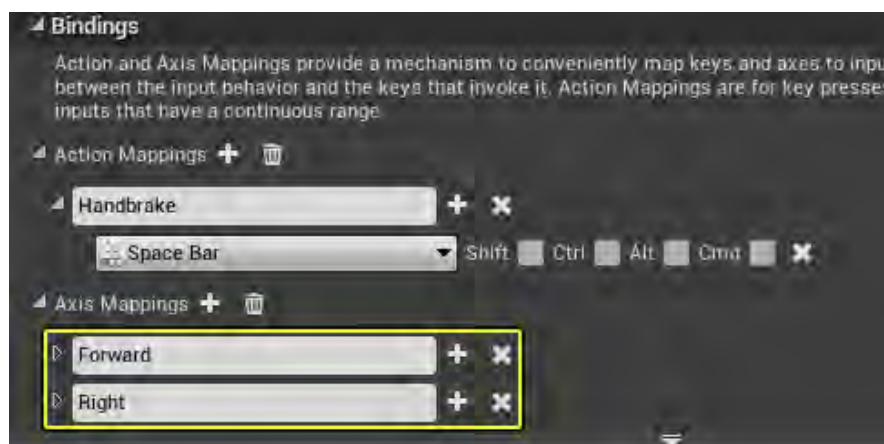
b, **Rename** the Action Mapping from "NewActionMapping_0" to **Handbrake**. Then, expand this property to use the selection box to change the key value from "None" to **Space Bar**.



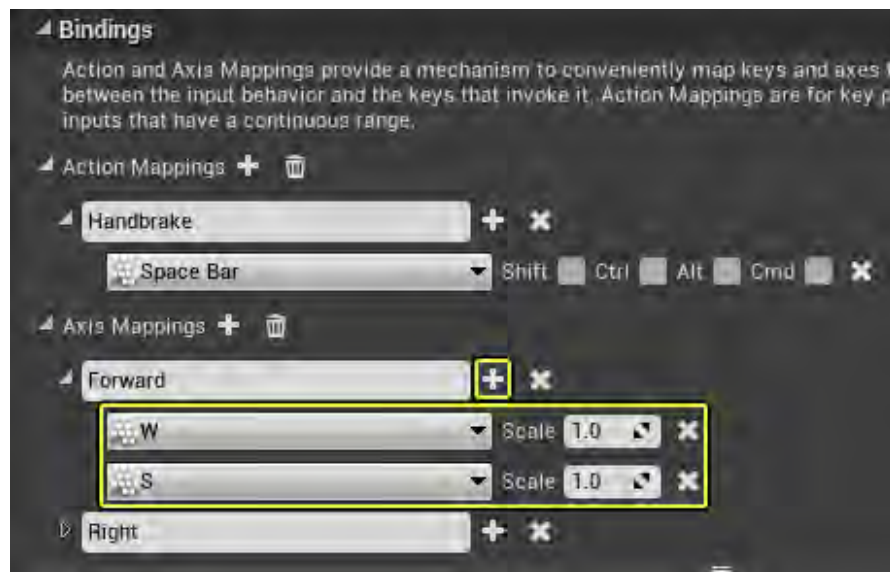
c. Next, we'll setup the **Axis Mappings** by clicking the + sign next to the Axis Mappings property. Repeat this so that you have two Axis Mappings, like this:



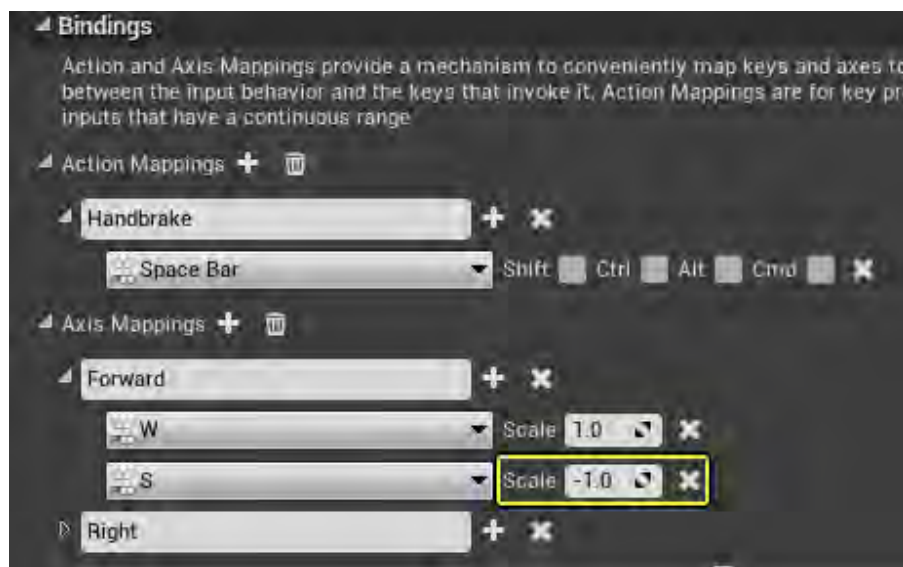
d. **Rename** the first Axis Mapping from "NewAxisMapping_0" to **Forward**. Then do the same for the second and rename it from "NewAxisMapping_1" to **Right**.



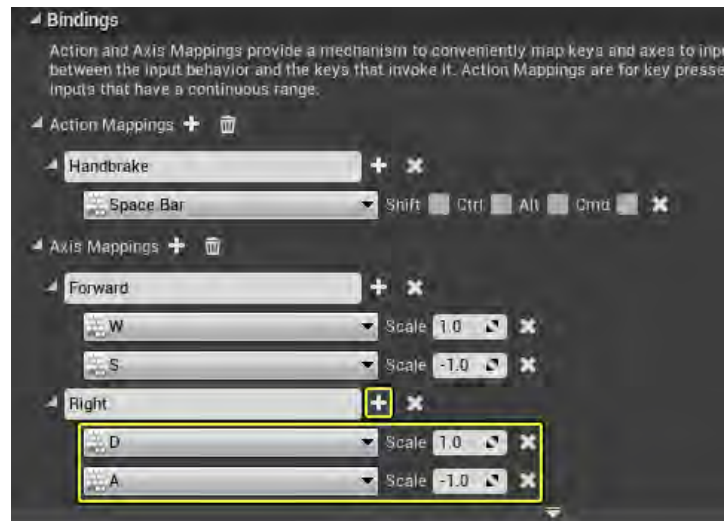
f. **Expand** the **Forward** Axis Mapping and then click the + sign once so that you have two Axis Mappings listed under "Forward". Then set the first "None" to use the **W** key for input. Set the second "None" to use the **S** key.



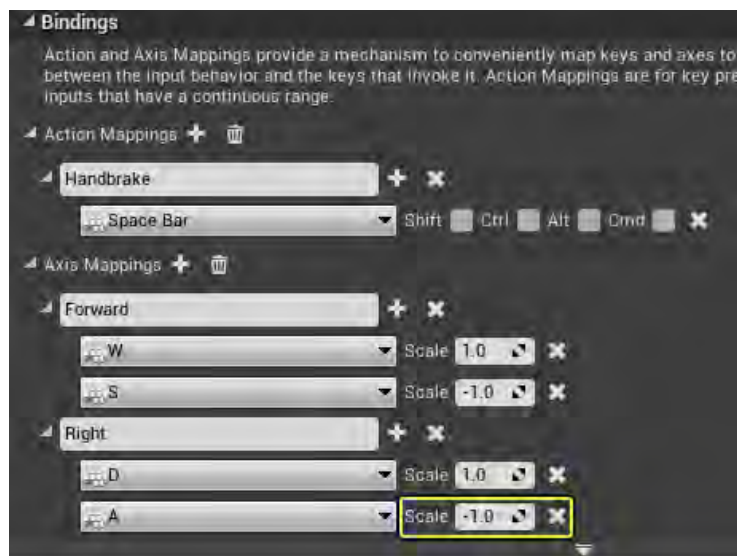
e. Next to the **S** key input under the **Forward** Axis Mapping, change the **Scale** value to be **-1**.



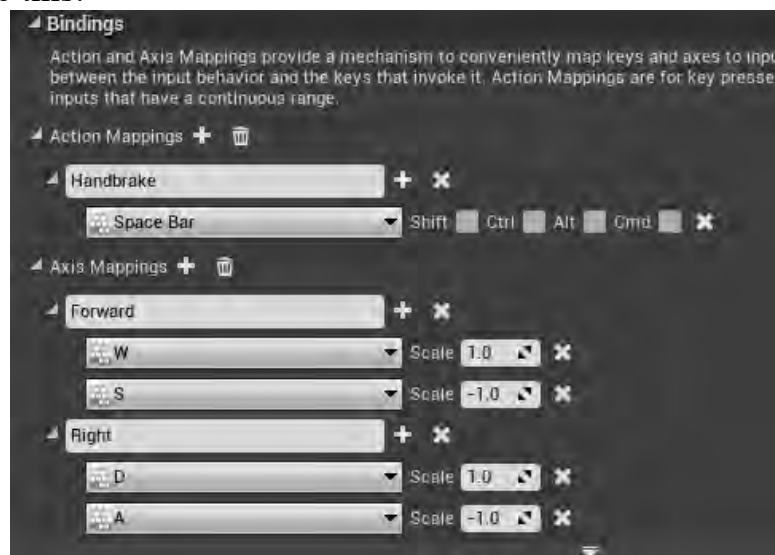
g. **Expand** the **Right** Axis Mapping and then click the + sign once so that you have two Axis Mappings listed under "Right". Then set the first "None" to use the **D** key for input. Set the second "None" to use the **A** key.



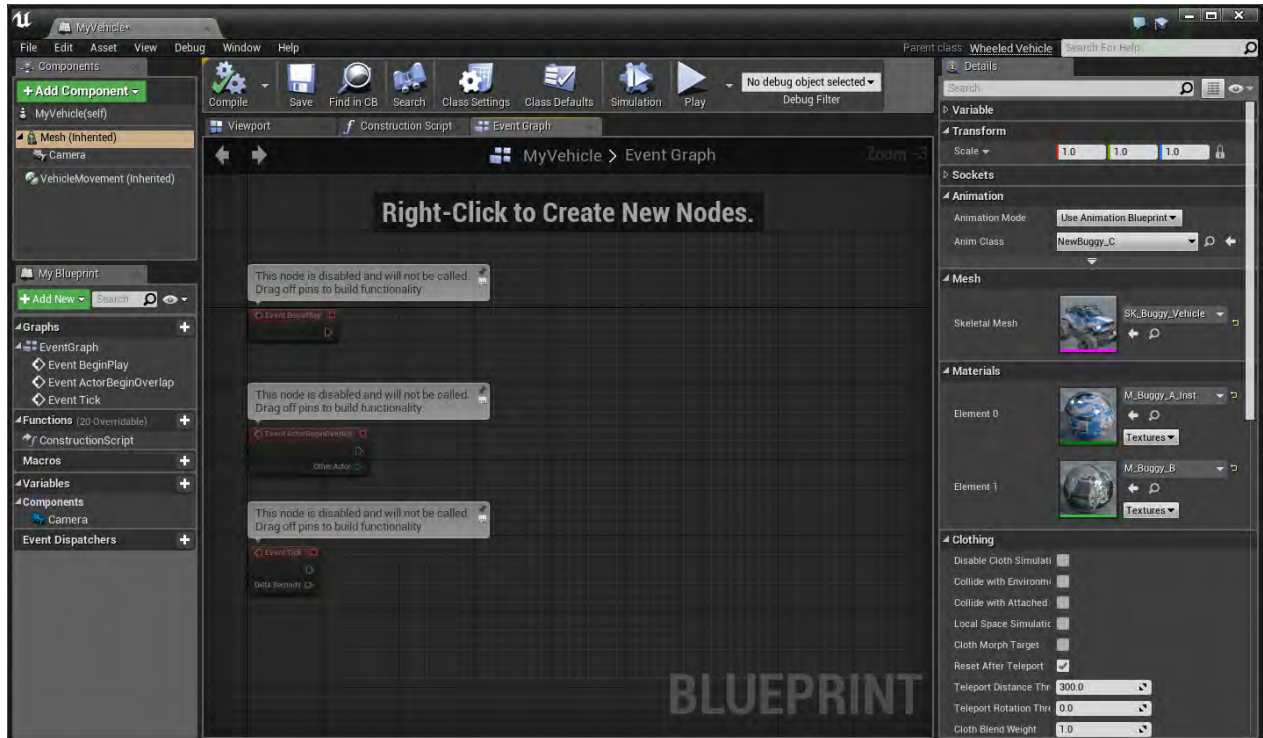
h. Next to the **A** key input under the **Right** Axis Mapping, change the **Scale** value to be **-1**.



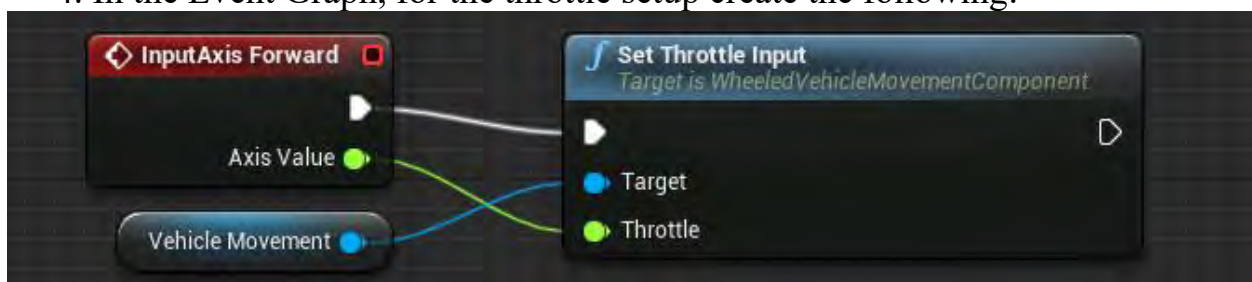
Once you're done setting the Action and Axis Mappings, your Bindings should look like this:



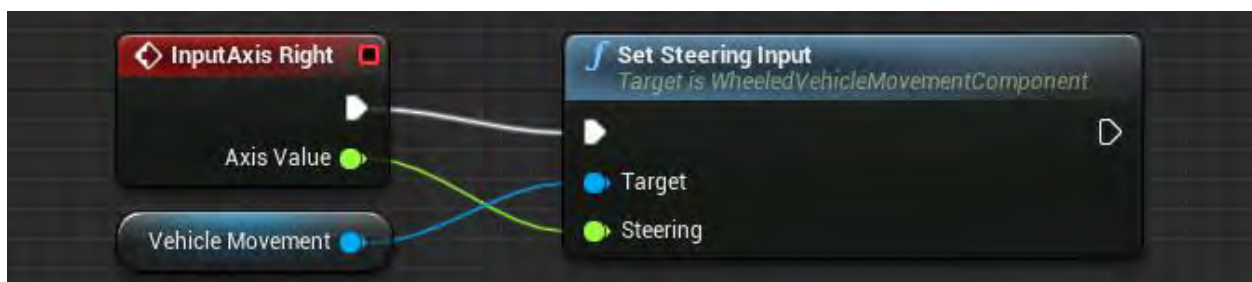
3. Now that you've got the controls set up, you'll need to actually do something with them. So, open up the **Vehicle Blueprint** that you created and locate the **Event Graph**, where you'll be able to call these Axis and Action events that you created in the Project Settings Input section.



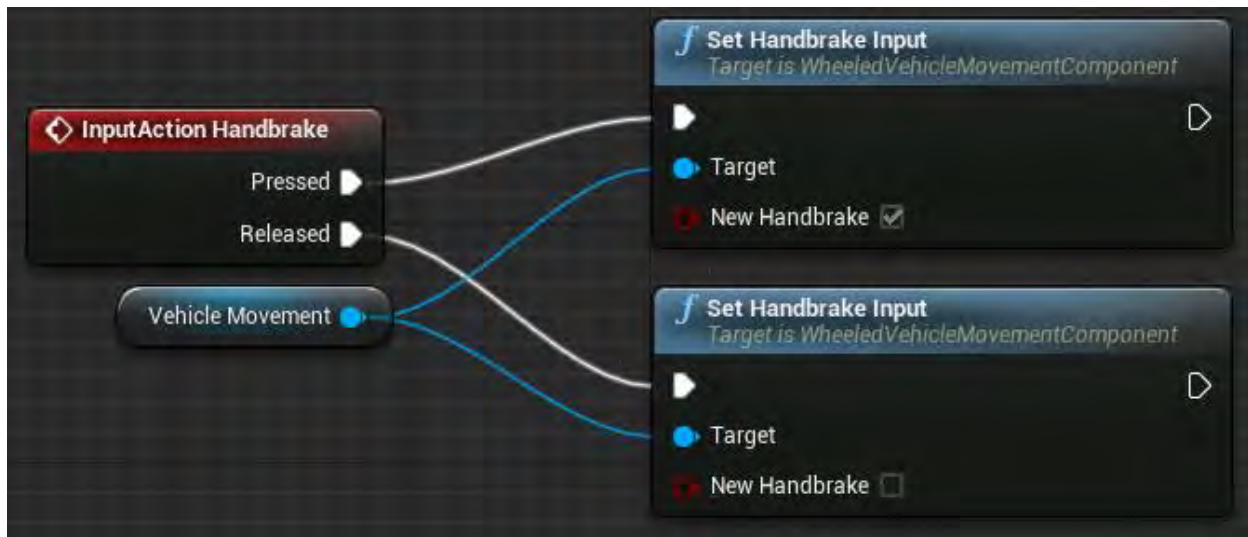
4. In the Event Graph, for the throttle setup create the following:



5. In the Event Graph, for steering setup the following:



6. In the Event Graph, for the handbrake setup the following:



End Result

Now with these Input Events added you have a fully functional vehicle that can take input movement! In the next and final step, you'll set up a new Game Mode that uses your Vehicle pawn as the default pawn when you launch a game.

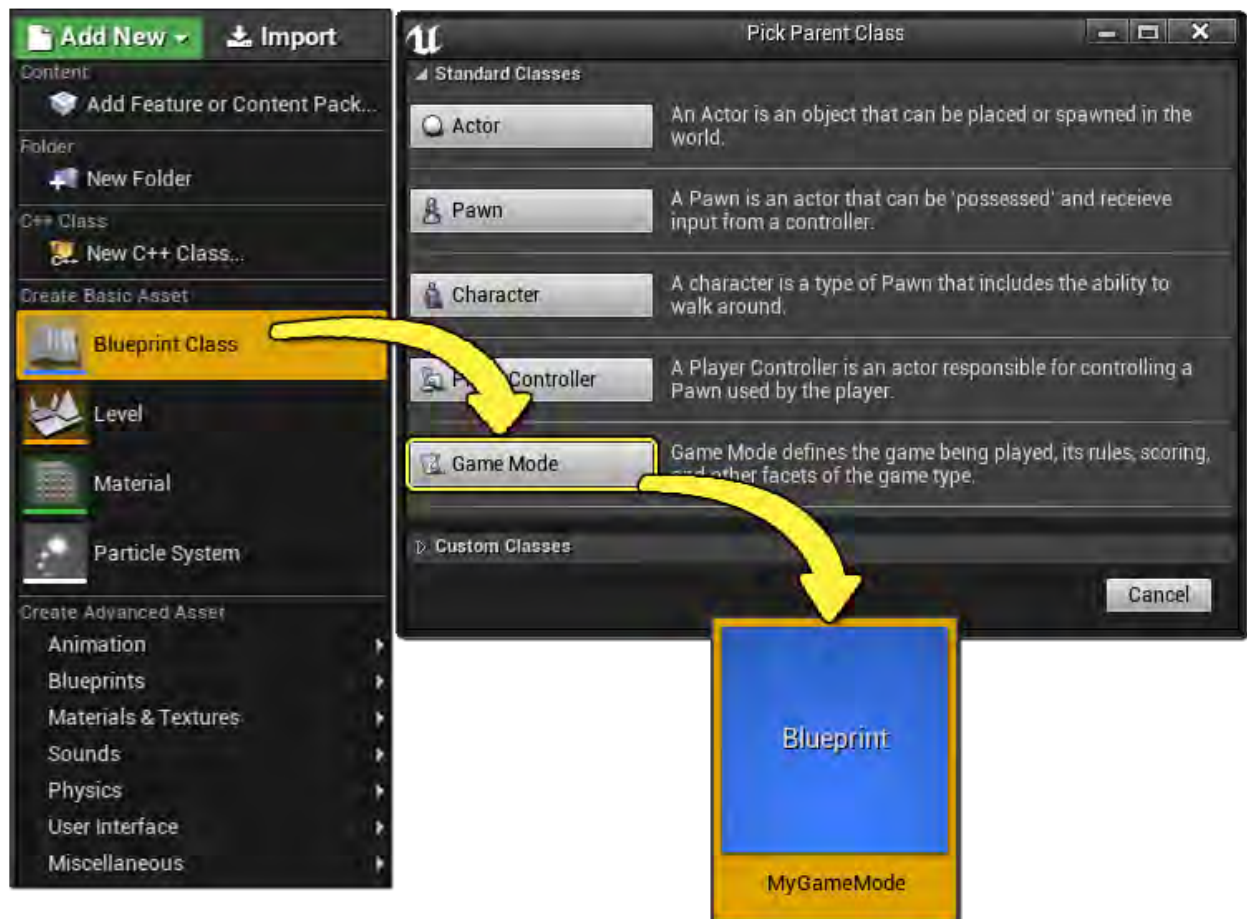
6 - Setting up a New Game Mode

1. In the previous step, you finished setting up our Vehicle Blueprint and assigned all of the controls that you need to make it work properly. In this last step, you'll finish by setting up a new Game Mode that uses your own Vehicle Blueprint as the Default Pawn for the game.

2. If you did not use a Blank Project Template when starting this guide you may find that you already have a Game Mode assigned that conflicts with the one you'll create in this step. Use the steps below to create a new Game Mode so that you can spawn in the vehicle you've created throughout this guide.

Creating a New Game Mode

1. In the **Content Browser**, click the **Add New** button and select **Blueprint Class** from the menu list.
2. In the **Pick Parent Class** window, select **Game Mode Base**, and then click the **Select** button to create the Game Mode Blueprint.



3. In the **Content Browser**, double-click to open the new Game Mode Blueprint so that you can edit it. In the **Details** under **Classes**, next to **Default Pawn Class**, use the drop-down selection to choose **Your Vehicle Blueprint** (which you created in Step 3).



After doing this, you can click **Save** and then **Close** this window.

4. In the Main Viewport window, in the **World Settings** tab under the **Game Mode** section, set **GameMode Override** to **Your Game Mode Blueprint**.



End Result



Congratulations! Now, when you Play in Editor (PIE) you should automatically spawn in with your very own vehicle that you created. You've learned everything you need to create your own working vehicles in UE4. As you worked your way through this guide, you learned:

- ✓ How to configure a TireConfig Data Asset.
- ✓ How to configure a Wheel Blueprint for your Front and Rear wheels.
- ✓ How to setup an Animation Blueprint specifically for a Vehicle.
- ✓ How to configure your Vehicle Blueprint with your Skeletal Mesh, Animation Blueprint, and Wheel Blueprint(s).
- ✓ How to configure your Axis Mappings and Bindings to control your Vehicle.
- ✓ How to create and assign a new Game Mode to spawn your vehicle.

Doing More with Vehicles in Unreal Engine

For additional resources with Vehicles you can take a look at the following pages:

- Vehicle Content Creation
- How to Build a Double Wishbone Suspension Vehicle
- Vehicle Center of Mass

Movement & Rotation in Unreal Engine

Ниже приведен материал статьи по движению, вращению в Unreal Engine, взятый в оригинале на английском из

<https://vrealmatic.com/unreal-engine/movement>

раскрывающий реализацию функций на C/C++.

Author: Vladimír Nepor. Published on 11. 8. 2023

C++ focused guide explaining how to move and rotate with actors, pawns, characters (just any movable object) in Unreal Engine.

a. Movement in Unreal Engine

Movement is usually processed in each frame, and as the frame time is not constant over time, movement is being used together with float DeltaTime value in Unreal Engine.

DeltaTime or also referred to as elapsed time is the time difference between the previous frame that was drawn and the current frame. Why is it so important? Shortly, by multiplying by **DeltaTime** we remove the dependence of the number of FPS (frames per second) on the speed of movement.

In Unreal Engine, DeltaTime is in seconds, then fps (frames per second) is calculate as **fps = 1 / DeltaTime**.

Deltatime value can be obtained by:

- It's a parameter of Tick function
- By asking UGameplayStatics::GetWorldDeltaSeconds(this);

Movement initializers

• Movement based on user's axes and actions input

SetupPlayerInputComponent function is defined in APawn class (and so it's available in all classes inherited from APawn).

• AI Movement

Functions MoveToLocation() and SimpleMoveToLocation() processes AI Movement **using a NavMesh** from current Location to a defined location.

• Movement with predefined trajectory and timing

Trajectory can be processed through AI MoveToLocation movement between points from the trajectory table or by converting the defined trajectory into inputs.

Actors Movement

Following examples presents a movement processed through functions defined in AActor class. As so, they are available in all classes inherited from AActor.

Take into notice, e.g. a Character animation is not processed automatically based on the movement processed through the function.

- **Example 1 - AActor::AddActorLocalOffset()**
- AExample::SetupPlayerInputComponent(UInputComponent* PlayerInputComponent)
- {
- Super::SetupPlayerInputComponent(PlayerInputComponent);
- **PlayerInputComponent->BindAxis(TEXT("MoveForward"), this, &AExample::MoveForward);**
- **PlayerInputComponent->BindAxis(TEXT("TurnToSides"), this, &AExample::TurnRight);**
- }
- void AExample::MoveForward(float value)
- {
- FVector DeltaLocation = FVector::ZeroVector;
- DeltaLocation.X = value * UGameplayStatics::GetWorldDeltaSeconds(this)
- * optionalSpeedCoefficient;
- bool bSweep = true;
- AddActorLocalOffset(DeltaLocation, bSweep);
- // If we work with a world space:
- //AddActorWorldOffset(DeltaLocation, bSweep);
- }
- void AExample::TurnRight(float value)
- {
- FRotator DeltaRotation = FRotator::ZeroRotator;
- DeltaRotation.Yaw = value * UGameplayStatics::GetWorldDeltaSeconds(this) * optionalTurnRateCoefficient;
- bool bSweep = true;
- AddActorLocalRotation(DeltaRotation, bSweep);
- }
- **Example 2 - AActor::SetActorLocation()**
- void AExampleClass::Tick(float DeltaTime)
- {
- FVector NewLocation = FMath::Lerp(StartLocation, TargetLocation, CurrentTime / MovementTime);
- SetActorLocation(NewLocation, bSweep);
- }

Sweeping bSweep

Sweeping is a second (optional) parameter of `AddActorLocalOffset()`, `()` and others. Sweeping is a check for an objects overlapping (collissions) while moving. When bool `bSweep` parameter is true, engine automaticly checks preset collisions configuration of Root Component of moving object before processing the movement and does not let it penetrate another object if it is not supposed to do so according to preset collisions configuration.

Pawn Movement

• Example 1 - `APawn::AddMovementInput()`

```

• ASimulationCharacter::SetupPlayerInputComponent(UInputComponent*
PlayerInputComponent)
• {
• // proces parents' classes functionality
• Super::SetupPlayerInputComponent(PlayerInputComponent);
• // keyboard - axis - deltaTime included automatically by the class function
• PlayerInputComponent->BindAxis(TEXT("MoveForward"), this,
&AExample::MoveForward);
• PlayerInputComponent->BindAxis(TEXT("MoveRight"), this,
&AExample::MoveRight);
• // keyboard - actions
• PlayerInputComponent->BindAction(TEXT("Jump"), EInputEvent::IE_Pressed, this,
&Character::Jump)
• // mouse
• PlayerInputComponent->BindAxis(TEXT("LookUp"), this,
&APawn::AddControllerPitchInput);
• PlayerInputComponent->BindAxis(TEXT("LookRight"), this,
&APawn::AddControllerYawInput);
• // controller
• PlayerInputComponent->BindAxis(TEXT("LookUpRate"), this,
&ASimulationCharacter::LookUpRate);
• }
• void ASimulationCharacter::MoveForward(float AxisValue)
• {
• AddMovementInput(GetActorForwardVector() * AxisValue);
• }
• void ASimulationCharacter::MoveRight(float AxisValue)
• {
• AddMovementInput(GetActorRightVector() * AxisValue);
• }
• void ASimulationCharacter::LookUpRate(float AxisValue)

```

```

• {
•   AddControllerPitchInput(AxisValue * RotationRate * GetWorld()-
>GetDeltaSeconds());
}

```

You may notice in the example above, that in a case of Jump we refer to ACharacter instead of to APawn as we do in a case of the looking based on mouse input. That's because Jump function is defined by AController and thus it is not available for APawn characters in default.

Example 2 - UPawnMovementComponent::AddInputVector()

Following example is used in the default **Third Person** Project Template in Unreal Engine. The pack features a playable character controlled by keyboard and mouse. Characters animations are included, processed through Movement Component that is native part of ACharacter..h

```

#pragma once
#include "CoreMinimal.h"
#include "GameFramework/Character.h"
#include "InputActionValue.h"
#include "MyProjectTmpCharacter.generated.h"
UCLASS(config=Game)
class AMyProjectTmpCharacter : public ACharacter
{
    GENERATED_BODY()
    /** MappingContext */
    UPROPERTY(EditAnywhere, BlueprintReadOnly, Category = Input, meta =
(AllowPrivateAccess = "true"))
    class UInputMappingContext* DefaultMappingContext;

    /** Jump Input Action */
    UPROPERTY(EditAnywhere, BlueprintReadOnly, Category = Input, meta =
(AllowPrivateAccess = "true"))
    class UInputAction* JumpAction;

    /** Move Input Action */
    UPROPERTY(EditAnywhere, BlueprintReadOnly, Category = Input, meta =
(AllowPrivateAccess = "true"))
    class UInputAction* MoveAction;
    /** Look Input Action */
    UPROPERTY(EditAnywhere, BlueprintReadOnly, Category = Input, meta =
(AllowPrivateAccess = "true"))

```

```

class UInputAction* LookAction;

public:
    AMyProjectTmpCharacter();

protected:

    /** Called for movement input */
    void Move(const FInputActionValue& Value);

    /** Called for looking input */
    void Look(const FInputActionValue& Value);

protected:
    // APawn interface
    virtual void SetupPlayerInputComponent(class UInputComponent*
PlayerInputComponent) override;

    // To add mapping context
    virtual void BeginPlay();
};
.cpp constructor
.cpp Movement

```

Movement by Impulse

It works based on initially provided impulse (force with direction) and then letting the physics (gravity and so on) take over

Movement component

UMovementComponentA special type of component that can be included into a class. After setting some parameters on the UMovementComponent, this component handles the movement for the actor. Unreal Engine has built-in types of movement components made for specific objects, such as:

• UProjectileMovementComponent

An Unreal Engine movement component for projectiles. It updates the position of another component during its tick.

```

// .h file
UPROPERTY(VisibleAnywhere, Category = "Movement")
class UProjectileMovementComponent* ProjectileMC;
UFUNCTION()
void OnHit(UPrimitiveComponent* HitComp, AActor* OtherActor,
UPrimitiveComponent* OtherComp, FVector NormalImpulse, const FHitResult& Hit);

```

```

// .cpp file
#include "GameFramework/ProjectileMovementComponent.h"
AProjectile::AProjectile()
{
    ProjectileMP
    CreateDefaultSubobject<UProjectileMovementComponent>(TEXT("Projectile Movement Component"));
    ProjectileMP->MaxSpeed = 1200.f;
    ProjectileMP->InitialSpeed = 2100.f;
}
AProjectile::BeginPlay()
{
    ProjectileMesh->OnComponentHit.AddDynamic(this, &AProjectile::OnHit);
//Collision Presets must be Enabled (Query and Physics) for handling Hit events
}
AProjectile::OnHit(UPrimitiveComponent* HitComp, AActor* OtherActor,
UPrimitiveComponent* OtherComp, FVector NormalImpulse, const FHitResult& Hit)
{
    // ...
}

```

a. Casting

Anytime there's need a connection or access between two classes, when one is derived from the another, there's possible to Cast from child class to parent class and each other. This can be done with code `Cast<type>(Pointer)`, see examples Getting APlayerController and APawn derived class.

Example 1 - Getting APlayerController reference

APlayerController is necessary if we need for example GetHitResultUnderCursor information. To get it, there may be used APawn::GetController() for getting controller which is a constant function of APawn class. The return type is a pointer to AController.

As our APlayerController is a derived function of AController, there's possible to **Cast<>()** from the parent AController to its child APlayerController and each other. With casting, there's possible to access properties of the required class.

```

void AExample::BeginPlay()
{
    Super::BeginPlay();
}

```

```

    // APlayerController* PlayerController; defined in .h file
    PlayerController = Cast<APlayerController>(GetController());
}
void AExample::Tick(float DeltaTime)
{
    Super::Tick(DeltaTime);
    if(!PlayerController) return;

    FHitResult HitResult;
    PlayerController->GetHitResultUnderCursor(ECollisionChannel::ECC_Visibility,
bTraceComplex, HitResult); // fills HitResult variable
    // Debug Visualization of pointer location target considering collisions
    DrawDebugSphere(GetWorld(), HitResult.ImpactPoint, 20.f, 12, FColor::Red, false);
    RottateTo(HitResult.ImpactPoint);
}
void AExample::RottateTo(FVector LookAtTarget)
{
    // Rotate from current rotation to aim to pointer location
    // ToTarget = TraceHitLocation - CurrentObjectLocation
    FVector ToTarget = LookAtTarget - ObjectMesh->GetComponentLocation();
    FRotator LookAtRotation = ToTarget.Rotation();
    ObjectMesh->SetWorldTotation(FMath::RInterpTo(ObjectMesh-
>GetComponentRotation(), LookAtRotation, UGameplayStatics::GetWorldDeltaSeconds(this),
10.f));
}

```

Example 2 - Getting APawn derived class refference

AExamplePlayerPawn is derived from APawn that includes construction of function GetPlayerPawn() that we need. As our AExamplePlayerPawn is derived from APawn, there's possible to cast.

```

// .h file
class AExamplePlayerPawn* PlayerPawn;
// .cpp file
PlayerPawn = Cast<AExamplePlayerPawn>(UGameplayStatics::GetPlayerPawn(this,0));

```

b. Local / World Transforms (Location, Rotation, Scale properties)

Each Actor-type of asset has an own Transform while within the Actor, there may be many components in a hierarchical structure, that have own Transform.

As clear, a Transform or each Component in the hierarchy has a parent with an own Transform. How do these "hierarchical" Transform work?

Relative position of each actor is the position to its parent actor. In the case of the image above, the BP_Character has a position relative to the world and holding a gun within a component attached to the class that has relative position to the BP_Character.

Local Coordinate System

Looking at the coordinate system, when a parent actor (BP_Character) moves or rotates, all its child actors (Gun) moves automatically as well as they have relative position to their parent (BP_Character).

Transforming between Global and Local space

- Global space has the base at the center of the world and the Actor is relative
- Local space has the base at the Self actor and all other is relative

It is important to distinguish between the Global and Local space especially when deal with a rotation - e.g. an angle of aiming. If a character turn around for any angle, the directions changes. Example below shows a state after turning around 180 degrees - if we would not reflect local world transform, there would be played right animation for moving left and left animation for moving right according to the position change in the Global Space.

Vector Transform

There's need to take care about the Space in which BP nodes returns the data. Usually we would get Transform in the Global space while e.g. animations work with local space. To achieve a proper functionality, there's need to convert returned Global Space to Local Space. Unreal Engine has 2 nodes for that. We use Inverted value as we transform from global to the local space.

- Inverse Transform Direction for Inverted Direction Vector Transform. Position Vector is basically a Vector that provides a difference between the current location (or zero point in the coordination system) and an arrow that points to where something should be - let's say gun. If transformed to the Global World, the arrow would go from 0,0,0 Vector of The World to the Gun again. As we need the opposite, we use the inverted value. Shortly, it's full transform.

- Inverse Transform Location for Inverted Position Vector Transform that tells us where and how fast the character should be going in the direction. Shortly, it informs only about rotation and scale.