

МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
имени М.В. ЛОМОНОСОВА  
МЕХАНИКО-МАТЕМАТИЧЕСКИЙ ФАКУЛЬТЕТ

На правах рукописи

\_\_\_\_\_

**Чертополохов Виктор Александрович**

**ВИЗУАЛЬНАЯ И ДИНАМИЧЕСКАЯ СОСТАВЛЯЮЩИЕ  
ИМИТАЦИИ УПРАВЛЯЕМОГО ДВИЖЕНИЯ  
НА СТЕНДАХ-ТРЕНАЖЕРАХ**

1.1.7. Теоретическая механика, динамика машин

**ДИССЕРТАЦИЯ**

на соискание учёной степени

кандидата физико-математических наук

Научные руководители:

доктор физико-математических наук, профессор

Александров Владимир Васильевич

доктор физико-математических наук

Лемак Степан Степанович

Москва — 2023

## Оглавление

	Стр.
Введение . . . . .	5
<b>Глава 1. Задача динамической имитации управляемого движения . . . . .</b>	<b>17</b>
1.1 Устройство вестибулярного аппарата . . . . .	17
1.2 Физическая постановка задачи динамической имитации вектора гравито-инерциальных сил, действующих на пилота .	18
1.2.1 Силы, действующие на виртуального пилота в модели самолета . . . . .	18
1.2.2 Задача динамической имитации вектора гравито-инерциальных сил, действующих на пилота . .	23
1.3 Алгоритм динамической имитации ориентации вектора перегрузки . . . . .	28
1.4 Кинематика трехстепенной платформы опорного типа . . . .	30
1.5 Алгоритм предварительной идентификации геометрических параметров подвижного стенда . . . . .	32
1.5.1 Постановка задачи поиска геометрических параметров стенда-платформы . . . . .	33
1.5.2 Алгоритм идентификации геометрических параметров подвижного стенда . . . . .	35
1.6 Построение границы множества допустимых положений для трехстепенной платформы опорного типа . . . . .	37
1.6.1 Задача построения рабочей области для трехстепенной платформы опорного типа . . . . .	40
1.7 Структура алгоритма динамической имитации быстрых угловых и линейных движений . . . . .	45
1.7.1 Построение простейшего алгоритма оптимальной остановки на границе и возврата в начало координат .	48
1.8 Реализация алгоритма имитации быстрых линейных и угловых движений на примере трехстепенной платформы опорного типа . . . . .	50

1.8.1	Синтез алгоритмов динамической имитации . . . . .	50
1.8.2	Экспериментальные результаты . . . . .	51
<b>Глава 2. Визуализация виртуального окружения при</b>		
	<b>управлении динамической системой . . . . .</b>	<b>54</b>
2.1	Принципы создания визуализации . . . . .	54
2.2	Задача построения визуализации . . . . .	59
2.2.1	Математический аппарат задания преобразований пространства . . . . .	59
2.2.2	Визуализация сцены с учетом положения виртуальной камеры . . . . .	62
2.3	Фовеальный рендеринг как метод повышения частоты кадров при визуализации . . . . .	65
2.3.1	Принцип работы окулографической системы . . . . .	66
2.3.2	Реализация фовеального рендеринга . . . . .	68
<b>Глава 3. Задача синхронизации визуальной и динамической</b>		
	<b>составляющих имитации . . . . .</b>	<b>72</b>
3.1	Постановка задачи синхронизации визуальной и динамической составляющих имитации . . . . .	75
3.2	Отслеживание угловых движений человека и стенда с применением оптической системы . . . . .	81
3.3	Построение прогноза углового движения подвижного стенда с учетом его динамики и времени задержки системы отслеживания движений . . . . .	87
3.3.1	Численный эксперимент . . . . .	88
3.3.2	Сравнение алгоритмов прогнозирования . . . . .	89
3.4	Построение управления стендом с учетом порога чувствительности вестибулярного аппарата . . . . .	91
3.4.1	Постановка задачи управления с использованием измерения ускорения и учетом порога чувствительности вестибулярного аппарата человека . . . . .	92
3.4.2	Структура скользящего управления . . . . .	94
3.4.3	Робастное управление с использованием интегрального скользящего режима . . . . .	101

<b>Глава 4. Реализация алгоритмов синхронизации</b>	
<b>динамической и визуальной составляющих</b>	
<b>имитации для стенда на базе многозвеного</b>	
<b>робота-манипулятора . . . . .</b>	<b>107</b>
4.1 Модель подвижного стенда . . . . .	107
4.1.1 Зависимость ускорений расчетной точки от перемещений звеньев манипулятора . . . . .	110
4.1.2 Динамическая модель робота-манипулятора . . . . .	111
4.1.3 Определение ограничений на перемещение расчетной точки в вертикальной плоскости . . . . .	112
4.1.4 Алгоритм управления остановкой на границе множества допустимых положений и возврата на стартовую дугу . . . . .	115
4.1.5 Алгоритм торможения и возврата . . . . .	117
4.1.6 Определение времени переключения управления . . . .	120
4.1.7 Построение рабочей области стенда. . . . .	125
4.2 Численное моделирование управления роботом-манипулятором с учетом порога чувствительности вестибулярного аппарата . . . . .	127
4.3 Скользящее управление с применением регуляризации на границе множества допустимых положений . . . . .	132
4.4 Реализация симулятора в Unity 3D и анализ результатов . . .	134
4.5 Методика максиминного тестирования качества управления механическими системами . . . . .	136
4.5.1 Седловая точка дифференциальной игры первого этапа методики тестирования . . . . .	139
4.5.2 Редукция к геометрической игре для программных стратегий тестирования . . . . .	140
4.6 Тестирование качества управления роботом-манипулятором .	141
<b>Заключение . . . . .</b>	<b>145</b>
<b>Обозначения . . . . .</b>	<b>148</b>
<b>Список литературы . . . . .</b>	<b>149</b>

## Введение

Появление новых динамических и визуальных симуляторов для имитации различных процессов стало важным шагом в развитии тренажерных систем различных типов. Создаваемые тренажеры предназначены для обучения операторов решению широкого спектра задач, связанных с управлением сложными системами. Наиболее эффективны тренажеры для моделирования ситуаций, которые влекут за собой существенные издержки при ошибочных действиях оператора. Одной из сфер деятельности человека, требующей углубленной подготовки, является управление летательными аппаратами. Использование авиатренажеров — общепринятая мировая практика при подготовке пилотов гражданской авиации. Полет не является естественным методом передвижения для человека. Следовательно, любое сложное действие во время полета может вызвать затруднения у неподготовленного пилота.

Одной из задач обучения является выработка соответствующей сенсомоторной реакции у обучаемого на информацию, поступающую по разным каналам восприятия. Во время полета пилот ориентируется не только на визуальную информацию, но и на информацию, получаемую посредством проприоцептивных и вестибулярных механизмов. Тренировка сенсомоторных навыков производится при помощи систем создания динамических стимулов — подвижных стендов-тренажеров. Эти системы имитируют влияние возникающих в полете ускорений на чувствительные органы человека — механорецепторы (в число которых входит и вестибулярный аппарат человека).

Попытки создания систем динамической имитации начались вскоре после появления первой авиационной техники и возникновения понимания о возможности полета аппаратов тяжелее воздуха. Так первые центрифуги — прототипы систем имитации подъема на орбиту Земли, были сконструированы К.Э. Циолковским в 1879-1903 годах [108] для проверки влияния перегрузок на животных. Прообраз современных авиационных симуляторов был создан С. Тичером в 1910 году [38] и представлял собой компоненты биплана, поставленные на бочкообразные конструкции. Система приводилась в движение с помощью рычагов двумя операторами и, естественно, не была синхронизирована с какой-либо математической моделью самолета. Первым

коммерческим динамическим стендом стал симулятор Э. Линка [38], имитирующий полет по приборам для различных моделей летательных аппаратов, созданный в 1929 году. Наиболее распространенный в наше время вид авиасимуляторов основан на шестистепенной платформе Гью-Стюарта, предложенной независимо в 1954-1967 годах В.Е. Гью, К. Каппелем и Д. Стюартом [90].

Для расширения возможностей тренажеров требуется применение нескольких типов имитационных воздействий одновременно. Создание новых образцов авиационной и космической техники повлекло за собой необходимость совершенствования комплекса подготовки к полету с использованием новейших достижений в области технологии визуальной имитации. Несмотря на то, что кинематография была изобретена братьями Люмьер в конце 19 века [106], видоизменять демонстрируемое изображение в соответствии с математической моделью имитируемого объекта стали относительно недавно [29].

Классической конструкцией тренажерного стенда можно считать систему, представляющую собой установленный на динамическую платформу макет кабины в натуральную величину [110] вместе с размещенной в нем системой визуализации. Дальнейшее развитие систем визуализации и прогресс в области разработки человеко-машинных интерфейсов привели к появлению более компактных систем виртуальной реальности. А. Сазерленд в 1966 году начал разработку первого шлема виртуальной реальности. В 1968 году был создан головной дисплей, изображение на котором управлялось с помощью компьютера [54]. Кроме того, в шлеме была применена система синхронизации изображения по данным с датчиков, регистрирующих движение головы пользователя. Первые шлемы виртуальной реальности были настолько тяжелыми, что их приходилось подвешивать к потолку. Задержка в изменении изображения у ранних образцов могла составлять около секунды. Тем не менее, на базе этих технологий в Национальном управлении по авиации и исследованию космического пространства (NASA) США в 70-х годах начались исследования в области использования виртуальной реальности для подготовки космических экипажей [15].

Одновременно развивались методы создания систем подготовки к космическому полету, в разработке которых участвовал Московский Университет. В 1977 году, по инициативе Центра подготовки космонавтов СССР, в МГУ была сформирована научная группа, специализирующаяся на стыке

двух дисциплин - биомеханики и математики. Группа возглавлялась В.А. Садовничим и В.В. Александровым, а основной целью ее работы стало разработка математического обеспечения авиационных и космических тренажеров.

Коллектив успешно решил задачу динамической имитации подъема космического корабля и воссоздания "физиологической невесомости" в земных условиях с использованием центрифуги ЦФ-18 [108]. За эту работу в 1988 году была присуждена Государственная премия СССР. Следует отметить, что на тот момент при моделировании подъема пилотируемого космического аппарата на орбиту Земли визуальная имитация не требовалась. Однако, в последующих работах коллектива, посвященных имитации управляемого спуска с орбиты, необходимость хотя бы простейшей визуализации уже учитывалась [73].

В 1991 году в структуре механико-математического факультета МГУ была создана лаборатория математического обеспечения имитационных динамических систем (МОИДС) под руководством В.В. Александрова. Коллектив лаборатории продолжил работы в области динамической имитации и начал исследования в области визуальной имитации процесса управления транспортными средствами [76; 86]. В 2013 году лабораторией была введена в эксплуатацию Панорамная система виртуальной реальности (ПСВР) МГУ, включающая экранную систему визуализации и трехстепенную подвижную платформу [70; 92]. Настоящая научная работа посвящена созданию математического и программного обеспечения для ПСВР и других перспективных систем виртуальной реальности.

В данной работе под термином виртуальной реальности понимается набор технологий, которые построены на обратной связи между человеком и виртуальным пространством, синтезируемым компьютером. Для обеспечения обратной связи с помощью специализированных устройств воспроизводятся изображения и создаются динамические/прочие имитирующие воздействия, которые полностью или частично замещают реальные ощущения человека (в том числе перегораживают поле зрения). Под виртуальным пространством будем понимать воспроизводимую на компьютере модель среды, которую требуется симитировать для человека, и с которой необходимо обеспечить взаимодействие [49]. Хотя первые исследования в области виртуальной реальности проводились еще в 60-е годы, массовое практическое приме-

нение технология получила только в последние десятилетия [41; 45; 47; 64]. На данный момент одной из основных тенденций развития систем виртуальной реальности стало увеличение интерактивности моделируемой среды. Перспективы применения виртуальной реальности в тренажерах представлены в работах [95; 96].

Интерактивная виртуальная реальность должна быть способна предоставлять пользователю возможность взаимодействовать с любым виртуальным объектом в пределах его досягаемости и давать возможность управлять смоделированными сложными системами. Манипулирование объектами осуществляется по законам синтезированного мира с помощью специализированных систем отслеживания движений человека, по показаниям которых перестраивается виртуальное изображение. Как отмечалось ранее, типичный подход к созданию стендов-тренажеров заключался в размещении полноразмерного макета кабины и экрана системы визуализации непосредственно на движущейся платформе. Однако в последнее время появляется все больше альтернативных технологических решений, в которых система визуализации не связана жестко с платформой:

- экран визуализации остается неподвижным, а изображение корректируется в соответствии с положением платформы [83];
- система визуализации фиксируется на голове пользователя (например, сочетание стенда на базе робота-манипулятора [3; 4; 35; 37; 60] и шлема виртуальной реальности).

В результате возникает потребность не только в адекватном управлении платформой для создания динамических стимулов, но и в обеспечении согласованности между визуальными и динамическими воздействиями. В дальнейшем, термин "синхронизация визуальных и динамических воздействий" будет означать "согласование", то есть не только совмещение по времени, но и пространственную координацию. Рассогласование имитационных воздействий выше допустимых пороговых значений существенно снижает эффективность тренажера и может вызывать "болезнь движения" [21; 93], что становится существенным ограничением применимости подобных систем. Несмотря на наличие публикаций о стендах-тренажерах, основанных на передовых технологиях виртуальной реальности [31; 57] и ряде исследований на тему рассогласования воздействий в стендах [13; 36], в открытых источниках не были найдены полные алгоритмы синхронизации визуальной и динамиче-



ской имитации, а только упоминания о необходимости их использования [24]. В связи с этим, рассмотрим достижения индустрии в решении некоторых локальных подзадач, входящих в состав задачи синхронизации: в частности, задачи идентификации геометрических параметров стенда и задачи построения устойчивого к динамическим возмущениям (робастного) управления стендом.

Обе вышеуказанные задачи возникают из-за отсутствия точной информации о параметрах динамического стенда. Можно выделить ряд работ, посвященных решению задачи идентификации *геометрических* параметров стенда-платформы и отслеживанию ее положения [2; 12; 20; 26; 30; 67]. В данных публикациях предполагается размещение определенного количества датчиков в заранее специфицированных местах на платформе.

Более перспективным представляется создание метода предварительной идентификации геометрических параметров стенда, при котором расположение измерительных средств на платформе может быть произвольным (но при этом невырожденным). Однако, такой метод решения задачи идентификации позволит избежать рассогласования в кинематике движения стенда, но не в его *динамике*.

Перейдя к обсуждению следующей подзадачи, стоит отметить одну из основных проблем при создании стендов-тренажеров — необходимость имитировать динамику линейных и угловых ускорений. Это существенно отличается от классических задач управления роботизированными стендами, при которых требуется следовать заданной траектории и поддерживать заданную скорость. Постановка задачи имитации ускорений приводит к высоким требованиям к робастности предлагаемого управления: возмущающие силы не должны приводить к существенным отклонениям от желаемого динамического профиля.

Существуют различные алгоритмы нелинейного управления, обеспечивающие высокую точность реализации движения при наличии возмущений и неопределенностей в знании модели системы [8; 14; 44; 51]: одним из наиболее эффективных методов является скользящее управление [40; 53; 65]. Однако, такой вид управления не лишен определенных недостатков, таких как высокое потребление энергии и явление чаттеринга, способное вызвать угрозу разрушения объекта управления. Возникающий в системах управления с использованием скользящего режима первого порядка чаттеринг делает про-

блематичным применением данного алгоритма для имитации ускорений в связи с появлением больших импульсных помех. Чтобы избежать этих недостатков, можно использовать специальное расширение скользящего алгоритма: интегральный скользящий режим (ИСР). В отличие от скользящего режима первого порядка, при котором в системе возникает разрыв в правой части уравнений движения [113; 114], в ИСР разрыв возникает в производных управления, а само управление остается непрерывным. Также, теоретически, ИСР может сразу приводить систему на желаемую скользящую поверхность без переходного режима [27; 39; 66]. Подобно режиму первого порядка, интегральный скользящий режим робастен, но чувствителен к несогласованным неопределенностям [16].

В работе предлагается модификация ИСР на базе усредненного субградиента [32], учитывающая порог чувствительности вестибулярного аппарата при построении согласованной визуально-динамической имитации. Использование линейного критерия качества позволило повысить точность управления вблизи нуля ошибки имитации профиля ускорений по сравнению с квадратичным критерием. Проверка робастности управления осуществлялась путем поиска наихудших возмущений в рамках методики максиминного тестирования качества управления [68; 69; 71].

В качестве ключевого результата представлен метод синхронизации визуальной и динамической составляющих имитации и решения связанных с ними подзадач. Применимость этого метода продемонстрирована на примере имитации управляемого полёта летательного аппарата с использованием двух типов систем: трёхстепенной подвижной платформы с неподвижным экраном и модели динамического стенда на базе робота-манипулятора и шлема виртуальной реальности. Важно отметить, что общие принципы согласования воздействий не зависят от типа стенда и типа имитируемого транспортного средства.

**Целью** данной работы является разработка методики согласования визуальных и динамических воздействий на стендах-тренажерах.

Для достижения поставленных целей необходимо было решить следующие **задачи**:

1. Разработать методы предварительной идентификации геометрических параметров подвижных стендов;

2. Решить задачу отслеживания угловых движений подвижного стенда и головы человека;
3. Разработать решение задачи оптимальной остановки на границе множества допустимых положений платформы для построения областей активной фазы динамической имитации;
4. Обеспечить высокую частоту дискретизации визуализации для улучшения качества визуальной имитации и снижения вероятности возникновения дискомфорта у пользователя;
5. Создать алгоритм управления подвижным стендом по ускорению на базе интегрального скользящего режима;
6. Разработать метод прогнозирования углового движения подвижного стенда с учетом его динамики и времени задержки системы отслеживания движений;
7. Провести тестирование указанных алгоритмов.

**Научная новизна:**

1. Предложен метод синхронизации и координации визуальной и динамической имитации управляемых движений летательного аппарата;
2. Разработан алгоритм идентификации геометрических параметров подвижного стенда опорного типа, функционирующий при произвольном расположении маркеров оптической системы отслеживания движений в случае выполнения достаточно общих условий невырожденности;
3. Получено решение задачи оптимальной по быстродействию остановки на границе множества допустимых положений платформы, позволяющее построить область активной фазы динамической имитации;
4. Разработан алгоритм управления подвижным стендом по ускорению, основанный на использовании интегрального скользящего режима. В алгоритме учтены пороги чувствительности вестибулярной системы и ограничения на движение стенда. Сформулирована и доказана теорема о сходимости ошибки имитации к известной окрестности нуля при работе данного алгоритма;

**Теоретическая и практическая значимость.** Согласованная работа систем визуальной и динамической имитаций оказывает одно из ключевых воздействий на реалистичность тренажерной системы в целом. Рассогласование имитационных воздействий может вызывать болезнь движения

у пилота. Одна из причин возникновения болезни движения — сенсорный конфликт, т.е. получение непривычного набора информации от разных сенсорных систем. Предложенный алгоритм согласования воздействий позволит уменьшить данный эффект.

Метод построения рабочей области стенда через решение задачи оптимальной остановки позволит эффективно учитывать имитационные возможности стенда на этапе разработки алгоритмов динамической имитации. Теоретический результат, позволяющий расширить понятие "поверхности скольжения" до "слоя скольжения" имеет также прикладное значение: дает возможность автоматически учитывать порог чувствительности вестибулярного аппарата человека, тем самым расширяя динамические возможности имитационного стенда.

Результаты, полученные автором работы, использованы в работе НЦМУ «Сверхзвук» в 2020-2023 годах, а также применимы в задачах Центра подготовки космонавтов имени Ю. А. Гагарина и Центрального аэрогидродинамического института имени профессора Н. Е. Жуковского.

**Методология и методы исследования.** Задача формирования визуальной и динамической составляющих имитации решена с использованием следующих методов:

1. Задача об остановке кабины стенда на границе множества допустимых положений поставлена как задача быстрогодействия и решена с использованием принципа максимума Понтрягина;
2. Идентификация геометрических параметров подвижного стенда проводится с помощью численных методов оптимизации, таких как метод Ньютона с модификацией Исаева-Сонина и симплекс-метод Нелдера-Мида;
3. Управление стендом по ускорению строится с использованием интегрального скользящего режима;
4. Для исследования устойчивости системы с предложенным управлением применяется метод Ляпунова;
5. Математическая модель подвижного стенда строится с использованием метода Лагранжа второго рода.

**Основные положения, выносимые на защиту:**

1. Реализация синхронизации и координации визуальной и динамической имитации управляемых движений на стендах без жесткого за-

- крепления экрана на подвижной платформе осуществима путем решения следующих задач: а) предварительной идентификации геометрических параметров подвижного стенда; б) отслеживания угловых движений как подвижного стенда, так и головы человека; в) построения алгоритма управления стендом с учетом порогов чувствительности вестибулярного аппарата; г) прогноза углового движения подвижного стенда с учетом его динамики и времени задержки системы отслеживания движений; д) визуализации виртуального окружения с частотой кадров выше физиологически обусловленной пороговой величины.
2. Разработанный метод идентификации геометрических параметров подвижного стенда опорного типа не требует априорной информации о расположении маркеров оптической системы отслеживания движений на подвижных частях стенда при выполнении достаточно общих условий невырожденности;
  3. Решение задачи оптимальной остановки на границе множества допустимых положений для рассмотренных типов стендов выделяет в их фазовом пространстве максимально возможную область активной фазы динамической имитации.
  4. Разработанный алгоритм управления подвижным стендом на базе интегрального скользящего режима, учитывающий порог чувствительности вестибулярного аппарата, устойчив к возмущениям и позволяет привести ошибку имитации в окрестность нуля за конечное время.

**Достоверность и обоснованность результатов** обеспечивается проведенным численным моделированием каждой из решенных в диссертации задач, а также практическим применением ряда результатов в рамках работы панорамной системы виртуальной реальности МГУ имени М.В. Ломоносова. Результаты находятся в соответствии с результатами, полученными другими авторами.

**Апробация работы.** Основные результаты работы были представлены на научно-исследовательском семинаре имени А. Ю. Ишлинского кафедры прикладной механики и управления механико-математического факультета МГУ, семинаре семинара имени В.В. Румянцева по аналитической механике и теории устойчивости под руководством проф. А. А. Зобовой и проф.

Е.И. Кугушева, семинаре отдела номер 7 Института прикладной математики им. М. В. Келдыша РАН, а также докладывались на следующих конференциях:

- IX Международная научно-практическая конференция «Запись и воспроизведение объёмных изображений в кинематографе и других областях», Россия, Москва, 17-18 апреля 2017 г.
- XIII мультikonференция по проблемам управления, Конференция «Управление в аэрокосмических системах» имени академика Е. А. Микрина (УАКС — 2020) — ЦНИИ «Электроприбор», Россия, Москва, 6-8 октября 2020 г.
- Исторические исследования в контексте науки о данных: информационные ресурсы, аналитические методы и цифровые технологии — Москва, 04-06 декабря 2020 г.
- 22nd IAA Humans in Space Symposium — Дубай, О.А.Э., 11-14 ноября 2019 г.
- XIII Всероссийская (с международным участием) конференция Биомеханика - 2018 — Краснодарский край, с. Дивноморское, Россия, 28 мая – 1 июня 2018 г.
- Display Week 2021 — онлайн, 17-21 мая 2021 г.
- 1-я Всероссийская Конференция с международным участием "Биомехатронные системы: задачи, теория, технологии", Махачкала, Россия, 18-22 сентября 2021 г.

Отдельные элементы данной работы легли в основу следующего зарегистрированного программного обеспечения:

- Программное обеспечение имитатора персонально управляемого колесного транспорта. Авторы: Чертополохов В. А., Бурлаков Д. С., Лебедев А. В., #2016619103, 2016 г.
- Программное обеспечение системы видеофиксации движений. Авторы: Бурлаков Д. С., Чертополохов В. А., #2016660867, 2016 г.
- Программное обеспечение для компактной системы виртуальной реальности. Авторы: Чертополохов В. А., Бурлаков Д. С., Латов В. В., #2018666818, 2018 г.

**Публикации.** Основные результаты данной диссертации опубликованы в 13 публикациях, из них 4 — научные статьи, опубликованные в рецен-

зируемых журналах, индексируемых в международных базах Scopus, WoS, RSCI.

**Личный вклад.** Все выносимые на защиту результаты и положения были получены лично автором или при его определяющем участии. Постановка задачи динамической имитации и оценка результатов принадлежит первому руководителю (В. В. Александрову), методика решения задач максиминного тестирования принадлежит второму руководителю (С. С. Лемаку). Автором сформулирована постановка задачи визуализации и синхронизации визуальной и динамической имитации, предложен метод решения этой задачи путем декомпозиции на несколько подзадач. Автором разработан алгоритм идентификации геометрических параметров подвижного стенда опорного типа, также автор принимал участие в создании программного обеспечения для Панорамной системы виртуальной реальности МГУ и анализировал экспериментальные результаты. Совместно с научными руководителями предложена методология построения множества активной фазы динамической имитации, получены аналитические и численные результаты в этой области. Автору принадлежит идея модификации скользящей переменной, при которой учитывается порог чувствительности вестибулярного аппарата. Автором доказана устойчивость замкнутой системы при наличии модифицированного таким образом управления с использованием скользящего режима и проведено численное моделирование, подтверждающее аналитические результаты.

**Объем и структура работы.** Диссертация состоит из введения, четырёх глав, списка обозначений и заключения. Полный объём диссертации составляет 160 страниц, включая 66 рисунков и 4 таблицы. Список литературы содержит 117 наименований.

Глава 1 посвящена задаче динамической имитации управляемых движений. В данной главе приведена постановка задачи, виды функционалов и предложено решение задачи на примере трёхстепенной платформы опорного типа.

В главе 2 описана задача визуальной имитации в системах виртуальной реальности и представлен общий способ решения данной задачи.

Глава 3 содержит обобщение информации из предыдущих глав. Сформулирован метод синхронизации визуальной и динамической имитации. В

этой главе также приведено решение ряда подзадач, включая построение управления стендом с использованием интегрального скользящего режима.

Глава 4 описывает модель стенда-тренажера нового типа, основанного на комплексе, состоящем из промышленного робота-манипулятора и шлема виртуальной реальности. Представлено решение задачи определения области активной фазы динамической имитации и смоделирована работа алгоритма согласования визуальной и динамической имитации. Приведена методика максиминного тестирования и её использование для проверки робастности предложенного управления.



## Глава 1. Задача динамической имитации управляемого движения

<sup>1</sup>В данной главе представлена формализация задачи динамической имитации управляемых движений летательного аппарата, проанализированы и описаны некоторые алгоритмы, используемые для создания имитационных воздействий на подвижных стендах.

### 1.1 Устройство вестибулярного аппарата

Для определения своего положения в пространстве различные виды животных, такие как рыбы, птицы, рептилии и млекопитающие, используют вестибулярный аппарат. У млекопитающих этот орган находится во внутреннем ухе и состоит из двух основных частей: отолитовых органов и полукружных каналов (рис. 1.1).

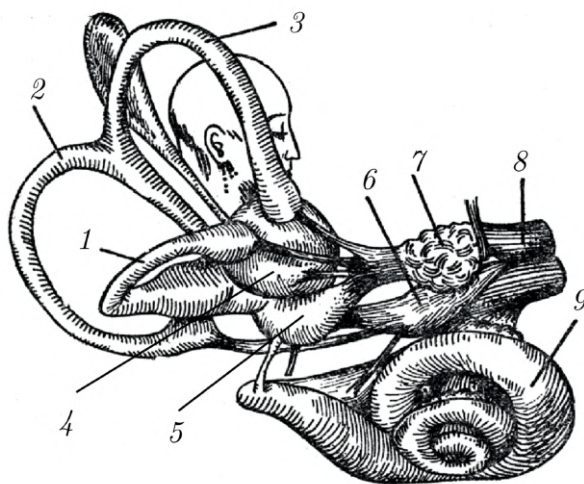


Рисунок 1.1 — 1, 2, 3 — горизонтальный, задний и передний полукружный канал соответственно; 4, 5 — отолитов аппарат; 6, 7 — нервные ганглии; 8 — вестибулярный нерв; 9 — улитка

Отолитовые органы чувствительны к линейным ускорениям и рывкам, в то время как полукружные каналы реагируют на угловую скорость и угловые ускорения. Шесть полукружных каналов расположены в плоскостях,

---

<sup>1</sup>При подготовке данного раздела диссертации использованы следующие публикации, выполненные автором лично или в соавторстве, в которых, согласно Положению о присуждении ученых степеней в МГУ, отражены основные результаты, положения и выводы исследования: [5;83]

образующих углы от  $84$  до  $98^\circ$  между собой. Можно предположить, что каналы попарно перпендикулярны. Каждый из каналов чувствителен к угловому ускорению в определенном направлении [99]. Одна из математических моделей процессов, происходящих в полукружных каналах, была разработана в лаборатории математического обеспечения имитационных динамических систем МГУ совместно с лабораторией нейрофизиологии Автономного университета штата Пуэбла (Мексика) [59; 77; 109]. В дальнейшем текст вестибулярный аппарат будет рассматриваться как механорецептор, для которого проводится имитация динамических воздействий (динамическая имитация).

## 1.2 Физическая постановка задачи динамической имитации вектора гравито-инерциальных сил, действующих на пилота

Предположим, что известна модель движения тяжелого летательного аппарата (ЛА) и его геометрические и динамические параметры. Также предполагается, что известны все силы, действующие на ЛА, и вектор состояния ЛА как функция времени. На основе этих данных можно определить силы, которые действуют на виртуального пилота в модели самолета.

### 1.2.1 Силы, действующие на виртуального пилота в модели самолета

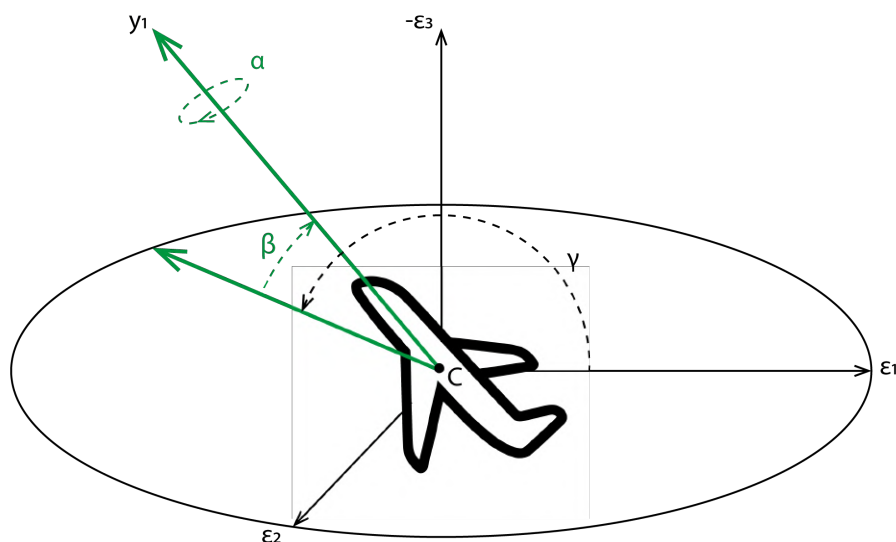


Рисунок 1.2 — Углы Крылова

Выберем систему координат, связанную с летательным аппаратом: начало координат положим в точке  $C$  — центре масс самолета,  $Sy_1$  направим вдоль главной (продольной) оси самолета,  $Sy_2$  — вдоль правого крыла,

$Cy_3$  определим таким образом, чтобы система координат образовала правую тройку.

Самолет перемещается относительно системы координат  $O\xi_1^*\xi_2^*\xi_3^*$ , которую в дальнейшем будем считать связанной с Землей (например, с определенной точкой аэродрома). Движение происходит в поле силы тяжести. Предполагается, что ось  $O\xi_3^*$  направлена вертикально вниз (параллельно направлению силы тяжести). Введем систему отсчета  $C\xi_1\xi_2\xi_3$ , связанную с центром масс самолета  $C$ , которая не совершает вращательных движений, и оси этой системы направлены аналогично системе отсчета  $O\xi_1^*\xi_2^*\xi_3^*$ .

В дальнейшем будем использовать систему углов Крылова [94], которая задает поворот от системы координат  $C\xi_1\xi_2\xi_3$  к жестко связанной с самолетом системе координат  $Cy_1y_2y_3$ .

Зададим последовательность поворотов через эти углы — рысканье  $\gamma$ , тангаж  $\beta$  и крен  $\alpha$  (рис. 1.2).

$$C\xi_1\xi_2\xi_3 \xrightarrow[\xi_3\xi_3']{\gamma} C\xi_1'\xi_2'\xi_3' \xrightarrow[\xi_2'y_2']{\beta} Cy_1'y_2'y_3' \xrightarrow[y_1'y_1]{\alpha} Cy_1y_2y_3. \quad (1.1)$$

Матрицы перехода, задающие вращение относительно трех осей координат, имеют следующий вид:

$$R_{\xi_3}(\gamma) = \begin{bmatrix} \cos \gamma & \sin \gamma & 0 \\ -\sin \gamma & \cos \gamma & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad (1.2)$$

$$R_{\xi_2'}(\beta) = \begin{bmatrix} \cos \beta & 0 & -\sin \beta \\ 0 & 1 & 0 \\ \sin \beta & 0 & \cos \beta \end{bmatrix}, \quad (1.3)$$

$$R_{y_1'}(\alpha) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \alpha & \sin \alpha \\ 0 & -\sin \alpha & \cos \alpha \end{bmatrix}. \quad (1.4)$$

Угловая скорость самолета  $\omega^p = (\omega_1^p, \omega_2^p, \omega_3^p)^T$  в связанной с самолетом системе координат получается из кинематических уравнений Пуассона [89]:

$$\varpi^p = R_{y_1'}(\gamma)^T R_{\xi_2'}(\beta)^T R_{\xi_3}(\alpha)^T \frac{d}{dt} (R_{\xi_3}(\alpha) R_{\xi_2'}(\beta) R_{y_1'}(\gamma)).$$

где  $\varpi^p$  — кососимметрическая матрица угловой скорости, определяемая как:

$$\varpi^p = \begin{bmatrix} 0 & -\omega_3^p & \omega_2^p \\ \omega_3^p & 0 & -\omega_1^p \\ -\omega_2^p & \omega_1^p & 0 \end{bmatrix}. \quad (1.5)$$

Здесь и далее верхний индекс  $p$  указывает на принадлежность к модели летательного аппарата.

Допустим ряд упрощений: масса самолета постоянна (топливо сгорает равномерно и медленно, рассматривается достаточно короткий временной интервал), положение центра масс не изменяется, сила тяжести постоянна, а Земля "локально" считается плоской и неподвижной.

Пусть  $M$  — масса самолета,  $R^e$  — вектор равнодействующей сил тяги двигателей,  $R^{aero}$  — вектор равнодействующей аэродинамических сил,  $g$  — вектор ускорения свободного падения на высоте полета ЛА.

В данной работе используется готовая компьютерная реализация математической модели летательного аппарата, созданная на языке C#, которая представляет собой "черный ящик" с известным входом в виде управления самолетом и выходом, который будет описан далее.

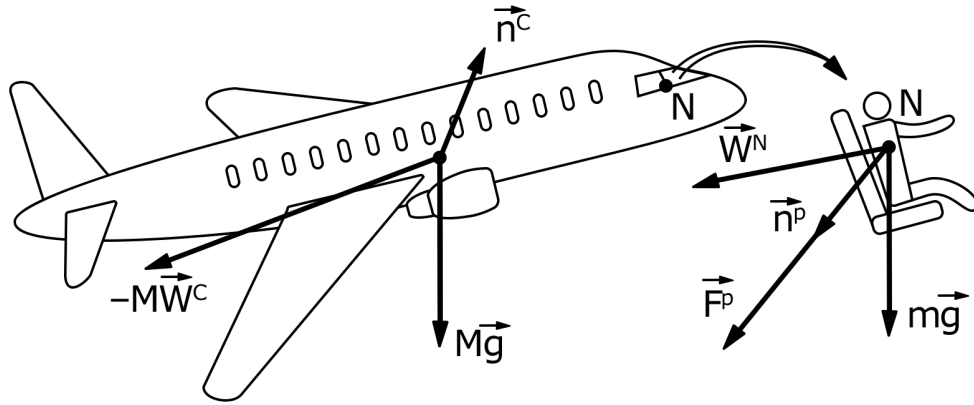


Рисунок 1.3 — Силы, приложенные к центру масс самолета и к расчетной точке на корпусе пилота

Введем вектор перегрузки для самолета как отношение суммы всех действующих на самолет сил (кроме силы тяжести), к абсолютной величине силы тяжести, действующей на самолет [98]:

$$n^C \approx \frac{1}{g_0 M} (R^e + R^{aero}) = \frac{1}{g_0} (W^C - g).$$

Здесь  $g_0$  — абсолютная величина ускорения свободного падения на характерной высоте полета ЛА,  $W^C$  — вектор суммарного ускорения центра масс летательного аппарата в системе отсчета, связанной с Землей, задаваемый как:

$$W^C = \frac{R^e + R^{aero}}{M} + g.$$

Рассмотрим движение чувствительной массы механорецептора вестибулярного аппарата пилота. Предположим, что масса находится в упруго-вязком подвесе в жестком корпусе с коэффициентами упругости  $\hbar$  и вязкости  $\wp$  соответственно. Во время полета тело пилота изменяет свое положение относительно самолета, однако считаем эти изменения незначительными, поэтому проводим вычисления в расчетной точке ( $N$ ), которая совпадает с положением чувствительной массы в начальный момент времени (рис. 1.3). Введем систему координат  $Nx_1x_2x_3$  с центром в точке  $N$  и осями, сонаправленными с  $Cy_1y_2y_3$ . Тогда уравнения динамики чувствительной массы можно записать следующим образом:

$$m\ddot{x}^p + \wp\dot{x}^p + \hbar x^p = m(g + W^N),$$

$$\begin{aligned} W^N &:= -W^C - W_{N,tr} - W_{N,rel} - W_{N,cor} = \\ &= -W^C - [\dot{\omega}^p, r(N)] - [\omega^p, [\omega^p, r(N)]] - 2[\omega^p, \dot{x}^p]. \end{aligned} \quad (1.6)$$

Здесь  $x^p$  — положение чувствительной массы в системе координат  $Nx_1x_2x_3$ ,  $m$  — ее масса,  $r(N)$  — радиус вектор расчетной точки  $N$  в системе координат, связанной с центром масс самолета,  $W_{N,tr}$  — переносное ускорение,  $W_{N,cor}$  — кориолисово,  $W_{N,rel}$  — относительное. В дальнейшем кориолисово ускорение считаем незначительным из-за малых перемещений и скоростей перемещения пилота в кабине и поэтому не учитываем в задаче. Векторную сумму сил инерции и гравитации называем *гравито-инерциальными* силами.

Обозначим  $W^\alpha := -W_{N,tr} - W_{N,rel}$  и вектор гравито-инерциальных сил  $F^p$ :

$$F^p = mg + mW^N = m(g + W^N) = mg_0 \frac{g + W^N}{g_0},$$

$$W^N = -W^C + W^\alpha.$$

Введем вектор перегрузки для чувствительной массы в расчетной точке  $N$  следующим образом:

$$n^p \approx \frac{F^p}{mg_0} = \frac{g + W^N}{g_0} = -n^C + \frac{1}{g_0}W^\alpha. \quad (1.7)$$

Из-за наличия собственной массы механорецептора, силы инерции воспринимаются направленными в противоположную сторону от действительного ускорения ЛА. В итоге получаем, что пилот во время полета чувствует (за счет механорецепторов) те же перегрузки, что действуют на самолет (с точностью до знака и с учетом инерциальных сил, возникающих в результате собственного вращения воздушного судна относительно центра масс). Динамическую имитацию движения летательного аппарата осуществляем, получая из математической модели летательного аппарата  $n^p$  — вектор перегрузки для чувствительной массы механорецептора пилота в расчетной точке.

В случае, когда имеется полная информация о перегрузке для центра масс, возможно самостоятельно рассчитать вектор перегрузки для чувствительной массы в расчетной точке  $N$ , используя соотношения (1.6) и (1.7) (кориолисовую силу, как и ранее, не учитываем). Для удобства перепишем соотношения, применив матричное представление векторного произведения [72]:

$$n^p(N) = -n^C + \frac{1}{g_0}(\dot{\varpi}^{pT} - \varpi^{p2})r(N).$$

Данное представление будет использоваться в дальнейшем при описании задачи имитации на стендах-тренажерах динамических воздействий, возникающих при полете летательного аппарата. Аналогичное представление перегрузки можно использовать для имитации движения других транспортных средств. В частности, для проверки применимости алгоритмов динамической имитации была разработана и протестирована компьютерная модель колесного транспорта собственной реализации, описание которой выходит за рамки данной работы.

### 1.2.2 Задача динамической имитации вектора гравито-инерциальных сил, действующих на пилота

Для тренировки сенсомоторных навыков необходимы системы создания динамических стимулов (англ. Motion Cueing), максимально точно имитирующих для механорецепторов возникающие в полете ускорения в качестве реакции на корректирующее или возмущающее действие пилота-оператора на органы управления. Для этого применяются специализированные подвижные платформы, на которых обычно размещаются экраны систем визуализации (рисунок 1.4).



Рисунок 1.4 — Пример системы подвижности с двумя степенями свободы в составе авиасимулятора

На тренажере невозможно точно воспроизвести все силы, действующие на пилота на протяжении всего полета, что связано с механическими ограничениями стенда. Поэтому для получения реалистичной имитации требуется создавать алгоритмы управления стендом, позволяющие эмулировать невозпроизводимые на стенде движения за счет воспроизводимых.

На сегодняшний день стенды для имитации воздействий, оказываемых на пилота во время его движения на летательном аппарате или каком-либо другом транспортном средстве, реализованы по большей части на основе платформ опорного типа и центрифуг. Алгоритмы имитации на этих стендах представлены в работах [58; 74; 112].

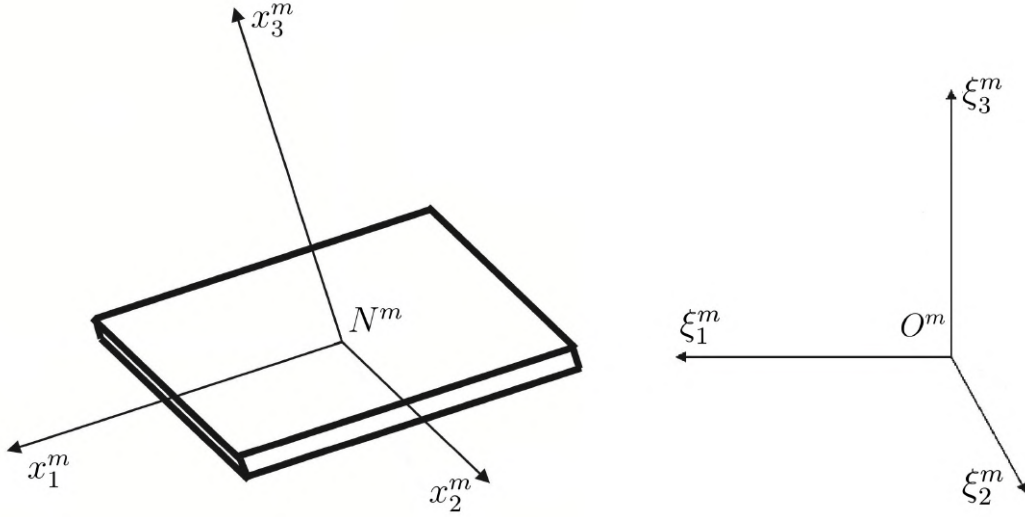


Рисунок 1.5 — Подвижная система координат  $N^m x_1^m x_2^m x_3^m$  и система координат  $O^m \xi_1^m \xi_2^m \xi_3^m$ , связанная с комнатой

Рассмотрим стенд опорного типа. Будем считать, что механическая часть динамического стенда состоит из подвижной и опорной частей, стержней фиксированной и переменной длины, а также шарниров. Предположим, что на платформе расположено кресло, полностью идентичное тому, которое должно находиться в имитируемом объекте (летательном аппарате). Рассмотрим модельный стенд, с неподвижным основанием которого свяжем систему координат  $O^m \xi_1^m \xi_2^m \xi_3^m$ . В то же время, жестко свяжем с подвижной частью платформы в некоторой расчетной точке систему координат  $N^m x_1^m x_2^m x_3^m$  (рис. 1.5).

Введем вектор обобщенных координат  $q(t) \in \mathbb{R}^s$  для стенда-платформы с  $s$  степенями свободы. Пусть динамика стенда описывается системой вида:

$$\ddot{q} = \Phi_a(\dot{q}, q) + \Phi_b(q)u + \Xi(\dot{q}, q, t). \quad (1.8)$$

Управления  $u(\cdot) \in U = \{u(\cdot) \in KC[t_0, t_k] | u(t) \in U\}$ , начальные условия  $q(0) \in Q^0$ , где  $Q^0, U$  — замкнутые и ограниченные множества, матрицы  $\Phi_a$  и  $\Phi_b$  известны,  $\Xi$  — ошибки моделирования и возмущения.

Аналогично ситуации реального полета, уравнения динамики чувствительной массы в точке  $N^m$  можно записать как:

$$m\ddot{x} + \wp\dot{x} + \hbar x = m(g + W^{mN}).$$

Здесь, как и ранее,  $x$  обозначает положение чувствительной массы в системе координат  $N^m x_1^m x_2^m x_3^m$ ,  $m$  — ее масса,  $W^{mN} = W^{mN}(\dot{q}, q, t)$  — ускорения, вызванные силами инерции при управляемом движении платформы.



Таким образом, вектор перегрузки для чувствительной массы в расчетной точке  $N^m$  подвижной платформы можно представить, как и в (1.7):

$$n := \frac{g + W^{mN}}{g_0}. \quad (1.9)$$

В идеальном случае чувствительная масса должна осуществлять движения, полностью идентичные тем, которые она совершала бы в имитируемом полете. Для этого необходимо подобрать такое управление  $u(\cdot) \in U$ , что

$$|n(t) - n^p(t)| \equiv 0, \forall t \geq 0.$$

Будем считать, что имитация производится для некоторого конкретного маневра самолета (например, для посадки, взлета, координированного разворота). На практике, из-за механических ограничений стенда и системы управления, достичь равенства для большинства маневров не представляется возможным.

По этой причине задачу динамической имитации в общем случае можно формулировать в виде задачи оптимизации:

$$J_n = \|n - n^p\| \rightarrow \inf_{u \in U}. \quad (1.10)$$

Предположим, что решение задачи (1.10) существует в рассматриваемом классе управлений. Тогда можно заменить операцию  $\inf$  на  $\min$ :

$$J_n = \|n - n^p\| \rightarrow \min_{u \in U}. \quad (1.11)$$

Здесь в качестве нормы можно использовать  $L^1$  - норму,  $L^2$  - норму,  $L^\infty$  - норму и другие. Например, если известен отрезок  $[t_0, t_k]$ , когда происходит имитация, можно рассмотреть следующий вариант функционала:

$$J_n = \int_{t_0}^{t_k} (n(t) - n^p(t))^T Q (n(t) - n^p(t)) dt, \quad (1.12)$$

где матрица  $Q$  — диагональная матрица  $3 \times 3$  и  $Q > 0$ .

На практике системы имитации работают с некоторой частотой дискретизации, поэтому возможно ставить следующую задачу оптимизации на шаге  $[t_{s-1}, t_s]$ :

$$J_{n_{step}} = (n(t_s) - n^p(t_s))^T Q (n(t_s) - n^p(t_s)) \rightarrow \min_{u \in U}.$$

В большинстве случаев управление можно считать постоянным на каждом шаге. При  $(t_s - t_{s-1}) \rightarrow 0$  задача формулируется следующим образом:

$$\left. \frac{d}{dt} J_n(u(t)) \right|_{t=t_s} \rightarrow \min_{u \in U}. \quad (1.13)$$

Таким образом, ставится задача построения локально оптимального управления для  $\forall \tau \in [t_0, t_k]$ .

Описанный выше подход позволяет осуществлять имитацию для одной чувствительной массы. В предположении, что угловая скорость платформы в связанной системе координат задается как  $\omega^T = (\omega_1, \omega_2, \omega_3)$ , для любой другой расчетной точки  $M^m$  с радиус-вектором  $r(M^m)$  из точки  $N^m$  можно пересчитать перегрузку:

$$n(M^m) = n + \frac{1}{g_0}(\dot{\varpi}^T - \varpi^2)r(M^m). \quad (1.14)$$

Здесь матрица  $\varpi$  составляется из угловой скорости  $\omega$  аналогично (1.5). В случае нескольких чувствительных масс из соотношения (1.14) следует, что требуется учитывать не только перегрузку, но также угловую скорость и угловое ускорение [72]. Тогда дополнительно можно рассмотреть следующие задачи:

$$J_\omega = \|\omega - \omega^p\| \rightarrow \min_{u \in U}, \quad (1.15)$$

$$J_{\dot{\omega}} = \|\dot{\omega} - \dot{\omega}^p\| \rightarrow \min_{u \in U}. \quad (1.16)$$

Норма вводится аналогично задаче (1.11).

В работе [52] приведено предположение, что при быстрых движениях отолитовые органы воспринимают рывок (производную ускорения). Тогда задачу (1.11) можно представить в следующем виде:

$$J_{\dot{n}} = \|\dot{n} - \dot{n}^p\| \rightarrow \min_{u \in U}. \quad (1.17)$$

Зачастую возможности стенда не позволяют решать задачи (1.11) и (1.17). В этом случае ставится задача не симитировать точное движение чувствительной массы, а обеспечить схожее с реальным полетом направление ее отклонения. В этом случае возникает задача имитации ориентации вектора перегрузки, когда перегрузки  $n$  и  $n^p$  должны быть ориентированы

относительно пилота похожим образом:

$$J_{\frac{n}{|n|}} = \left\| \frac{n}{|n|} - \frac{n^p}{|n^p|} \right\| \rightarrow \min_{u \in U}. \quad (1.18)$$

Во всех случаях минимизация производится за счет выбора соответствующего управления для подвижного стенда, управление зависит от желаемых угловых и линейных ускорений, рывка и угловых скоростей, конфигурации и ограничений стенда. В реальной практике создания тренажерных систем применяется следующий подход: данные о перегрузке и угловой скорости фильтруются с помощью фильтра высоких и низких частот [63], "медленная" составляющая поступает на вход алгоритма (1.18), "быстрая" составляющая поступает на вход алгоритмов (1.11) и (1.15) (рисунок 1.6).

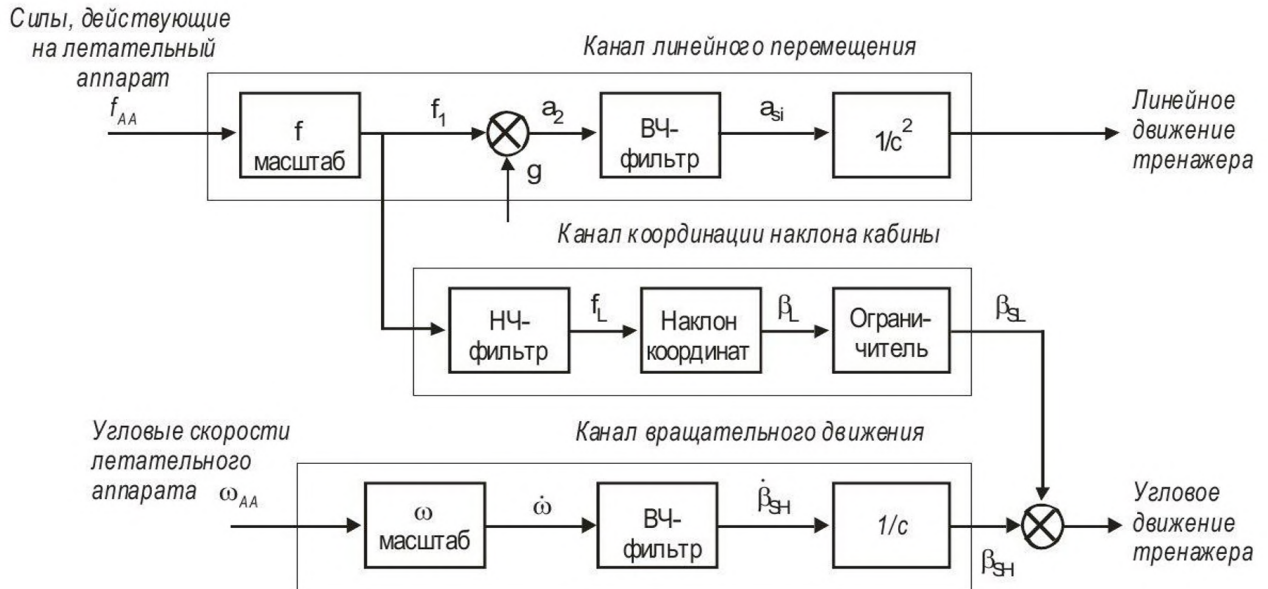


Рисунок 1.6 — Типовой алгоритм динамической имитации [107]

Итоговый функционал, который можно применять в задаче динамической имитации, разумно представить как сумму нескольких из приведенных выше функционалов с некоторыми весами, подбираемыми под особенности конкретного динамического стенда и имитируемых маневров, например:

$$J = \lambda_1 J_{\frac{n}{|n|}} + \lambda_2 J_n + \lambda_3 J_\omega. \quad (1.19)$$

### 1.3 Алгоритм динамической имитации ориентации вектора перегрузки

Будем считать, что в начальный момент времени самолет и стенд расположены неподвижно, горизонтально (рисунок 1.7).

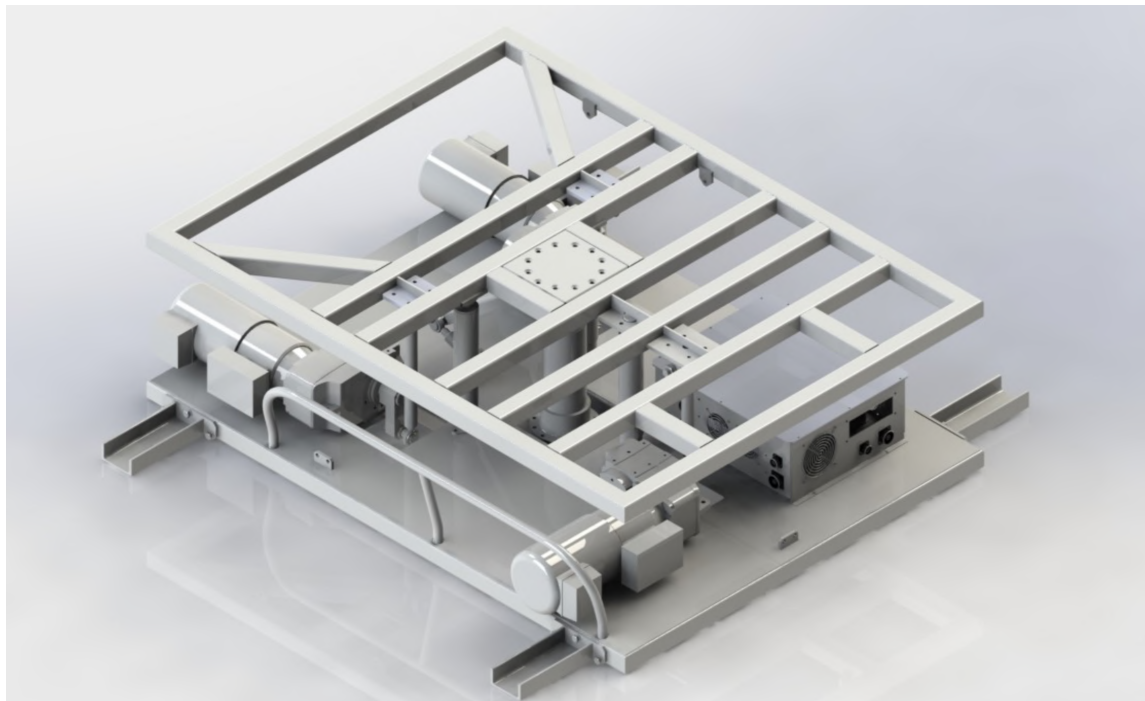


Рисунок 1.7 — Пример подвижной платформы

На пилота, находящегося на неподвижной платформе действует в качестве гравито-инерциальных сил только одна сила тяжести. Для отслеживания изменения ориентации вектора гравито-инерциальных сил относительно платформы, введем матрицу поворота  $R$  аналогично (1.1) в последовательности курс(1.2)-тангаж(1.3)-крен(1.4).

После поворота платформы на три угла вектор силы тяжести будет иметь следующие компоненты в подвижной СК:

$$\begin{bmatrix} 0 \\ 0 \\ g_0 m \end{bmatrix} \xrightarrow{R} g_0 m \begin{bmatrix} -\sin \beta \\ \sin \alpha \cos \beta \\ \cos \alpha \cos \beta \end{bmatrix}$$

Заметим, что ориентация вектора не зависит от угла рысканья. Теперь рассмотрим ЛА. Предположим, что перегрузка для пилота в кабине вирту-

ального самолета имеет следующие компоненты:

$$n^p = \begin{bmatrix} n_1^p \\ n_2^p \\ n_3^p \end{bmatrix}.$$

Сформулируем задачу имитации ориентации вектора перегрузки в случае отсутствия геометрических ограничений:

$$\begin{bmatrix} -\sin \beta - \frac{n_1^p}{|n^p|} \\ \sin \alpha \cos \beta - \frac{n_2^p}{|n^p|} \\ \cos \alpha \cos \beta - \frac{n_3^p}{|n^p|} \end{bmatrix} = 0 \quad (1.20)$$

Одно из решений системы (1.20):

$$\beta = -\arctg\left(\frac{n_1^p}{\sqrt{(n_2^p)^2 + (n_3^p)^2}}\right), \quad \alpha = \arctg\left(\frac{n_2^p}{n_3^p}\right). \quad (1.21)$$

В случае, когда не допускается переворот платформы, по значению желаемого вектора перегрузки можно однозначно определить крен и тангаж имитационной платформы.

Точное решение задачи имитации ориентации возможно для свободно вращающейся платформы (например, карданов подвес центрифуги), но для стендов опорного типа существуют определенные ограничения на углы поворота.

Приведем следующие определения в общем виде:

**Определение 1.** Будем называть  $\Omega^w$  **множеством допустимых положений** стенда, если оно содержит все возможные кинематические конфигурации  $\xi^m = (\xi_1^m, \xi_2^m, \xi_3^m, \alpha, \beta, \gamma)$ , где  $\xi_1^m, \xi_2^m, \xi_3^m$  — декартовы координаты некоторой точки кабины, а  $\alpha, \beta, \gamma$  — углы поворота кабины относительно системы отсчета, связанной с комнатой (рисунок 1.5).

**Определение 2.** **Рабочей областью** или **областью активной фазы имитации** стенда будем называть множество  $\Omega^a \subset \Omega^w$ , на границе которого прерывается процесс имитации и осуществляется торможение в пределах множества допустимых положений.

В случае, если углы из (1.21), выходят за пределы рабочей области стенда  $\Omega^a(t)$ , будем искать решение в следующем виде:

$$\xi = \Pi\{\hat{\xi}\}, \quad (1.22)$$

где  $\hat{\xi}$  — вычисленные в соответствии с (1.21) углы и некоторые значения остальных координат, допускающих данные углы,  $\Pi\{\cdot\}$  — оператор проектирования следующего вида:

$$\Pi\{\hat{\xi}\} = \begin{cases} \hat{\xi}, & \text{если } \hat{\xi} \in \Omega^a(t), \\ \xi^* \in \Omega^a(t) : \xi^* = \min_{\xi \in \Omega^a(t)} \|\hat{\xi} - \xi\|, & \text{если } \hat{\xi} \notin \Omega^a(t). \end{cases} \quad (1.23)$$

#### 1.4 Кинематика трехстепенной платформы опорного типа

Рассмотрим кинематические соотношения динамического стенда на базе трехстепенной платформы опорного типа.

В качестве иллюстрации рассмотрим модель динамического стенда панорамной системы виртуальной реальности МГУ имени М.В. Ломоносова. Он состоит из платформы, на которой установлены три кривошипно-шатунных механизма, преобразующих вращательное движение валов приводных двигателей в поступательное движение соответствующих кривошипов. Платформа перемещается с помощью этих механизмов. Обозначим за  $r$  длину кривошипа, который крепится к ротору двигателя, и за  $R$  длину шатуна, который соединяется с кривошипом через сферический шарнир. К платформе шатун также крепится с помощью сферического шарнира.

В центре неподвижной части платформы вертикально расположен телескопический стержень, который с помощью карданного вала соединен с подвижной частью платформы. Стержень не имеет двигателей, и его длина пассивно изменяется при движении платформы. Продольные шлицы, расположенные вдоль всей длины стержня, предотвращают его вращение вокруг своей оси.

Управление двигателями платформы осуществляется с компьютера путем передачи трех управляющих значений на контроллер, которые можно записать в виде вектора-столбца  $u = (A, B, C)^T$ . Контроллер линейно преобразует эти значения в углы поворота шагового двигателя и управляет перемещением платформы в новую позицию.

Аналогично предыдущему разделу, рассмотрим углы Крылова  $\gamma$ ,  $\beta$ ,  $\alpha$  в качестве рыскания, тангажа и крена. Поворот по углу рыскания  $\gamma$  равен нулю из-за конструкции платформы. Введем высоту  $h$  точки крепления карданного вала выдвижного стержня в системе координат  $O^m \xi_1^m \xi_2^m \xi_3^m$  (рис. 1.5).

Предположим, что углы поворота двигателей, равные углам поворота кривошипа и задаваемые вектором-столбцом  $u_{ang} = (a, b, c)^\top$ , линейно зависят от управляющих значений:

$$u_{ang} = (a, b, c)^\top = (k_A[A - A_0], k_B[B - B_0], k_C[C - C_0])^\top,$$

где  $k_A, k_B, k_C$  — масштабные коэффициенты,  $A_0, B_0, C_0$  — смещения нуля.

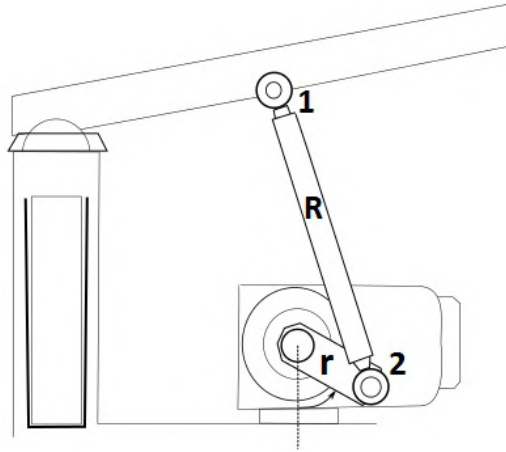


Рисунок 1.8 — Электродвигатель, соединенный с платформой

В дальнейшем множество допустимых положений для этой платформы будет строиться в пространстве углов  $\alpha, \beta$  и высоты  $h$  — доступных степеней свободы платформы.

Запишем систему уравнений, связывающую поворот и высоту платформы с управляющими значениями  $u$ . Рассмотрим один из блоков двигателя (рис. 1.8). В неподвижной системе  $O^m \xi_1^m \xi_2^m \xi_3^m$  координаты шарнира 2, соединяющего шатун и кривошип, имеют вид:

$$(D_{A1}, D_{A2}, D_{A3}) = (V_{A1} - r \sin a, V_{A2}, V_{A3} - r \cos a), \quad (1.24)$$

где  $(V_{A1}, V_{A2}, V_{A3})$  — координаты точки крепления кривошипа к двигателю в неподвижной системе координат,  $a$  — угол поворота кривошипа.

В то же время, координаты шарнира 1, расположенного в точке крепления шатуна к платформе, в неподвижной системе координат  $O^m \xi_1^m \xi_2^m \xi_3^m$  определяются положением платформы  $(\beta, \alpha, h)$ :

$$\begin{aligned} (E_{A1}, E_{A2}, E_{A3}) = & (\cos \alpha S_{A1} + \sin \alpha S_{A3}, \\ & \sin \beta \sin \alpha S_{A1} + \cos \beta S_{A2} - \sin \beta \cos \alpha S_{A3}, \\ & h - \cos \beta \sin \alpha S_{A1} + \sin \beta S_{A2} + \cos \beta \cos \alpha S_{A3}), \end{aligned} \quad (1.25)$$

где  $(S_{A1}, S_{A2}, S_{A3})$  — координаты точки крепления кривошипа к двигателю в подвижной системе координат  $N_{x_1^m x_2^m x_3^m}^m$ .

Из того, что длина шатуна равна  $R$ , можно получить уравнение, связывающее повороты кривошипа, шатуна и положение платформы. Аналогичная операция проводится с остальными блоками двигателей. Выпишем полученные кинематические соотношения:

$$\begin{cases} (D_{A1} - E_{A1})^2 + (D_{A2} - E_{A2})^2 + (D_{A3} - E_{A3})^2 - R^2 = 0, \\ (D_{B1} - E_{B1})^2 + (D_{B2} - E_{B2})^2 + (D_{B3} - E_{B3})^2 - R^2 = 0, \\ (D_{C1} - E_{C1})^2 + (D_{C2} - E_{C2})^2 + (D_{C3} - E_{C3})^2 - R^2 = 0, \end{cases} \quad (1.26)$$

где  $D_{ij}$  —  $j$ -я координата точки крепления шатуна к кривошипу  $i$ -го двигателя,  $E_{ij}$  —  $j$ -я координата точки крепления шатуна  $i$ -го двигателя к платформе в неподвижной системе координат  $O^m \xi_1^m \xi_2^m \xi_3^m$ ,  $i \in \{A, B, C\}$  (здесь и далее  $\{\}$  обозначает множество из указанных элементов),  $j = 1, 2, 3$ .

Таким образом, для данной платформы набор геометрических параметров представляет собой вектор

$$\mathbf{td} = (k_A, k_B, k_C, A_0, B_0, C_0, V_{ij}, S_{ij}, R, r). \quad (1.27)$$

Здесь  $V_{ij}$  —  $j$ -я координата точки крепления кривошипа к  $i$ -му двигателю в неподвижной системе координат  $O^m \xi_1^m \xi_2^m \xi_3^m$ , тогда как  $S_{ij}$  —  $j$ -я координата точки крепления кривошипа к  $i$ -му двигателю в подвижной системе координат  $N_{x_1^m x_2^m x_3^m}^m$ ,  $i \in \{A, B, C\}$ ,  $j = 1, 2, 3$ .

Подставляя (1.24), (1.25) и аналогичные вектора для других двигателей в систему (1.26) можно получить соотношения, связывающие величины  $(\beta, \alpha, h)$  и управляющие значения  $u = (A, B, C)^\top$ .

После разрешения системы (1.26) становится возможным однозначно определять положение платформы  $(\beta, \alpha, h)$  через поданные управления  $u = (A, B, C)^\top$ . Дополнительно появляется возможность построить множество допустимых положений платформы.

## 1.5 Алгоритм предварительной идентификации геометрических параметров подвижного стенда

Набор приведенных ранее параметров может быть известен неточно для реальной системы по ряду объективных причин. В результате это приводит к некорректному позиционированию платформы, и как следствие, к нарушению качества имитации.



Был разработан алгоритм идентификации геометрических параметров стенда опорного типа. Для этого используются полученные теоретически геометрические соотношения для платформы, которые позволяют преобразовать углы поворота роторов двигателей в ориентацию платформы в неподвижной системе координат. Уточнение геометрических параметров стенда будет проводиться на основе полученных экспериментальных данных.

Опишем метод идентификации геометрических параметров, для платформы опорного типа в общем виде. Для этого необходимо выписать преобразования между системами координат и составить вектор геометрических параметров.

Введем вектор геометрических параметров  $\mathbf{td} \in TD$ , который включает все длины, положения точек крепления элементов стенда и другие параметры. Для трехстепенной платформы опорного типа данный вектор приведен в (1.27).

### 1.5.1 Постановка задачи поиска геометрических параметров стенда-платформы

Пусть из внешних измерений оптической системы возможно построить матрицу преобразования для реальной кабины  $T^u \in \mathbb{R}^{4 \times 4}$  (от реальной неподвижной системы  $O\xi_1^r\xi_2^r\xi_3^r$ , к подвижной  $Cx_1^rx_2^rx_3^r$  — связанной с платформой) при некотором известном управлении  $u^*(\cdot)$ , приводящем платформу в стационарное положение. Подробное описание матричного представления преобразований пространства будет приведено в главе 2.

Введем функционал  $F(u^*, \mathbf{td}, T_m^u, T^u) = \|(T^u)^{-1}T_m^u - E\|_F$ , получаемый из определенных теоретически геометрических соотношений для платформы и реального (измеренного) расположения связанной системы координат. Здесь  $E$  — единичная матрица,  $\|\cdot\|_F$  — матричная норма Фробениуса,  $T_m^u \in \mathbb{R}^{4 \times 4}$  — преобразование, задающее теоретически рассчитанный поворот и перенос при конкретном  $u^*$  от неподвижной системы  $O^m\xi_1^m\xi_2^m\xi_3^m$ , к подвижной  $N^mx_1^mx_2^mx_3^m$  — связанной с подвижной платформой. Будем использовать приблизительные измерения вектора параметров модели  $\mathbf{td}$  в качестве начального приближения.

Как было указано ранее при теоретическом описании модели стенда, геометрические параметры могли быть заданы неточно. Вследствие этого

возникает разница (ошибка) между определенными эмпирически и теоретически положениями стенда, которую характеризует введенный функционал.

Для проведения эксперимента примем, что количество возможных поворотов роторов двигателей является дискретным. В первом приближении сочетание всех возможных заданных конечных положений ротора определяет множество допустимых положений платформы. Рассмотрим управления  $u^*$ , приводящие систему в каждое из заданных таким образом положений, назовем множество таких управлений  $G$ . Отметим, что размер данного множества требуется выбрать разумно большим для получения большей точности идентификации.

Задача идентификации параметров платформы заключается в подборе такого вектора параметров модели  $\mathbf{td}$ , чтобы

$$\mathbb{J} = \sum_{u^*} F(u^*, \mathbf{td}, T_m^u, T^u) \rightarrow \min_{\mathbf{td} \in TD}.$$

Отметим, что нам не известно точное положение системы координат, связанной с платформой. Поэтому экспериментальное определение положения платформы осуществляется с помощью внешней системы видеоанализа, маркеры которой установлены на платформе. Пример метода оценки поворота объекта с помощью системы видеоанализа будет приведен в главе 3. Особенность предложенного алгоритма идентификации заключается в том, что расположение маркеров системы видеоанализа на платформе может быть произвольным при условии, что количество маркеров не менее 4 штук, и они не расположены в одной плоскости.

Будем считать, что основание стенда неподвижно и совпадает с расположением реальной (физически связанной с комнатой, где находится стенд) неподвижной системы координат  $O\xi_1^r \xi_2^r \xi_3^r$ .

Используя систему видеоанализа для проведения эксперимента получаем координаты светоотражающих маркеров  $P_i^u$  в системе координат  $O\xi_1^r \xi_2^r \xi_3^r$  для всех управляющих значений  $u^*$ , где  $i$  — индекс маркера. Показания  $P_i^u$  будем усреднять на некотором отрезке времени после остановки платформы, чтобы уменьшить влияние шума.

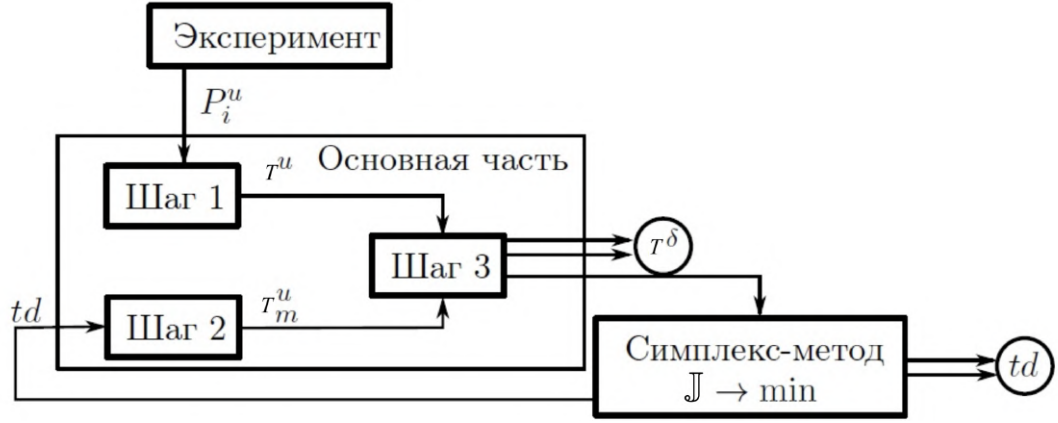


Рисунок 1.9 — Блок-схема алгоритма идентификации

### 1.5.2 Алгоритм идентификации геометрических параметров подвижного стенда

Как было отмечено ранее, задачу идентификации невозможно решить напрямую, поскольку нам неизвестно положение системы координат, связанной со стендом в экспериментальном случае. На рисунке 1.9 приведен алгоритм идентификации параметров, основанный на решении нескольких отдельных задач оптимизации.

Предварительно проводится эксперимент, во время которого измеряются положения маркеров внешней системы видеоанализа на конечном наборе управлений платформой  $u^* \in G$ .

Основная часть алгоритма идентификации параметров состоит из трех шагов:

- 1) Поиск системы координат, связанной с подвижной частью реального стенда, и ее поворота  $T^u$  относительно неподвижной системы для каждого  $u^* \in G$ .
- 2) Построение модельных преобразований  $T_m^u$  для каждого управления  $u^* \in G$ , используя геометрические параметры **td** (изначально заданные неточно).
- 3) Коррекция реальной подвижной системы координат в предположении, что модельная и реальная неподвижные системы координат совпадают, а подвижные также должны совпадать при каждом известном управлении  $u^* \in G$ .

На выходе основной части алгоритма получается значение функционала  $\mathbb{J}$ , который минимизируется численно за счет варьирования геометрических параметров модели платформы **td**.

## Шаг 1. Поиск системы координат, связанной с подвижной частью реального стенда

Найдем средневзвешенную относительно маркеров точку, которая будет совпадать с центром системы координат  $Cx_1^r x_2^r x_3^r$ , связанной с подвижной частью платформы по формуле:

$$C^u = \sum_i \omega_i P_i^u / \sum_i \omega_i,$$

где  $\omega_i$  - соответствующий вес для маркеров.

Найдем преобразование  $T^u$  поворота системы  $Cx_1^r x_2^r x_3^r$  относительно  $O\xi_1^r \xi_2^r \xi_3^r$  для каждого поданного управления  $u^*$ . Смещенные координаты маркеров:  $P_i'^u = P^u - C^u$ . При этом  $\sum_i \omega_i P_i'^u = 0$ . Координаты маркеров в системе  $Cx_1^r x_2^r x_3^r$  обозначим  $S_i'$ . Они не зависят от управления и жестко связаны с подвижной частью. Чтобы найти  $S_i'$  и  $T^u$  используется итерационный метод, где в качестве начального приближения  $S_i'^0 = P_i'^{u_0}$  — усредненное значение координат на первом отработанном цикле управления.

1)  $P_i'^u = T^{uN} \cdot S_i'^N + \Theta_i^{uN}$ , где  $\Theta_i^{uN}$  ошибка для конкретного управления  $u$ , маркера  $i$  на шаге  $N$ .

2)  $S_i'^{N+1} = \sum_{u \in G} P_i'^u \cdot T^{uN} / \sum_{u \in G} 1$ , при условии  $\sum_i \omega_i S_i'^{N+1} = 0$ , а  $\sum_{u \in G} 1$  - количество узлов пути.

## Шаг 2. Построение модельных уравнений платформы

Для модельной системы координат  $N^m x_1^m x_2^m x_3^m$  находим матрицу  $T_m^u$  преобразования относительно системы координат  $O^m \xi_1^m \xi_2^m \xi_3^m$  в точках  $u^* \in G$ , используя модель платформы.

## Шаг 3. Поиск взаимной ориентации модельной и реальной системы координат

Теперь необходимо найти преобразование от модельной  $N^m x_1^m x_2^m x_3^m$  системы к  $Cx_1^r x_2^r x_3^r$ , в предположении что  $O\xi_1^r \xi_2^r \xi_3^r$  и  $O^m \xi_1^m \xi_2^m \xi_3^m$  совпадают.

Чтобы решить эту задачу, необходимо найти минимум функционала:

$$\mathbb{J} \rightarrow \min_{T^\delta}.$$

Для этого используется модифицированный метод Левенберга–Марквардта [78] для использования его при наличии ограничений типа равенства.

На вход основной (внешней) части алгоритма подаются геометрические параметры платформы **td**, а также результаты эксперимента в виде набора  $u^*$  и соответствующих координат маркеров  $P_i^u$ . В то же время, на выходе получаем  $T^\delta = T^u T_m^{u^{-1}}$  на всем наборе управлений, а также функционал  $\mathbb{J}$  после минимизации по  $T^u$ . Исходя из этого же функционала, происходит коррекция параметров платформы **td**:

$$\mathbb{J} \rightarrow \min_{\mathbf{td} \in TD}.$$

Тем самым решается еще одна задача оптимизации на пространстве  $TD$ , в данном случае ввиду высокой размерности пространства геометрических параметров, относительно невысокой точности целевой функции и наличия большого количества локальных минимумов, для ее решения применяется симплекс-метод Нелдера-Мида [34]. Полученные параметры в дальнейшем используются в кинематических и динамических уравнениях движения стенда-платформы.

## 1.6 Построение границы множества допустимых положений для трехстепенной платформы опорного типа

Построим ограничения  $\Omega^w$  (определение 1), используя систему (1.26). Полученная система нелинейных уравнений с параметрами  $(\alpha, \beta, h)$  относительно неизвестных  $(A, B, C)$  была решена численным методом Ньютона с модификацией Исаева-Сониной [88]. В качестве примера рассматривались значения параметров динамического стенда панорамной системы виртуальной реальности МГУ имени М.В. Ломоносова. В эксперименте была применена оптическая система отслеживания движений (трекинга) ARTtrack, производства компании A.R.T. GmbH (Германия) [55].

Для построения множества были использованы следующие параметры, найденные в результате идентификации параметров платформы методом, описанным ранее:

$$\begin{aligned} A_0 &= -5365.56, B_0 = -5450.50, C_0 = -3994.74, \\ V_{A1} &= -414.36, V_{A2} = 0, V_{A3} = 0, V_{B1} = 208.49, V_{B2} = -309.76, V_{B3} = 0, \\ V_{C1} &= 208.49, V_{C2} = 309.76, V_{C3} = 0, S_{A1} = -414.36, S_{A2} = 0, S_{A3} = -1.11, \\ S_{B1} &= 208.49, S_{B2} = -309.76, S_{B3} = -1.11, S_{C1} = 208.49, S_{C2} = 309.76, \\ S_{C3} &= -1.11, r = 102.38, R = 380. \end{aligned}$$

В системе (1.26) возникает неоднозначность решения, которую устраним предположением, что коленвал всегда направлен "наружу" платформы. Вид множества, построенного с указанными параметрами, приведен на рисунке 1.10.

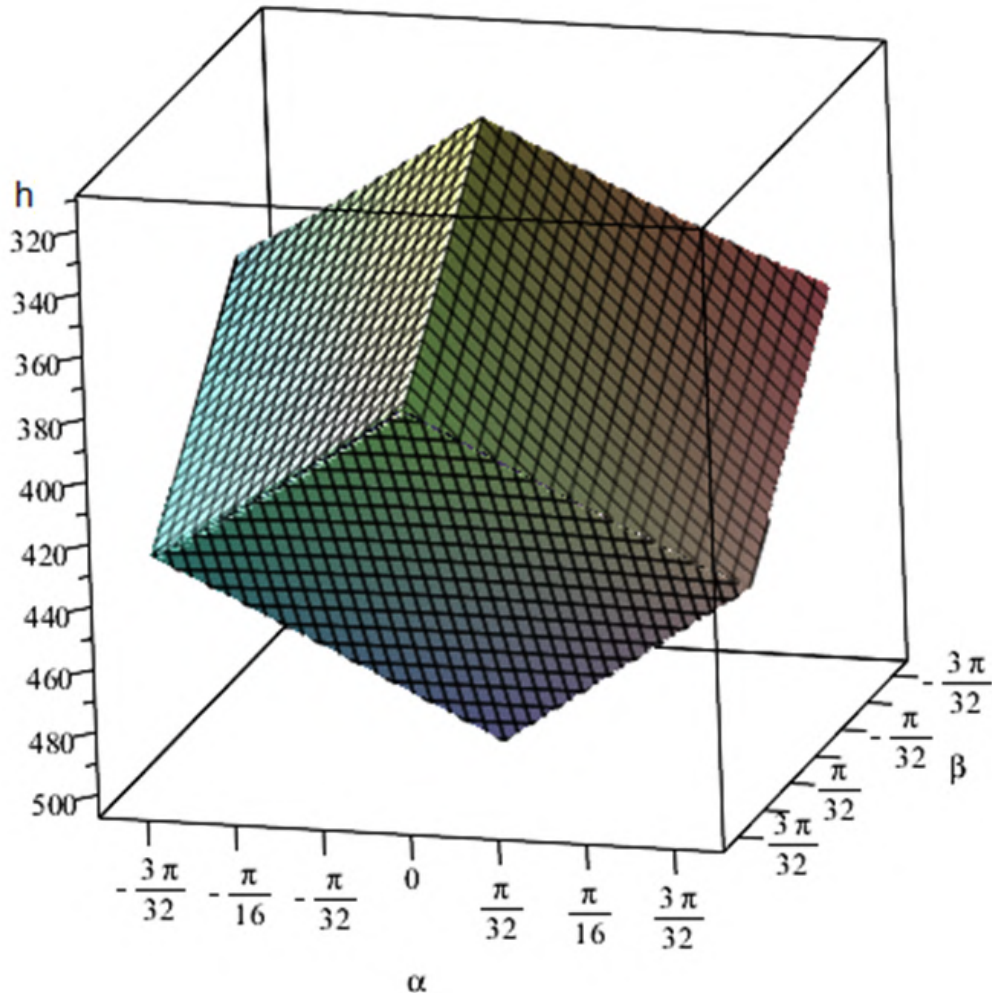


Рисунок 1.10 — Множество допустимых положений стенда-платформы в пространстве  $(\alpha, \beta, h)$  (углы представлены в радианах, высота в мм)

Можно аппроксимировать границу данного множества плоскостями для упрощения его использования в алгоритмах динамической имитации. Для дальнейших расчетов произведем нормировку полученной фигуры. Сделаем это, поместив точку отсчета в центр фигуры и поделив все координаты на их максимальное значение. Были найдены следующие коэффициенты нормировки:  $\alpha_{norm} = 19.2^\circ$ ,  $\beta_{norm} = 20.1^\circ$ ,  $h_{norm} = 110\text{мм}$ . Введем новые координаты:  $x = \alpha/\alpha_{norm}$ ,  $y = \beta/\beta_{norm}$ ,  $z = h/h_{norm}$ .

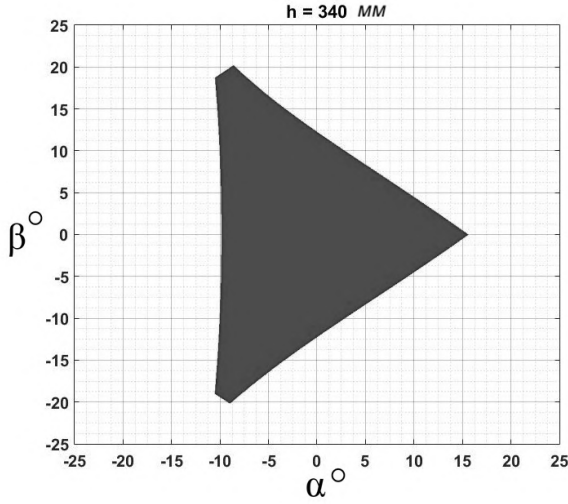


Рисунок 1.11 — Срез фигуры на высоте  $h_{max}/3$

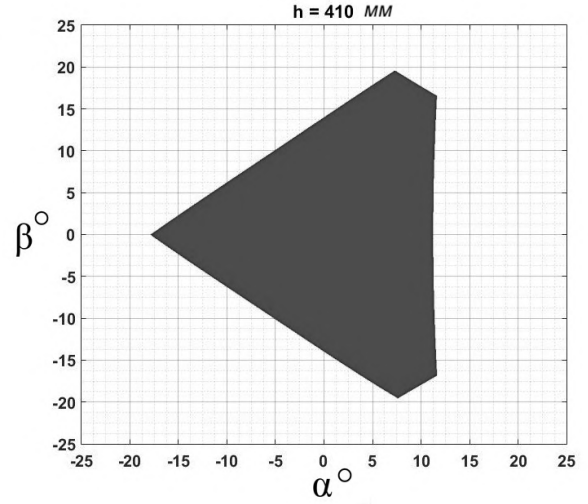


Рисунок 1.12 — Срез фигуры на высоте  $2h_{max}/3$

Выделим 6 граней получаемой фигуры, близкой по своей форме к многограннику (рис. 1.10), и аппроксимируем их плоскостями. В результате аппроксимации были получены следующие плоскости:

$$0.7655x + 1.0338y - 1.0174z - 1 = 0,$$

$$0.7655x + 1.0338y - 1.0174z + 1 = 0,$$

$$0.7655x - 1.0338y - 1.0174z + 1 = 0,$$

$$0.7655x - 1.0338y - 1.0174z - 1 = 0,$$

$$1.2361x + 1.0074z - 1 = 0,$$

$$1.2361x + 1.0074z + 1 = 0.$$

Следующим этапом необходимо оценить, насколько хорошо плоскости аппроксимируют грани исходной фигуры. Для этого построим график ошибки, показывающий расстояние от точек грани до аппроксимирующей плоскости. В качестве примера возьмем одну произвольную грань (рис 1.13). Как видно из рисунка, ошибка приближения на три порядка меньше характерных размеров фигуры. Полученные аналитические представления плоскостей достаточно точно приближают реальные грани фигуры, но при использовании аналитически заданных границ будет захватываться часть области, в которой платформа физически не может оказаться. Сузим множество, сдвинув плоскости к началу координат таким образом, чтобы все точки граней оказались вне фигуры, составленной из плоскостей. После сдвига плоскостей,

объем фигуры отличается от исходной на величину порядка 0,01%, что можно считать допустимо малой величиной.

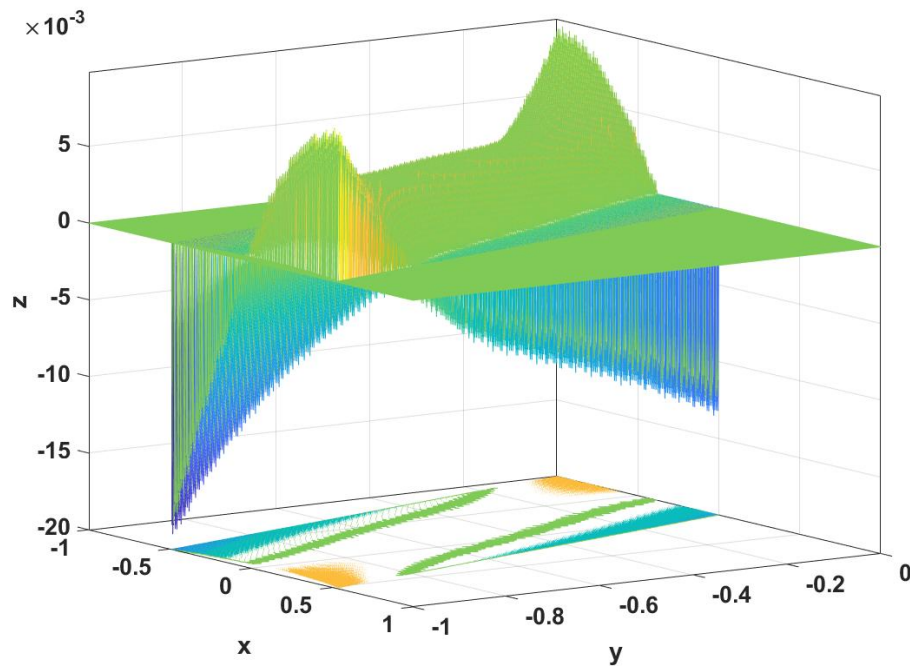


Рисунок 1.13 — Ошибка аппроксимации одной из границ множества допустимых положений платформы плоскостью

Полученную после преобразования границу множества допустимых положений платформы можно описать следующими плоскостями:

$$\begin{aligned}
 0.7655x + 1.0338y - 1.0174z - 0.99 &= 0, \\
 0.7655x + 1.0338y - 1.0174z + 0.99 &= 0, \\
 0.7655x - 1.0338y - 1.0174z + 0.99 &= 0, \\
 0.7655x - 1.0338y - 1.0174z - 0.99 &= 0, \\
 1.2361x + 1.0074z - 0.99 &= 0, \\
 1.2361x + 1.0074z + 0.99 &= 0.
 \end{aligned} \tag{1.28}$$

### 1.6.1 Задача построения рабочей области для трехстепенной платформы опорного типа

Как можно заметить из формы построенного множества допустимых положений, при изменении высоты платформы будут изменяться и максимальные значения углов, которые можно воспроизвести. Кроме того, при изменении значения одного из углов, также будут меняться максимальные значения и для другого угла. Это означает, что во время управления плат-



формой придется на каждом шаге имитации пересчитывать текущие максимальные значения по всем степеням свободы. Кроме этого, возникнет проблема оптимального использования доступного диапазона имитации: насколько большая амплитуда движения допустима по одной из степеней свободы, чтобы оставалась возможность реалистичной имитации по остальным степеням свободы.

Чтобы избежать подобных проблем, впишем в область допустимых положений платформы прямоугольный параллелепипед. Вписывание прямоугольного параллелепипеда в область допустимых положений уменьшит возможности, но позволит сделать алгоритмы имитации независимыми.

Также существуют различные режимы динамической имитации, отличающиеся приоритетом между описанными ранее задачами. Для самолета можно привести следующие примеры: движение по взлетно-посадочной полосе и движение по глиссаде. Во втором случае будут преобладать изменения ориентации вектора перегрузки, а в первом случае будет преобладать имитация быстрых изменений перегрузки по вертикальной оси (момент касания ЛА земли, неровности земли). Для колесного транспорта можно рассмотреть движение с большим количеством поворотов и движение по пересеченной местности. В различных режимах управления объектом могут преобладать движения по разным степеням свободы.

Для построения рабочей области в виде вписанного параллелепипеда максимального объема воспользуемся методом множителей Лагранжа [6].

С целью обеспечения максимального использования ресурсов платформы для динамической имитации объекта во всех режимах управления, предлагается адаптивно менять параметры параллелепипеда. Увеличение ресурсов для одной степени свободы будет происходить за счет уменьшения ресурсов другой. Таким образом, во время работы алгоритмов имитации, параллелепипед будет адаптироваться под текущие потребности.

### **Поиск рабочей области максимального объема**

Приведем постановку и способ решения задачи при помощи метода множителей Лагранжа. Ставится задача на условный экстремум:

$$\begin{cases} \varphi_0(\mathbf{x}) \rightarrow \max, \\ \varphi_i(\mathbf{x}) \leq 0, & i = 1, \dots, m. \end{cases} \quad (1.29)$$

Ее область определения (допустимое множество)  $\Omega$  имеет вид

$$\Omega = \{\mathbf{x} \in \mathbb{R}^n | \varphi_i(\mathbf{x}) \leq 0, i = 1, \dots, m\}.$$

Алгоритм решения задачи:

1. Составляем функцию Лагранжа:

$$L(\mathbf{x}, \lambda) = \sum_{i=0}^m \lambda_i \varphi_i(\mathbf{x}).$$

2. Выписываем необходимое условие экстремума:

$$\begin{cases} \frac{\partial L}{\partial \mathbf{x}_j} = \sum_{i=0}^m \lambda_i \frac{\partial \varphi_i(\mathbf{x})}{\partial \mathbf{x}_j} = 0, & j = 1, \dots, n, \\ \lambda_i \varphi_i(\mathbf{x}) = 0, & i = 1, \dots, m. \end{cases} \quad (1.30)$$

3. Решаем полученную систему.

4. Проверяем, лежат ли найденные точки в области  $\Omega$  и являются ли они экстремальными для задачи (1.29).

Впишем параллелепипед максимального объема. Поскольку требуется, чтобы отклонения от среднего положения в положительном и отрицательном направлениях были равны, определение координат вершин параллелепипеда может быть осуществлено следующим образом:

$$\begin{aligned} v_1 &= \{-x, -y, -z\}, & v_5 &= \{-x, -y, z\}, \\ v_2 &= \{-x, y, -z\}, & v_6 &= \{-x, y, z\}, \\ v_3 &= \{x, -y, -z\}, & v_7 &= \{x, -y, z\}, \\ v_4 &= \{x, y, -z\}, & v_8 &= \{x, y, z\}. \end{aligned} \quad (1.31)$$

Введем функцию:

$$F_i(v_j) = A_i v_{j1} + B_i v_{j2} + C_i v_{j3} + D_i, \quad (1.32)$$

где  $A_i, B_i, C_i, D_i$  — коэффициенты плоскости из (1.28),  $v_{jk}$  — координаты точки.

Введем параметры  $X, Y, Z$  для осей  $x, y, z$  соответственно, которые будут отвечать за гарантированный ресурс по каждой из степеней свободы. Этот подход можно рассматривать как резервирование определенной области и решение задачи максимизации для оставшейся.

Поскольку осуществляется поиск максимального объема рабочей области, можно ввести следующую функцию:

$$\varphi_0(x,y,z) = (x - X)(y - Y)(z - Z).$$

Для определения области  $\Omega$ , необходимо задать ограничения для каждой из 8 точек. Каждая из точек должна находиться либо на границе области допустимых положений платформы, либо внутри нее. Это означает, что для каждой из 8 точек имеются по 6 ограничений, в итоге получаем 48 неравенств.

При составлении системы (1.30), получим 3 уравнения при взятии частных производных от функции Лагранжа и 48 уравнений, задающих ограничения. Получаем нелинейную систему из 51 уравнения. Возможно провести некоторые упрощения.

Для упрощения воспользуемся симметрией фигуры. Отметим, что фигура симметрична относительно плоскости, проведенной через ее центр, и симметрична относительно начала координат. Поэтому можно ограничиться рассмотрением только двух точек и ограничениями по трем плоскостям на точку. Также ограничим значения  $x,y,z$ , они должны быть больше  $X,Y,Z$  соответственно. Очевидно, на параметры  $X,Y,Z$  накладываются те же самые ограничения, что и на  $x,y,z$  — параметры должны находиться внутри области достижимости. Для решения задачи выберем точки  $v_2$  и  $v_4$ . Выпишем ограничения на эти точки:

$$\begin{aligned} F_5(v_4) &\leq 0, & -F_6(v_2) &\leq 0, \\ F_1(v_4) &\leq 0, & F_1(v_2) &\leq 0, \\ -F_3(v_4) &\leq 0, & -F_3(v_2) &\leq 0. \end{aligned}$$

Выпишем функцию Лагранжа:

$$\begin{aligned} L(x,y,z,\lambda) = & \lambda_0(x - X)(y - Y)(z - Z) - \\ & \lambda_1(x - X) - \lambda_2(y - Y) - \lambda_3(z - Z) + \\ & (\lambda_4 F_5(v_4) + \lambda_5 F_1(v_4) - \lambda_6 F_3(v_4)) + \\ & (-\lambda_7 F_6(v_2) + \lambda_8 F_1(v_2) - \lambda_9 F_3(v_2)) \quad (1.33) \end{aligned}$$

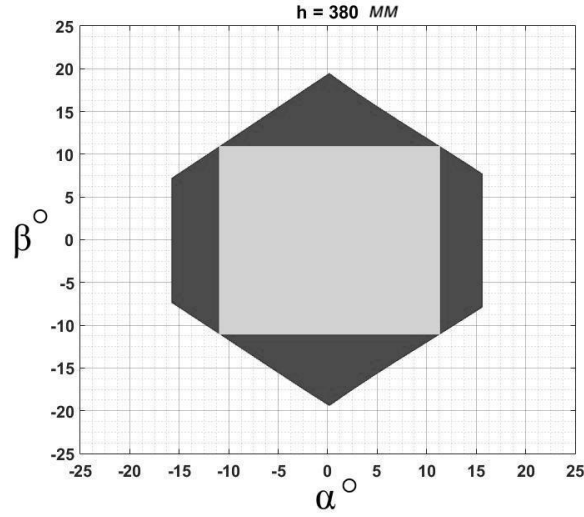


Рисунок 1.14 — Пример вписанного параллелепипеда в пространстве углов

Выпишем полную систему:

$$\left\{ \begin{array}{ll} \frac{\partial L}{\partial x} = 0, & \lambda_4 F_5(v_4) = 0, \\ \frac{\partial L}{\partial y} = 0, & \lambda_5 F_1(v_4) = 0, \\ \frac{\partial L}{\partial z} = 0, & \lambda_6 F_3(v_4) = 0, \\ \lambda_1(x - X) = 0, & \lambda_7 F_6(v_2) = 0, \\ \lambda_2(y - Y) = 0, & \lambda_8 F_1(v_2) = 0, \\ \lambda_3(z - Z) = 0, & \lambda_9 F_3(v_2) = 0. \end{array} \right. \quad (1.34)$$

где  $L$  задано в (1.33), точки  $v_2$  и  $v_4$  заданы в (1.31), а плоскости в выражении (1.28).

Для известных параметров трехстепенной платформы система имеет следующее приближенное решение при  $x = \alpha/\alpha_{norm}$ ,  $y = \beta/\beta_{norm}$ ,  $z = h/h_{norm}$ :

$$\left\{ \begin{array}{l} \beta_{max} \approx 8.7 \cdot (5 + 7.7X - 5.2Y - 5Z) \cdot 10^{-2}, \\ \alpha_{max} \approx 3.2 \cdot (10 - 7.7X + 20.7Y - 10.1Z) \cdot 10^{-2}, \\ h_{max} \approx 3.3 \cdot (10 - 7.7X - 10.3Y + 20.3Z) \cdot 10^{-2}. \end{array} \right. \quad (1.35)$$

Полученное решение позволяет описать вписанный параллелепипед в исходных координатах:

$$\Omega = \{(\alpha, \beta, h) \in \mathbb{R}^3 \mid |\alpha| \leq \alpha_{max}, |\beta| \leq \beta_{max}, |h| \leq h_{max}\}.$$

Заметим, что величины  $\beta_{max}$ ,  $\alpha_{max}$  и  $h_{max}$  линейно зависят от  $X$ ,  $Y$  и  $Z$ . Это указывает на то, что изменение гарантированных ресурсов по каждой из степеней свободы приводит к пропорциональному изменению размеров рабочей области. Исходя из знаков коэффициентов в решении, можно заключить, что увеличение параметра  $X$  приводит к уменьшению  $\alpha_{max}$  и  $h_{max}$ , но увеличению  $\beta_{max}$ . Аналогично, увеличение параметра  $Y$  приводит к увеличению  $\alpha_{max}$ , но уменьшению  $\beta_{max}$  и  $h_{max}$ . Увеличение параметра  $Z$  приводит к увеличению  $h_{max}$ , но уменьшению  $\alpha_{max}$  и  $\beta_{max}$ . Это позволяет сделать вывод о том, как изменение гарантированных ресурсов по каждой из степеней свободы влияет на размеры рабочей области в разных направлениях.

Пример вписанного параллелепипеда в пространстве углов, представленный на рисунке 1.14, иллюстрирует, как оптимальное распределение ресурсов может обеспечить максимальный объем рабочей области. Важно учитывать это в процессе проектирования алгоритмов динамической имитации для трехстепенной платформы, чтобы обеспечить эффективное использование ее возможностей.

## 1.7 Структура алгоритма динамической имитации быстрых угловых и линейных движений

Имитация только ориентации вектора перегрузки имеет ряд недостатков. Таким образом, возникает необходимость частичной имитации изменения модуля перегрузки, так как пилоты отчётливо ощущают его во время полёта, наряду с ориентацией вектора перегрузки. Определим передний фронт как производную вектора перегрузки [72].

В ряде ситуаций возникает потребность в имитации изменения модуля перегрузки. Например, атмосферные возмущения могут оказывать существенное влияние на летательный аппарат (ЛА).

Рассмотрим пример прямолинейного движения ЛА на постоянной высоте. График зависимости высоты от времени в полёте ЛА представлен на рисунке 1.15. На рисунках представлены записи траекторий, полученные из симулятора динамики тяжелого летательного аппарата при выполнении взлета-координированного разворота на небольшой высоте-посадки.

Если аппарат движется с постоянной скоростью и поддерживает заданную высоту, то график выглядит плавным и ровным (рис. 1.16).

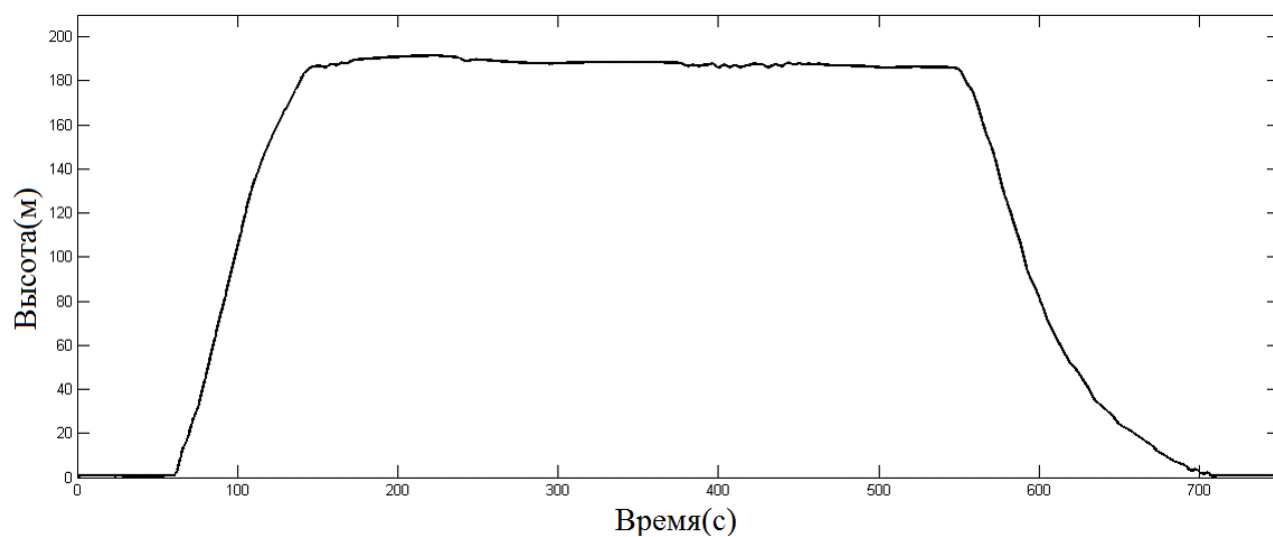


Рисунок 1.15 — График движения ЛА

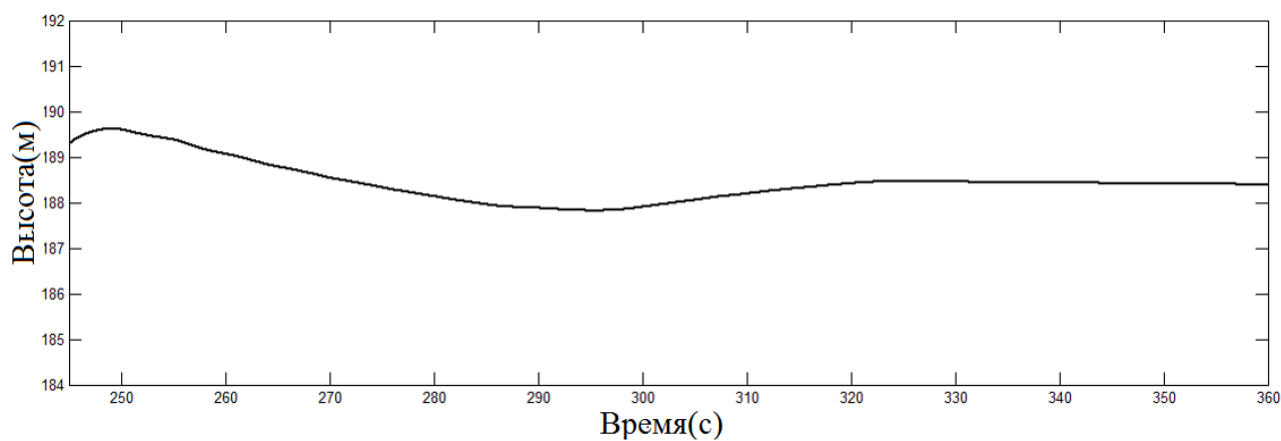


Рисунок 1.16 — Отрезок движения с постоянной скоростью

Допустим, лётчик решает изменить скорость движения, оставаясь на текущей высоте. В этом случае необходимо изменить угол атаки. Во время изменения угла происходит переходный колебательный процесс, который явно ощущается пилотом (рис. 1.17).

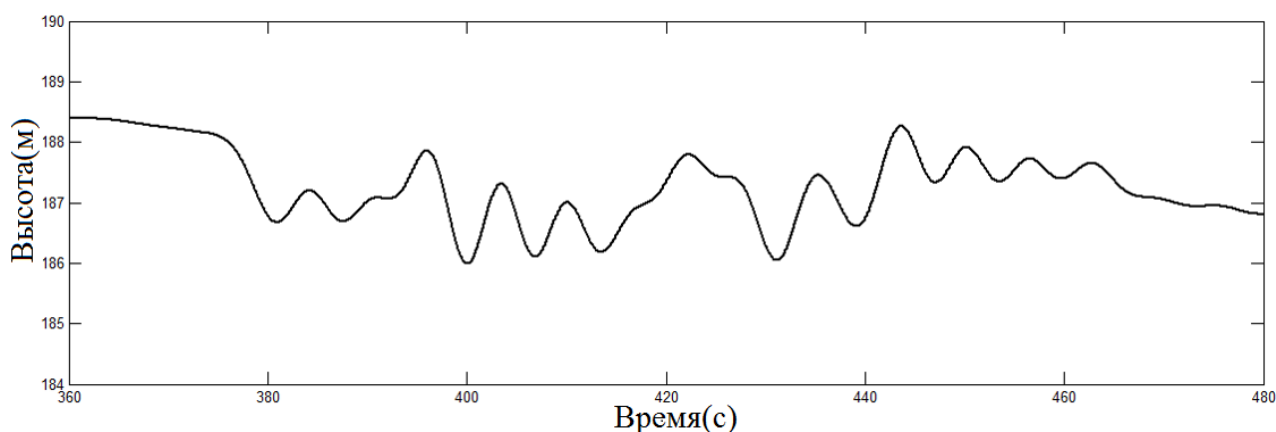


Рисунок 1.17 — Отрезок движения в момент изменения скорости движения

Необходимо имитировать изменение перегрузки на платформе. Учитывая порог чувствительности линейных ускорений у человека (пилот не ощущает ускорения ниже определенного порога), имитация не выполняется для всех движений, которые вызывают ускорение ниже порога чувствительности. Имитация прекращается, когда ускорение становится ниже порога чувствительности или положение платформы достигает границ рабочей области.

Учитывая наличие ограничений, процесс динамической имитации можно разбить на три этапа (рис. 1.18):

1. Активная фаза — процесс поиска таких управлений стендом, чтобы решалась задача минимизации функционала (1.19).
2. Остановки — торможения на границе.
3. Возврата в некоторое стартовое положение или область.

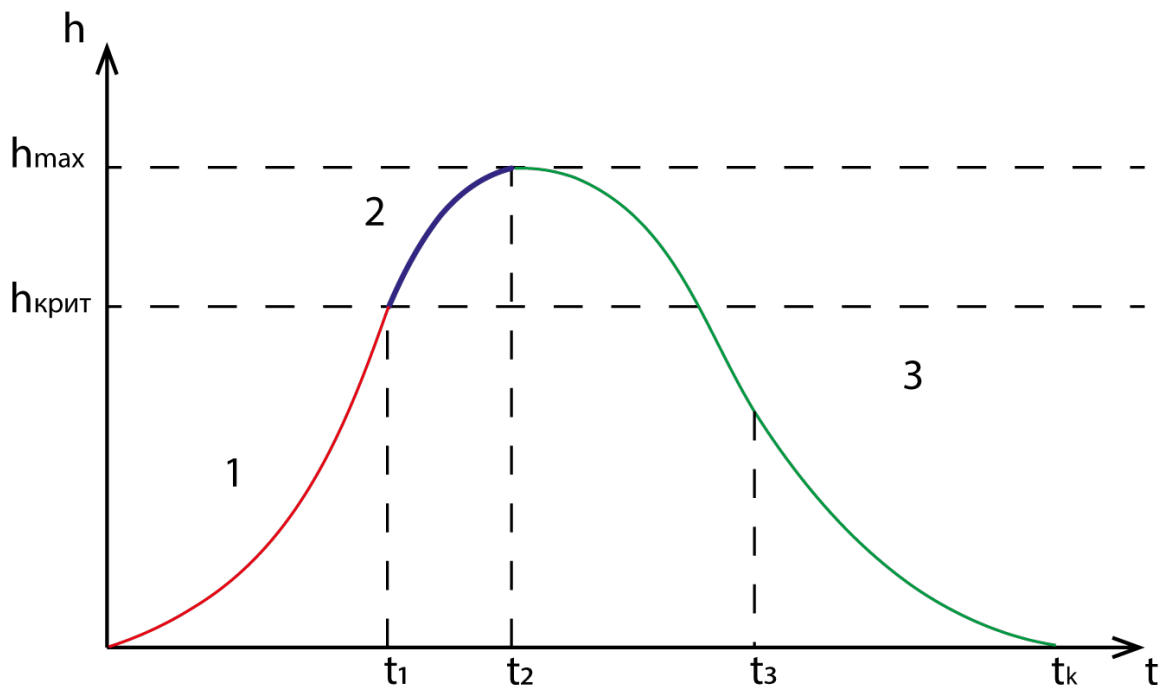


Рисунок 1.18 — Пример имитирующего движения

Считаем, что движение в процессе имитации удовлетворяет ограничениям стенда, включая ограничения на моменты, развиваемые электродвигателями, прочность конструкции и т.д. Одно из ключевых ограничений для стенда заключается в соблюдении условия, чтобы его движение не выходило за границы множества допустимых положений. Таким образом, предполагается, что процесс динамической имитации будет осуществляться в пределах обозначенного множества  $\Omega$ , полностью включенного во множество допустимых положений.

Возврат в "стартовое" положение на последнем этапе динамической имитации не является непосредственной частью процесса имитации. Поэтому во время этого этапа человек не должен ощущать движение. Это достигается благодаря тому, что движение осуществляется с ускорениями и угловыми скоростями, ниже пороговых значений чувствительности отолитовых органов и полукружных каналов человека.

### 1.7.1 Построение простейшего алгоритма оптимальной остановки на границе и возврата в начало координат

#### Математическая формулировка задачи

При моделировании желаемой траектории для первой (активной) фазы используется динамика модели транспортного средства. Для второй и третьей фазы рассмотрим построение траектории на примере одной координаты. Отметим, что процесс возврата не является частью имитации движения, и необходимо избегать удара. Условие отсутствия удара можно определить как требование к движению платформы с ускорениями ниже определенного порогового значения  $\epsilon_h$ .

Пусть  $h$  обозначает высоту платформы, причем нулевое положение определяется как среднее положение  $([h_{min} + h_{max}]/2)$ . В начальный момент времени платформа имеет некоторое отклонение от нулевого положения и начальную скорость.

Обозначим конечный момент времени за  $t_k$ . Цель состоит в том, чтобы вернуть платформу в нулевое положение за минимально возможное время, то есть минимизировать  $t_k$ . Введем  $x_1 = h$ ,  $x_2 = \dot{h}$ , и сформулируем задачу:

$$\begin{cases} \dot{x}_1 = x_2, \\ \dot{x}_2 = u, \end{cases} \quad (1.36)$$

$$\begin{aligned} x_1(0) &= h(0), & x_1(t_k) &= 0, \\ x_2(0) &= \dot{h}(0), & x_2(t_k) &= 0, \end{aligned}$$

$$J = t_k \rightarrow \min,$$

Оптимальное управление можно получить с использованием принципа максимума Понтрягина [105], аналогично работе [72]. Управление имеет величину  $-\epsilon_h$  или  $\epsilon_h$  с не более чем одним переключением (рисунок 1.19).



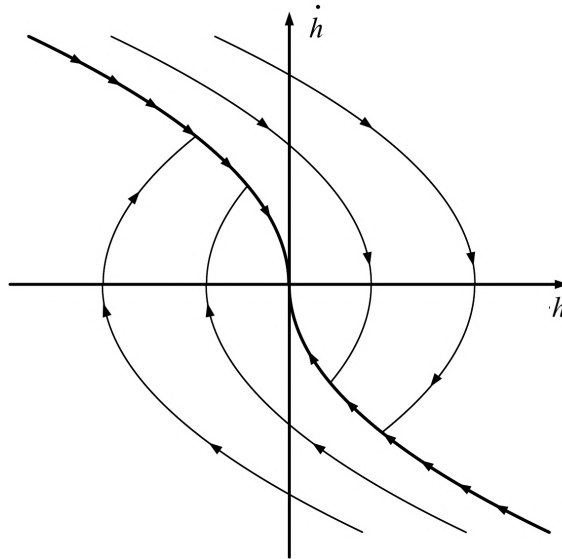


Рисунок 1.19 — Синтез оптимального управления

### Построение рабочей области

Не из всех начальных положений на фазовой плоскости  $(h, \dot{h})$  можно остановиться и вернуться, не нарушая ограничения по ускорению. Рассмотрим вышеописанную оптимальную задачу в обратном времени и построим представление области активной фазы имитации в фазовом пространстве (рисунок 1.20).

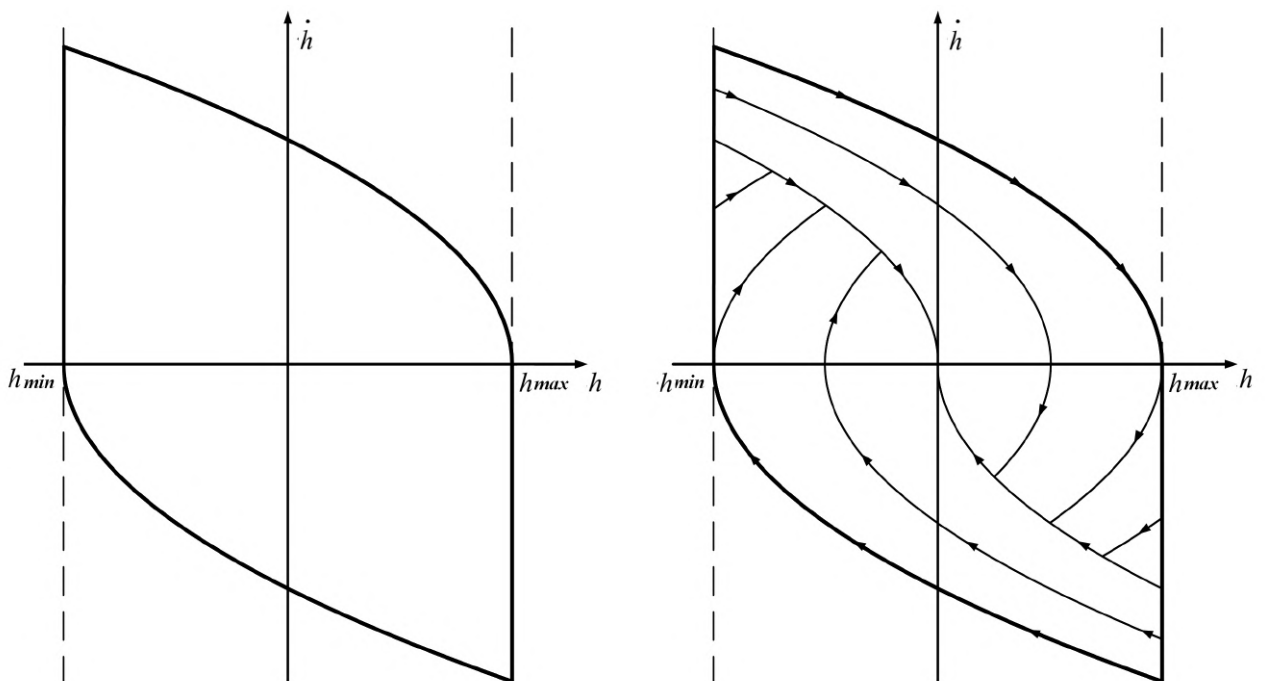


Рисунок 1.20 — Рабочая область в фазовом пространстве

## 1.8 Реализация алгоритма имитации быстрых линейных и угловых движений на примере трехстепенной платформы опорного типа

### 1.8.1 Синтез алгоритмов динамической имитации

Как упоминалось ранее, процесс имитации состоит из трех этапов: активная фаза имитации, торможение и возврат. Имитируются только те дви-

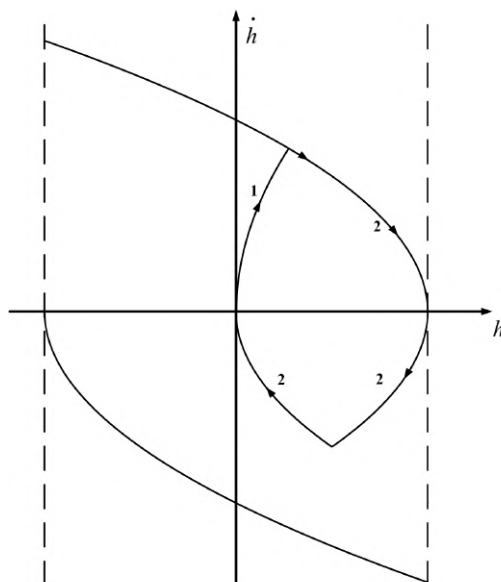


Рисунок 1.21 — Имитация быстрых движений с учетом остановки на границе

жения, ускорение которых превышает порог чувствительности. Переход с первого этапа на второй происходит в момент выхода платформы на границу области управления или когда ускорение становится ниже порогового значения.

Особое внимание следует уделить этапу возврата. На данном этапе ускорения, которые необходимо имитировать, не игнорируются. Если имеется запас ресурсов для имитации, происходит переход к активной фазе имитации. Предлагается следующий подход: если платформа находится на этапе возврата в правой полуплоскости ( $h > 0$ ), имитация ускорений с отрицательными значениями разрешается. Если платформа находится на этапе возврата в левой полуплоскости ( $h < 0$ ), имитация ускорений с положительными значениями разрешается.

Теперь, когда алгоритмы управления для каждого случая известны, рассмотрим их взаимодействие. Алгоритм имитации ориентации вектора перегрузки изменяет угол платформы, в то время как алгоритм имитации быст-

рых линейных движений контролирует координаты. Построение параллелепипеда обеспечивает полную независимость алгоритма имитации быстрых движений.

Таким образом, решение задачи вписывания прямоугольного параллелепипеда в область допустимых положений платформы обеспечивает полную независимость управления трехстепенной подвижной платформой опорного типа по каждой из степеней свободы от других степеней свободы. Это позволяет применять алгоритм имитации ориентации вектора перегрузки независимо от алгоритма имитации быстрых линейных движений.

### 1.8.2 Экспериментальные результаты

Для проверки работоспособности метода и оценки качества динамической имитации, эксперименты были проведены на панорамной системе виртуальной реальности МГУ. В ходе экспериментов оценивалась эффективность каждого из предложенных алгоритмов динамической имитации.

Исследование показало, что алгоритм имитации ориентации вектора перегрузки работает корректно с точностью до единиц градусов, а алгоритмы имитации угловых ускорений и переднего фронта модуля перегрузки успешно имитируют большую часть ускорений, превышающих порог чувствительности. Однако, алгоритмы не справляются с имитацией пиковых ускорений, возникающих в момент возврата алгоритма в ноль, когда нет ресурсов для их имитации.

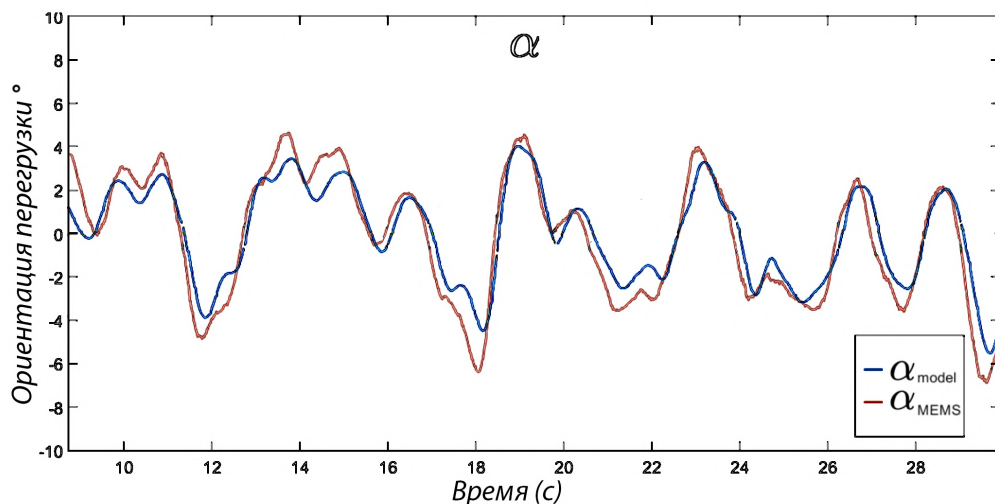


Рисунок 1.22 — Сравнение угла  $\alpha$

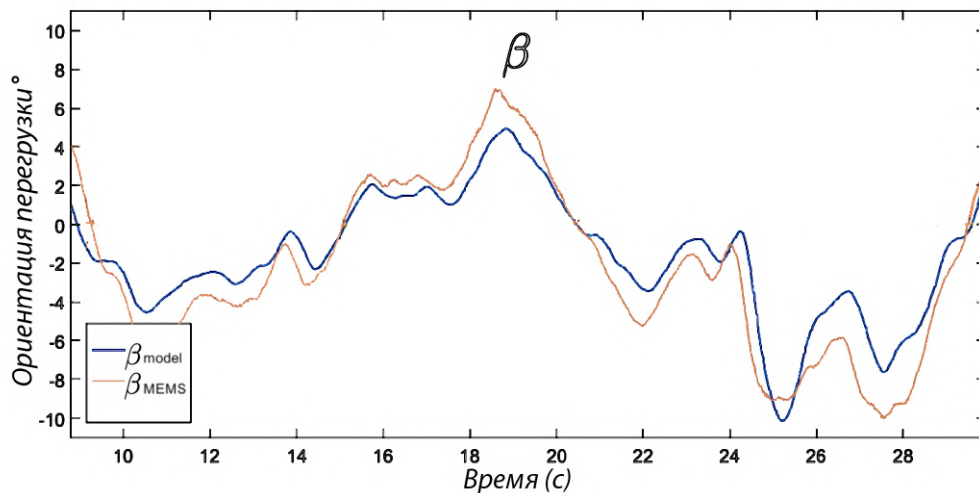


Рисунок 1.23 — Сравнение угла  $\beta$

Одним из основных проблем при реализации оказалось использование целочисленного формата времени, измеряемого в миллисекундах. Это привело к ошибкам при переключении на стадии возврата, что повлияло на точность возврата в ноль. Кроме того, была выявлена проблема досрочного прекращения имитации быстрых движений. Если движение выходило за границы области достижимости, расчёты точки пересечения не производились, и имитация прерывалась. В результате, движения, выходящие за область управления за один такт, не имитировались вовсе. Все обнаруженные ошибки и замечания были устранены в дальнейшем.

Рисунки (1.22 - 1.24) иллюстрируют работу алгоритмов динамической имитации на примере части маневра, приведенного ранее. На рисунках (1.22 - 1.23) представлено сравнение углов ориентации вектора перегрузки, полученных с математической модели, и реальных углов платформы, полученных с помощью инерциальных датчиков ( $\alpha_{MEMS}$ ,  $\beta_{MEMS}$ ). Рисунок (1.24) демонстрирует работу алгоритма имитации переднего фронта модуля перегрузки на фазовой плоскости. Отметим, что форма траекторий качественно соответствует полученным ранее теоретическим результатам.

Экспериментальные результаты подтверждают общую работоспособность предложенных алгоритмов динамической имитации. Алгоритм имитации ориентации вектора перегрузки показал корректную работу, в то время как алгоритмы имитации угловых ускорений и переднего фронта модуля перегрузки успешно имитировали большую часть ускорений, превышающих порог чувствительности, с некоторыми ограничениями при пиковых ускорениях.

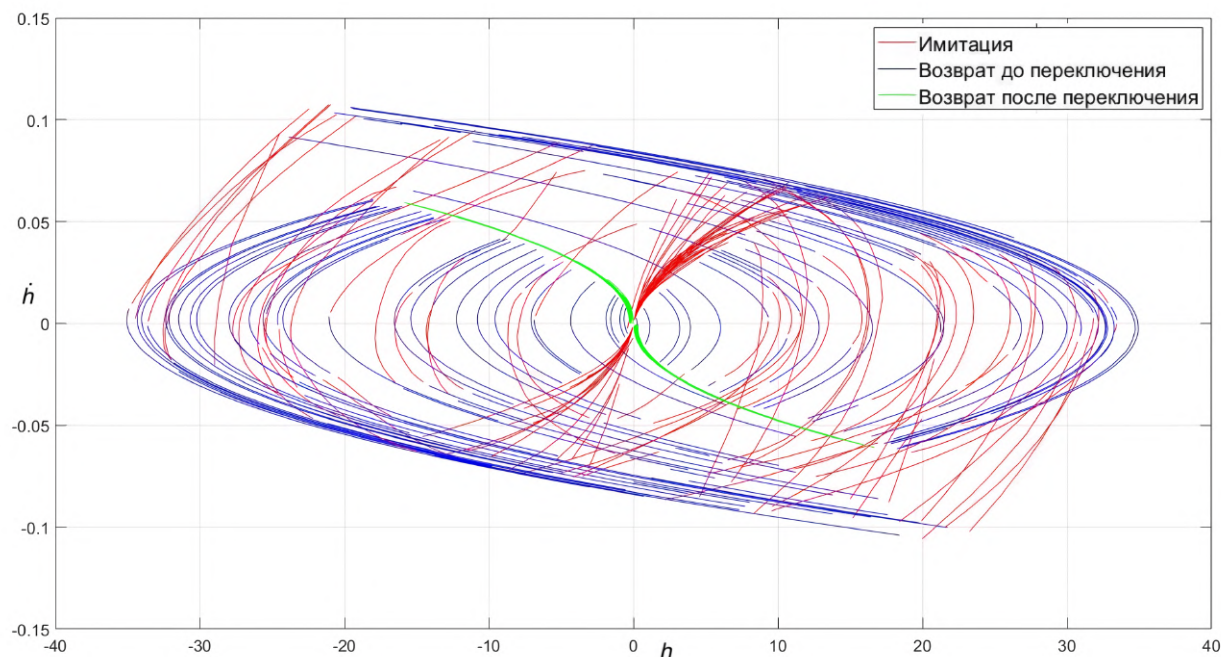


Рисунок 1.24 — Работа алгоритма имитации переднего фронта модуля перегрузки

Результаты исследования показывают, что алгоритмы динамической имитации обеспечивают качественное моделирование движений, что позволяет использовать их для разработки систем виртуальной реальности и тренажеров. Однако стоит отметить, что для оптимизации и повышения точности имитации потребовалось учесть обнаруженные проблемы и разработать методы их решения.

Одним из возможных подходов к решению проблемы ограниченности ресурсов стенда является использование робота-манипулятора в качестве основы для стенда. Этот подход позволяет повысить гибкость и масштабируемость системы, а также обеспечить более широкий диапазон сценариев имитации. Детальное описание этого варианта и его особенностей представлено в главе 4.

## **Глава 2. Визуализация виртуального окружения при управлении динамической системой**

<sup>1</sup>В предыдущей главе были подробно рассмотрены алгоритмы динамической имитации, которые позволяют эффективно обучать пользователей на тренажере и симуляторе, создавая для них реалистичные условия. Однако, для достижения наилучших результатов в обучении, также крайне важно предоставить пилоту качественную визуализацию виртуального окружения. В данной главе будут изучены основные принципы создания визуализации, используемые технологии и их особенности, а также рассмотрим примеры систем отображения, подходящих для различных задач.

### **2.1 Принципы создания визуализации**

В некоторых простых ситуациях достаточно воспроизвести приборную панель и необходимую индикацию в натуральную величину на стандартном мониторе, обеспечивающем моноизображение. Однако симуляция на мониторе не задействует периферическое зрение и не создает "эффекта погружения" [33], поскольку при повороте головы испытуемый видит рабочую комнату.

Бинокулярное зрение человека позволяет оценивать расстояние до объектов и определять их объем. Отсутствие объемности изображения затрудняет обучение пилота определенным навыкам, связанным с оценкой расстояний. В связи с этим возникает необходимость создания стереовизуализации, которая задействует периферическое зрение.

### **Системы отображения экранного типа**

Для полноценной визуальной имитации используются панорамные экраны различных конструкций и шлемы виртуальной и дополненной реальности. Рассмотрим подробно технологии, применяемые в экранных системах и в шлемах виртуальной реальности.

---

<sup>1</sup>При подготовке данного раздела диссертации использованы следующие публикации, выполненные автором лично или в соавторстве, в которых, согласно Положению о присуждении ученых степеней в МГУ, отражены основные результаты, положения и выводы исследования: [1; 22; 70; 91; 111; 116]

Технология панорамных экранов, задействующих периферическое зрение, была разработана в первой половине XX века. На Всемирной выставке в Нью-Йорке в 1939 году представили панорамный экран многопроекторной системы Vitarama [100]. Несмотря на то, что эта технология предоставляла большой угол обзора, она не могла моделировать объем изображения.

Давно существующая технология стереоизображений имитировала эффект объема, однако не обеспечивала достаточно широкого угла обзора для пользователя. Интеграция панорамных экранов с методами стереоизображений значительно расширила возможности визуализации.

Современные методы визуализации включают разнообразные конфигурации проекционных систем, различающихся по ряду параметров. Одним из основных является форма экрана (рисунок 2.1).

Для точной визуализации событий в виртуальной среде необходимо регулярно обновлять и перестраивать изображение. Процесс создания и отображения виртуальной сцены на основе специального описания называется рендеринг (rendering). Частота обновления изображения должна превышать 24 Гц [87] для центрального зрения и 60 Гц для периферического, чтобы пользователь не замечал обновлений. Реалистичность визуальной имитации и уровень погружения улучшаются благодаря стереоскопическому изображению.

### **Стереоскопическое изображение**

Трехмерное изображение объекта формируется в мозге человека на основе данных, полученных обоими глазами. Мозг анализирует две проекции одного изображения на сетчатку глаз с разным смещением — стереопару. Стереопара естественным образом возникает при сведении зрительных осей глаз на одной точке, что позволяет видеть объект с разных ракурсов и оценивать расстояние до него. Однако, при взгляде на плоский экран, человек видит одну плоскую картинку, из-за чего стереопара не формируется естественным образом.

В некоторых случаях, для создания реалистичного изображения достаточно имитировать бесконечное удаление визуализируемых объектов (например, при имитации полета на большой высоте) [38]. Это достигается с помощью коллиматорных зеркальных систем [46]. Изображение проецируется на полупрозрачный экран под углом 45 градусов к наблюдателю, отражается

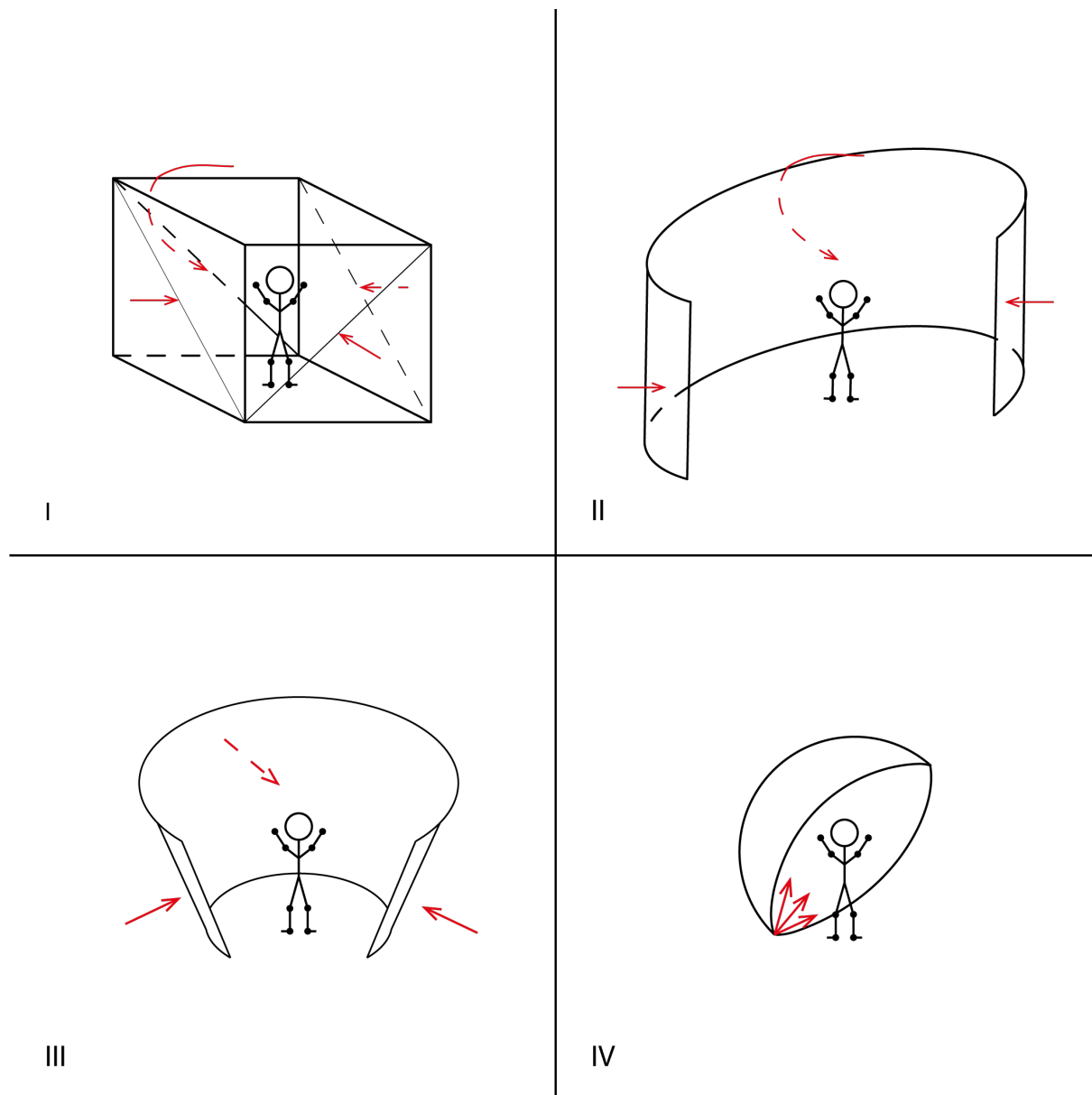


Рисунок 2.1 — Типы экранов: I — CAVE, II — цилиндрический экран, III — iCone display, IV — iDome display

от сферического зеркала и направляется к наблюдателю. Проектор размещается так, чтобы его линза находилась в оптическом центре зеркала, создавая иллюзию бесконечного удаления плоского изображения.

В общем случае, для создания 3D-эффекта на экране, необходимо искусственно формировать стереопару и проецировать разные изображения стереопары на сетчатку каждого глаза. При этом изображение рассчитано на наблюдателя, находящегося в строго определенной и заранее известной точке. Создание и воспроизведение стереопары позволяет пользователю воспринимать объемные объекты и сцены, улучшая ощущение погружения в виртуальную среду.



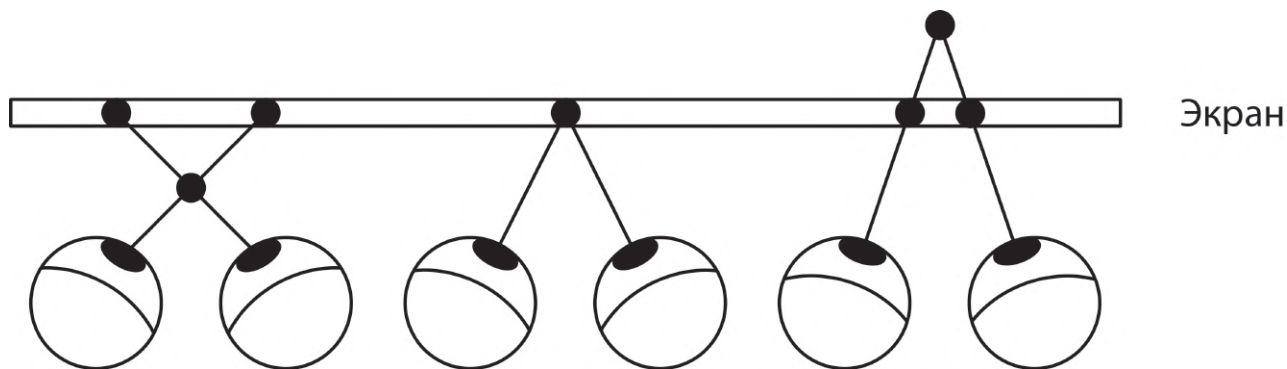


Рисунок 2.2 — Наглядная схема создания стереоскопического изображения

Для разделения изображений между глазами используются специальные очки с фильтрами, известные как 3D-очки. В основном существуют четыре типа 3D-очков:

- **Анаглифные очки:** Вместо линз эти очки имеют светофильтры красного и синего/зеленого цветов.
- **Поляризационные очки:** Вместо линз эти очки имеют поляризационные фильтры различной направленности.

Анаглифные и поляризационные очки используются для разделения двух изображений, одновременно выводимых на экран. В этих случаях на экран проецируются два изображения с разными углами или цветовыми спектрами (анаглиф) либо через разные поляризующие фильтры (поляризация). Отражаясь от экрана, свет проходит через фильтры в очках и достигает сетчатки глаз. Таким образом, каждый глаз видит только одно изображение, что позволяет зрителю воспринимать стереопару и создает 3D-эффект.

- **Зеркальные 3D-очки:** На экране отображаются два неперекрывающихся изображения для левого и правого глаз. В очках установлены поворотные зеркала, которые направляют изображение с правой части экрана на правый глаз и с левой части на левый глаз. Эта технология подходит только для просмотра с неподвижной головой относительно экрана, так как повороты головы нарушают создаваемую очками стереопару.
- **Активные очки (Active shutter 3D system):** Под воздействием управляющего сигнала очки попеременно закрывают левый и правый глаз. В момент закрытия одного из глаз на экране воспроизводится изображение, предназначенное для этого глаза. Таким образом, на разные глаза подаются разные изображения, образующие стереопару.

Дальнейшее развитие технологии представляет собой носимые головные дисплеи (Head Mounted Display, HMD), также известные как шлемы виртуальной реальности. Концепция HMD была предложена в 1960-е годы XX века. В 1968 году профессор Массачусетского технологического института А. Сазерленд и его ученик Б. Спроул создали первый рабочий прототип шлема виртуальной реальности [54]. В то время шлем был настолько тяжелым, что его приходилось подвешивать к потолку. В настоящее время существует множество моделей шлемов виртуальной реальности, но основные принципы их работы остаются неизменными.

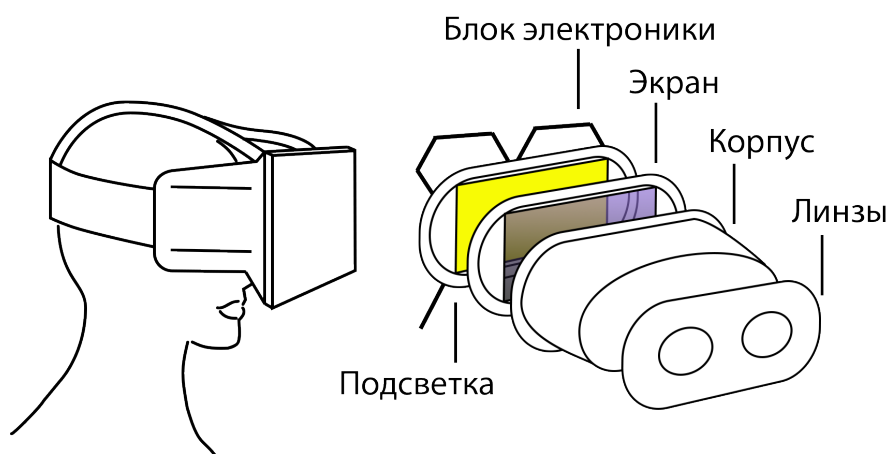


Рисунок 2.3 — Структура шлема виртуальной реальности

Современный шлем виртуальной реальности (рис. 2.3) обычно состоит из следующих компонентов [61]:

- Дисплей или другой источник изображения (например, жидкокристаллический дисплей);
- Оптическая система, состоящая из линз для каждого глаза, имитирующая бесконечное расстояние до экрана и создающая стереоэффект;
- Система отслеживания положения головы пользователя, необходимая для корректировки изображения в соответствии с положением и ориентацией головы относительно фиксированного пространства;
- Крепления, фиксирующие все компоненты устройства на голове пользователя.

Стереопара и отображение сцены зависят от конкретной точки пространства, в которой находится пользователь. Для достоверной визуальной имитации в виртуальной среде важно обновлять изображения, предоставля-

емые стереоэкранам, с учётом перемещений пользователя. Вследствие этого возникает задача отслеживания физического перемещения пользователя относительно неподвижного окружающего пространства. Такая задача также возникает при работе с неподвижным экраном, относительно которого человек может перемещаться (самостоятельно или на подвижной платформе).

## 2.2 Задача построения визуализации

### 2.2.1 Математический аппарат задания преобразований пространства

Аналогично предыдущей главе, будем использовать углы Крылова для определения поворотов пространства. Произвольное угловое перемещение твердого тела относительно исходной системы координат описывается углами рысканья  $\gamma$ , тангажа  $\beta$  и крена  $\alpha$ .

Для решения задачи визуализации, помимо описания поворотов связанной системы координат, необходимо также учитывать смещения начала координат этой системы, изменение масштаба осей и их растяжение.

Обобщенная матрица  $T$  преобразования размером  $4 \times 4$  для трехмерных координат имеет следующий вид:

$$T = \left[ \begin{array}{ccc|c} & & & x_T \\ & R & & y_T \\ & & & z_T \\ \hline a & b & c & S \end{array} \right]. \quad (2.1)$$

Эта матрица состоит из четырех отдельных частей:

- Матрица  $R^{3 \times 3}$  осуществляет линейное преобразование в виде изменения масштаба, сдвига и вращения.
- Матрица  $(x_T, y_T, z_T)^{3 \times 1}$  производит перенос.
- Матрица  $(a, b, c)^{1 \times 3}$  — преобразования в перспективе.
- Скалярный элемент  $S^{1 \times 1}$  выполняет общее изменение масштаба.

Рассмотрим более подробно, как выглядят простые операции в терминах обобщенной матрицы.

Трехмерный перенос:

$$\begin{bmatrix} 1 & 0 & 0 & x_T \\ 0 & 1 & 0 & y_T \\ 0 & 0 & 1 & z_T \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} x + x_T \\ y + y_T \\ z + z_T \\ 1 \end{bmatrix}. \quad (2.2)$$

Трехмерное изменение масштаба ( $r_x \neq r_y \neq r_z$ ):

$$\begin{bmatrix} r_x & 0 & 0 & 0 \\ 0 & r_y & 0 & 0 \\ 0 & 0 & r_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} r_x \cdot x \\ r_y \cdot y \\ r_z \cdot z \\ 1 \end{bmatrix}. \quad (2.3)$$

Если  $r_x = r_y = r_z = s$ , то общее изменение масштаба можно выполнить за счет четвертого диагонального элемента:

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & s \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} x \\ y \\ z \\ s \end{bmatrix} = \begin{bmatrix} \frac{x}{s} \\ \frac{y}{s} \\ \frac{z}{s} \\ 1 \end{bmatrix}. \quad (2.4)$$

Трехмерное искажение (дисторсия) осуществляется за счет недиагональных элементов матрицы  $T_{4 \times 4}$ :

$$\begin{bmatrix} 1 & h_1 & h_2 & 0 \\ h_3 & 1 & h_4 & 0 \\ h_5 & h_6 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} x + h_1y + h_2z \\ y + h_3x + h_4z \\ z + h_5x + h_6y \\ 1 \end{bmatrix}. \quad (2.5)$$

Матрицы перехода, задающие вращение относительно трех осей координат имеют вид:

$$T_x(\alpha) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \alpha & \sin \alpha & 0 \\ 0 & -\sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (2.6)$$

$$T_y(\beta) = \begin{bmatrix} \cos \beta & 0 & -\sin \beta & 0 \\ 0 & 1 & 0 & 0 \\ \sin \beta & 0 & \cos \beta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (2.7)$$

$$T_z(\gamma) = \begin{bmatrix} \cos \gamma & \sin \gamma & 0 & 0 \\ -\sin \gamma & \cos \gamma & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (2.8)$$

В данном разделе представлены преобразования, которые помещают объекты из виртуального мира (сцены) на виртуальный прямоугольный экран, в зависимости от конкретной точки зрения виртуального глаза (также называемого "виртуальной камерой"). Выбор плоского прямоугольного экрана обусловлен техническими и историческими причинами, хотя его форма не соответствует форме человеческой сетчатки.

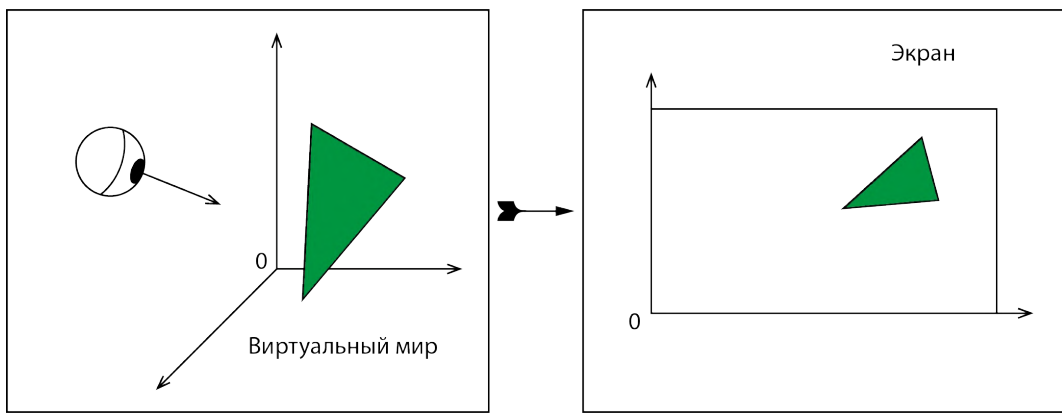


Рисунок 2.4 — Преобразование из виртуальной сцены в проекцию на экране

Геометрические модели в виртуальном пространстве представляют собой поверхности в  $\mathbb{R}^3$  и включают бесконечное количество точек. Однако, поскольку компьютерные представления должны быть конечными, модели определяются с использованием примитивов, каждый из которых представляет бесконечный набор точек. Наиболее простым и полезным примитивом является треугольник в трёхмерном пространстве, как показано на рисунке 2.4. Участок плоской поверхности, соответствующий всем точкам "внутри" и на границе треугольника, полностью определяется координатами вершин треугольника в трёхмерном пространстве.

Допустим, что модель движущегося объекта представлена в виде сетки из треугольников. Чтобы переместить объект, нужно применить одну и ту же трансформацию к каждой вершине каждого треугольника. Используя комбинацию сдвига и вращения, модель может быть размещена в любом месте и с любой ориентацией в виртуальном мире  $\mathbb{R}^3$ .

### 2.2.2 Визуализация сцены с учетом положения виртуальной камеры

В данном разделе рассматривается процесс преобразования моделей в виртуальном мире для их отображения на виртуальном экране, а также способы формирования изображений с использованием виртуальных камер (согласно формализму, представленному в [23]). Для достижения корректной работы виртуальной реальности, "камера" должна представлять один из двух виртуальных глаз человека, размещенных в виртуальной среде.

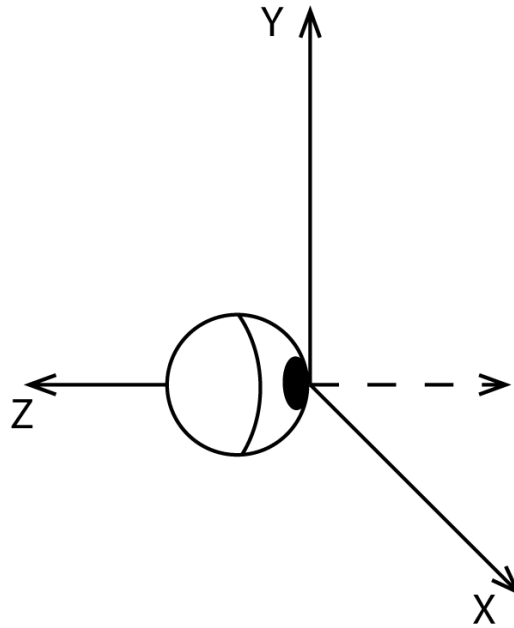


Рисунок 2.5 — Виртуальный глаз-"камера"

На рис. 2.5 изображен виртуальный глаз, направленный вдоль оси  $-z$  в связанной системе координат. Положение глаза таково, что ось  $x$  ориентирована вправо, а ось  $y$  — вверх. Предполагается, что глаз является объектом, который мы хотим разместить в виртуальном мире  $\mathbb{R}^3$  в определенной позиции  $e = (e_1, e_2, e_3)$  и ориентации, задаваемой матрицей  $R_{eye}$ .

Введем понятие виртуального экрана как объекта, находящегося на определенном расстоянии перед виртуальным глазом-камерой. Считаем расстояние известным и зависящим от конструкции системы визуализации. Виртуальный экран должен соответствовать реальному экрану по пропорциям и угловому размеру, однако может быть расположен ближе к виртуальному глазу, чем реальный экран к реальному глазу. Рассмотрим задачу отображения единичного объекта (твердого тела) на виртуальном экране (рис. 2.6). Финальная проекция будет выводиться на экран системы визуализации.

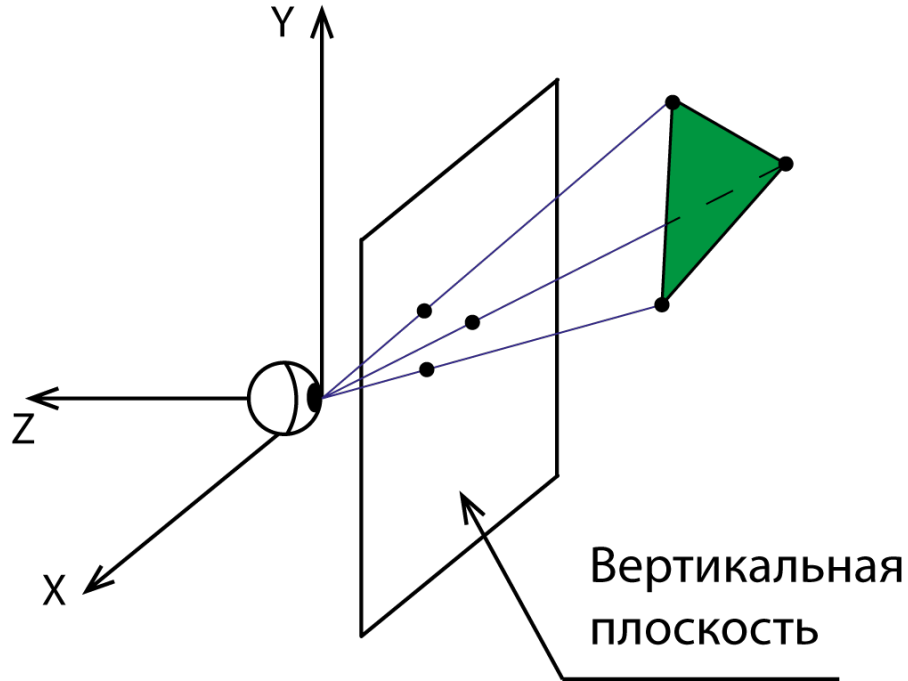


Рисунок 2.6 — Представление задачи визуализации как проекции изображения набора треугольников на плоскость экрана

Определим матрицу  $T_{rb}$  как преобразование, применяемое к точкам объекта (треугольник на рис. 2.4). Необходимо переместить все модели в виртуальном мире в систему отсчета виртуального глаза, представленную на рисунке 2.5. Это требует применения обратного преобразования  $-e$  и  $R_{eye}^\top$ . Обозначим его как  $T_{eye}$ :

$$T_{eye} = \left[ \begin{array}{ccc|c} R_{eye}^\top & 0 \\ \hline 0 & 0 & 0 & 1 \end{array} \right] \left[ \begin{array}{cccc} 1 & 0 & 0 & -e_1 \\ 0 & 1 & 0 & -e_2 \\ 0 & 0 & 1 & -e_3 \\ 0 & 0 & 0 & 1 \end{array} \right]. \quad (2.9)$$

Тогда преобразование, задающее положение объекта в системе координат глаза будет выглядеть как  $T_{eye}T_{rb}$ .

Далее необходимо определить центральную проекцию объекта на экран с учётом положения глаза. Важным требованием графического программного обеспечения является представление результата в безразмерной канонической форме для вывода на физический дисплей. Ещё одним требованием, связанным с ограниченными вычислительными ресурсами, является определение области, в которой возможна проекция на плоскость. Это ограничение определяется усечённым конусом, называемым фрустумом (рис. 2.7). Для этого вводятся плоскости на расстояниях  $z = n$  и  $z = f$ . Виртуальный экран

располагается на ближней плоскости, а центральная проекция должна поместить все точки внутри усечённой пирамиды на этом виртуальном экране.

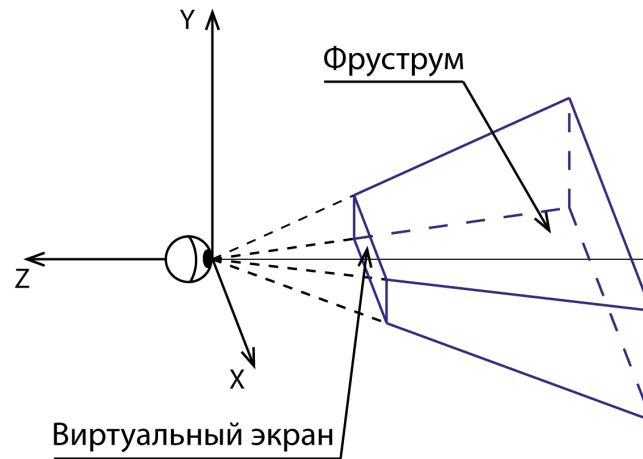


Рисунок 2.7 — Допустимая область визуализации относительно виртуального глаза

Преобразование осуществляется в два этапа. Сначала усечённая пирамида преобразуется в прямоугольник (рис. 2.8) с помощью преобразования  $T_p$ :

$$T_p = \begin{bmatrix} n & 0 & 0 & 0 \\ 0 & n & 0 & 0 \\ 0 & 0 & n & -fn \\ 0 & 0 & 1 & 0 \end{bmatrix}. \quad (2.10)$$

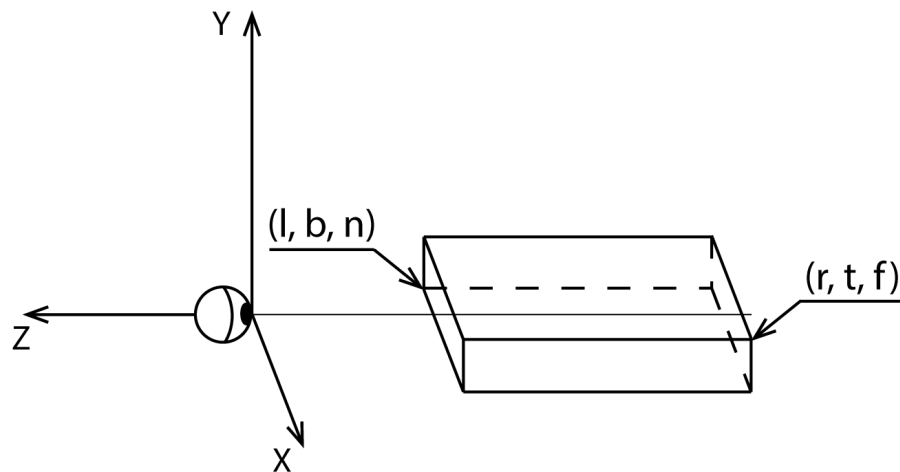


Рисунок 2.8 — Преобразованный фрустум относительно виртуального глаза

Затем прямоугольник преобразуется в единичный куб с вершинами, координаты которых равны  $(\pm 1, \pm 1, \pm 1)$ :



$$T_{st} = \begin{bmatrix} \frac{2}{r-l} & 0 & 0 & -\frac{r+l}{r-l} \\ 0 & \frac{2}{t-b} & 0 & -\frac{t+b}{t-b} \\ 0 & 0 & \frac{2}{n-f} & -\frac{n+f}{n-f} \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (2.11)$$

После применения двух предыдущих преобразований координаты изображения находятся в диапазоне от -1 до 1. Затем требуется преобразовать проецируемые точки к координатам, используемым для индексации пикселей на физическом дисплее. Пусть  $m$  — количество пикселей по горизонтали, а  $s$  — количество пикселей по вертикали. Например,  $s = 1080$  и  $m = 1920$  для дисплея FullHD. Дисплей проиндексирован строками от 0 до  $s - 1$  и столбцами от 0 до  $m - 1$ , причём точка  $(0, 0)$  находится в нижнем левом углу. В таком случае преобразование имеет следующий вид:

$$T_{vp} = \begin{bmatrix} \frac{m}{2} & 0 & 0 & \frac{m-1}{2} \\ 0 & \frac{s}{2} & 0 & \frac{s-1}{2} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (2.12)$$

Итоговая последовательность преобразований для визуализации виртуального объекта для конкретного глаза определяется следующим образом:

$$T_{viz} = T_{vp}T_{st}T_pT_{eye}T_{rb}. \quad (2.13)$$

Важно отметить, что для каждого объекта в сцене  $T_{rb}$  является уникальным. Для создания стереоэффекта данное преобразование необходимо вычислять для каждого из глаз человека.

### 2.3 Фовеальный рендеринг как метод повышения частоты кадров при визуализации

Современные системы виртуальной реальности оснащены экранами с высоким разрешением, что позволяет обеспечить четкое видение даже небольших виртуальных объектов. Человеческий глаз обладает наиболее четким зрением в центральной области сетчатки (в пределах 5 угловых градусов), где расположены светочувствительные рецепторы с наивысшей плотностью, называемые *фовеа*.

Одним из факторов, вызывающих укачивание при использовании систем виртуальной реальности, является снижение частоты кадров при воспроизведении плохо оптимизированных виртуальных сцен. Для корректного визуального восприятия движения, преобразование (2.13) должно выполняться с частотой не менее 60 кадров в секунду [56]. Возможным решением для повышения частоты кадров является использование биологических особенностей человеческого зрения, а именно применение метода "фовеального рендеринга". Фовеальный рендеринг предполагает отображение сцены в высоком разрешении только в области, попадающей в центральное поле зрения пользователя. Часть сцены, на которую смотрит пользователь, определяется с помощью окулографа [22] (системы отслеживания направления взгляда). Особенно актуальным становится применение данной методики в авиасимуляторах в случае имитации посадки, полета над жилыми кварталами и в прочих высоконагруженных сценариях.

В последнее время были представлены новые исследования, посвященные фовеальному рендерингу и его различным реализациям [7; 18]. Однако, одной из проблем такого подхода является мерцание периферического изображения из-за низкого разрешения и эффекта алиасинга (наложения частот, вызывающего на изображении характерную "лестницу" из пикселей).

В рамках данной работы была предложена модификация стандартного алгоритма фовеального рендеринга, которая позволяет уменьшить эффект мерцания периферического изображения. В качестве решения применялось искусственное масштабирование изображения с использованием Nvidia Deep Learning Super Sampling (DLSS).

В отличие от традиционных подходов, которые вызывают размытие сцены за пределами центрального поля [28], DLSS был добавлен для обеспечения приемлемого качества изображения даже при возможных ошибках в отслеживании движения глаз, а также для уменьшения мерцания пикселей. Эта идея схожа с концепцией DeepFovea [19], однако использует широко распространенное решение от Nvidia [62].

### **2.3.1 Принцип работы окулографической системы**

Зрительное восприятие человека основано на последовательной фиксации взгляда на постоянно изменяющихся отдельных точках. Переход взгляда с одной точки на другую происходит при помощи саккады — быстрого ко-

ординированного движения глаз [117]. Типичная продолжительность такого движения составляет от 0,01 до 0,1 секунды, а угловая скорость вращения глаза может достигать  $500^\circ/\text{с}$ . Для того чтобы эффективно визуализировать изображение, требуется определить точку остановки взора в середине саккадического движения. Исходя из этого, желаемая частота отслеживания движений глаз должна быть равна 250 Гц или более [22].

Один из распространенных подходов к отслеживанию движений глаз и окулографии заключается в использовании разнесенных по пространству фотодатчиков и системы обработки изображений. Проецируемое изображение с фотодатчика передается на внешний компьютер для дальнейшей обработки и определения положения объекта. Однако, подобное решение имеет ряд недостатков, таких как большой объем данных для передачи и обработки, низкая скорость и высокое энергопотребление.

Для решения этих проблем была предложена альтернативная схема — видеосистема на кристалле [10]. Данная система разработана для упрощенного поиска темных или светлых объектов на однородном фоне с частотой обработки до 1000 Гц. Она позволяет определить координаты, площадь и яркость точки, а также может хранить в буфере до 50 объектов на каждом кадре с указанной частотой. Полученная информация позволяет найти зрачок (темный объект на светлом фоне) и определить направление взгляда человека. Система работает в широком спектральном диапазоне, что позволяет использовать разнообразные конструкции для отслеживания взгляда. В результате, устройство обладает компактными размерами, небольшим весом, высокой точностью и низким энергопотреблением, что делает его подходящим для носимых систем и виртуальной реальности.

По сравнению с другими окулографами, данный подход сочетает в себе высокую точность и скорость работы, которые обычно недоступны в портативных системах. Камеры размещены за линзами (рисунок 2.9), что обеспечивает комфорт при использовании. Гарнитура виртуальной реальности с окулографом отличается от гарнитуры без окулографа только наличием инфракрасной подсветки вокруг линз. Благодаря возможностям системы на кристалле, расчеты могут быть проведены даже на слабых портативных компьютерах, что значительно упрощает интеграцию данной технологии в существующие комплексы.

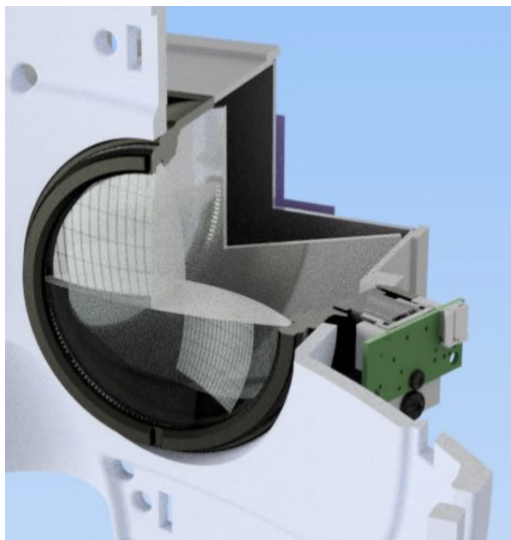


Рисунок 2.9 — Линзовая схема шлема виртуальной реальности с окулографом

### 2.3.2 Реализация фовеального рендеринга

Алгоритм фовеального рендеринга был успешно интегрирован в компактную систему виртуальной реальности, разработанную в МГУ. Тестовая виртуальная сцена была создана на основе Unreal Engine 4.25.1 (и впоследствии конвертирована для версии Unreal Engine 4.26). Общее количество треугольников сцены, с учетом применения уровней детализации для объектов, составило 21,8 миллиона. Сцена содержала следующие элементы:

- трехмерную модель ландшафта в виде полигонального меша,
- копию ландшафта с отключенным отображением поверхности, реализованную с помощью встроенных методов Unreal Engine,
- элементы растительности, размещенные на поверхности "невидимого" ландшафта,
- трехмерные модели фоновой застройки,
- трехмерные модели ряда высокополигональных объектов.

Изображение формировалось с использованием ранее описанных преобразований и отображалось на гарнитуре виртуальной реальности Total Vision VR4 с частотой обновления дисплея 60 Гц и суммарным разрешением для обоих глаз 2880x1440 пикселей. Сцена тестировалась на ноутбуке, оснащенном процессором Intel Core i9-9880H 2,3 ГГц, 32 ГБ оперативной памяти и видеоадаптером Nvidia GeForce RTX 2080.

Для оценки производительности алгоритма было проведено две серии сравнительных тестов, состоящих из 5 и 12 экспериментов соответственно.

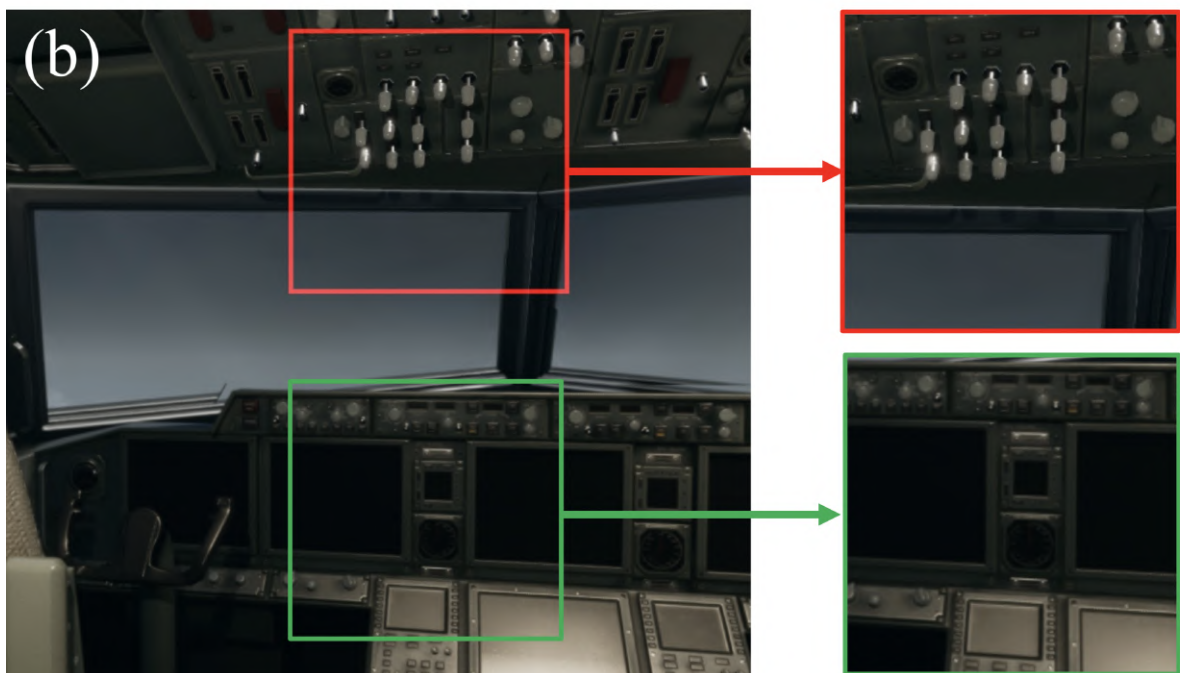
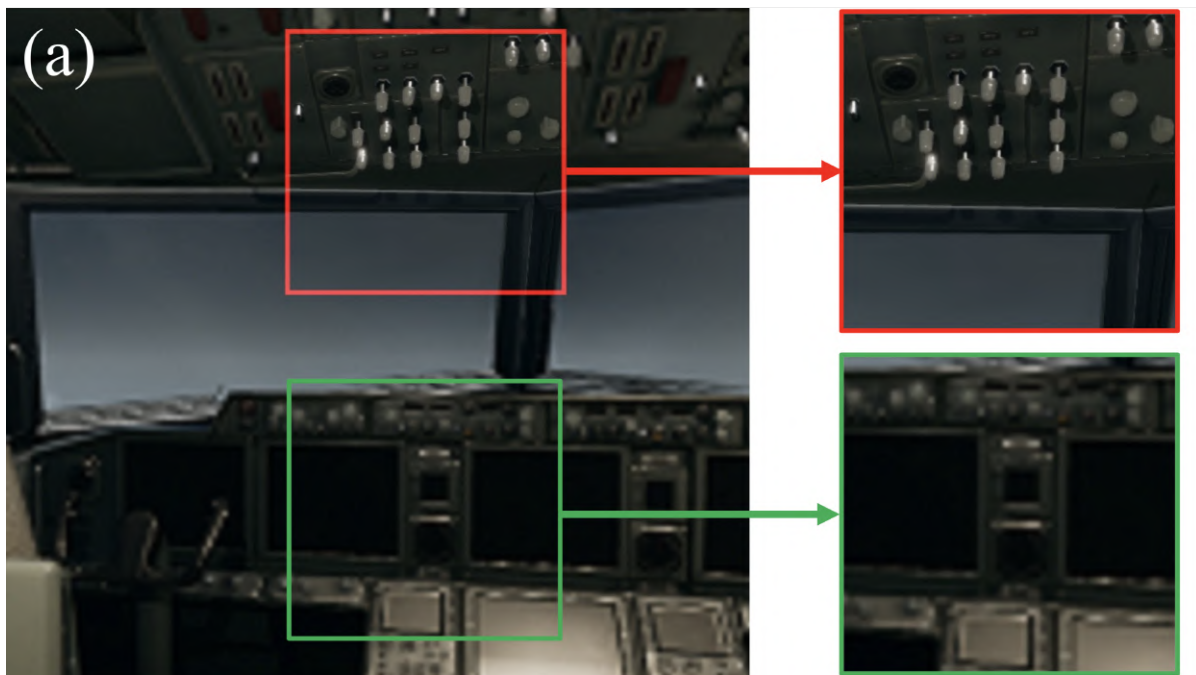


Рисунок 2.10 — Реконструкция фовеального дисплея: типовой фовеальный рендеринг (а) и предлагаемый подход (б). Красный квадрат показывает зону фовеа, а зеленый квадрат — периферическую зону

Каждый тест включал 20 минут пребывания пользователя в виртуальной среде. Во время тестов проводилось сравнение между рендерингом в полном разрешении и модифицированным фовеальным режимом рендеринга. В режиме полного разрешения сцена отображалась с разрешением 2880x1440 пикселей. В режиме фовеального рендеринга базовое разрешение составля-

ло 960x480 пикселей и было увеличено с использованием технологии Nvidia Deep Learning Super Sampling (DLSS). Рендеринг с полным разрешением применялся в области, ограниченной 10 угловыми градусами (рисунок 2.10).

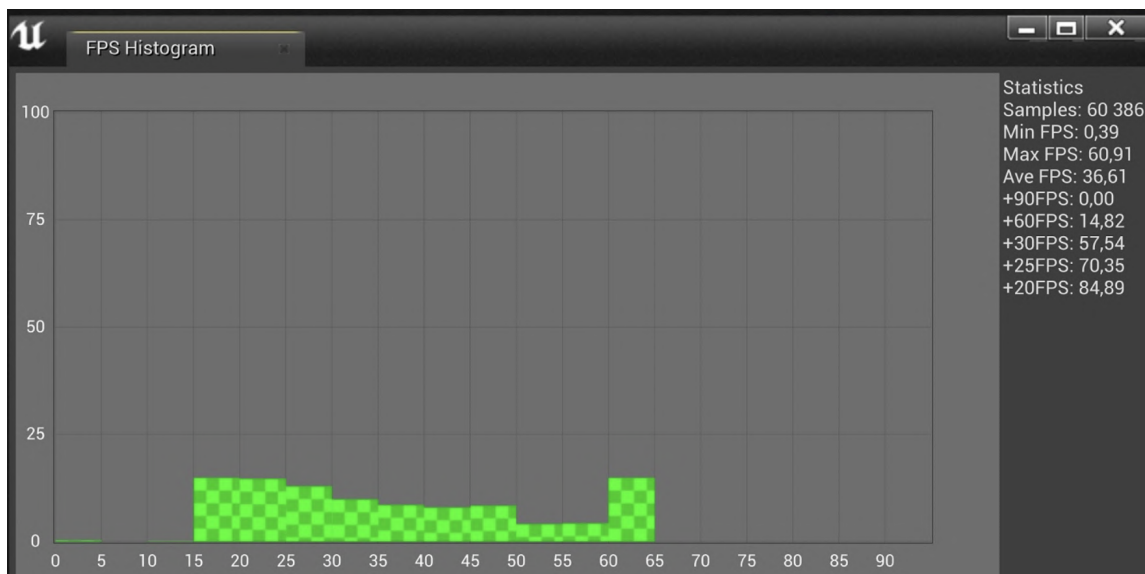


Рисунок 2.11 — Анализ частоты кадров для сцены с полным разрешением

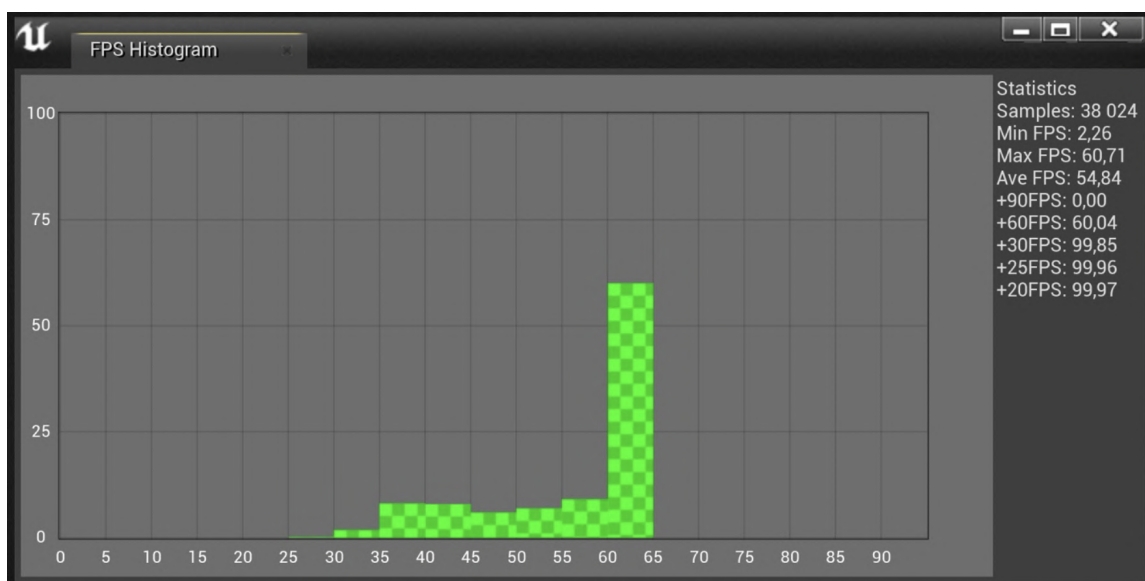


Рисунок 2.12 — Анализ частоты кадров для фовеального рендеринга сцены

В ходе проведенных экспериментов было обнаружено, что максимальное количество кадров в секунду (FPS) для обоих режимов достигает одинаковых значений, вплоть до 60 Гц в определенные моменты исследования сцены пользователем. Тем не менее, средние значения FPS демонстрируют существенную разницу:

- 36.61 FPS для сцены в полном разрешении (рисунок 2.11),
- 54.84 FPS для сцены с применением фовеального рендеринга (рисунок 2.12).

Полученные результаты свидетельствуют о приросте производительности на 50% при использовании фовеального рендеринга по сравнению с полным разрешением. Кроме того, существенно возрастает доля времени, когда достигается комфортное для человеческого восприятия значение частоты кадров в 60 Гц. Сравнение стандартного и модифицированного фовеального рендеринга не выявило статистически значимых различий в производительности. Однако это приводит к заметному субъективному улучшению комфорта восприятия периферийного изображения и снижению утомляемости пользователя.

В целом, интеграция алгоритма фовеального рендеринга в компактную систему виртуальной реальности позволила достичь высокой производительности и качества изображения, что является критически важным для комфортного и реалистичного взаимодействия пользователя с виртуальной средой. Использование технологии Nvidia DLSS для масштабирования изображения обеспечивает качественное отображение даже при возможных ошибках в отслеживании движения глаз, уменьшая мерцание пикселей в периферической зоне и улучшая общий пользовательский опыт.

### Глава 3. Задача синхронизации визуальной и динамической составляющих имитации

<sup>1</sup>Реалистичность имитационных сред виртуальной реальности определяется степенью соответствия визуальных образов и реальных движений пользователя. В данной работе рассматривается подвижный механический стенд и система визуализации, которые должны в совокупности обеспечивать имитацию управляемого объекта, например, самолета.

Предположим, что математическая модель, используемая для имитации движения самолета [72], вычисляется в реальном времени в определенном виртуальном пространстве. Рассмотрим пространство стенда как реальную комнату, в которой находятся пользователь и подвижный стенд-тренажер.

Аналогично главе 1, в реальном пространстве определим несколько систем координат: основную неподвижную систему, связанную с Землей и определенной точкой комнаты, подвижную систему, привязанную к стенду, и систему, связанную с головой пилота. В виртуальном пространстве, по аналогии с главой 2, также введем три системы координат: привязанную к Земле, креслу виртуального самолета и виртуальной камере-голове пилота.

Человеческий организм способен определить свое перемещение относительно системы отсчета, связанной с Землей. Вестибулярный аппарат человека, описанный в главе 1, через короткую трехнейронную цепочку соединен с глазодвигательными мышцами. Реакция глаз на движение головы происходит рефлекторно, обходя высшие структуры центральной нервной системы. Компенсаторное движение глаза относительно движения головы называется *вестибулоокулярным рефлексом* (ВОР).

Этот рефлекс играет ключевую роль в стабилизации взгляда относительно неподвижной системы отсчета во время движения человека. Благодаря этому рефлексу, при фиксации взгляда на точке и пассивных поворотах головы, сохраняется четкость видения. Нарушение механизма вестибулоокулярного рефлекса может привести к возникновению болезни движения, так-

---

<sup>1</sup>При подготовке данного раздела диссертации использованы следующие публикации, выполненные автором лично или в соавторстве, в которых, согласно Положению о присуждении ученых степеней в МГУ, отражены основные результаты, положения и выводы исследования: [9–11;81;91]



же известной как тренажерная болезнь, симптомом которой является укачивание.

Человек обладает способностью определения положения своего тела относительно Земли (условно неподвижной системы координат), поэтому наиболее точная имитация движения возможна с использованием модели, построенной относительно неподвижной системы координат.

При создании визуализации, синхронизация с динамической имитацией должна происходить относительно неподвижной системы координат. Возможно, возникнет предположение, что система визуализации должна быть жестко связана с комнатой. Однако в мировой практике существуют различные варианты расположения систем визуализации, применяемые в тренажерных комплексах:

1. Жесткая механическая связь с неподвижной комнатой.
2. Жесткая механическая связь со стендом (креслом пилота).
3. Жесткая механическая связь с головой пилота (шлем виртуальной реальности).

Выбор конкретного варианта обусловлен особенностями технической реализации. Вестибулярный аппарат ориентируется на неподвижную систему координат, и для создания корректной визуализации (как виртуального окружения, так и кокпита самолета) необходимо программно учитывать перемещения всех остальных систем.

”Идеальным” случаем можно считать полную согласованность перемещений связанных со стендом и пилотом систем координат и их виртуальных аналогов. Однако, из-за известных ограничений, это практически невозможно достичь. В условиях виртуальной реальности, информация от вестибулярного сенсора не всегда совпадает с представленным изображением из-за геометрических ошибок позиционирования виртуальной камеры, ограничений перемещений платформы и наличия задержек в формировании изображения.

Можно рассмотреть ”вырожденный” случай: когда система визуализации жестко связана с креслом пилота, при этом используется коллиматорный проекционный дисплей, имитирующий бесконечное удаление изображения.

Предполагается, что пилот пристегнут к креслу [72]. Такие предположения позволяют исключить малые смещения головы и построить визуализацию в форме, представленной в главе 2 (выражение (2.13)):

$$T_{viz} = T_{vp}T_{st}T_pT_{eye}T_{rb}.$$

Следует отметить, что в данном случае операторы преобразования визуализации  $T_{vp}$  (2.12),  $T_{st}$  (2.11),  $T_p$  (2.10) зависят только от параметров системы визуализации и рассчитываются заранее. В то время как оператор  $T_{rb}$ , определяющий перемещение виртуальных объектов относительно виртуальной начальной точки, вычисляется на каждом этапе симуляции. Оператор, задающий положение виртуальной камеры-”глаза”  $T_{eye}$  (2.9), заслуживает особого внимания. В классических тренажерных системах экранного типа он вычисляется однократно в зависимости от конфигурации экрана и его размещения на подвижной платформе, а  $T_{rb}(t)$  задается сразу в системе координат, связанной с виртуальной камерой.

Во всех рассмотренных сценариях без использования коллиматорного экрана, величина  $T_{eye}(t)$  становится зависимой от динамики движения платформы и головы пилота относительно начальной точки отсчета.

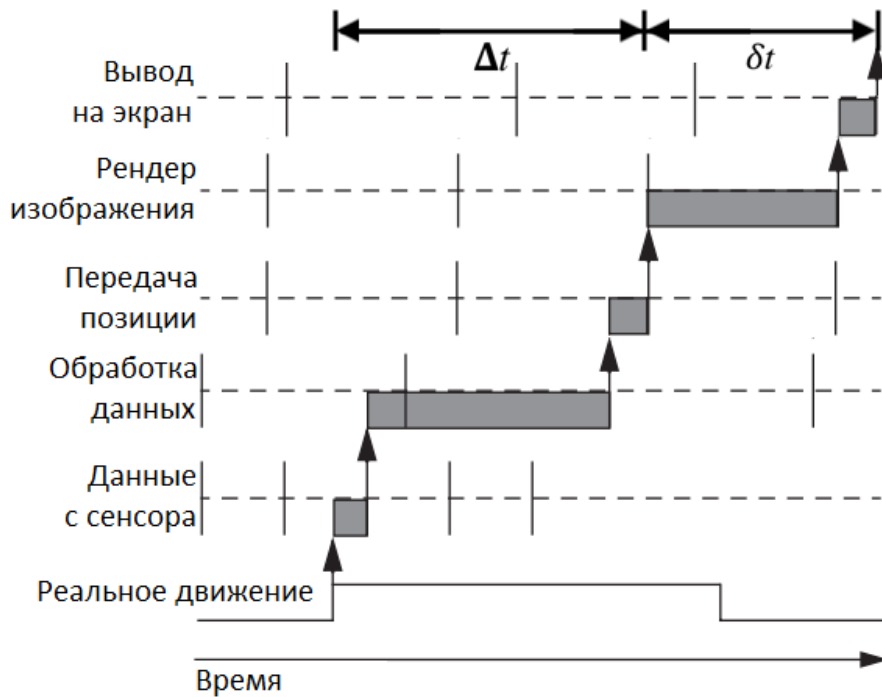


Рисунок 3.1 — Структура задержек в системе отслеживания движений и системе визуализации

Таким образом, возникает потребность в применении специализированных систем, которые могут отслеживать движения пользователя и платформы-стенда. Видеоанализ [50] является распространенным методом для

этой цели, хотя также были предложены гибридные методы отслеживания движений. Эксперименты, проведенные на панорамной системе виртуальной реальности МГУ, показали, что даже при использовании современных систем видеоанализа задержка между реальным движением и его воспроизведением в виртуальном пространстве может достигать 0.2 секунды. Задержка обусловлена временем на получение и обработку данных с системы видеоанализа, их передачи на графическую станцию, формирования и вывода на экран итогового изображения (см. рисунок 3.1).

В дальнейшем, без потери общности, будем рассматривать случай, когда система визуализации фиксирована неподвижно относительно комнаты. При этом, для формирования изображения будем учитывать перемещения как самой платформы, так и головы человека.

### 3.1 Постановка задачи синхронизации визуальной и динамической составляющих имитации

Рассмотрим метод получения преобразования  $T_{eye}(t)$  (2.9). Предположим, что имеется виртуальная среда с имитацией системы (*виртуальный летательный аппарат*). В данном контексте, в отличие от представленного в первой главе, предположим, что перемещение летательного аппарата моделируется с использованием графического программного обеспечения.

Для анализа данной системы введем несколько систем координат (рисунок 3.2). Обозначим систему координат, связанную с неподвижным виртуальным пространством, как  $O\xi^p\eta^p\zeta^p$ . Систему координат, ассоциированную с виртуальным креслом пилота, обозначим как  $M_P x^{PM} y^{PM} z^{PM}$ . Систему координат, связанную с системой виртуальных камер, обозначим как  $O_{Ph} x^{Ph} y^{Ph} z^{Ph}$ .

Кроме того, введем системы координат, которые соответствуют реальному пространству, где находится подвижный стенд. Обозначим систему координат, связанную с комнатой, как  $O_{init} \xi \eta \zeta$ . Систему координат, ассоциированную с креслом пилота на подвижном стенде, обозначим как  $M x^M y^M z^M$ . Наконец, систему координат, связанную с головой человека (при условии, что центр расположен между глазами), обозначим как  $O_h x^h y^h z^h$ .

Программная реализация модели летательного аппарата функционирует как "черный ящик", предоставляя доступ к выходу модели в виде перегрузки  $n^p(t)$ , угловой скорости  $\omega^p(t)$  и преобразования  $T_x^p(t)$  (рисунок 3.3),

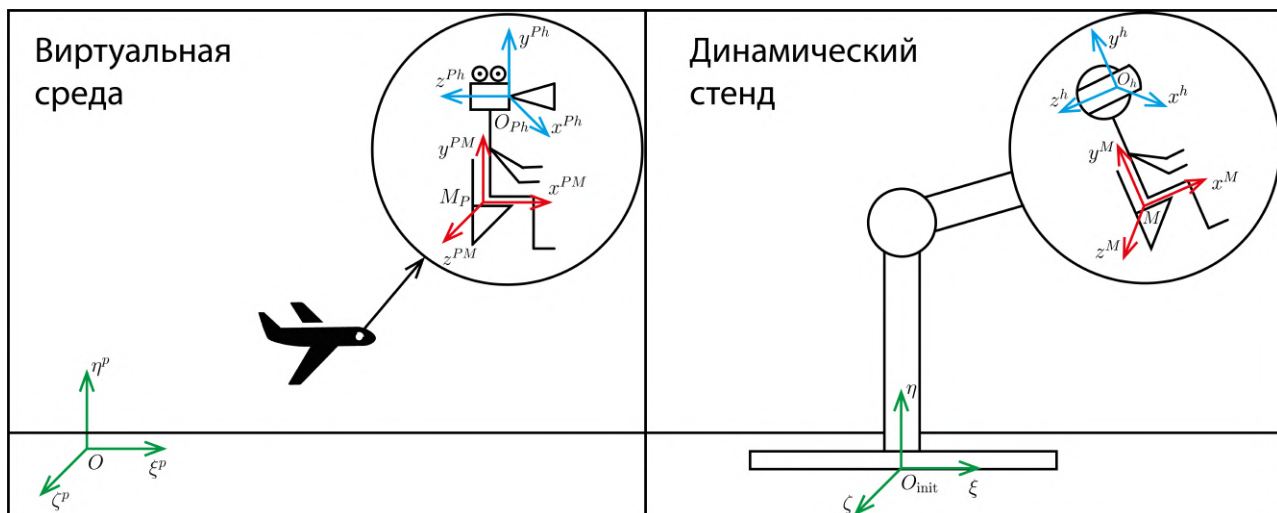


Рисунок 3.2 — Виртуальная среда, реальное пространство и системы координат

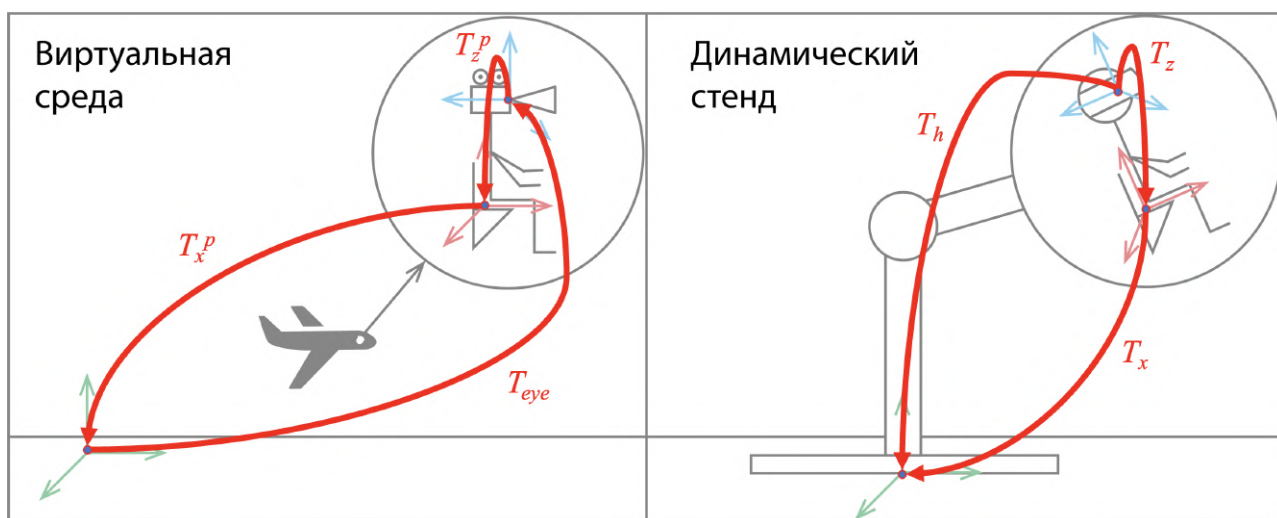


Рисунок 3.3 — Преобразования систем координат

определяющего положение кресла пилота в виртуальной среде. Введем оператор  $T_z^p(t)$ , задающий положение виртуальной "головы" в системе координат, связанной с виртуальным креслом пилота. Будем считать, что каждая из двух виртуальных "камер", соответствующих глазам пилота, задается сдвигом  $T_{shift}$  относительно головы (величина 0,032 соответствует половине среднестатистического межглазного расстояния для человека):

$$T_{shift} = \begin{bmatrix} 1 & 0 & 0 & \pm 0,032 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Тогда искомое преобразование может быть выражено следующим образом:

$$T_{eye}(t) = T_{shift}(T_x^p(t)T_z^p(t))^{-1}. \quad (3.1)$$

Модифицированное итоговое преобразование (2.13), решающее задачу визуализации, примет вид:

$$T_{viz}(t) = T_{vp}T_{st}T_pT_{shift}(T_x^p(t)T_z^p(t))^{-1}T_{rb}(t). \quad (3.2)$$

Поскольку  $T_x^p(t)$  известно из компьютерной модели, основная задача заключается в определении виртуального преобразования  $T_z^p(t)$ , согласующего виртуальное с реальным движением. Отметим, что для решения данной задачи нет необходимости синхронизировать визуальные и динамические воздействия с точностью, превышающей пороги чувствительности отолитовых органов и полукружных каналов [72].

Пороги чувствительности полукружных каналов и отолитовых органов уникальны для каждого человека и могут значительно различаться. В контексте создания сложных систем виртуальной реальности, где высокая точность является приоритетом, можно было бы адаптировать имитацию к индивидуальным особенностям каждого пользователя. Однако, для пилотажных стендов, где универсальность важнее учета физиологических особенностей каждого человека, можно использовать усредненные пороговые значения.

Согласно литературным данным, абсолютный порог чувствительности полукружных каналов к угловому ускорению варьируется в пределах от  $0,035^\circ/c^2$  до  $8,2^\circ/c^2$ . Среднестатистическое значение порога часто указывается в диапазоне  $\sim 1 - 3^\circ/c^2$  [101;104]. В качестве порога чувствительности полукружных каналов к угловой скорости приводят значения  $\sim 2^\circ/c$  [99].

Что касается порога чувствительности отолитовых органов, он также значительно различается в разных источниках: от  $2\text{см}/c^2$  [104] до  $20\text{см}/c^2$  [99]. Важно отметить, что по разным направлениям порог чувствительности может отличаться.

Рассмотрим схему взаимосвязи различных видов имитации и других блоков имитационного стенда (рисунок 3.4).

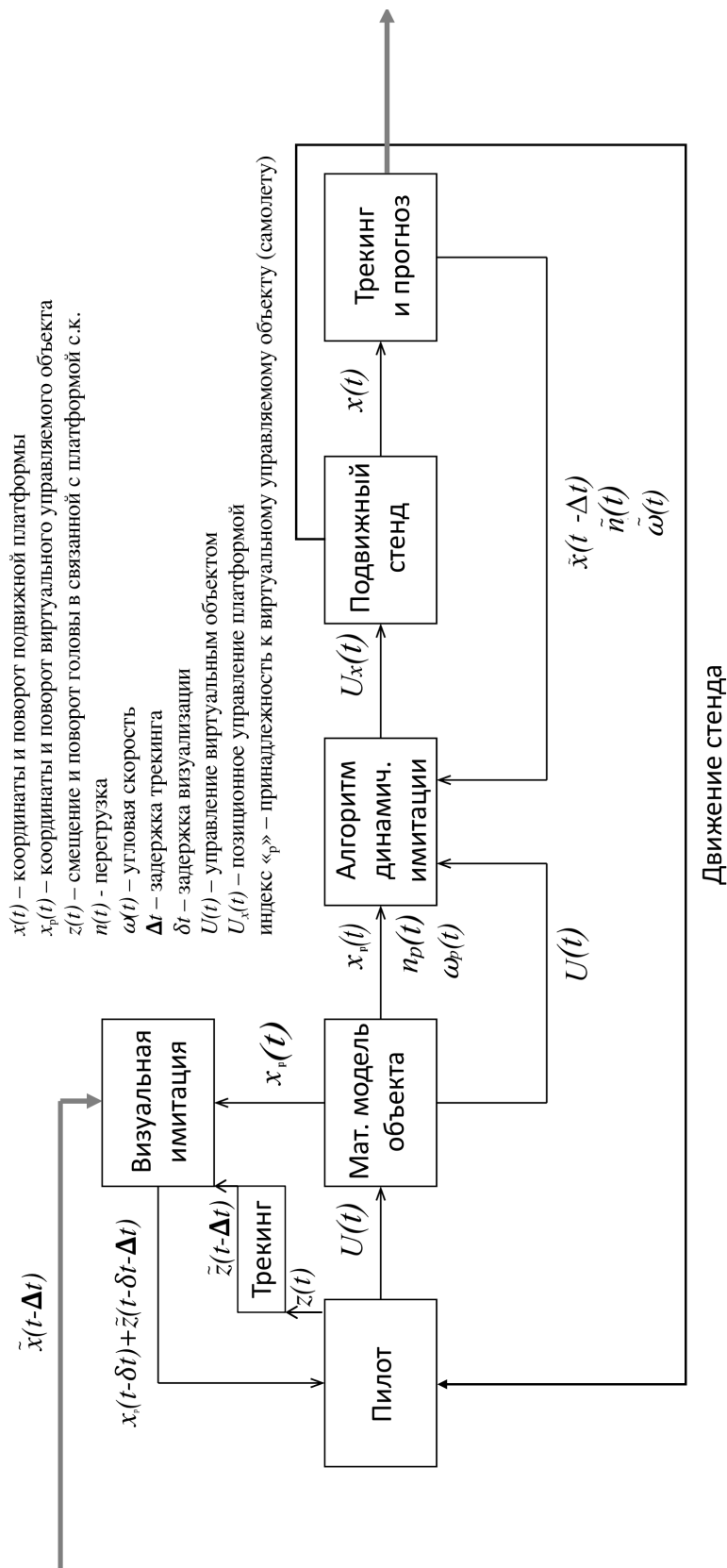


Рисунок 3.4 — Схема имитационного стенда

Рассмотрим преобразования в *реальном* пространстве (пространстве стенда). Пусть из показаний системы отслеживания движений в момент времени  $t$  можно получить  $\tilde{T}_x(t) = T_x(t - \Delta t) + \xi_x(t)$  — оценку преобразования между неподвижной системой координат комнаты и системой координат, связанной с креслом пилота в симуляторе (рисунок 3.3), где  $\Delta t$  — задержка системы трекинга (рисунок 3.1),  $\xi_x(t)$  — ошибка оценки.

Допустим, существует неизвестная динамика головы человека относительно кресла, а движение в подвижной системе координат обозначается как  $T_z(t)$  (рисунок 3.3). Предполагается, что система трекинга позволяет в момент времени  $t$  получить оценку положения головы относительно неподвижной системы координат, которую можно обозначить как  $\tilde{T}_h(t) := T_x(t - \Delta t)T_z(t - \Delta t) + \xi_h(t)$ , где  $\xi_h(t)$  — ошибка оценки.

В реальной имитационной системе процессы визуализации и динамической имитации осуществляются на дискретных временных интервалах. Будем называть этот временной интервал "шагом имитаций".

**Определение.** *Визуальная и динамическая составляющие имитации будут считаться **синхронными** (согласованными), если выполняются следующие условия:*

1. *Визуальная имитация реализуется с матрицей преобразования  $T_{eye}(t)$  (согласно представлению (3.2)), таким образом, чтобы для каждого шага имитации  $t_j$  выполнялось:*

$$\sum_{i=1}^3 [\omega_i^{Phead}(t_j) - \omega_i^{head}(t_j)]_{\mu_i+} \rightarrow \min_{T_{eye}}. \quad (3.3)$$

Здесь  $\omega_i^{Phead}$  обозначает угловую скорость виртуального "глаза"-камеры в виртуальном пространстве,  $\omega_i^{head}$  представляет угловую скорость головы человека в реальности (на подвижном стенде),  $\mu = (\mu_1, \mu_2, \mu_3)^\top$  — вектор порогов чувствительности полукружных каналов вестибулярного аппарата по угловой скорости,  $[\cdot]_{\mu_i+}$  — оператор:

$$[x]_{\mu_i+} = \begin{cases} x, & \text{если } |x| \geq \mu_i, \\ 0, & \text{если } |x| < \mu_i. \end{cases} \quad (3.4)$$

2. *Динамическая имитация выполняется с точностью, соответствующей вектору известных порогов чувствительности отолитовых органов вестибулярного аппарата  $\epsilon = (\epsilon_1, \epsilon_2, \epsilon_3)^\top$ .*

В наиболее простом случае, когда отсутствуют задержки ( $\Delta t \equiv 0$ ) и ошибки трекинга ( $\xi_x(t) \equiv \xi_h(t) \equiv 0$ ), задача поиска преобразования между виртуальным креслом и виртуальной камерой становится тривиальной. Однако, поскольку в реальной системе присутствуют задержки, необходимо использовать прогнозирование динамики стенда для создания визуализации с упреждением.

Кроме того, для реализации синхронизированных визуальной и динамической составляющих имитации необходимо найти такое управление стендом, чтобы динамическая имитация обеспечивалась с точностью, соответствующей вектору порогов чувствительности отолитовых органов  $\epsilon$ . Формально задачу динамической имитации быстрых линейных движений (1.11), приведенную в главе 1, можно переформулировать следующим образом:

$$J_w = \sum_{i=1}^3 \lambda_i [w_i(t_j) - w_i^p(t_j)]_{\epsilon_i+} \rightarrow \min_{u \in U}. \quad (3.5)$$

Здесь  $\lambda_i$  — весовые коэффициенты,  $w^p(t_j) \in \mathbb{R}^3$  — желаемое ускорение, получаемое из модели самолета, а  $w(t_j) \in \mathbb{R}^3$  — фактически реализуемое на стенде ускорение (и то, и другое можно получить из перегрузок  $n^p$  и  $n$  на  $j$ -м шаге имитации соответственно). Применение указанной нормы способствует повышению чувствительности алгоритма имитации при малых значениях отклонения от эталона по сравнению, например, с евклидовой и функционалами более высокой степени.

Задача имитации ориентации вектора перегрузки и быстрых угловых движений остается неизменной. В дальнейшем, будем считать, что на  $j$ -м шаге имитации сначала вычисляется управление стендом  $u(t_j)$ , а потом решается задача визуализации.

Поскольку из данных системы трекинга нам доступна оценка  $\tilde{T}_z(t)$ , отражающая реальное движение с некоторой неточностью и задержкой  $\Delta t$ , будем прогнозировать динамику стенда на величину этой задержки. Для снижения неопределенности, связанной с неточным знанием геометрических параметров системы, применим алгоритм предварительной идентификации этих величин. Идентификацию массовых параметров системы проводить не будем, а вместо этого применим робастный алгоритм управления, снижающий влияние неопределенностей и внешних возмущений. В конце, с целью повышения точности прогноза, будем использовать методы сокращения величины задержки визуализации.



Таким образом, решение задачи синхронизации включает:

1. Предварительной идентификации геометрических параметров подвижного стенда.
2. Отслеживания угловых движений как подвижного стенда, так и головы человека с целью получения преобразования  $T_{eye}(t)$ .
3. Построения алгоритма управления стендом с учетом порогов чувствительности вестибулярного аппарата.
4. Прогноза углового движения подвижного стенда с учетом его динамики и времени задержки системы отслеживания движений.
5. Визуализации виртуального окружения с частотой кадров выше физиологически обусловленной пороговой величины (60 Гц).

Важно отметить, что согласованная имитация возможна только при нахождении кабины стенда внутри определенной области. Поиск этой области также является актуальной задачей и обычно осуществляется совместно с решением первой задачи. В главе 1 была рассмотрена первая задача из списка выше, а в главе 2 — четвертая задача. Далее в работе будут представлены методы решения остальных задач, не напрямую связанных между собой, но необходимых для решения поставленной задачи синхронизации визуальных и динамических составляющих имитации. В главе 4 будет рассмотрена модель стенда на базе робота-манипулятора, для которого проведено численное моделирование и оценка возможности решения задачи синхронизации визуальной и динамической имитации.

### **3.2 Отслеживание угловых движений человека и стенда с применением оптической системы**

Для отслеживания и прогнозирования углового движения стенда была разработана методика, предназначенная для использования с оптической системой трекинга. Система включает набор инфракрасных камер, излучатели, отражающие маркеры и специализированное программное обеспечение. Это программное обеспечение выполняет первичную обработку сигналов, отраженных от маркеров, и с высокой точностью определяет линейное и угловое положения маркеров, объединенных в "тело". Информация об изменении углового положения представляется в виде матрицы направляющих косинусов, на основе которой можно вычислить углы поворота. Однако для

экспериментальных целей требуется также определить угловую скорость (и аналогично, угловое ускорение) вращения "тела".

Поскольку информация от системы трекинга поступает дискретно с фиксированным временным шагом, математически задача сводится к определению функции по известному значению интеграла на заданном отрезке, без наложения ограничений на искомую функцию. В технических приложениях такую задачу рассматривают как поиск производной зашумленного сигнала. Существуют различные методы решения данной задачи [80; 115], однако выбор наилучшего метода затруднительный, так как каждый имеет свои преимущества и ограничения. В данном разделе представлен способ настройки стандартного фильтра, учитывающий специфику решаемой задачи.

Рассмотрим следующую постановку задачи: имеется зашумленная запись значений  $R_i, i = 0, \dots, n$ , матрицы ориентации  $R = r_{ij}$  с постоянным временным шагом  $\delta t = t_i - t_{i-1}, i = 1, \dots, n$ . Частота записи составляет 60 Гц. Требуется определить угловую скорость вращения тела.

Для достижения этой цели можно использовать кинематические соотношения, которые связывают изменение параметров, определяющих ориентацию тела, с его угловой скоростью вращения. В проведенном эксперименте, аналогично первой главе, использовалась оптическая система трекинга ARTtrack, изготовленная немецкой компанией A.R.T. GmbH [55].

Одним из видов кинематических соотношений являются уравнения Пуассона. Они определяют изменение матрицы ориентации  $R$  в зависимости от угловой скорости  $\omega$  вращения твердого тела [102]:

$$\dot{R} = \begin{bmatrix} 0 & \omega_z & -\omega_y \\ -\omega_z & 0 & \omega_x \\ \omega_y & -\omega_x & 0 \end{bmatrix} R, \quad (3.6)$$

где  $\omega_x, \omega_y, \omega_z$  — проекции вектора  $\omega$  на жестко связанные с телом оси.

В каждый момент времени можно численно определить производную матрицы ориентации с использованием первой разности и, ссылаясь на уравнения Пуассона, найти значения проекций угловой скорости  $\omega$  на оси тела в моменты времени  $t_i$ :  $\omega_{xi}, \omega_{yi}, \omega_{zi}$ .

Однако работа с несглаженными данными может привести к большим ошибкам в определении производной. Тем временем, сглаживание матрицы ориентации требует особой аккуратности и имеет свои особенности из-за необходимости поддержания ортонормальности ее строк и столбцов. По-

этому было предложено использовать кинематические уравнения для углов Эйлера–Крылова поворота тела  $\gamma, \beta, \alpha$  (следуя терминологии [89], угловое перемещение первого рода) вместо уравнений Пуассона, сглаживать данные и затем определять угловые скорости. Для оценки полученных результатов сравнивались исходная запись значений матрицы ориентации во времени и результаты интегрирования уравнений Пуассона при найденных значениях проекций угловой скорости  $\omega$  на оси тела. Использовались соотношения, связывающие элементы матрицы ориентации и углы поворота, в случае  $\beta \neq \pi/2$ :

$$\sin \alpha \cos \beta = -r_{32}, \sin \beta = -r_{31}, \sin \gamma \cos \beta = -r_{21}, \quad (3.7)$$

и выражения для угловой скорости через углы поворота и их производные [89]:

$$\omega_x = \dot{\alpha} \cos \beta \cos \gamma + \dot{\beta} \cos \gamma, \quad \omega_y = -\dot{\alpha} \cos \beta \sin \gamma + \dot{\beta} \sin \gamma, \quad \omega_z = \dot{\gamma} + \dot{\alpha} \sin \beta. \quad (3.8)$$

Сглаживание углов поворота осуществлялось двумя методами: с использованием быстрого преобразования Фурье (БПФ) и фильтрации высоких частот, а также с применением фильтра Савицкого–Голея [48].

Благодаря БПФ можно представить сглаживаемую функцию в виде частичной суммы ряда Фурье и вычислить производную. В качестве тестового случая использовалась запись следующего движения: сначала ”тело” остается неподвижным, затем совершает колебательные движения вокруг вертикальной оси, снова становится неподвижным, повторяет цикл колебаний и вновь оказывается в состоянии покоя. Запись содержит  $n = 9203$  точек. График зависимости угла поворота  $\gamma_i = \gamma(t_i)$  от времени представлен на рисунке 3.5. Углы  $\alpha$  и  $\beta$  остаются небольшими на протяжении всего эксперимента. С использованием функций пакета программ Matlab было выполнено БПФ и построены частичные суммы рядов  $\alpha^f(t), \beta^f(t), \gamma^f(t)$ , в которых исключены все члены с частотой  $\nu$  выше  $\nu^c$ . Результаты сравнивались для  $\nu^c = 1; 3; 10$  Гц. Во всех случаях достигается достаточно хорошее качество аппроксимации, причем оно улучшается (с точки зрения минимума максимального значения модуля ошибки) при переходе от  $\nu^c = 1$  Гц к  $\nu^c = 3$  Гц и практически не меняется при переходе от  $\nu^c = 3$  Гц к  $\nu^c = 10$  Гц. Затем вычислялись значения  $\dot{\alpha}_i, \dot{\beta}_i, \dot{\gamma}_i$  как производные частичных сумм  $\alpha^f(t), \beta^f(t), \gamma^f(t)$  в точках  $t_i, i = 0, \dots, n$ ;  $\dot{\alpha}_0, \dot{\beta}_0, \dot{\gamma}_0$  полагались равными нулю.

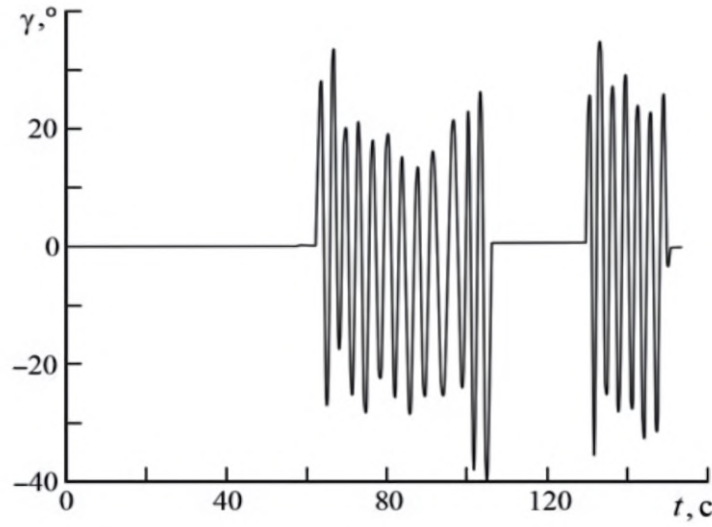


Рисунок 3.5 — График зависимости угла поворота  $\gamma$  от времени

Используя соотношение (3.8), уравнения Пуассона были проинтегрированы, применяя кубическую интерполяцию для угловых скоростей  $\omega_x(t), \omega_y(t), \omega_z(t)$  между узлами. Затем, в соответствии с (3.7), определялись  $\tilde{\alpha}_i^f, \tilde{\beta}_i^f, \tilde{\gamma}_i^f$  — значения углов поворота, восстановленные на основе угловой скорости. Можно подтвердить ранее сделанный вывод о достаточно хорошем приближении углов поворота во всех рассмотренных случаях. При этом аппроксимация улучшается (с точки зрения минимума максимального значения модуля ошибки) при переходе от  $\nu^c = 1$  Гц к  $\nu^c = 3$  Гц и практически не изменяется при переходе от  $\nu^c = 3$  Гц к  $\nu^c = 10$  Гц.

Однако стоит отметить существенное различие (с точки зрения максимального значения модуля угловой скорости) в полученных аппроксимациях для угловых скоростей  $\tilde{\omega}_x(t), \tilde{\omega}_y(t), \tilde{\omega}_z(t)$  при переходе от  $\nu^c = 1$  Гц к  $\nu^c = 3$  Гц и к  $\nu^c = 10$  Гц. В связи с этим был сделан вывод о необходимости предварительного определения максимальной частоты  $\nu^c$  на основе априорных данных о характере движения при использовании данного подхода в исследуемой задаче. Разность между исходным  $\gamma_i$  и восстановленным  $\tilde{\gamma}_i^f$  сигналами для  $\nu^c = 3$  Гц представлена на рисунке 3.6.

Альтернативный метод с использованием фильтра Савицкого-Голея заключается в следующем: выбираются размеры окна (количество точек, на основе которых рассчитываются коэффициенты, должно быть нечетным) и порядок полинома для аппроксимации данных. Коэффициенты полинома определяются с помощью метода наименьших квадратов. В данном случае были выбраны следующие параметры: для угла  $\gamma$  сглаживание проводилось

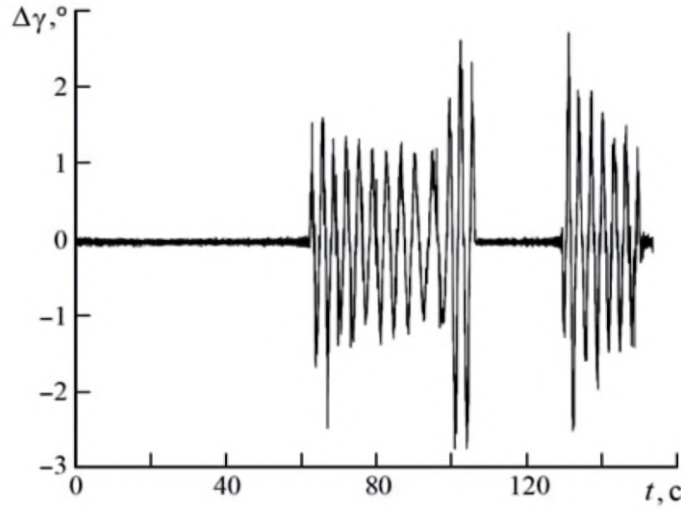


Рисунок 3.6 — Разность между исходным и восстановленным значениями  $\gamma$  при  $\nu^c = 3$  Гц

по 15 точкам (соответствует интервалу времени 0,25 с в записи) с полиномом 4-й степени; для углов  $\alpha$  и  $\beta$  сглаживание проводилось по 61 точке (соответствует интервалу времени 1 с в записи) с полиномом 2-й степени.

Так были найдены производные  $\dot{\alpha}_i$  и  $\dot{\beta}_i$  в точках  $t_i, i = 0, \dots, n - 1$ ; предполагалось, что  $\dot{\alpha}_0 = \dot{\alpha}_n = \dot{\beta}_0 = \dot{\beta}_n = 0$ . Затем, с применением соотношений (3.8), рассчитывались  $\omega_x(t_i), \omega_y(t_i), \omega_z(t_i)$  и производилось интегрирование уравнений Пуассона. Аналогично предыдущему методу, для угловых скоростей использовалась кубическая интерполяция между узлами. В соответствии с (3.7), определялись восстановленные по угловой скорости значения углов поворота  $\tilde{\alpha}_i^v, \tilde{\beta}_i^v, \tilde{\gamma}_i^v$ . Разность между исходным  $\gamma(t_i)$  и восстановленным  $\tilde{\gamma}_i^v$  сигналами представлена на рисунке 3.7.

Следует отметить, что максимальная разность между исходными и восстановленными значениями угла  $\gamma$  (диапазон изменения которого в эксперименте значительно превышает остальные два угла) на рисунке 3.7 почти в 3 раза меньше, чем на рисунке 3.6. Восстановленные значения углов достаточно близки к измеренным, что свидетельствует о потенциальной применимости данного метода для определения угловой скорости. На рисунке 3.8 представлена проекция вектора угловой скорости  $\tilde{\omega}_z(t)$ , вычисленная с использованием этого подхода.

Исключительно важно отметить, что данный метод требует априорных сведений о характере движения объекта. Вместо использования максимальной частоты, следует определить степень интерполирующего полинома и ко-

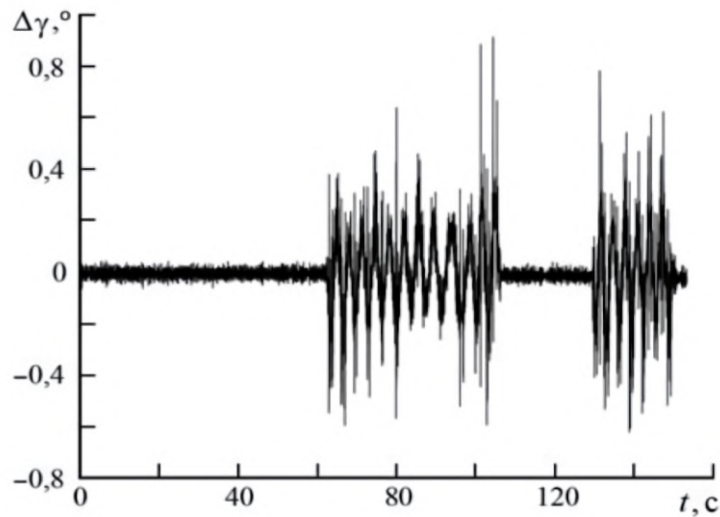


Рисунок 3.7 — Разность между исходным и восстановленным значениями  $\gamma$

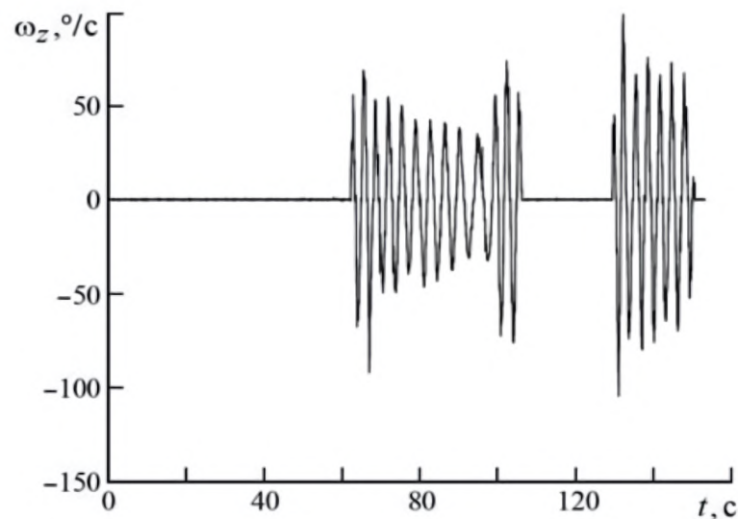


Рисунок 3.8 — Вычисленная проекция  $\omega_z$  вектора угловой скорости

личество точек сглаживания. Фильтр Савицкого-Голея, используемый в этом подходе, позволяет обрабатывать данные в реальном времени с минимальной задержкой, которая зависит от размера выбранного окна сглаживания.

Таким образом, было разработано и настроено программное обеспечение, обеспечивающее определение угловой скорости вращения тела при применении системы трекинга. Полученные оптические данные могут быть интегрированы с инерциальной информацией, обеспечивая комплексный подход к отслеживанию движений. Для реализации этого использовалась гибридная стратегия, основанная на совместном оптическом и инерциальном трекинге [83; 91].

### 3.3 Построение прогноза углового движения подвижного стенда с учетом его динамики и времени задержки системы отслеживания движений

Как было упомянуто ранее, данные, полученные от системы трекинга, поступают с некоторой задержкой. Аналогичным образом, процесс визуализации изображения также испытывает задержку. В связи с этим возникает необходимость использования прогнозирования движения объектов, определяющих положение камеры в сцене, с учетом суммарной задержки всех систем.

В данном разделе рассмотрим два алгоритма прогнозирования преобразования от связанной с креслом самолета системы координат к системе координат виртуальной камеры для случая, когда экран системы визуализации остается неподвижным (в результате чего виртуальная камера также не изменяет свою ориентацию относительно системы координат комнаты), а динамическая платформа перемещается независимо. Такая комбинация характерна, например, для Панорамной системы виртуальной реальности МГУ [83].

Первый рассматриваемый алгоритм основан на прогнозировании положения виртуальной камеры при предположении, что следующие несколько шагов по времени (обычно порядка сотых долей секунды) управление, подаваемое пилотом, остается неизменным. В этом случае для прогноза используются модель транспортного средства (в частном случае — летательного аппарата) и модель подвижной платформы.

Второй алгоритм основывается на прогнозировании положения виртуальной камеры при предположении, что в течение следующих нескольких шагов по времени угловая скорость движения платформы  $\omega \equiv \text{const}$ .

Сравнение работы алгоритмов прогнозирования движения может быть выполнено с использованием функционала следующего вида:

$$J(t, \Delta t) = \sum_{t_j} \sum_{i=1}^3 [\tilde{\omega}_i^{\text{real}}(t_j) - \tilde{\omega}_i(t_j)]_{\mu_i+}, \quad (3.9)$$

где  $\tilde{\omega}^{\text{fore}}$  — оценка угловой скорости стенда, полученная из прогноза,  $\tilde{\omega}$  — фактическая оценка угловой скорости объекта (получаемая, например, описанным в предыдущем разделе методом),  $\mu = (\mu_1, \mu_2, \mu_3)^\top$  — вектор порогов

чувствительности полукружных каналов вестибулярного аппарата по угловой скорости,  $[\cdot]_{\mu_i+}$  — оператор (3.4).

На каждом временном шаге записываются значения функционала для каждого из методов. На основе проведенных экспериментов сравниваются полученные данные, и делается вывод о том, какой из алгоритмов обеспечивает лучшее решение задачи минимизации функционала.

### 3.3.1 Численный эксперимент

В рамках проведения численного эксперимента использовалась модель движения транспорта [72], реализованная с помощью графического движка Unity 3D. Модель была разработана на языке программирования C#. Система дифференциальных уравнений решалась с использованием метода Рунге-Кутты четвертого порядка [78], что обеспечивало точность и стабильность численного решения.

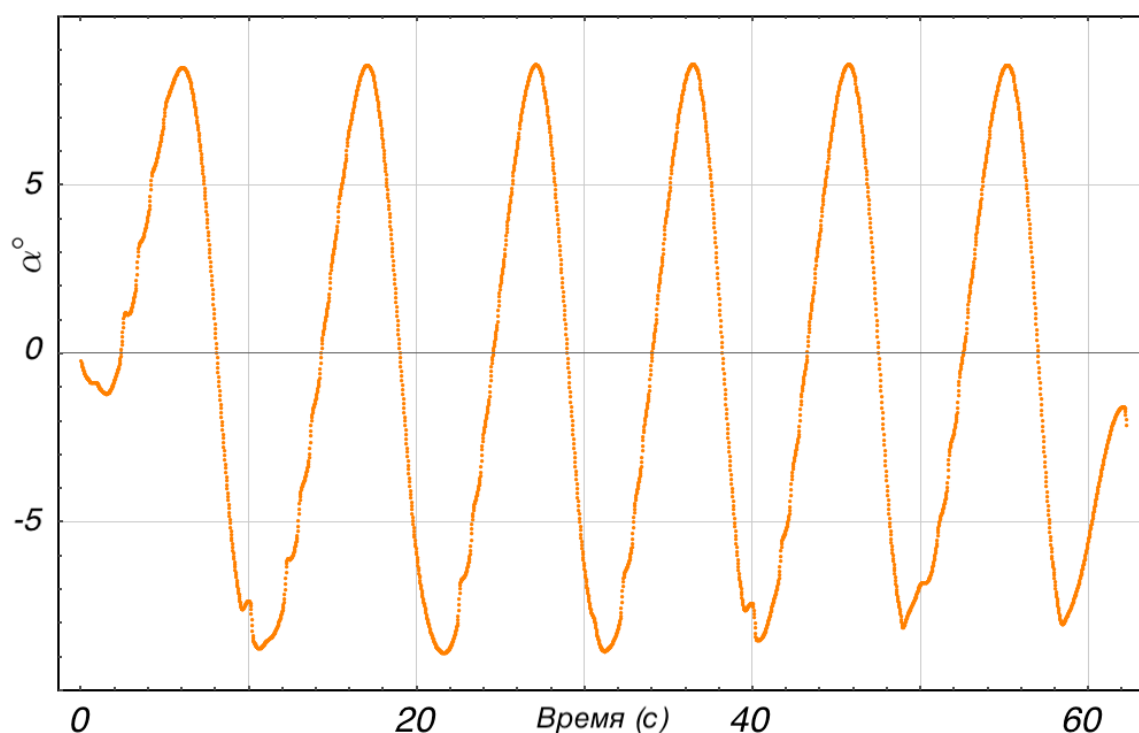


Рисунок 3.9 — Движение с синусоидальным управлением

Эксперимент был разделен на два этапа. На первом этапе, в ходе программной реализации, на каждом кадре записывались полные данные о движении модели транспорта и платформы в соответствии с алгоритмом управления, учитывая фиксированные задержки трекинга. Под полными данными понимаются не только углы наклона, необходимые для дальнейшей оценки



алгоритмов прогнозирования, но и информация о скорости движения транспорта и его координатах на каждом кадре. На втором этапе, в соответствии с записанными данными, выполнялся алгоритм прогнозирования движения, и результаты сравнивались.

Для эксперимента был выбран модельный сценарий управления, обеспечивающий периодические изменения угла тангажа у транспортного средства. В качестве алгоритма динамической имитации применялся алгоритм имитации ориентации вектора перегрузки, описанный в главе 1.

### 3.3.2 Сравнение алгоритмов прогнозирования

Рассмотрим, например, движения платформы только по оси тангажа. На рисунке 3.9 представлен график, иллюстрирующий угол наклона виртуального экрана относительно неподвижной системы координат. Ось абсцисс отображает время, ось ординат — угол наклона виртуального экрана  $\alpha(t)$  в соответствующий момент времени.

Данные, полученные после применения алгоритмов прогнозирования, анализировались и сравнивались путем вычисления функционала (3.9), предложенного в предыдущем разделе. В качестве порогов чувствительности выбирались значения  $\mu_1 = \mu_2 = \mu_3 = 2^\circ/\text{с}$ . Сравнивались также максимальные отклонения прогнозируемых углов от фактических на соответствующем временном шаге. Примеры отклонений при разных величинах задержки представлены на рисунке 3.10.

Для сравнения алгоритмов была составлена таблица 1. Столбцы таблицы соответствуют значению  $\Delta t$  - временному шагу, на котором выполнялся прогноз. Строки таблицы, начиная со второй, по порядку представляют: функционал  $J_c$  вида (3.9) для алгоритма прогнозирования на основе управления, функционал  $J_l$  вида (3.9) для алгоритма линейного прогнозирования, абсолютные максимальные отклонения  $\Delta\alpha_c$  и  $\Delta\alpha_l$  для соответствующих алгоритмов прогнозирования.

Основываясь на таблице 1, можно сделать вывод о том, что алгоритм линейного прогнозирования по сравнению с прогнозированием на основе управления демонстрирует более высокие значения как функционала (3.9), так и максимального отклонения, независимо от временного шага прогноза. Анализ данных, представленных в таблице и на графике, позволяет отметить, что линейный прогноз оказывается почти таким же эффективным

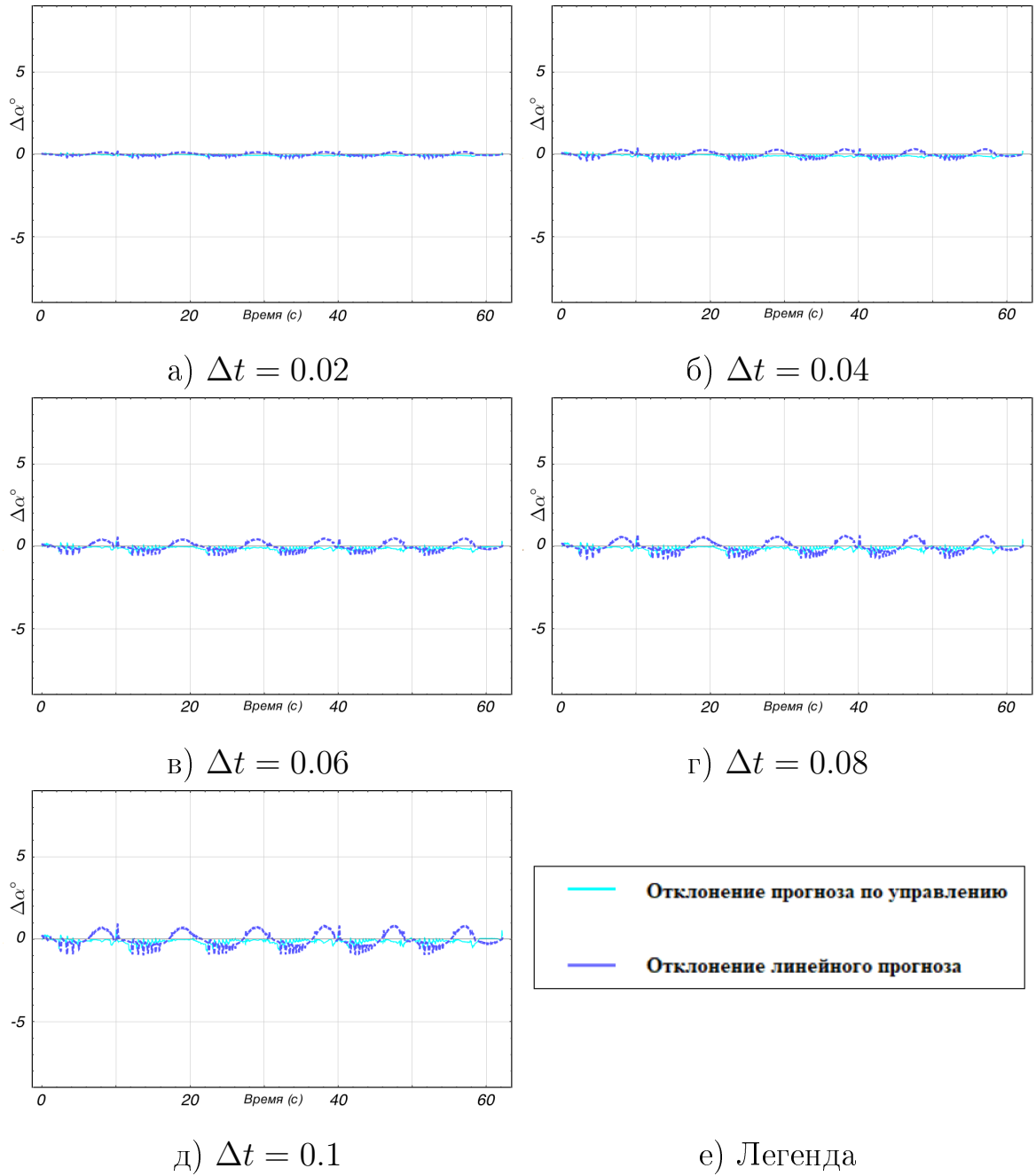


Рисунок 3.10 — Графики отклонений оригинальных и прогнозируемых данных для различных алгоритмов прогноза

при небольших значениях задержки, как и прогноз на основе модели. Однако при больших значениях задержки прогноз на основе модели становится значительно лучше. Полученные результаты подтверждают, что линейный прогноз может быть использован для решения задачи синхронизации визуальной и динамической составляющих имитации при малых значениях задержки (например, при стабильной частоте кадров в 60 Гц). Для прогноза движения головы, в свою очередь, приходится использовать только линейный прогноз вследствие отсутствия модели управления.

$\Delta t$	0.02	0.04	0.06	0.08	0.1
$J_c$	234.4	851.9	1798.6	2977.7	4172.4
$J_l$	330.5	1322.2	2975	5289	8264.8
$\max  \Delta\alpha_c $	0.15	0.28	0.45	0.62	0.77
$\max  \Delta\alpha_l $	0.2	0.39	0.59	0.78	0.97

Таблица 1 — Сводная таблица данных о движении

В целом, исследование показывает, что выбор алгоритма прогнозирования в значительной степени зависит от конкретных условий и требований к задаче. При определении подходящего алгоритма необходимо учитывать характеристики системы и ожидаемые значения задержек в контурах визуальной и динамической составляющих имитации.

### 3.4 Построение управления стендом с учетом порога чувствительности вестибулярного аппарата

В предыдущих разделах были рассмотрены методы решения задач, связанных с созданием визуализации, достаточно точной для обеспечения синхронизации с порогом чувствительности полукружных каналов вестибулярного аппарата. Как указывалось в начале главы, для достижения синхронизации визуальной и динамической составляющих имитации необходимо также имитировать перегрузки с точностью до порогов чувствительности отолитовых органов вестибулярного аппарата.

Для выполнения активной фазы динамической имитации необходимо обеспечить требуемый вектор ускорения для платформы, учитывая пороги чувствительности вестибулярного аппарата. Традиционная задача управления роботами формулируется в терминах перемещения рабочего органа по за-

данной траектории с определенной скоростью [65]. В данной работе рассматривается задача отслеживания линейного ускорения кабины, расположенной на рабочем органе робота, с точностью до порогов чувствительности механорецепторов вестибулярного аппарата человека, определенных для каждого направления — вектора  $\epsilon = (\epsilon_1, \epsilon_2, \epsilon_3)^\top$ .

Формирование алгоритма управления, который решает поставленную задачу, является основным фокусом данного раздела. Разработанный алгоритм управления должен быть оптимальным с точки зрения нестрого выпуклого функционала качества, который описывает разницу между ускорением в расчетной точке подвижного стенда и желаемым ускорением (полученным из модели самолета). Рассматриваемое управление будет анализироваться с точки зрения сходимости ошибки имитации к окрестности нуля за конечное время. Алгоритм создается для случая наличия неполной информации о модели стенда.

### 3.4.1 Постановка задачи управления с использованием измерения ускорения и учетом порога чувствительности вестибулярного аппарата человека

Предполагается, что ускорения расчетной точки известны или определены путем вычитания гравитации из корректно сориентированных акселерометров. Задача состоит в разработке такого управления  $u(t)$ , чтобы ускорение конечной точки  $w(t) \in \mathbb{R}^3$  могло повторять эталонное ускорение  $w^p(\cdot) \in C^2[\mathbb{R}_+]$ ,  $w^p(t) \in \mathbb{R}^3$ , полученное из модели самолета, с точностью до допустимого вектора порогов чувствительности  $\epsilon$ .

Для создания алгоритма управления предлагается использовать не только ошибку отслеживания ускорения (с учетом порогов чувствительности), но и ее производную.

Определим разницу между ускорениями как  $\Delta(t) = w^p(t) - w(t)$ . Теперь введем "сглаженную" разницу между ускорениями  $\delta(t)$ , которая учитывает пороги чувствительности:

$$\delta(t) := [\Delta(t) \circ \text{Sig}(\Delta(t)) - \epsilon]^{(1+\nu)} \circ \text{Sig}^{1/\mu}(\Delta(t)). \quad (3.10)$$

Здесь  $\mu > 0$  является нечетным целым числом. Скалярное значение  $0 < \nu < 1$  может быть выбрано малым, чтобы производная по времени  $\delta(t)$  была близка к производной по времени  $\Delta(t)$  вне зоны чувствительности. Произведение  $\circ$  соответствует покомпонентному умножению. Векторная

операция  $Sig(\cdot)$  представляет собой векторный сглаженный сигнум, сформированный из  $\tanh$ -функций (гиперболический тангенс).

Производную  $\delta(t)$  по времени можно вычислить следующим образом:

$$\frac{d}{dt}\delta = \Sigma(\Delta) \circ \frac{d}{dt}\Delta, \quad (3.11)$$

где  $\Sigma(\cdot)$  — функция от  $\Delta$ :

$$\begin{aligned} \Sigma(\Delta) &:= [\Delta \circ Sig(\Delta) - \epsilon]^\nu \circ \\ &\left[ \frac{1}{q} [\Delta \circ Sig(\Delta) - \epsilon] \circ Sig^{(1-\mu)/\mu}(\Delta) \circ Sig'(\Delta) + \right. \\ &\left. (1 + \nu) (\Delta \circ Sig'(\Delta) + Sig(\Delta)) \circ Sig^{1/\mu}(\Delta) \right]. \end{aligned} \quad (3.12)$$

Здесь штрих обозначает частную производную по  $\Delta$ . Используя выражение (3.12), можно показать, что векторная функция  $\Sigma(\Delta)$  ограничена, если  $\Delta$  ограничена (так как условие Липшица выполняется).

Следует отметить, что внутри инвариантной области, определенной как  $Sig(\Delta) \circ \Delta - \epsilon < 0$  (выражение  $<$  понимается поэлементно), выполняется следующее равенство:

$$\Sigma(\Delta) = 0.$$

Для определения значения производной ошибки ускорения необходимо вычислить изменение эталонных ускорений во времени:

$$h_w(t) = \frac{d}{dt}w^p. \quad (3.13)$$

где известную функцию  $h_w : \mathbb{R}^3 \rightarrow \mathbb{R}^3$  считаем ограниченной и непрерывной с ограниченной производной:

$$\left\| \frac{d}{dt}h_w \right\| \leq dH_+.$$

где  $dH_+$  — положительный скаляр.

Для оценки качества управления введем функционал  $J(\delta)$  вида (3.5):

$$J(\delta) = \|\delta\|_1 = \sum_{i=1}^3 \lambda_i |\delta_i|, \quad \sum_{i=1}^3 \lambda_i = 1, \lambda_i > 0. \quad (3.14)$$

Скаляр  $\delta_i$  обозначает  $i$ -ю компоненту вектора  $\delta$ .

Формулировка задачи управления может быть представлена как оптимизация этого нестрого выпуклого функционала. Предположим, что существует минимум указанного функционала. Задача оптимизации формулируется следующим образом:

$$J(\delta(t)) \xrightarrow{t \rightarrow \infty} \min_{u(\cdot) \in U_{adm}}. \quad (3.15)$$

Так как функционал негладкий, вместо градиента применим субградиент  $a(\delta)$  (по отношению к  $\Delta$ ), который определяется следующим образом:

$$a(\delta) = \begin{bmatrix} \lambda_1 \text{sign}(\delta_1) & 0 & 0 \\ 0 & \lambda_2 \text{sign}(\delta_2) & 0 \\ 0 & 0 & \lambda_3 \text{sign}(\delta_3) \end{bmatrix} \Sigma(\Delta). \quad (3.16)$$

### 3.4.2 Структура скользящего управления

Введем вспомогательную векторную переменную  $s(t)$ , которую назовем скользящей:

$$s = \dot{\Delta} + \frac{\delta + \eta}{t + \theta} + \frac{1}{t + \theta} \int_0^t a(\delta(t_s)) dt_s. \quad (3.17)$$

Здесь  $\theta \equiv \text{const} > 0$ ,  $\eta = -\theta \frac{d}{dt} \Delta|_{t=0} - \frac{d}{dt} \delta|_{t=0}$ .

Управление будет разработано с использованием интегрального скользящего режима. Интегральный скользящий режим может применяться к специальному классу поверхностей скольжения, основанных на усредненном субградиенте, оптимальных с точки зрения линейного функционала качества. Этот метод получил название интегрального скользящего режима с усредненным субградиентом (ИСП-УСГ).

В работе [43] предложена поверхность скольжения, оптимизирующая функционал вида (3.14). В отличие от упомянутой работы, в нашей задаче скользящая переменная включает ускорение вместо скоростей и координат, и учитывает порог чувствительности. В результате, вместо поверхности скольжения возникает подпространство, которое можно назвать "слоем скольжения".

Для скользящей переменной можно доказать следующее утверждение:

**Лемма 1.** *В случае, если с некоторого момента времени  $t_0 > 0$  за счет изменения имитирующего ускорения  $w(t)$  можно обеспечить нахождение на скользящей поверхности  $s(t) \equiv 0 \forall t > t_0 \geq 0$ , то существует такой скаляр  $\Psi$ , что верна оценка*

$$J(\delta) \leq \frac{\Psi}{t + \theta}.$$

## Доказательство

Для упрощения записи выкладок в доказательстве используем нижний индекс "t", обозначающий зависимость от времени. Введем вспомогательную переменную  $\theta_t := t + \theta$  и определим переменную  $\zeta_t$ , удовлетворяющую следующим соотношениям:

$$\left. \begin{aligned} \dot{\zeta}_t &= -a(\delta_t), \\ \zeta_{t_0} &= 0, \Delta_{t_0} \text{ известно}, \\ \zeta_t &= \theta_t \dot{\Delta}_t + \delta_t + \eta, \quad t \geq t_0 \geq 0, \\ t_0 &\text{ — момент, с которого мы оказываемся в "слое скольжения"}. \end{aligned} \right\} \quad (3.18)$$

Производная разницы ускорений  $\dot{\Delta}$ , при котором система находится в "слое скольжения" (то есть  $s(t) \equiv 0$ ), удовлетворяет уравнению:

$$\dot{\Delta} = -\frac{\delta + \eta}{t + \theta} - \frac{1}{t + \theta} \int_0^t a(\delta(t_s)) dt_s. \quad (3.19)$$

Внутри зоны чувствительности (при оптимальном режиме) верно  $\delta^* \equiv \delta(\Delta_t^*) \equiv 0$ . Отметим, что для выпуклого функционала (3.15) выполняется следующее свойство:

$$a(\delta_t)^\top (\delta_t - \delta_t^*) \geq J(\delta_t) - J_t^* \quad (3.20)$$

где  $J_t^* \equiv 0$  — наилучшее значение  $J$ .

Выпишем следующее соотношение:

$$\begin{aligned} \frac{d}{dt} \left[ \frac{1}{2} \|\zeta_t\|^2 - \zeta_t^\top \delta^* \right] &= \dot{\zeta}_t^\top (\zeta_t - \delta^*) = \\ &= -a^\top(\delta_t) \left[ \theta_t \dot{\Delta}_t + \delta_t + \eta - \delta^* \right] = \\ &= -a^\top(\delta_t) (\delta_t - \delta^*) - a^\top(\delta_t) (\theta_t \dot{\Delta}_t + \eta). \end{aligned}$$

Из определения субградиента следует:

$$a^\top(\delta_t) \dot{\Delta}_t \geq \frac{d}{dt} J(\delta_t) = \frac{d}{dt} [J(\delta_t) - J_t^*].$$

Тогда, вместе с соотношением (3.20), оно дает:

$$\frac{d}{dt} \left[ \frac{1}{2} \|\zeta_t\|^2 - \zeta_t^\top \delta^* \right] \leq -[J(\delta_t) - J_t^*] - \theta_t \frac{d}{dt} [J(\delta_t) - J_t^*] - a^\top(\delta_t) \eta.$$

Получаем:

$$J(\delta_t) - J_t^* \leq -\frac{d}{dt} \left[ \frac{1}{2} \|\zeta_t\|^2 - \zeta_t^\top \delta^* \right] - \theta_t \frac{d}{dt} [J(\delta_t) - J_t^*] - a^\top(\delta_t) \eta.$$

Затем, интегрируя это неравенство на интервале  $[t_0, t]$  (с применением интегрирования по частям), получаем:

$$\begin{aligned} \int_{t_0}^t [J(\delta_\tau) - J_\tau^*] d\tau &\leq \frac{1}{2} \left( \|\zeta_{t_0}\|^2 - \|\zeta_t\|^2 \right) + \\ &+ (\zeta_t - \zeta_{t_0})^\top \delta^* - \theta_t [J(\delta_t) - J_t^*] + \theta_{t_0} [J(\delta_{t_0}) - J_{t_0}^*] + \\ &+ \int_{t_0}^t [J(\delta_\tau) - J_\tau^*] \dot{\theta}_\tau d\tau - \left[ \int_{t_0}^t a^\top(\delta_\tau) d\tau \right] \eta. \end{aligned}$$

Введем  $\Psi = (t_0 + \theta)J(\delta_{t_0}) + \frac{1}{2}\|\eta\|^2$ .

Поскольку  $\dot{\theta}_\tau \equiv 1$ ,  $J_t^* \equiv 0$ ,  $\delta^* \equiv 0$ , и применяя определение  $\zeta_t$ , получаем:

$$\begin{aligned} \theta_t [J(\delta_t) - J_t^*] &\leq \theta_{t_0} [J(\delta_{t_0}) - J_{t_0}^*] + \\ &+ \frac{1}{2} \left( \|\zeta_{t_0}\|^2 - \|\zeta_t\|^2 \right) + (\zeta_t - \zeta_{t_0})^\top \delta^* + \zeta_t^\top \eta = \\ &(t_0 + \theta) [J(\delta_{t_0}) - J_{t_0}^*] + \left( \frac{1}{2} \|\zeta_{t_0}\|^2 - \zeta_{t_0}^\top \delta^* \right) + \\ &+ \frac{1}{2} \|\delta^* - \eta\|^2 - \underbrace{\frac{1}{2} \left[ \|\zeta_t\|^2 - 2\zeta_t^\top (\delta^* - \eta) + \|\delta^* - \eta\|^2 \right]}_{\|\zeta_t - (\delta^* - \eta)\|^2} \leq \\ &(t_0 + \theta) [J(\delta_{t_0}) - J_{t_0}^*] - \frac{1}{2} \|\zeta_t - (\delta^* - \eta)\|^2 + \\ &+ \left( \frac{1}{2} \|\zeta_{t_0}\|^2 - \zeta_{t_0}^\top \delta^* \right) + \frac{1}{2} \|\delta^* - \eta\|^2 \leq \Phi, \end{aligned}$$

что доказывает лемму 1.

□

Для построения управления введем ряд вспомогательных матриц, определенных конструкцией стенда. Предположим, что динамика стенда может быть представлена следующим образом:

$$\ddot{q} = A^{-1}(q)(u - C(q, \dot{q})\dot{q} - G(q)), \quad (3.21)$$

Здесь  $q(t) \in \mathbb{R}^n, n \in \mathbb{N}$  — обобщенные координаты,  $u(t) \in \mathbb{R}^k, k \in \mathbb{N}$  — управление, представляющее собой моменты сил, выдаваемых двигателями.

Систему можно переписать в виде:

$$\ddot{q} = \Phi_a(\dot{q}, q) + \Phi_b(q)u. \quad (3.22)$$

$$\Phi_a(\dot{q}, q) = -A^{-1}(q)(C(q, \dot{q})\dot{q} + G(q)) \in \mathbb{R}^{n \times n}, \quad \Phi_b(q) = A^{-1}(q) \in \mathbb{R}^{n \times k}.$$

Отметим, что правая часть уравнения (3.22) может быть известна неточно из-за наличия возмущений.



Для построения интегрального скользящего режима требуется расширить исходную систему дифференциальных уравнений (3.22), включив в качестве переменных  $u$  и  $\dot{u}$ . Для их формирования требуется найти изменение ускорения в пространстве обобщенных координат представляется следующим образом:

$$\ddot{q} = F_a(\ddot{q}, \dot{q}, q) + F_b(\dot{q}, q)u + \Phi_b(q)\dot{u}. \quad (3.23)$$

Матрицы  $F_a \in \mathbb{R}^{n \times n}$  и  $F_b \in \mathbb{R}^{n \times k}$  получаются из дифференцирования  $\Phi_a$  и  $\Phi_b$  соответственно.

Предположим, что ускорение расчетной точки может быть представлено следующим образом:

$$\begin{aligned} \frac{d}{dt} w(\ddot{q}, \dot{q}, q) &= A_3(q) \ddot{q} + A_2(\dot{q}, q) \ddot{q} + A_1(\dot{q}, q) \dot{q} := \\ &:= \frac{\partial w(\ddot{q}, \dot{q}, q)}{\partial \ddot{q}} \ddot{q} + \frac{\partial w(\ddot{q}, \dot{q}, q)}{\partial \dot{q}} \ddot{q} + \frac{\partial w(\ddot{q}, \dot{q}, q)}{\partial q} \dot{q}. \end{aligned} \quad (3.24)$$

Поскольку в предложенной скользящей переменной фигурирует производная от разницы между желаемым и подаваемым ускорениями (с учетом порога чувствительности), выразим  $\dot{\delta}$  через обобщенные координаты и их производные:

$$\dot{\delta} = \Sigma(\Delta) \circ (h_w - A_3(q) \ddot{q} - A_2(\dot{q}, q) \ddot{q} - A_1(\dot{q}, q) \dot{q}).$$

С учётом (3.23), можно получить дифференциальное уравнение для производной ошибки:

$$\dot{\Delta} = \tilde{F} + G_1 u + G_2 \dot{u}, \quad (3.25)$$

где

$$\begin{aligned} \tilde{F} &= h_w - A_3(q) F_a(\ddot{q}, \dot{q}, q) - \\ &- A_2(\dot{q}, q) \Phi_a(\dot{q}, q) - A_1(\dot{q}, q) \dot{q}, \\ G_1 &= -A_3(q) F_b(\dot{q}, q) - A_2(\dot{q}, q) \Phi_b(q), \\ G_2 &= -A_3(q) \Phi_b(q). \end{aligned} \quad (3.26)$$

Таким образом, производную по времени от  $\delta$  можно представить в следующем виде:

$$\dot{\delta} = \Sigma(\Delta) \circ (\tilde{F} + G_1 u + G_2 \dot{u}). \quad (3.27)$$

Теперь наша задача состоит в определении закона управления стендом. Для этого построим управляющее воздействие  $u(t)$  с использованием *инте-*

гравльного скользящего режима следующим образом:

$$\begin{aligned}
\ddot{u} &= \begin{cases} G_2^* \tau^c, & \text{если } \|\delta\| = 0, \\ G_2^* \tau, & \text{если } \|\delta\| > 0, \end{cases} \\
\tau &= -k \text{Sign}(s) - \xi_1(u, \dot{u}, t) - \xi_2(\delta, \dot{\delta}, t), \\
\tau^c &= -\xi_1(u, \dot{u}, t) - \xi_2(\delta, \dot{\delta}, t), \\
\xi_1(u, \dot{u}, t) &= \left(\frac{d}{dt} G_1\right) u + \left(G_1 + \frac{d}{dt} G_2\right) \dot{u}, \\
\xi_2(\delta, \dot{\delta}, t) &= -\frac{\delta + \eta}{(t + \theta)^2} + \frac{\dot{\delta} + a(\delta)}{(t + \theta)} - \frac{1}{(t + \theta)^2} \int_0^t a(\delta(t_s)) dt_s, \\
\text{Sign}(s) &= [\text{sign}(s_1), \text{sign}(s_2), \text{sign}(s_3)]^\top.
\end{aligned} \tag{3.28}$$

Здесь  $k$  — скалярный коэффициент усиления,  $G_2^* = G_2^\top [G_2 G_2^\top]^{-1}$  обозначает псевдоинверсию Мура-Пенроуза [42]. Предполагается, что такая матрица существует (вырожденный случай рассмотрен в работе [17]). Примем  $u(0) = 0, \dot{u}(0) = 0$ .

Сформулируем **основную теорему**:

**Теорема 1.** Пусть управление системой (3.22) представлено в виде (3.28) со скользящей переменной (3.17) и с известными матрицами (3.26).

При этом желаемое ускорение кинематически реализуемо и  $\exists f^+ \equiv \text{const}$  такая, что для  $\forall t > 0$  выполнено условие  $\left\| \frac{d}{dt} \tilde{F} \right\| \leq f^+$ .

Тогда для  $\forall t_{\text{reach}} > 0$  можно выбрать коэффициент усиления  $k = f^+ + \rho$ , где  $\rho = \frac{\|s(0)\|}{t_{\text{reach}}} > 0$ , в результате чего управление вида (3.28) обеспечит выход системы на скользящую поверхность  $s(t) \equiv 0 \ \forall t \geq t_{\text{reach}}$ .

При этом  $\forall \beta > \max\{\epsilon_1, \epsilon_2, \epsilon_3\} \ \exists t_{\text{conv}} > t_{\text{reach}}$  такое, что

$$\|\Delta(t)\| \leq \beta \quad \forall t \geq t_{\text{conv}}.$$

### Доказательство

Здесь, как и ранее, для упрощения записи выкладок в доказательстве будет применяться нижний индекс  $_t$ , обозначающий зависимость от времени. Положив  $t_0 = 0$ , представим производную скользящей переменной в виде:

$$\dot{s}_t = \ddot{\Delta}_t - \frac{\delta_t + \eta}{(t + \theta)^2} + \frac{\dot{\delta}_t + a(\delta_t)}{(t + \theta)} - \frac{1}{(t + \theta)^2} \int_{t_0}^t a(\delta_{t_s}) dt_s. \tag{3.29}$$

где

$$\ddot{\Delta}_t = \frac{d}{dt}\tilde{F}_t + \left(\frac{d}{dt}G_{1,t}\right)u_t + \left(G_{1,t} + \frac{d}{dt}G_{2,t}\right)\dot{u}_t + G_{2,t}\ddot{u}_t.$$

где  $\tilde{F}_t$ ,  $G_{1,t}$  и  $G_{2,t}$  удовлетворяют (3.26).

Пусть  $\|\delta\| > 0$ . Для доказательства того, что ошибка имитации практически устойчива, рассмотрим кандидата на функцию Ляпунова, заданную:

$$V(s_t) = \frac{1}{2} \|s_t\|^2.$$

Производная  $\dot{V}$  в силу системы:  $\dot{V} = s_t^\top \dot{s}_t$ . Используя вид производной  $s_t$ , заданный (3.29), в  $\dot{V}$ , получаем следующее дифференциальное уравнение:

$$\dot{V} = s_t^\top \left( \ddot{\Delta}_t - \frac{\delta_t + \eta}{(t+\theta)^2} + \frac{\dot{\delta}_t + a(\delta_t)}{(t+\theta)} - \frac{1}{(t+\theta)^2} \int_{t_0}^t a(\delta_{t_s}) dt_s \right).$$

Используя форму  $\ddot{\Delta}_t$  из (3.25), производную кандидата на функцию Ляпунова можно окончательно записать в виде:

$$\begin{aligned} \dot{V} = s_t^\top & \left( \frac{d}{dt}\tilde{F}_t + \left(\frac{d}{dt}G_{1,t}\right)u_t + \right. \\ & + \left(G_{1,t} + \frac{d}{dt}G_{2,t}\right)\dot{u}_t + G_{2,t}\ddot{u}_t - \frac{\delta_t + \eta}{(t+\theta)^2} + \\ & \left. + \frac{\dot{\delta}_t + a(\delta_t)}{(t+\theta)} - \frac{1}{(t+\theta)^2} \int_{t_0}^t a(\delta_{t_s}) dt_s \right). \end{aligned}$$

Подставляя определения  $\xi_1$  и  $\xi_2$ , получаем следующее дифференциальное уравнение:

$$\dot{V} = s_t^\top \left( \frac{d}{dt}\tilde{F}_t + \xi_1(u_t, \dot{u}_t, t) + G_{2,t}\ddot{u}_t + \xi_2(\delta_t, \dot{\delta}_t, t) \right).$$

Если существует псевдообратная матрица для  $G_{2,t}$ , то, используя вид управления из (3.28), можно определить производную кандидата на функцию Ляпунова как:

$$\begin{aligned} \dot{V} = s_t^\top & \left( \frac{d}{dt}\tilde{F}_t + \xi_1(u_t, \dot{u}_t, t) + \xi_2(\delta_t, \dot{\delta}_t, t) + \right. \\ & \left. + G_{2,t}G_{2,t}^\top [G_{2,t}G_{2,t}^\top]^{-1} \tau_t \right). \end{aligned}$$

После подстановки  $\tau_t$  получается:

$$\dot{V} = s_t^\top \left( \frac{d}{dt}\tilde{F}_t - k \text{Sign}(s_t) \right) \leq \|s_t\| f^+ - k \sum_{i=1}^3 |s_{i,t}|.$$

Учитывая, что  $\|s_t\| \leq \sum_{i=1}^3 |s_{i,t}|$ , получаем:

$$\dot{V} \leq (f^+ - k) \|s_t\|.$$

Взяв определение  $V$  и коэффициент усиления  $k = f^+ + \rho$ , получаем следующее неравенство:

$$\dot{V} \leq -\rho\sqrt{2V}.$$

Интегрирование неравенства дает  $\sqrt{V_t} \leq \sqrt{V_{t_0}} - \frac{\rho(t-t_0)}{\sqrt{2}}$ .

Можно заметить, что если

$$\sqrt{V_{t_0}} - \frac{\rho(t-t_0)}{\sqrt{2}} = 0,$$

то для всех

$$t \geq t_{reach} := \frac{\sqrt{2V_{t_0}}}{\rho} + t_0.$$

получаем

$$V_t \equiv 0.$$

Это доказывает сходимость  $V$  за конечное время.

Отсюда следует, что  $s_t \equiv \dot{s}_t \equiv 0$  для всех  $t \geq t_{reach}$ . Первое утверждение теоремы доказано.

Второе утверждение теоремы следует напрямую из леммы 1, поскольку сходимость  $J(\delta_t)$  (и как следствие  $\delta_t$ ) к нулю означает сходимость  $\Delta_t$  к окрестности нуля с точностью до порогов чувствительности:

$$\limsup_{t \rightarrow \infty} \|\Delta_t\| \leq \max\{\epsilon_1, \epsilon_2, \epsilon_3\}.$$

Применив определение практической устойчивости, получаем, что  $\forall \beta > \max\{\epsilon_1, \epsilon_2, \epsilon_3\} \exists t_{conv} > t_{reach}$ :

$$\|\Delta_t\| \leq \beta \quad \forall t > t_{conv}. \quad (3.30)$$

Теорема доказана.  $\square$

Таким образом, теорема показывает, что можно добиться необходимого качества имитации путем подбора соответствующего коэффициента усиления. Однако на практике существует ряд факторов, ухудшающих свойства устойчивости, которые следует учитывать, и которые связаны с особенностями динамической системы и системы управления.

Одним из таких факторов при применении алгоритмов, использующих скользящий режим, является частота дискретизации управляющего компьютера, на котором выполняются вычисления как для управления, так и для визуализации полета. Уменьшение частоты дискретизации может привести к ухудшению качества управления из-за эффекта размазывания сигнала, когда возникает потеря информации о быстро меняющихся состояниях системы, затрудняющая принятие корректных решений по управлению.

Современные системы визуализации требуют значительной вычислительной мощности, что делает актуальной задачу оптимизации потребления ресурсов (например, увеличение частоты визуализации методом фовеального рендеринга, упомянутым ранее). Таким образом, различные подзадачи, составляющие решение задачи синхронизации визуальной и динамической имитации, взаимно дополняют и подкрепляют друг друга. Однако необходимо тщательно сбалансировать частоту дискретизации и вычислительные ресурсы, чтобы обеспечить стабильность и высокое качество управления.

### 3.4.3 Робастное управление с использованием интегрального скользящего режима

В управлении, описанном в предыдущем разделе, полное знание параметров системы может рассматриваться как потенциальное ограничение. В связи с этим, настоящий раздел нацелен на смягчение этого требования и представление робастной версии предложенного управления, основанного на усредненном субградиенте.

В данном случае, матрица  $A(q)$  из уравнений (3.21) не известна точно:

$$A(q) = A^N(q) + A^U(q),$$

где  $A^N(q)$  обозначает известную часть матрицы, а  $A^U(q)$  представляет собой неопределенности. Отметим, что  $A^N(q)$  всегда обратима, учитывая механическую природу матрицы инерции. Учитывая свойства данной матрицы, можно отметить, что матрица неопределенности  $A^U(q)$  ограничена с точки зрения выбранной нормы, то есть:

$$\|A^U(q)\|_F \leq A^{U+}, \quad A^{U+} \in \mathbb{R}_+. \quad (3.31)$$

Воспользовавшись матричным тождеством Вудбури [42], можно показать, что обратная к  $A(q)$  матрица также определена неточно:

$$A^{-1}(q) = (A^N(q))^{-1} - A^{IU}(q), \quad (3.32)$$

где

$$A^{IU}(q) = (A^N(q))^{-1} \left( (A^U(q))^{-1} + (A^N(q))^{-1} \right)^{-1} (A^N(q))^{-1}.$$

Обратим внимание на определения  $G_1$  и  $G_2$ , представленные в (3.26). В этом случае будут верны следующие тождества:

$$\begin{aligned} G_1(\dot{q}, q) &= G_1^N(\dot{q}, q) + G_1^U(\dot{q}, q), \\ G_1^N(\dot{q}, q) &= -A_3(q) \left( \frac{d}{dt} (A^N(q))^{-1} - (A^N(q))^{-1} \cdot \right. \\ &\quad \cdot C(\dot{q}, q) (A^N(q))^{-1} \Big) - A_2(\dot{q}, q) (A^N(q))^{-1}, \\ G_1^U(\dot{q}, q) &= A_3(q) \left( \frac{d}{dt} (A^{IU}(q))^{-1} - (A^{IU}(q))^{-1} \cdot \right. \\ &\quad \cdot C(\dot{q}, q) (A^{IU}(q))^{-1} \Big) + A_2(\dot{q}, q) (A^{IU}(q))^{-1}, \end{aligned} \quad (3.33)$$

$$\begin{aligned} G_2(q) &= G_2^N(q) + G_2^U(q), \\ G_2^N(q) &= -A_3(q) (A^N(q))^{-1}, \\ G_2^U(q) &= A_3(q) A^{IU}(q). \end{aligned} \quad (3.34)$$

Учитывая ограниченность  $A_2$  и  $A_3$  (которые происходят из свойств прямой кинематики), а также их производных, можно ввести следующие ограничения:

$$\|G_i^U\| \leq G_i^+, \quad \left\| \frac{d}{dt} G_i^U \right\|_F \leq dG_i^+, \quad i = 1, 2. \quad (3.35)$$

Используя описание  $G_2(q)$  из (3.33), можно определить временную эволюцию  $\delta_t$  с помощью следующего соотношения:

$$\begin{aligned} \dot{\delta}_t &= \tilde{F} + (G_1^N(\dot{q}, q) + G_1^U(\dot{q}, q)) u_t + \\ &\quad + (G_2^N(\dot{q}, q) + G_2^U(\dot{q}, q)) \dot{u}_t. \end{aligned} \quad (3.36)$$

Теперь рассмотрим производную ошибки имитации ускорения с известными и неизвестными частями системы, представленными  $\tilde{F}_t$ ,  $G_1$  и  $G_2$ , удовлетворяющими (3.33) и (3.34). Управление  $u$  зададим в следующей форме:

$$\ddot{u}_t = \begin{cases} [G_2^N]^* \tau^c & \text{если } \|\delta\| = 0, \\ [G_2^N]^* \tau & \text{если } \|\delta\| > 0, \end{cases}$$

$$\tau = -k_t \text{Sign}(s) - p_{known}, \quad (3.37)$$

$$\begin{aligned} \tau^c &= -p_{known}, \\ p_{known} &= \left( \frac{d}{dt} G_1^N + \frac{G_1^N}{t+\theta} \right) u_t + \left( G_1^N + \frac{d}{dt} G_2^N(q) + \frac{G_2^N(q)}{t+\theta} \right) \dot{u}_t - \\ &\quad - \frac{\delta_t + \eta}{(t+\theta)^2} + \frac{\dot{\delta}_t + a(\delta(\Delta_t))}{(t+\theta)} - \frac{1}{(t+\theta)^2} \int_0^t a(\delta(t_s)) dt_s. \end{aligned}$$

**Теорема 2.** Пусть управление системой (3.22) представлено в виде (3.37) со скользящей переменной (3.17) и с известными матрицами (3.33) и (3.34). При этом желаемое ускорение кинематически реализуемо и  $\exists f^+ \equiv \text{const}$  такая, что для  $\forall t > 0$  выполнено условие  $\left\| \frac{d}{dt} \tilde{F} \right\| \leq f^+$ .

Тогда если выбрать коэффициент усиления  $k_t$  следующим образом:

$$\begin{aligned} k_t &= c_0 + c_1 \|u_t\| + c_2 \|\dot{u}_t\| + c_3, \quad c_3 > 0, \\ c_0 &= \sigma f^+ (1 + G_1^+ + G_2^+) + \\ &\quad (G_1^+ + G_2^+) \|p_{known}\|, \\ c_1 &= 2\sigma \left( dG_1^+ + \frac{G_1^+}{\theta} \right) (1 + G_1^+ + G_2^+) + \\ &\quad (G_1^+ + G_2^+) \|p_{known}\|, \\ c_2 &= 3\sigma \left( G_1^+ + dG_2^+ + \frac{G_2^+}{\theta} \right) (1 + G_1^+ + G_2^+) + \\ &\quad (G_1^+ + G_2^+) \|p_{known}\|, \quad \sigma > 5, \end{aligned} \quad (3.38)$$

будет существовать  $t_{reach} \geq 0$  такое, что  $s(t) \equiv 0 \quad \forall t \geq t_{reach}$ .

Кроме того,  $\forall \beta > \max\{\epsilon_1, \epsilon_2, \epsilon_3\} \quad \exists t_{conv} > t_{reach}$  такое, что

$$\|\Delta(t)\| \leq \beta \quad \forall t \geq t_{conv}.$$

### Доказательство

Здесь, как и ранее, для упрощения записи выкладок в доказательстве будет применяться нижний индекс  $_t$ , обозначающий зависимость от времени. Рассмотрим производную  $\delta_t$ , представленную в (3.36), и определение "слоя скольжения", введенное в (3.17). Производная  $s_t$  удовлетворяет следующему соотношению:

$$\dot{s}_t = \ddot{\Delta}_t + \left( -\frac{\delta_t + \eta}{(t+\theta)^2} + \frac{\dot{\delta}_t + a(\delta_t)}{(t+\theta)} \right) - \\ - \frac{1}{(t+\theta)^2} \int_{t_s=0}^t a(\delta_{t_s}) dt_s.$$

где

$$\ddot{\Delta}_t = \frac{d}{dt} \tilde{F}_t + \left( \frac{d}{dt} G_{1,t} \right) u_t + \\ + \left( G_{1,t} + \left( \frac{d}{dt} G_{2,t} \right) \right) \dot{u}_t + G_{2,t} \ddot{u}_t.$$

Теперь, используя представление  $G_{2,t}$  из (3.33), получим:

$$\dot{s}_t = p_{unknown} + p_{known} + G_{2,t}^N \ddot{u}. \quad (3.39)$$

где

$$p_{unknown} = \frac{d}{dt} \tilde{F}_t + \frac{\tilde{F}_t}{t+\theta} + \left( \frac{d}{dt} G_{1,t}^U + \frac{G_{1,t}^U}{t+\theta} \right) u + \\ + \left( G_{1,t}^U + \frac{d}{dt} G_{2,t}^U(q) + \frac{G_{2,t}^U(q)}{t+\theta} \right) \dot{u} + \left( G_{2,t}^U(q) \right) \ddot{u}. \quad (3.40)$$

Учитывая границы  $\tilde{F}_t$ ,  $G_{1,t}^U$  и  $G_{2,t}^U$  и их производные, обратим внимание, что норма вектора  $p_{unknown}$  удовлетворяет следующему неравенству:

$$\|p_{unknown}\| \leq a_0 + a_1 \|u\| + a_2 \|\dot{u}\| + \\ + \|(G_{2,t}^U(q)) \ddot{u}\|. \quad (3.41)$$

где  $a_0 = \sigma(f_0 + f_1)$ ,  $a_1 = 2\sigma \left( dG_1^+ + \frac{G_1^+}{\theta} \right)$ ,  $a_2 = 3\sigma \left( G_1^+ + dG_2^+ + \frac{G_2^+}{\theta} \right)$  и  $\sigma > 5$ .

В соответствии управлением представленным в (3.37), справедливо следующее ограничение для  $\|(G_{2,t}^U(q)) \ddot{u}\|$ :

$$\|(G_{2,t}^U(q)) \ddot{u}\| \leq b_0 + b_1 \|u\| + b_2 \|\dot{u}\|. \quad (3.42)$$

где  $b_0 = (\sigma(f_0 + f_1) + \|p_{known,t}\|) (G_1^+ + G_2^+)$ ,  
 $b_1 = \left( 2\sigma \left( dG_1^+ + \frac{G_1^+}{\theta} \right) + \|p_{known,t}\| \right) (G_1^+ + G_1^+)$  и  
 $b_2 = \left( 3\sigma \left( G_1^+ + dG_2^+ + \frac{G_2^+}{\theta} \right) + \|p_{known,t}\| \right) (G_1^+ + G_1^+).$

Таким образом, норма  $\|p_{unknown,t}\|$  ограничена следующим выражением:

$$\|p_{unknown,t}\| \leq c_0 + c_1 \|u_t\| + c_2 \|\dot{u}_t\|. \quad (3.43)$$

где  $c_i = a_i + b_i > 0$ ,  $i = 0, 1, 2$ .



Пусть  $\|\delta\| > 0$ . Для продолжения доказательства предлагается кандидат на функцию Ляпунова, зависящий от  $s_t$ :

$$V = \frac{1}{2} \|s_t\|^2$$

Рассмотрим изменение  $s_t$  в контексте динамики функции  $V$ . Получаем следующее дифференциальное уравнение:

$$\dot{V} = s_t^\top \dot{s}_t = s_t^\top (p_{unknown} + p_{known} + G_{2,t}^N \ddot{u}).$$

Учитывая вид управления из (3.37), производная функции  $V$  удовлетворяет следующему соотношению:

$$\begin{aligned} \dot{V} &= s_t^\top (p_{unknown,t} - k_t \text{Sign}(s_t)) \leq \\ &\leq \|s_t\| \|p_{unknown,t}\| - k_t \sum_{i=1}^3 |s_{i,t}|. \end{aligned}$$

Правая часть производной по времени для кандидата на функцию Ляпунова имеет следующую верхнюю границу:

$$\dot{V} \leq (\|p_{unknown,t}\| - k_t) \|s_t\|.$$

Возьмем коэффициент усиления  $k_t$  следующим образом:

$$k_t = c_0 + c_1 \|u_t\| + c_2 \|\dot{u}_t\| + c_3.$$

Получим следующее неравенство:

$$\dot{V} \leq -c_3 \sqrt{2V}.$$

Интегрирование неравенства дает:

$$\sqrt{V_t} \leq \sqrt{V_{t_0}} - \frac{c_3 (t - t_0)}{\sqrt{2}}$$

Опираясь на тот же технический анализ, что и в предыдущей теореме, получаем сходимость функции  $V_t$  за конечное время, то есть:

$$V_t \equiv 0 \quad \forall t \geq t_{reach}.$$

Доказательство остальной части утверждения полностью повторяет предыдущую теорему.  $\square$

Доказательство вышеуказанной теоремы успешно подтверждает теоретическую возможность создания системы управления, которая решает одну из ключевых подзадач синхронизации динамической и визуальной составляющих имитации (а именно, задачу динамической имитации с точностью до порогов чувствительности отолитового органа вестибулярного аппарата).

Однако стоит обратить внимание на то, что реализация предложенного алгоритма предполагает перемещение системы по всем трем линейным степеням свободы, а также достаточно широкий диапазон этих перемещений для обеспечения высококачественной имитации. В связи с этим, стенд панорамной системы виртуальной реальности МГУ, рассмотренный в первой главе, не считается подходящим для внедрения данного подхода, что вынуждает искать альтернативные решения.

Как один из возможных вариантов, рассматривается стенд на базе робота-манипулятора. Эта концепция была упомянута ранее и будет детально разобрана в последующей главе. Использование робота-манипулятора позволит преодолеть ограничения, связанные с перемещением системы и обеспечит более точную и реалистичную имитацию.

## Глава 4. Реализация алгоритмов синхронизации динамической и визуальной составляющих имитации для стенда на базе многозвеного робота-манипулятора

<sup>1</sup>Представим модель подвижного стенда, достаточного для реализации приведенного в предыдущей главе алгоритма управления. В первую очередь, определим механическую конфигурацию стенда, выбрав между разработкой уникальной конструкции или адаптацией существующей стандартной конструкции для решения поставленной задачи. Учитывая ограничения трехступенной платформы, упомянутой в первой главе, исследуем возможность использования промышленного робота-манипулятора в качестве основы для создания стенда.

Благодаря развитию компактных технологий виртуальной реальности и доступности промышленных роботов-манипуляторов с высокой грузоподъемностью (более 1 тонны), стало возможным создание стендов на их базе [3; 4]. В данной главе будет проведена численная моделирование алгоритмов динамической имитации для системы с динамическими параметрами, аналогичными многозвенному роботу-манипулятору Fanuc m2000iA 900L (рис. 4.1). Определим рабочую область стенда и численно смоделируем работу алгоритма управления, описанного в предыдущей главе.

### 4.1 Модель подвижного стенда

Рассмотрим модель робота-манипулятора, состоящую из многозвенной системы с кабиной, расположенной на рабочем органе. Для геометрического описания движения робота необходимо определить координаты, скорости и ускорения одной или нескольких расчетных точек кабины манипулятора относительно неподвижной системы отсчета. На рис. 4.2 изображена структура робота-манипулятора. Точка  $M$  обозначает центр подвеса (крепления) кабины. Для упрощения модели примем размеры кабины несущественными, а чувствительные массы пилота располагаются в точке  $M$ . Также рассмот-

---

<sup>1</sup>При подготовке данного раздела диссертации использованы следующие публикации, выполненные автором лично или в соавторстве, в которых, согласно Положению о присуждении ученых степеней в МГУ, отражены основные результаты, положения и выводы исследования: [9; 17; 25; 70; 79]



Рисунок 4.1 — Fanuc m2000iA 900L

рим  $O_1$  и  $O_2$  — точки, расположенные на осях вращения цилиндрических шарниров манипулятора для первого и второго звеньев соответственно.

Аналогично главе 3, обозначим неподвижную систему координат, связанную с комнатой, как  $O_{\text{init}}\xi\eta\zeta$ . Введем также несколько подвижных систем координат, связанных с каждым из подвижных элементов (рис. 4.2). Особый интерес представляет система координат  $O_1xyz$ , связанная с плоскостью движения манипулятора, где  $y$  сонаправлен с  $\eta$ .

Будем считать, что кабина, закрепленная в точке  $M$ , вращается вокруг своего центра масс, а моменты инерции кабины незначительны и не будут учитываться при управлении первыми двумя звеньями. В первом приближении считаем задачу управления кабиной решенной и не рассматриваем её.

Введем параметры устройства, описывающие его физические свойства:  $m_i$ ,  $i = 1, 2$  — массы первого и второго (вместе с креслом) звена,

$l_i$ ,  $i = 1, 2$  — длины звеньев,

$s_i$ ,  $i = 1, 2$  — расстояния от оснований звеньев до их центров масс,

$d_i$ ,  $i = 1, 2$  — расстояния от центров масс до следующих по конструкции соседних соединений,

Можно заметить, что  $l_i = s_i + d_i$ .

$I_0$  — момент инерции основания вокруг вертикальной оси вращения,  
 $I_i, i = 1, 2, 3$  — главные моменты инерции 1го звена,  
 $I_i, i = 1, 2, 3$  — главные моменты инерции 2го звена.

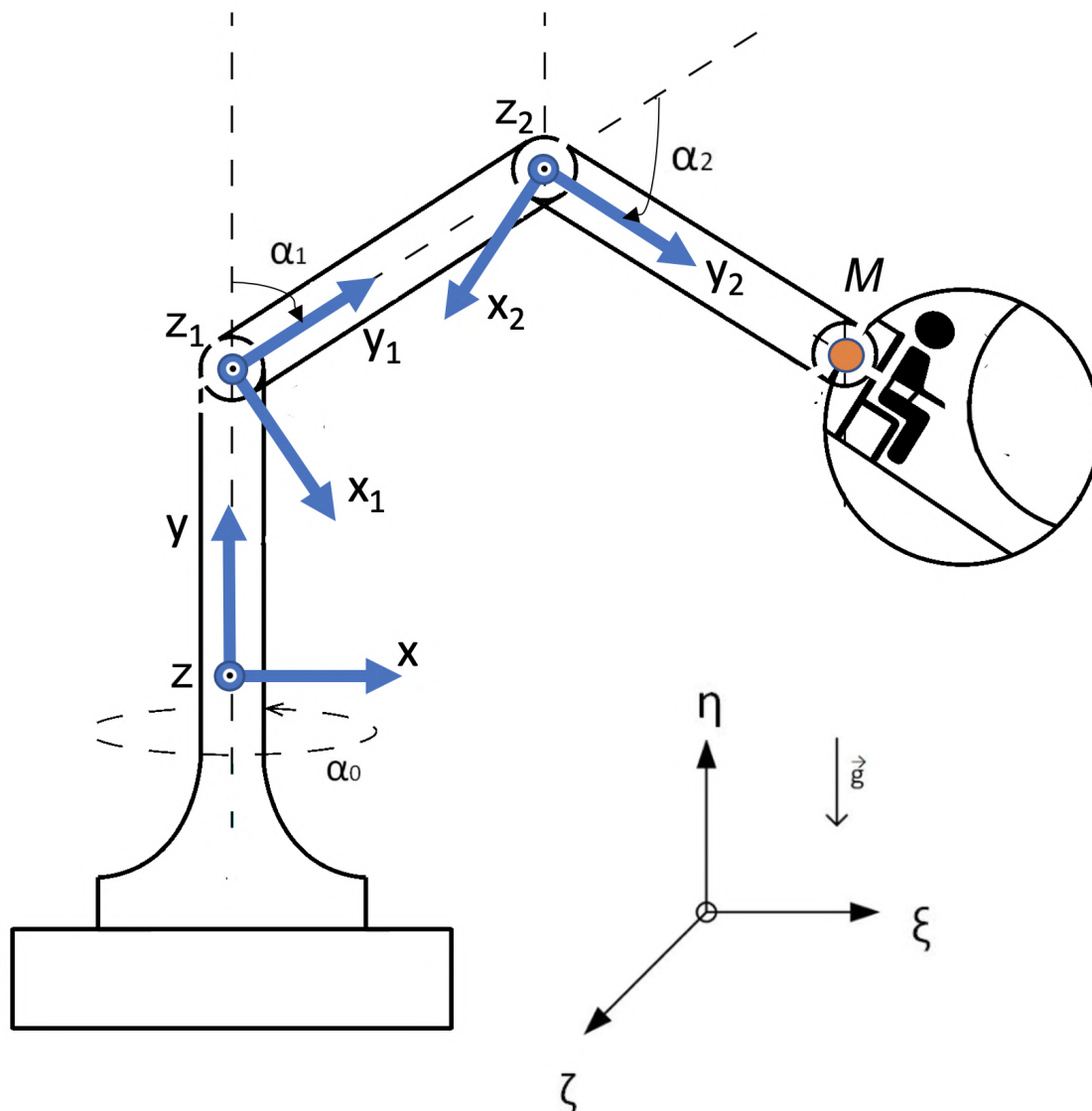


Рисунок 4.2 — Структура стенда на базе руки-манипулятора

Введем обобщенные координаты:

$$q = \begin{bmatrix} \alpha_0 \\ \alpha_1 \\ \alpha_2 \end{bmatrix}.$$

### 4.1.1 Зависимость ускорений расчетной точки от перемещений звеньев манипулятора

Для разработки алгоритма управления роботом-манипулятором необходимо знать координаты, скорости и ускорения расчетной точки манипулятора относительно неподвижной системы отсчета. Нашей целью является определение соотношений, связывающих ускорение расчетной точки станда с движением в пространстве обобщенных координат.

Рассмотрим структуру робота, представленную на рисунке 4.2.

Для определения ускорения точки крепления кабины  $w_M$ , выразим его через компоненты ускорения, связанные с поворотами и движением звеньев манипулятора. Здесь  $k_i$  обозначает соответствующие оси:

$$\begin{aligned} w_M = w_{abs}(\ddot{q}, \dot{q}, q) = & l_1(-\ddot{\alpha}_0 k_{-z} + \alpha_0 \dot{\alpha}_0 k_x) - \dot{\alpha}_0^2 l_{O_1} k_y + l_1(\ddot{\alpha}_1 k_{x_0} + \alpha_1 \dot{\alpha}_1 k_{y_0}) - \\ & - \dot{\alpha}_1^2 l_1 k_{z_0} + l_2(\ddot{\alpha}_2 k_{x_1} + \alpha_2 \dot{\alpha}_2 k_{y_1}) - \dot{\alpha}_2^2 l_2 k_{z_1}, \\ [k_{x_1} \ k_{y_1} \ k_{z_1}] = & R_1[k_{x_0} \ k_{y_0} \ k_{z_0}] = R_1 R_0[k_x \ k_y \ k_z], \\ [k_{x_0} \ k_{y_0} \ k_{z_0}] = & R_0[k_x \ k_y \ k_z], \\ [k_x \ k_y \ k_z] = & R[k_{x_{init}} \ k_{y_{init}} \ k_{z_{init}}]. \end{aligned} \quad (4.1)$$

Матрица поворота  $R$  определяет поворот вокруг вертикальной оси ( $y$ -оси) на угол  $\alpha_0$ :

$$R = \begin{pmatrix} \cos(\alpha_0) & 0 & -\sin(\alpha_0) \\ 0 & 1 & 0 \\ \sin(\alpha_0) & 0 & \cos(\alpha_0) \end{pmatrix}.$$

Матрица поворота  $R_0$  определяет поворот вокруг  $z$ -оси на угол  $\alpha_1$ :

$$R_0 = \begin{pmatrix} \cos(\alpha_1) & -\sin(\alpha_1) & 0 \\ \sin(\alpha_1) & \cos(\alpha_1) & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

Матрица поворота  $R_1$  определяет поворот вокруг  $z$ -оси на угол  $\alpha_2 - \alpha_1$ :

$$R_1 = \begin{pmatrix} \cos(\alpha_2 - \alpha_1) & -\sin(\alpha_2 - \alpha_1) & 0 \\ \sin(\alpha_2 - \alpha_1) & \cos(\alpha_2 - \alpha_1) & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

Таким образом, мы выразили ускорение точки крепления кабины  $w_{abs}$  через компоненты ускорения, связанные с поворотами и движением звеньев манипулятора.

Производную ускорения центра кабины можно представить следующим образом:

$$\begin{aligned} \frac{d}{dt}w_{abs}(\ddot{q},\dot{q},q) &= \frac{\partial w_{abs}(\ddot{q},\dot{q},q)}{\partial \ddot{q}}\ddot{\dot{q}} + \\ &+ \frac{\partial w_{abs}(\ddot{q},\dot{q},q)}{\partial \dot{q}}\ddot{q} + \frac{\partial w_{abs}(\ddot{q},\dot{q},q)}{\partial q}\dot{q} := \\ &:= A_3(q)\ddot{\dot{q}} + A_2(\dot{q},q)\ddot{q} + A_1(\dot{q},q)\dot{q}. \end{aligned} \quad (4.2)$$

#### 4.1.2 Динамическая модель робота-манипулятора

Для вывода уравнений движения системы был использован метод Лагранжа второго рода. Их можно представить в следующем виде:

$$\ddot{q} = A^{-1}(q)(u - C(q,\dot{q})\dot{q} - G(q)), \quad (4.3)$$

где  $u$  — векторное управление системой (моменты сил в двигателях).

Ненулевые элементы матрицы  $A(q) \in \mathbb{R}^{3 \times 3}$  определяются следующим образом:

$$\begin{aligned} a_{11} &= I_0 + I_2 \cos^2 \alpha_1 + \Pi_2 \cos^2(\alpha_1 + \alpha_2) + I_1 \sin^2 \alpha_1 + (s_1^2 m_1 + d_1^2 m_1) \sin^2 \alpha_1 + \\ &+ 2s_2 d_1 m_2 \sin \alpha_1 \sin(\alpha_1 + \alpha_2) + \Pi_1 \sin^2(\alpha_1 + \alpha_2) + s_2^2 m_2 \sin^2(\alpha_1 + \alpha_2), \\ a_{22} &= I_3 + \Pi_3 + s_1^2 m_1 + s_2^2 m_2 + d_1^2 m_2 + 2s_2 d_1 m_2 \cos \alpha_2, \\ a_{33} &= \Pi_3 + s_2^2 m_2, \\ a_{23} &= a_{32} = \Pi_3 + s_2^2 m_2 + s_2 d_1 m_2 \cos \alpha_2. \end{aligned}$$

Ненулевые элементы матрицы  $C(\dot{q},q) \in \mathbb{R}^{3 \times 3}$ :

$$\begin{aligned} c_{11} &= ((I_1 - I_2 + s_1^2 m_1 + d_1^2 m_2) \sin(2\alpha_1) + (\Pi_1 - \Pi_2 + s_2^2 m_2) \sin(2(\alpha_1 + \alpha_2))), \\ c_{12} &= 2s_2 d_1 m_2 \sin(2(\alpha_1 + \alpha_2)), \\ c_{13} &= 2(s_2 d_1 m_2 \sin \alpha_1 + (\Pi_1 - \Pi_2 + s_2^2 m_1) \sin(\alpha_1 + \alpha_2)) \cos(\alpha_1 + \alpha_2), \\ c_{21} &= -\dot{\alpha}_0(I_1 - I_2 + s_1^2 m_1 + d_1^2 m_2) \sin(2\alpha_1) + (\Pi_1 - \Pi_2 + s_2^2 m_2) \sin(2(\alpha_1 + \alpha_2)) + \\ &+ 2s_2 d_1 m_2 \sin(2(\alpha_1 + \alpha_2))/2, \\ c_{22} &= 2\dot{\alpha}_2 s_2 d_1 m_2 \sin \alpha_2, \end{aligned}$$

$$c_{23} = \dot{\alpha}_2 s_2 d_1 m_2 \sin \alpha_2.$$

Ненулевые элементы вектора  $G(q) \in \mathbb{R}^{1 \times 3}$ :

$$g_1 = -g(s_1 m_1 + d_1 m_2) \sin \alpha_1 + s_2 m_2 \sin(\alpha_1 + \alpha_2),$$

$$g_2 = -g s_2 m_2 \sin(\alpha_1 + \alpha_2).$$

Как было показано в предыдущей главе, система, представленная в (4.3), может быть преобразована следующим образом:

$$\ddot{q} = \Phi_a(\dot{q}, q) + \Phi_b(q)u. \quad (4.4)$$

$$\Phi_a(\dot{q}, q) = -A^{-1}(q) (C(q, \dot{q})\dot{q} + G(q)) \in \mathbb{R}^{6 \times 6}, \quad \Phi_b(q) = A^{-1}(q) \in \mathbb{R}^{6 \times 6}.$$

Следует отметить, что правая часть уравнения (4.4) может быть определена с погрешностью из-за наличия возмущений, таких как ошибки в определении массовых характеристик стенда, неточная реализация моментов двигателей и влияние движения кабины на динамику.

Далее рассмотрим ускорение в пространстве обобщенных координат:

$$\ddot{q} = F_a(\ddot{q}, \dot{q}, q) + F_b(\dot{q}, q)u + \Phi_b(q)\dot{u}, \quad (4.5)$$

где матрицы  $F_a \in \mathbb{R}^{6 \times 6}$ ,  $F_b \in \mathbb{R}^{6 \times 6}$ :

$$F_a(\ddot{q}, \dot{q}, q) = -\frac{d}{dt}A^{-1}(q) (C(q, \dot{q})\dot{q} + G(q)) -$$

$$-A^{-1}(q) \left( \frac{d}{dt}C(q, \dot{q})\dot{q} + C(q, \dot{q})\Phi_a(\dot{q}, q) + \frac{d}{dt}G(q)\dot{q} \right),$$

$$F_b(\dot{q}, q) = \frac{d}{dt}A^{-1}(q) - A^{-1}(q)C(q, \dot{q})A^{-1}(q).$$

Таким образом, были определены матрицы, необходимые для реализации алгоритма управления с использованием интегрального скользящего режима.

### 4.1.3 Определение ограничений на перемещение расчетной точки в вертикальной плоскости

Для практического применения алгоритмов управления стендом, необходимо определить множество допустимых положений и рабочую область стенда. Определение множества допустимых положений стенда на основе робота-манипулятора требует учета геометрических ограничений на перемещение звеньев.



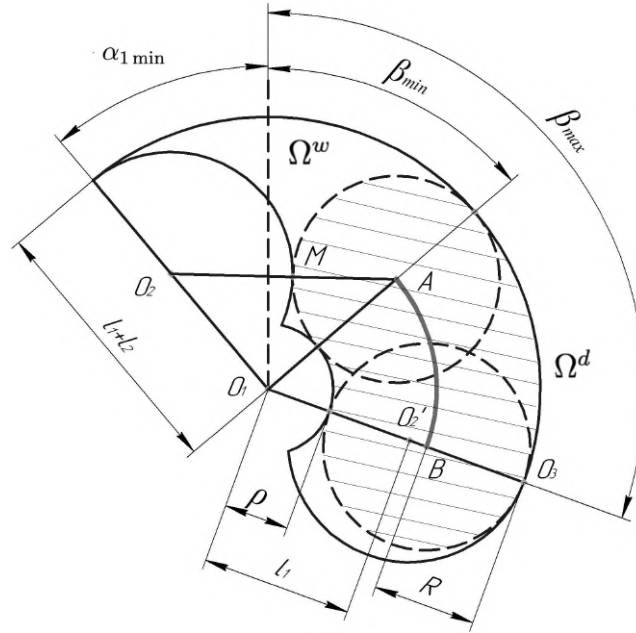


Рисунок 4.3 — Множество допустимых положений

Для удобства вычисления множества допустимых положений стенда введем вспомогательные величины  $l_1 = s_1 + d_1$  и  $l_2 = s_2 + d_2$  — длины первого и второго звеньев соответственно.

При движении стенда в вертикальной плоскости координаты расчетной точки стенда  $M$  определяются следующим образом:

$$\begin{cases} x = l_1 \sin \alpha_1 + l_2 \sin(\alpha_1 + \alpha_2), \\ y = l_1 \cos \alpha_1 + l_2 \cos(\alpha_1 + \alpha_2). \end{cases} \quad (4.6)$$

На рис. 4.3 изображено множество  $\Omega^w$  — геометрически допустимых положений точки  $M$  в вертикальной плоскости. Контур множества  $\Omega^w$  состоит из дуг нескольких окружностей. Радиус большей дуги с центром в точке  $O_1$  равен  $l_1 + l_2$ , а радиус меньшей  $\rho$  может быть найден по формуле:

$$\rho = \sqrt{l_1^2 + l_2^2 - 2l_1l_2 \cos \alpha_{2\max}}. \quad (4.7)$$

Множество  $\Omega^w$  имеет несимметричную форму. Для решения задачи динамической имитации предпочтительно иметь область, в которой возможно равномерное движение с максимальным ускорением во всех направлениях от определенного стартового положения. В множество  $\Omega^w$  можно вписать семейство окружностей радиуса  $R = l_1 + l_2 - \rho$ , центры которых образуют дугу  $\overset{\frown}{AB}$ . Эти окружности формируют область  $\Omega^d$  (заштрихована на рис. 4.3), которая вложена в  $\Omega^w$ . Назовем дугу  $AB$  "стартовой". Координаты  $a$  и

$b$  дуги  $\overset{\circ}{AB}$  можно записать в виде:

$$(a, b) = \{(R + \rho) \sin \beta, (R + \rho) \cos \beta\}. \quad (4.8)$$

где  $\beta \in [\beta_{\min}, \beta_{\max}]$ , а параметры  $\beta_{\min}$  и  $\beta_{\max}$  подлежат определению.

Рассмотрим на рис. 4.3 треугольник  $O_1O_2A$ :  $O_1O_2 = l_1$ ,  $O_1A = \rho + R$ ,  $O_2A = l_2 + R$ .  $O_2M = l_2$  и  $MA = R$  (радиусы  $O_2M$  и  $MA$  перпендикулярны касательной к соответствующим окружностям в точке  $M$  и образуют отрезок  $O_2A$ ). Используя теорему косинусов, получаем:

$$\cos(\angle O_2O_1A) = \cos(\alpha_{1\min} + \beta_{\min}) = \frac{l_1^2 + (\rho + R)^2 - (l_2 + R)^2}{2l_1(\rho + R)}.$$

Отсюда выражаем:

$$\beta_{\min} = \arccos \frac{\rho^2 + l_1^2 - l_2^2 + 2R(\rho - l_2)}{2l_1(\rho + R)} - \alpha_{1\min}. \quad (4.9)$$

Докажем, что  $\beta_{\max} = \alpha_{1\max}$ . Рассмотрим нижнюю дугу множества  $\Omega^w$  с центром в  $O'_2$  и радиусом  $l_2$ . Определим окружность с центром в точке  $B$  и радиусом  $R$ , удовлетворяющую условию  $2R > l_2$ . В случае  $\beta > \alpha_{1\max}$ , точки окружности с центром в  $B$  и радиусом  $R$  выходят за границы множества  $\Omega^w$ .

Пусть  $\overset{\circ}{AB}$  обозначает "стартовую" дугу, с которой начинается и на которой заканчивается процесс имитации. Радиус этой дуги определяется как сумма  $\rho + R$ , где  $R$  — радиус наибольшей окружности, полностью лежащей внутри  $\Omega^d$ , и центр которой принадлежит дуге  $\overset{\circ}{AB}$ ;  $\rho$  — радиус дуги, образующей левую границу  $\Omega^d$ . Значения  $R$  и  $\rho$  зависят от параметров робота-манипулятора и могут быть вычислены по следующим формулам:

$$\rho = \sqrt{l_1^2 + l_2^2 - 2l_1l_2 \cos \alpha_{2\max}},$$

$$R = (l_1 + l_2 - \rho)/2,$$

где  $\alpha_{2\max}$  — наибольшее значение угла  $\alpha_2$  для второго звена робота-манипулятора.

Минимальный и максимальный углы  $\beta_{\min}$  и  $\beta_{\max}$  могут быть найдены с использованием следующих формул:

$$\beta_{\min} = \arccos \frac{\rho^2 + l_1^2 - l_2^2 + 2R(\rho - l_2)}{2l_1(\rho + R)} - \alpha_{1\min}; \quad \beta_{\max} = \alpha_{1\max},$$

где  $\alpha_{1\min}$  и  $\alpha_{1\max}$  — минимальное и максимальное значения для угла первого звена робота-манипулятора.

В результате мы получаем множество допустимых положений для расчетной точки кабины стенда, в пределах которого будет проводиться имитация перегрузки.

#### 4.1.4 Алгоритм управления остановкой на границе множества допустимых положений и возврата на стартовую дугу

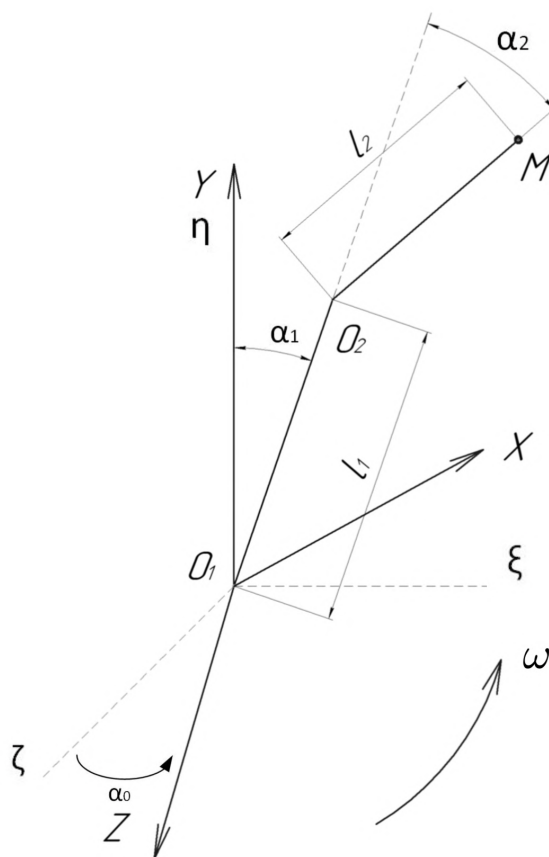


Рисунок 4.4 — Модель стенда с учетом вращения вокруг вертикали

Опираясь на результаты предыдущего раздела, определим область в фазовом пространстве, из которой возможно выполнить "плавную" остановку на границе множества допустимых положений (то есть с ускорением ниже порога чувствительности вестибулярного аппарата) с последующим возвратом на стартовую дугу.

Для решения данной задачи построим модель движения расчетной точки манипулятора, учитывающую вращение исполняющего механизма манипулятора вокруг вертикальной оси с известной угловой скоростью.

Рис. 4.4 представляет кинематическую схему манипулятора, вращающегося с постоянной угловой скоростью вокруг вертикальной оси. Во вращающейся системе координат возникают дополнительные ускорения: цен-

тробежное и кориолисово. Обозначим вектор вращения системы координат как  $\vec{\Omega} = \begin{pmatrix} 0 \\ \omega \\ 0 \end{pmatrix}$ . Тогда в расчетной точке  $M$  центробежное ускорение

$\vec{a}_c = \begin{pmatrix} \omega^2 x \\ 0 \\ 0 \end{pmatrix}$ . Кориолисово ускорение равно:

$$\vec{a}_k = 2 \begin{bmatrix} \vec{e}_x & \vec{e}_y & \vec{e}_z \\ \dot{x} & \dot{y} & \dot{z} \\ 0 & \omega & 0 \end{bmatrix} = \begin{pmatrix} -2\dot{z}\omega \\ 0 \\ 2\dot{x}\omega \end{pmatrix}.$$

Рассмотрим кинематическое управление  $u = (u_1, u_2)^\top$  расчетной точкой и запишем систему в следующем виде:

$$\begin{cases} \ddot{x} = u_1 + \omega^2 x - 2\dot{z}\omega, \\ \ddot{y} = u_2, \\ \ddot{z} = 2\dot{x}\omega. \end{cases} \quad (4.10)$$

Значения угловой скорости вращения манипулятора относительно малы, что позволяет пренебречь кориолисовым ускорением. Рассмотрим задачу остановки и возврата в некоторую начальную точку с величинами ускорения меньше порогов чувствительности вестибулярного аппарата  $\epsilon_1$  по оси  $x$  и  $\epsilon_2$  по оси  $y$ .

Так как процесс возврата является критическим для обеспечения качества имитации (имитация не производится во время остановки и возврата), строим алгоритм управления таким образом, чтобы минимизировать время возврата, обозначаемое как  $t_k$ . Тогда задача формулируется следующим образом:

$$\begin{cases} \ddot{x} = u_1 + \omega^2 x, \\ \ddot{y} = u_2, \\ J = t_k \rightarrow \min_{u_1, u_2}. \end{cases}$$

Начальные условия задачи определяются как:

$$x(0) = x_0, \quad \dot{x}(0) = v_1, \quad y(0) = y_0, \quad \dot{y}(0) = v_2.$$

Конечные условия задачи формулируются следующим образом:

$$x(t_k) = x_k, \quad y(t_k) = y_k, \quad \dot{x}(t_k) = 0, \quad \dot{y}(t_k) = 0.$$

Предполагаем, что  $(x_k, y_k) \in \check{A}B$ .

Воспользуемся вращением манипулятора для поддержания определенного значения модуля кажущегося ускорения равным  $\omega^2 x(t_k)$ . Особенность имитационных стендов заключается в том, что даже незначительные центробежные силы, несущественные для механических компонентов стенда, воспринимаются сенсорной системой человека, находящегося в тренажере. Можно ввести ограничения на управление, основываясь на необходимости остановки с ускорением ниже порога чувствительности вестибулярного аппарата. Однако ограничения на управление  $u_1$  будут несимметричными:

$$|u_1 + \omega^2 x(t_k)| \leq \epsilon_1, \quad |u_2| \leq \epsilon_2.$$

Здесь, как и ранее,  $\epsilon = (\epsilon_1, \epsilon_2, \epsilon_3)^\top$  представляет собой вектор порогов чувствительности отолитового органа вестибулярного аппарата. Считаем, что возникающее по тангенциальному направлению кориолисово ускорение меньше порога чувствительности  $\epsilon_3$ .

В результате, мы получаем задачу оптимального управления, которая заключается в минимизации времени возврата  $t_k$  с учетом ограничений на ускорения и управление, обеспечивая при этом сохранение комфорта для человека находящегося в тренажере. Данная задача может быть решена с использованием принципа максимума Понтрягина [105].

#### 4.1.5 Алгоритм торможения и возврата

Представим систему (4.10) в альтернативной форме:

$$\left\{ \begin{array}{l} \dot{x}_1 = x_2 \\ \dot{x}_2 = u_1 + \omega^2 x_1, \quad |u_1 + \omega^2 x_k| \leq \epsilon_1, \\ \dot{x}_3 = x_4 \\ \dot{x}_4 = u_2, \quad |u_2| \leq \epsilon_2, \\ \dot{x}_5 = 1, \\ J = x_5(t_k) \rightarrow \min_{u_1, u_2}. \end{array} \right. \quad (4.11)$$

Начальные условия:

$$x_1(0) = x_0, \quad x_2(0) = v_1, \quad x_3(0) = y_0, \quad x_4(0) = v_2.$$

Конечные условия:

$$x_1(t_k) = x_k, \quad x_2(t_k) = 0, \quad x_3(t_k) = y_k, \quad x_4(t_k) = 0.$$

Система 4.11 может быть разделена на две подсистемы, для которых задача решается отдельно. Для каждой подсистемы используем принцип максимума. Функция Понтрягина имеет вид:

$$H = \psi_1 x_2 + \psi_2(u_1 + \omega^2 x_1) + \psi_3 x_4 + \psi_4 u_2 + \psi_5.$$

Для максимальности  $H$  требуется выполнение следующих условий:

$$\begin{cases} u_1 = \epsilon_1 \text{sign}(\psi_2) - \omega^2 x_k, \\ u_2 = \epsilon_2 \text{sign}(\psi_4). \end{cases}$$

Сопряженная система представляет собой следующие уравнения:

$$\begin{cases} \dot{\psi}_1 = -\omega^2 \psi_2, \\ \dot{\psi}_2 = -\psi_1, \\ \dot{\psi}_3 = 0, \\ \dot{\psi}_4 = -\psi_3, \\ \dot{\psi}_5 = 0. \end{cases}$$

$\psi_2 = C_1 e^{\omega t} + C_2 e^{-\omega t}$  и  $\psi_4 = C_4 - C_3 t$  могут менять знак не более одного раза, поэтому переключений управления для  $u_1$  и  $u_2$  может быть не более одного. На фазовой плоскости  $x_1 O x_2$  траекториями являются гиперболы:

$$\begin{aligned} u_1 = \epsilon_1 + \omega^2 x_k : \quad & \begin{cases} x_1 = A_1 e^{\omega t} + A_2 e^{-\omega t} - \frac{\epsilon_1}{\omega^2} - x_k, \\ x_2 = \omega(A_1 e^{\omega t} - A_2 e^{-\omega t}), \end{cases} \\ u_1 = -\epsilon_1 + \omega^2 x_k : \quad & \begin{cases} x_1 = A_1 e^{\omega t} + A_2 e^{-\omega t} + \frac{\epsilon_1}{\omega^2} - x_k, \\ x_2 = \omega(A_1 e^{\omega t} - A_2 e^{-\omega t}). \end{cases} \end{aligned}$$

В настоящей постановке задачи  $x_1(t_k) = x_k > 0$ ,  $x_2(t_k) = 0$ . Очевидно в точку с такими координатами можно попасть не из любой другой точки на фазовой плоскости  $x_1 O x_2$ . Рисунок 4.5 иллюстрирует область управляемости на плоскости, представляющей собой полосу между двумя прямыми  $x_2 = -x_k - x_1 - \frac{\epsilon_1}{\omega^2}$  и  $x_2 = -x_k - x_1 + \frac{\epsilon_1}{\omega^2}$ . На рисунке также изображена траектория  $EF$ , проходящая через точку  $(x_k, 0)$ , которая является линией переключения управления. Синтез управления выполняется следующим образом:

1. для точек области управляемости, расположенных слева от линии переключения, управление положительно по знаку ( $u_1 = \epsilon_1 + \omega^2 x_k$ );
2. для точек области управляемости, расположенных справа от линии переключения, управление отрицательно по знаку ( $u_1 = -\epsilon_1 + \omega^2 x_k$ );
3. в момент переключения, система попадает на траекторию переключения  $EF$  и движется по ней.

Ширина области управляемости определяется величиной  $\epsilon_1$ . За пределами этой области траектории направлены прочь от нее, и невозможно подобрать управление, которое приведет в точку  $(x_k, 0)$ .

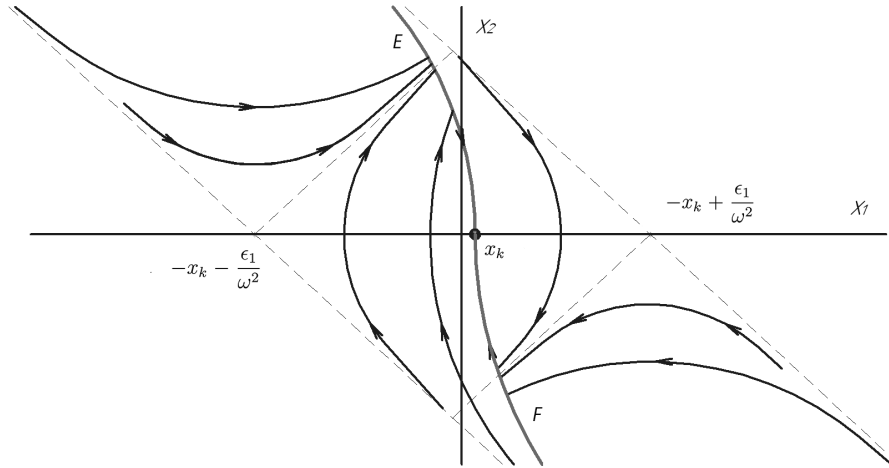


Рисунок 4.5 — Синтез управления в координатах  $x_1$  и  $x_2$

На фазовой плоскости  $x_3 O x_4$  траекториями являются параболы:

$$u_2 = \epsilon_2 : \begin{cases} x_3 = B_1 + B_2 t + \frac{\epsilon_2}{2} t^2, \\ x_4 = B_2 + \epsilon_2 t. \end{cases}$$

$$u_2 = -\epsilon_2 : \begin{cases} x_3 = B_1 + B_2 t - \frac{\epsilon_2}{2} t^2, \\ x_4 = B_2 - \epsilon_2 t. \end{cases}$$

Эти параболы изображены на рисунке 4.6, где  $CD$  — линия переключения управления  $u_2$ , которая также является траекторией системы. Синтез управления на этой плоскости аналогичен синтезу для подсистемы с координатами  $x_1, x_2$ :

1. для точек фазовой плоскости, лежащих слева от линии переключения, управление положительно по знаку ( $u_2 = \epsilon_2$ ).
2. для точек фазовой плоскости, лежащих справа от линии переключения, управление отрицательно по знаку ( $u_2 = -\epsilon_2$ ).

3. в момент переключения, система попадает на линию переключения  $CD$  и движется по ней.

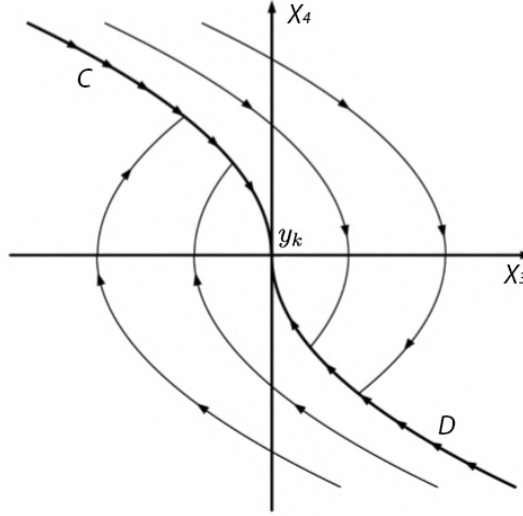


Рисунок 4.6 — Синтез управления в координатах  $x_3$  и  $x_4$

Обратим внимание на возможную разницу во времени достижения конечной точки по разным координатам. Формально, в этот момент оптимальная задача для координаты с меньшим временем достижения вырождается. Компенсация данного эффекта может быть достигнута путём введения более строгих ограничений на управление по координате, которая достигает конечной точки первой.

#### 4.1.6 Определение времени переключения управления

**Для управления по оси  $x$ .** Обозначим за  $u_+ = \epsilon_1 + \omega^2 x_k$  и  $u_- = \epsilon_1 - \omega^2 x_k$ .

Определим время переключения в случае, когда до момента переключения управление равно  $u_+$ , а после переключения —  $-u_-$ . Рассмотрим следующую систему:

$$\begin{cases} \dot{x}_1 = x_2, \\ \dot{x}_2 = \omega^2 x_1 + u_+, \end{cases}$$

с начальными условиями:  $x_1(0) = x_0$ ,  $x_2(0) = v_1$ . Решениями данной системы являются функции:

$$\begin{cases} x_1 = A_1 e^{\omega t} + A_2 e^{-\omega t} - \frac{u_+}{\omega^2}, \\ x_2 = \omega(A_1 e^{\omega t} - A_2 e^{-\omega t}). \end{cases}$$



Используя начальные условия, найдём константы:

$$\begin{cases} x_1(0) = x_0 = A_1 + A_2 - \frac{u_+}{\omega^2}, \\ x_2(0) = v_1 = \omega(A_1 - A_2). \end{cases}$$

Таким образом, получаем:

$$\begin{cases} A_1 = \frac{1}{2} \left( x_0 + \frac{v_1}{\omega} + \frac{u_+}{\omega^2} \right), \\ A_2 = \frac{1}{2} \left( x_0 - \frac{v_1}{\omega} + \frac{u_+}{\omega^2} \right). \end{cases}$$

Рассмотрим выражение координат для траектории после момента переключения:

$$\begin{cases} x_1 = C_1 e^{\omega(t-t_k)} + C_2 e^{-\omega(t-t_k)} + \frac{u_-}{\omega^2}, \\ x_2 = \omega(C_1 e^{\omega(t-t_k)} - C_2 e^{-\omega(t-t_k)}). \end{cases}$$

Траектория проходит через  $(x_k, 0)$ , что позволяет определить коэффициенты:

$$C_1 = C_2 = \frac{1}{2} \left( x_k - \frac{u_-}{\omega^2} \right).$$

В момент переключения управления, система описывается следующим образом:

$$\begin{cases} A_1 e^{\omega t_1} + A_2 e^{-\omega t_1} - \frac{u_+}{\omega^2} = C_1 e^{\omega(t_1-t_k)} + C_2 e^{-\omega(t_1-t_k)} + \frac{u_-}{\omega^2}, \\ A_1 e^{\omega t_1} - A_2 e^{-\omega t_1} = C_1 e^{\omega(t_1-t_k)} - C_2 e^{-\omega(t_1-t_k)}. \end{cases} \Rightarrow$$

$$\begin{cases} 2A_1 e^{\omega t_1} - \frac{u_+}{\omega^2} = 2C_1 e^{\omega(t_1-t_k)} + \frac{u_-}{\omega^2}, \\ 2A_2 e^{-\omega t_1} - \frac{u_+}{\omega^2} = 2C_2 e^{-\omega(t_1-t_k)} + \frac{u_-}{\omega^2}. \end{cases} \Rightarrow$$

$$\begin{cases} 2 \left( A_1 e^{\omega t_1} - C_1 e^{\omega(t_1-t_k)} \right) = \frac{u_- + u_+}{\omega^2}, \\ 2 \left( A_2 e^{-\omega t_1} - C_2 e^{-\omega(t_1-t_k)} \right) = \frac{u_- + u_+}{\omega^2}. \end{cases} \Rightarrow$$

$$\left( A_1 e^{\omega t_1} - C_1 e^{\omega(t_1-t_k)} \right) \left( A_2 e^{-\omega t_1} - C_2 e^{-\omega(t_1-t_k)} \right) = \frac{(u_- + u_+)^2}{\omega^4} \Rightarrow$$

$$A_1 A_2 + C_1 C_2 - A_2 C_1 e^{-\omega t_k} - A_1 C_2 e^{\omega t_k} = \frac{(u_- + u_+)^2}{4\omega^4}.$$

Обозначим  $p = e^{\omega t_k}$ , тогда, умножив на  $p$ , получаем уравнение:

$$A_1 C_2 p^2 + \left( \frac{(u_- + u_+)^2}{4\omega^4} - A_1 A_2 - C_1 C_2 \right) p + A_2 C_1 = 0,$$

Решением уравнения является:

$$p = \frac{A_1 A_2 + C_1 C_2 - \frac{(u_- + u_+)^2}{4\omega^4} + \sqrt{\left( \frac{(u_- + u_+)^2}{4\omega^4} - A_1 A_2 - C_1 C_2 \right)^2 - 4A_1 A_2 C_1 C_2}}{2A_1 C_2}.$$

Момент переключения выражается через  $p$  следующим образом:

$$t_1 = \frac{1}{\omega} \ln \left( \frac{u_- + u_+}{2\omega^2(A_1 - C_1/p)} \right).$$

Теперь найдем время переключения для случая, когда до момента переключения управление равно  $-u_-$ , а после переключения  $-u_+$ . Рассмотрим следующую систему:

$$\begin{cases} \dot{x}_1 = x_2, \\ \dot{x}_2 = \omega^2 x_1 - u_-. \end{cases}$$

Начальные условия:  $x_1(0) = x_0$ ,  $x_2(0) = v_1$ .

Решениями системы являются функции:

$$\begin{cases} x_1 = A_1 e^{\omega t} + A_2 e^{-\omega t} + \frac{u_-}{\omega^2}, \\ x_2 = \omega(A_1 e^{\omega t} - A_2 e^{-\omega t}). \end{cases}$$

Используя начальные условия, определим постоянные интегрирования:

$$\begin{cases} x_1(0) = x_0 = A_1 + A_2 + \frac{u_-}{\omega^2}, \\ x_2(0) = v_1 = \omega(A_1 - A_2). \end{cases}$$

Таким образом, получаем:

$$\begin{cases} A_1 = \frac{1}{2} \left( x_0 + \frac{v_1}{\omega} - \frac{u_-}{\omega^2} \right), \\ A_2 = \frac{1}{2} \left( x_0 - \frac{v_1}{\omega} - \frac{u_-}{\omega^2} \right). \end{cases}$$

Траектория после переключения управления соответствует  $u_+$  и её координаты записываются следующим образом:

$$\begin{cases} x_1 = C_1 e^{\omega t} + C_2 e^{-\omega t} - \frac{u_+}{\omega^2}, \\ x_2 = \omega(C_1 e^{\omega t} - C_2 e^{-\omega t}). \end{cases}$$

Такая траектория проходит через точку  $(x_k, 0)$ , откуда определяем коэффициенты:

$$C_1 = C_2 = \frac{1}{2} \left( x_k + \frac{u_+}{\omega^2} \right).$$

Далее аналогично предыдущему случаю вычислим время переключения. Введем переменную  $p = e^{\omega t_k}$  и запишем уравнение:

$$p = \frac{A_1 A_2 + C_1 C_2 - \frac{(u_- + u_+)^2}{4\omega^4} + \sqrt{\left( \frac{(u_- + u_+)^2}{4\omega^4} - A_1 A_2 - C_1 C_2 \right)^2 - 4A_1 A_2 C_1 C_2}}{2A_1 C_2}.$$

Теперь, используя значение  $p$ , можно выразить время переключения управления:

$$t_1 = \frac{1}{\omega} \ln \left( -\frac{u_- + u_+}{2\omega^2(A_1 - C_1/p)} \right).$$

### Для управления по оси $y$ .

Определим время переключения для случая, когда до момента переключения управление равно  $\epsilon_2$ , а после переключения —  $-\epsilon_2$ . Исследуем следующую систему:

$$\begin{cases} \dot{x}_3 = x_4, \\ \dot{x}_4 = \epsilon_2. \end{cases}$$

Начальные условия:  $x_3(0) = y_0$ ,  $x_4(0) = v_2$ .

Функции, являющиеся решениями системы, имеют вид:

$$\begin{cases} x_3 = \epsilon_2 \frac{t^2}{2} + B_2 t + B_1, \\ x_4 = \epsilon_2 t + B_2. \end{cases}$$

Учитывая начальные условия, получаем:

$$\begin{cases} x_3 = \epsilon_2 \frac{t^2}{2} + v_2 t + y_0, \\ x_4 = \epsilon_2 t + v_2. \end{cases}$$

Траектория после момента переключения управления записывается следующим образом:

$$\begin{cases} x_3 = -\epsilon_2 \frac{t^2}{2} + B_2 t + B_1, \\ x_4 = -\epsilon_2 t + B_2. \end{cases}$$

Эта траектория проходит через точку  $(y_k, 0)$ , на основании чего определяются коэффициенты:

$$\begin{cases} x_3 = y_k - \epsilon_2 \frac{(t_k - t)^2}{2}, \\ x_4 = \epsilon_2(t_k - t). \end{cases}$$

В момент переключения управления  $t_2$ :

$$\begin{cases} \epsilon_2 \frac{t_2^2}{2} + v_2 t_2 + y_0 = y_k - \epsilon_2 \frac{(t_k - t_2)^2}{2}, \\ \epsilon_2 t_2 + v_2 = \epsilon_2(t_k - t_2). \end{cases}$$

Выразим из второго уравнения:

$$t_k - t_2 = t_2 + \frac{v_2}{\epsilon_2}.$$

Подставим это значение в первое уравнение и получим:

$$\epsilon_2 t_2^2 + 2v_2 t_2 + y_0 - y_k + \frac{v_2^2}{2\epsilon_2} = 0.$$

Решив уравнение, найдём момент переключения управления:

$$t_2 = \frac{-v_2 + \sqrt{\frac{v_2^2}{2} + \epsilon_2(y_k - y_0)}}{\epsilon_2}.$$

Рассмотрим теперь случай, когда до момента переключения управление равно  $-\epsilon_2$ , а после переключения равно  $\epsilon_2$ . Исследуем следующую систему уравнений с начальными условиями  $x_3(0) = y_0$  и  $x_4(0) = v_2$ :

$$\begin{cases} \dot{x}_3 = x_4, \\ \dot{x}_4 = -\epsilon_2. \end{cases}$$

Начальные условия:  $x_3(0) = y_0$ ,  $x_4(0) = v_2$ .

Решениями данной системы являются функции:

$$\begin{cases} x_3 = -\epsilon_2 \frac{t^2}{2} + B_2 t + B_1, \\ x_4 = -\epsilon_2 t + B_2. \end{cases}$$

С учётом начальных условий:

$$\begin{cases} x_3 = -\epsilon_2 \frac{t^2}{2} + v_2 t + y_0, \\ x_4 = -\epsilon_2 t + v_2. \end{cases}$$

Выражение для траектории после момента переключения управления выглядит следующим образом:

$$\begin{cases} x_3 = \epsilon_2 \frac{t^2}{2} + B_2 t + B_1, \\ x_4 = \epsilon t + B_2. \end{cases}$$

Данная траектория проходит через точку  $(y_k, 0)$ . Выразим коэффициенты с учетом этого условия:

$$\begin{cases} x_3 = y_k + \epsilon_2 \frac{(t - t_k)^2}{2}, \\ x_4 = \epsilon_2(t - t_k). \end{cases}$$

На момент переключения управления  $t_2$ :

$$\begin{cases} -\epsilon_2 \frac{t_2^2}{2} + v_2 t_2 + y_0 = y_k + \epsilon_2 \frac{(t_2 - t_k)^2}{2}, \\ -\epsilon_2 t_2 + v_2 = \epsilon_2(t_2 - t_k). \end{cases}$$

Из второго уравнения получаем:

$$t_2 - t_k = \frac{v_2}{\epsilon_2} - t_2.$$

Подставляем в первое и получаем:

$$\epsilon_2 t_2^2 - 2v_2 t_2 - y_0 + y_k + \frac{v_2^2}{2\epsilon_2} = 0.$$

Решив уравнение, найдём момент переключения управления:

$$t_2 = \frac{v_2 + \sqrt{\frac{v_2^2}{2} + \epsilon_2(y_0 - y_k)}}{\epsilon_2}.$$

#### 4.1.7 Построение рабочей области станда.

Ранее в этой главе было показано, что у манипулятора есть ограниченное пространство для перемещения, обозначаемое как  $\Omega^d$ . Алгоритм управления должен гарантировать, что скорость движения конечной точки на границе области  $\Omega^d$  равна нулю, чтобы избежать выхода за пределы геометрических ограничений.

На рисунке 4.7 показана эта область. Рассмотрим точку  $K$ . Её ближайшая граничная точка множества  $\Omega^d$  — это точка  $K_B$ . При движении из  $K$  к

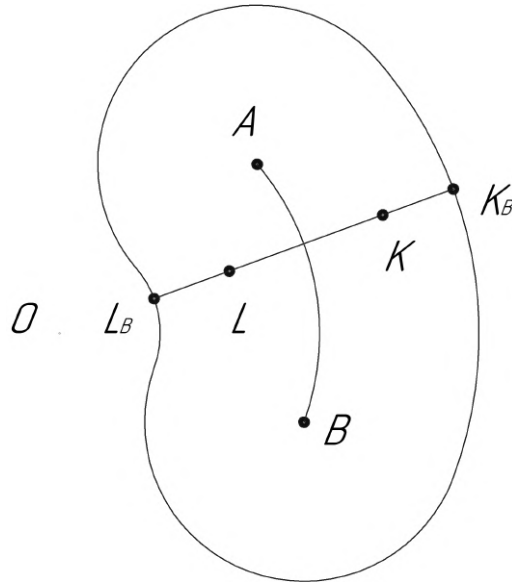


Рисунок 4.7 — Точки множества допустимых положений и соответствующие точки на границе

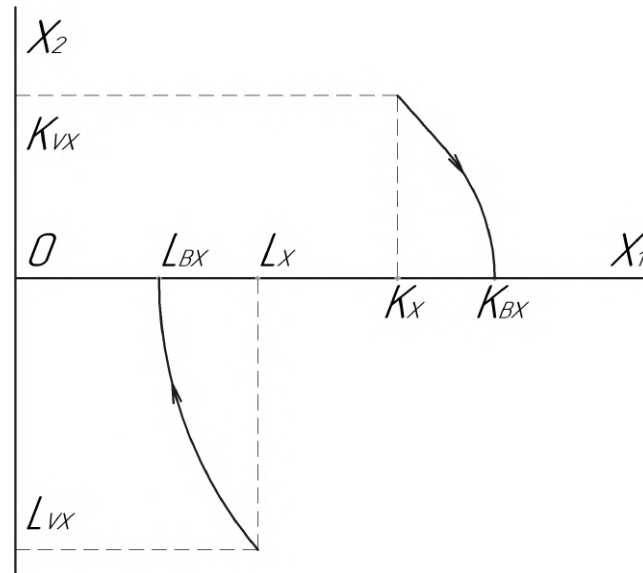


Рисунок 4.8 — Нахождение скорости движения к границе по оси  $x$

$K_B$  торможение происходит с ускорением по оси  $x$ , равным  $-(\epsilon_1 + \omega^2 x(t_k))$ . Таким образом, мы можем построить траекторию движения через точку  $K_{BX}$  на фазовой плоскости  $x_1 O x_2$  и найти соответствующую скорость  $K_{VX}$  для точки  $K_X$ , как показано на рис. 4.8.

Аналогично, для точки  $L$  ближайшей граничной точкой является  $L_B$ . Движение к ней происходит с ускорением по оси  $x$ , равным  $\epsilon_1 - \omega^2 x(t_k)$ . Мы можем построить траекторию через точку  $L_{BX}$  и найти скорость  $L_{VX}$  для точки  $L_X$ , что также показано на рис 4.8.

Подобным образом можно определить скорость по оси  $y$ . Основное различие заключается в том, что траектории на фазовой плоскости  $x_3 O x_4$  будут

иметь форму парабол, а не гипербол (вывод этого свойства основан на решении задачи быстродействия, описанной ранее). Итак, с учетом условия нулевой скорости на границе множества допустимых положений, можно определить область управляемости для каждой координаты, принимая во внимание геометрические ограничения манипулятора.

Применяя принцип суперпозиции, можно построить множество в 4D-пространстве  $(x, \dot{x}, y, \dot{y})$ , сечения которого представляют собой рабочие области установки при различных скоростях (области активной фазы динамической имитации). Если выйти за пределы этого множества, это может привести к выходу за пределы множества допустимых положений и, следовательно, к возможной поломке имитационного стенда. Очевидно, что форма этого множества зависит от геометрических параметров стенда, угловой скорости вращения стенда  $\omega$  и ограничений  $\epsilon_1$  и  $\epsilon_2$ .

Таким образом, алгоритм управления должен быть построен таким образом, чтобы обеспечить нулевую скорость на границе области  $\Omega^d$  и учесть геометрические ограничения, параметры стенда и ограничения на ускорение. С учетом этого можно определить область управляемости для каждой координаты и гарантировать надежную работу имитационного стенда.

#### **4.2 Численное моделирование управления роботом-манипулятором с учетом порога чувствительности вестибулярного аппарата**

Для получения численных результатов было проведено моделирование с использованием графического движка Unity 3D и кода, написанного на C#. Численное интегрирование осуществлялось с помощью метода Рунге–Кутты 4-го порядка [78], а величина шага интегрирования составляла 0,008.

Параметры смоделированной роботизированной руки приведены в таблице 2.

Был произведен расчет рабочей области с условием возврата в набор из десяти различных точек стартовой дуги  $\overset{\circ}{AB}$ . Полученные результаты для этих точек оказались весьма схожими (отличие менее 1 процента), что позволяет в первом приближении пренебречь положением точки на стартовой дуге.

Основной интерес представляет определение максимально допустимых скоростей движения кабины в направлении к границе множества допусти-

Таблица 2 — Параметры роботизированной руки

Параметр	Величина	Параметр	Величина
$l_1$	0.5м	$l_2$	1.7м
$d_1$	2.2м	$d_2$	2.3м
$m_1$	2700кг	$m_2$	1200кг
$I_1$	2989кгм <sup>2</sup>	$I_2$	2195кгм <sup>2</sup>

мых положений, чтобы иметь возможность гарантировано остановиться на границе. На рисунке 4.9 приведен график модуля максимальной допустимой скорости кабины в зависимости от положения манипулятора  $(x, y, \sqrt{\dot{x}^2 + \dot{y}^2})$ . Моделирование проводилось при  $x(t_k) = 1457$  мм,  $\omega = 0.05$  рад/с,  $\epsilon_1 = \epsilon_2 = 14$  мм/с<sup>2</sup>.

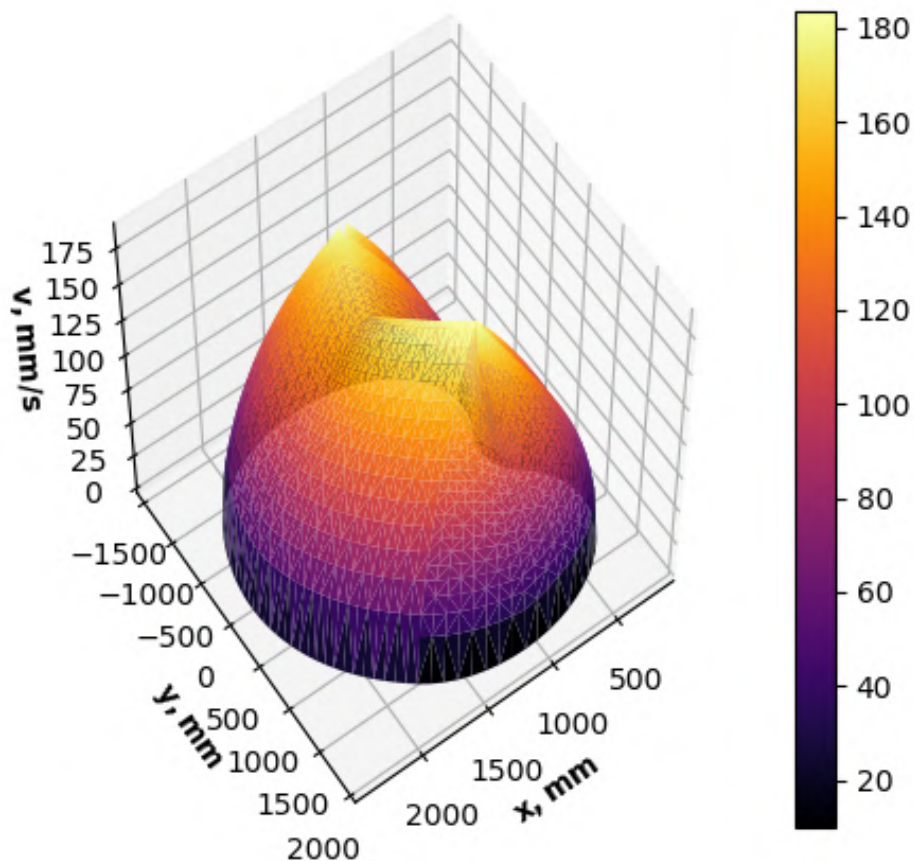


Рисунок 4.9 — Зависимость модуля максимальной скорости движения к границе от координат расчетной точки



После определения рабочей области стенда, был применен метод управления с использованием интегрального скользящего режима (3.28), описанный в главе 3, к модели стенда. Рассмотрим задачу перемещения роботизированной руки в плоскости  $(x, y)$ .

Собранные данные о маневрировании в турбулентности, полученные в симуляторе полета, были взяты в качестве эталонных ускорений, и смоделированы синтетические периодические ускорения со схожими амплитудо-частотными характеристиками. Моделирование проводилось с учетом ограничений на положение концевой точки, полученном ранее.

Предложенное в главе 3 управление с интегральным скользящим режимом сравнивалось с управлением по обратной связи (ПД):

$$u(t) = -318x(t) - 30\dot{x}(t),$$

и с управлением в виде скользящего режима первого порядка с поверхностью скольжения  $s = x(t) - \dot{x}(t)$ , в котором управляющий сигнал фильтруется в реальном времени:

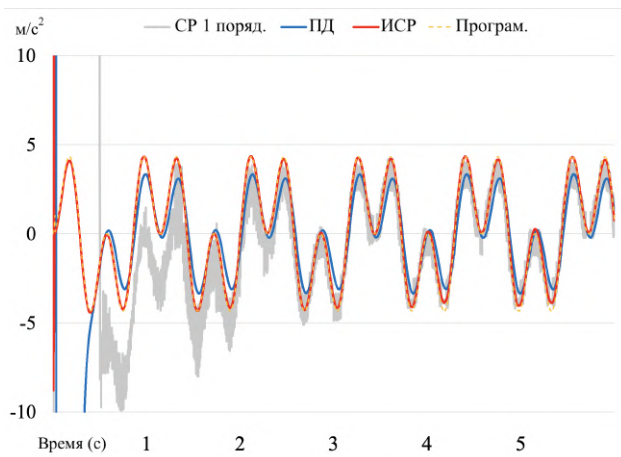
$$u(t) = 10\text{Sign}(s).$$

Здесь  $x(t)$  обозначает разницу между желаемым и фактическим положением расчетной точки кабины, аналогично главе 3.

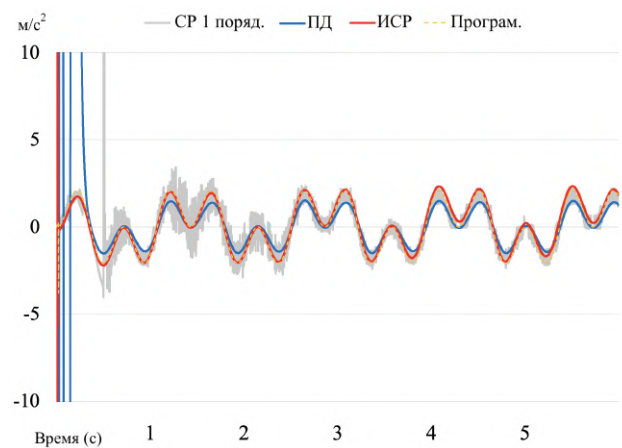
Рисунки 4.10а и 4.10б демонстрируют ускорения по вертикальной и горизонтальной осям в плоскости движения первых двух звеньев манипулятора. Изображения показывают, что управление с интегральным скользящим режимом сходится к требуемым ускорениям (с точностью до порога чувствительности) менее чем за 0.1 секунды, в то время как другие варианты управления имеют значительную задержку на протяжении всего интервала.

Траектории расчетной точки сравниваются на рисунках 4.10в и 4.10г. Следует отметить, что только управление с интегральным скользящим режимом не сходится к желаемой траектории, что, однако, не является критичным, поскольку соответствует постановке задачи: отслеживание (имитация) только ускорений.

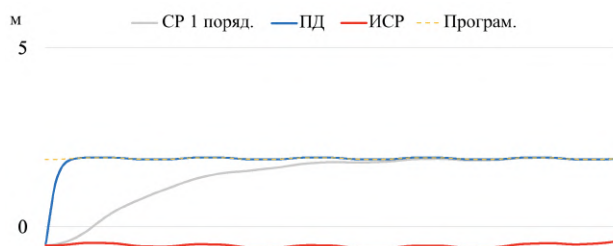
На рисунке 4.10д наблюдается, что предложенное управление с интегральным скользящим режимом приводит к меньшим значениям абсолютной величины ошибки ускорения. Если учесть чувствительность (выраженную величиной  $\delta$ ), то на отдельных участках она достигает нуля для предложенного управления. В этом случае можно говорить о полной синхронизации,



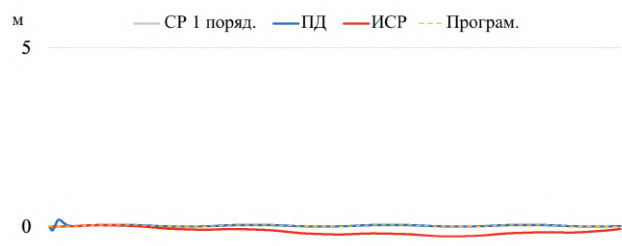
а) Ускорения по оси  $y$ .



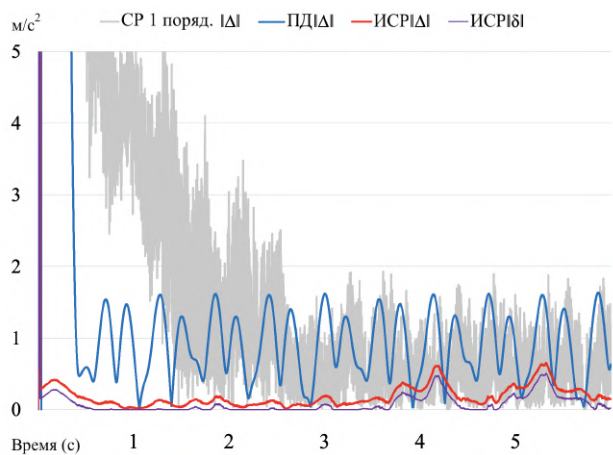
б) Ускорения по оси  $x$ .



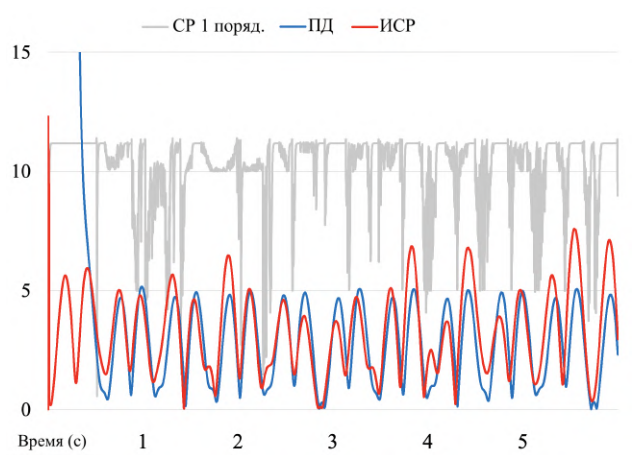
в) Траектория движения по оси  $y$ .



г) Траектория движения по оси  $x$ .



д) Сравнение  $\|\delta\|$  и  $\|\Delta\|$  ( $\epsilon = 0.1$ ).



е) Модуль управления  $u$ .

Рисунок 4.10 — Сравнение результатов для скользящего режима на базе усредненного субградиента (ИСП), скользящего режима первого порядка (СП 1-го порядка) и управления с обратной связью (ПД) для двух координат

при условии выполнения остальных критериев синхронизации, указанных в главе 3.

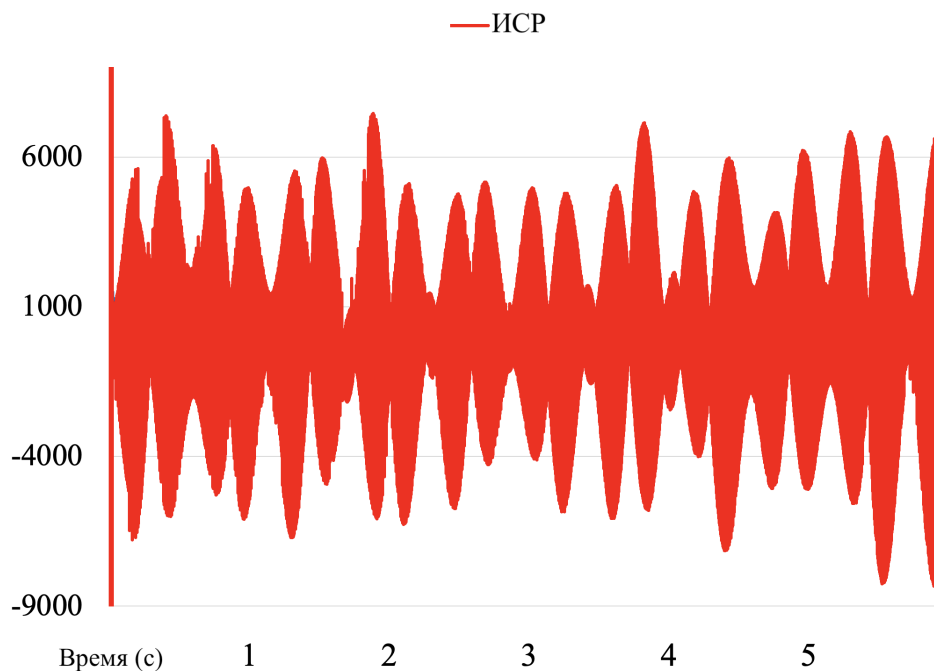


Рисунок 4.11 — Пример работы управления  $\tau$  в расширенной системе.

Таблица 3 — Сравнение качества работы алгоритмов с интегральным скользящим режимом (ИСП-УСГ), пропорционально-дифференциального (ПД), скользящего режима 1го порядка (СР)

Критерий	ИСП-УСГ $\Delta$	ИСП-УСГ $\delta$	ПД $\Delta$	СР 1-го пор. $\Delta$
Среднеkv. ошибка	<b>0.07</b>	<b>0.02</b>	1.03	0.52
Средн. абс. ошибка	<b>0.21</b>	<b>0.1</b>	0.91	0.62

Таблица 3 иллюстрирует сравнение среднеквадратичной ошибки и средней абсолютной ошибки для систем управления, использующих скользящий режим первого порядка, пропорционально-дифференциальный регулятор и управление с интегральным скользящим режимом после третьей секунды моделирования. Эти числовые результаты подчеркивают эффективность управления с применением интегрального скользящего режима.

Рисунок 4.10е отображает абсолютные величины управляющего сигнала для разных алгоритмов управления. Он также демонстрирует отсутствие чаттеринг-эффекта для управления с интегральным скользящим режимом  $u$ . Чаттеринг-эффект проявляется только во второй производной  $\ddot{u} = G_2^* \tau$  (см. рисунок 4.11).

### 4.3 Скользящее управление с применением регуляризации на границе множества допустимых положений

Стоит отметить, что при приближении к границам допустимых положений станда (см. рисунок 4.3) возникает проблема вырождения матрицы  $G_2$ , определенной в уравнении (3.26) и используемой в управлении (3.28). В результате обратная матрица перестает существовать, и для построения корректного управления требуется применение регуляризации. Подробное описание процесса регуляризации и модифицированное доказательство аналога теоремы 1 можно найти в статье [17]. Модифицированная версия управления также была протестирована на модели манипулятора.

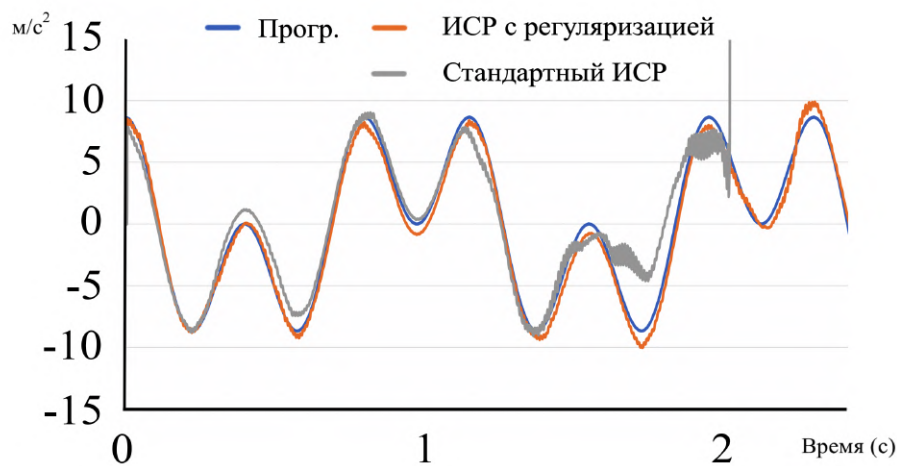


Рисунок 4.12 — Ускорения по оси  $x$  для ИСП с регуляризацией, без регуляризации и эталонные

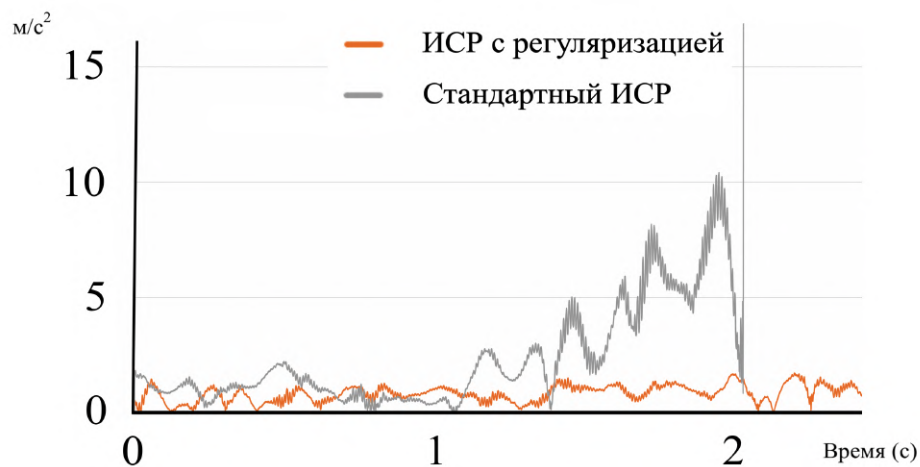


Рисунок 4.13 — Сравнение  $\|\delta\|$  для ИСП с регуляризацией и без регуляризации

Сравнение перемещений роботизированной руки проводилось в двух измерениях ( $x$ ,  $y$ ). В качестве эталонных ускорений использовались данные, аналогичные предыдущим тестам, однако начальное положение кабины стенда бралось ближе к границе множества допустимых положений. С целью оценки эффективности предложенного метода управления с применением скользящего режима и регуляризации, результаты сравнивались с аналогичным управлением без регуляризации. Рисунок 4.12 демонстрирует имитацию ускорений по оси  $x$ . Полученные результаты подтверждают, что ускорения, полученные с использованием управления с применением скользящего режима, сходятся к эталонным значениям. Предложенная модификация позволяет избежать сингулярного состояния на 2-й секунде.

После 2-й секунды, в случае управления без учета регуляризации, наблюдаются скачки в управляющем сигнале, ускорениях и траектории. Это связано с приближением системы к положению, в котором манипулятор ”выпрямляется”, и определитель матрицы  $G_2$  становится близким к нулю. В результате, управляющий сигнал резко возрастает.

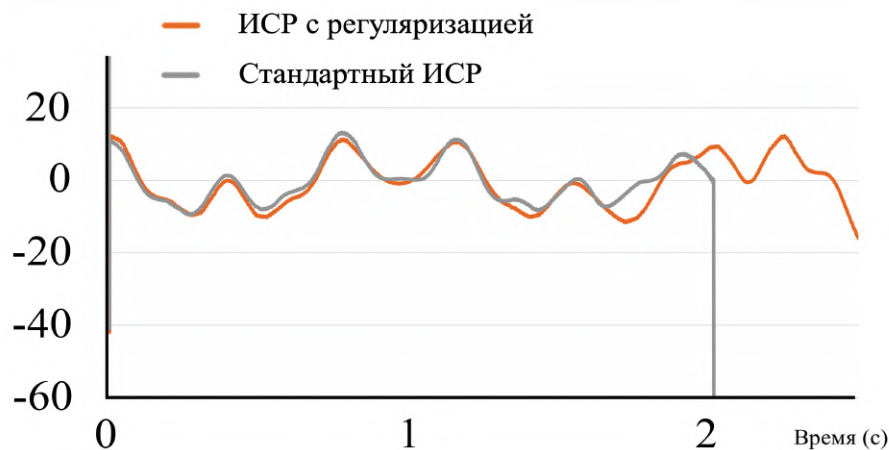


Рисунок 4.14 — Оценка значения управления  $u_1$

Для сравнительных целей исследовались различные управления, и для оценки их эффективности измерялись величины ошибки имитации  $|\Delta|$  (рис. 4.13) и крутящий момент в двигателе первого звена  $u_1$  (рис. 4.14). Управление без регуляризации демонстрирует увеличение ошибок вблизи области вырождения.

Траектории конечного исполнительного устройства, полученные с использованием выбранных управлений, сравниваются на рисунке 4.15. Эти траектории согласуются с изменением управляющего воздействия  $u_1$ , пред-

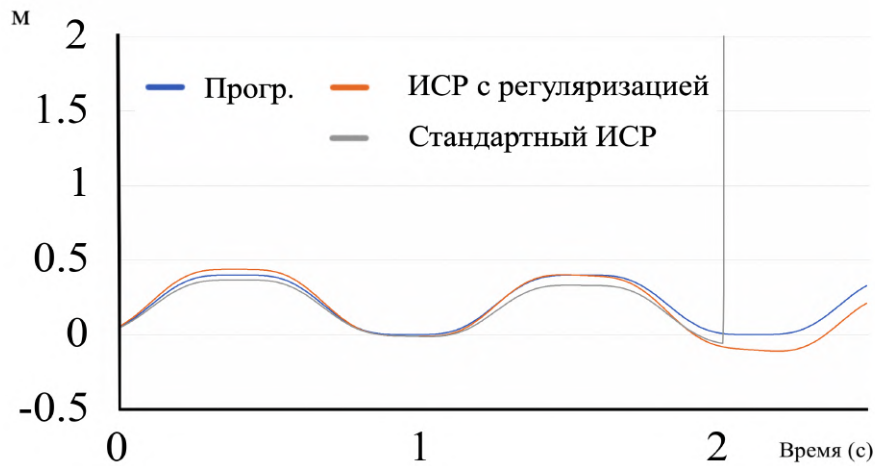


Рисунок 4.15 — Траектория системы по оси  $x$

ставленного на рисунке 4.14. Траектории могут расходиться со временем, поскольку задача управления заключается в имитации ускорений, а не повторении траектории.

Численные результаты подтверждают, что разработанное управление способно проходить через зону сингулярности и возвращаться в обычную зону, не прерывая имитацию.

#### 4.4 Реализация симулятора в Unity 3D и анализ результатов

В рамках данного исследования был разработан и использован симулятор в Unity 3D, реализующий динамику [72] летательного аппарата. Симулятор позволяет проводить моделирование маневров самолета в реальном времени, что обеспечивает достоверность данных при проверке разработанных алгоритмов управления. В Unity 3D также была реализована модель стенда на базе робота-манипулятора, что позволило интегрировать ее с симулятором и провести тестирование разработанного управления с интегральным скользящим режимом в реальном времени. Рисунок 4.16 демонстрирует окно симулятора и общий вид стенда с роботом-манипулятором.

В ходе проведения маневров в симуляторе были получены графики, отображающие динамику ускорений и перемещений роботизированной руки. Рисунок 4.17 показывает пример графика ускорений, полученного при выполнении одного из маневров.

Анализ результатов тестирования демонстрирует, что разработанный симулятор на основе Unity 3D успешно моделирует динамику самолета и маневры в реальном времени, предоставляя более достоверные данные для



Рисунок 4.16 — Вид на окно симулятора в Unity 3D

проверки алгоритмов управления. Интеграция симулятора с моделью стенда на базе робота-манипулятора способствует комплексному тестированию разработанных методов управления и оценке их эффективности при выполнении различных маневров.

Рисунок 4.18 иллюстрирует ошибку имитации для приведенного маневра. Результаты тестирования показали, что предложенный метод регуляризованного скользящего управления обеспечивает высокую точность и устойчивость работы стенда, что делает его применимым для разработки стендов-тренажеров.

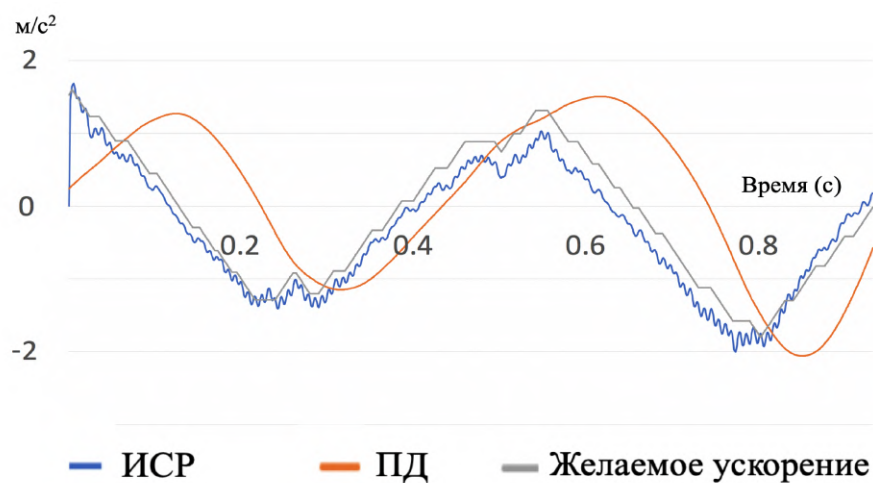


Рисунок 4.17 — Ускорения по оси  $x$  для ИСР и ПД-регуляторов



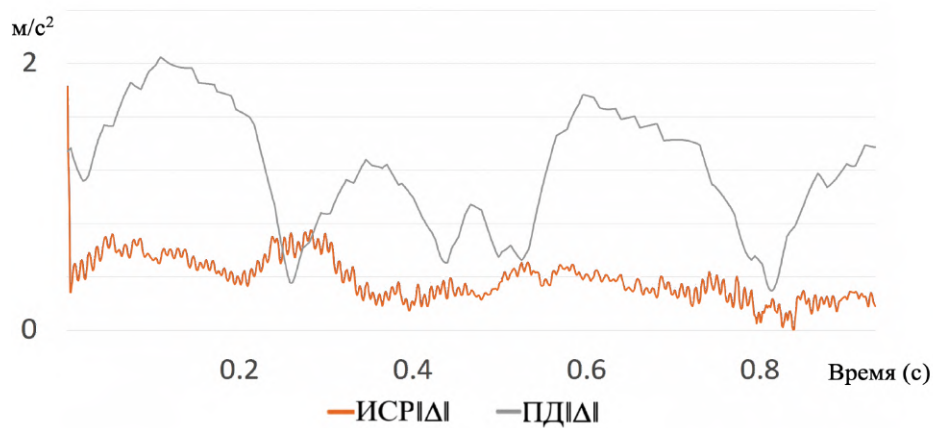


Рисунок 4.18 — Сравнение  $\|\Delta\|$  для ИСР и ПД-регуляторов

В ходе данного исследования было выполнено ограниченное число тестов и маневров. Разработка и проведение дополнительных сценариев тестирования может способствовать более глубокому анализу эффективности и точности предложенных методов управления. Однако, ручной выбор специфических маневров может потребовать значительных ресурсов, времени и привлечения опытных пилотов для оценки реалистичности тестовых ситуаций. Учитывая указанные ограничения, был выбран альтернативный подход — методика максиминного тестирования. Данный подход позволяет генерировать наиболее неблагоприятные возмущения в рамках игровой задачи, обеспечивая тщательное изучение разработанного алгоритма управления. В следующем разделе подробно описана указанная методика и представлены результаты тестирования.

#### 4.5 Методика максиминного тестирования качества управления механическими системами

Согласно теореме, представленной в главе 3, алгоритм автоматического управления с использованием интегрального скользящего режима решает задачу имитации, даже при наличии ошибок моделирования и возмущений (например, неточности исполнительных механизмов). Однако практическая реализация алгоритма может существенно отличаться от теоретических результатов, особенно при наличии фиксированного шага дискретизации в вычислителе. В связи с этим, необходимо оценить качество работы алгоритма, для чего предлагается использовать методику максиминного тестирования качества управления [70].



Опишем данную методику. Предположим, что динамическая система представлена уравнением:

$$\dot{x} = f(x, u, v) \quad (4.12)$$

Начальные условия заданы как  $x(0) \in X^0$ , управления  $u(\cdot) \in U$  и действующие в процессе движения возмущения  $v(\cdot) \in V$ . Введем вектор всех возмущений  $w(\cdot) = (x(0), v) \in W$ , где  $W = X^0 \times V$ , причем  $u$  и  $w$  будем считать независимыми. Будем рассматривать задачу как игру двух игроков: отвечающего за управление и отвечающего за возмущения.

Введем функционал качества вида  $J(w, u) = \|x(t_k)\|$ , зависящий от управлений и возмущений.

Пусть задачей управления является приближение функционала качества к нулю, возмущений - отдаление от нуля. *Стратегией* игрока называется совокупность правил, определяющих выбор его действия в зависимости от сложившейся ситуации.

Для объективной оценки качества стратегий управления системой с точки зрения функционала  $J$ , представим данную задачу в виде *дифференциальной игры* [103].

Тогда задачей игрока, отвечающего за возмущения, является ухудшение (максимизация) функционала даже при наилучших управлениях, то есть получение:

$$J_0 = \max_{w \in W} \min_{u \in U} J(w, u).$$

В то время как задача управляющего игрока представляет собой улучшение (минимизацию) функционала даже при наихудших возмущениях:

$$J^0 = \min_{u(\cdot) \in U} \max_{w(\cdot) \in W} J(w, u).$$

Будем рассматривать такие системы (4.12), для которых  $J_0$  и  $J^0$  всегда существуют при ограниченных  $W$  и  $U$ . Введенная таким образом антагонистическая игра обозначается как  $\Gamma = (V, U, J)$ .

**Теорема 3.** Для любой  $\Gamma$  верны следующие неравенства [103]:

$$J_0 = \max_{w(\cdot) \in W} \min_{u(\cdot) \in U} J \leq \min_{u(\cdot) \in U} \max_{w(\cdot) \in W} J = J^0 \leq \max_{w(\cdot) \in W} J = J(\tilde{w}^0(\tilde{u}), \tilde{u}) = \tilde{J}. \quad (4.13)$$

Здесь  $\tilde{u}$  — произвольное управление, а  $\tilde{w}^0(\tilde{u})$  — наихудшие возмущения для данных управлений.

Необходимо протестировать качество управления согласно разработанной методике. При этом, для оценки качества управления используется  $J_0 = \max \min J$ , поэтому тестирование будем называть *максиминным*. В методике можно выделить три этапа:

**1-й этап** — предварительный. На этом этапе осуществляется поиск  $J_0$  и оптимальной (наихудшей для управляющего игрока) стратегии для возмущений  $w^*$ .

**2-й этап** — основной. На втором этапе производится процесс испытания с непосредственным участием тестируемого алгоритма управления, при этом на управляемый объект воздействуют наихудшие возмущения, найденные на первом этапе. В конце второго этапа получим значение функционала качества  $\tilde{J} = \|x(t_k)\|$ .

**3-й этап** — заключительный. На этом этапе происходит сравнение наилучшего  $J_0$  и реального  $\tilde{J}$  значений функционалов. Здесь будет использоваться сравнение вида  $100 \cdot J_0 / \tilde{J}$ , что обеспечивает оценку качества управления по столбальной шкале.

Таким образом можно тестировать любой заданный автоматический алгоритм управления.

На рисунке 4.19 представлена функциональная схема компьютерного варианта системы тестирования.

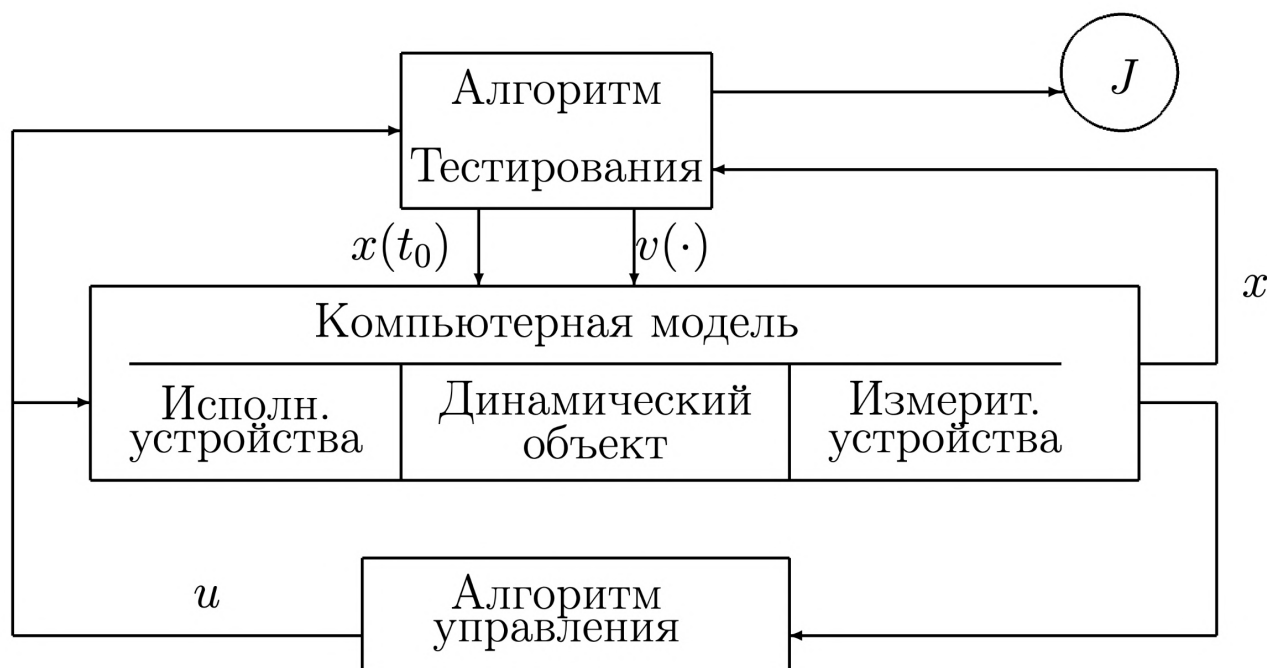


Рисунок 4.19 — Схема методики тестирования

Для реализации данной схемы необходимо иметь в распоряжении модель функционирования управляемого объекта, измерительных и исполнительных устройств. Сам тестируемый алгоритм стабилизации может быть представлен как "черный ящик" с входом и выходом. Блок алгоритма тестирования подает в компьютерную модель наихудшие возмущения, найденные на первом этапе.

Таким образом, на втором этапе, путем компьютерного и (или) имитационного моделирования возмущений, создается в некотором смысле наихудшая среда для функционирования автоматической системы управления.

Важнейшим свойством предложенной схемы тестирования является возможность объективного сравнения между собой нескольких представленных для тестирования алгоритмов управления. Далее рассмотрим метод поиска наихудших возмущений.

#### 4.5.1 Седловая точка дифференциальной игры первого этапа методики тестирования

Обычно в процессе игры игрок делает выбор в зависимости от конкретной ситуации. Однако при реализации первого этапа тестирования предпочтительно найти *программную* стратегию для наихудших возмущений. Программными называются такие стратегии, при которых решения приняты игроком заранее (независимо от сложившейся ситуации) и зависят только от времени. По этой причине их проще запрограммировать в компьютерной модели для второго этапа тестирования.

Кроме этого, из (4.13) следует, что у управляющего игрока есть объективный шанс получить оценку 100 только при выполнении условия  $J_0 = J^0$ :

$$\max_{w(\cdot) \in W} \min_{u(\cdot) \in U} J = \min_{u(\cdot) \in U} \max_{w(\cdot) \in W} J. \quad (4.14)$$

Покажем, что этим условиям удовлетворяет *седловая точка* (точка равновесия). Это такая пара стратегий  $(u^*, w^*)$ , что для любых  $u \in U$  и  $w \in W$  выполняется неравенство:

$$J(w, u^*) \leq J(w^*, u^*) \leq J(w^*, u) \quad (4.15)$$

Как видно из определения, стратегия, обеспеченная седловой точкой, является программной и оптимальной для обоих игроков - то есть от нее не выгодно отклоняться ни тому, ни другому.

Сформулируем еще одну теорему, которую в дальнейшем будем использовать для нахождения точки равновесия:

**Теорема 4.** *Для того чтобы пара стратегий  $(w^*, u^*)$  была точкой равновесия антагонистической игры  $\Gamma = (W, U, J)$ , необходимо и достаточно, чтобы было выполнено неравенство [103]:*

$$J(w^*, u^*) = \max_{w \in W} \min_{u \in U} J(w, u) \geq J(w, u^*), \forall w \in W. \quad (4.16)$$

#### 4.5.2 Редукция к геометрической игре для программных стратегий тестирования

Предположим, что имеется некоторая программная траектория и программное управление системой (например, желаемое движение при имитации ускорений). Рассмотрим линейную систему в отклонениях от программной траектории вида:

$$\dot{x} = A(t)x + B(t)u + C(t)v. \quad (4.17)$$

где матрицы  $A(t)$ ,  $B(t)$ ,  $C(t)$  являются кусочно-постоянными внутри шага дискретизации.

Начальные условия  $x(0) \in X^0$ , управления  $u(\cdot) \in U$ , где  $U = [u(\cdot) \in KC[0, t_k] \mid \|u(t)\| \leq u^{max}]$ . Введем вектор всех возмущений  $w(\cdot) = (v, x(0)) \in W$ , где  $W = [w(\cdot) \in KC[0, t_k] \mid w(t) \in S]$ , причем  $u$  и  $w$  будем считать независимыми. Здесь  $KC$  обозначает кусочно-непрерывную функцию, а  $S$  — некоторое ограниченное множество.

Применим способ решения максиминной задачи, использующий разложение  $x = s - q$  на возмущаемую и управляемую подсистемы [70]:

$$\dot{s} = A(t)s + Cv, \quad s(0) = x(0), \quad (4.18)$$

$$\dot{q} = A(t)q - Bu, \quad q(0) = 0. \quad (4.19)$$

Введем понятие множества *достижимости* как множества всех таких состояний, в которые можно привести динамическую систему при помощи допустимого управления/возмущения из начальной точки (начального состояния) за заданный промежуток времени. Математически множество достижимости по управлениям записывается как:

$$D_u(t_k, q(0)) = \{q(t_k) \mid \dot{q} = A(t)q - Bu, q(0) = 0, \forall u \in U\}$$

В то время как по возмущениям оно является объединением множеств достижимости при всех возможных начальных условиях  $s(0) = x(0) \in X^0$  :

$$D_w(t_k, X^0) = \bigcup_{x(0) \in X^0} D_v(t_k, s(0)) = \{s(t_k) \mid \dot{s} = A(t)s + Cu, s(0) = x(0), \forall v \in V\}.$$

Каждая точка во множестве достижимости соответствует некоторым управлениям/возмущениям. Если построить множества достижимости  $D_w$  и  $D_u$  в одном пространстве переменных, то исходная дифференциальная игра  $\Gamma$  сведется к геометрической игре  $\Gamma_1 = (D_w, D_u, \rho)$ . Здесь функционал качества  $J(w, u) = \|x(t_k)\|$  выглядит как  $\rho(s, q) = \|s(t_k) - q(t_k)\|$  и представляет собой расстояние между точками из множеств  $D_w$  и  $D_u$ . А седловая "точка" является парой точек на плоскости, принадлежащих соответствующим множествам достижимости.

В такой постановке аналогом неравенства (4.16) из **теоремы 7** является следующее:

$$\rho(s(t_k)^*, q(t_k)^*) = \max_{s(t_k) \in D_w} \min_{q(t_k) \in D_u} \rho(s(t_k), q(t_k)) \geq \rho(s(t_k), q(t_k)^*), \forall s(t_k) \in D_w. \quad (4.20)$$

Таким образом, для седловой точки  $(s(t_k)^*, q(t_k)^*)$  сфера с центром в  $q(t_k)^*$  и радиусом  $\rho(s(t_k)^*, q(t_k)^*)$  должна включать в себя все  $D_w$  целиком.

Для поиска максимина, который будет использоваться в тестирующей стратегии, используется представление исходной задачи как комбинации задач Булгакова. Промежуточные шаги алгоритма используют методику решения задачи Булгакова о накоплении возмущений по  $k$  координатам, основанную на операции поиска максимума линейного функционала на выпуклом множестве [75; 82; 97]. Кроме этого, на каждом шаге алгоритма используется операция поиска проекции точки на выпуклое множество. Ее решение получим с помощью метода условного градиента [84; 85].

## 4.6 Тестирование качества управления роботом-манипулятором

Используем линейное приближение системы (4.4), описывающей динамику робота-манипулятора. Рассматриваются участки динамической имитации длительностью 1/60 секунды. В качестве невозмущенной траектории берется точное воспроизведение желаемых ускорений. Система линеаризуется

в окрестности этой траектории и записывается в виде (4.17). Для примера в дальнейшем на графиках рассмотрим плоское движение без первой по счету обобщенных переменных и трех последних (соответствующих кабине) степеней свободы.

Считаем, что геометрические параметры системы идентифицированы в соответствии с приведенным ранее методом и система абсолютно жесткая.

Действующие в процессе движения возмущения вводятся как паразитные моменты сил в работе (вызванные в том числе движением кабины), и ошибки, соответствующие неточной реализации программного управления двигателями.

Зададим вид и ограничения на возмущения исходя из опыта и понимания особенностей задачи. В дальнейшем эти данные можно скорректировать до значений, полученных экспериментальным путем. Действующие в процессе движения возмущения будем искать в следующей форме:  $v_i(t) = k_i \sin(s_i t + p_i) + n_i$ ,  $i = 1, 2$ . Здесь  $k_i$ ,  $s_i$  и  $p_i$  берутся из заданного дискретного множества,  $n_i \in [-n_i^{\max}, n_i^{\max}]$ . Начальные возмущения соответствуют ошибкам трекинга (ошибкам определения угловой скорости звеньев) и также берутся из известного дискретного множества. Ограничения на дополнительное управление и возмущения введем как:  $|u| \leq 0.01u_{\max}^*$ ,  $n_i^{\max} = 0.0033u_{\max}^*$ , где  $u^*$  — программное управление. В контексте игровой задачи в качестве функционала использовалась взвешенная норма углов и угловой скорости звеньев в конце каждого шага имитации, заданная как  $J(w, u) = \|x(t_k)\|$ . Важно отметить, что данное значение функционала увеличивается при росте интеграла от ошибки имитации  $\|\delta\|$ . Основываясь на вкладе каждой составляющей в ошибку на каждом шаге тестирования, пересчитывались веса в норме. Шаг имитации и шаг дискретизации модели управления не совпадают. Движение стенда рассматривалось только в плоскости манипулятора.

Численное тестирование трех типов алгоритмов проводилось в среде Unity 3D для всех начальных условий, при которых седловая точка существует. На каждом тесте выставлялась оценка качества управления, полученная алгоритмом. В таблице 4 приведены полученные алгоритмами, реализованными в Unity 3D, средние оценки и среднеквадратичное отклонение от данных оценок, высчитанное на всем множестве начальных условий для всех участков имитируемой траектории. Заметим, что качество работы пред-

ложенного алгоритма с использованием интегрального скользящего режима является наилучшим среди всех тестируемых, однако наиболее стабильная оценка получается у скользящего режима первого порядка.

Таблица 4 — Сравнение качества работы алгоритмов управления с интегральным скользящим режимом (ИСП-УСГ), пропорционально-дифференциального (ПД), скользящего режима 1го порядка (СР) по стобалльной шкале

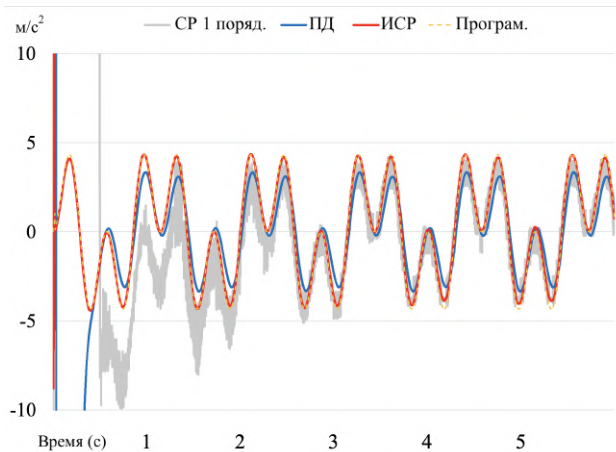
Критерий	ИСП-УСГ	ПД	СР
Средняя	<b>98</b>	68	87
Среднекв. откл.	<b>1.18</b>	13.6	0.7

Как пример, рассмотрим одну из реализаций работы алгоритмов управления при наличии возмущений и сравним поведение возмущенной системы с невозмущенной.

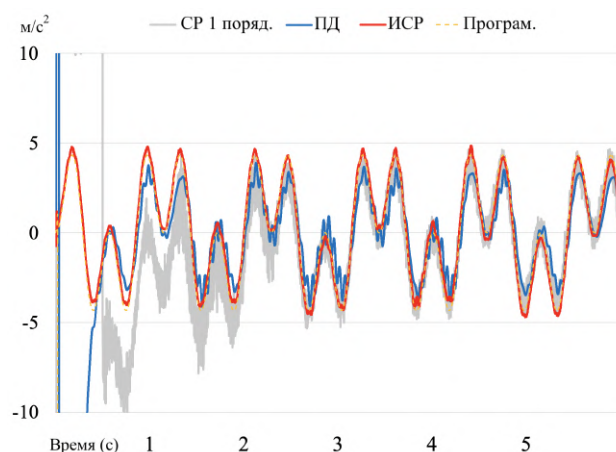
На рисунках 4.20а и 4.20б для оси у, 4.20в и 4.20г для оси х, видно, что общий вид формируемых ускорений существенно меняется только для пропорционально-дифференциального управления. На рисунках 4.20д и 4.20е, можно заметить, что максимальная ошибка после 3-х секунд для пропорционально-дифференциального управления превышает в 1,5 раза максимальную ошибку, полученную при использовании скользящих режимов.

Следует отметить, что средняя ошибка достаточно сильно возрастает в условиях применения интегрального скользящего режима управления. При этом, максимальная ошибка увеличивается в 1,2 раза, однако, несмотря на это, она все равно остается значительно меньше, чем при применении других вариантов управления.

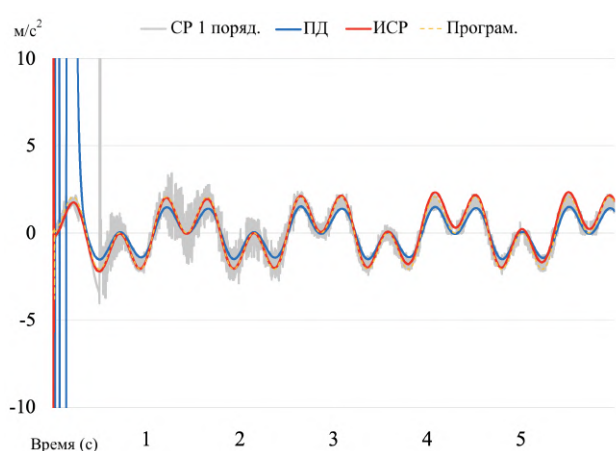
Таким образом, на основе проведенного анализа было продемонстрировано, что использование интегрального скользящего режима управления обеспечивает устойчивость системы к возмущениям. Это делает его предпочтительным вариантом для использования в составе решения задачи синхронизации визуальной и динамической имитации.



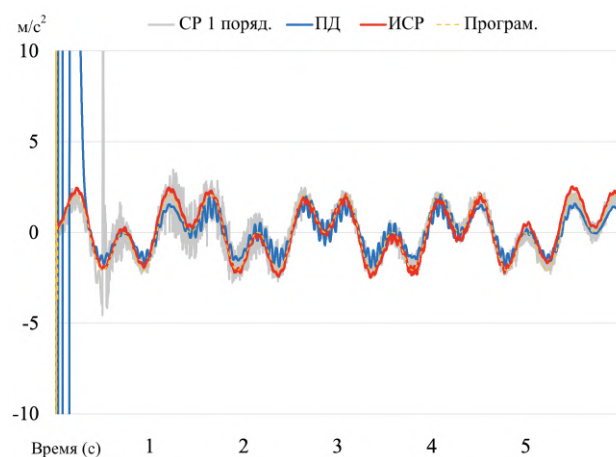
а) Ускорения по оси  $y$  без возмущений



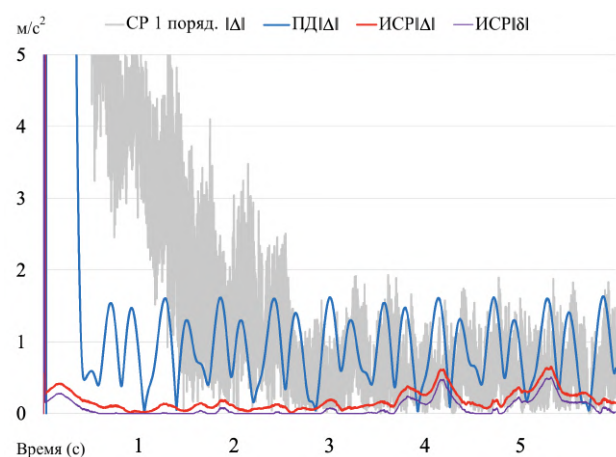
б) Ускорения по оси  $y$  при наихудших возмущениях



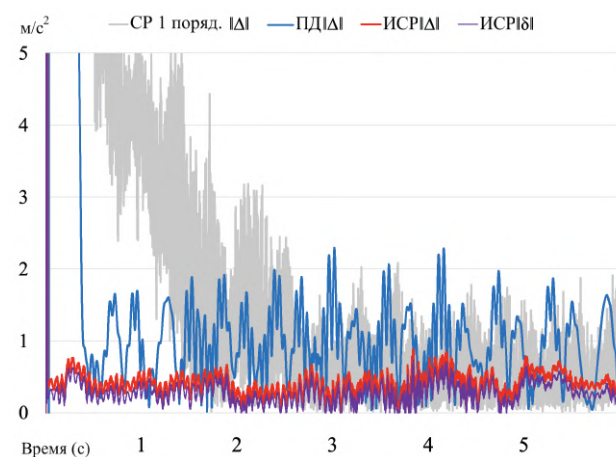
в) Ускорения по оси  $x$  без возмущений



г) Ускорения по оси  $x$  при наихудших возмущениях



д)  $\|\delta\|$  и  $\|\Delta\|$  ( $\epsilon = 0.1$ ) без возмущений



е)  $\|\delta\|$  и  $\|\Delta\|$  ( $\epsilon = 0.1$ ) при наихудших возмущениях

Рисунок 4.20 — Сравнение результатов при наличии возмущений, полученных с помощью методики максиминного тестирования качества управления



## Заключение

Основные результаты работы заключаются в следующем.

Первая глава посвящена детальному изложению задачи динамической имитации управляемых движений. В ней представлены различные виды функционалов и обсуждается решение этой задачи на примере трехстепенной платформы опорного типа. Разработан алгоритм идентификации геометрических параметров платформы и построено множество допустимых положений стенда. Метод множителей Лагранжа использован для построения вписанного в полученное множество параллелепипеда, что в свою очередь позволило разделить алгоритмы имитации ориентации вектора перегрузки и имитации быстрых движений. В подтверждение полученных результатов приведены данные экспериментов на стенде, расположенном в МГУ имени М.В. Ломоносова.

Во второй главе рассмотрена задача визуальной имитации в системах виртуальной реальности. Представлен общий способ решения данной задачи и описана модификация системы визуализации методом фовеального рендеринга, которая позволяет детально визуализировать только ту часть изображения, на которую смотрит пользователь.

В третьей главе сформулирован метод синхронизации и координации визуальной и динамической имитации, что является обобщением результатов предыдущих глав. В рамках этой главы также приведено решение ряда подзадач: отслеживания углового движения стенда с помощью оптической системы трекинга и прогноза углового движения стенда. Предложена и обоснована стратегия управления стендом с использованием интегрального скользящего режима, учитывающего порог чувствительности вестибулярного аппарата. Доказана теорема о практической устойчивости ошибки имитации при использовании данного алгоритма.

В четвертой главе описана модель стенда-тренажера нового типа, основанного на комплексе из промышленного робота-манипулятора и шлема виртуальной реальности. В рамках этой главы представлены уравнения динамики стенда с использованием метода Лагранжа второго рода, а также предложено решение задачи определения области активной фазы динами-

ческой имитации. Приведена методика максиминного тестирования и её использование для проверки робастности предложенного управления.

Таким образом, была успешно решена задача синхронизации и координации визуальной и динамической составляющих имитации. Однако возможности для улучшения синхронизации существуют, и они могут быть достигнуты путем оптимизации механических параметров стенда, увеличения числа его степеней свободы и усовершенствования математической модели визуальной составляющей.

Одно из возможных направлений развития данной темы - это создание прототипа стенда и разработка объективных критериев оценки качества согласования с учетом визуальной и динамической имитации. Основой для таких критериев может служить моделирование вестибулоокулярного рефлекса, что позволит сформировать общий метод настройки параметров алгоритмов управления стендом, исходя из конкретных маневров транспортного средства и объективных показателей качества имитации.

Кроме того, применение адаптивных алгоритмов управления может обеспечить автоматическую корректировку параметров стенда в зависимости от условий эксплуатации, предпочтений оператора или специфических требований обучения. Вместе с этим, разработка алгоритмов для определения оптимальных параметров стенда с использованием методов машинного обучения и оптимизации может улучшить качество имитации и обеспечить более точное воспроизведение реальных ситуаций.

Для обеспечения более полного и реалистичного опыта в симуляторе, можно интегрировать другие варианты имитации, такие как аудио или тактильные сигналы. Следует также учесть индивидуальные различия между пользователями, такие как возраст, опыт пилотирования или физиологические особенности, чтобы выявить их влияние на эффективность синхронизации визуальной и динамической составляющих имитации.

Наконец, расширение исследования на различные типы транспортных средств и сценарии использования, включая судоходство или космические аппараты, позволит определить применимость разработанных алгоритмов и методов в более широком контексте. Исследование совместимости предложенных решений с различными типами транспортных средств и сценариями эксплуатации может способствовать разработке универсальных подходов к синхронизации визуальной и динамической имитации.

Результаты представленного исследования и предложенные направления развития могут значительно способствовать улучшению качества имитационных стендов и обучения операторов различных транспортных средств. Улучшение синхронизации визуальной и динамической составляющих имитации не только повысит эффективность обучения, но и может предотвратить возможные ошибки и аварии, связанные с несоответствием имитационного опыта реальным условиям управления транспортными средствами. Таким образом, дальнейшее развитие данной темы имеет важное практическое значение для обеспечения безопасности и надежности транспортных систем.

### **Благодарности.**

Автор выражает глубокую признательность научным руководителям: Владимиру Васильевичу Александрову и Степану Степановичу Лемак за их поддержку, знания и наставления. Также автор благодарит друзей и коллег, принимавших ключевое участие в становлении молодежного коллектива лаборатории МОИДС МГУ: Даниила Бурлакова, Маргариту Белоусову, Екатерину Проскурякову, Максима Мироненко, Виталия Хлопикова, Анну Кручинину и Василия Латонова.

Автор искренне благодарит Александра Семеновича Позняка и коллег из его научной группы: Исаака Чаиреза Орья и Ольгу Андрианову за сотрудничество, обмен опытом и множество новых идей и концепций.

Автор выражает благодарность Александру Моисеевичу Формальскому и Дмитрию Игоревичу Бугрову за дельные советы и помощь с корректурой работы. Особую признательность автор выражает коллегам и студентам молодежного коллектива, с кем совместно работали над вопросами создания алгоритмов динамической имитации: Григорию Бугрий, Артуру Мухамедову, Максиму Дементьеву, Павлу Сухочеву, Михаилу Мартынову, Михаилу Макарову и Никите Соколову.

За дискуссии по теме диссертации и ценные замечания автор благодарит Михаила Юрьевича Проскурякова.

Теплые слова благодарности автор адресует своей матери Марине Юрьевне Мирошниченко, отцу Александру Викторовичу Чертополохову и бабушке Тамаре Михайловне Соляник за любовь, поддержку и веру в успех работы.

## Обозначения

БПФ — быстрое преобразование Фурье.

ВР — виртуальная реальность.

Визуальная имитация — одна из технологий виртуальной реальности, нацеленная на построение визуального окружения для пользователя.

Динамическая имитация — одна из технологий виртуальной реальности, нацеленная на создание динамических стимулов для механорецепторов пользователя (в том числе вестибулярного аппарата).

ИСП — интегральный скользящий режим.

Концевой эффектор — рабочий орган робота-манипулятора.

ПД-регулятор — пропорционально-дифференцирующий метод управления.

Рендеринг — термин в компьютерной графике, обозначающий процесс получения изображения по модели с помощью компьютерной программы. Часть процесса визуализации.

СР — скользящий режим.

Субградиент — обобщение градиента на случай негладких функций.

Трекинг — отслеживание движений динамического объекта.

УСГ — усредненный субградиент.

Фовеальный рендеринг — метод рендеринга, при котором отображается в максимальной четкости только та часть изображения, на которую смотрит пользователь.

Фруструм — усеченный конус видимости в системах визуализации.

Чаттеринг — проблема низкочастотных колебаний (автоколебаний), обусловленных статическими и динамическими неидеальностями релейных элементов.

## Список литературы

1. Alexandrov V. V., Chertopolokhov V. A. Human eye's sharp vision area stabilization for VR headsets // SID Symposium Digest of Technical Papers. — Vol. 52. — 2021. — P. 376–378.
2. Andre N., Martinet P. Vision-based self-calibration and control of parallel kinematic mechanisms without proprioceptive sensing // Intelligent Service Robotics. — 2009. — Vol. 2, № 2. — P. 71–80.
3. Bellmann T., Heindl J., Hellerer M., Kuchar R., Sharma K., Hirzinger G. The DLR robot motion simulator part I: Design and setup // 2011 IEEE International Conference on Robotics and Automation. — 2011. — P. 4694–4701.
4. Bellmann T., Otter M., Hirzinger G. The DLR robot motion simulator part II: Optimization based path-planning // 2011 IEEE International Conference on Robotics and Automation. — 2011. — P. 4702–4709.
5. Belousova M. D., Chertopolokhov V. A., Kruchinina A. P. Using virtual reality technology for bobsleigh simulation // 12th International Symposium on Computer Science in Sport Proceedings. — Moscow, 2019. — P. 126–127.
6. Bertsekas A., Dimitri P. Constrained Optimization and Lagrange Multiplier Methods. — New York: Academic Press, 1982. — 400 p. — ISBN: 1-886529-04-3.
7. Chang C., Cui W., Gao L. Foveated holographic near-eye 3D display // Optical Express. — 2020. — Vol. 28, № 2. — P. 1345–1356.
8. Chen K. Y. Robust optimal adaptive sliding mode control with the disturbance observer for a manipulator robot system // International Journal of Control, Automation and Systems. — 2018. — № 16. — P. 1701–1715.
9. Chertopolokhov V. A., Andrianova O. G., Hernandez-Sanchez A., Mireles C., Poznyak A. S., Chairez I. Averaged sub-gradient integral sliding mode control design for cueing end-effector acceleration of a two-link robotic arm // ISA Transactions. — 2023. — Vol. 133. — P. 134–146.

10. Chertopolokhov V. A., Khinenzon E. V., Pravdivtsev A. V., Grigoryan G. L. High-frequency optical solution for human motion tracking and eye-tracking // Journal of Physics: Conference Series. — 2020. — Vol. 1560, №1, Art. 012020. — P. 1–8. <https://doi.org/10.1088/1742-6596/1560/1/012020>.
11. Chertopolokhov V. A. On the Problem of Synchronization of Virtual and Real Movements for Virtual Reality Systems // Journal of Physics: Conference Series. — 2021. — Vol. 2056, №1, Art. 012052. — P. 1–7. <https://doi.org/10.1088/1742-6596/2056/1/012052>.
12. Du G., Zhang P. Online Serial Manipulator Calibration Based on Multisensory Process Via Extended Kalman and Particle Filters // IEEE Transactions on Industrial Electronics. — 2014. — Vol. 61, № 12. — P. 6937–6946.
13. Duc-An P., Duc-Toan N. False cue influence on motion cue quality for 10 motion cueing algorithms // Science Progress. — 2021. — Vol. 104, №3. <https://doi.org/10.1177/00368504211036857>.
14. Faulwasser T., Weber T., Zometa P., Findeisen R. Implementation of nonlinear model predictive path-following control for an industrial robot // IEEE Transactions on Control Systems Technology. — 2017. — № 25. — P. 1505–1511.
15. Fisher S. S., Humphries J., McGreevy M., Robinett W. The Virtual Environment Display System // Proceedings of the Workshop on Interactive 3-D Graphics. — 1986. — Vol. 1. — P. 77–87.
16. Hamayun M. T., Christopher E., Alwi H. Integral Sliding Mode Control Variable // Studies in Systems, Decision and Control. — Springer, 2016. — Vol. 61. — P. 17–37.
17. Hernandez-Sanchez A., Chairez I., Poznyak A. S., Andrianova O. G., Chertopolokhov V. A. Cueing end-effector acceleration of a two-link robotic arm by dynamic averaged sub-gradient integral sliding mode control // Asian Journal of Control. — 2022. — P. 1–11. <https://doi.org/10.1002/asjc.2994>.

18. Hong J., Kim Y., Hong S., Shin C., Kang H. Near-eye foveated holographic display // *Imaging and Applied Optics* 2018. — Optical Society of America, 2018. — P. 1–4. <https://doi.org/10.1364/3D.2018.3M2G.4>.
19. Kaplanyan A., Sochenov A., Leimkuhler T., Okunev M., Goodall T., Rufo G. DeepFovea: Neural Reconstruction for Foveated Rendering and Video Compression using Learned Statistics of Natural Videos // *ACM Transactions on Graphics (TOG)*. — 2019. — Vol. 36. — P. 1–13. <https://doi.org/10.1145/3355089.3356557>.
20. Khalil W., Besnard S. Self Calibration of Stewart-Gough Parallel Robots Without Extra Sensors // *IEEE Transactions on Robotics and Automation*. — 1999. — Vol. 15, № 6. — P. 1116–1121.
21. Kornilova L. N., Temnikova V. V., Sagalovitch S. V., et al. Effect of otoliths upon function of the semicircular canals after long-term stay under conditions of microgravity // *Physiological Journal*. — 2007. — Vol. 93, № 2. — P. 28–138.
22. Kruchinina A. P., Chertopolokhov V. A., Grigoryan G. L., Belyaev V. V. Prediction of saccadic eye movement for foveated rendering // *SID Symposium Digest of Technical Papers*. — Vol. 51. — 2020. — P. 803–806.
23. LaValle S. M. *Virtual Reality*. — Cambridge University Press, 2023. — 426 p. — ISBN: 9781108182874.
24. Lee W., Kim J., Cho J. A driving simulator as a virtual reality tool // *Proceedings. 1998 IEEE International Conference on Robotics and Automation*. — Vol. 1. — 1998. — P. 71–76.
25. Lemak S. S., Chertopolokhov V. A., Makarov M. A. Set of dynamic restrictions imposed on robotic arm-based motion simulator phase coordinates // *Journal of Physics: Conference Series*. — 2021. — Vol. 1864, №1. Art. 012133. — P. 1–6. <https://doi.org/10.1088/1742-6596/1864/1/012133>.
26. Masory O., Wang J., Zhuang H. Kinematic modeling and calibration of a Stewart platform // *Advanced Robotics*. — 1997. — Vol. 11, № 5. — P. 519–539.

27. Matthews G. P., DeCarlo R. A. Decentralized tracking for a class of interconnected nonlinear systems using variable structure control // IEEE Transactions on Industrial Electronics. — 1988. — № 24. — P. 187–193.
28. Matthews S., Uribe-Quevedo A., Theodorou A. Rendering Optimizations for Virtual Reality Using Eye-Tracking // 2020 22nd Symposium on Virtual and Augmented Reality (SVR). — 2020. — P. 398–405.
29. Mazuryk T., Gervautz M. Virtual reality — history, applications, technology and future. — 1996. — P. 1–72.
30. Meng Y., Zhuang H. Self-Calibration of Camera-Equipped Robot Manipulators // The International Journal of Robotics Research. — 2001. — Vol. 20, № 11. — P. 909–921.
31. Miermeister P., Lächele M., Boss R., Masone C., Schenk C., Tesch J., Kerger M. et al. The CableRobot simulator large scale motion platform based on cable robot technology // 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). — 2016. — P. 3024–3029.
32. Nazin A. V., Nemirovsky A. S., Tsybakov A. B., Juditsky A. B. Algorithms of robust stochastic optimization based on mirror descent method // Automation and Remote Control. — 2019. — Vol. 80, № 9. — P. 1607–1627.
33. Nechvatal J. Critical Distances: A Study of the Affinity Between Artistic Ideologies Based in Virtual Reality and Previous Immersive Idioms. — LAP Lambert Academic Publishing AG and Co KG, 2009. — 592 p. — ISBN: 978-3838304458.
34. Nelder J.A., Mead R. A simplex method for function minimization // The Computer Journal. — 1965. — Vol. 7. — P. 308–313.
35. Nelson K., Black T., Creighton D., Naha-vandi S. A Simulation-Based Control Interface Layer for a High-Fidelity Anthropomorphic Training Simulator // Interservice/Industry Training, Simulation & Education Conference (I/ITSEC). — 2010. — P. 1–11.
36. Ng A., Chan L., Lau H. A Study of Cybersickness and Sensory Conflict Theory Using a Motion-Coupled Virtual Reality System // 2018 IEEE



- Conference on Virtual Reality and 3D User Interfaces (VR). — 2018. — P. 643–644.
37. Oliveira W. R., Matheus A., Rodamilans G., Nicola R., Arjoni D., Trabasso L. G., Villani E., da Silva E. T. Evaluation of the Pilot Perception in a Robotic Flight Simulator with and without a Linear Unit // AIAA Scitech 2019 Forum. — 2019. — P. 7–13. <https://doi.org/10.2514/6.2019-0713>.
  38. Page R. L. Brief history of flight simulation // SimTecT 2000 proceedings. — 2000. — P. 11–17.
  39. Pan Y., Yang C., Pan L., Yu H. Integral sliding mode control: performance, modification, and improvement // IEEE Transactions on Industrial Informatics. — 2018. — № 14. — P. 3087–3096.
  40. Parra-Vega V., Arimoto S., Liu Y. H., Hirzinger G., Akella P. Dynamic sliding PID control for tracking of robot manipulators: theory and experiments // IEEE Transactions on Robotics and Automation. — 2003. — № 19. — P. 967–976.
  41. Plancher G., Tirard A., Gyselinck V., Nicolas S., Piolino P. Using virtual reality to characterize episodic memory profiles in amnesic mild cognitive impairment and Alzheimer's disease: influence of active and passive encoding // Neuropsychologia. — 2012. — Vol. 50, № 5. — P. 592–602.
  42. Poznyak A. S. Advanced Mathematical Tools for Control Engineers: Volume 1: Deterministic Systems. — Elsevier, 2010. — Vol. 1. — 808 p. — ISBN: 9780080446745.
  43. Poznyak A. S., Nazin A. V., Alazki H. Integral Sliding Mode Convex Optimization in Uncertain Lagrangian Systems Driven by PMDC Motors: Averaged Subgradient Approach // IEEE Transactions on Automatic Control. — 2021. — Vol. 66, № 9. — P. 4267–4273.
  44. Rahimi H. N., Howard I., Cui L. Neural adaptive tracking control for an uncertain robot manipulator with time-varying joint space constraints // Mechanical Systems and Signal Processing. — 2018. — №112. — P. 44–60.

45. Rizzo A., Buckwalter J., Neumann U., Kesselman C., Thiebaut M. Basic issues in the application of virtual reality for the assessment and rehabilitation of cognitive impairments and functional disabilities // *CyberPsychology & Behavior*. — 1998. — Vol. 1, № 1. — P. 59–78.
46. Rolfe J. M., Staples K. J. *Flight Simulation*. — Cambridge University Press, 1988. — 282 p. — ISBN: 0521020891.
47. Rothbaum B. O., Hodges L. F., Ready D., Graap K., Alarcon R. D. Virtual reality exposure therapy for Vietnam veterans with posttraumatic stress disorder // *Clinical Psychiatry*. — 2001. — Vol. 62, № 8. — P. 617–622.
48. Savitzky A., Golay M. J. E. Smoothing and differentiation of data by simplified least squares procedures // *Analytical Chemistry*. — 1964. — Vol. 36, № 8. — P. 1627–1639.
49. Schroeder R. Defining Virtual Worlds and Virtual Environments // *Journal of Virtual Worlds Research*. — 2008. — Vol. 1, №1. <https://doi.org/10.4101/jvwr.v1i1.294>.
50. Sharma A., Agarwal M., Sharma A., Dhuria P. Motion Capture Process, Techniques and Applications // *International Journal on Recent and Innovation Trends in Computing and Communication*. — 2013. — Vol. 1, № 4. — P. 251–257.
51. Slotine J. J., Weiping L. Adaptive manipulator control a case study // *IEEE Transactions on Industrial Electronics*. — 2005. — № 33. — P. 1392–1400.
52. Soyka F., Teufel H., Beykirch K., Giordano P.R., Butler J., Nieuwenhuizen F., Buelthoff H. Does Jerk Have to be Considered in Linear Motion Simulation? // *Proceedings of the AIAA Modeling and Simulation Technologies Conference (AIAA 2009)*. — 2009. — P. 1–4. <https://doi.org/10.2514/6.2009-6245>.
53. Su Y., Zheng C., Mercorelli P. Robust approximate fixed-time tracking control for uncertain robot manipulators // *Mechanical Systems and Signal Processing*. — 2020. — №135. — P. 2891–2902.

54. Sutherland I. E. A head-mounted three-dimensional display // *Proceedings of the December 9-11, 1968, Fall Joint Computer Conference, Part I.* — Vol. 33. — Association for Computing Machinery, 1968. — P. 757–764.
55. System user manual. ARTtrack, TRACKPACK & DTrack. — Weilheim: A.R.T. GmbH, 2018. — 275 p.
56. Tcha-Tokey K., Loup-Escande E., Christmann O., Richir S. Effects of Interaction Level, Framerate, Field of View, 3D Content Feedback, Previous Experience on Subjective User eXperience and Objective Usability in Immersive Virtual Environment // *International Journal of Virtual Reality.* — 2017. — Vol. 17, № 3. — P. 27–51.
57. Teufel H., Nusseck H., Beykirch K., Butler J., Kerger M., Bulthoff H. MPI Motion Simulator: Development and Analysis of a Novel Motion Simulator // *AIAA Modeling and Simulation Technologies Conference and Exhibit.* — AIAA, 2007. — P. 1–14. <https://doi.org/10.2514/6.2007-6476>.
58. Truszczyński O., Kowalczyk K. The new Polish centrifuge as a dynamic flight simulator. New application and ideas // *The Polish Journal of Aviation Medicine and Psychology.* — 2012. — № 3. — P. 71–80.
59. Vega R., Alexandrov V. V., Alexandrova T. B., Soto E. Mathematical model of the cupula-endolymph system with morphological parameters for the axolotl (*Ambystoma tigrinum*) semicircular canals // *Open Medical.* — 2008. — Vol. 2. — P. 138–148.
60. Venrooij J., Pretto P., Katliar M., Nooij S., Nesti A., Lachele M., de Winkel K. et al. Perception-based motion cueing: validation in driving simulation // *DSC 2015 Europe: Driving Simulation Conference & Exhibition.* — 2015. — P. 153–161.
61. Veron H., Hezel P. J., Southard D. A. Head mounted displays for virtual reality // *Proceedings of the SPIE.* — 1994. — Vol. 2218. — P. 41–50.
62. Watson A. Deep Learning Techniques for Super-Resolution in Video Games // *Neural and Evolutionary Computing.* — 2020. — P. 1–4. <https://doi.org/10.48550/arXiv.2012.09810>.

63. White A. D., Hall J. R. Initial Validation of an R&D Simulator With Large Amplitude Motion // Flight Mechanics Panel Symposium on Piloted Simulation Effectiveness. — 1992. — P. 1–24.
64. Zehner B. Mixing Virtual Reality and 2D Visualization - Using Virtual Environments as Visual 3D Information Systems for Discussion of Data from Geo- and Environmental Sciences // GRAPP 2010 - Proceedings of the International Conference on Computer Graphics Theory and Applications. — 2010. — P. 364–369.
65. Zhang L., Liu L., Wang Z., Xia Y. Continuous finite-time control for uncertain robot manipulators with integral sliding mode // IET Control Theory and Applications. — 2018. — № 12. — P. 1621–1627.
66. Zhao L., Dai L., Xia Y., Li P. Attitude control for quadrotors subjected to wind disturbances via active disturbance rejection control and integral sliding mode control // Mechanical Systems and Signal Processing. — 2019. — Vol. 129. — P. 531–545.
67. Zhuang H. Self-Calibration of Parallel Mechanisms with a Case Study on Stewart Platforms // IEEE Transactions on Robotics and Automation. — 1997. — Vol. 13, № 3. — P. 340–348.
68. Александров В.В. Тестирование качества стабилизации нестационарных движений // Вестник МГУ. Серия 1: Математика. Механика. — 1997. — № 3. — С. 51–54.
69. Александров В. В., Блаженнова-Микулич Л. Ю., Гутierrez-Ариас И. М., Лемак С. С. Максимальное тестирование точности стабилизации и седловые точки в геометрических играх // Вестник МГУ. Серия 1: Математика. Механика. — 2005. — № 1. — С. 55–69.
70. Александров В. В., Бугров Д. И., Лемак С. С., Лебедев А. В., Тихонова К. В., Чертополохов В. А., Шуленина Н. Э. Новые задачи физико-математического практикума. Часть II. Тестирование качества сближения устройства спасения космонавта с международной космической станцией. — Москва : Издательство попечительского совета механико-математического факультета МГУ, 2015. — 56 с. — ISBN: 978-5-4294-0008-2.

71. Александров В. В., Герра Л., Каленова И. Н., Трифонова А. В. Минимаксная стабилизация и максиминное тестирование линейных управляемых систем // Вестник МГУ. Серия 1: Математика. Механика. — 1999. — № 5. — С. 58–65.
72. Александров В. В., Ишлинский А. Ю., Садовничий В. А., Глазков Ю. Н., Воронин Л. И. Математические задачи динамической имитации аэрокосмических полетов. — Издательство Московского Университета, 1995. — 160 с.
73. Александров В. В., Лемак С. С. Максиминный контроль качества коррекции траектории управляемого спуска с орбиты // Труды конференции по теории колебаний и управлению (Москва, 29 ноября 2000 года). — 2000. — С. 47–49.
74. Александров В. В., Лемак С. С. Алгоритмы динамической имитации для стенда-тренажера пилотируемых полетов на базе центрифуги с управляемым кардановым подвесом // Фундаментальная и прикладная математика. — 2018. — Т. 22. — С. 3–16.
75. Александров В. В., Лемак С. С., Парусников Н. А. Оптимизация динамики управляемых систем. — М.: Издательство Московского университета, 2000. — 304 с. — ISBN: 5-211-04267-0.
76. Александров В. В., Лемак С. С., Садовничий В. А. Максиминное тестирование качества визуального управления устройством спасения космонавта // Вестник Московского государственного университета леса. — 2005. — № 4. — С. 6–11.
77. Александров В. В., Садовничий В. А., Александрова Т. Б., Сидоренко Г. Ю., Шуленина Н. Э. Формирование выходной информации в вертикальных полукружных каналах // Доклады Академии наук. — 2011. — Т. 441, № 1. — С. 122–125.
78. Бахвалов Н. С. Численные методы. Том I: анализ, алгебра, обыкновенные дифференциальные уравнения. — М.: Наука, 1975. — 632 с.
79. Белоусова М. Д., Чертополохов В. А., Кручинина А. П. О целенаправленном движении руки человека для построения копирующего управления антропоморфным роботом // Биомеханика-2018: материалы XIII

- Всероссийской (с международным участием) конференции. — Издательство Южного федерального университета Ростов-на-Дону, 2018. — С. 10–11.
80. Бобылев А. Н., Болотин Ю. В., Воронов А. В., Кручинин П. А. О двух модификациях метода наименьших квадратов в задаче восстановления утерянной информации системы видеоанализа по показаниям акселерометра // Российский журнал биомеханики. — 2012. — Т. 16, № 1. — С. 89–101.
81. Бугров Д. И., Лебедев А. В., Чертополохов В.А. Оценка угловой скорости вращения тела при помощи системы трекинга // Вестник Московского университета. Серия 1: Математика. Механика. — 2014. — № 1. — С. 68–71.
82. Булгаков Б. В. О накоплении возмущений в линейных колебательных системах // Доклады Академии Наук СССР. — 1946. — Т. 51. — С. 5–8.
83. Бурлаков Д. С., Латонов В. В., Чертополохов В. А. Идентификация параметров модели подвижной платформы опорного типа // Фундаментальная и прикладная математика. — 2018. — Т. 22, № 2. — С. 73–88.
84. Васильев Ф. П. Методы решения экстремальных задач. — М.: Наука, 1980. — 552 с. — ISBN: 978-00-1424477-0.
85. Демьянов В. Ф., Рубинов А. М. Приближенные методы решения экстремальных задач. — Издательство ЛГУ, 1968. — 182 с.
86. Долгов П. П., Киршанов В. Н., Иродов Е. Ю., Гаврик И. Н., Коренной В. С., Онуфриенко Ю. И., Чудинов А. П. и др. Экспериментальные исследования возможности применения систем виртуальной реальности при моделировании на центрифуге управляемого движения лунохода // Пилотируемые полеты в космос. — 2020. — № 3(36). — С. 91–108.
87. Доценко В. И., Штефанова О. Ю., Якушев А. Г. Коэффициент стабилизации взора при компьютерном анализе нистагма как объективный интегральный показатель оценки вестибулярной функции // Клиническая нейрофизиология и нейрореабилитация: Материалы 2-й научно-практической конференции с международным участием. — 2014. — С. 62–64.

88. Исаев В. К., Сонин В. В. Об одной модификации метода Ньютона численного решения краевых задач // Вычислительная математика и математическая физика. — 1963. — Т. 3, № 6. — С. 1114–1116.
89. Ишлинский А. Ю. Ориентация, гироскопы и инерциальная навигация. — М.: Наука, 1976. — 670 с.
90. Красовский А. А., Кудиенко А. В. Пилотажно-навигационные и комплексные тренажеры. — ВВИА им. проф. Н.Е. Жуковского, 1984. — 204 с.
91. Кручинина А. П., Латонов В. В., Чертополохов В. А. Обзор технологий визуальной имитации в тренажерных системах // Пилотируемые полеты в космос. — 2019. — № 3. — С. 89–107.
92. Кручинина А. П., Чертополохов В. А. Применение технологий виртуальной реальности в спорте // Материалы Всероссийской научно-практической конференции по вопросам спортивной науки в детско-юношеском спорте и спорте высших достижений. — ГКУ "ЦСТиСК" Москомспорта, 2016. — С. 196–201.
93. Кручинина А. П., Якушев А. Г. Анализ математической модели вестибулярного стимулятора – качелей Хилова // Труды XVII Международного научно-технического семинара / ГУАП СПб. — Алушта, 2008. — 09. — С. 255–256.
94. Крылов А. Н. Избранные труды. — Издательство Академии наук СССР, 1958. — 804 с.
95. Крючков Б. И., Усов В. М. Создание моделей виртуальной реальности, как способ обучения космонавтов взаимодействию с роботом-помощником экипажа и как условие определение потенциальных областей его полезного применения // Труды международной научно-технической конференции "Экстремальная робототехника" / Политехника-сервис. — 2013. — С. 230–244.
96. Крючков Б. И., Усов В. М., Михайлюк М. В. Применение методов захвата движений в космической робототехнике при инженерно-психологическом проектировании человеко-машинного взаимодействия // Пилотируемые полеты в космос. — 1986. — № 4. — С. 57–58.

97. Лемак С. С. Максиминный контроль качества стабилизации космических объектов. Диссертация на соискание степени доктора физико-математических наук. — 2004. — 174 с.
98. Лесовский А. С. Анализ факторов, влияющих на изменение перегрузки при полете самолета в турбулентной атмосфере // Научный вестник МГТУ ГА. — 2006. — № 110. — С. 20–25.
99. Орлов В. И. Вестибулярная функция. — СПб.: Наука, 1998. — 248 с. — ISBN: 5-02-026097-5.
100. Осипов М. П. Системы виртуальной реальности. — Нижегородский университет, 2012. — 48 с.
101. Отчет № 2368. Анализ математических моделей полукружных каналов вестибулярной системы. — Институт механики МГУ, 1980. — 63 с.
102. Парусников Н. А., Морозов В. М., Борзов В. И. Задача коррекции в инерциальной навигации. — Издательство Московского Университета, 1982. — 176 с.
103. Петросян Л. А., Зенкевич Н. А., Шевкопляс Е. В. Теория игр. 2-е издание. — БХВ-Петербург, 2012. — 432 с. — ISBN: 978-5-9775-0484-3.
104. Покровский В. М., Коротько Г. Ф. Физиология человека. Том 2. — Медицина, 1998. — Т. 2. — 368 с. — ISBN: 5-225-04729-7.
105. Понтрягин Л. С., Болтянский В. Г., Гамкрелидзе Р. В., Мищенко Е. Ф. Математическая теория оптимальных процессов. — М.: Наука, 1983. — 392 с.
106. Раев О. Н. Генезис киноискусства // Инновационные технологии в кинематографе и образовании. — 2020. — С. 54–62.
107. Руководство по критериям квалификационной оценки тренажерных устройств имитации полета. Том 1. Самолеты. Издание четвертое. — ИКАО, 2015. — 770 с. — ISBN: 978-92-9249-930-3.
108. Садовничий В. А., Александров В. В. Динамическая имитация аэрокосмических полетов // Математическое моделирование. — 1994. — Т. 6, № 6. — С. 5–8.



109. Садовничий В. А., Александров В. В., Александрова Т. Б., Вега Р., Сото Э. Информационный процесс в латеральных полукружных каналах // Доклады Академии наук. — 2011. — Т. 436, № 1. — С. 129–132.
110. Садовничий В. А., Александров В. В., Лемак С. С., Лобашов Е. С. Алгоритмы динамической имитации аэрокосмического полета на стендах типа центрифуги // Современные проблемы математики и механики. — 2009. — Т. 1, № 1. — С. 134–146.
111. Сухочев П. Ю., Чертополохов В. А. Full-immersive vr в кино: совмещение 360 видео и виртуальной сцены // Инновационные технологии в кинематографе и образовании: III Международная научно-практическая конференция. Материалы и доклады. — Москва, 2016. — С. 28–35.
112. Тимаков В. М., Прошин И. А., Прошкин В. Н. Совершенствование динамических стендов авиационных тренажеров на базе гидроприводов // Мехатроника, автоматизация, управление. — 2009. — С. 18–22.
113. Уткин В. А. Короткий комментарий к методу А.Ф. Филиппова продолжения решения на границе разрыва // Автоматика и телемеханика. — 2015. — № 5. — С. 165–174.
114. Филиппов А. Ф. Дифференциальные уравнения с разрывной правой частью. — М.: Наука, 1985. — 224 с.
115. Хэмминг Р. У. Цифровые фильтры. — Москва : Советское радио, 1980. — 224 с.
116. Чертополохов В. А., Григорян Г. Л. Применение технологии фовеального рендеринга для оптимизации производительности сложных сцен виртуальной исторической реконструкции // Исторические исследования в контексте науки о данных: информационные ресурсы, аналитические методы и цифровые технологии : материалы международной конференции. — ООО "МАКС Пресс", 2020. — С. 371–377.
117. Ярбус А. Л. Роль движений глаз в процессе зрения. — Москва : М.: Наука, 1965. — 166 с.