

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
АМУРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
(ФГБОУ ВО «АмГУ»)

Факультет математики и информатики
Кафедра информационных и управляющих систем
Направление подготовки /специальность 09.04.04 – Программная инженерия
Направленность (профиль) образовательной программы Управление разработкой программного обеспечения

ДОПУСТИТЬ К ЗАЩИТЕ
Зав. кафедрой
_____ А.В. Бушманов
«_____» _____ 2023 г.

МАГИСТЕРСКАЯ ДИССЕРТАЦИЯ

на тему: Компьютерное моделирование движения руки

Исполнитель студент группы 157-ом	_____	Л.Е. Осипенко
	(подпись, дата)	
Руководитель доцент, канд.техн.наук	_____	А.В. Бушманов
	(подпись, дата)	
Руководитель научного содержания программы магистратуры профессор, доктор техн.наук	_____	И.Е. Еремин
	(подпись, дата)	
Нормоконтроль доцент, канд.техн.наук	_____	Л.В. Никифорова
	(подпись, дата)	
Рецензент доцент, канд.техн.наук	_____	А.Н. Рыбалев
	(подпись, дата)	

Благовещенск 2023

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
АМУРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
(ФГБОУ ВО «АмГУ»)

Факультет математики и информатики
Кафедра информационных и управляющих систем

УТВЕРЖДАЮ

Зав. кафедрой

_____ А.В. Бушманов

подпись

И.О.Фамилия

« _____ » _____ 2023 г.

ЗАДАНИЕ

К выпускной квалификационной работе студента Осипенко Любови Евгеньевны

1. Тема выпускной квалификационной работы: Компьютерное моделирование движения руки

(утверждена приказом от 21.02.2023 № 442-уч)

2. Срок сдачи студентом законченной работы (проекта) 20.06.2023 г.

3. Исходные данные к выпускной квалификационной работе: предметная область, отчеты по практической подготовке, результаты выступления на научной конференции, реальное манипуляционное устройство (рука)

4. Содержание выпускной квалификационной работы (перечень подлежащих разработке вопросов): общая характеристика исследуемой задачи, алгоритмическое и программное обеспечение решения поставленной задачи, компьютерная реализация алгоритма для компьютерного моделирования движения руки

5. Дата выдачи задания: 30.01.2023 г.

6. Руководитель выпускной квалификационной работы: Бушманов Александр Вениаминович, доцент, канд.техн.наук

(фамилия, имя, отчество, должность, уч. степень, уч. звание)

Задание принял к исполнению (30.01.2023): _____

(подпись студента)

РЕФЕРАТ

Магистерская диссертация содержит 101 с., 69 рисунков, 7 таблиц, 54 источника

МАНИПУЛЯЦИОННОЕ УСТРОЙСТВО, ТРЕХМЕРНАЯ МОДЕЛЬ, SOLIDWORKS, MATLAB, SIMSCAPE, КИНЕМАТИКА И ДИНАМИКА МАНИПУЛЯТОРА, ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ, ПОЛЬЗОВАТЕЛЬСКИЙ ИНТЕРФЕЙС

Целью работы является создание разработка и исследование модели реального манипуляционного устройства, по внешнему виду напоминающего человеческую руку Разработанная компьютерная модель манипулятора может использоваться на различных производствах для исследования перемещений и угловых конфигураций в рабочей зоне манипулятора.

В данной работе описывается технология разработки трехмерной модели с помощью пакета прикладных программ Matlab, библиотеки SimScape и системы автоматизированного проектирования SolidWorks. Моделирование основано на решении задач кинематики и динамики с помощью которых на выходе модели будут получены необходимые координаты по заданным оператором угловым конфигурациям и угловые конфигурации для достижения координат. Также созданы графические интерфейсы для взаимодействия пользователя с моделью руки. Полученные результаты могут быть использованы на реальном устройстве.

СОДЕРЖАНИЕ

Введение	7
1 Общая характеристика исследуемой задачи	9
1.1 Обоснование актуальности темы проводимого исследования	9
1.2 Предметная область и объект исследования	10
1.3 Обзор существующих методов математического решения рассматриваемой задачи	13
1.4 Обзор программных сред для моделирования механизмов роботов	15
1.5 Постановка задачи для компьютерного моделирования	23
2 Алгоритмическое и программное обеспечение решения поставленной задачи	24
2.1 Предлагаемые алгоритмы для компьютерного моделирования движения руки	26
2.1.1 Алгоритм управления звеньями руки по заданию угла поворота	26
2.1.2 Алгоритм управления звеньями руки по заданию положения	28
2.1.3 Алгоритм управления кистью руки	30
2.2 Выбор профильного программного обеспечения и обзор его возможностей	31
2.3 Обоснование выбора программно-технического обеспечения	42
3 Компьютерная реализация алгоритма для компьютерного моделирования движения руки	45
3.1 Основные этапы разработки программного продукта	45
3.1.1 Описание математической модели объекта исследования	45
3.1.2 Построение модели руки на основе реального манипуляционного устройства	51
3.1.3 Программная реализация алгоритма управления звеньями руки	59
3.1.4 Программная реализация алгоритма по заданию положения руки	66
3.1.5 Программная реализация алгоритма по управлению кистью руки	74

3.1.6 Описание пользовательского интерфейса и возможностей оператора	76
3.2 Тестирование программного продукта	80
3.2.1 Тестирование программы для управления звеньями руки	81
3.2.2 Тестирование программы для управления положением руки	84
3.2.3 Тестирование программы для управления кистью руки	87
3.3 Анализ достоверности и практической значимости результатов моделирования	89
Заключение	90
Библиографические ссылки	91
Библиографический список	96

ОПРЕДЕЛЕНИЯ, ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ

ДХ-представление – представление Денавита-Хартенберга;

МР – манипуляционный робот;

ОС – операционная система;

ПК – персональный компьютер;

ПО – программное обеспечение;

САПР (CAD) – система автоматизированного проектирования;

СУ – система управления;

GUI – графический интерфейс пользователя (graphical user interface);

ODE – открытый физический движок (open dynamics engine).

ВВЕДЕНИЕ

Сегодня в различных сферах деятельности широко используются тип роботов такой как роботы-манипуляторы или манипуляционные устройства, исполнительными устройствами которых являются манипуляторы (механические руки). Манипулятором принято называть тип роботов с функциями, аналогичными функциями человеческой руки. Их область применения – это автоматизированные производства, которые являются важным направлением развития промышленности. Автоматизация применяется, к примеру, в автомобильной промышленности, электротехнике и электронике, металлообработке, машиностроение, медицине, сельском хозяйстве и т.д.

Роботы-манипуляторы и системы их управления являются одним из основных элементов автоматизации производств. Они способны улучшить эффективность, повысить качество выпускаемой продукции, уменьшить фактор человеческой ошибки. По своей сути роботы-манипуляторы – это электромеханический объект, в котором существуют зависимые и независимые взаимосвязи между элементами. Также у манипуляторов сложная кинематическая структура и существует необходимость синхронного управления звеньями. Управление манипулятором происходит с помощью запрограммированного микроконтроллера и сервоприводов. Разработанные системы управления необходимы для осуществления взаимодействия манипуляционного устройства с пользователем (оператором), для выполнения поставленных технологических задач.

Увеличение производства манипуляционных роботов и развитие систем их управления приводит к необходимости создания компьютерных моделей манипуляторов, которые помогут разработчикам систем управления проводить эксперименты, тестирование алгоритмов управления, наблюдать за поведением модели, внедрять в технологический процесс отечественные разработки, обучаться управлению на модели.

Целью проводимой квалификационной работы является разработка и исследование модели реального манипуляционного устройства, по внешнему виду напоминающего человеческую руку, включающая:

- математическую модель, описывающую движения механической руки в разных формах движения;
- 3D моделирование с применением прикладных программ.

Для достижения поставленной цели необходимо решить следующие задачи:

- проанализировать предметную область, изучить объект исследования и подходы к его моделированию;
- исследовать особенности кинематики и динамики объекта исследования;
- сформировать математическую модель для разных положений объекта исследования на основе полученных расчетов и данных;
- спроектировать программный продукт для компьютерного моделирования объекта исследования;
- проверить работоспособность программного продукта тестированием его функциональности;
- осуществить проверку адекватности модели и её уточнение.

Результаты исследования в основном могут быть применены при разработках систем управлений антропоморфными манипуляторами, их тестировании, а также в учебном процессе в робототехнике.

1 ОБЩАЯ ХАРАКТЕРИСТИКА ИССЛЕДУЕМОЙ ЗАДАЧИ

1.1 Обоснование актуальности темы проводимого исследования

На сегодняшний день широкое применение во многих сферах деятельности человека получили манипуляционные роботы. Развитие в области робототехники и систем управлений антропоморфными манипуляторами приводит к необходимости развития математического аппарата, разработки кинематических, динамических и компьютерных трехмерных моделей, которые более полно учитывают многозвенную структуру и подобные факторы.

Компьютерное моделирование многозвенных твердых систем является перспективным направлением изучения пространственных механизмов в том числе манипуляторов. По мере увеличения сложности проектируемых систем их аналитическое исследование становится все более трудным, а их прототипирование становится все более дорогим, поэтому компьютерное моделирование механических систем часто является основным (или единственным) доступным методом исследования. Компьютерное моделирование – метод, позволяющий строить модели, описывающие процессы так, как они проходили бы в действительности с применением компьютерных технологий. Такую модель можно «проиграть» во времени, а также проводить исследования над ней.

Исследование поможет разработчикам систем управления или оператору манипуляционного устройства промоделировать необходимое положение МР и получить необходимые конфигурации. В данном квалификационном исследовании моделирования движения манипуляционного устройства, похожего на человеческую руку, модель используется в качестве замены экспериментов на реальном объекте, что связано с рядом преимуществ:

- моделирование имеет значительно меньшие финансовые и временные затраты по сравнению с физическими экспериментами;
- все виртуальные эксперименты, проводимые над моделью повторяемы, что не может быть гарантировано для реального объекта исследования;

- модель полностью управляема и контролируема;
- удобность применения в учебных целях, так как исключает возможности повредить реальное устройство;
- позволит сократить взаимодействие на реальном устройстве и сохранить ему долгий срок службы.

1.2 Предметная область и объект исследования

Предметной областью проводимого исследования является робототехника – прикладная наука, занимающаяся разработкой и применением роботов и робототехнических систем для облегчения трудовой деятельности человека, а также является основой развития производства. Робототехническая система (робот) – это совокупность механизмов и устройств, формирующих единую систему в том числе имеющую распределенную сетевую структуру и выполняющую рабочие операции, связанные со сложными пространственными перемещениями, способную к целесообразному поведению в условиях изменяющейся внешней обстановки без непосредственного участия человека за счет автоматизации процессов [1].

Роботы и создаваемые на их основе робототехнические системы являются достаточно сложными устройствами. Включаемые в робототехнические системы роботы и обслуживаемое ими оборудование имеют разнообразные связи между собой: информационные, кинематические, программные и многие другие [2]. На данный момент времени разного типа роботы применяется в широких областях: в сборочных операциях, перемещениях и изготовлении на производствах, обслуживание оборудования, работа в экстремальных условиях и т.д.

Для работы где предусмотрена двигательная функция и необходимо заменить человека применяют манипулятор – совокупность кинематических звеньев, представляющих собой пространственный рычажной механизм и систему приводов. Кинематическая цепь манипулятора может быть замкнутой и разомкнутой. В последнем случае манипулятор считается антропоморфным человекоподобным роботом. Манипулятор совершает манипуляции, которые осуществля-

ются под управлением программируемого автоматического устройства или человека-оператора. Такие манипуляции аналогичны действиям руки человека. Манипулятор может быть, как самостоятельным устройством, так и находиться в составе более сложного роботизированного комплекса. Звенья манипулятора имеют соединения, допускающие вращательное или поступательное движение.

Основание манипулятора – это его база, которая закреплена и остается неподвижной, другие звенья совершают движения с помощью управления приводами. Привода воздействуют на звенья в соответствии с управляющими сигналами, которые формируются по заданным законам или алгоритмам движения от системы управления. Существует два типа заданий на перемещения:

- перемещение от одной точки к другой за заданный период времени – позиционное управление;

- перемещение захвата по заданной траектории и заданному во времени закону движения по этой траектории – контурное управление.

В первом случае задаются начальные и конечные точки, а траектория и скорость его не определяются. Точек на пути перемещения может быть большое количество. Примером такого управления служит цикловое управление. В этом случае захват движется между двумя точками. Во втором случае задается закон движения во времени. Это зависимость трех координат x , y , z от времени.

Объектом исследования в квалификационной работе является манипуляционное устройство или «рука», которая представляет собой перепрограммируемый многофункциональный манипулятор, предназначенный для осуществления определенных, заранее заданных перемещений предметов, деталей, инструментов или специальных приспособлений с целью выполнения различных работ. Манипуляционное устройство, модель которого разрабатывается в квалификационной работе является антропоморфным манипулятором с достаточно сложной кинематической конструкцией из множества звеньев, сопряженных посредством вращательно подвижных сочленений.

По внешнему виду такое устройство напоминает человеческую руку и в своем составе имеет: базу вокруг которой вращается рука, плечо, локоть, предплечье, запястье и кисть. Для управления скоростью и положением звеньев реального манипуляционного устройства используется микроконтроллер Arduino UNO R3, который обеспечивает перемещение его звеньев по заданному алгоритму, а для управления звеньями манипулятора используются устройство с электрическим мотором – сервомоторы (сервоприводы). Внешний вид реального антропоморфного манипуляционного устройства показан на рисунке 1. На рисунке показаны основные звенья манипулятора и вид движения между звеньями. Звенья соединены между собой с помощью шарниров, которые вращаются вокруг собственной системы координат.

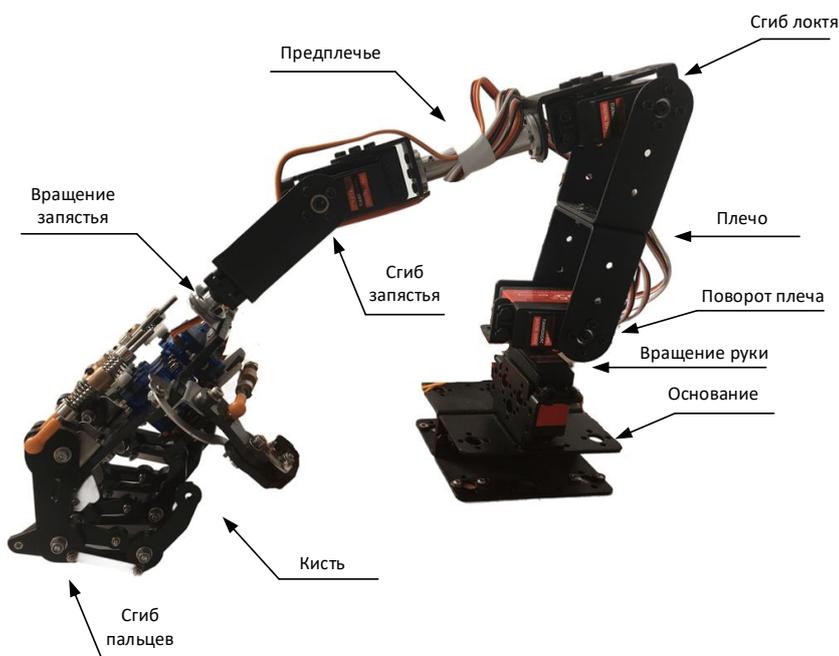


Рисунок 1 – Антропоморфное манипуляционное устройство

Важной характеристикой для манипуляционного устройства служит число степеней свободы. Минимально необходимое число переносных степеней подвижности для перемещения рабочего органа в любую точку свободной рабочей зоны равно трем [3]. Однако для повышения ловкости, достижения сложных схем движения, таких как обход препятствий, или увеличения скорости, манипуляторы часто оснащаются несколькими избыточными степенями подвижности.

Это, конечно, значительно увеличивает сложность и стоимость робота. В среднем современные манипуляторы имеют 4 – 6 степеней подвижности, но существуют и манипуляторы и с 8 – 9 такими степенями. В данной руке 6 степеней свободы. В создаваемой модели возможные перемещения относительно начального (нулевого) положения перечислены ниже:

- вращение манипулятора. Вращение манипулятора осуществляется вокруг основания от -180° до 180° ;
- сгиб плеча. Управление сгибом плеча происходит в пределах от -100° до 120° ;
- сгиб локтя. Сгибание локтевой части манипулятора происходит в пределах от -270° до 60° ;
- сгиб запястья. Сгибание кисти манипулятора происходит в пределах от -100° до 120° ;
- вращение кисти. Вращение кисти манипулятора происходит в пределах от -180° до 180° ;
- движения в кисти (сгиб пальцев).

В квалификационной работе создается математическая модель на основе объекта исследования – это приближенное описание на языке математики (отображение на математическом языке) основных закономерностей и наиболее важных свойств, присущих исследуемому оригиналу (объекту). Построение математической модели будет происходить по экспериментальным данным, полученным в ходе измерения на реальном объекте. С помощью созданной математической модели будет отстраиваться компьютерная модель руки в 3D с помощью прикладных программ. Далее с помощью программной среды оператор программы сможет приводить руку в заданные положения. Движения модели можно «проиграть» во времени, отследить её поведение на графиках и на видео.

1.3 Обзор существующих методов математического решения рассматриваемой задачи

Для моделирования движения будет разрабатываться компьютерная модель руки. Компьютерная модель руки будет создана с помощью прикладных

программ, ей можно будет управлять и наблюдать за движением. Модель даёт возможность более эффективно выполнить исследование, проектирование и оптимизацию движений руки. Создание компьютерной модели – это метод решения задач анализа и синтеза сложных систем с помощью отстроенной модели с применением компьютерных технологий. За счет моделирования появляется возможность получить количественные и качественные результаты, которые позволяют обнаружить ранее неизвестные свойства сложных систем, такие как их структура, динамика развития, устойчивость и полнота. Виртуальные среды позволяют быстро отлаживать различные программные алгоритмы, поскольку в них не используется реальное оборудование, что облегчает тестирование на реальном объекте, а также реализацию и оптимизацию путей движения.

Для реализации компьютерного моделирования необходимо изучить объект исследования, а именно составить его математическую модель. Построить хорошую математическую модель, точно описывающую оригинал, непросто. Процесс математического моделирования исследуемого оригинала условно можно разделить на этапы формирования задачи, выбор метода решения, идентификация и проверка [4].

На первом этапе формируется постановка задачи исследования:

- определяются искомые характеристики оригинала и исходные данные;
- описывается и анализируется проблемная ситуация;
- определяются законы, которыми описывается поведение оригинала;
- устанавливаются связи, действующие между отдельными элементами оригинала;
- определяются требования к исходной информации и математической модели.

На втором этапе следует приступить к созданию математической модели объекта исследования, отражающая важные особенности строения модели. Описанная математическая модель анализируется с помощью теоретических методов и дает важные характеристики и предварительную структуру оригинала. Также в этом этапе подбирается метод решения задачи математической модели,

а затем разрабатывается алгоритм для его решения с применением компьютерных технологий. Для этого необходим подбор подходящего пакета прикладных программ. Далее разрабатываются расчетные эксперименты и проводятся фактические расчеты

Третий этап моделирования обычно связан с идентификацией или проверкой описанной математической модели и с её улучшением. При идентификации модели переменные и параметры модели выбираются и оцениваются на основе статистических данных, полученных в результате наблюдений и экспериментов. Если обнаруживается, что математическая модель недостаточно точна, создается усовершенствованная математическая модель. Затем проводятся расчеты для дальнейших результатов моделирования, пока не будут достигнуты необходимые результаты.

Выбор метода решения задачи производимого во втором этапе определяется тем, к какому классу относится решаемая математическая задача, а класс решаемой математической задачи, в свою очередь, определяется классом математической модели. Классификацию разделяет по разным признакам. В нашем случае модель по фактору времени является физической, динамической, по характеру изучаемых процессов детерминированной, по способу реализации является имитационной моделью. Имитационными называются компьютерные модели, представляющие собой программу или комплекс программ, позволяющий воспроизводить процессы функционирования объекта в разных условиях.

1.4 Обзор программных сред для моделирования механизмов роботов

Для компьютерного моделирования движения руки необходимо выбрать среду, в которой модель будет создаваться. Перечислим наиболее популярные среды имитационного моделирования на сегодняшний день, позволяющие работать с робототехникой:

– Webots – это профессиональный симулятор роботов и является настольным приложением с открытым исходным кодом. Он предоставляет полную среду разработки для моделирования, программирования и имитации роботов.

Webots использует ODE, который позволяет точно имитировать физические свойства объектов, таких как скорость, инерции и трения.

– Gazebo – ПО для исследований в области робототехники. Обеспечивает моделирование динамики и кинематики роботизированных систем и механизмов. Программа Gazebo имеет гибкий дизайн и удобный интерфейс, поддерживающий одновременную работу с несколькими устройствами. Для работы в данной среде необходимо хорошее знание C++ и UNIX-систем [5].

– CoppeliaSim (V-REP) – это среда компьютерного моделирования, позволяющая пользователю создавать собственные модели динамических или статических объектов с их визуальным отображением, например, приводы или подвижные конструкции роботов. V-REP использует кинематические и физические движки (Bullet, ODE, Vortex, Newton Game Dynamics) для моделирования твердого тела. Каждый из них имеет более высокую точность моделирования в определенных задачах, но все они хорошо решают общие задачи моделирования роботов [6].

– SimSpark – это универсальная система моделирования для различных многоагентных симуляций. Он поддерживает разработку физических симуляций для исследований в области искусственного интеллекта и робототехники с использованием платформы приложений с открытым исходным кодом. Обычно используется в академических исследованиях и образовании. В данной программе используется агентное моделирование и в одной симуляции могут участвовать несколько программных агентов.

– Simscape Multibody (SimMechanics) – пакет расширения Simulink среды Matlab, которая предоставляет среду для имитационного моделирования многозвенных трёхмерных механических систем, таких как роботы, подвески транспортных средств, строительное оборудование и т.п., которые собираются из блоков, представляющих собой тела, кинематические соединения, а также силовые элементы, в блок-схему [7]. Модели сборок, включая массы, значения различных физических свойств, к примеру, инерции, размерности могут импортироваться из САПР.

Анализ наиболее распространенных сред моделирования применяющихся в робототехнике по критериям представлен в таблице 1.

Таблица 1 – Анализ сред моделирования для робототехники

Критерии	Webots	Gazebo	CoppeliaSim	SimSpark	Simscape Multibody
1	2	3	4	5	6
Разработчик (страна), год выхода	Cyberbotics Ltd.(Швейцария), 1998	Open Robotics (США), 2004	Coppelia Robotic (Швейцария), 2010	О. Обст и др. (+26) (Германия), 2003	The MathWorks (США)
Подходы	Системная динамика	Системная динамика	Системная динамика	Агентное моделирование	Системная динамика
Построение модели	Графическое, программное	Графическое, программное	Графическое, программное	Графическое, программное	Графическое, программное
Основной пользовательский интерфейс	GUI	GUI	GUI	GUI, сокет	GUI
Операционная среда	Windows 10, 64-разрядная версия Linux, macOS X	Linux	Windows, Linux, macOS	Linux, macOS, Windows	Windows, Linux, macOS
Анимация	Есть	Есть	Есть	Есть	Есть
Интеграция других платформ	MATLAB, ROS (Robotics Operating System), имеет API C/C++, Java, Python или URBI	C/C++, Python	Имеет API C/C++, Python, Java, Lua, Matlab или Octave	C++, Ruby	Язык программирования Matlab

1	2	3	4	5	6
Импорт работок САПР	Да, в формате VRML	Да, с помощью плагина SW2URDF или в формате SDF или URDF, но требует небыстрой настройки	Да, в формате URDF	Только графики сцены Ruby	Да, с применением CAD-транслятора
Имитация физических свойств объекта	Есть, использует ODE	Есть, использует ODE	Есть, использует Bullet, ODE, Vortex, Newton Game Dynamics	Есть, использует ODE	Есть, с помощью специализированных блоков
Возможность подключения к Arduino	Да, но требует изменения кода и настройки	Неизвестно	Неизвестно	Неизвестно	Есть, достаточно выполнить команду targetinstaller
Доступность	Свободная лицензия Apache 2	Свободная лицензия Apache 2	Двойная лицензия (коммерческая или GPL)	Свободная лицензия GPL	Проприетарная лицензия
Личный опыт использования	Нет	Нет	Нет	Нет	Есть

Для выбора среды моделирования движения руки из приведенного анализа выделим основные достоинства и недостатки сред. Анализ достоинств и недостатков для всех сред представлен в таблице 2.

Таблица 2 – Достоинства и недостатки представленных сред

Название среды	Достоинства	Недостатки
Webots	Бесплатный продукт, обладает инструментарием для тестирования моделей роботов и различных интерфейсов; поддерживает программное управление посредством популярных языков программирования.	Обладает ограниченным функционалом инструментария для конструирования модели робота.
Gazebo	Бесплатный продукт, обладает симулятором и инструментарием для создания конструкций.	Совместим только с Ubuntu Linux; разработка ведется на языке разметки xml и является крайне трудоемким процессом; слабо поддерживается разработчиками.
CoppeliaSim	Огромное количество возможностей для разработки и модификации робототехнических систем; поддержка популярных языков программирования и различных интерфейсов.	Использует скриптовый язык Lua, подключение других языков программирования проблематично; проблема с изменением положения объекта по оси Z.
SimSpark	Бесплатный продукт; поддерживает программное управление посредством популярных языков.	Обладает ограниченным функционалом инструментария для конструирования модели робота; приводы заменяются эффекторами SimSpark.
Simscape Multibody	Формулирует и решает уравнения движения для полной механической системы; есть возможность импортировать в свою модель полные сборки САПР; поддерживает подключения к Arduino.	Не встроенная библиотека – требует установки и дополнительной настройки в САПР; необходимость регистрации на сайте разработчика.

Для 3D моделирования изделий существуют программы САПР. Система автоматизированного проектирования (САПР) – сложный комплекс средств,

предназначенный для автоматизации проектирования. На сегодняшний день существует большое количество САПР, которые решают разные задачи. Они подразделяются на базовые и легкие, среднего уровня и тяжелые. Облегченная или базовая САПР предназначена для 2D-проектирования и создания 3D-моделей без использования сложных построений. Основные программы из базовых САПР:

- AutoCAD – программа для САПР, разработанная компанией Autodesk. Является одной из самых популярных систем. С ее помощи можно проектировать как в двумерной, так и в трехмерной среде;

- BricsCAD – альтернатива AutoCAD, объединяет в себе двумерное и трехмерное моделирование;

- Компас-График – универсальная отечественная САПР, для моделирования в 3D КОМПАС-График работает в связке с системой трехмерного моделирования КОМПАС-3D;

- nanoCAD – отечественная альтернатива AutoCAD. Включает модуль 3D – многофункциональный модуль, объединяющий два подхода: параметрическое и прямое объемное моделирование.

Системы автоматизированного проектирования среднего уровня самые популярные системы на рынке [8]. Эти системы оптимальны по соотношению цена/производительность, могут решать большинство задач проектирования и удовлетворяют потребности большинства клиентов. Основные программы из среднего уровня САПР:

- Autodesk Inventor – это профессиональный комплекс для 3D моделирования изделий объединяющий два подхода: трехмерное твердотельное и поверхностное параметрическое;

- SolidWorks – программа САПР, разработанная компанией Dassault Systems. Является трехмерным программным комплексом для автоматизации конструкторских работ промышленного предприятия;

– SolidEdge – это система трехмерного моделирования машиностроительных изделий, разработчиком является Siemens PLM Software. Используется технология моделирования на основе конструктивных элементов и дерева построения, так и технология вариационного прямого моделирования.

Тяжелые САПР предназначены для построения сложных изделий аэрокосмической промышленности, судостроении и т.д. Функционально тяжелая САПР может делать тоже, что и промежуточная система, но архитектура и используемые алгоритмы сильно отличаются в лучшую сторону. Основные программы из тяжелого уровня САПР:

– PTC Creo (Pro/ENGINEER) – САПР двухмерного и трехмерного параметрического проектирования сложных изделий от компании PTC. САПР PTC Creo широко используется в самых разных областях проектирования;

– NX – ведущая САПР от разработчиков Siemens PLM Software, используемая для разработки сложных изделий со сложной геометрией и большим количеством тесно связанных компонентов;

– Catia – САПР для проектирования сложных комплексных изделий от компании Dassault Systemes. В основном используется в авиа и кораблестроении.

Для дальнейшего выбора среды САПР для построения 3D модели руки необходимо проанализировать представленные выше программы по основным параметрам. Для анализа выбирается одна программа из каждой категории сложности. Анализ САПР для построения модели приведен в таблице 3.

Таблица 3 – Анализ САПР для построения модели

Критерии	AutoCAD	SolidWorks	PTC Creo
1	2	3	4
Разработчик (страна), год выхода	Autodesk(США), 1982	Dassault Systemes (Франция), 1995	PTC (США), 2011

1	2	3	4
Подходы к моделированию	Твердотельное, поверхностное и полигональное моделирование	Гибридное параметрическое моделирование: твердотельное моделирование, моделирование поверхностей, каркасное моделирование и их комбинация.	Прямое, параметрическое твердотельное моделирование
Операционная среда	Microsoft Windows, macOS, iOS, Android и Windows Phone	Windows	Windows
Читаемые форматы файлов	DXF и DWG, а также импорт 3DS, DGN, JT, SAT, PDF, STEP и некоторых других.	STL и SolidWorks Drawing	STEP, IGES, DXF, STL, VRML, AutoCAD DWG, DXF (импорт 3D с 2D), ACIS.
Создаваемые форматы файлов	DXF и DWG, а также экспорт DGN, SAT, STL, IGES, FBX и некоторых других	SolidWorks Drawing	PRT и ASM
Удобный интерфейс пользователя	Не достаточно интуитивен	Да	Да
Рекомендованные системные требования	ЦП: с тактовой частотой выше 3 ГГц. ГП: с 4 Гбайт видеопамати и поддержкой DirectX 11. RAM: 16 Гбайт. Жесткий диск: 7 Гбайт. ОС: Windows 8.1 / 10 64-разрядная.	ЦП: 4 ядра с тактовой частотой 3,3 ГГц или выше. RAM: 16 Гбайт или больше. Жесткий диск: от 5 Гбайт до 50 Гбайт / SSD-накопитель ГП: NVIDIA / AMD ОС: 64-битная Windows 7 SP1, Windows 8.1, Windows 10.	ЦП: от 2 ядер, 2 ГГц x86 64-разрядная версия. RAM: От 3 до 8 Гбайт. Жесткий диск: 2 Гбайта для установки. ГП: библиотека OpenGL. ОС: Windows 10 (64-разрядная версия) Linux 7 или 8 amd64.

На основе проведенных анализов представленных продуктов будет осуществлен выбор сред для построения трехмерной модели руки и моделирования её движения

1.5 Постановка задачи для компьютерного моделирования

Разрабатываемый программный продукт реализует человеко-машинный интерфейс и предоставляет возможность пользователю задавать траекторию движения или управлять звеньями манипуляционного устройства по отдельности. понимания того, как устроено манипуляционное устройство, каковы структура и свойства, как описываются его законы движения, а также для отработки алгоритмов разработчиком программ для управления манипуляционным устройством.

Программный продукт для моделирования движения создается для:

- понимания того, как устроено манипуляционное устройство, каковы структура и свойства, как описываются его законы движения;
- определения наилучших способов управления при заданных параметрах;
- получения угловой и пространственной конфигурации манипуляционного устройства;
- отработки алгоритмов разработчиком программ для управления манипуляционным устройством.

Программный продукт должен предусматривать следующую функциональность:

- интеграцию CAD-модели в систему управления моделью;
- результаты модели точно отображают действительное положение звеньев руки (манипулятора) согласно составленной математической модели;
- отображение результата работы модели и получение данных о текущем положении руки и захвата кисти в виде графиков доступных пользователю;
- управление точкой захвата через задание её положения;
- управление отдельными звеньями манипуляционного устройства.

После выполненной функциональности требуется сравнить результаты модели с расчетами и проверить её адекватность.

2 АЛГОРИТМИЧЕСКОЕ И ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ РЕШЕНИЯ ПОСТАВЛЕННОЙ ЗАДАЧИ

Механический манипулятор можно рассматривать как разомкнутую цепь, которая состоит из нескольких твердых тел (звеньев), последовательно соединенных вращательными или поступательными сочленениями, приводимыми в движение силовыми приводами [9]. Для воздействия на объекты необходимо чтобы один конец цепи был соединен с основанием, а другой свободен и оснащен рабочим органом, в данном случае кистью подобную человеческой руке, которая позволит выполнять различные технологические операции. Относительное движение сочленений передается звеньям, в результате рука занимает в пространстве заданное положение.

Существует две основные задачи кинематики: прямая и обратная. Первая задача заключается в том, чтобы по заданным параметрам звеньев привести руку в заданное положение и определить его. На рисунке 2 показана общая схема задач прямой и обратной кинематики.

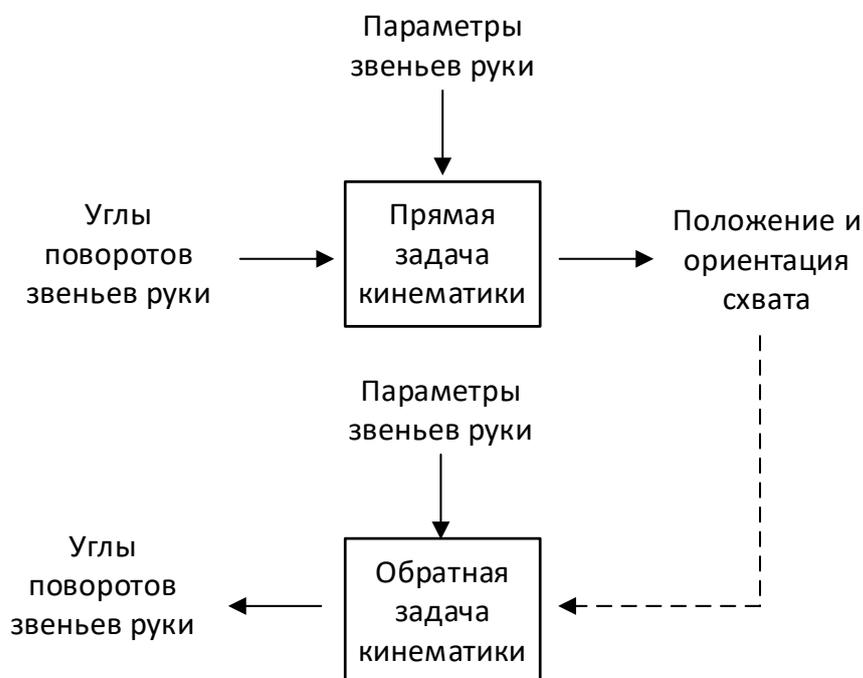


Рисунок 2 – Прямая и обратная задачи кинематики

Обратная задача необходима для того чтобы по заданному положению и при известных параметрах руки определить углы поворота звеньев руки.

Для понимания того, как должна функционировать разрабатываемая система управления моделью руки нужно составить диаграмму прецедентов. В системе моделирования движения руки в качестве актора (действующего лица) выступает пользователь/оператор модели руки. Прецедент соответствует отдельному сервису системы, определяет один из вариантов ее использования и описывает типичный способ взаимодействия пользователя с системой [10]. Диаграмма прецедентов представлена на рисунке 3.

В качестве прецедентов выделены следующие функциональные части системы:

- запуск моделирования;
- визуализация 3D модели;
- управление звеньями руки;
- согнуть кисть/локоть/плечо;
- повернуть руку/кисть;
- управление точкой захвата руки;
- управление кистью;
- вычисление управляющих команд;
- получение ответа от модели;
- вывод информации о текущем положении руки.

Пользователю предоставляется три варианта управления движением руки: управление звеньями руки, управление точкой захвата руки и управление кистью. Для управления звеньями руки у пользователя есть возможность задать угол поворота каждого привода, отвечающего за определённое звено в руке. При управлении точки захвата пользователь задает координаты точки или координаты траектории следования, при задании которых рука перемещается до их достижения. Управление кистью заключается в том, что пользователю предоставляется возможность сгибать пальцы кисти руки.



Рисунок 3 – Диаграмма прецедентов

Все команды решаются в отстроенной модели и после ответа системы пользователь получает визуализацию движения и возможность просмотреть результаты вычислений в виде графиков, на которых отображены изменение углов поворотов и изменение координат захвата во времени.

2.1 Предлагаемые алгоритмы для компьютерного моделирования движения руки

2.1.1 Алгоритм управления звеньями руки по заданию угла поворота

Для создания программного продукта требуется реализовать возможность управления 3D моделью с помощью спроектированной системы управления. После правильной отстройки модели и успешного запуска визуализация трехмерной модели должна открыться в отдельном окне. Пользователю предоставляется возможность задавать углы поворота для каждого звена отдельно. Для этого в специально отведенное поле вводится число из заданного диапазона начального и конечного положения каждого вала модели сервопривода, отвечающего за поворот звена. Диапазон на каждом сервоприводе установлен в зависимости от воз-

возможности сгиба руки. Число представляет последовательность цифр, которая перезаписывается во временный массив до момента обнаружения символа передачи команды «Ок». Поворот происходит после задания значения угла поворота на определенный вал модели сервопривода.

В модели руки и в реальном устройстве находится пять сервоприводов:

- сервопривод для вращения руки вокруг базы;
- сервопривод для поворота плеча;
- сервопривод для сгиба локтя;
- сервопривод для сгиба запястья;
- сервопривод для вращения запястья (кисти).

Информация о положении точки захвата в координатах и о занятых угловых положениях отображается в графиках зависимости от времени. В окне визуализации также можно увидеть все перемещения, производимые пользователем программы на всем интервале времени моделирования.

В случае задания числа, не входящего в угловой диапазон, выводится сообщение об ошибке. Задание угловых значений напрямую относится к решению прямой задачи кинематики. После того как пользователь программы вводит значение угла модель реагирует на изменение и визуализирует движения. Для расчета координат, которую заняла точка захвата (конечный эффектор) руки необходимо сформировать матрицу преобразования для прямой кинематики манипулятора, где значения трех строк последнего столбца будут означать x , y , z координаты. После работы с моделью необходимо остановить моделирование и посмотреть график изменений значений углов и координат во времени.

На рисунке 4 показан алгоритм работы программы для задания угла поворота каждого звена руки.

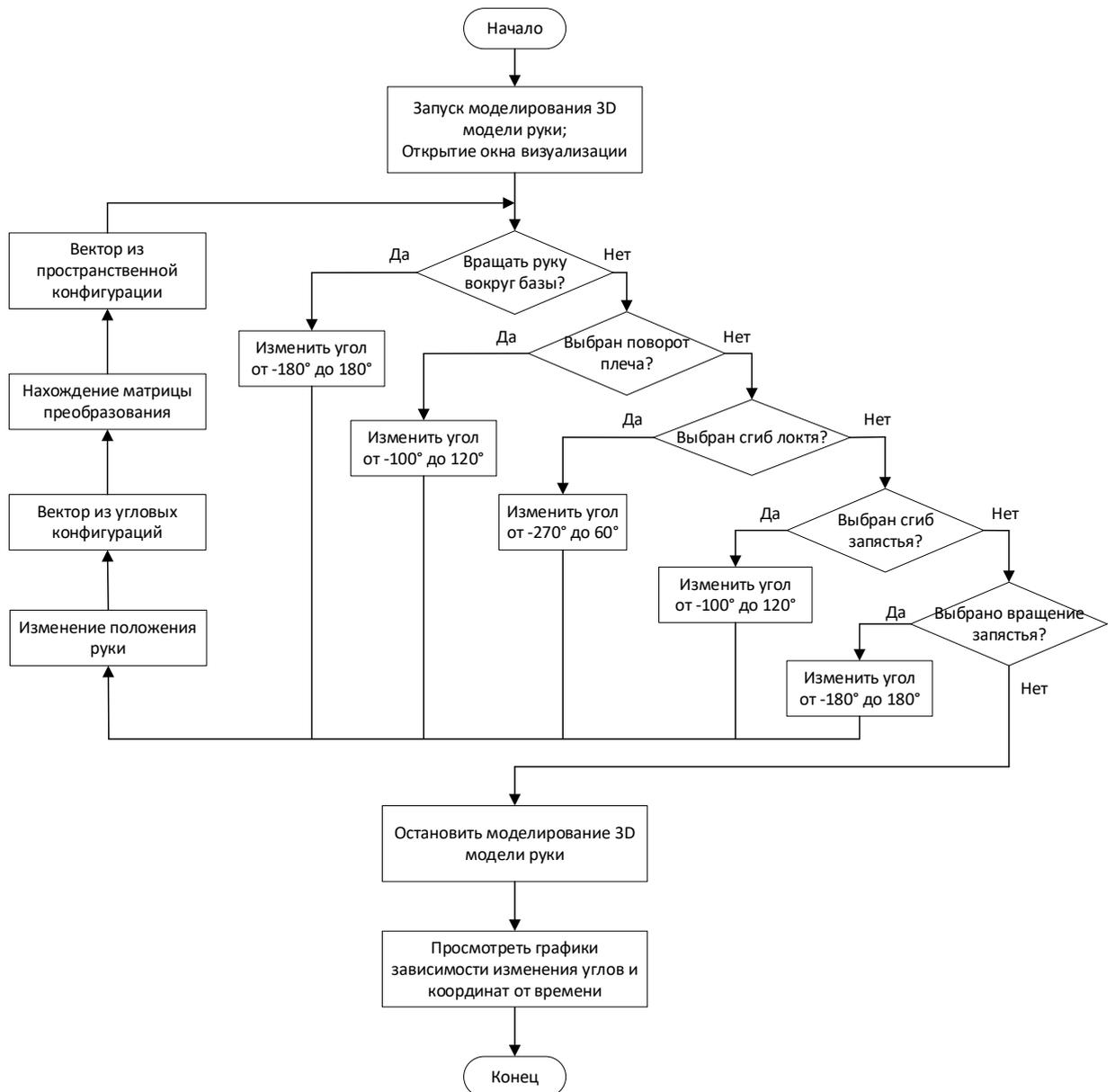


Рисунок 4 – Алгоритм управления звеньями руки

Данный тип управления относится к ручному, так как все углы поворота задаются вручную. Для того чтобы модель сама рассчитывала угловые значения необходимо создать алгоритм управления рукой по положению.

2.1.2 Алгоритм управления звеньями руки по заданию положения

Задание положения для модели руки заключается в том, чтобы ввести данные координат на перемещения точки захвата. Данные вводятся в специальный блок для ввода данных. Данные о перемещении и ориентации захвата обрабатываются и вычисляются в модели, затем происходит перемещение всех зависимых

от перемещения звеньев руки. Перемещение также можно наблюдать в окне визуализации. Результаты обработки команды выведены на графике изменения положения и углов поворота. На рисунке 5 показан алгоритм управления точкой захвата по положению.

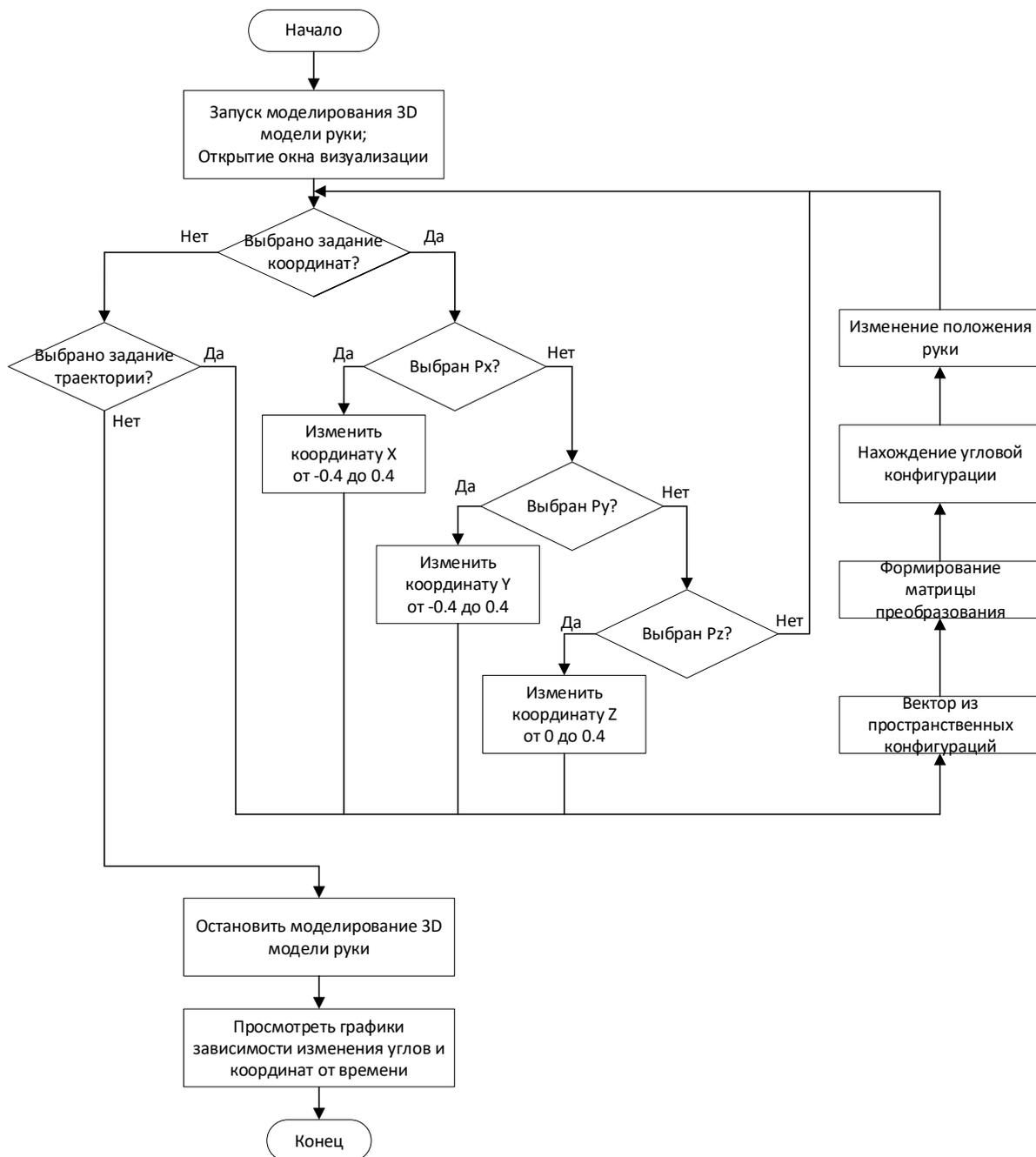


Рисунок 5 – Алгоритм управления положением руки

2.1.3 Алгоритм управления кистью руки

Управление кистью руки заключается в управлении пальцами руки. Каждый палец управляется за счет поворота вала сервопривода. На них будет подаваться угол поворота от пользователя программы. В зависимости от поданного в модель значения будет происходить сжатие или разжатие пальца. Сжатие и разжатие всех пальцев происходит до тех пор, пока валы сервоприводов не достигнут заданных критических значений угла поворота вала (в программе искусственно заданы минимальное и максимальное отклонения, как бы это соответствовало реальной руке человека). На рисунке 6 показан алгоритм управления кистью руки.

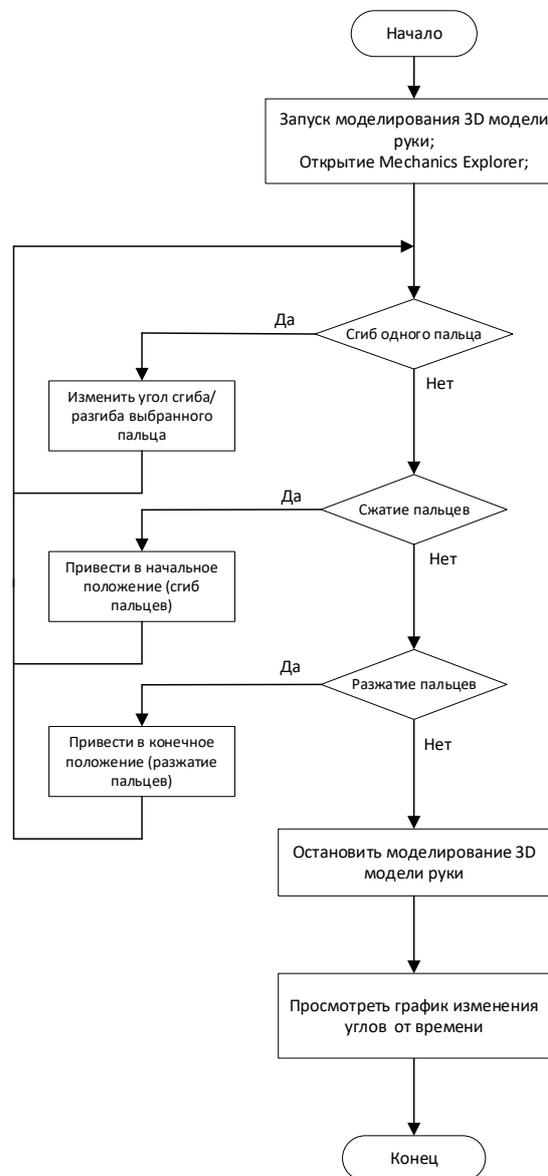


Рисунок 6 – Алгоритм управления кистью руки

Для достижения начального и конечного положения на каждый датчик вращения пальца подается необходимый угол. Далее необходимо выбрать программы в котором будут реализованы представленные алгоритмы.

2.2 Выбор профильного программного обеспечения и обзор его возможностей

Для того чтобы создать программу для компьютерного моделирования руки необходимо отстроить трехмерную модель в специальном программном комплексе SolidWorks. Данная программа – это САПР, которая активно применяется для автоматизации работ промышленного предприятия, а именно обеспечивает разработку разного рода изделий.

Сборки моделей могут быть импортированы из САПР SolidWorks, включая все значения масс, инерций, соединения и детализированную визуализацию. Затем к кинематическим парам прикладывается сила/момент или профиль движения, после чего система готова к моделированию. САПР SolidWorks позволяет осуществлять гибридное параметрическое моделирование: твердотельное моделирование, моделирование поверхностей, каркасное моделирование и их комбинации без ограничения степени сложности [11].

Построение модели ведется в SolidWorks Premium 2020 SP4.0. Включает функциональные возможности SolidWorks Standard и SolidWorks Professional, а также множество других подключаемых модулей [12].

Для моделирования движения манипулятора выбирается Simscape Multibody – это пакет расширения (библиотека) Simulink среды Matlab, которая обеспечивает среду для моделирования многосвязных трёхмерных механических систем, таких как роботы, подвески авто, строительная техника, производственное оборудование и т.п., которые собираются из блоков Simulink, представляющих собой тела, кинематические соединения, а также силовые элементы, в блочную диаграмму [13]. Программа для моделирования движения будет реализовываться в версии Matlab r2019b [14].

Для связи Matlab с отстроенной трехмерной моделью SolidWorks предусмотрен специальный плагин Simscape Multibody Link, который позволяет интегрировать CAD-сборки SolidWorks с электрическими, гидравлическими, управляющими и другими системами в среду Matlab/Simulink и моделировать всю систему [15]. В Simulink можно уточнить требования к механической системе и оптимизировать производительность на уровне системы.

Simscape Multibody Link позволяет экспортировать модели сборок SolidWorks CAD в Simscape Multibody. Чтобы загрузить и установить плагин необходимо скачать дистрибутив Simscape Multibody Link [16]. Версия дистрибутива должна соответствовать версии Matlab. Далее добавив файлы в корневую папку Matlab в командной строке Matlab необходимо ввести «install_addon ('smlink.r2019b.win64.zip '»)» для установки плагина, а также нужно зарегистрировать Matlab в качестве сервера автоматизации. После выполненных действий чтобы включить плагин необходимо в командной строке Matlab ввести «smlink_linksw».

Далее после запуска SolidWorks в строке меню SolidWorks в «Инструменты» выбрать «Добавления» как показано на рисунке 7 и установить флажок в разделе «Другие добавления» возле Simscape Multibody Link, как показано на рисунке 8.

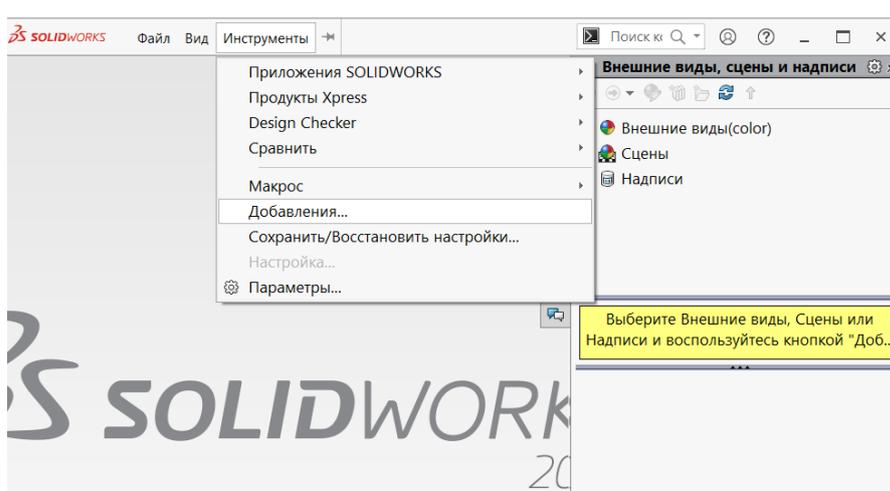


Рисунок 7 – Добавление Simscape в SolidWorks

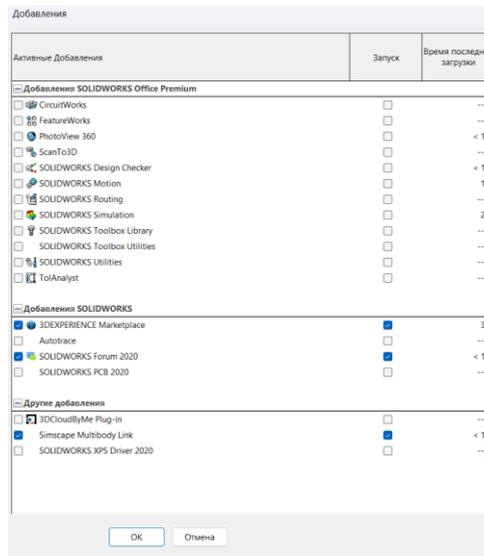


Рисунок 8 – Добавление Simscape в диалоговом окне

В итоге будет доступна возможность экспортировать сборку модели руки в программу Matlab. Плагин автоматически создает сборку многотельной имитационной модели из Solidworks при этом масса, инерция и расположение центра тяжести детали, топология и ограничения автоматически переносятся в имитационную модель Simulink.

Чтобы экспортировать модель из SolidWorks необходимо открыть файл с построенной моделью и перейти в раздел «Инструменты» как показано на рисунке 9. Далее в пункте Simscape MultiBody Link выбрать пункт Export и Simscape MultiBody. Программа предложит папку для сохранения XML-файла и файлы геометрии тела и после выбора начнет переформатировать модель. При безошибочном выполнении появляется окно с сообщением о сохранении файла XML-файла. После всего этого в диалоговом окне Matlab создается модель в Simulink с помощью функции «smimport('G:\Hand3d\Hand3d.xml')» [17].

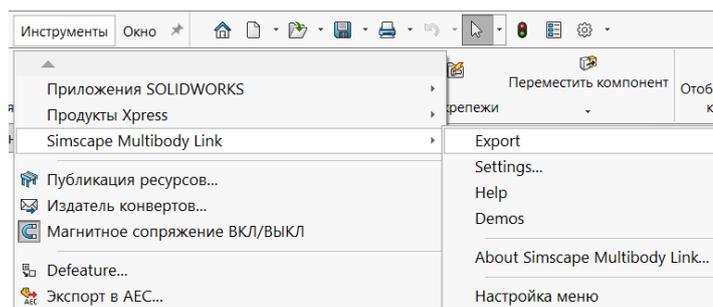


Рисунок 9 – Экспорт модели в Simscape Multibody

Результат в случае успешного выполнения функции: плагин успешно установлен, модель в Simulink построена. Результат в случае неудачного выполнения функции: вывод ошибки «неудачная установка плагина», SolidWorks выдаёт ошибку об отсутствии связи с Matlab (проверить установку сервера автоматизации). При экспорте в указанной папке создается файл с расширением XML и файлы с расширением STEP где хранятся данные о деталях и сборках.

Библиотека Simscape Multibody является частью пакета прикладных программ Matlab в частности графической среды программирования Simulink для моделирования, симуляции и анализа динамических систем. Simscape используется для исследования кинематики и динамики сложных механизмов, моделируя пространственное движение больших и сложных машин и механизмов на этапе инженерного проектирования. Это может быть эффективным для разного рода инженеров, конструкторов и т.п.

Дифференциальные уравнения описываются в виде специальных блоков Simscape, то есть все механические связи представлены удобной и наглядной блок-схемой. Блоки библиотеки Simscape Multibody – это разного рода модели механических устройств, имеющие между собой определенный вид связи. Библиотека позволяет менять положение этих моделей в соответствии с законами динамики. Модели в свою очередь описывают физическую структуру механизма и геометрические и кинематические связи компонентов. Simscape способен автоматически представлять эти структурные представления во внутренне эквивалентных математических моделях [18].

Различные механизированные устройства или системы моделируются с помощью Simscape для определения характеристики движения каждой части в пространстве относительно друг друга в определенной системе координат, для правильного определения отношений учитываются взаимосвязь и ограничения между отдельными элементами.

Библиотека Simscape состоит из двух библиотек верхнего уровня: библиотеки Foundation Library и Utilities. Внешний вид раздела библиотеки Simscape в Simulink Library Browser показан на рисунке 10.

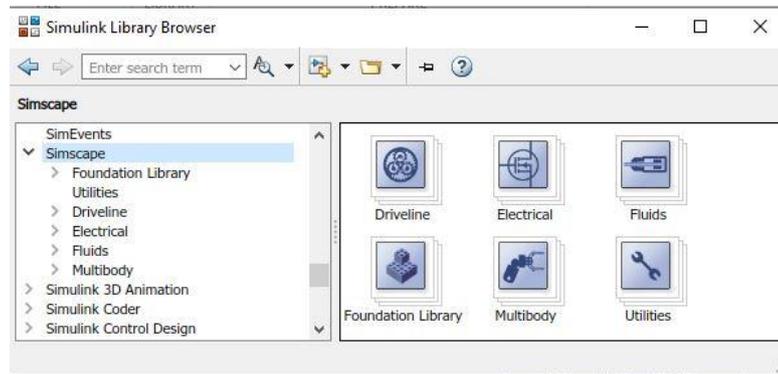


Рисунок 10 – Раздел Simscape в Simulink Library Browser

Первая библиотека считается основой и в ней содержатся основные физические элементы, базовые блоки, различные датчики. Все блоки распределены на физические дисциплины в которых применяется моделирование. Соединение тел в Simscape происходит через введенные ограничения и специальных соединений. Блоки соединений определяют максимально допустимые степени свободы вращения и смещения между соединенными телами, а блоки ограничений необходимы для того, чтобы уменьшить число степеней свободы. Основные блоки из раздела соединения показаны на рисунке 11.

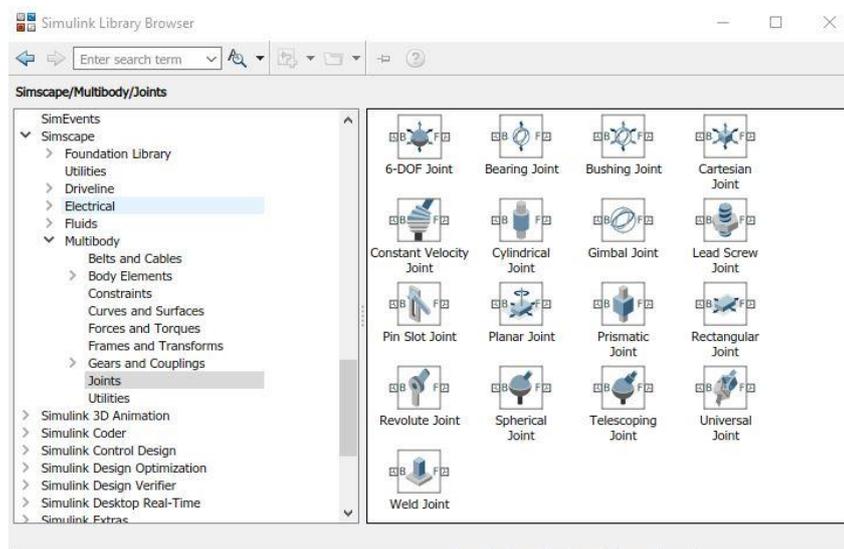
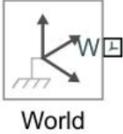
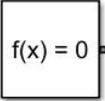
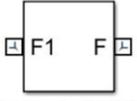
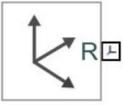


Рисунок 11 – Блоки из раздела соединений

Обзор используемых блоков библиотеки Simscape Multibody в квалификационном исследовании и их характеристика представлен в таблице 4.

Таблица 4 – Обзор блоков библиотеки Simscape Multibody

Блок	Описание блока
1	2
 <p>World</p>	<p>Блок World обеспечивает доступ к глобальной или земной системе координат. Это уникальная, фиксированная, декартова система координат, направленная по часовой стрелке, которая предопределена в каждой механической модели.</p> <p>Выход блока – порт W к которому присоединяется остальная часть тела. Порт фрейма, непосредственно соединенный с W идентифицируются с глобальной системой.</p>
	<p>Блок Mechanism Configuration задает механические параметры и параметры моделирования. Задаваемые свойства позволяют задать равномерную силу тяжести для всего механизма, а также определить дельту линеаризации.</p> <p>Порт C – это узел фрейма, который подключается к механизму с помощью соединительной линии на любом узле фрейма.</p>
	<p>Блок Solver Configuration определяет настройки решателя ode для использования в симуляции. Задаются такие параметры как допуск по консистенции равный 10^{-9}, постоянная времени фильтрации задана равной 0.001. Включен режим применять фильтрацию в 1D/3D соединениях, когда это необходимо.</p>
 <p>Transform</p>	<p>Определяет фиксированное 3D преобразование жесткого тела между двумя положениями. В разделе свойства задаются тип и параметры двух компонентов трансформации: Rotation и Translation. В них указываются данные полученные из SolidWorks.</p> <p>Порты B и F – это порты, представляющие первичное и вторичное расположение механизма соответственно.</p>
 <p>Base_1_RIGID</p>	<p>Блок, в котором находится блочная диаграмма тела Base_1_RIGID, представляет собой жесткое твердое тело, в данном случае основание руки. В него входят блоки: Reference Frame, Solid и Transform.</p>

1	2
 <p>ReferenceFrame</p>	<p>Этот блок обеспечивает систему отсчета для определения других систем координат. Система отсчета обычно инерциальная. Она может быть ускорена относительно глобальной системы. Порт R – это порт фрейма, идентифицированный как эталонный фрейм, порт фрейма, непосредственно подключенный к R, также идентифицируется как эталонный фрейм.</p>
 <p>Solid</p>	<p>Блок представляет собой твердотельные объекты с геометрией, материалами и визуальными свойствами, загруженными из CATIA, NX, SolidEdge и других форматов. Блоки SolidFile берут инерцию из геометрии и плотности, геометрии и массы или из определенного тензора инерции.</p> <p>В выпадающем списке "Свойства" указываются типы инерции, графа и каркаса, а также их конфигурации. Порт R – это порт фрейма, который представляет систему отсчета, связанную с геометрией. Для каждого нового фрейма дополнительно будет создан новый порт фрейма.</p>
 <p>Revolute</p>	<p>Блок представляет собой вращающийся шарнир, который работает между двумя деталями. Этот шарнир имеет вращательную степень свободы, которая представлена одним вращательным прототипом. Шарнир разграничивает начало двух кадров так, что оси z первичного и вторичного кадров совпадают, а оси x и y повторителя могут вращаться вокруг оси z.</p> <p>Выпадающий узел "Свойства" позволяет задать состояние, метод приведения в действие, функцию зондирования и внутренний механизм. После применения этих настроек система мониторинга отобразит соответствующие порты физических сигналов. Порты B и F – это порты фрейма, представляющие базовый и ведомый фреймы соответственно. Направление соединения определяется относительным перемещением ведомой детали по отношению к базовой детали.</p>

1	2
 <p data-bbox="316 589 437 618">Cylindrical</p>	<p data-bbox="544 271 1508 629">Этот блок обеспечивает цилиндрическое соединение между двумя деталями. Имеет поступательную и вращательную степени свободы и представлена вращательными призматическим прототипом вдоль одной оси. Это соединение может только вращаться, относительное движение в плоскости отсчета XY запрещено. Сначала рама толкателя поворачивается вокруг оси Z основания, а затем начальная координатная точка толкателя перемещается вдоль оси Z основания.</p> <p data-bbox="544 656 1508 846">В разделе «Свойства» определяются состояние, способ приведения в действие, чувствительную функцию и внутренний механизм этого первичного звена. Порты B и F – это порты, представляющие первичные и вторичные детали соответственно.</p>

Первые три блока World, Mechanism Configuration, Solver Configuration находятся в связке и задают базовые настройки для моделирования механизма. Стоит отметить то что, в Simscape обычные сигналы заменены механическими силами. Входы блока определяют начальное положение соответствующего механизма. Эти связи отвечают за передачу силовых воздействий, которыми обмениваются части механизма.

Работа любого блока раздела соединений основана на векторном методе, который применяется в кинематическом исследовании механических систем. Он позволяет определить положения, скорости и ускорения звеньев механизма [19]. Так, например, блок Revolute, который имеет одну вращательную степень свободы также использует векторный метод. Работа данного блока описывается матрицей направляющих косинусов вида:

$$M_{ba} = \begin{bmatrix} \cos(x_b, x_a) & \cos(x_b, y_a) & \cos(x_b, z_a) \\ \cos(y_b, x_a) & \cos(y_b, y_a) & \cos(y_b, z_a) \\ \cos(z_b, x_a) & \cos(z_b, y_a) & \cos(z_b, z_a) \end{bmatrix} \quad (1)$$

где $[x_a, y_a, z_a]$ – исходная система координат;

$[x_b, y_b, z_b]$ – повернутая система координат.

Стоит отметить что вращение происходит в правую сторону, вращение в обратную сторону отрицательно. Так как в блоке Revolute вращение происходит вокруг оси z , то матрица (1) примет вид, в которой описан переход из одной системы координат O_a в систему координат O_b :

$$M_{ba}^z = \begin{bmatrix} \cos(\varphi_{ba}) & -\sin(\varphi_{ba}) & 0 \\ \sin(\varphi_{ba}) & \cos(\varphi_{ba}) & 0 \\ 0 & 0 & 1 \end{bmatrix}. \quad (2)$$

На рисунке 12 показан переход из одной системы координат O_a в систему координат O_b .

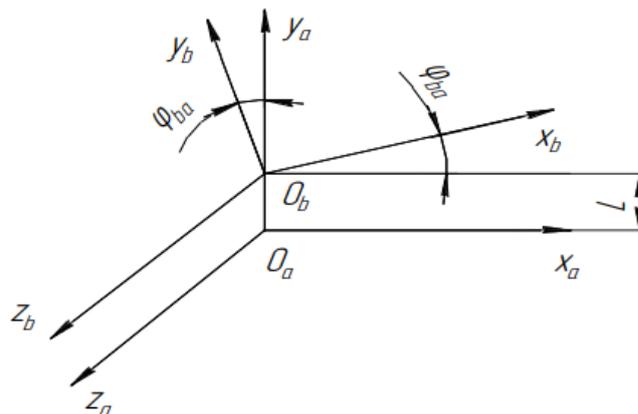


Рисунок 12 – Переход проекций вектора

Так как разрабатываемая модель руки имеет 6 степеней свободы его положение определяется шестью независимыми параметрами. Параметрами являются три координаты центра тяжести X , Y , и Z и три угла Эйлера (φ, ψ, θ) . В построенной модели Simulink блок Revolute имеет настройки, показанные на рисунке 13.

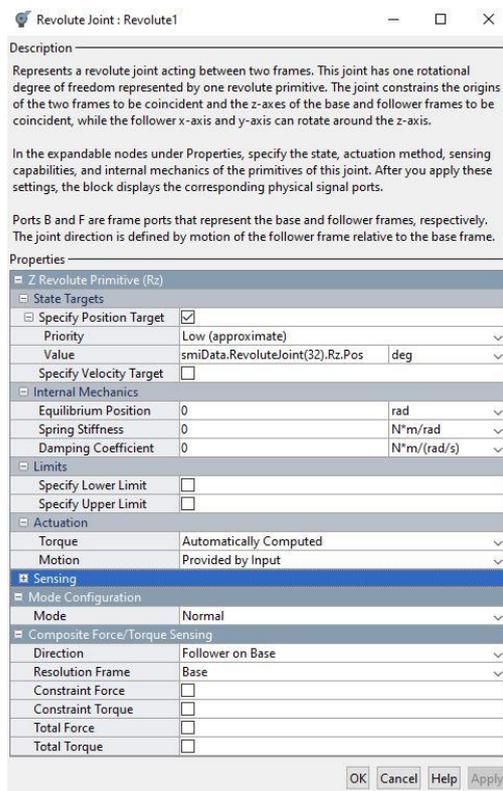


Рисунок 13 – Настройки блока Revolute

Блок Solid, а точнее File Solid характеризует твердотельный элемент со свойствами, выведенными из внешнего файла. Данный блок моделирует твердый элемент с его физическими свойствами, среди них геометрия, инерция и цвет, полученные из внешнего файла [20]. Файл должен иметь модель элемента руки и данные о геометрии тела. Некоторые форматы могут быть заданы вручную в случае необходимости. Среди поддерживаемых форматов CATIA, NX, SolidWorks (все распространенные приложения CAD). На рисунке 14 показаны настройки блока File Solid, полученные от SolidWorks. В левой части указаны полученные параметры из Solidworks, в правой части окно визуализации твердого тела, в данном случае это база руки с установленным сервоприводом для вращения всей руки вокруг основания. В данном окне доступны различные функции: виды на стороны детали, перемещение, вращение и приближение.

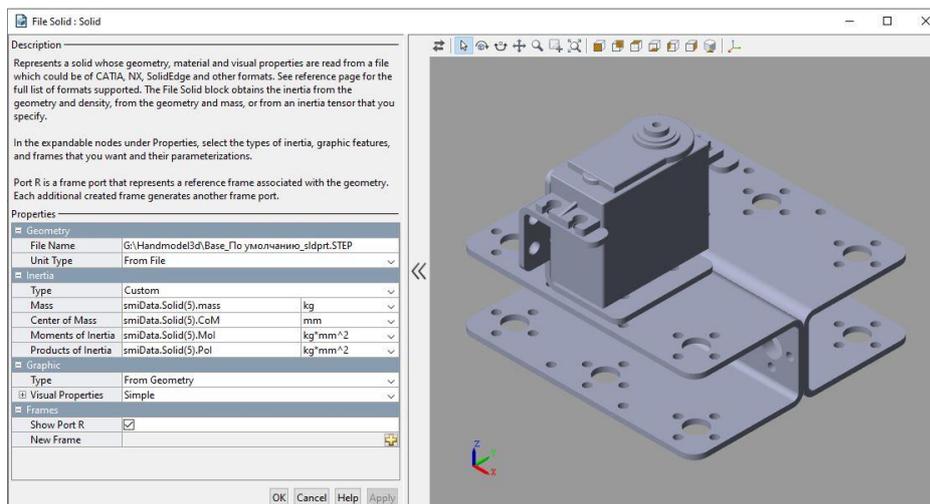


Рисунок 14 – Настройки блока File Solid

В Simulink за визуализацию модели отвечает Mechanics Explorer. Он является инструментом Simscape Multibody на основе OpenGL Silicon Graphics API. Инструмент включает панель визуализации, чтобы просмотреть модель, область просмотра в виде дерева, чтобы исследовать иерархию модели и панель свойств, чтобы исследовать отдельные параметры компонента.

Преимущества применения SolidWorks с библиотекой Simscape Multibody:

- меньше итераций при механическом проектировании, поскольку требования являются точными и последовательными;
- меньше механических прототипов, потому что ошибки выявляются раньше;
- снижение стоимости системы, поскольку компоненты не имеют слишком большой размер;
- сокращение времени простоя системы благодаря тестированию интегрированной системы с использованием виртуального ввода в эксплуатацию на цифровом двойнике;
- стандартный пользовательский графический интерфейс Windows.

Недостатки при работе с Simscape Multibody:

- в программный пакет Matlab библиотека не встроена, требует загрузки с сайта разработчика и установки отдельно;

- требуется дополнительная настройка в САПР SolidWorks;
- после экспортирования модели с помощью Simscape Multibody, блочная диаграмма требует серьезной доработки;
- в случае изменения сборки необходимо заново экспортировать и пере-страивать модель в Matlab.

2.3 Обоснование выбора программно-технического обеспечения

Разрабатываемое ПО предназначено для реализации имитационной модели для моделирования и управления моделью руки. Программа предназначена для задания управляющих воздействий на трехмерную модель руки и наблюдения изменения её положения. Применять ПО следует для анализа и исследования результатов моделирования для дальнейших работ с реальным манипуляционным устройством. Для корректной работы с моделью необходимо на ПК иметь установленный пакет Matlab R2019b с средой Simulink с установленной библиотекой Simscape, а также в отдельной папке должны быть расположены файлы: XML-файл, файлы геометрии тела, slx-файл.

Для создания проектных и инженерных моделей, системные требования представляют собой основные характеристики ПК для запуска программы. Так как отстраиваемая трехмерная модель создается в программе САПР SolidWorks, следует подбирать необходимое для стабильной работы техническое обеспечение рабочего места разработчика ПО руки, в нашем случае ПК. Версия программы выбирается по годам и предложенным обновлениям. Для построения модели руки была выбрана версия 2020 SP4.0. Данная версия выбрана за счет улучшенной производительности, в сравнении с прошлым годом. SolidWorks работает только на ОС Windows, поэтому на этой операционной системе и будет создаваться программное обеспечение моделирования движения руки.

К рекомендуемым требованиям, предъявляемые к аппаратной системе ПК относят:

- центральный процессор: четырехъядерный процессор с тактовой частотой 3.3 ГГц или выше;
- оперативная память: 16 Гбайт или больше;

- жесткий диск: от 5 Гбайт до 50 Гбайт, 19.4 Гбайт для установки всех продуктов SSD-накопитель рекомендуется для оптимальной производительности;
- графический процессор: NVIDIA Quadro (поддерживающие OpenGL 4.5), AMD FirePro, AMD Radeon Pro;
- операционная система: 64-битная Windows 7 SP1, Windows 8.1, Windows 10;
- сеть: широкополосное подключение к Интернету.

Проанализировав данные характеристики для создания модели руки используется ПК со следующими характеристиками:

- центральный процессор: шестиядерный Intel Core i5-8500 CPU 3 ГГц с включенным режимом Turbo Boost для разгона частоты до 3.9 ГГц;
- оперативная память: 16 Гбайт;
- жесткий диск: SSD-накопитель на 256 Гбайт и жесткий диск для хранения сборок и деталей на 1Тбайт;
- графический процессор: NVIDIA GeForce GTX 1070, поддерживает OpenGL 4.5;
- операционная система: 64-битная Windows 10;
- сеть: широкополосное подключение к Интернету.

В качестве программного обеспечения для создания модели манипулятора была выбрана библиотека Simscape Multibody в Matlab. Для программы Matlab версии R2019b для среды Windows необходимыми требованиями являются:

- операционная система: Windows 10;
- центральный процессор: любой процессор Intel или AMD x86–64 с четырьмя логическими ядрами и поддержкой набора команд AVX2;
- оперативная память: 8 Гбайт;
- жесткий диск: 31.5 Гбайт для установки, рекомендуется использовать SSD-накопитель;
- графический процессор: специальная видеокарта не требуется, но рекомендуется использовать видеокарту с аппаратным ускорением, поддерживающую OpenGL 3.3 с 1 Гбайт памяти GPU.

ПК полностью подходит под рекомендуемые системные требования для версии R2019b. Данная версия выбрана из-за повышенной производительности и увеличения возможностей для моделирования. Также начиная только с релизов 2018 года в Simulink реализован «Simulation Pacing». Данная функция необходима для замедления моделирования, чтобы можно было наблюдать за поведением системы. Моделирование будет выполняться с приближением к указанному темпу в окне настройки «Simulation Pacing», показанного на рисунке 15.

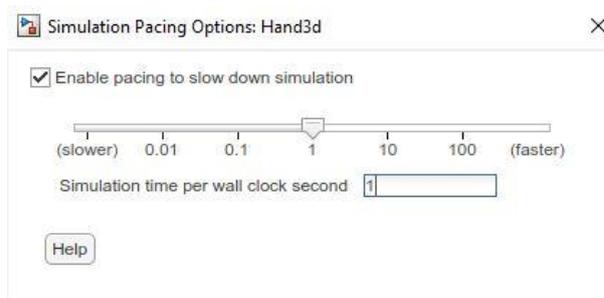


Рисунок 15 – Настройка замедления моделирования

Использование Simulink с библиотекой Simscape полезен и удобен тем, что дифференциальные уравнения записаны в виде структурной модели, то есть механическая система представляется связанной блочной диаграммой. Блоки пакета являются моделями механических устройств, положение которых в пространстве и относительно друг друга может меняться в соответствии с законами механики [21]. Моделирование механических систем и устройств осуществляется на основе законов кинематики, физики и механики.

Основная цель применения данной библиотеки для моделирования механических объектов – это показать, как движутся механические части системы и различных машин относительно друг друга в заданной системе координат, как на плоскости, так и в пространстве. При этом учитываются связи между отдельными объектами и различные их движения в соответствии с теми или иными ограничениями.

3 КОМПЬЮТЕРНАЯ РЕАЛИЗАЦИЯ АЛГОРИТМА ДЛЯ КОМПЬЮТЕРНОГО МОДЕЛИРОВАНИЯ ДВИЖЕНИЯ РУКИ

3.1 Основные этапы разработки программного продукта

3.1.1 Описание математической модели объекта исследования

Для создания модели движения руки в разных положениях необходимо исследовать особенности кинематики и динамики манипулятора и сформировать его математическую модель для разных положений руки. Математическая модель манипуляционного устройства – это приближенное описание на языке математики (отображение на математическом языке) основных закономерностей и наиболее важных свойств, присущих исследуемому оригиналу (объекту). С помощью созданной математической модели разрабатывается математическое моделирование – это метод, базирующийся на разработке и использовании математических моделей для исследования свойств, характеристик или поведения оригинала [22]. Для определения ориентации в пространстве манипулятора нужно решить две основные задачи: прямую и обратную задачи кинематики.

Решение задачи кинематики необходимо для того, чтобы аналитически описать пространственное расположение манипулятора в зависимости от времени, установить связь между значениями присоединенных координат манипулятора и занимаемым положением, а также чтобы определить ориентацию его конечного эффектора в декартовом пространстве [23]. Положение и ориентация твердого тела (или связанной с ним системы координат) в пространстве определяются шестью координатами: тремя линейными и тремя угловыми [24].

Кинематическая структура манипулятора может быть описана с помощью метода Денавита-Хартенберга. В данном методе кинематическая структура описывается в матричной форме. Данный метод применим в данном квалификационном исследовании, так как является наиболее удобным для выполнения пространственных преобразований, так как позволяет синтезировать кинематическую модель произвольного манипулятора путем задания взаимного положения

звеньев [25]. Согласно методу Денавита-Хартенберга решение состоит из четырех шагов: привязка систем координат к звеньям, определение параметров Денавита-Хартенберга, построение матриц однородного преобразования, расчет углов Эйлера по итоговой матрице вращения.

При решении рассматриваются две системы координат: исходная, связанная с землей и итоговая, связанная со конечным эффектором. Эти два набора координат связаны некоторым преобразованием:

$$k^0 = T_n^0 k^n, \quad (3)$$

где T_n^0 – преобразование, несущее информацию о линейном смещении и пространственной ориентации одной системы относительно другой.

Общая матрица однородного преобразования выражается следующим образом:

$$T_n^0 = \prod_{i=1}^n A_i^{i-1} = \begin{bmatrix} n_x & s_x & a_x & p_x \\ n_y & s_y & a_y & p_y \\ n_z & s_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} n_n^0 & s_n^0 & a_n^0 & p_n^0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (4)$$

где A_i – матрицы преобразования i -сочленения;

n – количество сочленений в манипуляторе;

n_n^0, s_n^0, a_n^0 – векторы направления осей x_n, y_n и z_n , относительно исходной системы координат;

p_n^0 – вектор линейного смещения начала координат системы $O_n x_n y_n z_n$ относительно $O_0 x_0 y_0 z_0$.

Обратная задача кинематики заключается в расчете обобщенных координат при заданных линейных и угловых координатах рабочего органа манипуля-

ционного устройства [26]. Эта задача сложнее, чем прямая, так как может привести к неопределенности решения, потому что одному и тому же положению точки захвата могут соответствовать разные положения всей руки.

Для описания математической модели необходимо произвести кинематический анализ манипуляционного устройства. Кинематика роботов относится к аналитическому изучению движения робота-манипулятора, без учета сил или моментов, вызывающие его движение. Формулировка подходящих кинематических моделей для механизма робота очень важна для анализа поведения промышленных манипуляторов. Манипуляционное устройство состоит из 5 звеньев: нулевое и первое звено – основание, плечо, предплечье, запястье и кисть. Каждое звено связано вращательными соединениями. На рисунке 16 показана кинематическая схема для нулевого положения манипулятора при котором все обобщенные координаты равны нулю. Также на этой схеме указаны размеры каждого звена, приведенные в миллиметрах.

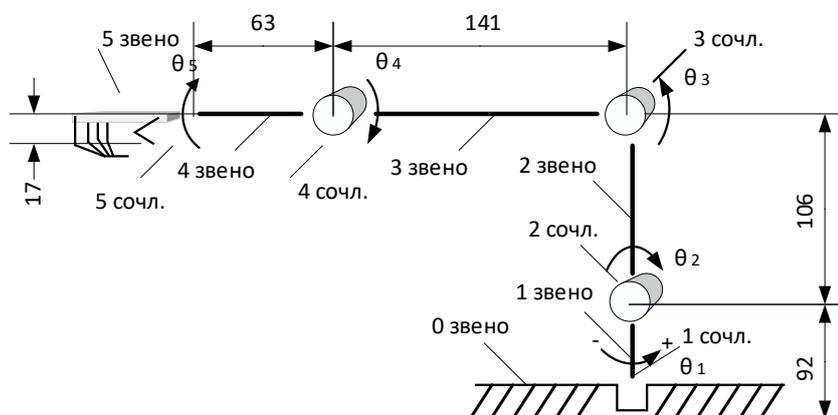


Рисунок 16 – Кинематическая схема манипуляционного устройства

Отметим, что пятое звено не участвует в определении положения конечного эффектора, так как служит только для его вращения. Для решения прямой задачи кинематики используется метод Денавита-Хартенберга описанный выше. Первым этапом этого метода служит привязка систем координат к звеньям. Для привязки системы координат к звеньям необходимо учитывать несколько правил:

- ось z_i выбирается так, чтобы она совпала с осью вращения последующего сочленения;
- ось x_i выбирается так, чтобы она пересекала и была перпендикулярна оси z_{i-1} ;
- ось y_i выбирается так, чтобы она дополняла систему координат до правой декартовой системы координат (по правилу правой руки).

Выбранные оси в кинематической структуре стоит учитывать при создании трехмерной модели в SolidWorks, так как они будут важны при дальнейшем исследовании модели. Системы координат каждого звена представлены на рисунке 17.

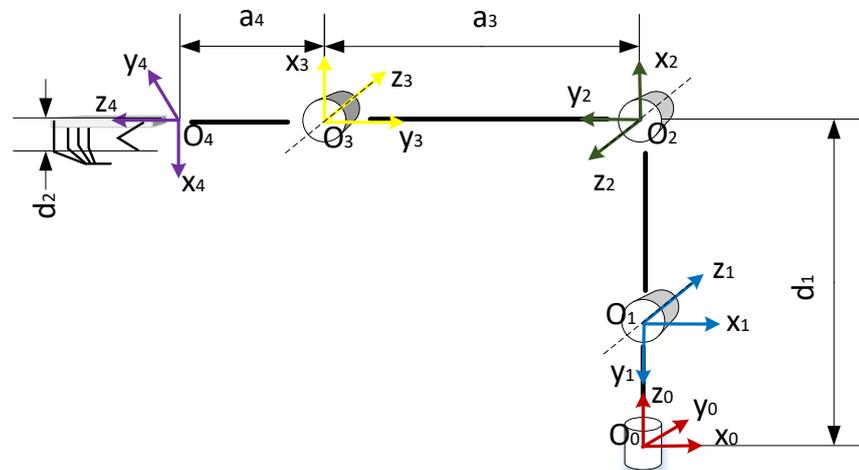


Рисунок 17 – Выбранные системы координат звеньев

Для определения четырех геометрических параметров Денавита-Хартенберга нужно воспользоваться следующими правилами:

- угол θ_i – это присоединенный угол. Значение угла поворота оси x_{i-1} вокруг оси z_{i-1} , чтобы ось x_{i-1} стала сонаправлена с осью x_i ;
- угол α_i – это линейное смещение, которое определяет расстояние, на которое необходимо повернуть ось z_{i-1} вокруг x_i , чтобы она стала сонаправлена с z_i ;
- a_i – расстояние между z_{i-1} и z_i вдоль оси x_i ;
- d_i – расстояние между x_{i-1} и x_i вдоль оси z_{i-1} .

Изображенное расположение звеньев (и связанных с ними систем координат) считается за нулевую конфигурацию. Это означает, что все обобщенные координаты для конкретно такого положения манипулятора являются нулевыми. Стоит отметить то, что знак углов определяется в соответствии с правилом правой руки. Параметры для ДХ-представления данной кинематической схемы устройства представлены в таблице 5.

Таблица 5 – Параметры для ДХ-представления

Звено i	θ_i	α_i	$a_i, \text{ м}$	$d_i, \text{ м}$
1	θ_1	-90	0	0,198
2	$-\theta_2$	0	0	0,017
3	θ_3	0	0,141	0
4	$-\theta_4$	-90	0,063	0

Далее в описанном методе Денавита-Хартенберга следует пункт построения матриц однородного преобразования. Они строятся по полученным параметрам используя выражение (4). В общем виде матрица преобразования будет выглядеть следующим образом:

$$\begin{aligned}
 T_i^{i-1} &= \begin{bmatrix} \cos(\theta_i) & -\sin(\theta_i) & 0 & 0 \\ \sin(\theta_i) & \cos(\theta_i) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \\
 &\times \begin{bmatrix} 1 & 0 & 0 & a_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\alpha_i) & -\sin(\alpha_i) & 0 \\ 0 & \sin(\alpha_i) & \cos(\alpha_i) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \\
 &= \begin{bmatrix} \cos(\theta_i) & -\cos(\alpha_i)\sin(\theta_i) & \sin(\theta_i)\sin(\alpha_i) & a_i \cos(\theta_i) \\ \sin(\theta_i) & \cos(\alpha_i)\cos(\theta_i) & -\cos(\theta_i)\sin(\alpha_i) & a_i \sin(\theta_i) \\ 0 & \sin(\alpha_i) & \cos(\alpha_i) & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}.
 \end{aligned} \tag{5}$$

Отдельные матрицы преобразования для каждого звена примут вид:

$$T_0^1(\theta_1) = \begin{bmatrix} \cos(\theta_1) & 0 & -\sin(\theta_1) & 0 \\ \sin(\theta_1) & 0 & \cos(\theta_1) & 0 \\ 0 & -1 & 0 & 0,198 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (6)$$

$$T_1^2(\theta_2) = \begin{bmatrix} \cos(\theta_2) & \sin(\theta_2) & 0 & 0 \\ -\sin(\theta_2) & \cos(\theta_2) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (7)$$

$$T_2^3(\theta_3) = \begin{bmatrix} \cos(\theta_3) & -\sin(\theta_3) & 0 & 0,141 \cdot \cos(\theta_3) \\ \sin(\theta_3) & \cos(\theta_3) & 0 & 0,141 \cdot \sin(\theta_3) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (8)$$

$$T_3^4(\theta_4) = \begin{bmatrix} \cos(\theta_4) & 0 & \sin(\theta_4) & 0,066 \cdot \cos(\theta_4) \\ -\sin(\theta_4) & 0 & \cos(\theta_4) & -0,066 \cdot \sin(\theta_4) \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (9)$$

Матрицей однородного преобразования, которая связывает все системы координат будет являться произведение матриц из формул (6-9) [27]. В полученной матрице выражения в 4 столбце, кроме 4 строчки характеризуют связь трех систем координат x , y , z . Так как модель руки направлена против положительной оси x , а конечный эффектор имеет сдвиг в отрицательную сторону по оси y необходимо поставить отрицательный знак перед выражениями отвечающие за координаты x и y .

Результирующими выражениями для нахождения координат будут являться с учетом направления точки захвата в отрицательную сторону по X и по Y :

$$P_x = -(0.063 \cos(\theta_1) \cdot \cos(\theta_4 + \theta_2 - \theta_3) + 0,141 \cos(\theta_1) \cos(\theta_2 - \theta_3) - 0,0175 \sin(\theta_1)), \quad (10)$$

$$P_y = -(0.063 \sin(\theta_1) \cdot \cos(\theta_4 + \theta_2 - \theta_3) + 0,141 \sin(\theta_1) \cos(\theta_2 - \theta_3) + 0,0175 \cos(\theta_1)), \quad (11)$$

$$P_z = 0.063 \sin(\theta_4 + \theta_2 - \theta_3) + 0,141 \sin(\theta_2 - \theta_3) + 0,198. \quad (12)$$

При начальных значениях углов ($\theta_1 = \theta_2 = \theta_3 = \theta_4 = 0$) значения $P_x = -0,203$; $P_y = -0,0175$; $P_z = 0,198$. Теперь необходимо построить трехмерную модель на основе проведенного анализа кинематической структуры манипуляционного устройства и собрать модель в программе Matlab в среде Simulink для изучения модели руки и управления ею, а также для проверки полученных результатов и оценки их точности.

Для реализации динамики манипулятора будет использоваться уравнение движения в конфигурационном пространстве:

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) = \tau, \quad (13)$$

где $q(t), \dot{q}(t), \ddot{q}(t)$ – векторы обобщенных координат, скоростей и ускорений;

$M(q)$ – матрица инерции размерности $[n \times n]$;

$C(q, \dot{q})$ – вектор кориолисовых и центробежных сил размерности $[n \times 1]$;

$g(q)$ – вектор гравитации размерности $[n \times 1]$;

$\tau(t)$ – вектор обобщенных сил и моментов размерности $[n \times 1]$.

3.1.2 Построение модели руки на основе реального манипуляционного устройства

Современные программные решения позволяют моделировать трехмерные объекты практически любой сложности, используя базовый инструментарий САД-модуля [28]. Общие требования к выполнению работ в форме электронной

модели изделия выполняются по ГОСТ 2.051 [29]. В ГОСТ 2.052 определяются три вида трехмерных электронных моделей: твердотельная, поверхностная и каркасная модели [30]. В квалификационном исследовании создается твердотельная модель, которая представляет собой трехмерное изображение объекта и имитирует все его физические и механические свойства, а также позволяет определить его числовые характеристики, такие как масса, инерция и т.п.

На рисунке 18 изображена структурная схема, на ней показаны схематично маршруты действий для того чтобы построить твердотельную модель детали.



Рисунок 18 – Структурная схема построения твердотельной модели

Построение будет вестись в системе автоматизированного проектирования SolidWorks 2020, так как эта среда была выбрана на этапе анализа сред для моделирования. Эта среда позволяет выполнять: гибридное параметрическое моделирование, в него входит твердотельное моделирование, а также моделирование поверхностей, каркасное моделирование и их комбинации; со спецификой выполнения, например, детали из различных материалов; проектирование сложных сборок.

Построение каждой детали в SolidWorks начинается с построения примитива. Например, с точки, отрезка, окружности или прямоугольника. Построение примитива происходит в режиме «Эскиз», как только примитив отстроен далее

происходит переход в режим «Модель» [31]. В этом режиме по команде «Вытянутая бобышка/основание» и при нужных настройках глубины создается объемная деталь.

Для облегчения общей сборки модели необходимо разделить манипуляционное устройство на различные элементы, а затем назначить взаимосвязи между ними так, чтобы осуществлялось вращательное движение между элементами. Первым элементом является база или основание вокруг которой вращается рука. На рисунке 19 показано реальное основание у манипуляционного устройства по размерам которой будет отстраиваться модель, продемонстрированная на рисунке 20.



Рисунок 19 – Основание реального устройства



Рисунок 20 – Твёрдотельная модель основания руки

На рисунке 21 показан внешний вид сервопривода (сервомотора) TD-8135MG, он используется для управления звеньями манипулятора. Физически сервопривод управляется с помощью импульсов переменной длительности. С помощью него выполняются вращательные движения между элементами. В манипуляционном устройстве установлено 5 сервоприводов 8135MG.



Рисунок 21 – Сервопривод TD-8135MG

На выходной вал сервопривода надевается рычаг для передачи вращательного движения на рабочий орган. Внешний вид модели рычага показан на рисунке 22.

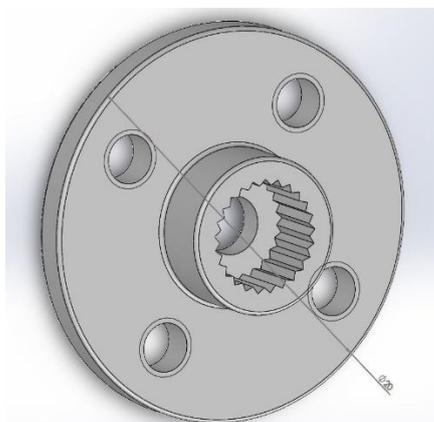


Рисунок 22 – Модель рычага

Далее сборка из формата SLDASM, собранная из деталей основания сохраняется как деталь в формате SLDPRT, это дает возможность облегчить модель и убрать все взаимосвязи не участвующие в движении руки. Следующая

сборка отвечает за движение плеча. Плечо реального устройства показано на рисунке 23. На рисунке 24 представлена его твердотельная модель.



Рисунок 23 – Плечо руки

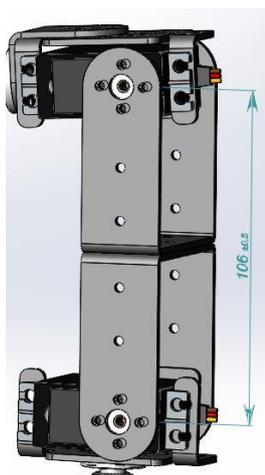


Рисунок 24 – Твердотельная модель плеча руки

Следующий элемент руки – предплечье. К нему присоединены отстроенное до этого плечо и элемент, отвечающий за сгиб запястья. На рисунке 25 показано предплечье руки на реальном манипуляционном устройстве. Сборка выполняется по тому же принципу, что описан выше. Вращающие движения локтя руки происходят за счет рычага на сервоприводе находящийся между деталью плеча и деталью предплечья. На рисунке 26 показана отстроенная твердотельная модель предплечья руки.



Рисунок 25 – Предплечье руки

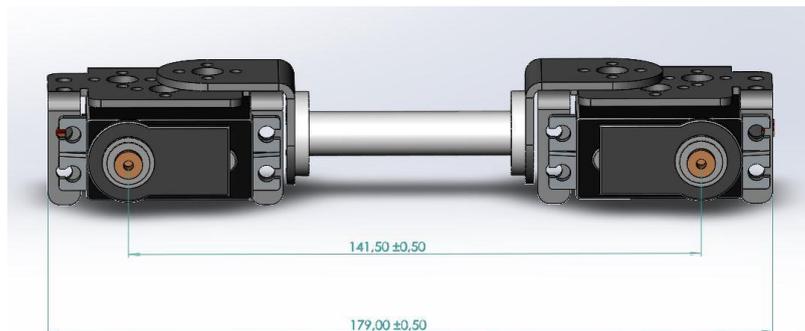


Рисунок 26 – Твёрдотельная модель предплечья руки

Далее к предплечью крепиться элемент – запястье, отвечающий за сгиб запястья руки. На рисунке 27 показано запястье руки. К данному элементу далее будет крепиться кисть. На рисунке 28 изображен внешний вид модели запястья.



Рисунок 27 – Запястье руки

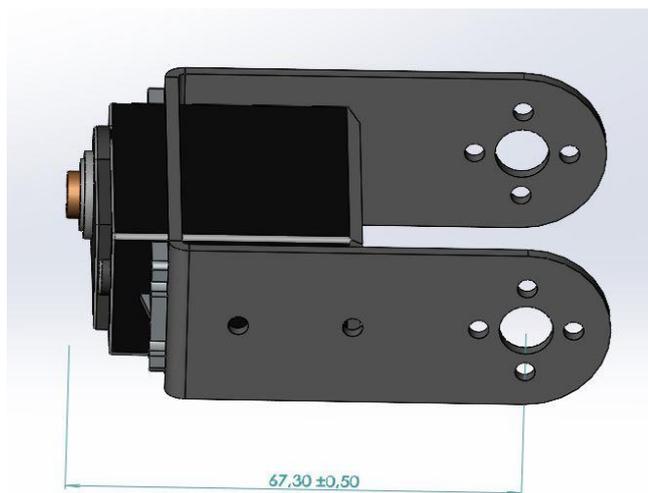


Рисунок 28 – Твёрдая модель запястья руки

Добавляя к основной сборке отстроенную модель запястья, а также взаимосвязи к рычагам на сервоприводах для совершения вращательных движений, основная часть манипуляционного устройства будет построена. Для полноценного манипуляционного устройства необходим рабочий орган или захватное устройство. В реальном устройстве установлена кисть, напоминающая человеческую, внешний вид которой показан на рисунке 29.



Рисунок 29 – Кисть руки

Управляется каждый палец с помощью сервопривода SG90, внешний вид которого показан на рисунке 30.



Рисунок 30 – Сервопривод SG90

Сборка кисти в SolidWorks производится по тому же принципу, что и все составные части руки. Сначала начинаем с отстройки отдельных деталей, затем объединяем детали в сборки, причем объединяем те части, где движения необходимые для подвижности руки не происходят. По этому принципу в сборку объединяем ладонь руки и сервоприводы. Пальцы кисти отличаются от образца, так как была создана форма более похожая на человеческие пальцы, также заменены амортизирующие пружины для передачи движения рычага на палец на более облегченный вариант. В общей сборке кисти 11 деталей: ладонь, 5 рычагов для сервоприводов, 5 передающих момент деталей и 5 сборок пальцев. В каждую сборку пальцев входят 3 фаланги пальца, жестко закрепленные для корректной работы модели. Внешний вид твердотельной модели кисти руки показан на рисунке 31.

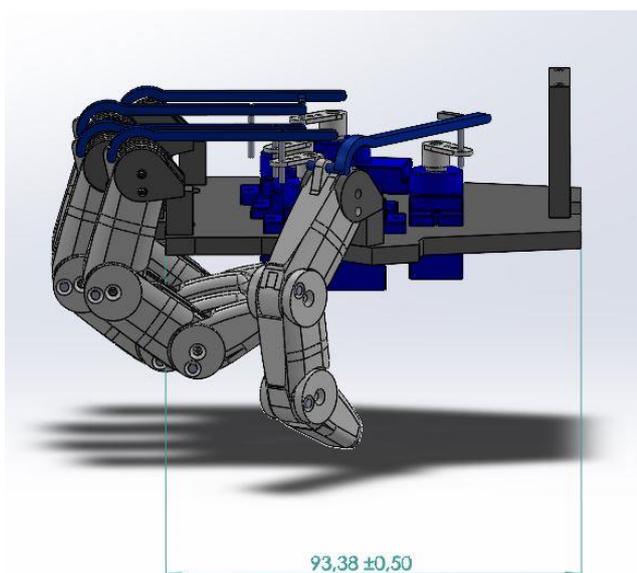


Рисунок 31 – Твердотельная модель кисти

Общая модель руки, полученная при объединении всех деталей и сборок, построенных выше в САПР SolidWorks показана на рисунке 32.

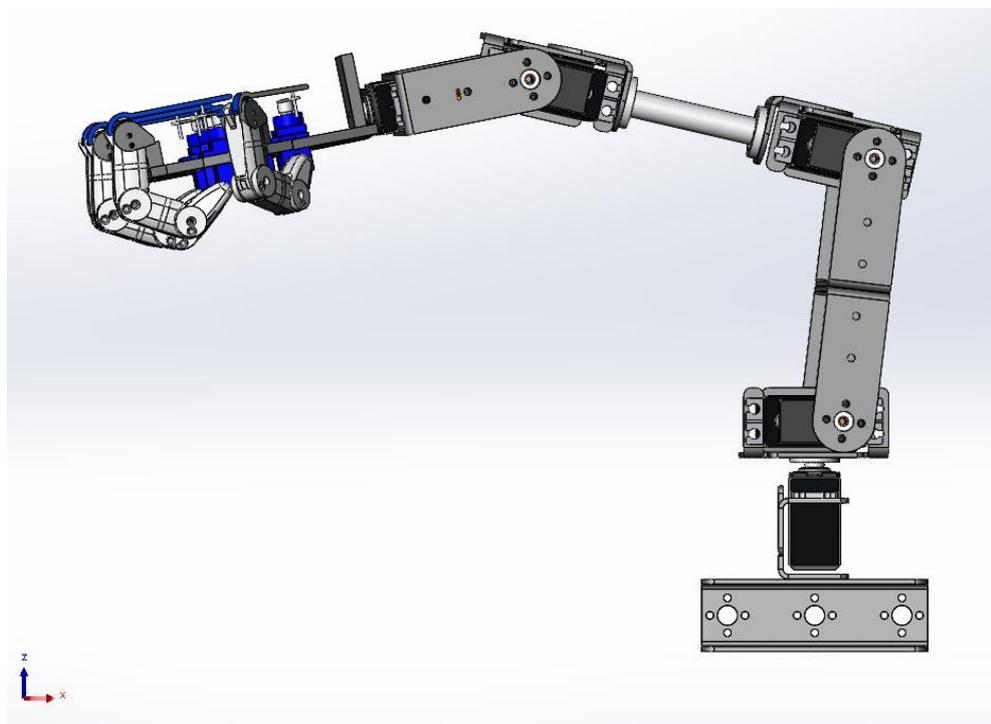


Рисунок 32 – Общий вид на модель руки в SolidWorks

Стоит еще раз отметить, что построение правой руки велось против оси X. Этот момент необходим для дальнейшего построения системы управления на основе кинематического анализа и построенной трехмерной модели. Также стоит отметить, что применение технологий САПР помогает разработчикам при моделировании сложных объектов, намного облегчает построение, изучение, анализ объектов, также есть возможность их улучшения. Позволяет легко проверять траектории движения отстроенных моделей любого типа в том числе и манипуляторов.

3.1.3 Программная реализация алгоритма управления звеньями руки

Для управления звеньями модели руки с помощью пакета прикладных программ Matlab необходимо создать модель переносом созданных файлов системой SolidWorks с помощью утилиты SimScape MultiBody Link в графическую

среду Simulink. При переносе всех файлов и хранящихся в них данных необходимо заблаговременно создать папку, в которой будут храниться все файлы, включая создаваемые файлы Matlab. Иначе файл с данными сохранится в корневой папке Matlab или в открытой папке в рабочей области. Это приведет к потере данных и блоки с деталями не будут определены. Правильная настройка отображена в окне Model Explorer на рисунке 33. В рабочей области Simulink-модели должен храниться файл smiData.

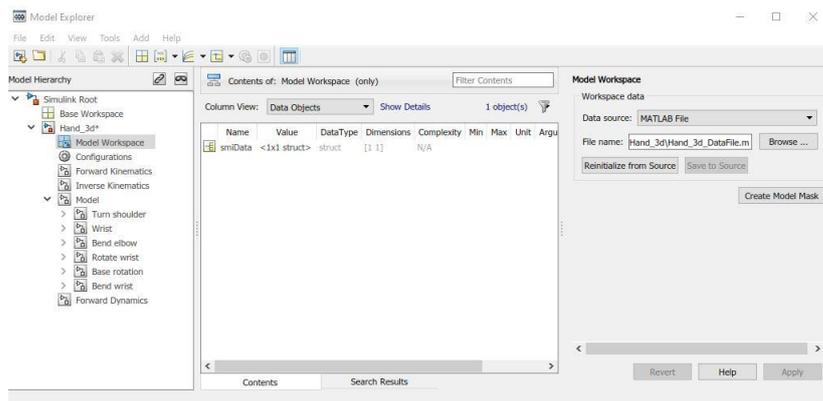


Рисунок 33 – Настройка в окне Model Explorer

После редактирования полученной схемы с блоками Simscape Multibody получим структурную схему из 6 блоков: вращение вокруг базы, поворот плеча, сгиб локтя, сгиб запястья, вращение запястья, кисть. Внешний вид общей схемы показан на рисунке 34. Далее схему поместим в подсистему с названием «Model Hand». Данная схема является физической моделью Simscape. На нее будут подаваться управляющие сигналы, которые будут высчитываться в датчиках «Revolute». Затем полученные результаты будут сняты при помощи блоков для вывода сигнала.

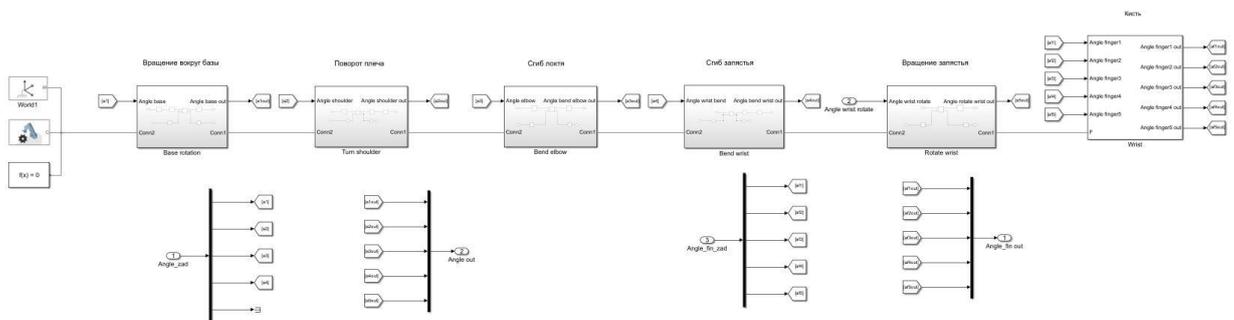


Рисунок 34 – Физическая модель руки

В каждой подсистеме, отвечающей за движение звеньев руки находится структурная схема из блоков библиотеки Simscape и Simulink. Описание каждого блока Simscape было представлено выше. На рисунке 35 показана схема в подсистеме «Вращение вокруг базы (Base rotation)». В данной схеме расположены четыре блока: «Transform», «Base», «Revolute», «Servo_Mount_1». Блок Rigid Transform необходим для преобразования одной системы координат в другую, иными словами, он вращает и переводит систему координат порта последователя (F) относительно основной системы координат порта (B). Блок «Base» необходим для определения положения и конфигураций тела базы руки. На рисунке 36 показана внутренняя схема блока «Base». Структурная схема блока «Servo_Mount_1» (рычаг на сервоприводе) отличается от «Base», тем что в нем находятся два блока «Transform», отвечающие за расположение систем координат относительно базы и следующей подсистемы «Turn shoulder».

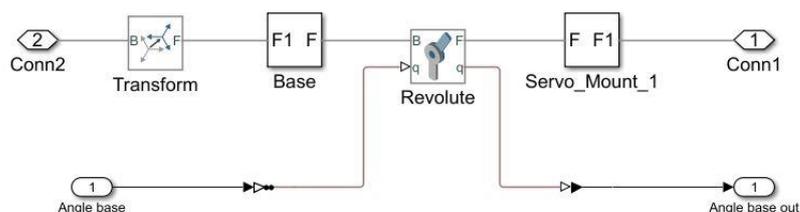


Рисунок 35 – Структурная схема подсистемы «Base rotation»

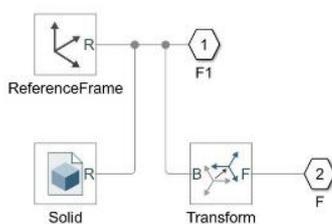


Рисунок 36 – Структурная схема блока «Base»

На рисунке 37 показана схема в подсистеме «Поворот плеча (Turn shoulder)». В данной схеме расположены четыре блока: «Rotatehand», «Revolute1», «Servo_Mount_2» и «Shoulder». Структурные схемы внутри блоков однотипны описанным выше схемам.

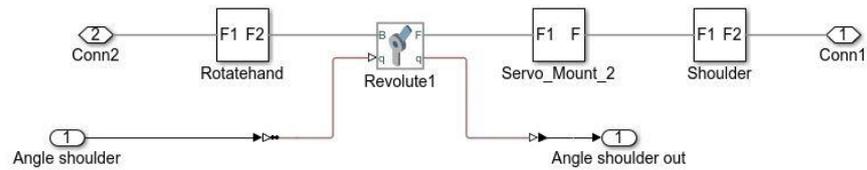


Рисунок 37 – Структурная схема подсистемы «Turn shoulder»

На рисунке 38 показана схема в подсистеме «Сгиб локтя (Bend elbow)». В данной схеме расположены три блока: «Servo_Mount_3», «Revolute2», и «Forearm».

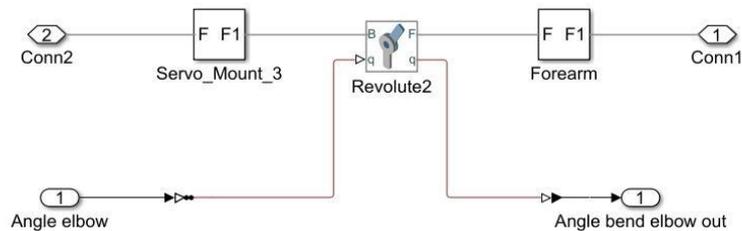


Рисунок 38 – Структурная схема подсистемы «Bend elbow»

На рисунке 39 показана схема в подсистеме «Сгиб запястья (Bend wrist)». В данной схеме расположены три блока: «Revolute3», «Servo_Mount_4» и «Wristhand».

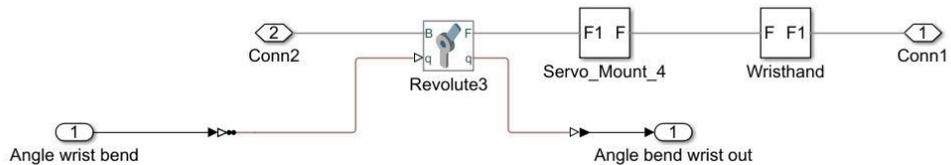


Рисунок 39 – Структурная схема подсистемы «Bend wrist»

На рисунке 40 показана схема в подсистеме «Вращение запястья (Rotate wrist)». В данной схеме расположены два блока: «Revolute4» и «Servo_Mount_5».

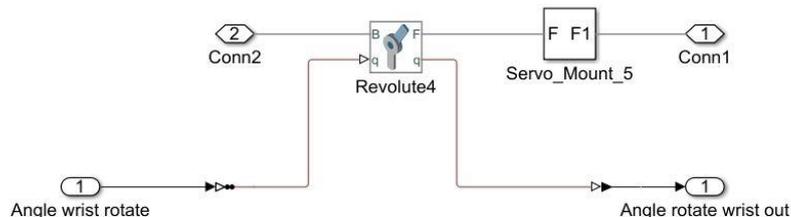


Рисунок 40 – Структурная схема подсистемы «Rotate wrist»

Для того чтобы приводить в движения звенья необходимо на блоки Revolute в каждой подсистеме подавать управляющие воздействия, в нашем случае – угол в радианах, на который будет сдвигаться каждая часть руки [32]. За конвертацию отвечает блок Simulink-PS Converter, в нем задается физическая единица измерения «Input signal unit» равной «rad» (измерение в радианах) и параметр «Input filtering order», который задает количество производных времени входного сигнала, обеспеченного путем фильтрации равный «Second-order filtering» (обеспечивает вычисление второй производной) [33].

Выходной сигнал представляет собой значение, которое занимает рука после подачи входного сигнала. Оно конвертируется в Simulink сигнал посредством блока PS-Simulink Converter, в нем задается физическая единица измерения «Input signal unit» равной 1.

Для задания углов на входы блоков «Revolute» в подсистеме «Model» в главном окне Simulink модели выделим область для ввода значений пользователем, для удобства выделено голубым цветом и имеет надпись: «Изменение углов поворота звеньев руки». Задание углов происходит во время моделирования в блоках Slider Gain, выполняющий скалярное усиление, которое можно изменить в процессе моделирования [34]. Каждый блок настроен на минимальный и максимальный угол отклонения от начального положения. Так как на вход блока «Revolute» необходимо подавать размерность углов в радианах, то после Slider Gain поместим блок, преобразующий градусы в радианы. На рисунке 41 показан внешний вид области для ввода значений углов.

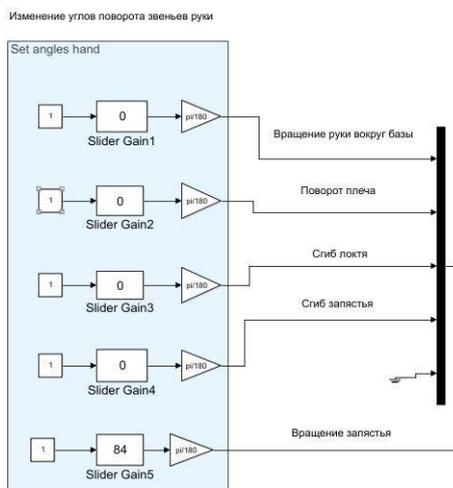


Рисунок 41 – Область для ввода значений углов поворота звеньев руки

Для управления рукой в её рабочем пространстве необходим алгоритм для того чтобы указывать звеньям робота поворачиваться на определенный угол, если требуется достичь определенной точки в рабочем пространстве. Описанный выше способ ввода значений углов вращения можно считать базовым решением. Он применим если пользователь знает на какие углы повернуть звенья для того чтобы переместить конечный эффектор в нужное положение. Чтобы определить положение точки захвата (конечного эффектора) по заданным углам поворота нужно воспользоваться описанной выше прямой кинематикой модели. В случае если углы поворота не известны, а требуется переместить точку захвата в необходимое положение требуется воспользоваться обратной кинематикой, реализация которой будет описана в следующем подразделе.

В квалификационном исследовании будут использоваться два метода решения прямой задачи кинематики. Первый метод основан на решении матрицы преобразования Денавита-Хартенберга, где по значению задаваемых углов находятся три координаты расположения точки захвата. Вычисление координат по этому методу будет вестись в m-файле для интерфейса. Во втором методе воспользуемся блоком из библиотеки Robotics System Toolbox, который предоставляет инструменты и алгоритмы для моделирования и тестирования для манипуляторов [35]. Для блоков из данной библиотеки требуется представление дерева

твёрдого тела. Для этого необходимо в m-файле задать команду «[hand_3d_model_Arm, Hand_3d_Arm] = importrobot('Hand3d');» где «Hand3d» имя Simulink-модели [36]. Перед вызовом данной команды прокомментируем блок, отвечающий за визуализацию кисти руки, так как он не влияет на расположение конечного эффектора. Промоделируем модель для создания в рабочей области структуру из данных расположения модели руки. Затем запустим подготовленный m-файл, который создаст объекты дерева твёрдого тела (rigidBodyTree) в рабочей области [37]. В нем описаны 5 структур для каждого тела модели руки. На рисунке 42 показан внешний вид структуры основания.

Property	Value
Name	'Body1'
Joint	1x1 rigidBodyJoint
Mass	0.0187
CenterOfMass	[-0.0127,-0.0132,0.0528]
Inertia	[7.8703e-05,7.1103e-05,3.1855e-05,1.4335e-05,1.6804e-05,-...
Parent	1x1 rigidBody
Children	1x1 cell
Visuals	0x0 cell

Рисунок 42 – Описание основания в дереве твёрдого тела

В разделе библиотеки Robotics System Toolbox выберем подраздел Manipulator Algorithms в котором находятся блоки для управления манипулятором разными методами. Для определения координат по положению манипулятора воспользуемся блоком Get Transform [38]. На вход этого блока необходимо вывести вектор $[N \times 1]$ положений суставов для количества N нефиксированных соединений в связанной модели дерева твёрдого тела, в данном манипуляторе вектор равен $[5 \times 1]$. Блок выводит матрицу преобразования $[4 \times 4]$. Однородное преобразование необходимо перевести в представление преобразования координат, за это отвечает блок Coordinate Transformation Conversion [39]. Внешний вид блочной диаграммы, которая вычисляет прямую кинематику показан на рисунке 43.

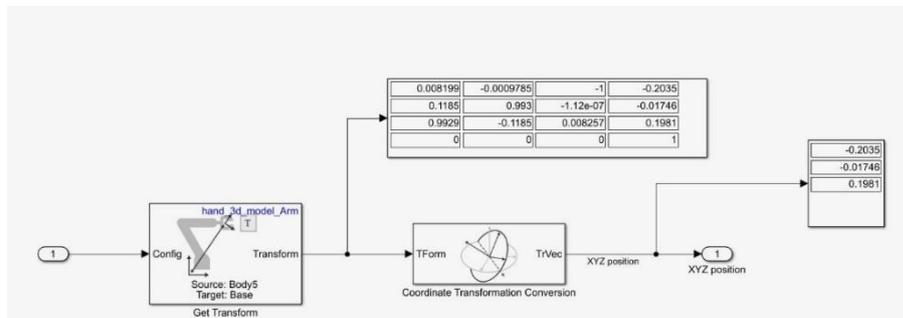


Рисунок 43 – Вычисление прямой кинематики в Simulink

Полученные координаты можно использовать для дальнейшего исследования движения модели руки.

3.1.4 Программная реализация алгоритма по заданию положения руки

Управление по положению руки будет реализовано в несколько этапов отличающихся друг от друга принципом задания входных значений, подаваемых на физическую модель SimScare. Первым этапом будет реализовано задание оператором координат на перемещение. Вторым этапом будет являться задание полиномиальной траектории, которой будет следовать манипулятор в течении заданного времени. Третьим этапом реализуется управление крутящими моментами для контроля заданного положения модели манипулятора по заданным оператором координат. С помощью этих этапов будут получены углы из физической модели, которые можно будет использовать разработчикам систем управления реальным манипулятором.

Для реализации первого этапа необходимо воспользоваться библиотекой Robotics System Toolbox и блоком Inverse Kinematics. Данный блок вычисляет положение конечного эффектора на основе заданной модели дерева твердого тела [40]. Для работы с блоком инверсной(обратной) кинематики необходимо произвести настройку внутри блока. Для достижения решения выберем алгоритм проекции градиента Бройдена Флетчера Голдфарба Шэнно (BFGS), который использует градиенты функции стоимости от прошлых итераций, чтобы сгенерировать аппроксимированную информацию о второй производной. Этот метод является алгоритмом по умолчанию и более устойчив при нахождении решений,

чем метод Levenberg-Marquardt [41]. Чтобы преобразовать вектор координатного преобразования в матрицу преобразований необходимо воспользоваться блоком Coordinate Transformation Conversion. В блоке Delay начальное состояние равно $[0; 0; 0; 0; 1.57]$. Значение 1,57 означает угол в радиан, на который нужно повернуть запястье, чтобы рычаги на кисти были направлены вверх. Вращение кисти не влияет на расположение конечного эффектора, поэтому модель построена таким образом, чтобы можно было независимо задавать угол для вращения запястья. Внешний вид подсистемы «Inverse Kinematics» показан на рисунке 44.

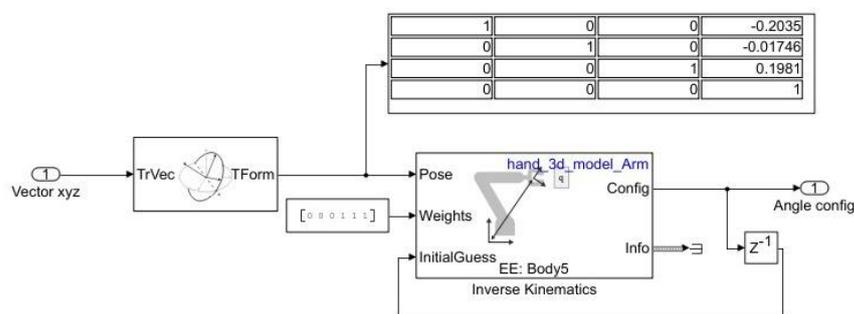


Рисунок 44 – Подсистема «Inverse Kinematics»

На блок Coordinate Transformation Conversion необходимо передать физическое местоположение в рабочей области, который в свою очередь передает матрицу однородного преобразования алгоритму обратной кинематики. Данный алгоритм возьмет точки рабочей области, по которым необходимо переместиться роботу манипулятору и сгенерирует углы суставов, которые будут переданы в симуляции. Датчик «Revolute» измерит углы на суставах и передаст данные алгоритму прямой кинематики. На рисунке 45 показана область для ввода значений координат, а также для ввода готовой траектории в Simulink-модели. Переключение между разными входными воздействиями на систему реализовано через переключатель «Switch2» на который подается сигнал 0 или 1 через блок «Gain17». Такой же переключатель установлен после блока «Inverse Kinematics», чтобы можно было задавать углы вручную. Переключение переключателей будет реализовано через интерфейс пользователя.

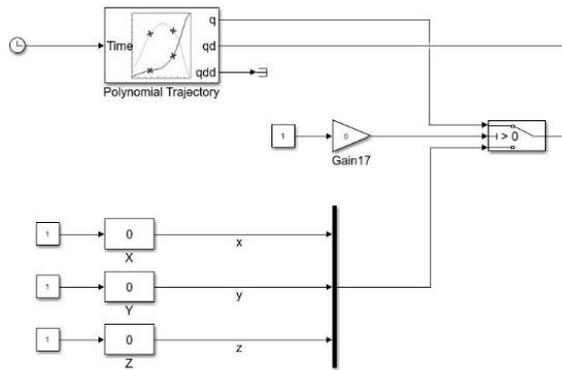


Рисунок 45 – Задание положение для модели руки

Значение координат вводится в блоки «Slider Gain». В них выставлено максимальное и минимальное отклонение для X, Y от -0,4 до 0,4. Для Z от 0 до 0,4. Заданная траектория, генерируемая алгоритмом обратной динамики может не совпадать с траекторией на выходе физической модели. Данная проблема будет решаться на третьем этапе.

Вторым этапом реализовывается следование руки по заданной траектории. Воспользуемся блоком Polynomial Trajectory из библиотеки Robotics System Toolbox в разделе Utilities. На рисунке 46 показано окно настройки блока где необходимо указать путьевые точки для трех координат, временные точки за которые необходимо пройти путь траектории. Матрица путьевых точек для x, y, z хранится в переменной pol1 и показана на рисунке 47.

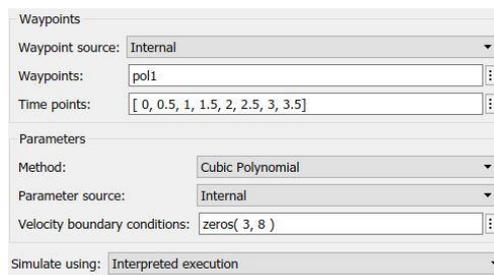


Рисунок 46 – Настройка блока Polynomial Trajectory

	1	2	3	4	5	6	7	8
1	-0.2035	-0.1018	-0.1018	-0.2035	-0.2035	-0.2035	-0.2035	-0.2035
2	-0.0175	-0.0175	0.0100	0.0100	0.0163	0.0163	0.1630	-0.0175
3	0.1981	0.1981	0.1981	0.1981	0.1981	0.0500	0.0500	0.1981

Рисунок 47 – Матрица путьевых точек

Блок Polynomial Trajectory генерирует траектории, чтобы переместиться через путевые точки в заданных моментах времени с помощью трех методов. На выходе блока выводятся положения, скорости и ускорения для достижения траектории на основе входа Time [42].

Для третьего этапа необходимо внести изменения в физическую модель, а именно настроить датчики «Revolute» для подачи крутящих моментов на рычаги сервоприводов. Для этого создадим новый файл Simulink в рабочей папке и помести туда физическую модель в подсистеме «Model Hand». Затем изменим настройки «Revolute» во всех подсистемах, кроме подсистемы вращения запястья, так как задание изменения угла поворота будет поступать напрямую от оператора. Simulink- PS Converter настраивается на измерение крутящего момента (N·m). Измененная подсистема вращения вокруг базы показана на рисунке 48. Внутренние настройки блока «Revolute» показаны на рисунке 49.

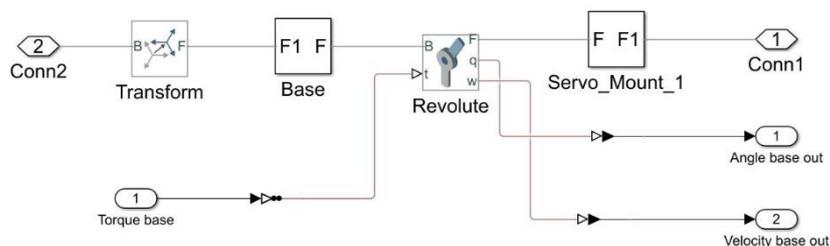


Рисунок 48 – Измененная структурная схема подсистемы «Base rotation»

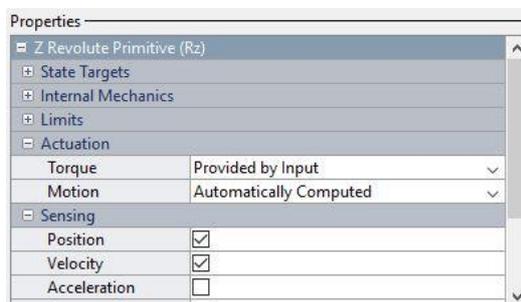


Рисунок 49 – Настройка блока «Revolute»

Для взаимодействия руки с окружением и возможности воздействовать на него с контрольным усилием необходимо отслеживать крутящий момент вместо

траектории положения шарнира (рычага). Для расчета крутящего момента необходимо воспользоваться уравнением динамики и на его основе создать вычисляемый контроллер крутящего момента, необходимый для получения заданной конфигурации и скорости, при условии, что переменные уравнения динамики руки, а именно: сила, затрачиваемая на ускорение; силы, для преодоления кориолисовых, центростремительных и центробежных сил; силы для преодоления гравитационных сил известны. В Simulink эти переменные могут быть получены с помощью блоков робототехнических манипуляторов из библиотеки Robotics System Toolbox.

Необходимо помнить, что если обойти физическую модель и подавать напрямую сигналы на блок прямой кинематики или динамики, то на выходе получится тоже самое, но когда сигнал пройдет через физическую модель, то появятся отклонения от заданных значений. Это связано с возмущающими воздействиями действующими на физическую модель, о которых набор Robotics System Toolbox не знает. Даже если будут учитываться силы, перечисленные выше, идеальной модели для отслеживания траектории достичь не получится. Для вывода необходимых крутящих моментов нужно использовать систему управления. Общая схема контура управления показана на рисунке 50. Блок «Расчитанный крутящий момент» необходим для расчета крутящего момента всех звеньев руки. Он будет использовать уравнение динамики и крутящие моменты необходимые для компенсации возмущения. Расхождения между требуемыми крутящими моментами и полученными будет устранено с помощью регулятора.

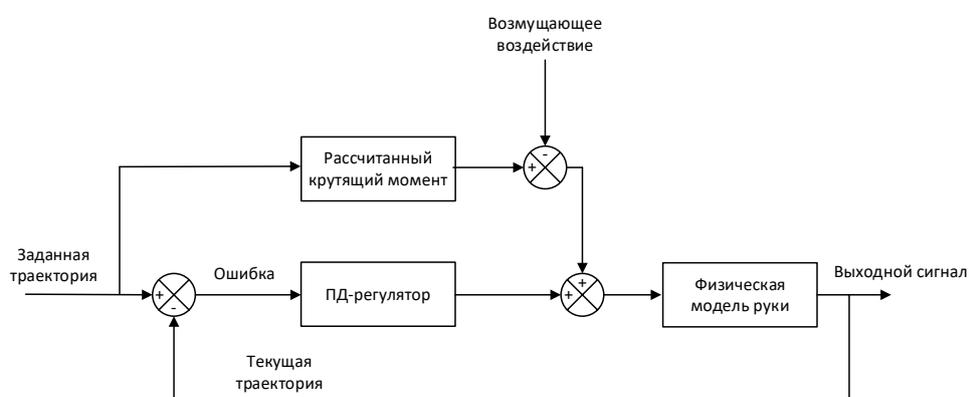


Рисунок 50 – Общая схема контура управления

На рисунке 51 показана подробная схема системы управления крутящими моментами в зависимости от заданных условий. Входящими сигналами будут являться ускорение, скорость и углы вращения.

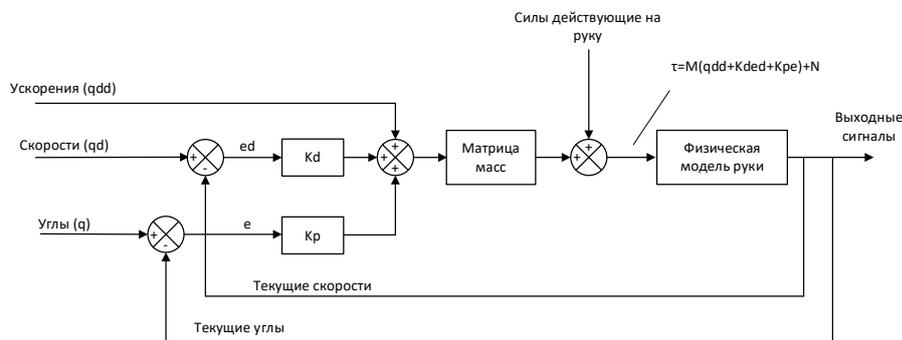


Рисунок 51 – Система управления крутящими моментами руки

В Simulink модели необходимо отстроить полученную схему управления положением манипулятора. Система управления получает информацию о положении сустава и скорости от модели руки и отправляет команды крутящего момента для приведения робота в желаемое положение сустава, рассчитанное с использованием обратной кинематики. На рисунке 52 показана блочная диаграмма подсистемы «Torque Controller», в которой находится система управления. На выходе подсистемы находятся необходимые крутящие моменты для приведения руки в заданное положение.

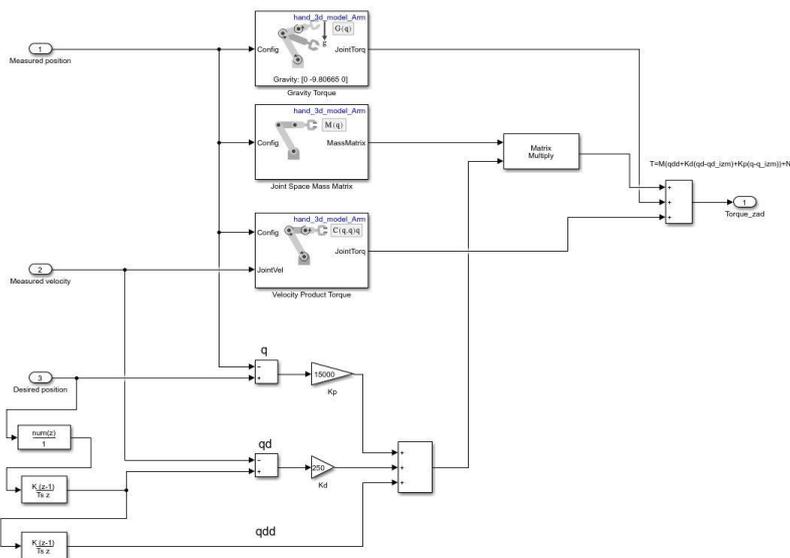


Рисунок 52 – Внешний вид подсистемы «Torque Controller»

В данную подсистему входят два блока вычисляющие крутящий момент на основе сил, действующих на руку и блок для вывода массовой матрицы совместного пространства для данной конфигурации руки, возвращенная как положительно определенная симметричная матрица. Данным блокам необходимо передать дерево твердого тела (RigidBodyTree). Для удобства структуры RigidBodyTree сохранены в рабочей папке Matlab.

Уравнение из выражения 13 в данном случае будет выглядеть следующим образом:

$$M(\ddot{q} + K_d \dot{q}_e + K_p q_e) + N = \tau, \quad (14)$$

где M – матрица масс совместного пространства;

\dot{q}_e, q_e – ошибка по скорости и ошибка по положению;

N – возмущающие воздействия в виде сил действующие на манипулятор;

τ – вектор крутящих моментов.

В системе управления вычисляющая крутящий момент есть два настраиваемых параметра коэффициент усиления K_p : пропорциональный коэффициент усиления и коэффициент усиления K_d : дифференциальный коэффициент усиления. Стандартный способ определения K_p и K_d заключается в вычислении их как:

$$K_p = \omega_n^2, \quad (15)$$

где ω_n – собственная частота колебаний системы.

$$K_d = 2\omega_n \xi, \quad (16)$$

где ξ – коэффициент затухания.

Значения K_p и K_d получены путем установки собственной частоты и коэффициента демпфирования как $\omega_n=120$ и $\xi=1$, чтобы сделать систему с критическим демпфированием [43]. Не зависимо от выбора данных коэффициентов рукой рано или поздно будет достигнуто заданное положение.

Для ввода в систему управления необходимой конфигурации воспользуемся подсистемой «Inverse Kinematics» из предыдущей отстроенной модели. На вход подаются желаемые значения координат x, y, z блоки для ввода также взяты из предыдущей Simulink-модели. Связь между подсистемами «Inverse Kinematics» и «Torque Controller» показана на рисунке 53.

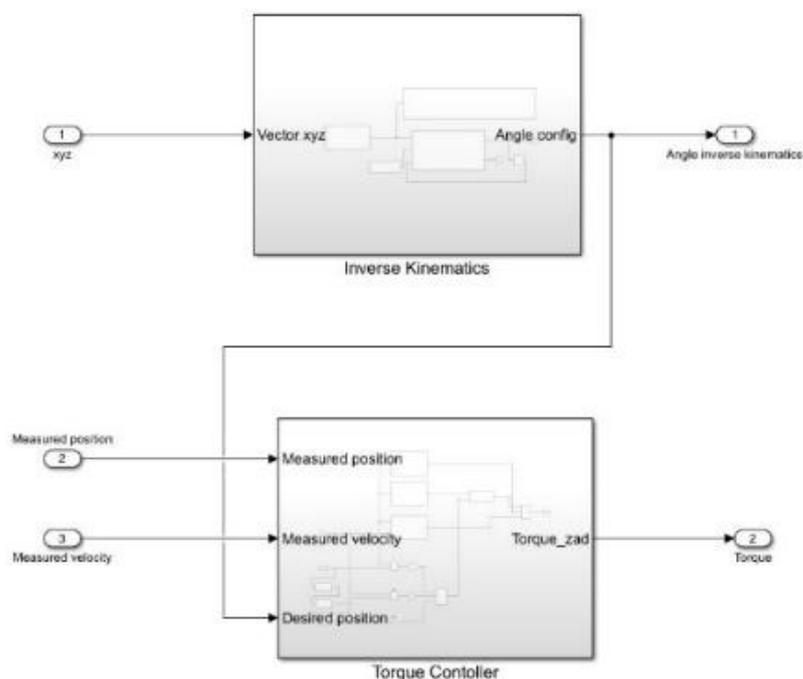


Рисунок 53 – Подсистема «Torque Control»

На рисунке 54 показана Simulink-модель, где входными данными в физическую модель являлась угловая конфигурация. На рисунке 55 демонстрируется готовая блочная диаграмма Simulink-модели для управления крутящими моментами подаваемых на физическую модель руки.

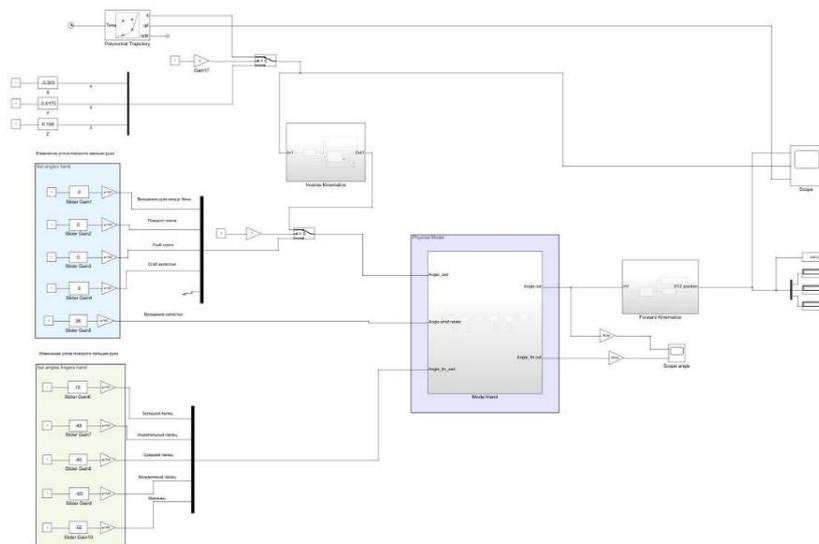


Рисунок 54 – Simulink-модель для ввода угловой конфигурации в физическую модель

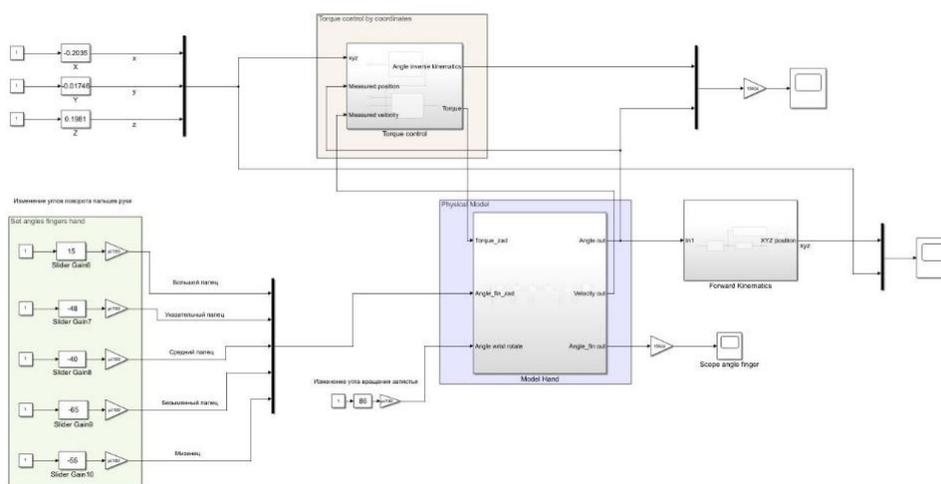


Рисунок 55 – Simulink-модель для ввода крутящих моментов в физическую модель

Для управления моделированием модели и задания входных значений будет разработан графический интерфейс пользователя. Описание управления кистью руки представлено ниже.

3.1.5 Программная реализация алгоритма по управлению кистью руки

Для реализации управления кистью необходимо задавать углы вращения на рычаги установленные на сервоприводах. Такой же принцип реализуется на

модели Simulink. При построении блочной диаграммы с помощью утилиты Simscape Multibody Link отстраивается физическая модель кисти. На рисунке 56 представлен внешний вид подсистемы «Wrist». К основанию кисти, за которое отвечает блок «BaseWrist» прикреплены пять пальцев. В данной подсистеме находится 12 преобразований координатных систем, которые определяют фиксированное трехмерное жесткое преобразование между двумя телами.

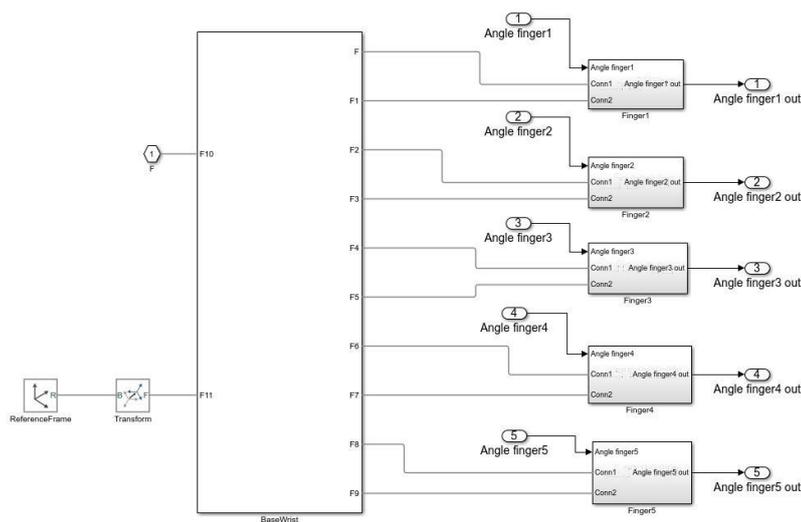


Рисунок 56 – Подсистема для управления кистью руки «Wrist»

Подсистема для приведения в движение пальца руки показан на рисунке 57. В данном исследовании примем допущение неподвижности фаланг пальцев так как для подвижности необходимо пересмотреть тип управления фалангами как на реальном объекте, так и в моделируемом.

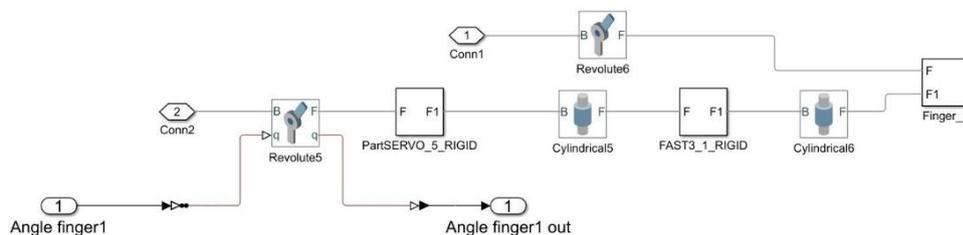


Рисунок 57 – Подсистема управления пальцем руки

Каждая подсистема пальцев однотипна той что представлена на рисунке 57. Угловое расположение рычагов в сжатом состоянии пальцев относительно

неподвижных сервоприводов было выбрано на основе расположения на реальном объекте. Задание углов для пальцев непосредственно из окна Simulink выполнено однотипно тому как управляются угловые перемещения звеньев руки. На рисунке 58 показан внешний вид области для ввода углов для каждого из пяти пальцев, на котором заданы значения углов для начального состояние – сжатие пальцев кисти руки.

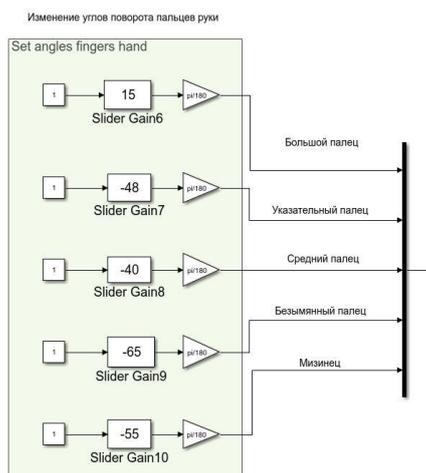


Рисунок 58 – Область для ввода значений углов поворота пальцев руки

Каждый Slider Gain настроен на максимальное и минимальное отклонение рычага. Для каждого пальца угол изменяется соответственно от 15 до 80 градусов; от - 48 до 15 градусов; от - 40 до 20 градусов; -65 до 5 градусов; от -55 до 5 градусов. Угловые значения зависят от расположения пальца относительно рычага и ладони

Для удобства управления моделью руки необходимо создать общее окно для ввода и вывода значений. Описание пользовательского интерфейса и возможностей оператора представлено в следующем разделе.

3.1.6 Описание пользовательского интерфейса и возможностей оператора

Для удобства ввода данных воспользуемся приложением с графическим пользовательским интерфейсом в Matlab. Графические пользовательские интерфейсы (GUI), также известные как приложения, обеспечивают управление про-

граммными приложениями в режиме "укажи и щелкни", устраняя необходимость для других изучать язык или вводить команды для запуска приложения [44].

Для первой Simulink-модели необходимо предусмотреть:

- область для ввода режима моделирования руки;
- область для задания углов вращения;
- область для задания координат;
- область для управления пальцами руки;
- область для выведения результатов расчета координат по ДХ-представлению.

Для второй Simulink-модели, где входными данными в физическую модель являются крутящие моменты необходимо предусмотреть:

- область для ввода режима моделирования руки;
- область для задания координат и угла вращения запястья;
- область для управления пальцами руки.

Создание GUI интерфейса сопровождается двумя файлами формата fig и m. В первом файле создается окно с интерактивными элементами управления с помощью инструмента GUIDE [45]. Во втором файле командный код для GUIDE и Simulink-модели. В данной работе будет четыре файла, по два на каждую модель.

Перечислим входящее в области интерфейса для первой модели Simulink. Область для ввода режима управления включает:

- кнопку «Запуск», которая запускает моделирование в окне Simulink;
- кнопку «Пауза», которая приостанавливает моделирование в окне Simulink;
- кнопку «Продолжить», которая продолжает моделирование после паузы в окне Simulink;
- кнопку «Стоп», которая останавливает моделирование в окне Simulink;
- кнопку «Нулевое положение», которая задает начальную конфигурацию для модели руки;

– кнопку «Отстроить траекторию», которая переключает «Switch1» и «Switch2» в окне Simulink для того чтобы передать сигнал от блока полиномиальной траектории в физическую модель.

Область для задания углов вращения включает:

– пять элементов Slider, настроенных для каждого угла поворота сустава соответственно;

– пять элементов Edit Text, настроенных для вывода значений каждого угла поворота сустава соответственно.

При перетаскивании любого из пяти ползунков переключается «Switch1» в Simulink для пропуска сигнала на изменение угла.

Область для задания координат включает:

– три элемента Slider, настроенных на изменения значений координат P_x , P_y , P_z ;

– три элемента Edit Text, настроенных для вывода значений координат соответственно.

При перетаскивании любого из трех ползунков переключается «Switch1» и «Switch2» в Simulink для пропуска сигнала на изменение координат.

Область для управления пальцами руки включает:

– две кнопки выпрямить и согнуть пальцы, передающие в Simulink-модель начальные и конечные угловые значения на каждый палец;

– пять ползунков для задания угловых значений вручную.

Область для выведения координат x , y , z из матрицы однородного преобразования из ДХ-представления, расчет, которой ведется в командном коде для GUI-интерфейса включает три элемента Edit Text.

Внешний вид первого интерфейса представлен на рисунке 59.

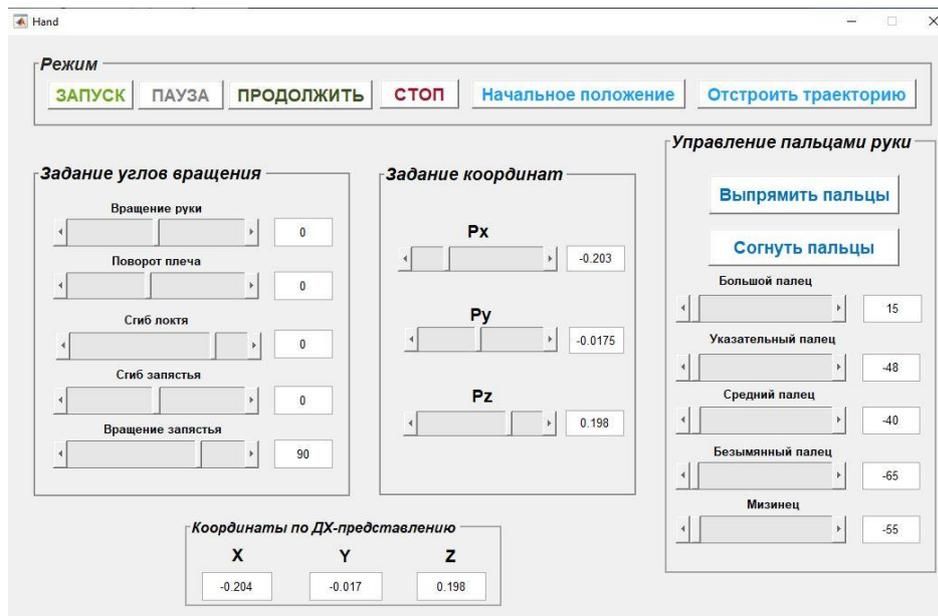


Рисунок 59 – Внешний вид GUI-интерфейса для первой Simulink-модели

Перечислим входящее в области интерфейса для второй модели Simulink.

Область для ввода режима управления включает:

- кнопку «Запуск», которая запускает моделирование в окне Simulink;
- кнопку «Пауза», которая приостанавливает моделирование в окне Simulink;
- кнопку «Продолжить», которая продолжает моделирование после паузы в окне Simulink;
- кнопку «Стоп», которая останавливает моделирование в окне Simulink;
- кнопку «Начальное положение», которая задает начальную конфигурацию для модели руки.

Область для задания входных воздействий включает:

- три элемента Slider, настроенных на изменения значений координат P_x , P_y , P_z ;
- три элемента Edit Text, настроенных для вывода значений координат соответственно;
- один элемент Slider для ввода значений угла для вращения запястья;
- один элемент Edit Text для вывода значений угла для вращения запястья.

Область для управления пальцами руки включает:

- две кнопки выпрямить и согнуть пальцы, передающие в Simulink-модель начальные и конечные угловые значения на каждый палец;
 - пять ползунков для задания угловых значений вручную.
- Внешний вид второго интерфейса представлен на рисунке 60.

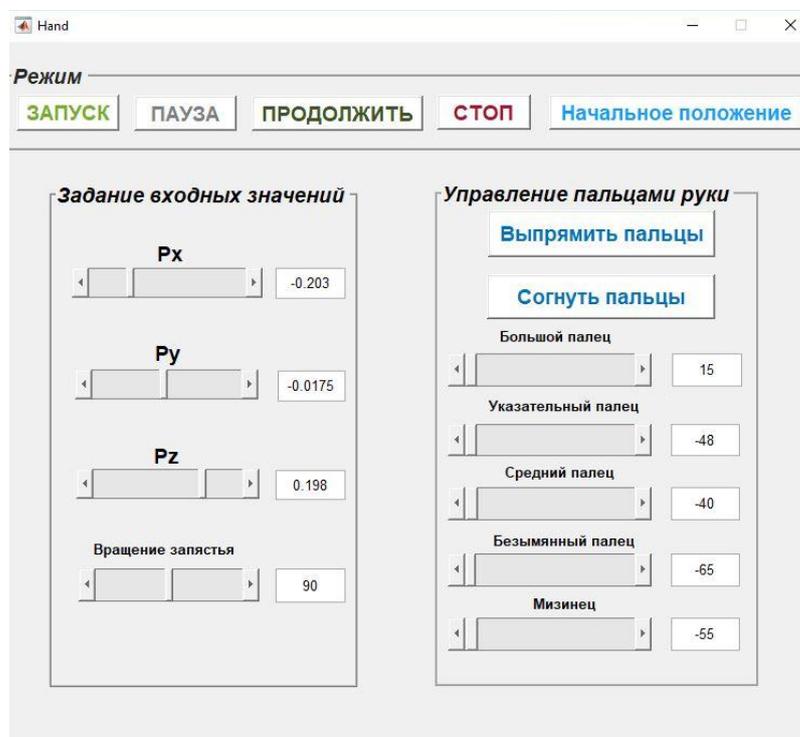


Рисунок 60 – Внешний вид GUI-интерфейса для второй Simulink-модели

Выходные данные доступны в окне Simulink в блоках Scope и Edit. Их можно посмотреть во время и после моделирования. После повторного запуска моделирования данные перезаписываются на новые.

3.2 Тестирование программного продукта

Тестирование – это процесс проверки того, что созданный программный продукт соответствует заданным требованиям и не содержит ошибок. Тестирование может включать в себя запуск программного обеспечения или его компонентов с использованием ручных или автоматизированных инструментов тестирования программного обеспечения [46]. Также могут быть протестированы отдельные компоненты или отдельные функции программного обеспечения.

В данном квалификационном исследовании требуется протестировать управление звеньями руки, управление по положению в трех разных вариациях и управление кистью руки.

3.2.1 Тестирование программы для управления звеньями руки

Внешний вид содержимого файла «Hand3d», где хранятся все необходимые файлы для моделирования движения руки показан на рисунке 61. Данный файл должен быть открыт в рабочей области Matlab.

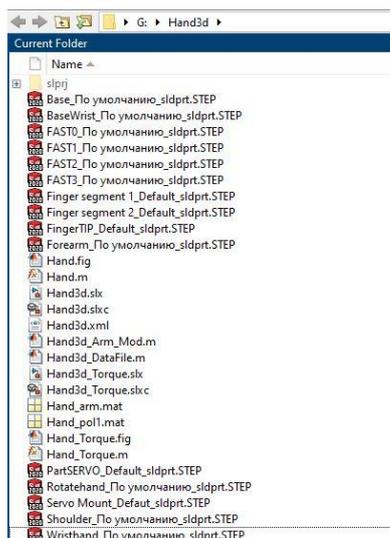


Рисунок 61 – Внешний вид содержимого файла «Hand3d»

В рамках тестирования функции программного продукта такой как управление звеньями руки необходимо выполнить следующие действия:

- запустить программу Matlab и задать в области «Current Folder» рабочую папку с моделью как показано на рисунке 61;
- открыть первую Simulink-модель «Hand3d.slx»;
- загрузить данные из файлов «Hand_arm.mat» и «Hand_poll.mat», которые хранят модель дерева твердого тела модели и матрицу положения для полиномиальной траектории;
- запустить файл «Hand.m» и запустить Simulink-модель нажав на кнопку «Запуск» в открывшемся окне GUI-интерфейса.

В рамках тестирования необходимо провести сравнение положения руки по трем координатам на основе вычисления матрицы однородного преобразования и вычисления блока «Forward Kinematics». Результаты сравнения представлены в таблице 6.

Таблица 6 – Сравнительный анализ вычислений прямой кинематики

№	Углы (θ_i), заданные оператором, °	Координаты x, y, z по ДХ-представлению, м	Координаты x, y, z полученные в Matlab, м
1	$\theta_1 = 0; \theta_2 = 0;$ $\theta_3 = 0; \theta_4 = 0;$	x = -0,204; y = -0,017; z = 0,198.	x = - 0,2035; y = - 0,01746; z = 0,1981.
2	$\theta_1=25; \theta_2=5;$ $\theta_3=0; \theta_4=0;$	x = -0,177; y = -0,010129; z = 0,21578.	x = - 0,174; y = - 0,0952; z = 0,2153.
3	$\theta_1 = -36; \theta_2 = -7;$ $\theta_3 = 5; \theta_4 = 10;$	x = -0,17251; y = 0,10432; z = 0,16649.	x = - 0,1788; y = - 0,01047; z = 0,1655.
4	$\theta_1 = -36; \theta_2 = -7;$ $\theta_3 = -22; \theta_4 = -30;$	x = - 0,16941; y = 0,010207; z = 0,21819.	x = - 0,1752; y = 0,01024; z = 0,2174.
5	$\theta_1=40; \theta_2=0;$ $\theta_3=-10; \theta_4=40;$	x = - 0,12647; y = - 0,12831; z = 0,27075.	x = - 0,1335; y = - 0,125; z = 0,2707.
6	$\theta_1=-10; \theta_2=0;$ $\theta_3=20; \theta_4=-15;$	x = - 0,18426; y = 0,015227; z = 0,11364.	x = - 0,1827; y = 0,01311; z = 0,1133.
7	$\theta_1 = -25; \theta_2 = -15;$ $\theta_3 = -70; \theta_4 = -30;$	x = - 0,13223; y = 0,042902; z = 0,34013.	x = - 0,1523; y = 0,0493; z = 0,3368.
8	$\theta_1 = 54; \theta_2 = 0;$ $\theta_3 = -30; \theta_4 = -30;$	x = - 0,09501; y = - 0,15975; z = 0,2685.	x = - 0,1051; y = - 0,01573; z = 0,2681.

Разница между методами расчетов в процентах составила от 0 – 5% в зависимости от координаты. Есть несколько факторов, которые влияют на расхождение: 3D форма модели; нахождение базовых систем координат в Matlab вокруг

которой вращается сустав; округление значений, как в расчетах матриц в командном коде, так и в Simulink-модели.

Все полученные координаты за время моделирования оператор может получить, введя команду в Command Window `get(out.xyz)`. В рабочей области появится структура с необходимыми данными. Также просмотр выходных данных доступен в блоке осциллографа Scope.

На рисунках 62 и 63 показаны окна Mechanics Explorers, демонстрирующие модель руки в начальной конфигурации и в последней из заданных.

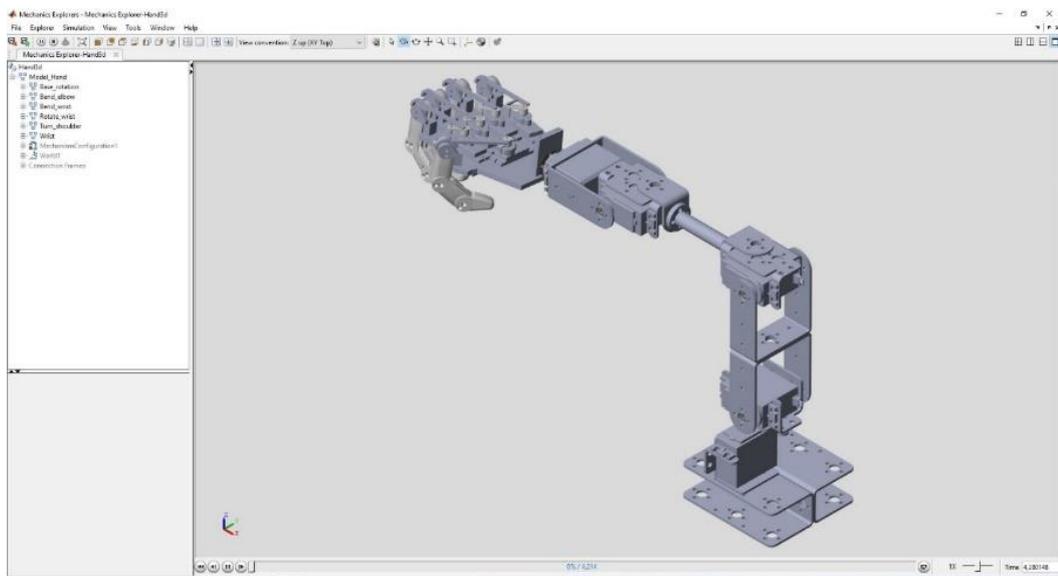


Рисунок 62 – Визуализация 3D модели при нулевой конфигурации руки

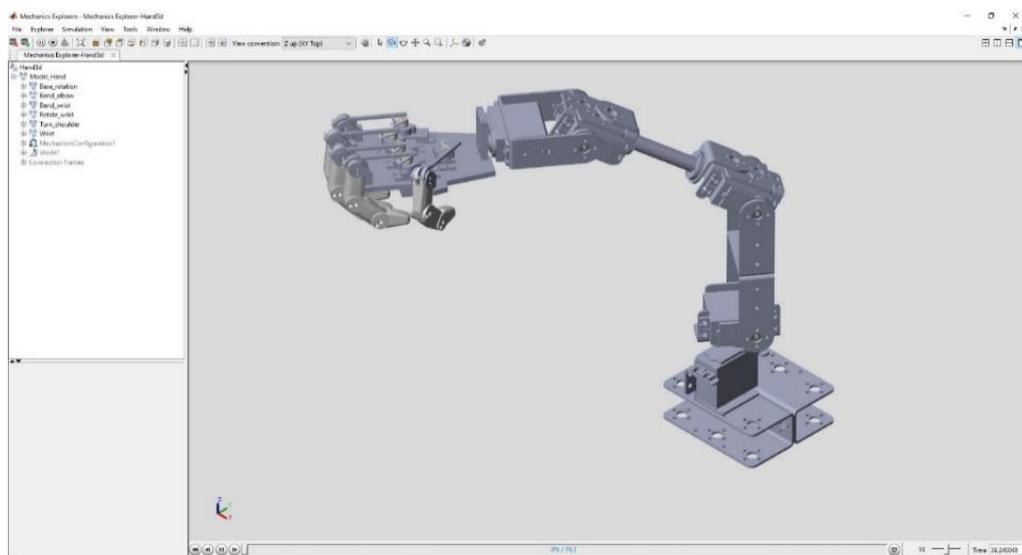


Рисунок 63 – Визуализация 3D модели при заданной конфигурации руки

Далее в Simulink-модель «Hand3d.slx» необходимо проверить на корректность работы подсистему обратной кинематики и построение траектории.

3.2.2 Тестирование программы для управления положением руки

Для проверки работоспособности модели руки при задании координат необходимо выполнить действия, описанные выше, в случае если оператор продолжает работать с моделью после задания угловых значений необходимо привести руку в начальное положение нажав на соответствующую кнопку в графическом интерфейсе.

Для тестирования необходимо выбрать координаты из рабочей области руки и ввести значение с помощью ползунков в области задания координат. В таблице 7 представлен анализ введенных координат в алгоритм обратной кинематики с координатами, полученными на выходе алгоритма прямой кинематики после прохождения физической модели.

Таблица 7 – Сравнительный анализ координат

№	Координаты x, y, z заданные оператором, м	Координаты x, y, z полученные в Matlab, м
1	$x = - 0,24234; y = - 0,012308; z = 0,22769.$	$x = - 0,242; y = - 0,01231; z = 0,227.$
2	$x = - 0,061314; y = - 0,11692; z = 0,17231.$	$x = - 0,06131; y = - 0,1169; z = 0,1723.$
3	$x = 0,055474; y = - 0,22154; z = 0,27077.$	$x = - 0,05547; y = 0,2215; z = 0,2708.$
4	$x = 0,09635; y = - 0,073846; z = 0,27696.$	$x = 0,09635; y = - 0,07385; z = 0,2769.$
5	$x = - 0,050474; y = - 0,19692; z = 0,12923.$	$x = - 0,05047; y = - 0,11969; z = 0,1292.$

По полученным данным из анализа можно сделать вывод о том, что модель руки точно следует входным значениям. На рисунке 64 показан вывод осциллографа, на котором видно, что входные и выходные сигналы накладываются друг

на друга. Небольшие отклонения в моменте допускаются, так как есть задержка в расчете.



Рисунок 64 – График входных и выходных положений руки

Для отстройки траектории необходимо остановить моделирование если оно было запущено, затем необходимо в окне интерфейса нажать на кнопку отстроить траекторию, это связано с тем, что переключатели Switch1 и Switch2 перед запуском модели нужно перевести в состояние >0 . Так как блок полиномиальной траектории выдает результат непрерывно в течении от 1 до 4 секунд как показано на рисунке 46, для наблюдения за ним требуются описанные выше условия, иначе путь руки начнет не с начальной заданной точки.

На рисунке 47 задана матрица для восьми путевых точек для каждой координаты, где первая и последняя путевые точки – это начальное положение руки. Необходимо протестировать следует ли рука заданным путевым точкам. На рисунке 65 показан вывод осциллографа, на нём видно, что модель руки точно следует заданным путевым точкам. На 4 секунде рука заняла последнее исходное положение. Также на полученном графике можно отследить скорость, с которой конечный эффектор двигался по точкам.

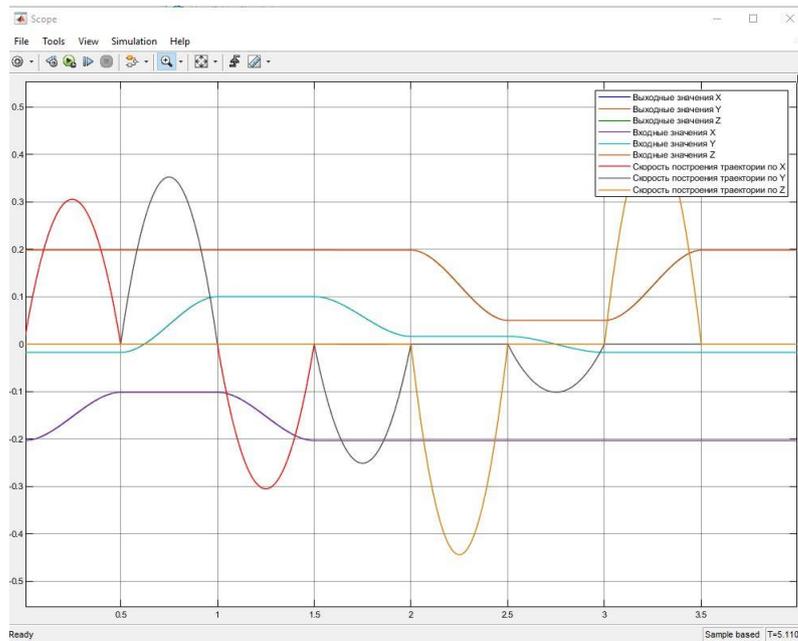


Рисунок 65 – График результата отслеживания траектории

Также в рамках данного тестирования необходимо проверить на работоспособность Simulink-модели для расчета крутящего момента для компенсации возмущений, действующие на руку. На вход модели поступает конфигурация положения из трех координат, которую необходимо ввести с помощью ползунков в GUI-интерфейсе. Для значений $x = -0,155$; $y = -0,0912$; $z = 0.16$ полученный график переходных процессов угловых значений на каждом шарнире показан на рисунке 66. На рисунке 67 показан график переходных процессов для координат.

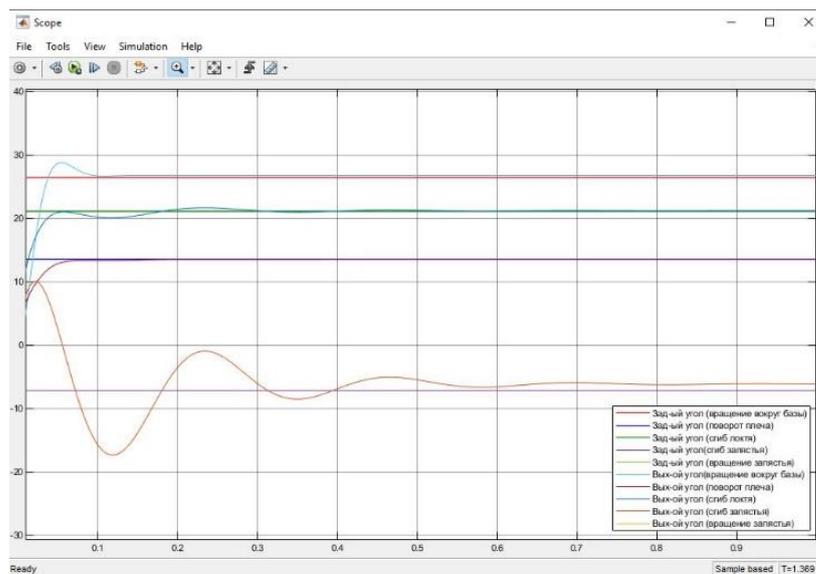


Рисунок 66 – Переходный процесс угловых значений шарниров

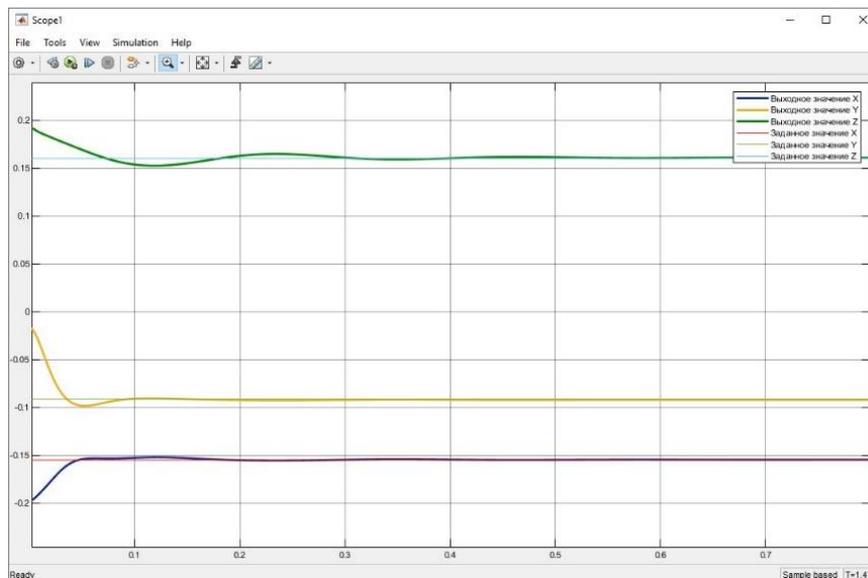


Рисунок 67 – Переходный процесс для заданных координат

Как видно из рисунков достижение заданных координат происходит быстро ($t_{\text{пер}} = 0,4$ сек.) и точно. В угловом значении ошибка в 1 градус наблюдается у шарнира для сгиба запястья, так как он находится последним из звеньев, на него могут действовать.

3.2.3 Тестирование программы для управления кистью руки

Управления кистью руки доступно как в первой, так и во второй модели Simulink в зависимости от типа входных воздействий подаваемых на манипулятор (руку). Реализация управления между ними не отличается, так как область для ввода значений и команд на пальцы кисти в GUI-интерфейсах идентична. По вводу значений на ползунках модель реагирует на изменение значений и это можно наблюдать на симуляции. Также доступно достижение крайних положений на всех пальцах одновременно.

На рисунке 68 показан график изменений углов на каждом пальце, а также достижение конечных положений всех пальцев одновременно. На рисунке 69 показано крайнее положение «Выпрямить пальцы».

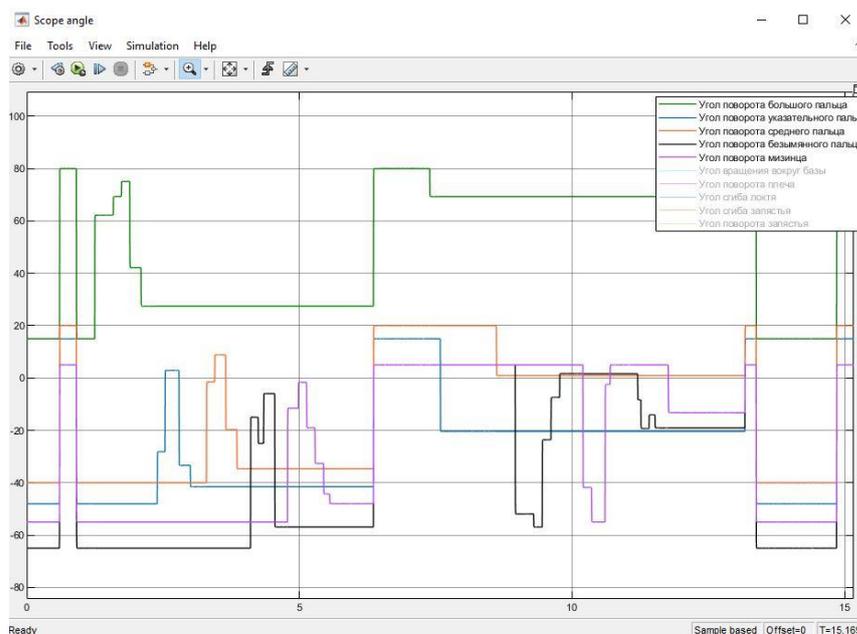


Рисунок 68 – Вывод графика изменения углов пальцев руки

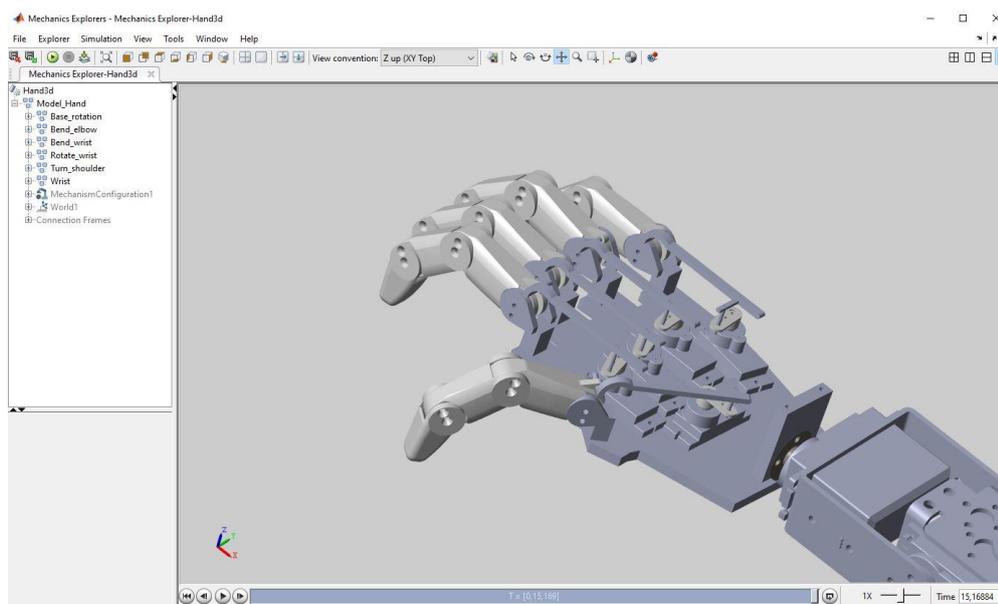


Рисунок 69 – Внешний вид крайнего положения «Выпрямить пальцы»

Цель проведения тестирования состояла в том, чтобы рассмотреть все возможные варианты работы программы, выявить недостатки и устранить их, если таковые имели место. В результате испытаний частей программы было доказано, что данная программа работает согласно алгоритму. Тестирование показало, что разработанный программный продукт работоспособен и может быть применен для дальнейшего изучения антропоморфных роботов-манипуляторов.

3.3 Анализ достоверности и практической значимости результатов моделирования

В рамках квалификационной работы была создана трехмерная модель на основе реального многозвенного антропоморфного манипуляционного устройства, конструктивно схожего с человеческой рукой. Данный тип роботов занимает важную роль в изучении и разработке систем управления, а также высоко интересен со стороны как промышленности, так и для научных исследований в области робототехники и не только. Для манипуляционных роботов разрабатываются различные подходы к решению задач построения алгоритмов управления робототехнической системой, вызванные их многозвенной структурой.

Для проведения экспериментов над конфигурацией реального оборудования затрачивается его срок службы, для недопущения выхода из строя дорого оборудования построена модель с системой управления для контроля положения руки с помощью задания углов и крутящего момента для вращения шарниров на физической модели с помощью пакета прикладных программ Matlab.

На компьютерной модели проверены результаты составленной кинематической структуры, также по результатам моделирования руки, можно сделать вывод, что модель точно следует заданным конфигурациям и проходит все заданные путевые точки при задании траектории. Отклонения между расчетами координат x , y , z с применением ДХ-представления и Matlab и алгоритмом кинематики с применением блоков библиотеки Robotics System Toolbox в зависимости от заданных углов оператором составила от 0 – 5 %, что соответствует 0 – 9 мм. Что говорит о правильности составления матрицы однородного преобразования и работы модели. Отклонения между заданными значениями координат и полученными после прохождения физической модели не наблюдаются. Полученные угловые конфигурации для заданных положений руки можно передавать в микроконтроллер реального манипуляционного устройства.

ЗАКЛЮЧЕНИЕ

В ходе выполнения магистерской диссертации была разработана трехмерная модель реального антропоморфного манипуляционного устройства, по внешнему виду напоминающего человеческую руку и создан программный продукт для моделирования ее движения.

Авторская разработка заключается в предложении системы управления моделью руки для моделирования её движения и детального изучения структуры манипуляционного устройства с использованием программного обеспечения САПР SolidWorks и библиотеки Simscape Multibody в графической среде моделирования Simulink в Matlab. Данные программы выбраны из-за возможности интеграции программы SolidWorks с Simulink с помощью плагина Simscape Multibody Link для экспортирования отстроенной модели. Стоит отметить то, что выбранный программный пакет SolidWorks и Matlab с библиотеками Simscape и Robotics System Toolbox может быть применен к любому типу манипуляционных роботов.

Для удобства управления моделью руки были созданы GUI-интерфейсы, через которые пользователь программы сможет задавать угловые конфигурации для шарниров, координаты для перемещения руки и режимы управления моделью. Полученные результаты во время моделирования, а точнее угловые положения, координаты рабочей области могут быть переданы напрямую в микроконтроллер реального манипуляционного устройства для достижения положения или выполнения необходимой траектории.

Разработанная компьютерная модель поможет разработчикам систем управления рукой проводить эксперименты и тестирование на модели и точно определять положение звеньев для достижения требуемой точки в пространстве, а также может быть применена в учебных целях для изучения структуры манипуляционных устройств.

БИБЛИОГРАФИЧЕСКИЕ ССЫЛКИ

- 1 Колюбин С. А. Динамика робототехнических систем. СПб.: Университет ИТМО, 2017. 117 с.
- 2 Хомченко В. Г. Робототехнические системы. Омск: Омский гос. ун-т, 2016. 195 с.
- 3 Юревич Е. И. Основы робототехники: 2-е изд. СПб.: БХВ – Петербург, 2005. 416 с.
- 4 Ахмадиев Ф. Г., Гильфанов Р. М. Математическое моделирование и методы оптимизации. Казань: Изд-во Казанск. гос. архит.-строит. ун-та, 2017. 178 с.
- 5 Gazebo [Электронный ресурс]: schem.ru. URL: <https://schem.net/software/gazebo.php> (Дата обращения: 20.09.2022).
- 6 Бжихатлов И. А. Моделирование робототехнических систем в программе V-REP: Учебно-метод пособие. СПб: Университет ИТМО, 2018. 59с.
- 7 SimScare: Моделирование и симуляция междисциплинарных физических систем [Электронный ресурс]: exponenta.ru. URL: <https://exponenta.ru/simscape> (Дата обращения: 20.10.2022).
- 8 Системы автоматизированного проектирования (САПР) [Электронный ресурс]: studopedia.ru. URL: https://studopedia.net/19_33144_sistemi-avtomatizirovannogo-proektirovaniya-sapr.html (Дата обращения: 20.10.2022).
- 9 Климчик А. С., Гомолицкий Р. И., Фурман Ф. В., Сёмкин К. И. Разработка управляющих программ промышленных роботов. Минск: Белорусский гос. ун-т, 2008. 131 с.
- 10 Разработка логической модели ПК [Электронный ресурс]: studbooks.net. URL: https://studbooks.net/2428166/informatika/razrabotka_logicheskoy_modeli (Дата обращения: 20.10.2022).
- 11 Что такое SolidWorks [Электронный ресурс]: dic.academic.ru URL: <https://dic.academic.ru/dic.nsf/ruwiki/133966> (Дата обращения: 20.10.2022).

12 Каталог решений SolidWorks [Электронный ресурс]: idtsoft.ru. URL: <https://www.idtsoft.ru/sites/default/files/fields/media/file/field-media-file/2021-02/solidworks-idt-katalog2020-all.pdf> (Дата обращения: 03.11.2022).

13 Simscape Multibody [Электронный ресурс]: docs.exponenta.ru. URL: <https://docs.exponenta.ru/physmod/sm/index.html> (Дата обращения: 03.11.2022).

14 Matlab r2019b [Электронный ресурс]: mathworks.ru. URL: https://www.mathworks.com/products/new_products/release2019b.html (Дата обращения: 03.11.2022).

15 Simscape Multibody Link [Электронный ресурс]: solidworks.com <https://www.solidworks.com/partner-product/simscape-multibody-link> (Дата обращения: 03.11.2022).

16 Установите плагин Simscape Multibody Link [Электронный ресурс]: docs.exponenta.ru. URL: <https://docs.exponenta.ru/physmod/sm/ug/installing-and-linking-simmechanics-link-software.html> (Дата обращения: 03.11.2022).

17 Smimport [Электронный ресурс]: docs.exponenta.ru. URL: <https://docs.exponenta.ru/physmod/sm/ref/smimport.html> (Дата обращения: 03.11.2022).

18 Мусалимов В.М., Заморуев Г.Б., Калапышина И.И., Перечесова А.Д., Нуждин К.А. Моделирование мехатронных систем в среде Matlab (Simulink / Simmechanics): учеб. пособие. СПб: НИУ ИТМО, 2013. 114 с.

19 Умнов В. П., Мишулин Ю. Е. Моделирование динамики механизмов и управляемого движения исполнительных устройств манипуляционных роботов: учеб. пособие. Владимир: Изд-во ВлГУ, 2019. 152 с.

20 File solid [Электронный ресурс]: docs.exponenta.ru. URL: <https://docs.exponenta.ru/R2020a/physmod/sm/ref/filesolid.html> (Дата обращения: 03.11.2022).

21 Мусалимов В.М., Заморуев Г.Б., Калапышина И.И., Перечесова А.Д., Нуждин К.А. Моделирование мехатронных систем в среде Matlab (Simulink / Simmechanics): учеб. пособие. С 5.

22 Ахмадиев Ф.Г., Гильфанов Р.М. Математическое моделирование и методы оптимизации: Учеб. пособие С 7.

23 Климчик А. С., Гомолицкий Р. И., Фурман Ф. В., Сёмкин К. И. Разработка управляющих программ промышленных роботов. С 14.

24 Борисов О. И., Громов В. С., Пыркин А. А. Методы управления робототехническими приложениями: учеб. пособие СПб.: Университет ИТ-МО, 2016. 108 с.

25 Осипенко Л. Е. Компьютерное моделирование движения руки / Молодёжь 21 века: шаг в будущее: материалы 23 региональной научно-практ. конф. 2022. Т.4. С. 199-201

26 Умнов В. П., Мишулин Ю. Е. Моделирование динамики механизмов и управляемого движения исполнительных устройств манипуляционных роботов: учеб. пособие. С 64.

27 Кинематическая модель роботизированного манипулятора Dobot magician [Электронный ресурс]: mnv.irgups.ru URL: <https://mnv.irgups.ru/kinematicheskaya-model-robotizirovannogo-manipulyatora-dobot-magician> (Дата обращения: 03.02.2023).

28 Гузненков В. Н., Журбенко П. А., Бондарева Т. П. SolidWorks 2016: трехмерное моделирование деталей и выполнение электронных чертежей: учеб. пособие М: МГТУ им Н.Э. Баумана, 2017. 124 с.

29 ГОСТ 2.051-2013 Единая система конструкторской документации (ЕСКД). Электронные документы. М. 2014. 17 с.

30 ГОСТ 2.052-2021 Единая система конструкторской документации. Электронная модель изделия. Общие положения М. 2022. 10 с.

31 Гузненков В. Н., Журбенко П. А., Бондарева Т. П. SolidWorks 2016: трехмерное моделирование деталей и выполнение электронных чертежей: учеб. пособие. С 12.

32 Revolute Joint [Электронный ресурс]: docs.exponenta.ru. URL: <https://docs.exponenta.ru/physmod/sm/ref/revolutejoint.html> (Дата обращения: 10.02.2023).

33 Simulink-PS Converter [Электронный ресурс]: docs.exponenta.ru. URL: <https://docs.exponenta.ru/physmod/simscaperef/simulinkpsconverter.html> (Дата обращения: 10.02.2023).

34 Slider Gain [Электронный ресурс]: docs.exponenta.ru. URL: <https://docs.exponenta.ru/simulink/slref/slidergain.html> (Дата обращения: 10.02.2023).

35 Robotics System Toolbox [Электронный ресурс]: docs.exponenta.ru. URL: <https://docs.exponenta.ru/robotics/index.html> (Дата обращения: 15.02.2023).

36 importrobot [Электронный ресурс]: docs.exponenta.ru. URL: <https://docs.exponenta.ru/R2021b/robotics/ref/importrobot.html> (Дата обращения: 15.02.2023).

37 rigidBodyTree [Электронный ресурс]: docs.exponenta.ru. URL: <https://docs.exponenta.ru/robotics/ref/rigidbodytree.html> (Дата обращения: 15.02.2023).

38 Get Transform [Электронный ресурс]: docs.exponenta.ru. URL: <https://docs.exponenta.ru/robotics/ref/gettransformblock.html> (Дата обращения: 15.02.2023).

39 Coordinate Transformation Conversion [Электронный ресурс]: docs.exponenta.ru. URL: <https://docs.exponenta.ru/R2020b/robotics/ref/coordinateconversion.html> (Дата обращения: 15.02.2023).

40 Обратная кинематика/Inverse Kinematics [Электронный ресурс]: docs.exponenta.ru. URL: <https://docs.exponenta.ru/R2020a/robotics/ref/inversekinematics.html> (Дата обращения: 15.02.2023).

41 Алгоритмы инверсной кинематики [Электронный ресурс]: docs.exponenta.ru. URL: <https://docs.exponenta.ru/R2020a/robotics/ug/inverse-kinematics-algorithms.html>. (Дата обращения: 10.03.2023).

42 Полиномиальная траектория [Электронный ресурс]: docs.exponenta.ru. URL: <https://docs.exponenta.ru/robotics/ref/polynomialtrajectory.html> (Дата обращения: 02.04.2023).

43 Perform Safe Trajectory Tracking Control Using Robotics Manipulator Blocks [Электронный ресурс]: mathworks.com URL: <https://www.mathworks.com/help/robotics/ug/perform-safe-trajectory-tracking.html> (Дата обращения: 10.04.2023).

44 Matlab GUI [Электронный ресурс]: mathworks.com URL: <https://www.mathworks.com/discovery/matlab-gui.html> (Дата обращения: 15.04.2023).

45 Создание простого приложения используя GUIDE [Электронный ресурс]: docs.exponenta.ru. URL: https://docs.exponenta.ru/R2020b/matlab/creating_guis/about-the-simple-guide-gui-example.html (Дата обращения: 20.04.2023).

46 Пероцкая В. Н., Градусов Д. А. Основы тестирования программного обеспечения: учеб. пособие. Владимир: Изд-во ВлГУ, 2017. 100 с.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

- 1 Артемова, А. О. Математическое моделирование, алгоритмы и программы управления манипуляторами: автореф. дис...канд. физ.-матем. наук : 05.13.18 / А. О. Артемова.– Ульяновск: Изд-во Ульянов. гос ун-та, 2013. – 21 с.
- 2 Ахмадиев, Ф. Г. Математическое моделирование и методы оптимизации: учеб. пособие / Ф. Г. Ахмадиев, Р. М. Гильфанов. – Казань : Изд-во Казанск. гос. архитектур.-строит. ун-та, 2017. – 178 с.
- 3 Бжихатлов, И. А. Моделирование робототехнических систем в программе V-REP: учеб.-метод. пособие/ И. А. Бжихатлов. – СПб : Университет ИТМО, 2018. – 59 с.
- 4 Борисов, И. И. Имитационное моделирование мехатронных систем / И. И. Борисов, С. А. Колюбин. – СПб. : Университет ИТМО, 2020. – 103 с.
- 5 Борисов, О. И. Методы управления робототехническими приложениями: учеб. пособие / О. И. Борисов., В. С. Громов., А. А. Пыркин. – СПб. : Университет ИТМО, 2016. – 108 с.
- 6 ГОСТ 2.051-2013 Единая система конструкторской документации (ЕСКД). Электронные документы. – Взамен ГОСТ 2.051–2006; введ. 2014–06–01. –М. : Стандартинформ, 2014. – 17 с.
- 7 ГОСТ 2.052-2021 Единая система конструкторской документации. Электронная модель изделия. Общие положения. – Взамен ГОСТ 2.052–2015; введ. 2021–01–29. –М. : Стандартинформ, 2022. – 11 с.
- 8 Горитов, А. Н. Расчёт динамических характеристик манипулятора / А. Н. Горитов, М. Ф. Молокова // Доклады ТУСУРа, том 20. – 2017– № 4.– С 113-116.
- 9 Горностаева, Т. Н. Математическое и компьютерное моделирование: учеб. пособие [Электронный ресурс] / Т. Н. Горностаева, О. М. Горностаев. – Режим доступа: <https://izd-mn.com/PDF/50MNNPU19.pdf> – 02.12.2022.

10 Гузенков, В. Н. SolidWorks 2016: трехмерное моделирование деталей и выполнение электронных чертежей: учеб. пособие / В. Н. Гузенков, П. А. Журбенко, Т. П. Бондарева. – М : МГТУ им Н.Э. Баумана, 2017. – 124 с.

11 Егоров, Е. Е. Моделирование работы манипуляционного робота в программном пакете Matlab Robotics Toolbox / Е. Е. Егоров // Политехнический молодежный журнал. – 2020. – № 01(42). – 12 с.

12 Звонарев, С. В. Основы математического моделирования: учеб. пособие / С. В. Звонарев. – Екатеринбург: Изд-во Урал. ун-та, 2019. – 112 с.

13 Зенкевич, С. Л. Основы управления манипуляционными роботами / С. Л. Зенкевич, А. С. Ющенко. – М.: Изд-во МГТУ им. Н. Э. Баумана, 2004. – 480 с.

14 Зиновьев, Д. В. Основы моделирования в SolidWorks: 1-е изд / Д. В. Зиновьев; под ред. М. И. Азанова. – М.: ДМК Пресс, 2017. – 240 с.

15 Зуева, С. В. Создание 3D модели манипулятора в Matlab Simulink [Электронный ресурс] / С. В. Зуева, А. С. Беляев // XV Международная научно-практическая конференция студентов аспирантов и молодых учёных. – Режим доступа: https://earchive.tpu.ru/bitstream/11683/46547/1/conference_tpu-2017-C04_p104-105.pdf. – 07.02.2023.

16 Иванова, С. Б. Роботизированные компьютерно-аппаратные комплексы широкого назначения: необходимость и проблемы создания / С. Б. Иванова, И. С. Сальников, Р. И. Сальников // Искусственный интеллект. – 2015. – № 0 (1). – С. 50 - 62.

17 Климчик, А. С. Разработка управляющих программ промышленных роботов / А. С. Климчик, Р. И. Гомолицкий, Ф. В. Фурман, К. И. Сёмкин. – Минск: Белорусский гос. ун-т, 2008. – 131 с.

18 Ключев, С. А. Компьютерное моделирование: учебно-методическое пособие / С. А. Ключев. – М.: Волжский политехнический институт, 2009. – 89 с.

19 Колюбин, С. А. Динамика робототехнических систем: учеб. пособие / С. А. Колюбин. – СПб.: Университет ИТМО, 2017. – 117 с.

20 Корецкий, А. В Компьютерное моделирование кинематики манипуляционных роботов / А. В. Корецкий, Н. В Осадченко. – М.: Издательство МЭИ, 2000. – 48 с.

21 Кумагина, Е. А. Модели жизненного цикла и технологии проектирования программного обеспечения: учебно-методическое пособие / Е. А. Кумагина, Е. А. Неймарк. – Нижний Новгород: Изд-во ННГУ, 2016. – 41 с.

22 Моделирование мехатронных систем в среде Matlab (Simulink / Simmechanics): учебное пособие для высших учебных заведений / В. М. Мусалимов [и др.]. – СПб: НИУ ИТМО, 2013. – 114 с.

23 Снисаренко, С. В. Кинематический анализ манипулятора с использованием Matlab [Электронный ресурс]/ С. В. Снисаренко, Н. А. Стасевич // Международная научно-техническая конференция. – Минск : БНТУ, 2021. – Режим доступа: <https://rep.bntu.by/handle/data/99060>. – 07.02.2023.

24 Надёжина, С. Э. Кинематическая модель роботизированного манипулятора dobot magician [Электронный ресурс] / С. Э. Надёжина, М. А. Капистер, С. В. Ковыршин, С. П. Круглов // Молодая наука Сибири: электрон. науч. журн. – 2021. – № . 1 – Режим доступа: <https://mnv.irkups.ru/toma/111-2021>. – 07.02.2023.

25 Осипенко, Л. Е. Компьютерное моделирование движения руки / Л.Е. Осипенко // Молодёжь 21 века: шаг в будущее: материалы 23 региональной научно-практической конференции. – 2022. – Т.4. – С. 199-201.

26 Осипенко, Л. Е. Управление движением 3D модели руки с помощью библиотеки Simscape в среде Matlab-Simulink / Л. Е. Осипенко // Лучшие научные исследования студентов и учащихся: сб. статей Международной научно-практической конференции. – Пенза: МЦНС «Наука и Просвещение». – 2022. – Ч.1. – С. 73-76.

27 Пероцкая, В. Н. Основы тестирования программного обеспечения: учеб. пособие / В. Н. Пероцкая, Д. А. Градусов. – Владимир : Изд-во ВлГУ, 2017. – 100 с.

28 Пикалев, Я. С. Анализ существующих симуляторов робототехнических систем / Я. С. Пикалев // Проблемы искусственного интеллекта. –2017. – № 1 (4). – С. 51-65.

29 Ровбо, М. А. Обзор средств имитационного моделирования коллективов роботов с элементами социальной организации / М. А. Ровбо, Е. Е. Овсянникова, А. А. Чумаченко // Программные продукты и системы. – 2017. – №3. – С. 425-434.

30 Свидетельство о государственной регистрации программы для ЭВМ 2023615751. Программа для моделирования движения руки / Л. Е. Осипенко, А. В. Бушманов ; Амурский государственный университет. – № 2023613233 ; Заявл. 22.02.2023; Оpubл 17.03.2023, Бюл. № 3.

31 Тарасевич, Ю. Ю. Математическое и компьютерное моделирование. Вводный курс / Ю. Ю. Тарасевич. – Изд. 6-е. – М.: ЛИБРОКОМ, 2013. –152 с.

32 Тертычный-Даури, В. Ю. Динамика робототехнических систем / В. Ю. Тертычный-Даури. – СПб.: НИУ ИТМО, 2012. – 128 с.

33 Тимофеев, Г. А. Теория механизмов и механика машин: учебник для вузов / Г. А Тимофеев. – М.: Издательство МГТУ им. Н.Э. Баумана, 2017.– 177 с.

34 Умнов, В.П. Моделирование динамики механизмов и управляемого движения исполнительных устройств манипуляционных роботов : учеб. пособие / В. П. Умнов, Ю. Е. Мишулин ; Владим. гос. ун-т им. А. Г. и Н. Г. Столетовых. – Владимир : Изд-во ВлГУ, 2019. – 152 с.

35 Хомченко, В. Г. Робототехнические системы : учебное пособие / В.Г. Хомченко. – Омск : Омский гос. ун-т, 2016. – 195 с.

36 Щербаков, В. С. Моделирование и визуализация движений механических систем в MATLAB: учеб. пособие / В. С. Щербаков, М. С. Корытов, А. А. Руппель, В. А. Глушец, С. А. Милюшенко. – Омск: Изд-во СибАДИ, 2007. – 84 с.

37 Эльберг, М. С. Имитационное моделирование: учеб. пособие / М. С. Эльберг, Н. С. Цыганков. – Красноярск: Сиб. федер. ун-т, 2017. – 128 с.

38 Юревич, Е. И. Основы робототехники / Е. И. Юревич. – 2-е изд., перераб. и доп. – СПб.: БХВ – Петербург, 2005. – 416 с.

39 Юревич, Е. И. Управление роботами и робототехническими системами / Е. И. Юревич. – СПб.: БХВ – Петербург, 2000. – 171 с.

40 autocad-lessons: Как работать в SOLIDWORKS [Электронный ресурс]. – Режим доступа: <https://autocad-lessons.com/kak-rabotat-v-solidworks/>. – 27.02.2023.

41 docs.exponenta.ru: Создание простого приложения используя GUIDE [Электронный ресурс]. – Режим доступа: https://docs.exponenta.ru/R2020b/matlab/creating_guis/about-the-simple-guide-gui-example.html – 20.04.2023.

42 docs.exponenta.ru: Установите плагин Simscape Multibody Link [Электронный ресурс] – Режим доступа: <https://docs.exponenta.ru/physmod/sm/ug/installing-and-linking-simmechanics-link-software.html>. – 03.11.2022.

43 exponenta.ru: ЦИТМ Экспонента [Электронный ресурс]. – Режим доступа: <https://exponenta.ru/>. – 12.03.2023.

44 exponenta.ru: Алгоритмы инверсной кинематики [Электронный ресурс]. – Режим доступа: <https://docs.exponenta.ru/R2020a/robotics/ug/inverse-kinematics-algorithms.html>. – 24.04.2023.

45 exponenta.ru: Полиномиальная траектория [Электронный ресурс]. – Режим доступа: <https://docs.exponenta.ru/robotics/ref/polynomialtrajectory.html> – 24.04.2023.

46 exponenta.ru: SimScare - моделирование и симуляция многодисциплинарных физических систем [Электронный ресурс]. – Режим доступа: <https://exponenta.ru/simscare>. – 20.10.2022.

47 Gazebo [Электронный ресурс]. – Режим доступа: <https://cxem.net/software/gazebo.php>. – 20.09.2022.

48 mathworks.com: Design position controlled manipulator using Simscape [Электронный ресурс]. – Режим доступа: <https://www.mathworks.com/help/robotics/ug/perform-safe-trajectory-tracking.html> – 24.04.2023.

49 mathworks.com: Robotics System Toolbox [Электронный ресурс]. – Режим доступа: <https://www.mathworks.com/products/robotics>. – 10.04.2023.

50 mathworks.com: Matlab GUI [Электронный ресурс]. – Режим доступа: <https://www.mathworks.com/discovery/matlab-gui.html>. – 20.04.2023.

51 Robot Kinematics: Forward and Inverse Kinematics [Электронный ресурс]. – Режим доступа: https://cdn.intechopen.com/pdfs/379/InTech-Robot_kinematics_forward_and_inverse_kinematics.pdf. – 10.03.2023.

52 Robotics Toolbox for Matlab (Release 7.1) [Электронный ресурс]. – Режим доступа: https://phoenix.goucher.edu/~jillz/cs325_robotics/robot.pdf. – 20.04.2023.

53 SimMechanics 2 User's Guide [Электронный ресурс]. – Режим доступа: <https://mecanismos2mm7.files.wordpress.com/2011/09/tutorial-sim-mechanics.pdf>. – 10.10.2022.

54 solidworks.com: Simscape Multibody Link [Электронный ресурс]. – Режим доступа: <https://www.solidworks.com/partner-product/simscape-multibody-link>. – 09.10.2022.