

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ  
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ  
ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ  
«БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ  
УНИВЕРСИТЕТ им. В.Г. ШУХОВА»  
(БГТУ им. В.Г. Шухова)

Институт информационных технологий и управляющих систем  
Кафедра «Техническая кибернетика»

Образовательная программа 221000.62 «Мехатроника и робототехника»  
(шифр, название специальности, направления)

Профиль «Мехатроника и робототехника»

( профиль, специализация, магистерская программа)

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА  
на тему:

Разработка трехступенного лабораторного  
манипулятора с механическим звахватом

Дипломник \_\_\_\_\_ Ерёмин Д.О.  
Зав.кафедрой \_\_\_\_\_ Рубанов В.Г.  
Руководитель \_\_\_\_\_ Юдин. Д.А.

К защите допустить  
Зав. кафедрой \_\_\_\_\_ /В.Г. Рубанов/  
« \_\_\_\_\_ » \_\_\_\_\_ 20 \_\_\_\_ г.

Белгород 2015г.

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ  
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ  
ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ  
«БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ  
УНИВЕРСИТЕТ им. В.Г. ШУХОВА»  
(БГТУ им. В.Г. Шухова)

**Институт информационных технологий и управляющих систем**  
**Кафедра «Техническая кибернетика»**

**Образовательная программа 221000.62 «Мехатроника и робототехника»**  
(шифр, название специальности, направления)

**Профиль «Мехатроника и робототехника»**

( профиль, специализация, магистерская программа)

Утверждаю:

Зав. кафедрой \_\_\_\_\_ В.Г. Рубанов  
« \_\_\_\_ » \_\_\_\_\_ 20 \_\_\_\_ г.

Задание

на выпускную квалификационную работу студента (ки)

**Ерёмина Дениса Олеговича**

1. Вид выпускной квалификационной работы **бакалаврская работа**  
(дипломный проект (работа), бакалаврская работа, магистерская диссертация)

2. Тема работы **«Разработка трехступенного лабораторного манипулятора с  
механическим захватом»**

утверждено приказом по университету от « **27** » **апреля** 2015г. № **2/685**

3. Срок сдачи студентом законченной работы **25.06.2015**

4. Исходные данные **техническая документация оборудования лаборатории систем  
управления техническими и робототехническими системами центра высоких  
технологий БГТУ им. Шухова**

5. Содержание выпускной квалификационной работы (перечень подлежащих разработке  
разделов)

**Глава 1 – Подходы к моделированию и управлению манипуляторами**

**Глава 2 – Требования к разрабатываемому устройству**

**Глава 3 – Разработка конструкции манипулятора**

**Глава 4 – Разработка динамической модели манипулятора**

**Глава 5 – Разработка аппаратного комплекса манипулятора**

**Глава 6 – Разработка системы управления манипулятором**

**Глава 7 – Тестирование работы манипулятора**

6. Перечень графического материала: **Цель и задачи; Кинематическая схема  
манипулятора; Проектирование схвата манипулятора; Проектирование деталей  
манипулятора; Компьютерная динамическая модель манипулятора; Динамические  
характеристики манипулятора; Определение рабочей зоны манипулятора;  
Характеристики манипулятора; Аппаратная реализация системы управления на  
основе Arduino; Программа управления манипулятором; Демонстрация работы  
манипулятора**

Консультанты по работе с указанием относящихся к ним разделов

Раздел	Консультант	Задание выдал (подпись, дата)	Задание принял (подпись, дата)
--------	-------------	----------------------------------	-----------------------------------

Дата выдачи задания « 27 » апреля 2015г.

Юдин Д.А.

(подпись руководителя)

Задание принял к исполнению

Ерёмин Д.О.

(подпись выпускника)

**КАЛЕНДАРНЫЙ ПЛАН**

№ п/п	Наименование этапов работы	Срок выполнения этапов работы	Примечание
<b>1</b>	<b>Анализ подходов к моделированию и управлению манипуляторами</b>	<b>28.04 – 03.05</b>	
<b>2</b>	<b>Определение требований к разрабатываемому устройству</b>	<b>03.05 – 09.05</b>	
<b>3</b>	<b>Разработка конструкции манипулятора</b>	<b>09.05 – 15.05</b>	
<b>4</b>	<b>Разработка динамической модели манипулятора и её исследование</b>	<b>15.05 – 21.05</b>	
<b>5</b>	<b>Разработка аппаратного комплекса манипулятора</b>	<b>21.05 – 27.05</b>	
<b>6</b>	<b>Разработка системы управления манипулятором</b>	<b>27.05 – 02.06</b>	
<b>7</b>	<b>Тестирование и отладка работы манипулятора</b>	<b>02.06 – 08.06</b>	
<b>8</b>	<b>Оформление пояснительной записки выпускной квалификационной работы</b>	<b>08.06 – 14.06</b>	

Дипломник Ерёмин Д.О.

Руководитель Юдин Д.А.

## Содержание

Введение.....	5
1. Подходы к моделированию и управлению манипуляторами.....	7
1.1. Типы манипуляторов .....	7
1.2. Классификация манипуляторов.....	11
1.3. Системы управления манипулятором.....	14
1.4. Способы моделирования манипуляторов.....	16
2. Требования к разрабатываемому устройству.....	19
3. Разработка конструкции манипулятора.....	21
3.1. Построение структурной кинематической схемы манипулятора...	21
3.2. Решение прямой и обратной задачи кинематики .....	23
3.3. Выбор сервоприводов.....	27
3.4. Проектирование деталей .....	29
3.4.1. Выбор САПР .....	29
3.4.2. Проектирование 1-го и 2-го звеньев .....	30
3.4.3. Проектирование схвата .....	31
3.4.4. Проектирование основания .....	34
3.4.5. Проектирование всей конструкции манипулятора .....	36
3.5. Определение диапазонов углов поворотов звеньев .....	39
4. Разработка динамической модели манипулятора.....	41
4.1. Система виртуального моделирования Adams .....	41

					<b>Разработка трехстепенного лабораторного манипулятора с механическим захватом</b>			
Изм.	Лист	№ докум.	Подпись	Дата				
Разработал	Ерёмин Д.О.				<b>Содержание</b>	Лит.	Лист	Листов
Проверил	Юдин Д.А.						3	74
Руководит.	Юдин Д.А.					БГТУ им. В.Г. Шухова МР-41		
Норм. Контр.	Гольцов Ю.А.							
Зав. каф.	Рубанов В.Г.							

4.2. Рабочая зона манипулятора .....	43
4.3. Динамические характеристики манипулятора.....	44
4.4. Грузоподъёмность манипулятора .....	49
4.5. Математическое описание динамической модели манипулятора ..	52
5. Разработка аппаратного комплекса манипулятора.....	56
5.1. Печать деталей на 3D-принтере .....	56
5.2. Ручное изготовление деталей .....	57
5.3. Сборка манипулятора .....	59
6. Разработка системы управления манипулятором.....	61
6.1. Разработка системы управления манипулятором на основе Arduino.....	61
6.2. Разработка компьютерной программы для управления манипулятором .....	62
7. Тестирование работы манипулятора.....	68
Заключение .....	70
Список литературы .....	72
Приложения .....	74

## Введение

Манипулятор - совокупность пространственного рычажного механизма и системы приводов, осуществляющая под управлением программируемого автоматического устройства или человека-оператора действия (манипуляции), аналогичные действиям руки человека.

Промышленные роботы предназначены для замены человека при выполнении основных и вспомогательных технологических операций в процессе промышленного производства. При этом решается важная социальная задача - освобождения человека от работ, связанных с опасностями для здоровья или с тяжелым физическим трудом, а также от простых монотонных операций, не требующих высокой квалификации. Гибкие автоматизированные производства, создаваемые на базе промышленных роботов, позволяют решать задачи автоматизации на предприятиях с широкой номенклатурой продукции при мелкосерийном и штучном производстве. Промышленные роботы являются важными составными частями современного промышленного производства [1].

Компьютерное моделирование робототехнических систем имеет огромное значение в областях науки и техники. Огромное множество трудоёмкой физической работы человека в настоящее время заменено роботами, а компьютерные устройства, системы компьютерной математики значительно облегчили громоздкие вычисления и преобразования, сведя их к минимуму.

Компьютерная модель - компьютерная программа, реализующая абстрактную модель некоторой системы. Компьютерные модели используются для получения новых знаний о моделируемом объекте или для приближенной оценки поведения математических систем, слишком сложных для аналитического исследования. Если возникнут вопросы, пишите на почту [den-2d@mail.ru](mailto:den-2d@mail.ru) ©

					<b>Разработка трехстепенного лабораторного манипулятора с механическим захватом</b>		
Изм.	Лист	№ докум.	Подпись	Дата	<b>Введение</b>		
Разработал	Ерёмин Д.О.						
Проверил	Юдин Д.А.						
Руководит.	Юдин Д.А.						
Норм. Контр.	Гольцов Ю.А.						
Зав. каф.	Руданов В.Г.						
					Лит.	Лист	Листов
						5	2
					БГТУ им. В.Г. Шухова МР-41		

Компьютерное моделирование является одним из эффективных методов изучения сложных систем. Компьютерные модели проще и удобнее исследовать, в тех случаях когда реальные эксперименты затруднены из-за финансовых или физических препятствий или могут дать непредсказуемый результат. Логичность и формализованность компьютерных моделей позволяет выявить основные факторы, определяющие свойства изучаемого объекта. [2]

Целью данной работы является разработка кинематической схемы трехстепенного манипулятора и построение трёхмерной модели его конструкции, моделирование компьютерной динамической модели манипулятора, выполнить анализ движения модели робота при различных условиях, определить некоторые характеристики робота на основе моделирования, проектирование отдельных деталей и изготовление их при помощи 3D печати, сборка робота, разработка системы управления манипулятором на основе Arduino [3].

Манипулятор по принципу действия напоминает человеческую руку. В нём присутствуют поворотные соединения, которые обеспечивают наклон в плечевом соединении и сгибание в локте, механический захват, который позволит роботу хватать и перемещать предметы в разных направлениях.

Отличительная черта данной конструкции – очень высокая гибкость, позволяющая роботу обходить многие препятствия. Кроме того, робот этого типа достаточно компактен, и обслуживаемая им зона гораздо больше, чем занимаемое роботом место.

Несмотря на очевидные преимущества конструкции, управлять таким роботом достаточно сложно. При перемещении каждого звена принцип минимального значения требуемого угла, и манипулятор движется не по прямой линии (как декартов, например), а выполняет довольно сложную траекторию, имитируя движения живой руки. В результате мысленное представление всех движений «руки» сильно затруднено, что создаёт трудности при программировании. [4]

					<b>Введение</b>	Лист
						6
Изм.	Лист	№ докум.	Подпись	Дата		

# 1. Подходы к моделированию и управлению манипуляторами

## 1.1. Типы манипуляторов

Первый вопрос, с которым сталкивается создатель манипулятора, - выбор его кинематической схемы, структуры его скелета. В процессе выполнения операций с объектами манипулирования в большинстве случаев манипуляторы имитируют движение рук человека. Поэтому структурная схема манипулятора должна обладать кинематическими характеристиками, аналогичными характеристикам руки человека.

Манипулятор должен иметь по крайней три степени подвижности, необходимые для перемещения инструмента в любую точку зоны обслуживания. Каждая степень подвижности манипуляционного робота управляется индивидуальным приводом, в результате чего исполнительный орган получает направленное вполне определенное движение. В современных манипуляторах используют электромеханические, гидравлические, пневматические или комбинированные приводы.

Существует большое количество схем манипуляторов, различным образом реализующих региональные движения, но наиболее распространенными в промышленности являются пять следующих схем с одноподвижными кинематическими парами:

**Декартовый тип.** Манипулятор (рис. 1.1), функционирующий в декартовой (прямоугольной) системе координат, прост в управлении и отличается высокой точностью действий. Схват манипулятора поступательно перемещается вдоль трех основных осей:  $x$ ,  $y$  и  $z$  (т. е. слева направо, вперед-назад и вверх-вниз).

					<b>Разработка трехстепенного лабораторного манипулятора с механическим захватом</b>		
Изм.	Лист	№ докум.	Подпись	Дата			
Разработал	Ерёмин Д.О.				<b>Подходы к моделированию и управлению манипуляторами</b>	Лит.	Лист
Проверил	Юдин Д.А.						
Руководит.	Юдин Д.А.					7	12
Норм. Контр.	Гольцов Ю.А.					БГТУ им. В.Г. Шухова МР-41	
Зав. каф.	Руданов В.Г.						



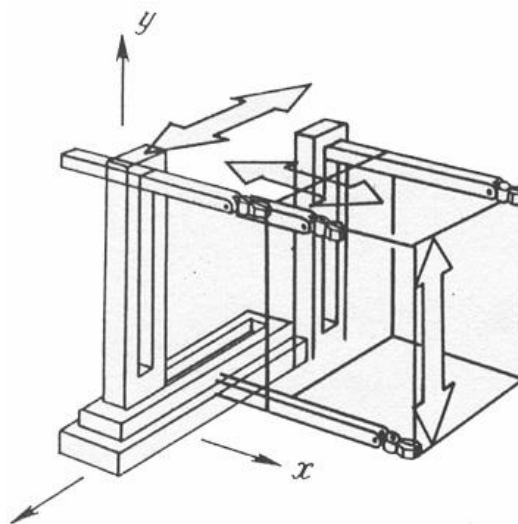


Рис. 1.1. Манипулятор декартового типа

**Цилиндрический тип.** Манипулятор (рис. 1.2), работающий в цилиндрической системе координат. Его схват может выдвигаться и втягиваться, а также перемещаться вверх и вниз вдоль стойки. Кроме того, весь узел манипулятора может поворачиваться вокруг оси основания, но не на полный оборот, что позволяет ему выполнять операции в окружающей цилиндрической зоне.

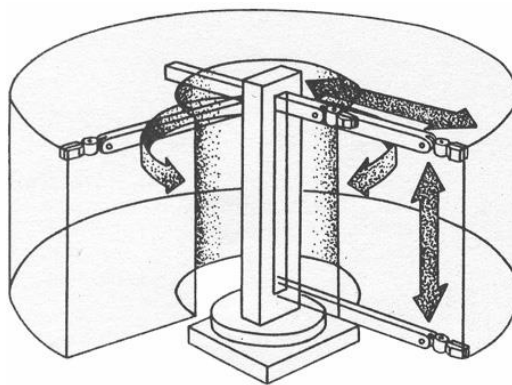


Рис. 1.2. Манипулятор цилиндрического типа

**Сферический тип.** Манипулятор (рис. 1.3), действующий в сферической (или полярной) системе координат. Его схват может выдвигаться и втягиваться. Вертикальные перемещения манипулятора достигаются путем поворота его в вертикальной плоскости в «плечевом» суставе. Весь узел манипулятора может также поворачиваться вокруг оси основания. Зона действия подобного манипулятора представляет усеченную

сферу. Первые модели промышленных роботов были сконструированы именно по этому принципу.

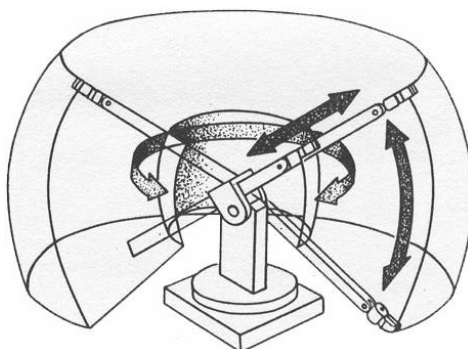


Рис. 1.3. Манипулятор сферического типа

**Ангулярный тип.** Шарнирный манипулятор (рис.1.4), действующий в ангулярной системе координат, не имеет поступательных кинематических пар, а имеет только вращательные кинематические пары. Манипулятор такого типа очень напоминает руку человека, поскольку имеет «плечевое» и «локтевое» сочленения, а также «запястье». Его зона обслуживания значительно больше, чем у роботов других типов. Он способен обходить препятствия гораздо более разнообразными путями и даже складываться, но вместе с тем он исключительно сложен в управлении.

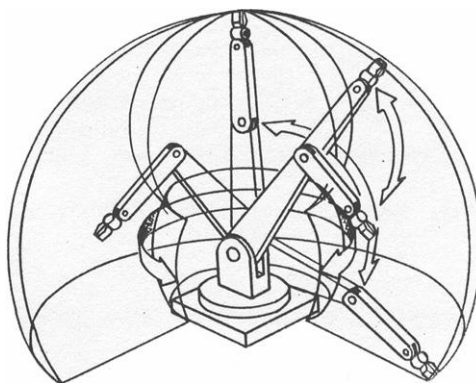


Рис. 1.4. Манипулятор ангулярного типа

**Тип SCARA.** Своеобразную схему имеет манипулятор системы SCARA (рис. 1.5), представляющий собой вариант манипулятора с цилиндрической системой координат. Все кинематические пары этого манипулятора располагаются в горизонтальной плоскости, благодаря чему механизм способен разворачиваться подобно складной ширме. Его зона обслуживания имеет цилиндрическую форму.

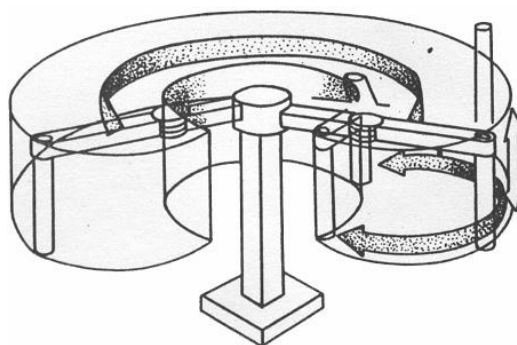


Рис. 1.5. Манипулятор SCARA типа

Перспективными представляются роботы еще двух типов. Первый из них, «**Spine**» (рис. 1.6), спроектирован специалистами фирмы «Спайн роботикс». В нем используется длинный хоботоподобный манипулятор, состоящий из множества чечевицеобразных дисков, которые соединены между собой двумя парами тросов, обеспечивающих натяжение. Тросы соединены с поршнями гидравлических цилиндров, которые, создавая натяжение, вызывают перемещение манипулятора. Специальные датчики передают на систему управления информацию о положении манипулятора и его кисти. Такой робот отличается чрезвычайно большой гибкостью, значительным радиусом действия и высокой маневренностью.

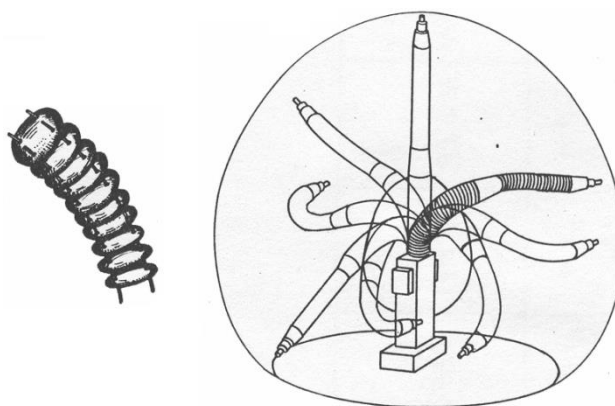


Рис. 1.6. Манипулятор Spine типа

Другой робот **маятникового типа** (рис. 1.7), IR B1000, разработан специалистами фирмы ASEA; его манипулятор подвешен подобно маятнику с двойным карданным подвесом и может перемещаться по направляющим относительно продольной и поперечной осей. По утверждению специалистов фирмы ASEA, это устройство движется в 1,5 раза быстрее, чем

традиционные манипуляторы, что обеспечивает высокую производительность. [5]

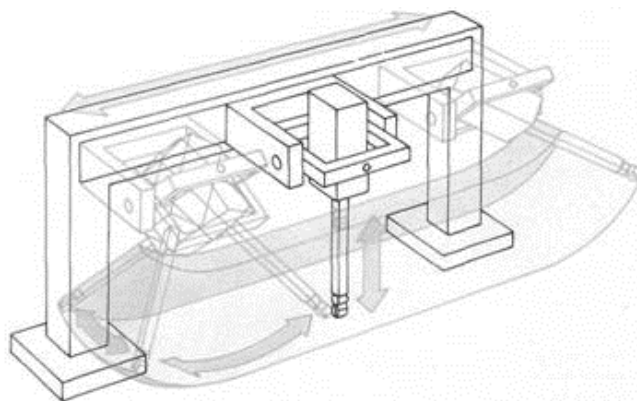


Рис. 1.6. Манипулятор маятникового типа

В данной работе будет разрабатываться манипулятор ангулярного типа т.к. вращательные кинематические пары наиболее просты в конструкции и легко реализуемы, также такая конструкция имеет довольно обширную рабочую зону и своей гибкостью позволяет обходить некоторые препятствия. Но реализация системы управления для такой конструкции будет вызывать некоторые трудности т.к. для передвижения рабочего органа манипулятора по декартовым координатам потребуется управлять углами поворотов звеньев робота, причём отклонения поворота каждого звена от нужного значения вносят суммирующее воздействие на погрешность положения рабочего органа.

## 1.2. Классификация манипуляторов

Манипуляторы составляют 85-90% всех роботов в мире.

**Классификация роботов по назначению** в промышленности:

- Керамическая промышленность: выдавливание керамического сырья, загрузка вальцовых (крокетных) машин, извлечение сформованных изделий, складирование, покрытие глазурью путем окунания, нанесение глазури пульверизатором, шлифовка изделия после обжига, загрузка и разгрузка печей;
- Стекольная промышленность: загрузка и разгрузка машин;

- Швейная промышленность: загрузка швейных машин;
- Деревообрабатывающая промышленность: покрытие лаком, сборка изделий, забивка гвоздей, закручивание винтов;
- Производство и обработка кожи: загрузка машин;
- Резинообрабатывающая промышленность: распознавание образов, манипулирование шинами;
- Асбестообрабатывающая промышленность: разрезка, обточка, шлифовка, штукатурка;
- Обработка пластиков: загрузка сырья, разгрузка машин;
- Мясообрабатывающая промышленность: рубка мяса.

#### **По степени универсальности:**

- универсальные (для выполнения разных операций совместно с различными видами оборудования);
- специализированные (выполняет одну операцию из нескольких возможных с различным оборудованием);
- специальные (выполняет конкретную операцию с одним типом оборудования).

#### **По виду технологических операций:**

- осуществляющие основные технологические операции;
- выполняющие вспомогательные технологические операции по обслуживанию технологического оборудования (средства автоматизации).

#### **По показателям, определяющим их конструкцию:**

- тип приводов робота (электрический, гидравлический, пневматический);
- грузоподъемность (сверхлегкие — до 1 кг; легкие — от 1 до 10 кг; средние 10,200 кг; тяжелые — 200,1000 кг; сверхтяжелые — свыше 1000 кг);
- количество манипуляторов (от 1 до 4 рук);

					<i>Подходы к моделированию и управлению манипуляторами</i>	Лист
Изм.	Лист	№ докум.	Подпись	Дата		12

- тип и параметры рабочей зоны манипуляторов (прямоугольная, цилиндрическая, сферическая, угловая (ангулярная) и различные их комбинации).);
- подвижность робота определяется наличием или отсутствием у него устройства передвижения (подвижный или стационарный).
- по способу размещения стационарные и подвижные роботы бывают напольными, подвесными (перемещаются по монорельсу), встраиваемые в другое оборудование (в станок или др.);
- по исполнению робота - зависит от назначения (нормальное, пылезащитное, теплозащитное, влагозащитное, взрывобезопасное и т.п.).

#### **По способу управления:**

- с программным управлением;
- с адаптивным управлением;
- с интеллектуальным управлением.

#### **По быстродействию движений:**

- малое быстродействие — до 0,5 м/с;
- среднее — линейные скорости от 0,5 до 1 м/с (~80 % роботов);
- высокое — свыше 1 м/с (~20 % роботов).

#### **По точности движений:**

- малая точность — при линейной погрешности от 1 мм и выше;
- средняя — от 0,1 до 1 мм (больше всего роботов);
- высокая — менее 0,1 мм.

#### **Параметры, определяющие технический уровень роботов:**

- надёжность;
- число одновременно работающих степеней подвижности;
- время программирования;
- удельная грузоподъёмность (отнесённая к массе робота);

- выходная мощность манипулятора (произведение грузоподъёмности на скорость перемещения), отнесённая к мощностям его приводов;
- относительные оценки габаритных параметров и т. п.

Эти параметры служат критериями качества, предназначенные для их оптимизации при проектировании и сравнительной оценки роботов. [6]

Разрабатываемый в данной работе трёхступенной манипулятор будет иметь следующие параметры классификации:

- степень универсальности - специальный, предназначен для переноски небольших лёгких предметов;
- тип приводов – электрический;
- грузоподъёмность - сверхлёгкий (до 1 кг);
- тип рабочей зоны – ангулярная;
- подвижность робота – стационарный;
- способ размещения – напольный;
- способ управления – программный и ручной;
- быстродействие движений – малое быстродействие (до 0,5 м/с);
- точность передвижений – малая точность (погрешность более 1 мм).

### 1.3. Системы управления манипулятором

Системами управления оснащены все манипуляторы, у которых перемещение подвижных звеньев осуществляются с помощью различных немеханических приводов.

Система управления современного манипулятора состоит из нескольких подсистем, выполняющих определенные информационные, управляющие, защитно-предупредительные, ограничительные и другие функции.

Различают три основных вида управления ПР: цикловое, позиционное, контурное. Цикловое управление программирует последовательность выполнения движений и условия начала и окончания движений; положение,

					<i>Подходы к моделированию и управлению манипуляторами</i>	<i>Лист</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		14

до которой идет движение, задаются на самом манипуляторе, а не в программе; скорость перемещения определяется характеристиками привода и также не задается в программе. При позиционном управлении команды подаются так, что перемещение рабочего органа происходит от точки к точке, причем положения точек задаются программой. Скорость перемещения между точками не контролируется и не реализуется. При контурном управлении движение рабочего органа происходит по заданной траектории с задаваемой скоростью. В программе задаются сами траектории и режимы движения. Контурное управление используется значительно в технологических работах.

Устройство управления и другие блоки системы управления при цикловом, позиционном и контурном управлениях могут быть реализованы на одинаковых или разных принципах и элементных базах. В особый вид обычно выделяется адаптивное управление, при котором осуществляется автоматическое изменение управляющих программ. В частности адаптация или приспособленность системы управления может заключаться в том, что устройства системы управления с помощью специальных датчиков определяют конфигурацию объекта и его положения.

Все основные ПР образуют систему управления. Объектом управления является исполнительное устройство (манипулятор плюс устройство передвижения, если оно есть). В исполнительное устройство также входят приводы. Все остальное оборудование робота предназначено для формирования и выдачи управляющих воздействий исполнительному устройству. Таким образом, устройство управления получает сигналы (от датчиков) и выдает сигналы (на приводы манипулятора).

Для пульта ручного управления основными являются связи с устройством управления. С пульта ручного управления могут осуществляться ввод программ, настройка. На пульт управления поступают сигналы о выполнении различных движений, а также о возможных нарушениях режимов работы и об отказах. Следует иметь в виду, что в

					<i>Подходы к моделированию и управлению манипуляторами</i>	Лист
Изм.	Лист	№ докум.	Подпись	Дата		15



устройство управления обычно поступают сигналы от внешних (по отношению к роботу) датчиков и систем (например, от систем управления обслуживаемым оборудованием). Устройство управления роботом также может быть связано с ЭВМ, координирующей работу нескольких единиц оборудования, например всего оборудования технологического участка или линии. В этих случаях эта ЭВМ как бы находится на более высоком этаже, на следующем уровне управления. Такая многоуровневая система управления типична для современных гибких производственных систем. [7]

В данной работе системой управления манипулятором будет микроконтроллер Arduino, который будет получать сигналы управления с компьютера через COM порт. Все вычисления будут выполняться на компьютере, микроконтроллер будет отвечать за повороты сервоприводов робота. Через программу в компьютере можно будет управлять манипулятором вручную, т.е. поворачивать звенья манипулятора или передвигать рабочую точку робота по декартовым осям координат. Также будет режим программирования, где рабочая точка робота будет перемещаться от положения к положению, т.е. реализуется позиционная система управления.

#### 1.4. Способы моделирования манипуляторов

Моделирование играет ключевую роль в области роботостроения, потому что оно позволяет проводить эксперименты, которые в ином случае были бы дорогими и/или требовали больших затрат времени [8].

С помощью компьютерной динамической модели робота-манипулятора можно определить диапазон углов поворотов звеньев и рабочую зону робота с учётом ограничений его конструкции, а также определить различные возникающие моменты и силы в узлах конструкции. Разработка модели робота играет ключевую роль при его проектировании.

Существуют различные подходы к разработке динамической модели

					<p><i>Подходы к моделированию и управлению манипуляторами</i></p>	Лист
						16
Изм.	Лист	№ докум.	Подпись	Дата		

робота-манипулятора. Есть математические и компьютерные динамические модели.

Математическая динамическая модель манипулятора может быть построена на основе известных законов ньютоновской или лагранжевой механики. Результатом применения этих законов являются уравнения, связывающие действующие в сочленениях силы и моменты с кинематическими характеристиками и параметрами движения звеньев. Таким образом, уравнения динамики движения реального манипулятора могут быть получены традиционными методами Лагранжа-Эйлера или Ньютона-Эйлера.

Вывод уравнений динамики движения манипулятора методом Лагранжа-Эйлера отличается простотой и единством подхода и основан на следующем:

1. На описании взаимного пространственного расположения систем координат  $i$ -го и  $(i-1)$ -го звеньев с помощью матрицы преобразования однородных координат.

2. На использовании уравнения Лагранжа-Эйлера

С целью получения более эффективных с вычислительной точки зрения алгоритмов расчета обобщенных сил и моментов используются уравнения Ньютона-Эйлера. Вывод уравнений Ньютона-Эйлера прост по содержанию, но весьма трудоёмок. Результатом является система прямых и обратных рекуррентных уравнений, последовательно применяемых к звеньям манипулятора.

Уравнения Лагранжа-Эйлера обладают низкой вычислительной эффективностью, это обусловлено тем, что для описания кинематической цепи используются матрицы преобразования однородных координат. Уравнения Ньютона-Эйлера обладают большей вычислительной эффективностью, что связано с их векториальным представлением и их рекуррентной природой. [9]

Для построения компьютерной динамической модели робота используются программы предназначенные для моделирования динамики и

					<i>Подходы к моделированию и управлению манипуляторами</i>	Лист
						17
Изм.	Лист	№ докум.	Подпись	Дата		

кинематики пространственных механических систем, например:

MSC.ADAMS [10]; UM [11]; Unigraphics NX [12]; CATIA [13]; SolidWorks [14].

Создание модели подразумевает описание всех ее характеристик: геометрических размеров, физических свойств, способов соединения подвижных и неподвижных частей, задание действующих сил и моментов, начального положения элементов модели и их скоростей. Этап тестирования модели включает в себя моделирование поведения частей модели под действием приложенных сил и заданных движений и выявление критических параметров, наиболее сильно влияющих на эффективность работы модели в целом.

В данной работе динамическая модель робота-манипулятора будет строиться в программной среде MSC.ADAMS, предназначенной для виртуального моделирования сложных механизмов. Широкие возможности программного пакета, высокая надёжность и малая трудоёмкость его использования позволяют исследовать множество вариантов механизмов, моделируя на компьютере реальные условия их работы, сравнивать и выбирать лучший вариант [15]. Также для сравнения с компьютерной моделью будет выведена математическая динамическая модель с использованием уравнений Лагранжа-Эйлера.

					<i>Подходы к моделированию и управлению манипуляторами</i>	<i>Лист</i>
						18
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		

## 2. Требования к разрабатываемому устройству

1. Разрабатываемое устройство должно обладать следующими свойствами:

- Основание, поворачивающееся вокруг своей оси, способное крепиться к горизонтальной плоской поверхности;
- Два звена, имеющих длину по 20 см с шарнирными соединениями;
- Механический хват, способный схватить лёгкие предметы диаметром до 7 см.

2. Построить кинематическую схему робота.

3. Разработать трёхмерную модель конструкции трехступенного манипулятора с механическим захватом.

4. В трёхмерной модели определить диапазоны углов поворотов звеньев с учётом ограничений конструкции робота.

5. Построить компьютерную динамическую модель робота, с помощью которой определить:

- Рабочую зону манипулятора;
- Скорости, ускорения, силы и моменты, возникающие в шарнирах звеньев манипулятора при различных условиях;
- Максимальную грузоподъёмность робота.

6. Изготовить детали звеньев и захвата манипулятора при помощи 3D печати.

## 7. Собрать манипулятор.

8. Разработать систему управления манипулятором на основе Arduino.

9. Разработать компьютерную программу для управления манипулятором, которая позволит:

- Вручную управлять поворотами каждого звена по отдельности а также схватом;

					Разработка трехступенного лабораторного манипулятора с механическим захватом											
Изм.	Лист	№ докум.	Подпись	Дата												
Разработал		Ерёмин Д.О.				Требования к разрабатываемому устройству				Лит.		Лист		Листов		
Проверил		Юдин Д.А.											19		2	
Руководит.		Юдин Д.А.								БГТУ им. В.Г. Шухова МР-41						
Норм. Контр.		Гольцов Ю.А.														
Зав. каф.		Руданов В.Г.														

- Управлять передвижением рабочей точки манипулятора вдоль осей X, Y, Z декартовой системы координат.
- Управлять скоростью движения манипулятора;
- Позволить программировать манипулятор для автоматического передвижения через конкретные точки.

10. Проанализировать работу конструкции манипулятора и программу управления манипулятором и сделать выводы.

					Требования к разрабатываемому устройству	Лист
						20
Изм.	Лист	№ докум.	Подпись	Дата		

### 3. Разработка конструкции манипулятора

#### 3.1. Построение структурной кинематической схемы манипулятора

Перед разработкой трёхмерной модели конструкции манипулятора необходимо составить его структурную кинематическую схему (Рис. 3.1). Для начала необходимо определить тип манипулятора. Было решено разработать манипулятор ангулярного типа т.к. вращательные кинематические пары наиболее просты в конструкции и легко реализуемы, также такая конструкция имеет довольно обширную рабочую зону и своей гибкостью позволяет обходить некоторые препятствия. Также по заданию манипулятор должен иметь всего 3 степени подвижности: поворот основания вокруг своей оси и повороты 1го и 2го звеньев. Чтобы схват захватывал объекты правильно, он должен быть направлен в определённой ориентации, для этого в кинематическую схему добавлен специальный механизм для удержания схвата в неизменной ориентации.

Кинематическая схема разрабатывалась на основе реального промышленного робота Fanuc M-410iB/450 [16] (Рис. 3.1), который имеет 4 степени подвижности и используется для переноски груза.

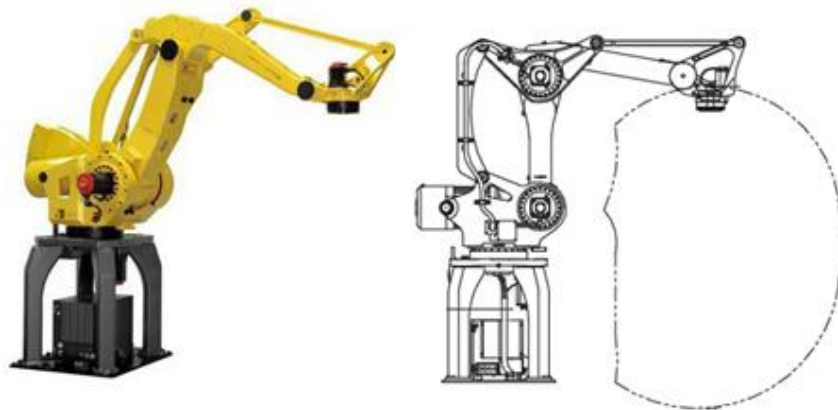


Рис. 3.1. Промышленный манипулятор Fanuc M-410iB/450 и его рабочая зона

					<b>Разработка трехстепенного лабораторного манипулятора с механическим захватом</b>		
Изм.	Лист	№ докум.	Подпись	Дата			
Разработал	Ерёмин Д.О.				<b>Разработка конструкции манипулятора</b>	Лит.	Лист
Проверил	Юдин Д.А.						21
Руководит.	Юдин Д.А.						20
Норм. Контр.	Гольцов Ю.А.					БГТУ им. В.Г. Шухова МР-41	
Зав. каф.	Руданов В.Г.						

Разработанная кинематическая схема трёхстепенного манипулятора указана на рисунке 3.1. На схеме указаны 1ое и 2ое звенья, которые поворачиваются на углы  $q_1$  и  $q_2$ , основание, вокруг которого манипулятор поворачивается на угол  $q_0$ , и схват, а также проставлены длины всех частей манипулятора.

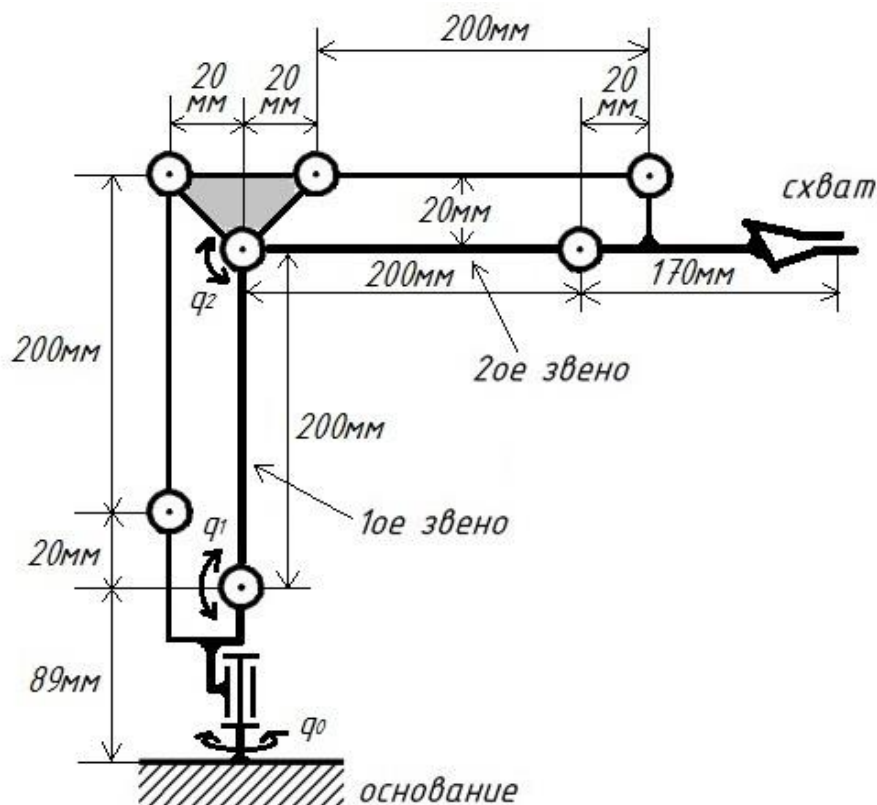


Рис. 3.1. кинематическая схема манипулятора

В этой кинематической схеме используются некоторые дополнительные звенья и элементы, которые образуют параллельный механизм, который позволяет схвату манипулятора оставаться всегда параллельно основанию независимо от того, на какие углы  $q_1$  и  $q_2$  повернутся звенья манипулятора. На рисунке 3.2(а,б,в,г) показан принцип работы параллельного механизма, на котором видно что при изменении углов поворота звеньев, схват робота сохраняет параллельное положение относительно основания. Данное свойство удобно для захвата, перемещения и установления объекта в новое положение, при этом сохраниться его вертикальная ориентация. Так же отпадает необходимость ставить 3е управляемое звено. Несмотря на множество шарнирных соединений в данной

схеме для ориентации манипулятора используется всего 3 сервопривода, которые расположены в точках углов поворотов звеньев  $q_0$ ,  $q_1$ ,  $q_2$ , ещё один сервопривод нужен для сжатия/разжатия схвата.

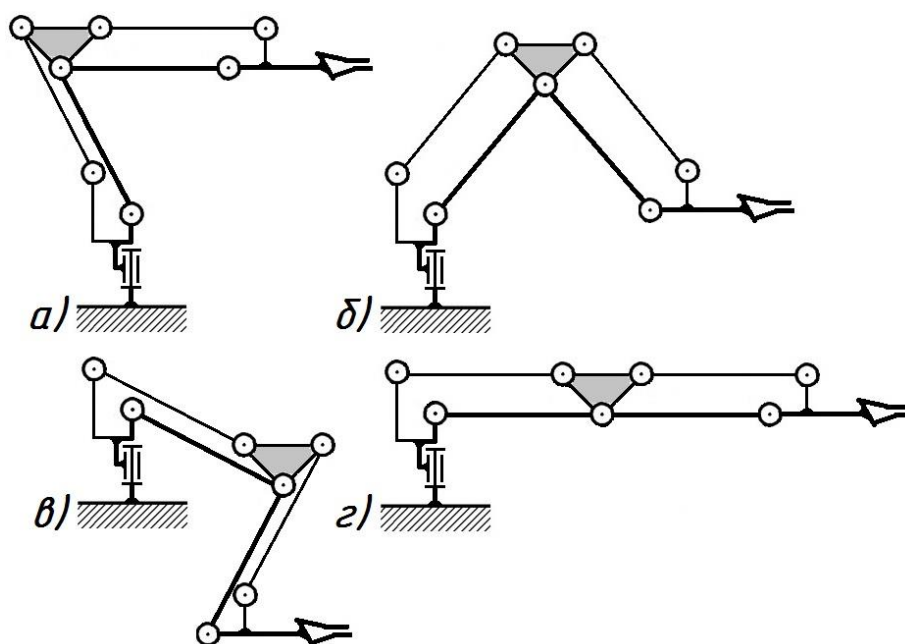


Рис. 3.2. принцип работы параллельного механизма

### 3.2. Решение прямой и обратной задачи кинематики

Прямая задача – это определение положений звеньев манипулятора при заданных углах поворота соединений и размеров звеньев.

Обратная задача – это определение углов поворота при заданных координатах положения рабочей точки манипулятора

Пред решением задач составим структурную схему (Рис. 3.3) с нужными для решения обозначениями.

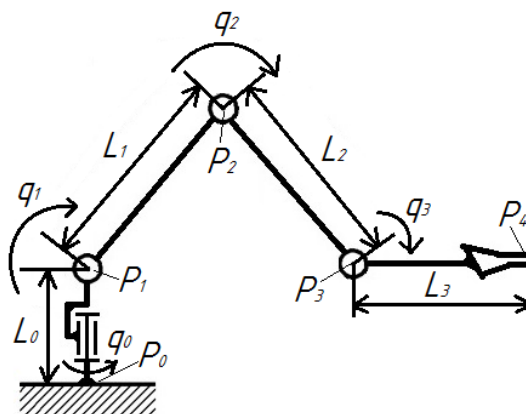


Рис 3.3. Структурная схема с обозначениями

					Разработка конструкции манипулятора	Лист
Изм.	Лист	№ докум.	Подпись	Дата		23



В данной схеме:

$q_0, q_1, q_2$  – углы поворота;

$L_0, L_1, L_2, L_3$  – длины звеньев;

$P_0, P_1, P_2, P_3, P_4$  – точки положения звеньев

Все вычисления для прямой и обратной задач проводились в среде Mathcad.

### Решение прямой задачи:

Задаём углы и расстояния:

$$q_0 := 65 \frac{\pi}{180} \quad q_1 := 15 \frac{\pi}{180} \quad q_2 := 35 \frac{\pi}{180} \quad q_2 := -(q_1 + q_2)$$

Задаём длины звеньев, мм:

$$L_0 := 90 \quad L_1 := 200 \quad L_2 := 200 \quad L_3 := 170$$

Матрицы переноса:

$$AL_0 := \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & L_0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad AL_1 := \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & L_1 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad AL_2 := \begin{pmatrix} 1 & 0 & 0 & L_2 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad AL_3 := \begin{pmatrix} 1 & 0 & 0 & L_3 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Матрицы поворота:

$$Aq_0 := \begin{pmatrix} \cos(q_0) & -\sin(q_0) & 0 & 0 \\ \sin(q_0) & \cos(q_0) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad Aq_1 := \begin{pmatrix} \cos(q_1) & 0 & \sin(q_1) & 0 \\ 0 & 1 & 0 & 0 \\ -\sin(q_1) & 0 & \cos(q_1) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$Aq_2 := \begin{pmatrix} \cos(q_2) & 0 & \sin(q_2) & 0 \\ 0 & 1 & 0 & 0 \\ -\sin(q_2) & 0 & \cos(q_2) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad AL_1 := \begin{pmatrix} \cos(q_3) & 0 & \sin(q_3) & 0 \\ 0 & 1 & 0 & 0 \\ -\sin(q_3) & 0 & \cos(q_3) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Вычисление матриц положения точек:

$$A_0 := \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$A_1 := A_0 \cdot Aq_0 \cdot AL_0 \quad A_2 := A_1 \cdot Aq_1 \cdot AL_1 \quad A_3 := A_2 \cdot Aq_2 \cdot AL_2 \quad A_4 := A_3 \cdot Aq_3 \cdot AL_3$$

Координаты точек манипулятора:

$$X := \begin{pmatrix} A0_{0,3} \\ A1_{0,3} \\ A2_{0,3} \\ A3_{0,3} \\ A4_{0,3} \end{pmatrix} \quad Y := \begin{pmatrix} A0_{1,3} \\ A1_{1,3} \\ A2_{1,3} \\ A3_{1,3} \\ A4_{1,3} \end{pmatrix} \quad Z := \begin{pmatrix} A0_{2,3} \\ A1_{2,3} \\ A2_{2,3} \\ A3_{2,3} \\ A4_{2,3} \end{pmatrix}$$

На рисунке 3.4 изображено решение **прямой задачи**

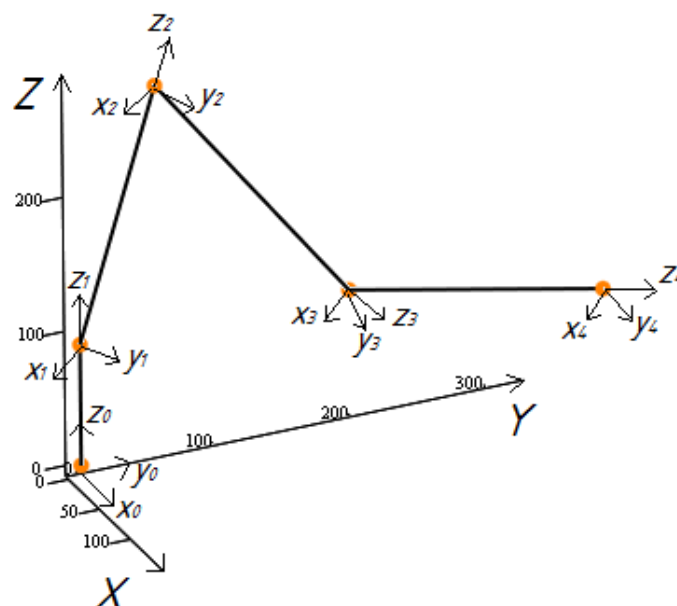


Рис 3.4. Решение прямой задачи

Для наглядного представления решения **обратной задачи** показана геометрическая схема на рисунке 3.5.

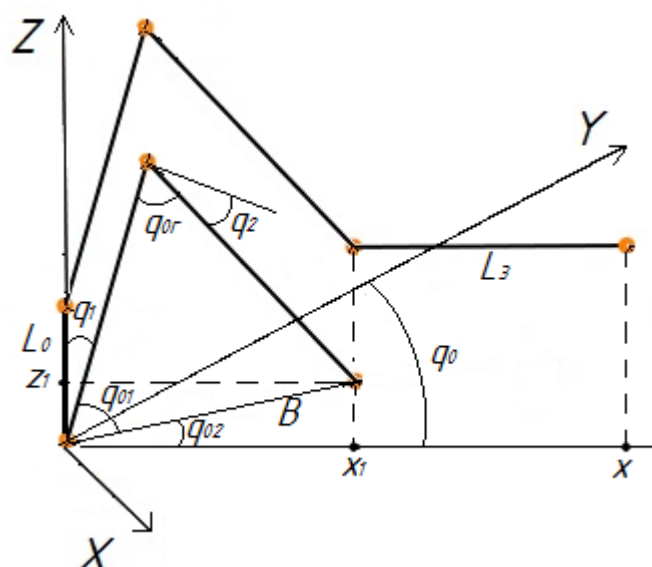


Рис 3.5. Решение обратной задачи

### Решение обратной задачи:

Задаём длины звеньев, мм:

$$L0 := 90 \quad L1 := 200 \quad L2 := 200 \quad L3 := 170$$

Задаём координаты положения точки схвата:

$$x := A4_{0,3} = 148 \quad y := A4_{1,3} = 317 \quad z := A4_{2,3} = 130$$

Решение задачи:

Находим проекцию точки  $P_4$  на плоскость OXY:

$$x := \sqrt{x^2 + y^2}$$

Находим проекцию точки  $P_3$  на плоскость OXY:

$$x1 := x - L3$$

Смещаем вниз координату  $z$  точки  $P_3$  на длину  $L_0$ :

$$z1 := z - L0$$

Находим расстояние между началом координат и точкой  $P_3$ :

$$B := \sqrt{x1^2 + z1^2}$$

Находим угол  $q_0$ :

$$Q0 := a \sin\left(\frac{y}{x}\right) = 1.134$$

1ое и 2ое звенья и отрезок  $B$  образуют треугольник, т.к. мы знаем их длины, то можем определить углы этого треугольника по формулам:

Найдём угол  $q_{01}$ :

$$q01 := a \cos\left(\frac{L1^2 + B^2 - L2^2}{2 \cdot B \cdot L2}\right) = 1.091$$

Найдём угол  $q_{0r}$ :

$$q0r := a \cos\left(\frac{L1^2 + L2^2 - B^2}{2 \cdot L1 \cdot L2}\right) = 0.96$$

Найдём угол  $q_{02}$ :

$$q02 := a \sin\left(\frac{z1}{B}\right) = 0.218$$

Найдя углы треугольника можно определить углы поворотов звеньев:

					Разработка конструкции манипулятора	Лист
Изм.	Лист	№ докум.	Подпись	Дата		26

$$Q1 := \frac{\pi}{2} - q01 - q02 \quad Q2 := \frac{\pi}{2} - q0r$$

Найдём значения углов в градусах:

$$q0 := Q0 \frac{180}{\pi} = 65 \quad q1 := Q1 \frac{180}{\pi} = 15 \quad q2 := Q2 \frac{180}{\pi} = 35$$

Решение показывает, что подставив значения координат из решения прямой задачи в обратную задачу и решив её, получим значения углов такие же как задавали в прямой задачи, что подтверждает правильность решения этих задач.

### 3.3. Выбор сервоприводов

Поворот звеньев манипулятора и сжатие/разжатие захвата будут осуществлять цифровые сервоприводы. При длине двух звеньев (по 20 см каждое) и с учетом размера захвата приблизительная максимальная длина манипулятора будет 57 см. Чтобы определить мощность, необходимую сервоприводу для поднятия манипулятора, нужно определить вес манипулятора. Для этого нужно учесть вес сервоприводов и деталей звеньев и захвата. Сервопривод весит в среднем 60 грамм, т.к. их 3, то общий их вес 180 грамм, остальные детали будут сделаны из пластика и металла, в целом вес робота должен быть около 500 грамм без основания. Допустим что масса робота примерно распределена равномерно вдоль всей его длины, следовательно, при закреплённом основании и расположении звеньев робота перпендикулярно силе тяжести, масса в рабочей точке робота будет 250 грамм (Рис. 3.6).

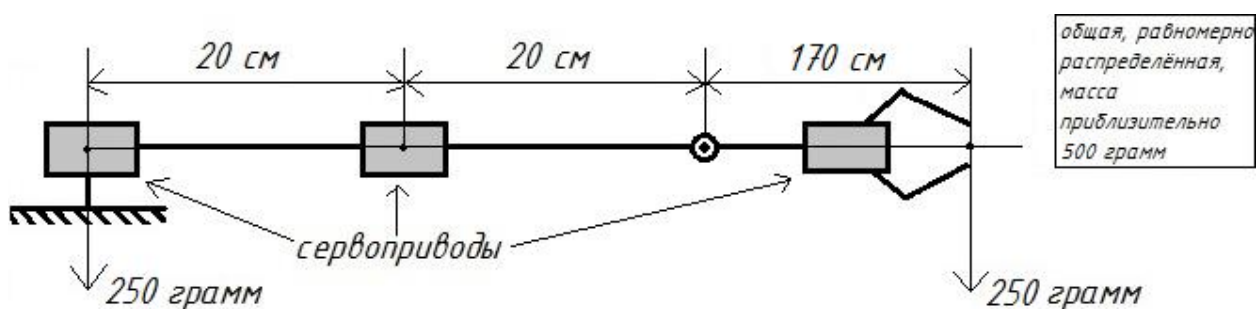


Рис. 3.6. Примерная оценка массы робота

					<i>Разработка конструкции манипулятора</i>	<i>Лист</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		<i>27</i>

При массе 250 грамм в рабочей точки и длине робота в 57 см, момент на валу сервопривода будет 14,25 кг\*см при условии, что захват робота будет без груза.

Доступные в данный момент сервоприводы с достаточно большим моментом были сервоприводы **DF15RMG** (Рис. 3.7) для работы 1го звена манипулятора и RDS3115 (Рис. 3.8) для работы 2го звена, эти сервоприводы практически идентичны по характеристикам и одинаковые по размерам.



Рис. 3.7. Внешний вид DF15RMG с дополнительными кронштейнами

Спецификации для сервопривода DF15RMG:

- угол поворота: 170°
- Скорость поворота: 60°/0.16s
- Момент: 19.3кг·см при 7.4V
- Металлические шестерни
- Размеры: 40x40x20 мм
- Вес: 65 г
- Рабочее напряжение: 5-7.4v
- Максимальный ток: 3A при 8.5V
- PPM Voltage: 3V-5V
- Разрешение: 2мкс
- Частота: до 4 КГц

[17]



Рис. 3.8. Внешний вид RDS3115 с дополнительными кронштейнами

Спецификации для сервопривода RDS3115:

- угол поворота: 180°
- Скорость поворота: 60°/0.16s
- Момент: 17кг·см при 7.2V

					Разработка конструкции манипулятора	Лист
Изм.	Лист	№ докум.	Подпись	Дата		28

- Металлические шестерни
- Размеры: 40x40x20 мм
- Вес: 60 г
- Рабочее напряжение: 4.8-8.4v [18]

Для работы захвата и поворота основания манипулятора будут использованы сервоприводы SR430 (Рис. 3.8). Они развивают меньший момент чем DF15RMG, т.к. для работы захвата и поворота основания не требуется высокий момент.



Рис. 3.8. Внешний вид SR430

Спецификации для сервопривода SR430:

- Рабочее напряжение: 6,0 — 7,4 В
- Скорость: 0,18 сек/60° при 6 В
- Усилие на валу: 4,2 кг/см при 6 В
- Скорость: 0,16 сек/60° при 7,4 В
- Усилие на валу: 5,3 кг/см при 7,4 В
- Угол вращения: 180°
- Размеры: 42 x 39,5 x 20,5 мм
- Вес: 44 гр.
- Шестерни редуктора: пластиковые [19]

### 3.4. Проектирование деталей

#### 3.4.1. Выбор САПР

Для проектирования 3D моделей деталей робота была выбрана программа Google SketchUp, т.к. она является одной из простейших программ для 3D моделирования и является бесплатной, а также имеет интерфейс на русском языке. Программа оснащена упрощенным набором инструментов, с которым пользователю работать максимально удобно - это позволит всего лишь за несколько часов освоить работу в программе и начать

					Разработка конструкции манипулятора	Лист
Изм.	Лист	№ докум.	Подпись	Дата		29

проектировать самые разные 3D-модели, от простейших геометрических фигур до полномасштабных красивых моделей [20].

### 3.4.2. Проектирование 1-го и 2-го звеньев

Сервоприводы DF15RMG уже обладают готовыми креплениями (Рис. 3.9) к ним, поэтому звено нужно спроектировать таким образом, чтобы оно могло крепиться к ним. Значит необходимо в детали сделать такие же отверстия как и в кронштейнах, чтобы их можно было стянуть болтами. Диаметр отверстий 3мм, расстояние между ним 10мм. Деталь будет сделана из алюминиевой рейки т.к. металл выдержит более высокие нагрузки и он менее деформируется по сравнению с пластиком.

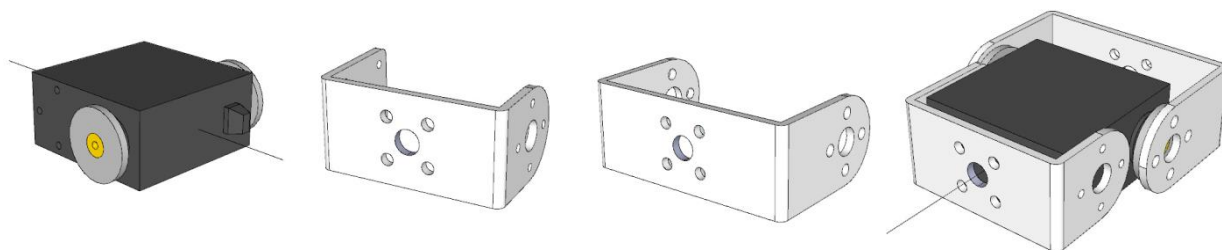


Рис. 3.9. 3D модель сервопривода DF15RMG и его кронштейнов

Стоит учесть, что сервоприводы, соединённые с рейкой образуют единое звено манипулятора, которое должно иметь длину 20см (Рис. 3.10), следовательно длину рейки нужно подобрать таким образом, чтобы выполнялось это условие.

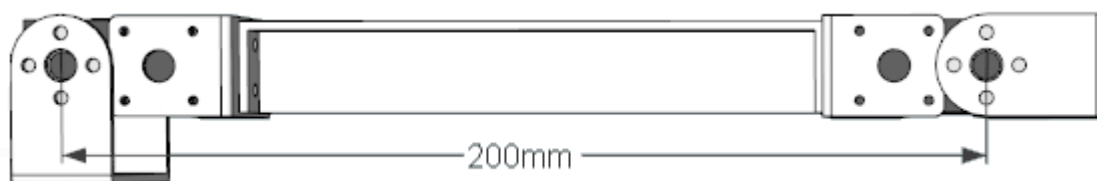


Рис. 3.10. Модель 1-го звена

В результате получится рейка длиной 12,8см (Рис. 3.11), которая образует с двумя сервоприводами 1ое звено длиной 20см.

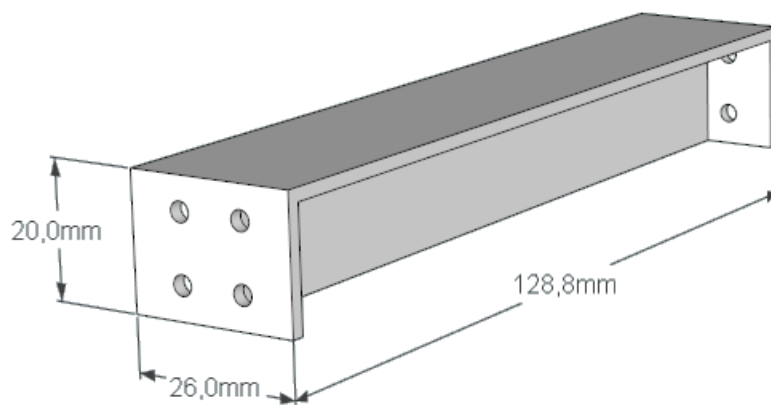


Рис. 3.11. Трёхмерный вид детали 1го звена с указанными размерами

Рейка 2го звена будет такая же как и для 1го звена, но только другой длины. 2ое звено образуют сервопривод, металлическая рейка и специальное крепление для схвата. Расстояние между осью вала сервопривода и осью вращения крепления для схвата составляет 20 см, что является длиной 2ого звена (Рис. 3.12).

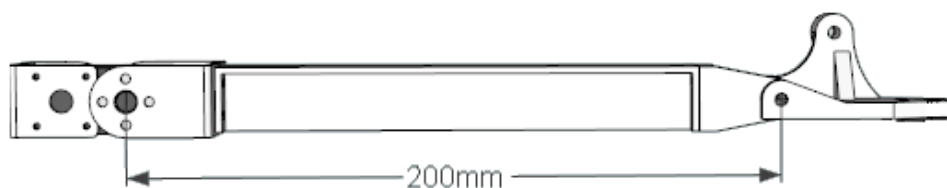


Рис. 3.12. Модель 2-го звена

В результате получится рейка 2го звена длиной 14,9см (Рис. 3.13).

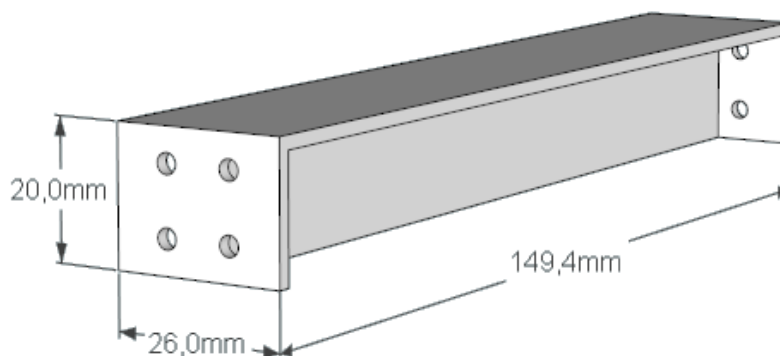


Рис. 3.13. Трёхмерный вид детали 2го звена с указанными размерами

### 3.4.3. Проектирование схвата

Модель схвата будет строится на основе готового чертежа (Рис. 3.14). Такой схват имеет 6 подвижных деталей, которые крепятся на основание (Рис. 3.15).



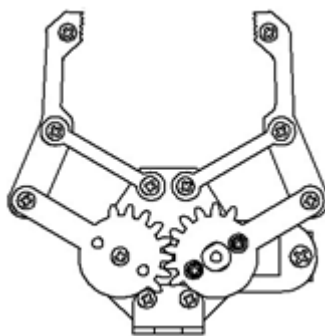


Рис. 3.14. Чертёж для построения модели схвата

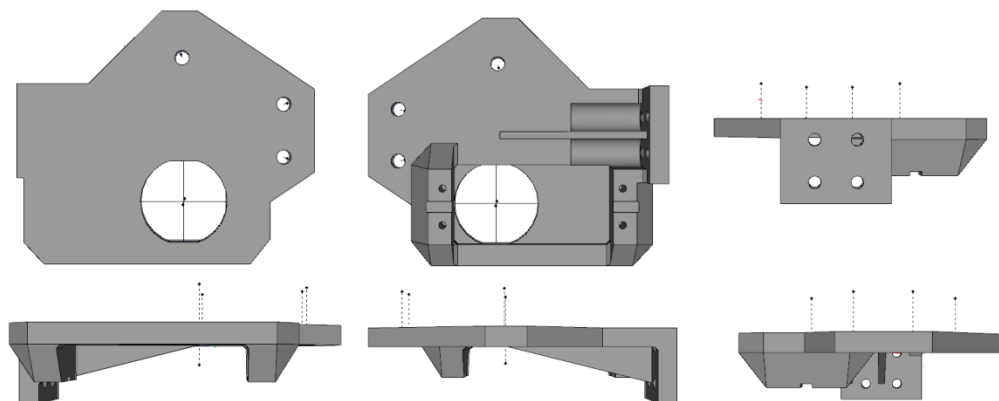


Рис. 3.15. Модель основания схвата с разных сторон

Расположение крепления для сервопривода было изменено, по сравнению с готовым чертежом, для того, чтобы схват мог шире раскрыться. Также на креплении добавлены специальные опоры для сервопривода SR430. В основании имеется специальное отверстие, чтобы вал сервопривода соединить с шестернёй, которая приводит механизм схвата в движение.

Расположение подвижных деталей и их форма остались как на чертеже. Все детали схвата (Рис. 3.17) должны крепиться между собой болтами диаметром 3мм и длиной желательно 15мм.

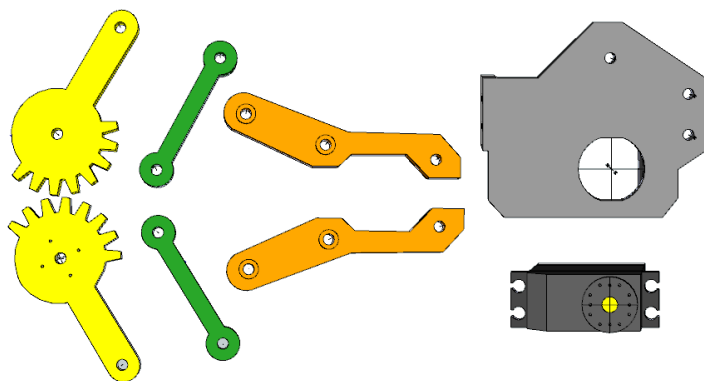


Рис. 3.17. Детали и основание схвата и сервопривод SR430

На рисунке 3.19 изображены 3D модели собранного схвата с разных ракурсов вместе с сервоприводом SR430.

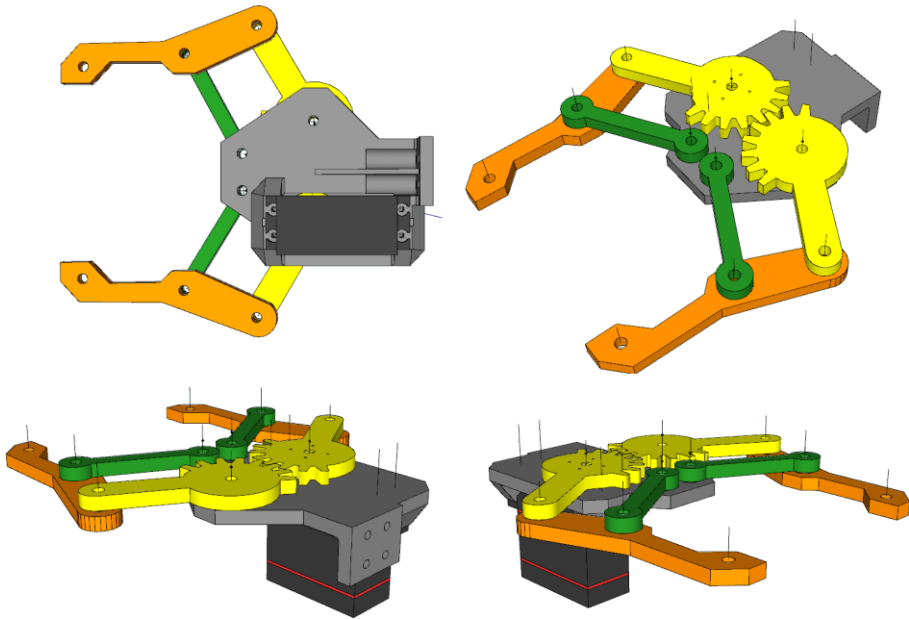


Рис. 3.19. Модель собранного схвата с разных ракурсов

Из рисунка 3.20 видно, что длина схвата меняется в зависимости от степени его раскрытия.

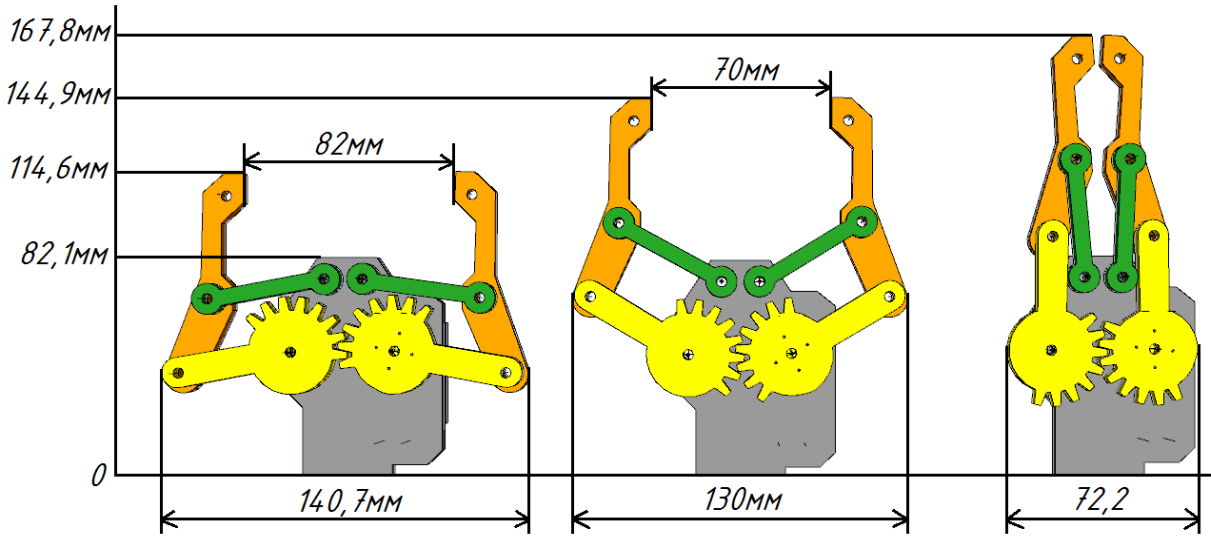


Рис. 3.20. Схват при различных положениях и проставленные размеры

### 3.4.4. Проектирование основания

На основание манипулятора будут идти самые большие нагрузки, также оно примет на себя весь его вес. К тому же к основанию предъявляется ряд требований: оно должно иметь возможность крепиться к плоской поверхности; поворачиваться вокруг своей оси; за вращение должен отвечать сервопривод SR430; основание должно быть достаточно широким, чтобы не опрокинуться при нагрузках и прочным, чтобы не сломаться от высоких нагрузок.

При проектировании основания было решено следующее: т.к. основание вращается оно должно быть цилиндрической формы, в нём должен поместиться сервопривод, оно должно иметь крепления для 1го звена манипулятора. Креплением 1го звена является кронштейн его сервопривода, сделаем в основании отверстия для крепления кронштейна (Рис. 3.21).

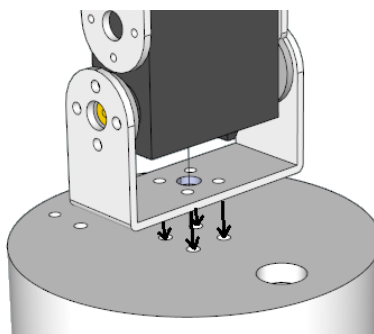


Рис. 3.21. Крепление 1го звена к основанию манипулятора

Внутри основания предусмотрены крепления для сервопривода SR430, они достаточно прочны чтобы выдержать нагрузки при вращении основания. Сервопривод крепится так, чтобы центр его вала находился по центру основания (Рис. 3.22).

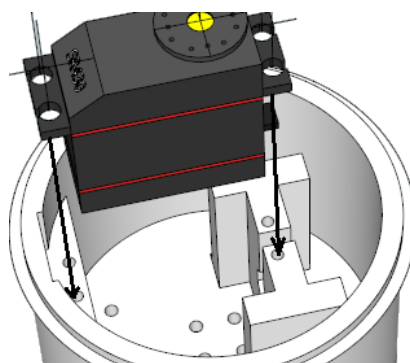


Рис. 3.22. Крепление сервопривода в основании манипулятора

					Разработка конструкции манипулятора	Лист
Изм.	Лист	№ докум.	Подпись	Дата		34

Чтобы основание вращалось, у него есть отдельная нижняя часть, относительно которой оно вращается (Рис. 3.23). Нижняя часть имеет по контуру 8 отверстий для крепления к плоской поверхности. Чтобы основание не вывалилось из своей нижней части, нижняя часть крепится болтами к валу сервопривода, который крепится к основанию (Рис. 3.24). При вращении вала сервопривода основание начнёт вращаться относительно нижней части.

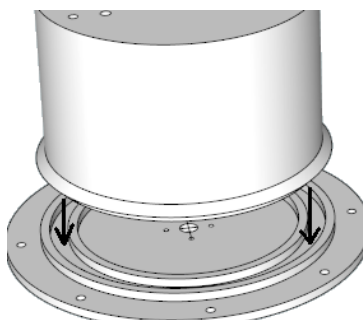


Рис. 3.23. Соединение основания с нижней частью

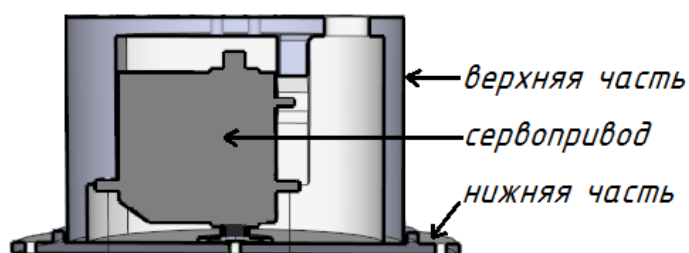


Рис. 3.24. Структура основания

На рисунках 3.25 и 3.26 указаны основные размеры деталей основания: верхней поворачивающейся части и нижней части, крепящейся к поверхности.

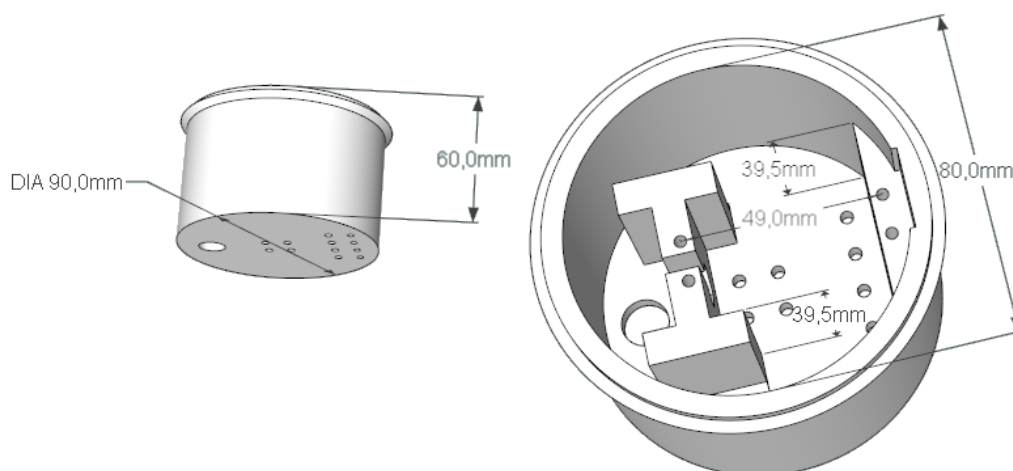


Рис. 3.25. Верхняя часть основания с указанными размерами

Изм.	Лист	№ докум.	Подпись	Дата

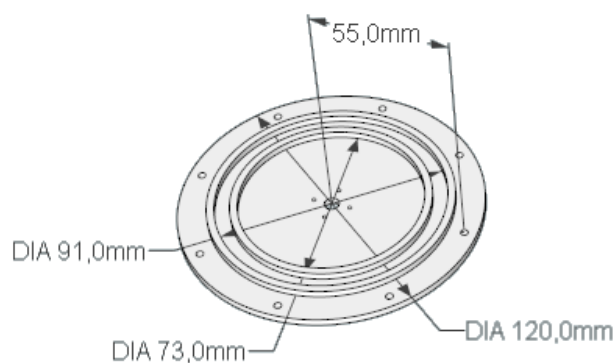


Рис. 3.26. Нижняя часть основания с указанными размерами

### 3.4.5. Проектирование всей конструкции манипулятора

Используя разработанные модели основания, звеньев и схвата манипулятора необходимо разработать модель всей конструкции манипулятора на основе его кинематической схемы (Рис. 3.1). Для этого нужно разработать модели деталей, которые должны образовать параллельный механизм согласно кинематической схеме. На рисунке 3.27 обозначены числами 1 – 5 те шарниры и звенья, которые нужно спроектировать для параллельного механизма.

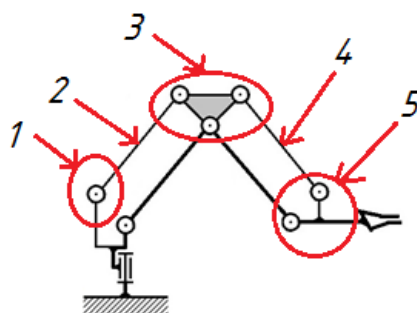


Рис. 3.27. Обозначение элементов параллельного механизма на кинематической схеме

Звенья под номерами 2 и 4 (Рис. 3.27) имеют одинаковую длину по 20 см. Следовательно детали этих звеньев можно спроектировать одинаковыми. На рисунке 3.28 изображена модель этих звеньев, они будут выполняться из алюминиевых пластин, чтобы иметь достаточную прочность при их длине.

Изм.	Лист	№ докум.	Подпись	Дата

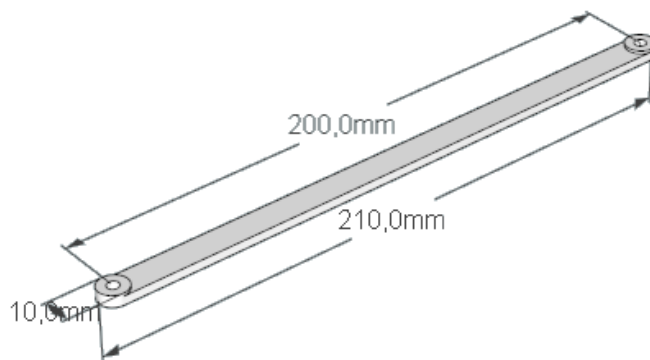


Рис. 3.28. Модель дополнительных звеньев для параллельного механизма

Модель детали, которая образует шарнирное соединение в точке 1 (Рис. 3.27) изображена на рисунке 3.29. Эта деталь будет изготавливаться при помощи 3D печати и будет крепиться к поворачивающемуся основанию манипулятора.

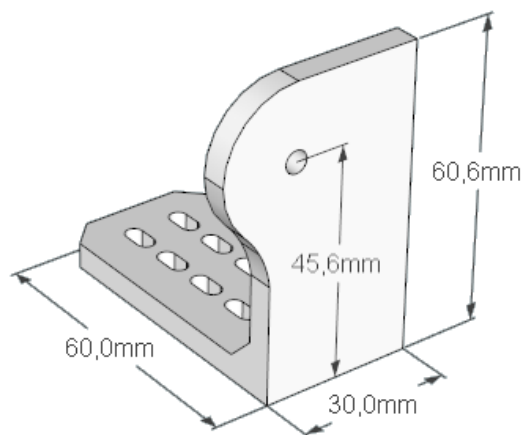


Рис. 3.29. Модель детали для шарнирного соединения

Модель детали, которая образует 3 шарнирных соединения в точке 3 (Рис. 3.27) изображена на рисунке 3.30. Эта деталь будет изготавливаться при помощи 3D печати и будет крепиться к валу сервопривода 2го звена манипулятора, а также будет соединять дополнительные звенья.

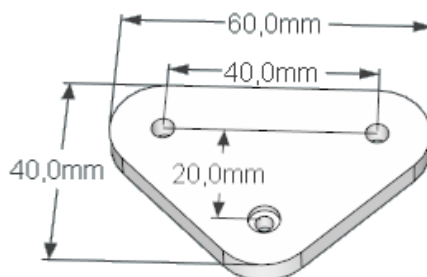


Рис. 3.30. Модель центральной детали параллельного механизма

Модель шарнирного механизма под номером 5 (Рис. 3.27) объединяет 2ое звено, схват и дополнительное звено манипулятора, она состоит из двух отдельных деталей, которые печатаются на 3D принтере и соединяются между собой длинным болтом. На рисунках 3.31 и 3.32 изображены эти детали по отдельности и объединённые вместе.

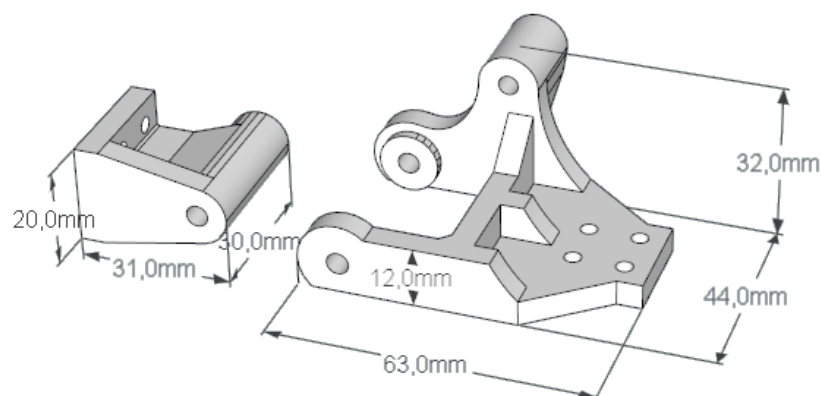


Рис. 3.31. Модель деталей шарнирного соединения манипулятора в области схвата

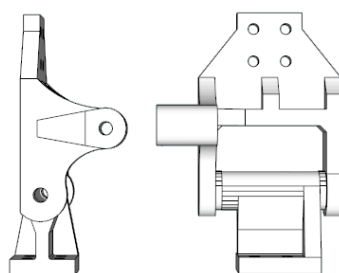


Рис. 3.32. Вид сбоку и сверху шарнирного соединения

Разработав модели всех деталей конструкции манипулятора, можно объединить их и получить модель полностью собранного манипулятора (Рис. 3.33). Чтобы лучше различить отдельные детали, они были окрашены в разные цвета.

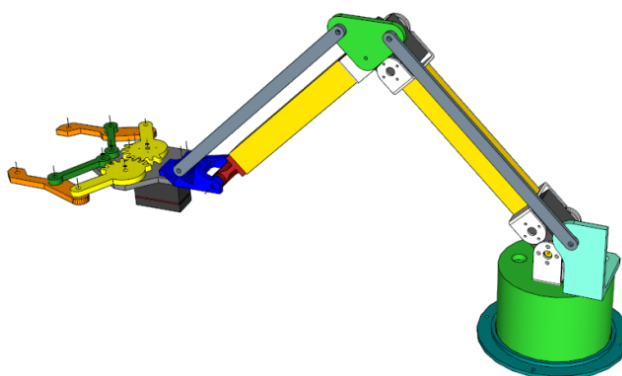


Рис. 3.33. Модель всей конструкции манипулятора

					Разработка конструкции манипулятора	Лист
Изм.	Лист	№ докум.	Подпись	Дата		38

На рисунке 3.34 показаны различные положения модели манипулятора и их кинематические схемы, тут же демонстрируется принцип работы параллельного механизма.

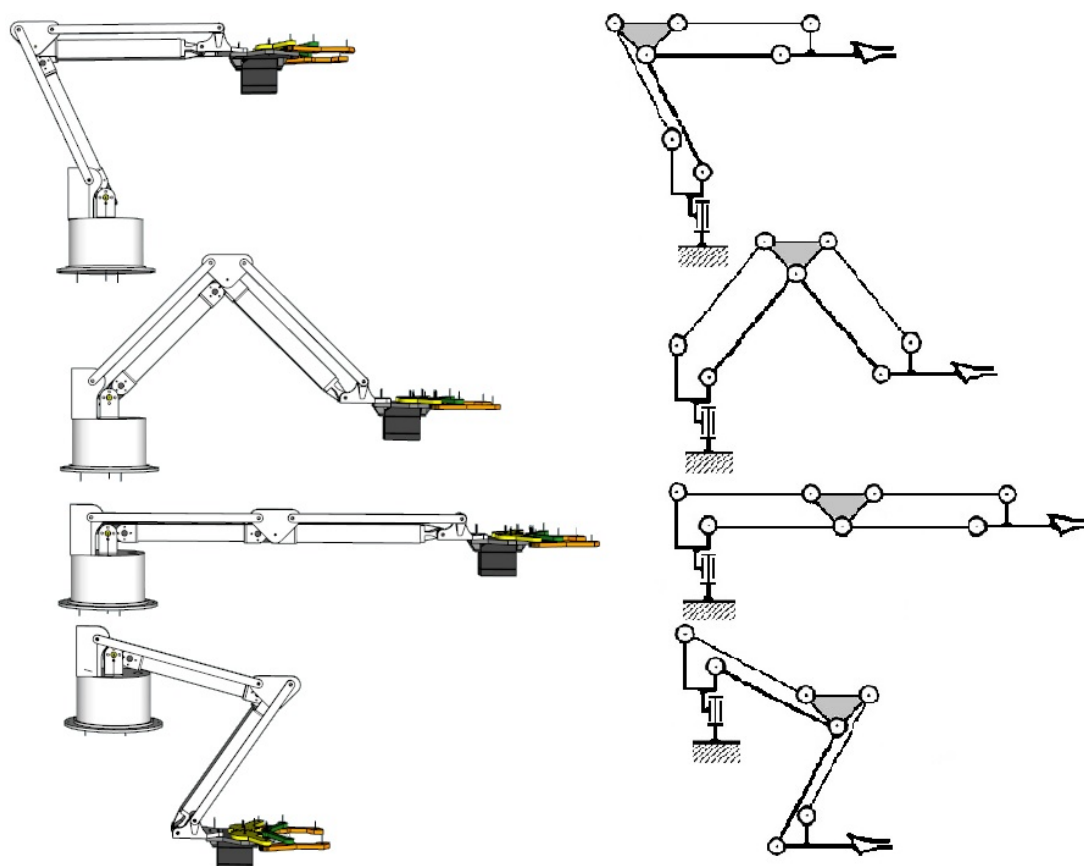


Рис. 3.34. Манипулятор в различных положениях

### 3.5. Определение диапазонов углов поворотов звеньев

Построив 3D модель манипулятора (Рис. 3.33) с учётом всех длин звеньев и размеров деталей, можно определить диапазоны углов поворотов звеньев робота относительно начального положения. Эти диапазоны будут учитываться при определении рабочей зоны динамической модели манипулятора.

Угол поворота основания  $q_0$ :

За поворот основания отвечает сервопривод SR430, диапазон его поворота составляет  $180^0$ . В конструкции робота нет препятствий, ограничивающих этот диапазон, следовательно робот будет поворачивать на  $\pm 90^0$  относительно начального положения (Рис. 3.35).



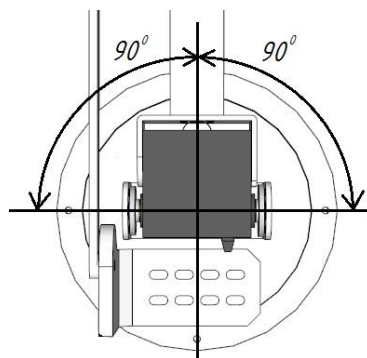


Рис. 3.35. Угол поворота основания  $q_0$

Угол поворота 1го звена  $q_1$ :

За поворот 1го звена отвечает сервопривод DF15RMG с диапазоном поворота  $170^\circ$ . Конструкция робота позволяет максимально наклонить звено в одну сторону на  $102^\circ$ , а на поворот в другую сторону  $38^\circ$  относительно начального положения (Рис. 3.36).

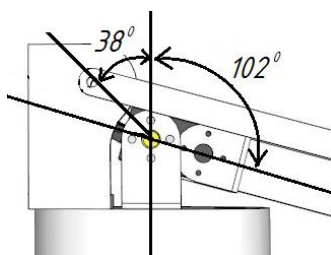


Рис. 3.36. Угол поворота 1го звена  $q_1$

Угол поворота 2го звена  $q_2$ :

За поворот 2го звена отвечает сервопривод RDS3115 с диапазоном поворота  $180^\circ$ . Конструкция робота позволяет максимально наклонить звено в одну сторону на  $48^\circ$ , поворот в другую возможен только на  $90^\circ$  (Рис. 3.37).

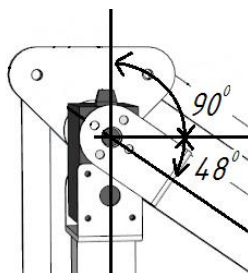


Рис. 3.37. Угол поворота 2го звена  $q_2$

Диапазоны поворотов углов были сделаны максимально возможными в одну сторону для того, чтобы сделать рабочую зону перед роботом как можно больше.

## 4. Разработка динамической модели манипулятора

### 4.1. Система виртуального моделирования Adams

Динамическая модель робота-манипулятора (Рис. 4.1) строилась в программе Adams View. При построении были учтены все размеры из кинематической схемы (Рис. 3.1) и 3D модели робота (Рис. 3.33).

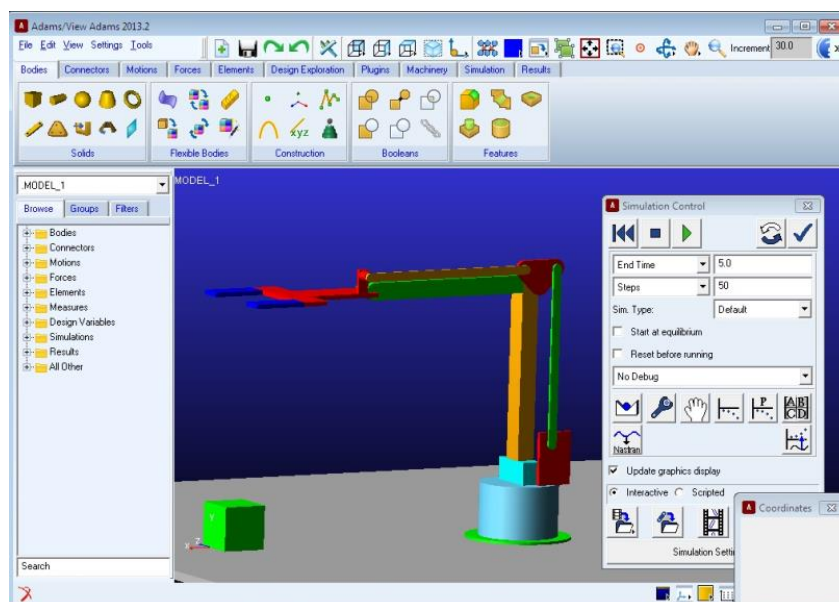


Рис. 4.1. Трёхмерная модель робота для моделирования динамики

ADAMS/View предназначен для создания, тестирования и оптимизации работы моделей механизмов и конструкций, состоящих из абсолютно твердых тел и их соединений (шарниров, нитей, пружин и т.д.).

Создание модели подразумевает описание всех ее характеристик: геометрических размеров, физических свойств, способов соединения подвижных и неподвижных частей, задание действующих сил и моментов, начального положения элементов модели и их скоростей.

Этап тестирования модели включает в себя моделирование поведения частей модели под действием приложенных сил и заданных движений и выявление критических параметров, наиболее сильно влияющих на эффективность работы модели в целом.

					<b>Разработка трехстепенного лабораторного манипулятора с механическим захватом</b>		
Изм.	Лист	№ докум.	Подпись	Дата			
Разработал	Ерёмин Д.О.				<b>Разработка динамической модели манипулятора</b>	Лит.	Лист
Проверил	Юдин Д.А.						Листов
Руководит.	Юдин Д.А.						41
Норм. Контр.	Гольцов Ю.А.					БГТУ им. В.Г. Шухова МР-41	
Зав. каф.	Руданов В.Г.						
							15

Построенная модель робота в Adams при входных данных: углы поворота, угловая скорость и масса звеньев, позволяет определить выходные данные: координаты положения звеньев в пространстве, силы, моменты, угловые ускорения действующие в сочленениях звеньев (Рис. 4.2).

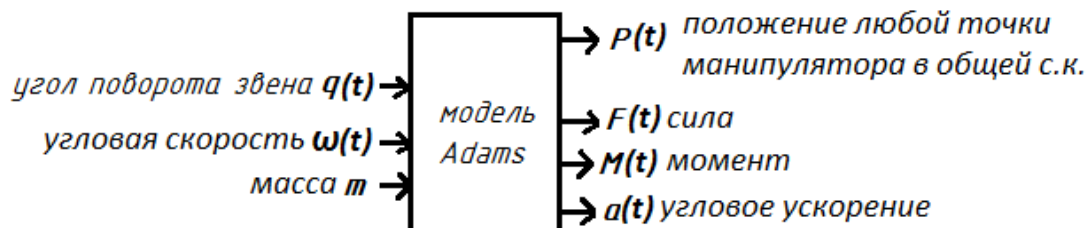


Рис. 4.2. Входные и выходные данные в модели манипулятора

Также возможны и другие варианты входных и выходных данных, например при заданном моменте найти угловую скорость и ускорение.

Основой для системы уравнений, описывающих динамику модели робота в программе Adams, послужили уравнения в форме Эйлера-Лагранжа с множителями

$$\begin{aligned} \dot{p}_i &= \frac{\partial L}{\partial q_i} + \sum_{k=1}^{m_a} \bar{F}_j \cdot \frac{\partial \bar{r}_k}{\partial q_i} - \sum_{j=1}^m \lambda_j \frac{\partial \Phi_j}{\partial q_i}, \\ p_i &= \frac{\partial L}{\partial \dot{q}_i} = 0, \\ u_i - \dot{q}_i &= 0, \quad i = 1, \dots, 6n, \\ \Phi_j(\{q_a\}, t) &= 0, \quad j = 1, m, \\ \bar{F}_k - \bar{f}_k(\{q_a\}, \{u_\beta\}, \{\lambda_\gamma\}, t) &= 0, \quad k = \overline{1, m_a}. \end{aligned}$$

Для задания движения твердого тела используются инерциальные глобальные координаты его центра масс и углы Эйлера. По умолчанию ориентация определяется последовательными поворотами вокруг главных центральных осей тела. Выбор одной из 24-х систем углов Эйлера должен быть сделан в процессе сборки модели, перед началом симуляции. Необходимость смены системы эйлеровых обобщенных координат, связана с вырождением матрицы связи проекций вектора угловой скорости на ортогональные оси и обобщенных скоростей [10].

## 4.2. Рабочая зона манипулятора

Была задана функция, с помощью которой модель робота повернула свои звенья по всему диапазону допустимых углов. На рисунке 4.3 показан скриншот движения модели манипулятора. При этом схват робота прошёлся по контуру своей рабочей зоны. Координаты точек контура были сохранены, по которым потом построилась рабочая зона на координатной плоскости. Контур рабочей зоны показан на плоскости по с осями  $OXZ$  на рисунке 4.4 и на плоскости с осями  $OXY$  на рисунке 4.5.

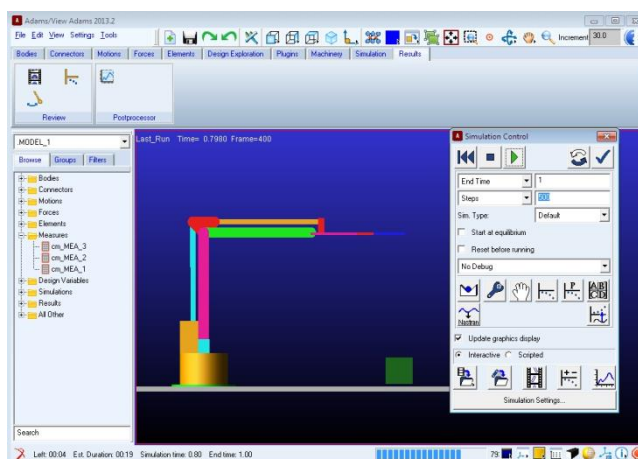


Рис. 4.3. Движение модели манипулятора для определения рабочей зоны  
Для большей наглядности была добавлена модель манипулятора на координатную плоскость с рабочей зоной.

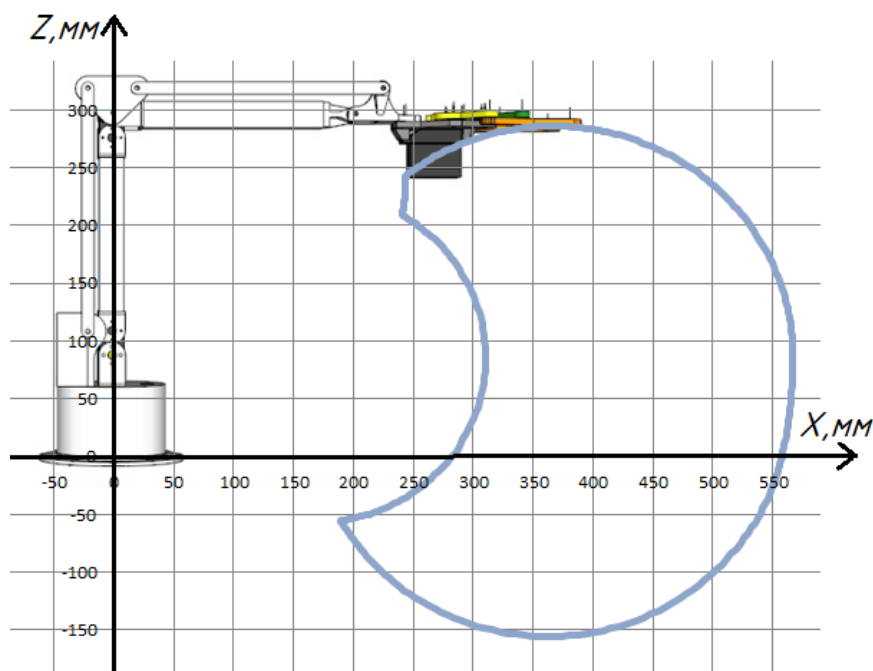


Рис. 4.4. Контур рабочей зоны с моделью манипулятора (вид сбоку)

					Разработка динамической модели манипулятора	Лист
Изм.	Лист	№ докум.	Подпись	Дата		43

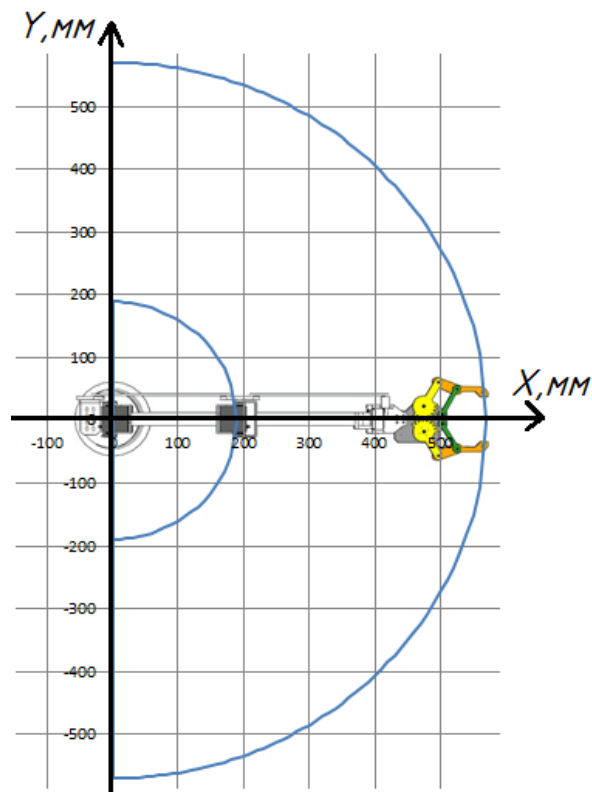


Рис. 4.5. Контур рабочей зоны с моделью манипулятора (вид сверху)

### 4.3. Динамические характеристики манипулятора

Нужно определить динамические характеристики в точках поворота звеньев манипулятора  $q_0$ ,  $q_1$ ,  $q_2$  (Рис. 4.6) при различных его положениях и передвижениях т.к. в этих точках расположены сервоприводы и важно понять какая будет действовать на них нагрузка.

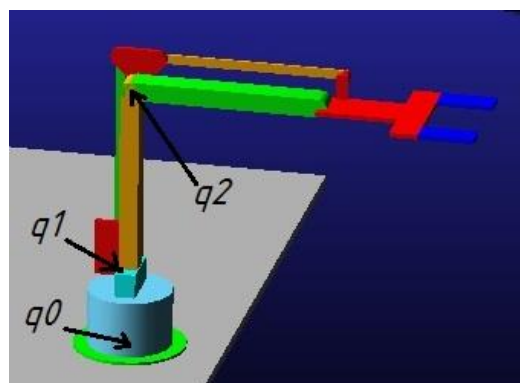


Рис. 4.6. точки в которых будут определены динамические характеристики  
Динамические характеристики будут следующими: сила  $F$ , момент  $M$ , угловая скорость  $\omega$ , угловое ускорение  $a$ . Чтобы их определить нужно задать движение звеньев модели манипулятора, а именно повернуть на

					Разработка динамической модели манипулятора	Лист
Изм.	Лист	№ докум.	Подпись	Дата		44

определённые углы с конкретной скоростью. Далее программа рассчитает массив данных, на основе которых построятся графики, по которым можно определить значения динамических характеристик при определённом угле поворота звена и в конкретный момент времени.

#### Определение динамических характеристик в точках $q_1$ и $q_2$

Было задано движение звеньев модели манипулятора таким образом, что он занимает 5 различных положений (Рис.4.7), при этом средняя угловая скорость поворота каждого звена составляла 1.57 рад/с, достигнув каждого положения манипулятор останавливался на 0.5 сек.

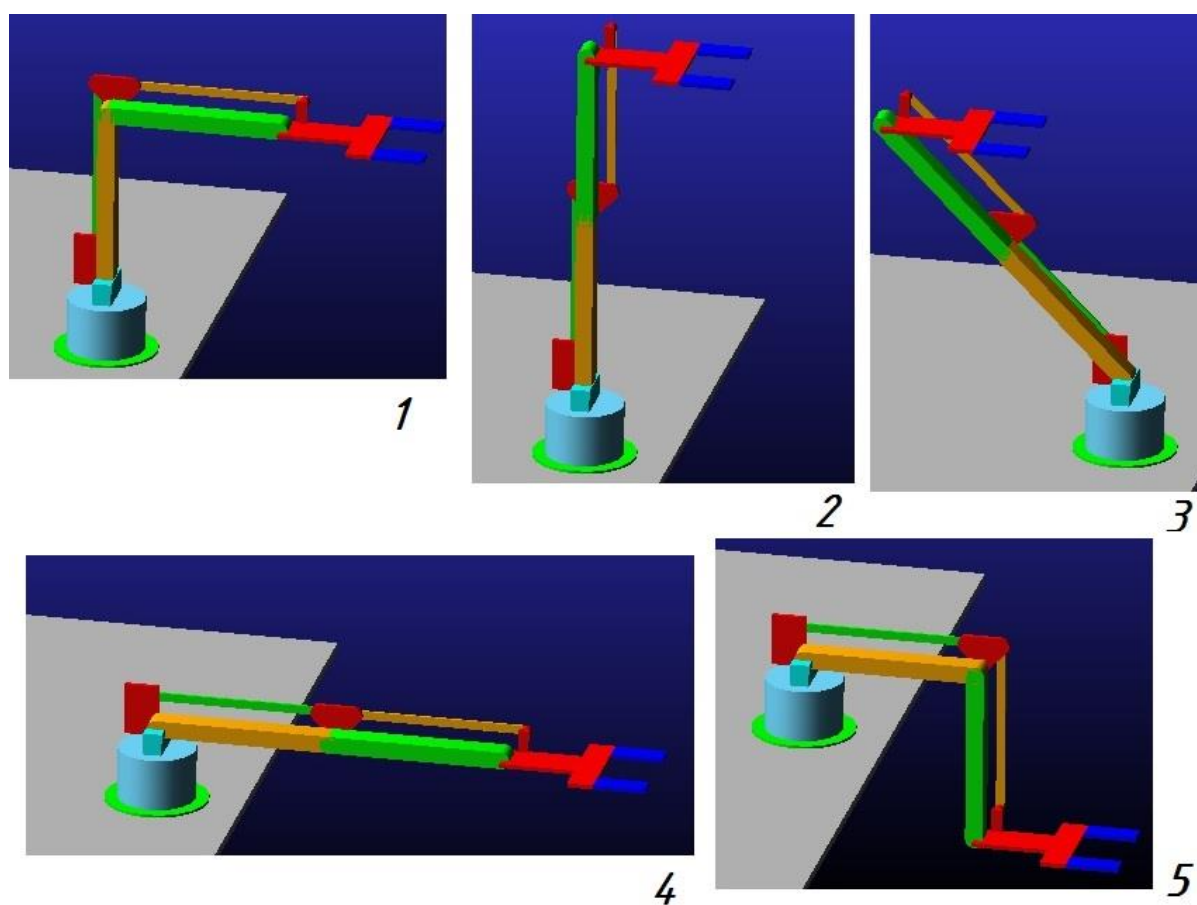


Рис. 4.7. Движение модели манипулятора для определения динамических характеристик

На рисунке 4.8 показаны графики: угол поворота  $q_1$  1ого звена, сила  $F$ , момент  $M$ , угловая скорость  $\omega$  и угловое ускорение  $a$ , зависящие от времени. В начальный момент времени, когда манипулятор в положении «1» угол поворота 1ого звена считается нулевым.

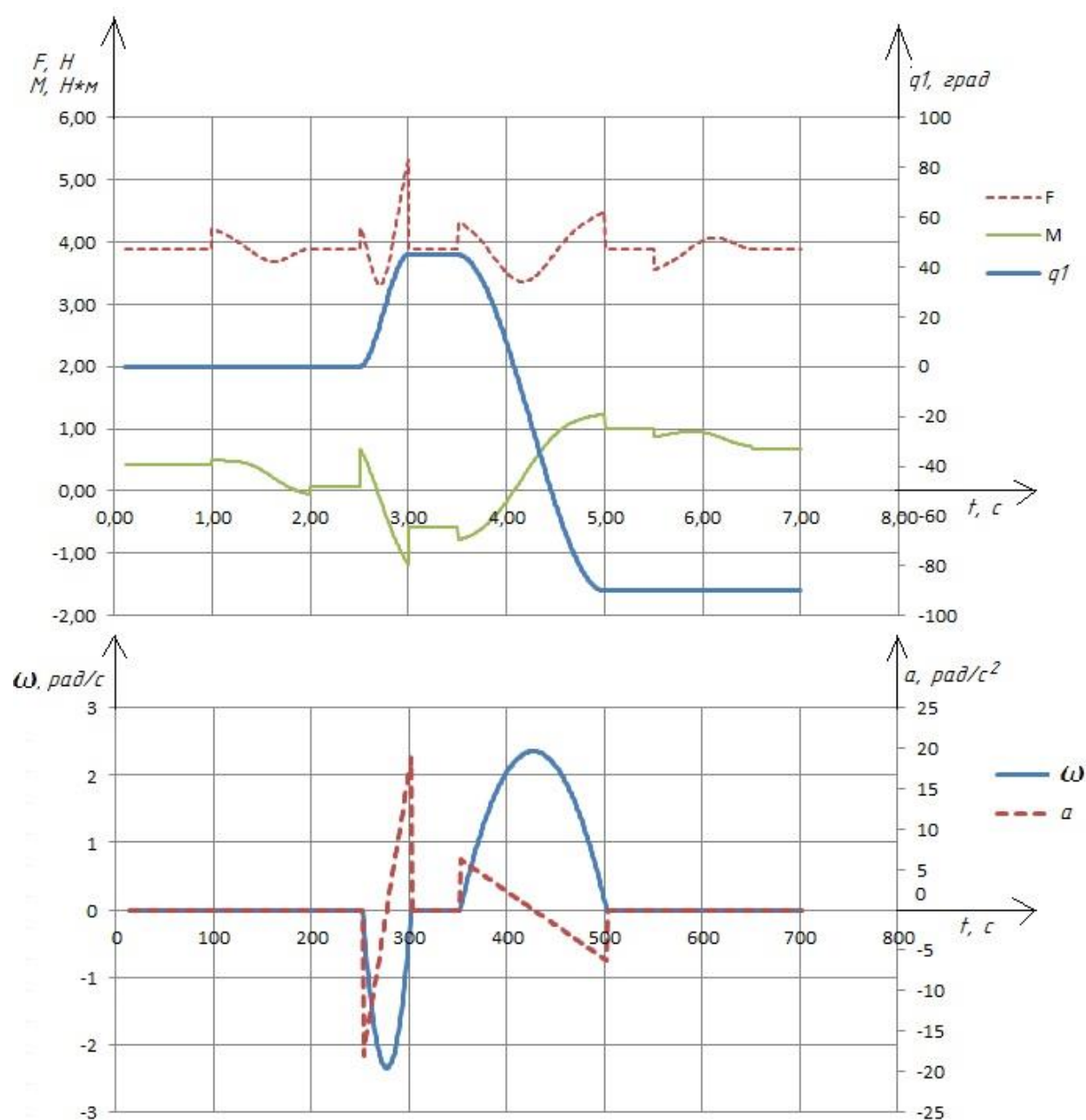


Рис. 4.8. Динамические характеристики манипулятора в точке  $q1$

Стоит заметить что в динамической модели угловая скорость изменяется плавно, а вращающий момент в среднем не превышает 1  $\text{H}\cdot\text{m}$ , что является допустимой нагрузкой реальный сервопривод.

Аналогично строятся графики динамических характеристик для точки  $q2$  (Рис. 4.9)



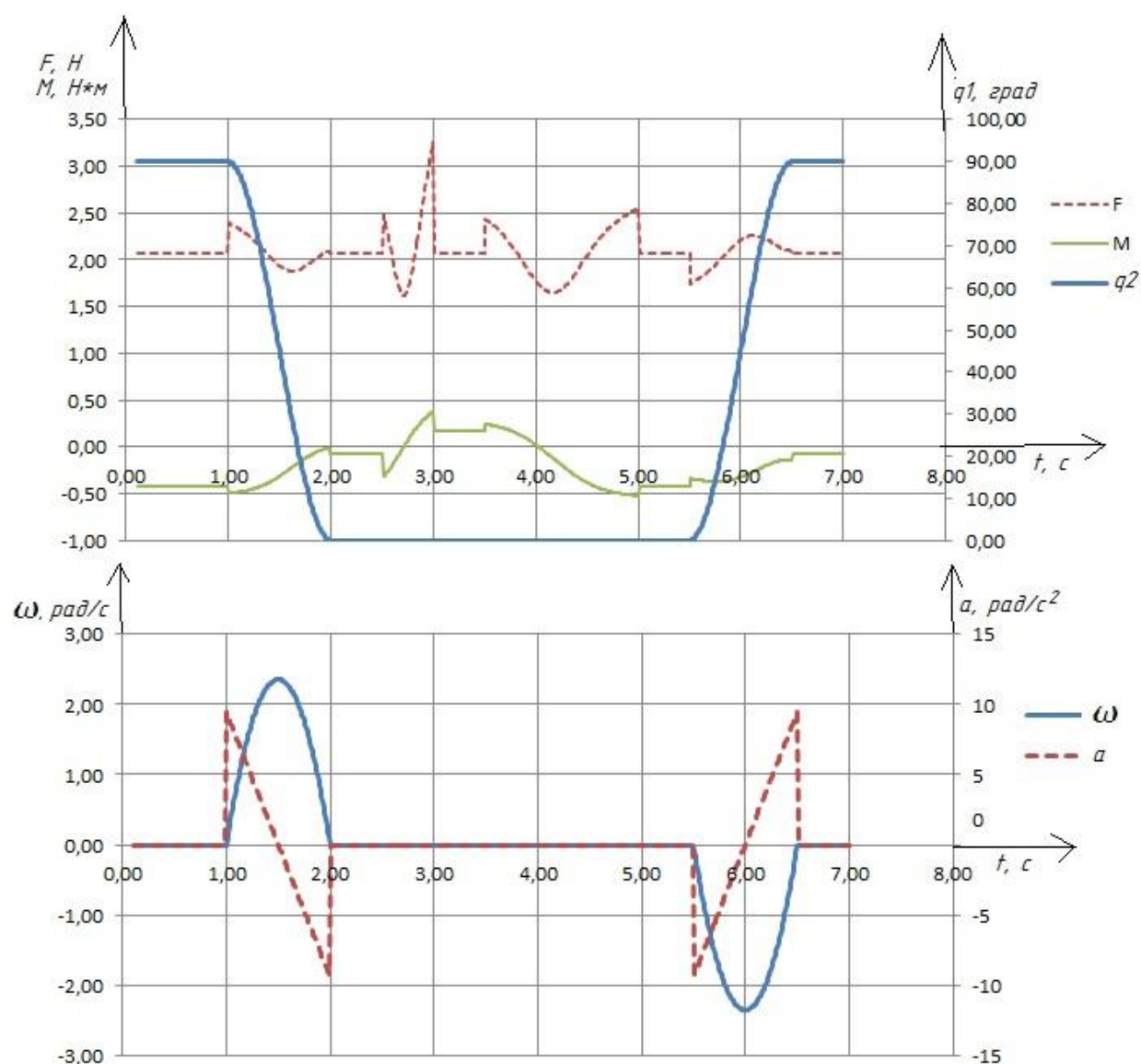


Рис. 4.9. Динамические характеристики манипулятора в точке  $q2$

Здесь угол  $q2$  поворота 2ого звена в начальный момент времени (положение «1») отклонён на 90 градусов.

#### Определение динамических характеристик в точке $q0$

Точка  $q0$  - это точка поворота основания манипулятора вокруг своей оси. Движение модели манипулятора пройдёт через 4 разных положения (Рис. 4.10). Сначала манипулятор поворачивает на 90 градусов с максимально вытянутой вперёд рукой, затем отклоняет руку назад и обратно поворачивается на 90 градусов.



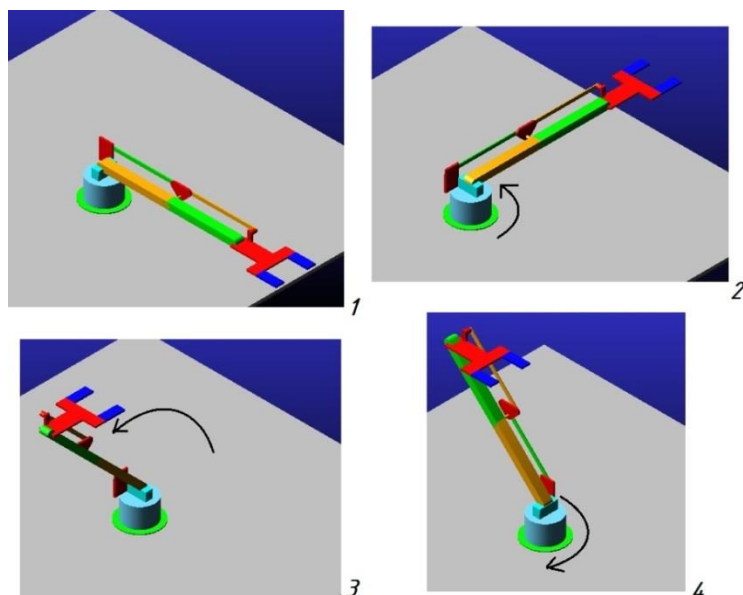


Рис. 4.10. Движение модели манипулятора для определения динамических характеристик

Графики динамических характеристик точки  $q_0$  показаны на рисунке 4.11

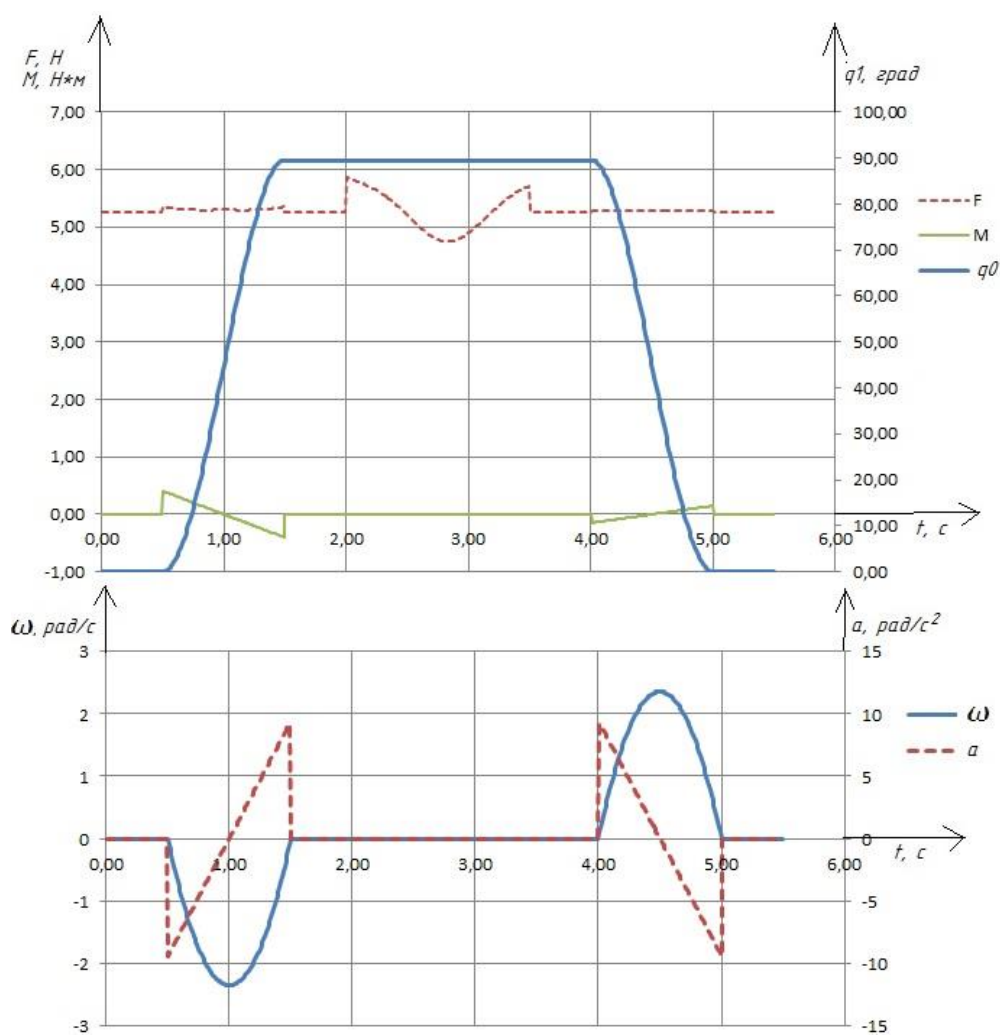


Рис. 4.11. Динамические характеристики манипулятора в точке  $q_0$

#### 4.4. Грузоподъёмность манипулятора

Грузоподъёмность манипулятора зависит от мощности его сервоприводов. Самая большая нагрузка идёт на сервопривод DF15RMG, отвечающий за поворот 1го звена на угол  $q_1$ . Он создаёт момент на валу  $19 \text{ кг*см} \approx 1.86 \text{ Н*м}$ . Максимальная грузоподъёмность – это тот максимальный вес, который манипулятор сможет поднять при максимально вытянутой вперёд руки.

Модель манипулятора ставится в положение с максимально вытянутой рукой вперёд, на ось 1го звена задаётся вращающий момент  $M=1.86 \text{ Н*м}$ , который сможет развить сервопривод, схват манипулятора держит груз, масса которого  $m$  постепенно увеличивается до тех пор, пока силы момента не хватит чтобы его поднять (Рис. 4.12). Максимальная массу которую смог поднять манипулятор = 150грамм (при 160грамм он уже опускается вниз).

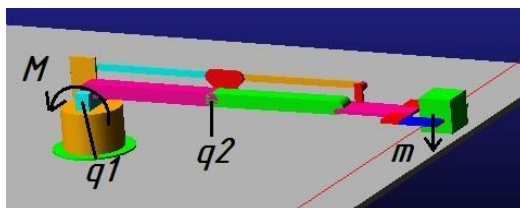


Рис. 4.12. Определение грузоподъёмности манипулятора

Таким же способом можно определить сколько поднимет сервопривод RDS3115, поворачивающий 2ое звено, с моментом  $M = 17 \text{ кг*см} \approx 1.67 \text{ Н*м}$  (Рис. 4.13).

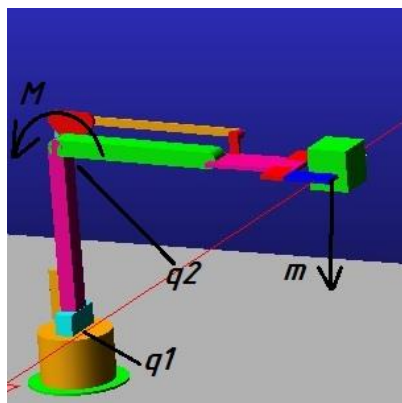


Рис. 4.13. Определение грузоподъёмности 2го звена

Максимальный вес, который смогло поднять 2ое звено составил 330 грамм.

					Разработка динамической модели манипулятора	Лист
Изм.	Лист	№ докум.	Подпись	Дата		49

Также с помощью динамической модели можно определить с какой скоростью манипулятор сможет повернуть свои звенья при определённом вращающем моменте и массе груза. Ниже показаны зависимости угловой скорости 1ого звена от времени и угла поворота  $q_1$  при вращающем моменте  $M=1.86 \text{ Н*м}$ , без груза (Рис. 4.14), с грузом массой 150 грамм (Рис. 4.15).

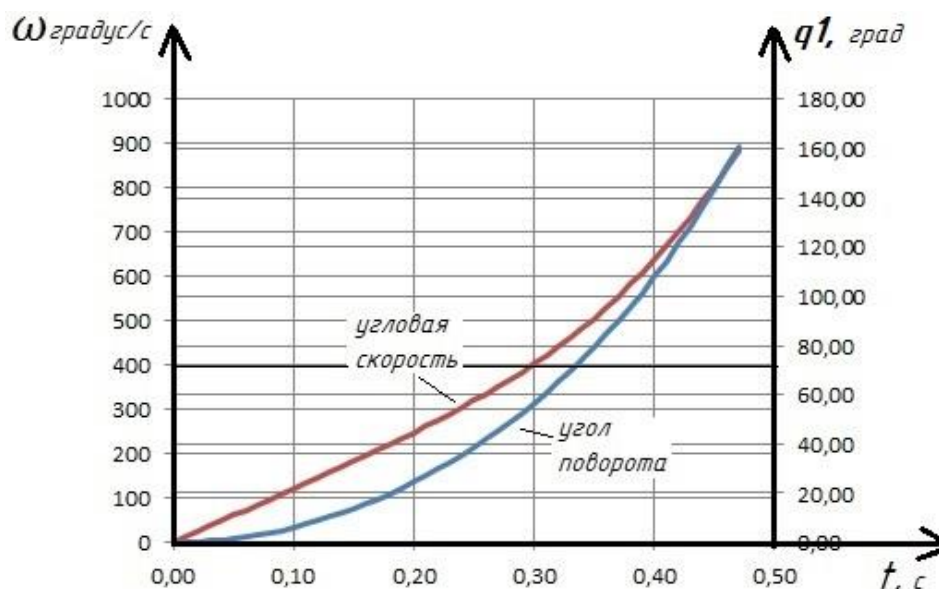


Рис. 4.14. Зависимость угловой 1ого звена скорости от времени (без груза)

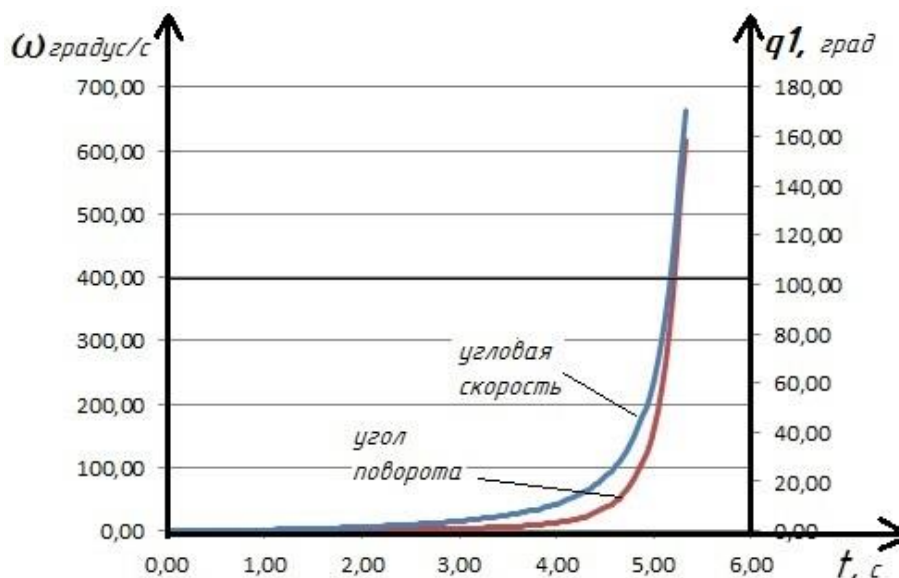


Рис. 4.15. Зависимость угловой 1ого звена скорости от времени (с грузом)

Такие же зависимости были определены для 2го звена с вращающим моментом  $M=1.67 \text{ Н*м}$  без груза (Рис. 4.16) и с грузом массой 150 грамм (Рис. 4.17).

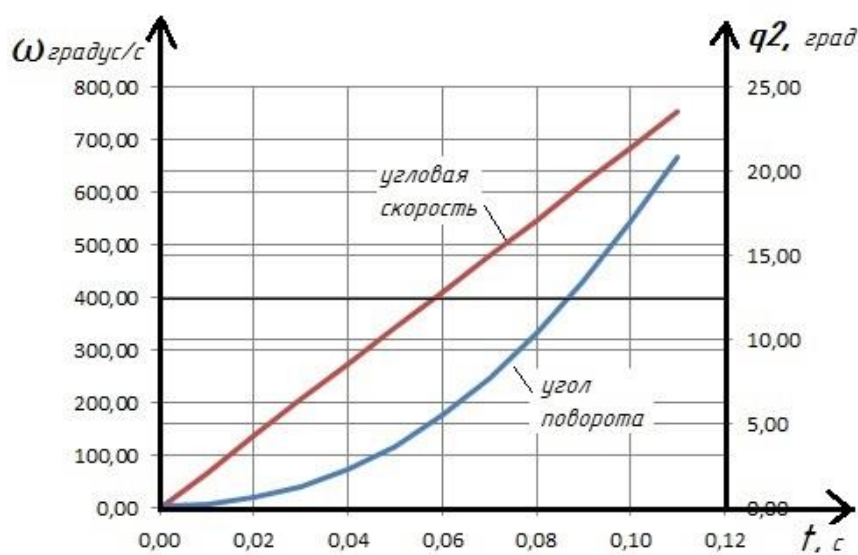


Рис. 4.16. Зависимость угловой 1ого звена скорости от времени (без груза)

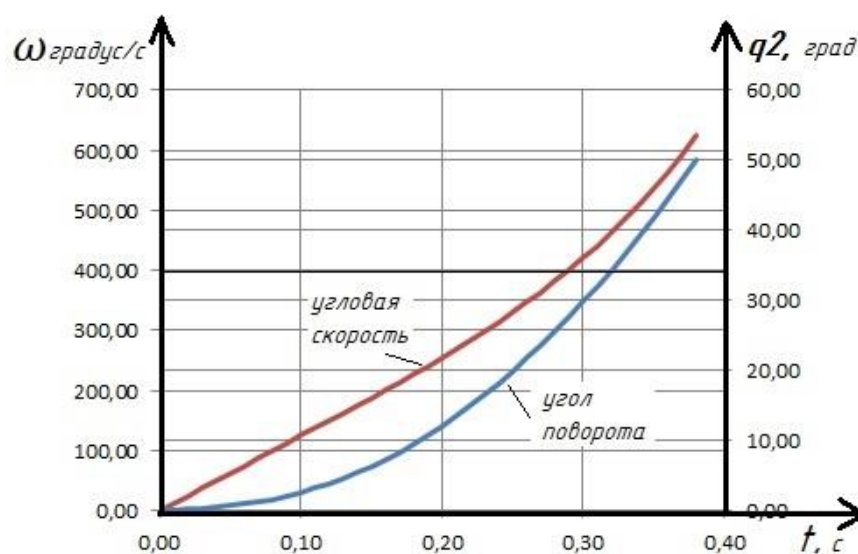


Рис. 4.17. Зависимость угловой 1ого звена скорости от времени (с грузом)

Стоит заметить что в реальности сервоприводы обоих звеньев не смогут вращаться быстрее 400 град/с (по техническим характеристикам). Так же в динамической модели не учитываются различные сопротивления и силы трения, поэтому реальные скорости окажутся куда ниже.

#### 4.5. Математическое описание динамической модели манипулятора

Для упрощения математического описания динамики манипулятора сделаем его поворотное основание неподвижным, тогда получим двухзвенный манипулятор (Рис. 4.18).

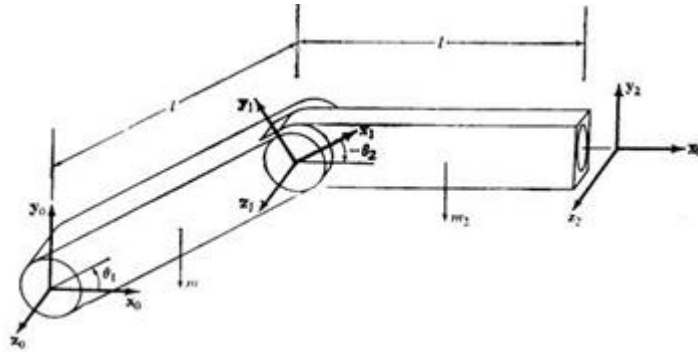


Рис. 4.18. Двухзвенный манипулятор

Для описания динамики движения манипулятора выполним применение уравнений Лагранжа-Эйлера.

присоединенными переменными являются  $\theta_1, \theta_2$ ;

-первое и второе звенья имеют массы  $m_1$  и  $m_2$ ;

-параметры звеньев имеют значения  $\alpha_1 = \alpha_2 = 0$ ;  $d_1 = d_2 = 0$ ;  $a_1 = a_2 = 0$ .

Тогда для матрицы имеем:  ${}^{i-1}A_i$  ( $i=1,2$ )

$${}^0A_1 = \begin{bmatrix} C_1 & S_1 & 0 & lC_1 \\ S_1 & C_1 & 0 & lS_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad {}^1A_2 = \begin{bmatrix} C_2 & -S_2 & 0 & lC_2 \\ S_2 & C_2 & 0 & lS_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad {}^0A_2 = \begin{bmatrix} C_{12} & -S_{12} & 0 & l(C_{12} + C_1) \\ S_{12} & C_{12} & 0 & l(S_{12} + S_1) \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

где  $C_i = \cos \theta_i$ ;  $S_i = \sin \theta_i$ ;  $C_{ij} = \cos [\theta_i + \theta_j]$ ;  $S_{ij} = \sin [\theta_i + \theta_j]$ .

В соответствии с определением матрицы  $Q_i$  для вращательного сочленения имеем:

$$Q_i = \begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix},$$

получаем:

$$U_{11} = \frac{\partial^0 A_2}{\partial Q_1} = Q_1^0 A_1 = \begin{bmatrix} -S_1 & -C_1 & 0 & -lS_1 \\ C_1 & -S_1 & 0 & lC_1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix},$$

аналогично для  $U_{21}$  и  $U_{22}$  получаем:

$$U_{21} = \frac{\partial^0 A_2}{\partial Q_1} = Q_1^0 A_1 = \begin{bmatrix} -S_{12} & -C_{12} & 0 & -l(S_{12} + S_1) \\ C_{12} & -S_{12} & 0 & l(C_{12} + C_1) \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix},$$

$$U_{22} = \frac{\partial^0 A_2}{\partial Q_2} = {}^0 A_1 Q_1^1 A_2 = \begin{bmatrix} -S_{12} & -C_{12} & 0 & -lS_{12} \\ C_{12} & -S_{12} & 0 & lC_{12} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}.$$

Полагая, что центробежные моменты инерции равны нулю, получим формулу для матрицы псевдоинерции  $J_i$ :

$$J_1 = \begin{bmatrix} \frac{1}{3}m_1 l^2 & 0 & 0 & -\frac{1}{2}m_1 l \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -\frac{1}{2}m_1 l & 0 & 0 & m_1 \end{bmatrix}; \quad J_2 = \begin{bmatrix} \frac{1}{3}m_2 l^2 & 0 & 0 & -\frac{1}{2}m_2 l \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -\frac{1}{2}m_2 l & 0 & 0 & m_2 \end{bmatrix}.$$

Определения слагаемых, описывающих центробежное и кориолисово ускорение:

$$h_1 = \sum_{k=1}^2 \sum_{m=1}^2 h_{1km} \dot{\theta}_k \dot{\theta}_m = h_{111} \dot{\theta}_1^2 + h_{112} \dot{\theta}_1 \dot{\theta}_2 + h_{121} \dot{\theta}_1 \dot{\theta}_2 + h_{122} \dot{\theta}_2^2,$$

$$h_1 = -\frac{1}{2}m_2 S_2 l^2 \theta_2^2 - m_2 S_2 l^2 \theta_1 \theta_2,$$

$$h(\theta, \dot{\theta}) = \begin{bmatrix} -\frac{1}{2}m_2 S_2 l^2 \theta_2^2 - m_2 S_2 l^2 \dot{\theta}_1 \dot{\theta}_2 \\ \frac{1}{2}m_2 S_2 l^2 \dot{\theta}_2^2 \end{bmatrix}.$$

Слагаемые, определяющие влияние гравитационных сил  $c = (c_1, c_2)^T$ :

$$c_1 = -(m_1 g U_{11}^1 \bar{r}_1 + m_2 g U_{21}^2 \bar{r}_2) = -m_1 (0, g, 0, 0) \begin{bmatrix} -S_1 & -C_1 & 0 & -lS_1 \\ C_1 & -S_1 & 0 & lC_1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \times \begin{bmatrix} -l/2 \\ 0 \\ 0 \\ 1 \end{bmatrix} -$$

$$-m_2 (0, g, 0, 0) \begin{bmatrix} -S_{12} & -C_{12} & 0 & -l(S_{12} + S_1) \\ C_{12} & -S_{12} & 0 & l(C_{12} + C_1) \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \times \begin{bmatrix} -l/2 \\ 0 \\ 0 \\ 1 \end{bmatrix} = \frac{1}{2} m_1 g l C_1 + \frac{1}{2} m_2 g l C_{12} + m_2 g l C_1;$$

$$c_2 = -m_2 g U_{22}^2 \bar{r}_2 = -m_2 (0, g, 0, 0) \begin{bmatrix} -S_{12} & -C_{12} & 0 & -lS_{12} \\ C_{12} & -S_{12} & 0 & lC_{12} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \times \begin{bmatrix} -l/2 \\ 0 \\ 0 \\ 1 \end{bmatrix} = -m_2 \left( \frac{1}{2} g l C_{12} - g l C_{12} \right)$$

Таким образом, вектор, определяющий влияние силы тяжести:

$$c(\theta) = \begin{bmatrix} c_1 \\ c_2 \end{bmatrix} = \begin{bmatrix} \frac{1}{2} m_1 g l C_1 + \frac{1}{2} m_2 g l C_{12} + m_2 g l C_1 \\ \frac{1}{2} m_2 g l C_{12} \end{bmatrix}.$$

Окончательно имеем уравнения описывающие динамику движения манипулятора, где  $\tau_i$  сила, которую должен развить силовой привод  $i$ -го сочленения, чтобы реализовать данное движение:

$$\tau(t) = D(\theta) \ddot{\theta}(t) + h(\theta, \dot{\theta}) + c(\theta),$$

$$\begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} = \begin{bmatrix} \frac{1}{3} m_1 l^2 + \frac{4}{3} m_2 l^2 + m_2 C_2 l^2 & \frac{1}{3} m_2 l^2 + \frac{1}{2} m_2 l^2 C_2 \\ \frac{1}{3} m_2 l^2 + \frac{1}{2} m_2 l^2 C_2 & \frac{1}{3} m_2 l^2 \end{bmatrix} \begin{bmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \end{bmatrix} +$$

$$+ \begin{bmatrix} -\frac{1}{2} m_2 S_2 l_2 \theta_2^2 - m_2 S_2 l^2 \theta_1 \theta_2 \\ \frac{1}{2} m_2 S_2 l^2 \theta_1^2 \end{bmatrix} + \begin{bmatrix} \frac{1}{2} m_1 g l C_1 + \frac{1}{2} m_2 g l C_{12} + m_2 g l C_1 \\ \frac{1}{2} m_2 g l C_{12} \end{bmatrix}.$$

Вычисление данных уравнений выполняется в программе Matlab. Код вычисления уравнений представлен в приложении 1.

Для сравнения работы математической и компьютерной динамических моделях манипулятора был вычислен график зависимости момента от времени в сочленении 2-го и 1-го звеньев. График компьютерной модели

был показан в главе 4.3 на рисунке 4.9. Для расчёта момента в упрощённой математической модели были использованы данные из расчётов в главе 4.3, а именно функции угловых скоростей и ускорений 1-го и 2-го звеньев, а так же их массы и длины. Это выполнялось для того, чтобы расчёт момента в математической модели выполнялся при тех же условиях, которые были заданы в компьютерной модели. На рисунке 4.19 представлено сравнение графиков момента, возникающего в сочленении 2-го и 1-го звена при движении манипулятора. Как видно форма графиков похожа, но в некоторой степени отличаются значения моментов на разных графиках. Такая разница происходит из-за того, что в математической модели рассчитывалась упрощённая схема двухзвенного манипулятора, а также не учитывалось влияние параллельного механизма манипулятора.

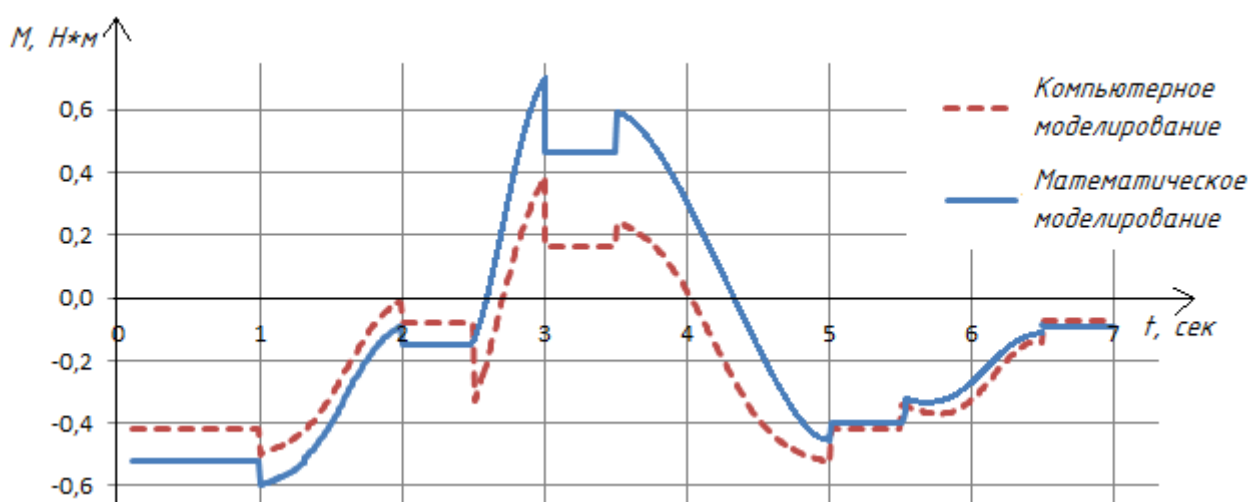


Рис. 4.19. Сравнение графиков динамической характеристики манипулятора, рассчитанных на компьютерной и математической динамических моделях



## 5. Разработка аппаратного комплекса манипулятора

### 5.1. Печать деталей на 3D-принтере

Детали схвата манипулятора детали изготавливались при помощи 3D печати на 3D принтере Makerbot Replicator. Процесс печати показан на рисунке 5.1. Детали печатались пустотелыми, для экономии материала, и получились лёгкими по весу, что хорошо скажется на грузоподъёмности манипулятора, при этом они обладают достаточной прочностью.



Рис. 5.1. Печать деталей для схвата манипулятора

Так же на принтере Makerbot Replicator печатались детали для шарнира, который соединяет хват и 2ое звено манипулятора, эти детали также являются пустотелыми для уменьшения веса. Все детали печатались по 3D моделям, разработанным в главе 3.4.

Всё остальное печаталось на 3D принтере 3DTouch. Это были детали основания и детали параллельного механизма. Они печатались с более грубой точностью чем детали схвата и не являются пустотелыми, при этом обладают значительной массой, но являются очень прочными. Прочность основания играет большую роль, т.к. на него действуют самые большие нагрузки. При этом большой вес основания не влияет на грузоподъёмность манипулятора. На рисунке 5.2 показаны напечатанные деталь основания и кронштейн для крепления звена параллельного механизма к основанию.

					<b>Разработка трехстепенного лабораторного манипулятора с механическим захватом</b>		
Изм.	Лист	№ докум.	Подпись	Дата			
Разработал		Ерёмин Д.О.			<b>Разработка аппаратного комплекса манипулятора</b>		
Проверил		Юдин Д.А.					
Рцководит.		Юдин Д.А.					
Норм. Контр.		Гольцов Ю.А.					
Зав. каф.		Руданов В.Г.					
						Лит.	Лист
							56
							5
						БГТУ им. В.Г. Шухова	
						МР-41	

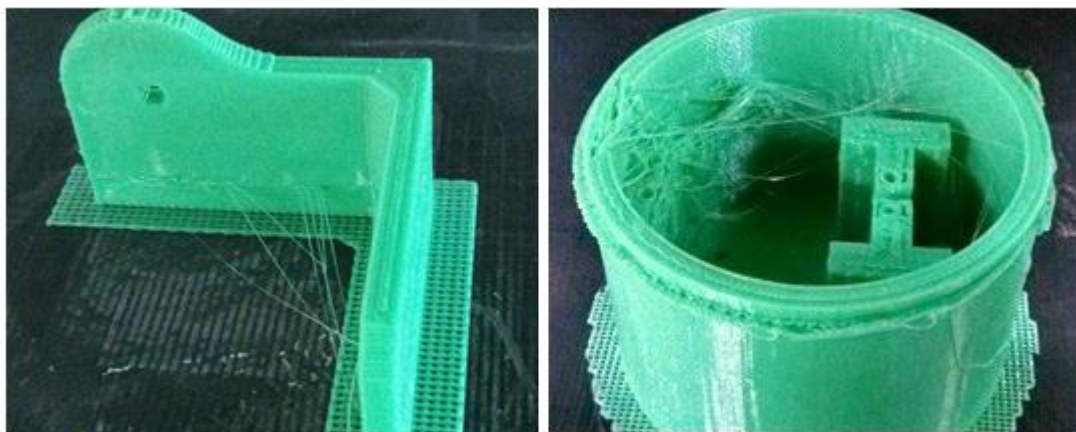


Рис. 5.2. Напечатанные деталь для основания и кронштейн

## 5.2. Ручное изготовление деталей

Детали 1го и 2го звена и дополнительных звеньев параллельного механизма манипулятора изготавливаются вручную из алюминиевых реек. Т.к. эти детали достаточно длинные, они должны обладать хорошей прочностью и жёсткостью. Алюминий обладает этими качествами в отличие от пластика, который более гибкий, и использование пластика привело бы к сильным колебаниям манипулятора во время движения.

Детали изготавливались согласно размерам 3D моделей из главы 3.4. На рисунке 5.3 изображена деталь 1го звена с прикреплёнными к ней кронштейнами сервоприводов.



Рис. 5.3. Деталь 1го звена

На деталь 2го звена кронштейн сервопривода крепится только с одной стороны (Рис. 5.4), к другой стороне будет крепиться шарнирное соединение схвата.



Рис. 5.4. Деталь 2го звена

Чтобы манипулятор стоял устойчиво и не опрокинулся при работе, было решено сделать для него специальную платформу, к которой будет крепиться основание манипулятора. Размеры платформы будут 50х50 см. Т.к. значительная часть рабочей зоны манипулятора находится ниже его основания (Рис. 4.4), было решено поднять основание над платформой на 10 см. На рисунке 5.5 изображена модель платформы с указанными размерами.

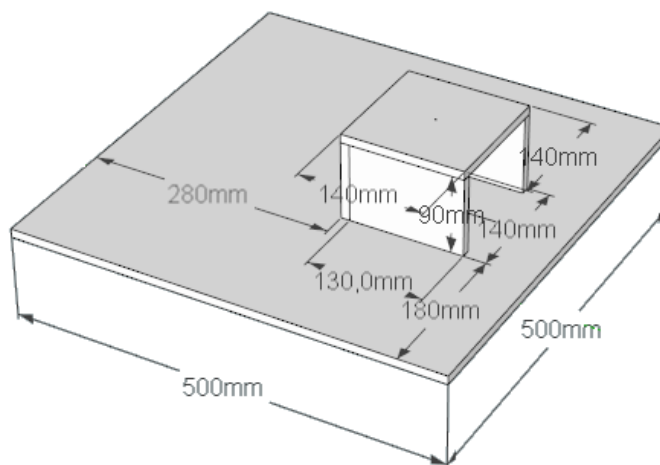


Рис. 5.5. Схема платформы для манипулятора

Платформа изготавливалась целиком из фанеры толщиной 10 мм, все элементы платформы скреплены саморезами (Рис. 5.6).



Рис. 5.6. Платформа для манипулятора

### 5.3. Сборка манипулятора

Все детали схвата скреплены болтами диаметром 4мм, они не слишком затянуты, чтобы хват работал без затруднений. На рисунке 5.7 изображен собранный хват (вид сверху и снизу).

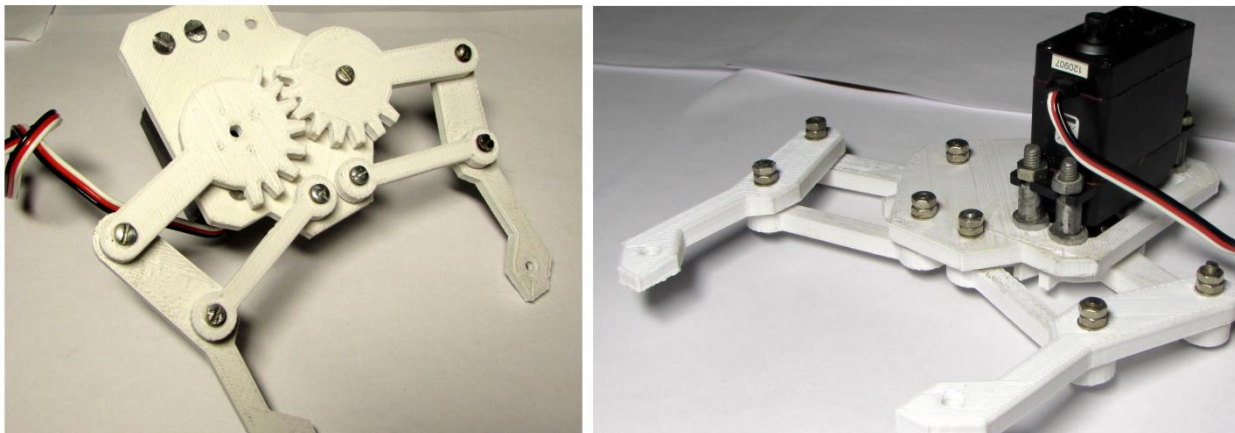


Рис. 5.7. Собранный хват манипулятора

На рисунке 5.8 изображено скрепленные 1ое и 2ое звено, это основная часть конструкции манипулятора, её вес составляет 260 грамм.



Рис. 5.8. 1ое и 2ое звенья манипулятора

Схват крепится к 2му звену через специальное шарнирное соединение (Рис. 5.9).



Рис. 5.9. Шарнирное соединение

Нижняя неподвижная часть основания манипулятора крепится болтами к платформе (Рис. 5.10), затем к ней крепится верхняя подвижная часть основания.





Рис. 5.10. Нижняя часть основания

Полностью собранный робот, установленный на платформу и подключенный к плате Arduino и блоку питания показан на рисунке 5.11



Рис. 5.11. Готовый манипулятор

## 6. Разработка системы управления манипулятором

### 6.1. Разработка системы управления манипулятором на основе Arduino

Система управления манипулятором разработана на основе Arduino Uno которая подключена к компьютеру через COM порт, по которому поступает на Arduino питание и команды для управления сервоприводами манипулятора. Четыре сервопривода подключены к Arduino к цифровым портам под номерами 6, 8, 10, 12 (Рис. 6.1). Сервоприводы подключены параллельно к отдельному источнику питания, номинальное напряжение питания 7В. Также земля (GND) сервоприводов подключена к Arduino к контакту GND.

Сервопривод “servo 0”, подключенный к pin 6 (Рис. 6.1), управляет поворотом основания манипулятора, модель сервопривода SR430. “servo 1”, подключенный к pin 8, управляет поворотом 1го звена манипулятора, модель сервопривода DF15RMG. “servo 2”, подключенный к pin 10, управляет поворотом 2го звена манипулятора, модель сервопривода RDS3115. “servo 3”, подключенный к pin 12, управляет сжатием/разжатием механического схвата манипулятора, модель сервопривода SR430.

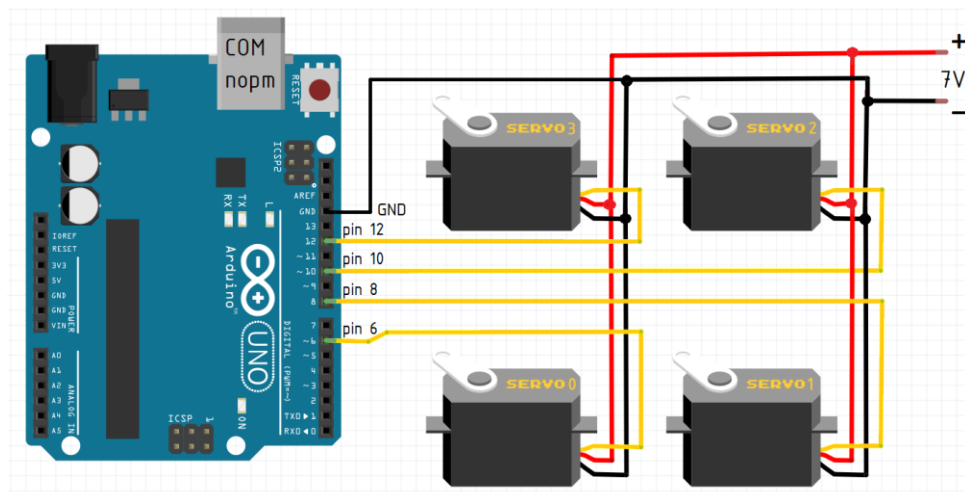


Рис. 6.1. Схема подключения сервоприводов к Arduino Uno

					<b>Разработка трехстепенного лабораторного манипулятора с механическим захватом</b>		
Изм.	Лист	№ докум.	Подпись	Дата			
Разработал	Ерёмин Д.О.				<b>Разработка системы управления манипулятором</b>		
Проверил	Юдин Д.А.						
Руководит.	Юдин Д.А.						
Норм. Контр.	Гольцов Ю.А.						
Зав. каф.	Рубанов В.Г.						
					Лит.	Лист	Листов
						61	7
					БГТУ им. В.Г. Шухова МР-41		

При подключении Arduino к питанию, в ней тут же запускается программа и посылает сигналы на сервоприводы, чтобы они повернулись в начальное положение. Далее программа начинает ожидать команды, поступающие в COM порт с компьютера. Команды являются строками, содержащими значения углов поворотов сервоприводов манипулятора. Как только поступает команда (строка) в COM порт, программа считывает данные углов поворота из этой строки и отправляет сигналы управления с новыми значениями на сервоприводы. Упрощённая блок-схема работы программы Arduino представлена на рисунке 6.2.

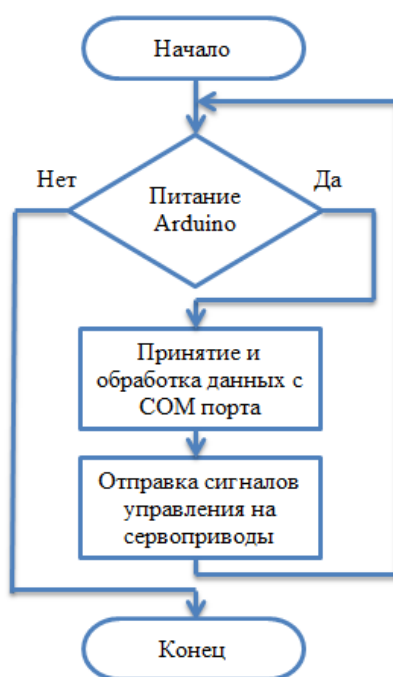


Рис. 6.2. упрощённая блок-схема программы Arduino

Код программы для Arduino представлен в приложении 2.

## 6.2. Разработка компьютерной программы для управления манипулятором

Компьютерная программа выполняет все вычисления для управления манипулятором. Она позволяет выполнить следующее: вручную управлять поворотами каждого звена по отдельности а также схватом; управлять передвижением рабочей точки манипулятора вдоль осей X, Y, Z декартовой системы координат; управлять скоростью движения манипулятора;

позволить программировать манипулятор для автоматического передвижения через конкретные точки.

Код программы написан на языке C# в среде Visual Studio [21]. Внешне программа выглядит в виде формы, на которой расположены отдельные панели, выполняющие свои функции, например: панель для ручного управления роботом, которое выполняется с помощью кнопок; панель настройки скорости движения манипулятора; панель программирования манипулятора, которая позволяет задать и запустить программу автоматического управления роботом; панель информации, где указываются данные о состоянии манипулятора. На рисунке 6.3 показана копия экрана программы в которой стоят обозначения отдельных её функций, которые далее будут более подробно описаны.

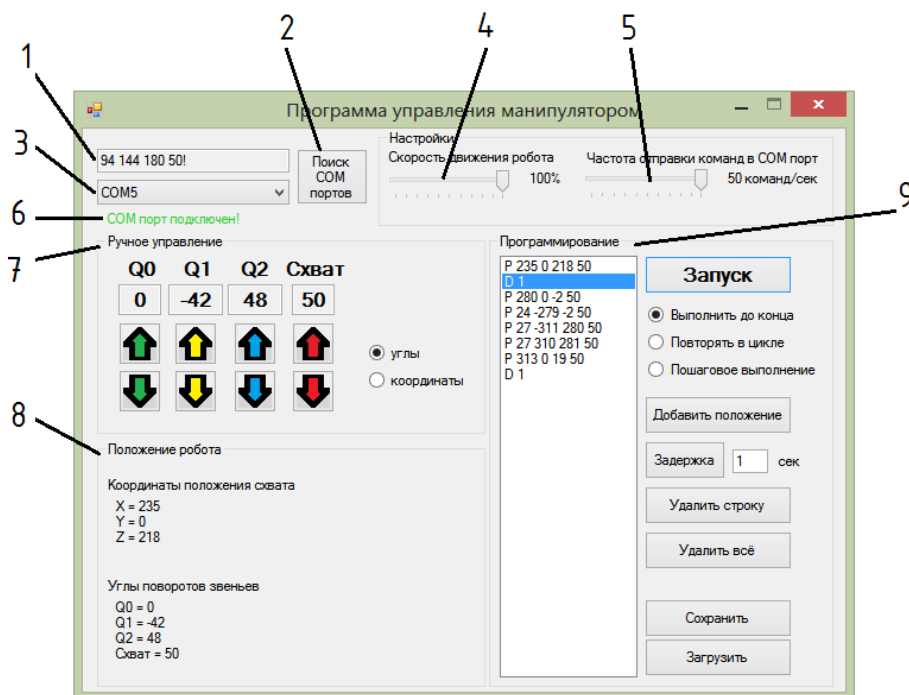


Рис. 6.3. Копия экрана программы управления манипулятором

1 – отображение строки (команды), посылаемой в COM порт. В строке содержатся значения углов поворотов для четырёх сервоприводов, которыми управляет плата Arduino.

2 – кнопка поиска доступных COM портов, используется когда уже после запуска программы подключили Arduino через COM порт. При запуске программа выполняет единожды поиск подключенных COM портов и



автоматически подключается к первому найденному.

3 – отображение названия COM порта в который программа посылает команды управления. Можно раскрыть список и выбрать другой доступный порт, если такой есть.

4 – ползунок регулировки скорости движения манипулятора. Чем больше значение, тем быстрее будут осуществляться повороты сервоприводов и тем быстрее будет перемещаться рабочий орган манипулятора.

5 – ползунок регулировки частоты отправки команд в COM порт. Чем выше значение, тем чаще плата Arduino будет получать новые данные о положении манипулятора, но тем сильнее загруженность порта. Команды отправляются лишь в тот момент, когда изменяется положение манипулятора.

6 – вывод информации о состоянии программы. Там указываются сообщения подключен или не подключен COM порт; возникает ли ошибка при передачи в COM порт; достиг ли робот края рабочей зоны.

7 – панель ручного управления манипулятором. Содержит 2 режима управления: “углы” и “координаты”. В первом режиме при помощи кнопок со стрелками осуществляется управление поворотами каждого звена по отдельности, где Q0 – угол поворота основания манипулятора, Q1 – угол поворота 1го звена, Q2 – угол поворота 2го звена относительно нулевого положения робота. Второй режим позволяет управлять передвижением только рабочим органом манипулятора, его можно перемещать по всем трём осям X, Y, Z декартовой системы координат. На рисунке 6.4 указаны направление координат, начало координат находится в самом низу по центру основания манипулятора, также робот стоит в нулевом положении, т.е. когда все углы поворотов звеньев равны нулю. Значения углов поворотов звеньев отсчитываются именно от этого положения. Также в обоих режимах можно управлять схватом, где число обозначает степень разжатия схвата (0 – схват полностью сжат, 90 – схват полностью разжат). Кнопка “Запуск” запускает программу с той команды, которая выделена. Есть три режима выполнения программы: 1) программа выполнится до конца и остановится; 2) дойдя до

					<i>Разработка системы управления манипулятором</i>	Лист
						64
Изм.	Лист	№ докум.	Подпись	Дата		

конца, программа начнёт повторно выполняться; 3) выполняется лишь одна команда. На рисунке 6.4 показан процесс выполнения программы. При выполнении программы блокируется ручное управление, а в панель информации добавляются данные о состоянии выполнения программы. Если остановить программу т.е. нажать кнопку “Стоп”, то можно будет продолжить программу с того же места, а если во время паузы отвести робота в сторону, то при продолжении выполнения программы он вернётся к тому положению, на котором был остановлен. Если при работе программы выделить мышкой другую команду, то робот начнёт её выполнять.

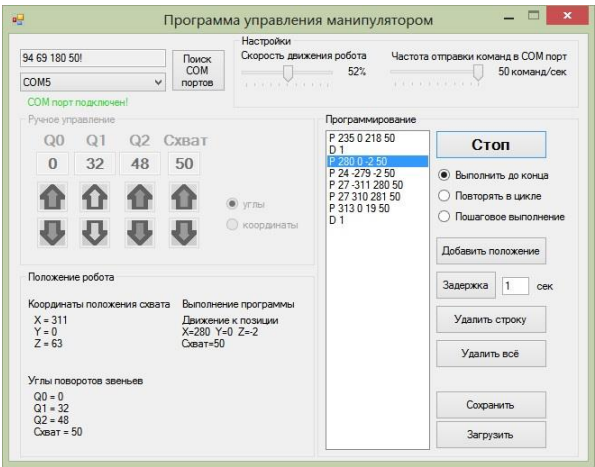


Рис. 6.4. Процесс выполнения программы

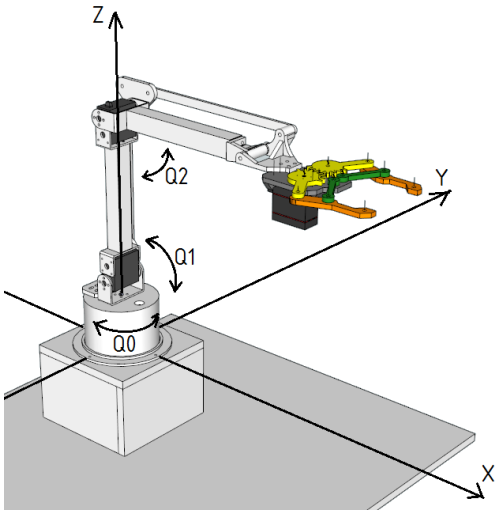


Рис. 6.4 Направление координатных осей рабочей зоны манипулятора

8 – панель информации. Выводит данные о положении манипулятора такие как углы поворотов звеньев, уровень сжатия схвата и координаты положения рабочей точки манипулятора.

9 – панель программного управления манипулятором. Содержит в себе поле, в котором отображаются строки команд управления. Команды бывают двух видов: “P X Y Z S” и “D T”, где P – point – команда точки с координатами X, Y, Z и уровнем сжатия схвата S, D – delay – команда задержки на время T секунд. Процесс программирования следующий: в ручном режиме подвести манипулятор к нужной точки и нажать кнопку “Добавить положение”, программа запомнит координаты этой точки и создаст команду. Создав программу из множества точек можно нажать кнопку “Запуск” и робот пройдёт через каждую точку в той последовательности, в которой они стоят. Кнопка “Задержка” добавляет команду задержки, где справа в поле можно указать время задержки. Можно сохранять или загрузить готовую программу. Она сохраняется в формате txt, где её можно полностью редактировать вручную. Если вписать несуществующие команды, то при выполнении программы они проигнорируются.

Ниже на рисунке 6.5 представлена упрощённая структура компьютерной программы управления манипулятором. Стрелочками указаны взаимосвязи отдельных подпрограмм и направления поступления информации.

В центре блок-схемы находится основная подпрограмма, которая содержит в себе все режимы управления манипулятором. При управлении поворотами звеньев робота решается прямая задача для определения координат рабочей точки, при переключении на режим управления координатами, эти координаты используются чтобы продолжить двигаться роботу с того же места, при этом решается обратная задача, чтобы определить углы поворотов звеньев. Данные об углах звеньев обрабатываются и посылаются в СОМ порт в виде строки. При программном управлении в командах программы считываются координаты точки, далее по этим координатам решается обратная задача и определяются углы поворотов звеньев, к которым робот стремится повернуть свои звенья, чтобы достичь заданной точки.

					<i>Разработка системы управления манипулятором</i>	Лист
						66
Изм.	Лист	№ докум.	Подпись	Дата		



Рис. 6.5. Упрощённая структура программы управления манипулятором  
Код программы представлен в приложении 3.

## 7. Тестирование работы манипулятора

В ходе тестирования работы манипулятора проверяются стабильность работы компьютерной программы и её связь с платой Arduino, наличие соединения сервоприводов с Arduino и подключение внешнего питания, работа манипулятора согласно заданной программы движения.

Сначала подключаются сервоприводы к Arduino согласно схеме на рисунке 6.1, далее Arduino подключается по COM порту к компьютеру. При подачи питания на сервоприводы манипулятор становится в начальное положение (Рис. 7.1).



Рис. 7.1. Начальное положение манипулятора

При запуске компьютерной программы управления манипулятором, она сразу определяет подключенный COM порт: выводится сообщение “COM порт подключен!”. Для проверки правильности подключения сервоприводов к Arduino в ручном режиме управления проверяется каждое звено по отдельности, т.е. при нажатии кнопки должен отреагировать соответствующий сервопривод. Все сервоприводы подключены верно. Для проверки стабильности программы внезапно отключаем COM порт Arduino от компьютера, программа вывела сообщение “COM порт не подключен!”.

					Разработка трехстепенного лабораторного манипулятора с механическим захватом							
Изм.	Лист	№ докум.	Подпись	Дата								
Разработал	Ерёмин Д.О.				Тестирование работы манипулятора				Лит.	Лист	Листов	
Проверил	Юдин Д.А.										68	2
Руководит.	Юдин Д.А.								БГТУ им. В.Г. Шухова МР-41			
Норм. Контр.	Гольцов Ю.А.											
Зав. каф.	Рубанов В.Г.											

робот остался в том же положении в каком и был и никаких действий самостоятельно не производит. После обратного подключения СОМ порта нажимаю кнопку “Поиск СОМ портов” и программа тут же находит этот СОМ порт и подключается к нему, теперь возможно дальнейшее управление манипулятором. При ручном управлении манипулятором по осям координат, рабочая точка передвигается строго вдоль оси, по которой её передвигаю. Все звенья работают синхронно, т.к. для каждого нового движения просчитывается обратная задача кинематики.

При движении манипулятора заметны некоторые небольшие резкие рывки сервоприводов что приводит к колебанию всей конструкции манипулятора. Это происходит из за того, что сервоприводы работают не с достаточно высокой точностью и на малые углы поворота (например в 1 градус) могут не отреагировать. При этом программа Arduino поворачивает сервоприводы с шагом в 0.01 градуса, т.е. с программной точки зрения движение должно происходить плавно.

Далее выполняется проверка программного управления. Подвожу в ручную положение рабочей точки манипулятора в нужную позицию и нажимаю кнопку “Добавить положение”, данная позиция сохраняется. Таким образом создаётся вся программа целиком и добавляю команды задержки в некоторых местах. Нажимаю кнопку “Запуск”, манипулятор начинает движение проходя через все запрограммированные точки и ожидает заданное количество времени на командах задержки, при движении манипулятора есть возможность регулировать скорость его движения в реальном времени.

					<p><i>Тестирование работы манипулятора</i></p>	Лист
						69
Изм.	Лист	№ докум.	Подпись	Дата		

## Заключение

В ходе выполнения работы была построена кинематическая схема трёхстепенного манипулятора, в которой использовался параллельный механизм для того чтобы схват манипулятора оставался параллельно основанию. Работа данной конструкции была смоделирована на динамической модели манипулятора где подтвердилась её работоспособность.

Также была построена 3D модель манипулятора, с учётом всех длин и размеров деталей, для определения её конструктивных ограничений, а именно диапазонов углов поворотов звеньев. Эти углы определили форму рабочей зоны, максимальный радиус которой получился 570мм от центра основания робота. Рабочая зона строилась при помощи динамической модели манипулятора, в которой манипулятор прошёлся через все крайние положения рабочей зоны.

В динамической модели были определены зависимости различных величин таких как: сила, момент, угловая скорость и угловое ускорение, которые действовали на оси поворотов звеньев и основания. Также было смоделировано как вращающий момент, который развивают реальные сервоприводы, действовал на движение звеньев манипулятора. Этот же момент использовался для определения грузоподъёмности манипулятора, которая составила 150 грамм при максимально вытянутой руке манипулятора.

На основе спроектированной трёхмерной модели были изготовлены детали для схвата, основания и ещё некоторые элементы манипулятора при помощи 3D печати из пластика и металлические детали звеньев, которые создавались вручную.

Если возникли вопросы, пишите на почту [den-2d@mail.ru](mailto:den-2d@mail.ru) ☺

					<b>Разработка трехстепенного лабораторного манипулятора с механическим захватом</b>			
Изм.	Лист	№ докум.	Подпись	Дата				
Разработал	Ерёмин Д.О.				<b>Заключение</b>			
Проверил	Юдин Д.А.							
Руководит.	Юдин Д.А.							
Норм. Контр.	Гольцов Ю.А.							
Зав. каф.	Руданов В.Г.							
						Лит.	Лист	Листов
							70	2
						БГТУ им. В.Г. Шухова МР-41		

Была разработана система управления манипулятором на основе платы Arduino Uno, которая получает сигналы управления с компьютера. Компьютерная программа позволяет управлять манипулятором в различных режимах как вручную – управление каждым звеном манипулятора по отдельности или передвижением рабочей точки по осям координат, так и управление в автоматическом режиме, когда программируются различные позиции, по которым должен пройти манипулятор.

Проверка работоспособности манипулятора показала, что робот отрабатывает точно все движения, которые ему задаются в программе. При этом были заметны некоторые небольшие рывки при движении конструкции манипулятора, это происходит из-за особенностей сервоприводов. Тестирование работы манипулятора позволило определить его следующие параметры: повторяемость, которая в среднем составляла 2 мм; реальная грузоподъёмность, равная от 50 до 100 грамм в зависимости от длины плеча, что является меньше чем при моделировании, т.к. реальная масса робота оказалась немного больше ожидаемой; рабочая зона, которая получилась такая же как и при моделировании.

Так же были получены новые знания и навыки построения 3D моделей деталей, работы с 3D принтером, сборки манипулятора, разработки и тестировании его компьютерной динамической модели, разработки системы управления и программировании на языке C#.

Разработанный в данной работе, манипулятор может эффективно использоваться в качестве лабораторного оборудования для изучения задач кинематики и динамики роботов.

					<i><b>Заключение</b></i>	<i>Лист</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		<i>71</i>



## Список литературы

1. Промышленные роботы и манипуляторы [электронный ресурс]. URL: [http://cncnc.ru/documentation/theory\\_of\\_mechanismus\\_and\\_machines/lect\\_19.htm](http://cncnc.ru/documentation/theory_of_mechanismus_and_machines/lect_19.htm) (дата обращения: 09.06.2015).
2. Ключев С.А. Компьютерное моделирование: Учебно-методическое пособие. – М.: Волжский политехнический институт, 2009. - 89 с.
3. Arduino [электронный ресурс]. URL: <https://www.arduino.cc/> (дата обращения: 09.06.2015).
4. Промышленные роботы манипуляторы [электронный ресурс]. URL: <http://robo24.ru/promyshlennye-roboty> (дата обращения: 09.06.2015).
5. Манипуляционные роботы [электронный ресурс]. URL: <http://refleader.ru/jgeyfsrnarna.html> (дата обращения: 10.06.2015).
6. Лекции по УРиПТС [электронный ресурс]. URL: <http://www.studfiles.ru/dir/cat39/subj1292/file13007/view135824/page2.html> (дата обращения: 10.06.2015).
7. Схиртладзе А.Г., Выходец В.И. Оборудование машиностроительных предприятий. – М.: РПК «Политехник», 2005. – 92 с.
8. Компьютерное моделирование [электронный ресурс]. URL: <http://www.infl.info/book/export/html/215> (дата обращения: 8.03.2015).
9. Гонсалес Р., Фу К., Ли К. Робототехника. – М.: Москва «Мир», 1989. – 620с.
10. Adams - MSC Software [электронный ресурс]. URL: <http://www.mssoftware.com/product/adams> (дата обращения: 10.06.2015).
11. Universal Mechanism – моделирование динамики механических систем [электронный ресурс]. URL: <http://www.umlub.ru/pages/index.php?id=1> (дата обращения: 10.06.2015).

					<b>Разработка трехстепенного лабораторного манипулятора с механическим захватом</b>		
Изм.	Лист	№ докум.	Подпись	Дата			
Разработал	Ерёмин Д.О.				<b>Список литературы</b>		
Проверил	Юдин Д.А.						
Руководит.	Юдин Д.А.						
Норм. Контр.	Гольцов Ю.А.						
Зав. каф.	Руданов В.Г.						
						Лит.	Лист
							72
							2
						БГТУ им. В.Г. Шухова	
						МР-41	

- 12.NX: Siemens PLM Software [электронный ресурс]. URL: [http://www.plm.automation.siemens.com/ru\\_ru/products/nx/](http://www.plm.automation.siemens.com/ru_ru/products/nx/) (дата обращения: 10.06.2015).
13. CATIA - Dassault Systèmes [электронный ресурс]. URL: <http://www.3ds.com/ru/produkty-i-uslugi/catia/> (дата обращения: 10.06.2015).
14. SolidWorks-Russia [электронный ресурс]. URL: <http://www.solidworks.ru/> (дата обращения: 10.06.2015).
15. Пособие по ADAMS [электронный ресурс]. URL: [http://mmm\\_samgu.ssu.samara.ru/polyakov/adams/Adams\\_pos\\_new.pdf](http://mmm_samgu.ssu.samara.ru/polyakov/adams/Adams_pos_new.pdf) (дата обращения: 8.03.2015).
16. Fanuc M-410iB Series [электронный ресурс]. URL: [http://www.fanucrobotics.com/cmsmedia/datasheets/M-410iB%20Series\\_15.pdf](http://www.fanucrobotics.com/cmsmedia/datasheets/M-410iB%20Series_15.pdf) (дата обращения: 11.06.2015).
17. Описание сервопривода DF15RMG [электронный ресурс]. Режим доступа: <http://www.electronshtik.ru/item/df15rmg-tilt-kit-20kg-1114314> (дата обращения: 12.06.2015).
18. Описание сервопривода RDS3115 [электронный ресурс]. Режим доступа: <http://www.aliexpress.com/snapshot/6494586405.html?orderId=65771461539714> (дата обращения: 12.06.2015).
19. Описание сервопривода SR430 [электронный ресурс]. Режим доступа: [http://robot-kit.ru/print\\_product\\_info.php/products\\_id/438](http://robot-kit.ru/print_product_info.php/products_id/438) (дата обращения: 12.06.2015).
20. Описание Google SketchUp 8 [электронный ресурс]. Режим доступа: <http://reviewsoft.ru/Windows/Audio-Video/Audio-Recorders/google-sketchup/> (дата обращения: 12.06.2015).
21. Visual Studio — Microsoft Developer Tools [электронный ресурс]. Режим доступа: <https://www.visualstudio.com/ru-ru/visual-studio-homepage-vs.aspx> (дата обращения: 13.06.2015).

					Список литературы	Лист
						73
Изм.	Лист	№ докум.	Подпись	Дата		

## Приложения

Приложение 1: Код Matlab для вычисления математической динамической модели манипулятора .....	75
Приложение 2: Код программы Arduino для управления сервоприводами манипулятора.....	76
Приложение 3: Код компьютерной программы на языке C# для управления манипулятором .....	78
Приложение 4: Презентация .....	93

					<b>Разработка трехстепенного лабораторного манипулятора с механическим захватом</b>		
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>	<b>Приложения</b>		
Разработал	Ерёмин Д.О.						
Проверил	Юдин Д.А.						
Руководит.	Юдин Д.А.						
Норм. Контр.	Гольцов Ю.А.						
Зав. каф.	Рубанов В.Г.						
					<i>Лит.</i>	<i>Лист</i>	<i>Листов</i>
						74	19
					БГТУ им. В.Г. Шухова МР-41		

## Код Matlab для вычисления математической динамической модели манипулятора

```

L1 = 0.2;
L2 = 0.2;
Q1 = pi/4; dQ1 = 0; ddQ1 = 0;
Q2 = pi/4; dQ2 = pi/8; ddQ2 = pi/6;
m1 = 0.21;
m2 = 0.18;
g = 9.80665;

C1 = cos(Q1); S1 = sin(Q1);
C2 = cos(Q1); S2 = sin(Q1);
C12 = cos(Q1+Q2); S12 = sin(Q1+Q2);

A1_0 = [C1 S1 0 L1*C1
        S1 C1 0 L1*S1
        0 0 1 0
        0 0 0 1];
A2_1 = [C2 -S2 0 L2*C2
        S2 C2 0 L2*S2
        0 0 1 0
        0 0 0 1];
A2_0 = A1_0 * A2_1;
MQ = [0 -1 0 0
      1 0 0 0
      0 0 0 0
      0 0 0 0];
U11 = MQ * A1_0;
U21 = MQ * A2_0;
U22 = A1_0 * MQ * A2_1;
J1 = [1/3*m1*L1^2 0 0 -1/2*m1*L1
      0 0 0 0
      0 0 0 0
      -1/2*m1*L1 0 0 m1];
J2 = [1/3*m2*L2^2 0 0 -1/2*m2*L2
      0 0 0 0
      0 0 0 0
      -1/2*m2*L2 0 0 m2];

D11 = 1/3*m1*L1^2 + 4/3*m2*L2^2 + m2*C2*L2^2;
D12 = 1/3*m2*L2^2 + 1/2*m2*L2^2*C2;
D22 = 1/3*m2*L2^2;
h1 = -1/2*m2*S2*L2^2*dQ2^2 - m2*S2*L2^2*dQ1*dQ2;
h2 = 1/2*m2*S2*L2^2*dQ1^2;
c1 = 1/2*m1*g*L1*C1 + 1/2*m2*g*L2*C12 + m2*g*L1*C1;
c2 = 1/2*m2*g*L2*C12;
D = [D11 D12
     D12 D22];
ddQ = [ddQ1
       ddQ2];
h = [h1
     h2];
c = [c1
     c2];

T = D*ddQ + h + c;

```

### Код программы Arduino для управления сервоприводами манипулятора

```
#include "Servo.h"
// Объекты Servo
Servo servo0;
Servo servo1;
Servo servo2;
Servo servo3;
String s="";
int i;
int q0,nq0 = 94; //начальное положение угла серво0
int q1,nq1 = 144; //начальное положение угла серво1
int q2,nq2 = 180; //начальное положение угла серво2
int q3,nq3 = 50; //начальное положение угла серво3
float p0,p1,p2,p3;

void setup()
{
  Serial.begin(9600);
  servo0.attach(6); //серво0 подключен к пин 6
  servo1.attach(8); //серво1 подключен к пин 8
  servo2.attach(10); //серво2 подключен к пин 10
  servo3.attach(12); //серво3 подключен к пин 12
  q0 = nq0;
  q1 = nq1;
  q2 = nq2;
  q3 = nq3;
  servo0.write(q0); // Повернуть серво0 на q0 градусов
  servo1.write(q1); // Повернуть серво1 на q1 градусов
  servo2.write(q2); // Повернуть серво2 на q2 градусов
  servo3.write(q3); // Повернуть серво3 на q3 градусов
  p0=q0;
  p1=q1;
  p2=q2;
  p3=q3;
}

void loop()
{
  while (Serial.available())
  { //считывание данных с COM порта
    char c = Serial.read(); //читаем символ
    s += c; //добавляем к строке
    if (c == '!') //если считали символ, строка закончилась
    {
      i=0;
      q0=0;
      q1=0;
      q2=0;
      q3=0;
      while (s[i] != ' ')
      {
        q0=q0*10;
        q0=q0+(int(s[i])-48);
        i++;
      }
      i++;
      while (s[i] != ' ')
      {
        q1=q1*10;
        q1=q1+(int(s[i])-48);
        i++;
      }
    }
  }
}
```

```

    }
    i++;
    while (s[i] != ' ')
    {
        q2=q2*10;
        q2=q2+(int(s[i])-48);
        i++;
    }
    i++;
    while (s[i] != '!')
    {
        q3=q3*10;
        q3=q3+(int(s[i])-48);
        i++;
    }
    s = "";
}
}
//повороты сервоприводов
if (p0<q0)
{
    p0=p0+0.01;
    servo0.write(p0); // Повернуть серво0 на p0 градусов
}
if (p0>q0)
{
    p0=p0-0.01;
    servo0.write(p0); // Повернуть серво0 на p0 градусов
}
if (p1<q1)
{
    p1=p1+0.01;
    servo1.write(p1); // Повернуть серво1 на p1 градусов
}
if (p1>q1)
{
    p1=p1-0.01;
    servo1.write(p1); // Повернуть серво1 на p1 градусов
}
if (p2<q2)
{
    p2=p2+0.01;
    servo2.write(p2); // Повернуть серво2 на p2 градусов
}
if (p2>q2)
{
    p2=p2-0.01;
    servo2.write(p2); // Повернуть серво2 на p2 градусов
}
if (p3<q3)
{
    p3=p3+0.01;
    servo3.write(p3); // Повернуть серво3 на p3 градусов
}
if (p3>q3)
{
    p3=p3-0.01;
    servo3.write(p3); // Повернуть серво2 на p2 градусов
}
delay(0.01);
}

```

## Код компьютерной программы на языке C# для управления манипулятором

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.IO.Ports; //подключение пространства имен с классом SerialPort
using System.IO;

namespace RobotProgram
{
    public partial class Form1 : Form
    {
        double q0 = 0, q1 = 69, q2 = 38, q3 = 50; //значения начальных углов поворотов
звеньев
        int s0 = 1, s1 = 1, s2 = 1, s3 = 1; //скорости поворотов звеньев (изменять не
нужно)
        //длины звеньев, мм
        int L0 = 90; //основание
        int L1 = 200; //1ое звено
        int L2 = 200; //2ое звено
        int L3 = 170; //схват
        //переменные координат и углов звеньев
        double X, Y, Z, Xm, Ym, Zm, qq0, qq1, qq2;
        //флаги нажатия кнопок управления
        bool f0U = false, f0D = false,
            f1U = false, f1D = false,
            f2U = false, f2D = false,
            f3U = false, f3D = false;

        bool flag = true;
        bool flag1 = false;
        bool start_flag = false;
        bool gotovo = true;
        bool f_time = false;
        int time1 = 0;
        string str_pos = "";
        string str_pos_1 = "";
        string str1 = "";
        string str_x = "";
        string str_y = "";
        string str_z = "";
        string str_s = "";
        string str_t = "";
        int xk, yk, zk, sk, tk;

        double k = 57.295779513;
        double q3r;
        string stroka = "";

        SerialPort Port1;
        public Form1()
        {
            InitializeComponent();
        }
    }
}
```

```

private void Form1_Load(object sender, EventArgs e)
{
    string[] myPort; //создаем массив строк
    myPort = System.IO.Ports.SerialPort.GetPortNames(); // в массив помещаем
доступные порты
    comboBox1.Items.AddRange(myPort); //теперь этот массив заносим в
список(comboBox)
    if (comboBox1.Items.Count != 0) comboBox1.SelectedIndex = 0; //если есть
порты выбираем 1ый
}

private void Form1_FormClosing(object sender, FormClosingEventArgs e)
{
    if (comboBox1.Items.Count != 0) //если есть доступный COM порт
    {
        if (Port1.IsOpen != false)
        {
            Port1.Close(); //при закрытии программы закрываем порт (если он
открыт)
        }
    }
}

private void button1_Click(object sender, EventArgs e) { }

private void button2_Click(object sender, EventArgs e) { }

////управление роботом////
////ручное управление////
private void timer1_Tick(object sender, EventArgs e)
{
    if (angles.Checked) //если выбрано управление углами
    {
        label1.Text = "Q0";
        label2.Text = "Q1";
        label3.Text = "Q2";
        //определение значений скоростей регулировки углов
        //s0 = Q0TB.Value; //скорость движения манипулятора
        s0 = 1;
        s1 = s0;
        s2 = s0;
        s3 = 1;
        //настройка скорости
        if ((q3r * k + s1) > 0)
        {
            s1 = (int)Math.Abs(Math.Round(q3r * k));
            s2 = s1;
            if (s1 == 0)
            {
                s1 = 1;
                s2 = 1;
            }
        }
    }
    //управление углами
    if (f0U)
    {
        q0 = q0 + s0; //увеличение q0
        if (q0 > 85) q0 = 85;
    }
    if (f0D)
    {
        q0 = q0 - s0; //уменьшение q0
        if (q0 < -85) q0 = -85;
    }
}

```



```

    }
    Q0T.Text = q0.ToString(); //вывод значения угла q0

    if (f1U)
    {
        q1 = q1 + s1; //увеличение q1
        if (q1 > 102) q1 = 102;
    }
    if (f1D && ((q3r * k) < 0))
    {
        q1 = q1 - s1; //уменьшение q1
        if (q1 < -42) q1 = -42;
    }
    Q1T.Text = q1.ToString(); //вывод значения угла q1

    if (f2U)
    {
        q2 = q2 + s2; //увеличение q2
        if (q2 > 48) q2 = 48;
    }
    if (f2D && ((q3r * k) < 0))
    {
        q2 = q2 - s2; //уменьшение q2
        if (q2 < -90) q2 = -90;
    }
    Q2T.Text = q2.ToString(); //вывод значения угла q2

    if (f3U)
    {
        q3 = q3 + s3; //увеличение q3
        if (q3 > 90) q3 = 90;
    }
    if (f3D)
    {
        q3 = q3 - s3; //уменьшение q3
        if (q3 < 0) q3 = 0;
    }
    Q3T.Text = q3.ToString(); //вывод значения угла q3
    //корректировка q1 и q2
    if ((q3r * k) > 0)
    {
        if (q1 < 0) q1 = q1 + s0;
        if (q2 < 0) q2 = q2 + s0;
    }

    pryamaya_zadacha(); //решение прямой задачи
    //присвоение координат
    X = Xm;
    Y = Ym;
    Z = Zm;
}
if (coordinates.Checked) //если выбрано управление координатами
{
    label1.Text = "X";
    label2.Text = "Y";
    label3.Text = "Z";
    //скорость движения манипулятора
    s0 = 1;
    s3 = 1;
    //настройка скорости
    if ((q3r * k + s1) > 0)
    {
        s0 = (int)Math.Abs(Math.Round(q3r * k));
    }
}

```

```

        if (s0 == 0) s0 = 1;
    }
    //управление координатами
    if (f0U && !((flag == false) && (X > 0)))
    {
        X = X + s0; //увеличение X
    }
    if (f0D)
    {
        X = X - s0; //уменьшение X
    }
    Q0T.Text = X.ToString(); //вывод значения X

    if (f1U && !((flag == false) && (Y > 0)))
    {
        Y = Y + s0; //увеличение Y
    }
    if (f1D && !((flag == false) && (Y < 0)))
    {
        Y = Y - s0; //уменьшение Y
    }
    Q1T.Text = Y.ToString(); //вывод значения Y

    if (f2U && ((q3r * k) < 0) && !((flag == false) && (Z > 0)))
    {
        Z = Z + s0; //увеличение Z
    }
    if (f2D && !((flag == false) && (Z < 0)))
    {
        Z = Z - s0; //уменьшение Z
    }
    Q2T.Text = Z.ToString(); //вывод значения угла Z
    //управление схватом
    if (f3U)
    {
        q3 = q3 + s3; //разжимать схват
        if (q3 > 90) q3 = 90;
    }
    if (f3D)
    {
        q3 = q3 - s3; //сжимать схват
        if (q3 < 0) q3 = 0;
    }
    Q3T.Text = q3.ToString(); //вывод значения угла q3

    obratnaya_zadacha(); //решение обратной задачи

    if (double.IsNaN(qq0) || double.IsNaN(qq1) || double.IsNaN(qq2))
    { //если при решении обратной задачи значение угла будет NaN, значит
    это уже недостижимая зона
        label9.ForeColor = Color.Red;
        label9.Text = "Достигнут край рабочей зоны!";
        flag = false; //ставим флаг, чтобы робот дальше не двигался
    }
    else
    {
        q0 = qq0;
        q1 = qq1;
        q2 = qq2;
        flag = true;
    }
    korrektirovka(); //корректировка углов поворотов звеньев
}
//вывод различных значений на экран

```

```

label10.Text = "X = " + Math.Round(X).ToString();
label11.Text = "Y = " + Math.Round(Y).ToString();
label12.Text = "Z = " + Math.Round(Z).ToString();

label13.Text = "Q0 = " + q0.ToString();
label14.Text = "Q1 = " + q1.ToString();
label15.Text = "Q2 = " + q2.ToString();
label19.Text = "Схват = " + q3.ToString();

if (start_flag) Start.Text = "Стон";
else Start.Text = "Запуск";
if ((gotovo == false) && (start_flag == true)) action();

if (start_flag)
{
    groupBox1.Enabled = false;
    angles.Checked = true;
}
else groupBox1.Enabled = true;

////программное управление////
if (start_flag && radioButton1.Checked) //одноразовое выполнение программы
{
    if (listBox1.Items.Count > 0) //если есть строки
    {
        try
        {
            string st = listBox1.Items[listBox1.SelectedIndex].ToString();
//если нет выделенных строк
        }
        catch
        {
            listBox1.SetSelected(0, true); //выделяем 1ую строку
        }
        if ((listBox1.Items[listBox1.SelectedIndex].ToString().Length > 0) &&
gotovo) //если есть строка
        {
            str_pos = listBox1.Items[listBox1.SelectedIndex].ToString();
//считывание строки-команды
            if (String.Equals(str_pos, str_pos_1) == false) gotovo = false;
//если поступила новая команда, ставим флаг на выполнение команды
        }
        if (gotovo) //если команда выполнена
        {
            if (listBox1.Items.Count > (listBox1.SelectedIndex + 1))
listBox1.SetSelected(listBox1.SelectedIndex + 1, true); //переключаемся на следующую
строку-команду
            else start_flag = false; //если строк-команд нет, программа
завершена
        }
    }
    else start_flag = false; //если строк-команд нет, программа завершена
}

if (start_flag && radioButton2.Checked) //выполнение в цикле
{
    if (listBox1.Items.Count > 0) //если есть строки
    {
        try
        {
            string st = listBox1.Items[listBox1.SelectedIndex].ToString();
//если нет выделенных строк
        }
        catch

```

```

        {
            listBox1.SetSelected(0, true); //выделяем 1ую строку
        }
        if ((listBox1.Items[listBox1.SelectedIndex].ToString().Length > 0) &&
gotovo) //если есть строка
        {
            str_pos = listBox1.Items[listBox1.SelectedIndex].ToString();
//считывание строки-команды
            if (String.Equals(str_pos, str_pos_1) == false) gotovo = false;
//если поступила новая команда, ставим флаг на выполнение команды
        }
        if (gotovo) //если команда выполнена
        {
            if (listBox1.Items.Count > (listBox1.SelectedIndex + 1))
listBox1.SetSelected(listBox1.SelectedIndex + 1, true); //переключаемся на следующую
строку-команду
            else listBox1.SetSelected(0, true); //если строк-команд нет,
переходим в начало
        }
    }
    else start_flag = false; //если строк-команд нет, программа завершена
}

if (start_flag && radioButton3.Checked) //пошаговое выполнение
{
    if (listBox1.Items.Count > 0) //если есть строки
    {
        try
        {
            string st = listBox1.Items[listBox1.SelectedIndex].ToString();
//если нет выделенных строк
        }
        catch
        {
            listBox1.SetSelected(0, true); //выделяем 1ую строку
        }
        if ((listBox1.Items.Count > (listBox1.SelectedIndex + 1))) //если
есть следующая строка
        {
            if ((listBox1.Items[listBox1.SelectedIndex + 1].ToString().Length
> 0) && gotovo && (flag1==false)) //если есть строка
            {
                if (flag1 == false)
listBox1.SetSelected(listBox1.SelectedIndex + 1, true); //переключаемся на следующую
строку-команду
                str_pos = listBox1.Items[listBox1.SelectedIndex].ToString();
//считывание строки-команды
                gotovo = false; //ставим флаг на выполнение команды
                flag1 = true; //ставим флаг, что начали выполнение программы
            }
            if (gotovo && flag1) //если команда выполнена
            {
                flag1 = false;
                start_flag = false; //программа завершена
            }
        }
    }
    else
    {
        start_flag = false; //если строк-команд нет, программа завершена
    }
}
else start_flag = false; //если строк-команд нет, программа завершена
}
////конец программного управления////

```

```

//вывод информации о состоянии программы
if (start_flag)
{
    label20.Text = "Выполнение программы";
    if (String.Equals(str1, "P")) //если это команда позиции
    {
        label21.Text = "Движение к позиции";
        label22.Text = "X=" + str_x + " Y=" + str_y + " Z=" + str_z;
        label23.Text = "Схват=" + str_s;
    }
    if (String.Equals(str1, "D")) //если это команда задержки
    {
        label21.Text = "Ожидание";
        label22.Text = "time="+time1.ToString();
        label23.Text = "";
    }
}
else
{
    label20.Text = "";
    label21.Text = "";
    label22.Text = "";
    label23.Text = "";
}
}
/////конец управления роботом/////

private void korrektirovka() //корректировка положения углов
{
    bool f1 = false;

    if (q0 > 85)
    {
        q0 = 85;
        f1 = true;
    }
    if (q0 < -85)
    {
        q0 = -85;
        f1 = true;
    }
    if (q1 > 102)
    {
        q1 = 102;
        f1 = true;
    }
    if (q1 < -42)
    {
        q1 = -42;
        f1 = true;
    }
    if (q2 > 48)
    {
        q2 = 48;
        f1 = true;
    }
    if (q2 < -90)
    {
        q2 = -90;
        f1 = true;
    }

    if ((q3r * k) > 0)

```

```

    {
        if (q1 < 0) q1 = q1 + s0;
        if (q2 < 0) q2 = q2 + s0;
        f1 = true;
    }

    pryamaya_zadacha();
    if (f1)
    {
        X = Xm;
        Y = Ym;
        Z = Zm;
        f1 = false;
    }
}

private void pryamaya_zadacha()
{
    //---Прямая_задача---//
    k = 57.295779513;
    double q0r = q0 / k;
    double q1r = q1 / k;
    double q2r = q2 / k;
    q3r = -(q1r + q2r);

    double sin_q0 = Math.Sin(q0r);
    double sin_q1 = Math.Sin(q1r);
    double sin_q2 = Math.Sin(q2r);
    double sin_q3 = Math.Sin(q3r);

    double cos_q0 = Math.Cos(q0r);
    double cos_q1 = Math.Cos(q1r);
    double cos_q2 = Math.Cos(q2r);
    double cos_q3 = Math.Cos(q3r);

    Xm = L2 * (cos_q0 * cos_q1 * cos_q2 - cos_q0 * sin_q1 * sin_q2) + L3 *
(cos_q3 * (cos_q0 * cos_q1 * cos_q2 - cos_q0 * sin_q1 * sin_q2) - sin_q3 * (cos_q0 *
cos_q1 * sin_q2 + cos_q0 * cos_q2 * sin_q1)) + L1 * cos_q0 * sin_q1;
    Ym = L2 * (cos_q1 * cos_q2 * sin_q0 - sin_q0 * sin_q1 * sin_q2) + L3 *
(cos_q3 * (cos_q1 * cos_q2 * sin_q0 - sin_q0 * sin_q1 * sin_q2) - sin_q3 * (cos_q1 *
sin_q0 * sin_q2 + cos_q2 * sin_q0 * sin_q1)) + L1 * sin_q0 * sin_q1;
    Zm = L0 - L2 * (cos_q1 * sin_q2 + cos_q2 * sin_q1) - L3 * (cos_q3 * (cos_q1 *
sin_q2 + cos_q2 * sin_q1) + sin_q3 * (cos_q1 * cos_q2 - sin_q1 * sin_q2)) + L1 * cos_q1;

    Xm = Math.Round(Xm);
    Ym = Math.Round(Ym);
    Zm = Math.Round(Zm);
}

private void obratnaya_zadacha()
{
    //---Обратная_задача---//
    double xx = Math.Sqrt(X * X + Y * Y);
    double x1 = xx - L3;
    double z1 = Z - L0;
    double B = Math.Sqrt(x1 * x1 + z1 * z1);

    qq0 = Math.Asin(Y / xx);
    qq1 = 1.570796327 - Math.Acos((L1 * L1 + B * B - L2 * L2) / (2 * B * L2)) -
Math.Asin(z1 / B);
    qq2 = 1.570796327 - Math.Acos((L1 * L1 + L2 * L2 - B * B) / (2 * L1 * L2));

    qq0 = Math.Round(qq0 * k);

```

```

        qq1 = Math.Round(qq1 * k);
        qq2 = Math.Round(qq2 * k);
    }

    //обработчики нажатия кнопок управления
    private void Q0U_MouseDown(object sender, MouseEventArgs e)
    {
        f0U = true;
    }

    private void Q0U_MouseUp(object sender, MouseEventArgs e)
    {
        f0U = false;
    }

    private void Q0D_MouseDown(object sender, MouseEventArgs e)
    {
        f0D = true;
    }

    private void Q0D_MouseUp(object sender, MouseEventArgs e)
    {
        f0D = false;
    }

    private void Q1U_MouseDown(object sender, MouseEventArgs e)
    {
        f1U = true;
    }

    private void Q1U_MouseUp(object sender, MouseEventArgs e)
    {
        f1U = false;
    }

    private void Q1D_MouseDown(object sender, MouseEventArgs e)
    {
        f1D = true;
    }

    private void Q1D_MouseUp(object sender, MouseEventArgs e)
    {
        f1D = false;
    }

    private void Q2U_MouseDown(object sender, MouseEventArgs e)
    {
        f2U = true;
    }

    private void Q2U_MouseUp(object sender, MouseEventArgs e)
    {
        f2U = false;
    }

    private void Q2D_MouseDown(object sender, MouseEventArgs e)
    {
        f2D = true;
    }

    private void Q2D_MouseUp(object sender, MouseEventArgs e)
    {

```

```

        f2D = false;
    }

    private void Q3U_MouseDown(object sender, MouseEventArgs e)
    {
        f3U = true;
    }

    private void Q3U_MouseUp(object sender, MouseEventArgs e)
    {
        f3U = false;
    }

    private void Q3D_MouseDown(object sender, MouseEventArgs e)
    {
        f3D = true;
    }

    private void Q3D_MouseUp(object sender, MouseEventArgs e)
    {
        f3D = false;
    }

    private void trackBar1_Scroll(object sender, EventArgs e)
    {
        //регулировка задержки таймера1
        //отвечает за скорость скорость перемещения робота
        int period1 = trackBar1.Value;
        period1 = (101 - period1);
        timer1.Interval = period1; //уровень задержки таймера
        labelTimer.Text = (101 - period1).ToString()+"%";
    }

    private void trackBar2_Scroll(object sender, EventArgs e)
    {
        //регулировка задержки таймера2
        //отвечает за частоту отправки команд управления
        int period2 = trackBar2.Value;
        double period21 = 1000 / period2;
        period21 = Math.Round(period21);
        timer2.Interval = (int)period21; //уровень задержки таймера
        label8.Text = period2.ToString() + " команд/сек";
    }

    private void comboBox1_SelectedIndexChanged(object sender, EventArgs e)
    {
        //выбирает доступный COM порт, если он есть
        if (comboBox1.SelectedItem.ToString() != "") Port1 = new
SerialPort(comboBox1.SelectedItem.ToString(), 9600);
    }

    private void Poisk_Click(object sender, EventArgs e)
    {
        //поиск доступных COM портов
        string[] myPort; //создаем массив строк
        myPort = System.IO.Ports.SerialPort.GetPortNames(); // в массив помещаем
доступные порты
        comboBox1.Items.Clear(); //очистка comboBox
        comboBox1.Items.AddRange(myPort); //теперь этот массив заносим в
список(comboBox)
        if (comboBox1.Items.Count != 0) comboBox1.SelectedIndex = 0;
    }

    private void timer2_Tick(object sender, EventArgs e)
    {

```



```

//---Отправка_команды_на_Arduino---//
//преобразование углов для сервоприводов
double q0s = Math.Round(94 - q0);
double q1s = Math.Round(102 - q1);
double q2s = Math.Round(132 + q2);
string komanda = ""; //очистка строки

if (System.IO.Ports.SerialPort.GetPortNames().Length == 0)
comboBox1.Items.Clear(); //если нет портов очистка comboBox

if (double.IsNaN(q0) || double.IsNaN(q1) || double.IsNaN(q2) ||
double.IsNaN(X) || double.IsNaN(Y) || double.IsNaN(Z))
{
    label9.ForeColor = Color.Red;
    label9.Text = "Выход из рабочей зоны!";
}
else
{
    komanda = q0s.ToString() + " " + q1s.ToString() + " " + q2s.ToString() +
" " + q3.ToString() + "!"; //создание строки значений углов поворотов звеньев
    textBox1.Text = komanda;

    if (komanda != stroka)
    {
        if (comboBox1.Items.Count != 0) //если есть доступный COM порт
        {
            label9.ForeColor = Color.LimeGreen;
            if (flag) label9.Text = "COM порт подключен!";
            try
            {
                Port1 = new SerialPort(comboBox1.SelectedItem.ToString(),
9600);

                Port1.Open(); //открываем порт
                Port1.Write(komanda); //отправка строки значений углов
                Port1.Close(); // закрываем порт
            }
            catch (Exception)
            {
                label9.ForeColor = Color.Red;
                if (flag) label9.Text = "Ошибка подключения по COM порту!";
            }
        }
        else
        {
            label9.ForeColor = Color.Red;
            if (flag) label9.Text = "COM порт не подключен!";
        }
    }
    stroka = komanda;
    //---Конец_отправки---//
}
private void button1_Click_1(object sender, EventArgs e)
{
    string position = "P " + X + " " + Y + " " + Z + " " + q3;
    //добавление в список позицию робота
    if ((listBox1.Items.Count == 0) || (listBox1.SelectedIndex ==
(listBox1.Items.Count - 1)))
    {
        listBox1.Items.Add(position);
        listBox1.SetSelected(listBox1.Items.Count - 1, true);
    }
    else

```

```

        {
            listBox1.Items.Insert(listBox1.SelectedIndex + 1, position);
        }
    }
    private void button2_Click_1(object sender, EventArgs e)
    {
        string delay = "D " + textBox2.Text;
        //добавление в список позицию робота
        if ((listBox1.Items.Count == 0) || (listBox1.SelectedIndex ==
(listBox1.Items.Count - 1)))
        {
            listBox1.Items.Add(delay);
            listBox1.SetSelected(listBox1.Items.Count - 1, true);
        }
        else
        {
            listBox1.Items.Insert(listBox1.SelectedIndex + 1, delay);
        }
    }
    private void button3_Click(object sender, EventArgs e)
    {
        //удаление выбранной строки
        int p = listBox1.SelectedIndex;
        if (p >= 0) listBox1.Items.RemoveAt(p);
        if (p - 1 >= 0) listBox1.SetSelected(p - 1, true);
        if ((p == 0) && (listBox1.Items.Count >= 1)) listBox1.SetSelected(p, true);
    }
    private void button4_Click(object sender, EventArgs e)
    {
        listBox1.Items.Clear(); //очистить поле
    }
    private void button6_Click(object sender, EventArgs e)
    {
        //сохранение текста программы движения робота
        Stream myStream;
        SaveFileDialog saveFileDialog1 = new SaveFileDialog();
        saveFileDialog1.Filter = "txt files (*.txt)|*.txt";
        saveFileDialog1.FilterIndex = 2;
        saveFileDialog1.RestoreDirectory = true;
        if (saveFileDialog1.ShowDialog() == DialogResult.OK)
        {
            if ((myStream = saveFileDialog1.OpenFile()) != null)
            {
                StreamWriter sw = new StreamWriter(myStream);
                for (int i = 0; i < listBox1.Items.Count; i++)
                {
                    listBox1.SelectedIndex = i;
                    sw.WriteLine(listBox1.SelectedItem.ToString());
                }
                sw.Close();
                myStream.Close();
            }
        }
    }
    private void button7_Click(object sender, EventArgs e)
    {
        //загрузка текста программы движения робота
        OpenFileDialog openFileDialog = new OpenFileDialog();
        string line;
        if (openFileDialog.ShowDialog() == DialogResult.OK)
        {
            listBox1.Items.Clear(); //очистить поле
            string fileName = openFileDialog.FileName;
            FileStream sss = File.Open(fileName, FileMode.Open, FileAccess.Read);

```

```

        if (sss != null)
        {
            StreamReader reader = new StreamReader(sss);
            while ((line = reader.ReadLine()) != null)
                listBox1.Items.Add(line);
            sss.Close();
        }
    }
}
private void action()
{
    //выполнение команды программы
    if (String.Equals(str_pos, str_pos_1)==false) //если поступила новая команда
    {
        str_pos_1 = str_pos;
        str1 = "";
        str_x = "";
        str_y = "";
        str_z = "";
        str_s = "";
        str_t = "";
        int i = 0;
        //считывание команды
        if (str_pos.Length > 0)
        {
            while ((i < str_pos.Length) && (Char.Equals(str_pos[i], ' ') ==
false))
            {
                str1 = str1 + str_pos[i];
                i = i + 1;
            }
            if (String.Equals(str1, "P") == true) //если это команда позиции
            {
                i = i + 1;
                while ((i < str_pos.Length) && (Char.Equals(str_pos[i], ' ') ==
false))
                {
                    str_x = str_x + str_pos[i];
                    i = i + 1;
                }
                i = i + 1;
                while ((i < str_pos.Length) && (Char.Equals(str_pos[i], ' ') ==
false))
                {
                    str_y = str_y + str_pos[i];
                    i = i + 1;
                }
                i = i + 1;
                while ((i < str_pos.Length) && (Char.Equals(str_pos[i], ' ') ==
false))
                {
                    str_z = str_z + str_pos[i];
                    i = i + 1;
                }
                i = i + 1;
                while ((i < str_pos.Length) && (Char.Equals(str_pos[i], ' ') ==
false))
                {
                    str_s = str_s + str_pos[i];
                    i = i + 1;
                }
                xk = int.Parse(str_x);
                yk = int.Parse(str_y);
                zk = int.Parse(str_z);
            }
        }
    }
}

```

```

        sk = int.Parse(str_s);
        ///////////////////////////////////
        //---Обратная_задача---//
        double xx = Math.Sqrt(xk * xk + yk * yk);
        double x1 = xx - L3;
        double z1 = zk - L0;
        double B = Math.Sqrt(x1 * x1 + z1 * z1);
        qq0 = Math.Asin(yk / xx);
        qq1 = 1.570796327 - Math.Acos((L1 * L1 + B * B - L2 * L2) / (2 *
B * L2)) - Math.Asin(z1 / B);
        qq2 = 1.570796327 - Math.Acos((L1 * L1 + L2 * L2 - B * B) / (2 *
L1 * L2));

        qq0 = Math.Round(qq0 * k);
        qq1 = Math.Round(qq1 * k);
        qq2 = Math.Round(qq2 * k);
        ///////////////////////////////////
    }
    if (String.Equals(str1, "D")) //если это команда задержки
    {
        i = i + 1;
        while ((i < str_pos.Length) && (Char.Equals(str_pos[i], ' ') ==
false))

        {
            str_t = str_t + str_pos[i];
            i = i + 1;
        }
        tk = int.Parse(str_t);
    }
}

//выполнение команды
if (String.Equals(str1, "P")) //если это команда позиции
{
    if (q0 < qq0)
    {
        q0 = q0 + 1;
        if (q0 > 85) q0 = 85;
    }
    if (q0 > qq0)
    {
        q0 = q0 - 1;
        if (q0 < -85) q0 = -85;
    }
    //Q0T.Text = q0.ToString(); //вывод значения угла q0
    if (q1 < qq1)
    {
        q1 = q1 + 1;
        if (q1 > 102) q1 = 102;
    }
    if ((q1 > qq1) && ((q3r * k) < 0))
    {
        q1 = q1 - 1;
        if (q1 < -42) q1 = -42;
    }
    //Q1T.Text = q1.ToString(); //вывод значения угла q1
    if (q2 < qq2)
    {
        q2 = q2 + 1;
        if (q2 > 48) q2 = 48;
    }
    if ((q2 > qq2) && ((q3r * k) < 0))
    {
        q2 = q2 - 1;
    }
}

```

```

        if (q2 < -90) q2 = -90;
    }
    //Q2T.Text = q2.ToString(); //вывод значения угла q2
    if (q3 < sk)
    {
        q3 = q3 + 1;
        if (q3 > 90) q3 = 90;
    }
    if (q3 > sk)
    {
        q3 = q3 - 1;
        if (q3 < 0) q3 = 0;
    }
    //Q3T.Text = q3.ToString(); //вывод значения угла q3
    if ((q3r * k) > 0)
    {
        if (q1 < 0) q1 = q1 + 1;
        if (q2 < 0) q2 = q2 + 1;
    }
    pryamaya_zadacha();
    X = Xm;
    Y = Ym;
    Z = Zm;
    //если все углы достигнуты, команда выполнена
    if ((q0 == qq0) && (q1 == qq1) && (q2 == qq2) && (q3 == sk)) gotovo =
true;
    }
    if (String.Equals(str1, "D")) //если это команда задержки
    {
        if (time1 == 0) time1 = tk*10;
        f_time = true;
    }
}
private void Start_Click(object sender, EventArgs e)
{
    if (start_flag == false) start_flag = true;
    else start_flag = false;
}
private void timer3_Tick(object sender, EventArgs e)
{
    if (f_time && start_flag)
    {
        if (time1 == 1)
        {
            f_time = false;
            gotovo = true;
        }
        if (time1 > 0) time1 = time1 - 1;
    }
}
private void textBox2_KeyPress(object sender, KeyPressEventArgs e)
{
    //ограничение на ввод только цифр
    if (Char.IsDigit(e.KeyChar) || (e.KeyChar == 8))
    {
    }
    else
    {
        e.Handled = true;
    }
}
}
}
}

```



Тема дипломной работы:

# Разработка трехстепенного лабораторного манипулятора с механическим захватом

Презентация

Выполнил:

студент группы МР-41

Ерёмин Д.О.

Дипломный руководитель:

к.т.н. Юдин Д.А.



# Цель и задачи

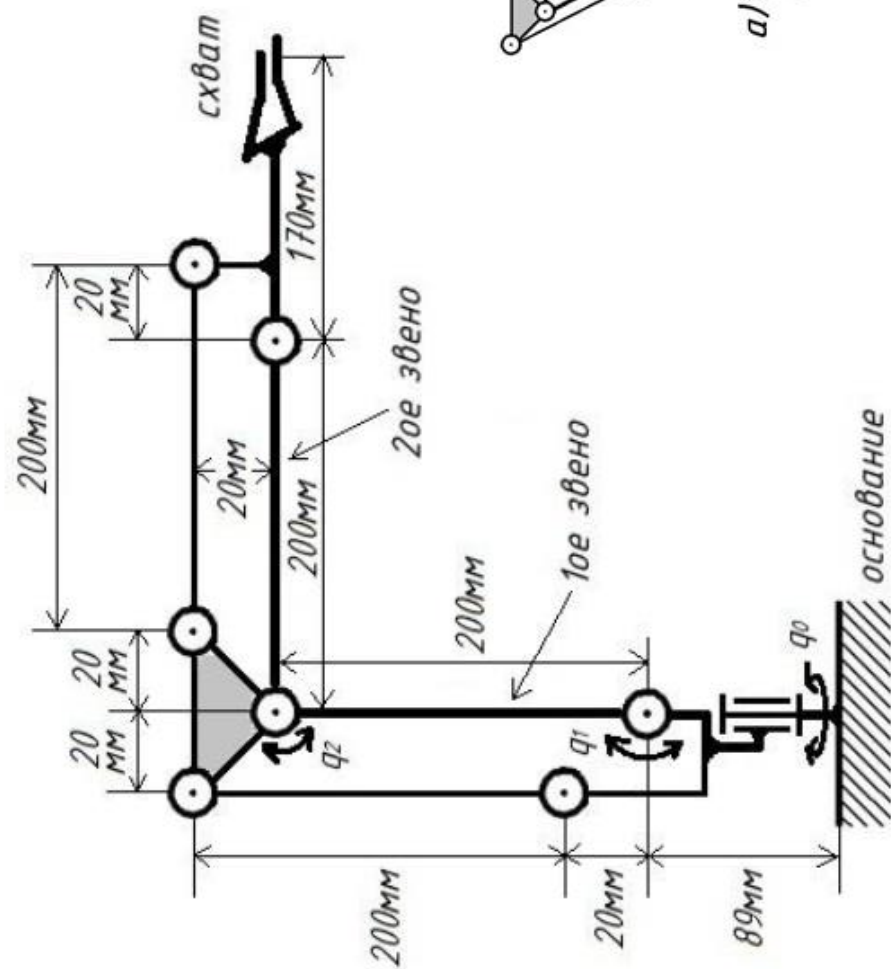
## Цель:

Разработка трехстепенного лабораторного манипулятора с механическим захватом для исследования методов управления и моделирования манипуляционных робототехнических систем

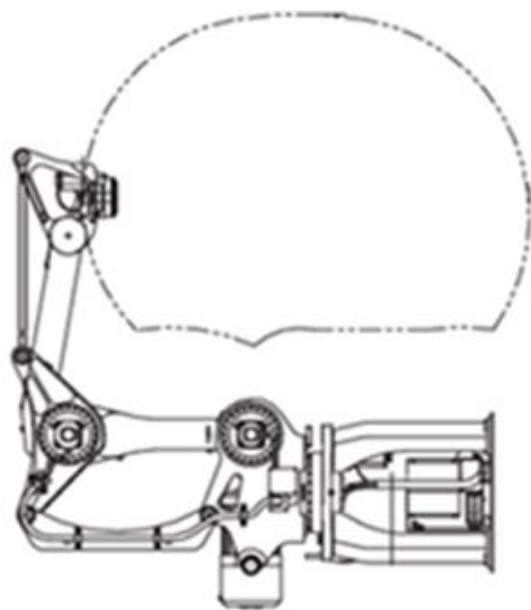
## Задачи:

- Разработать манипулятор, который должен обладать следующими свойствами:
  - Иметь три степени подвижности
  - Длины звеньев должны составлять по 20 см
  - Наличие механического схвата
- Разработать трёхмерную модель конструкции манипулятора
- Построить компьютерную динамическую модель манипулятора
- Разработать компьютерную программу для управления манипулятором

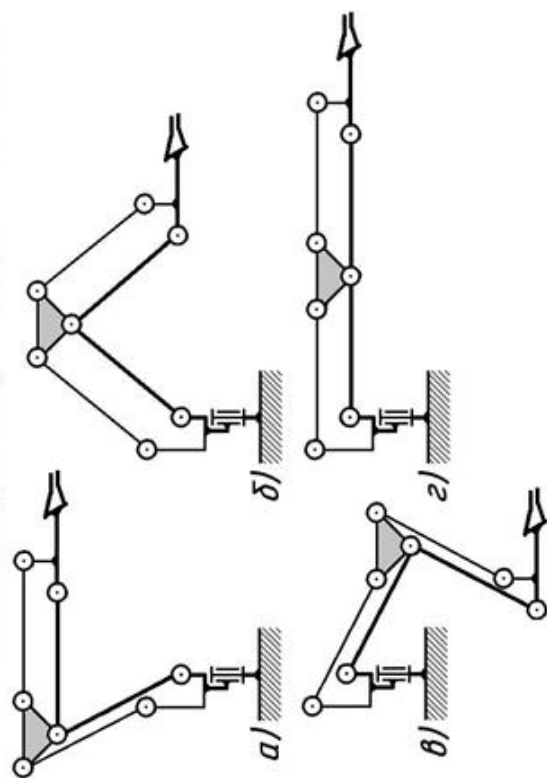
# Кинематическая схема манипулятора



Кинематическая схема манипулятора



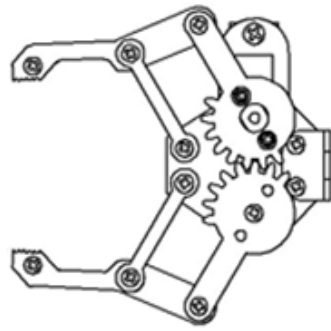
Манипулятор Fanuc M-410iB



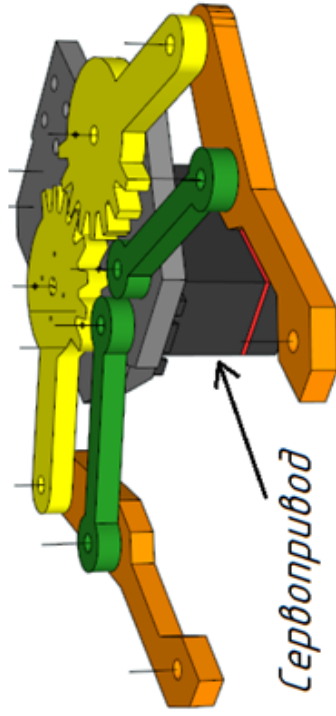
Различные положения



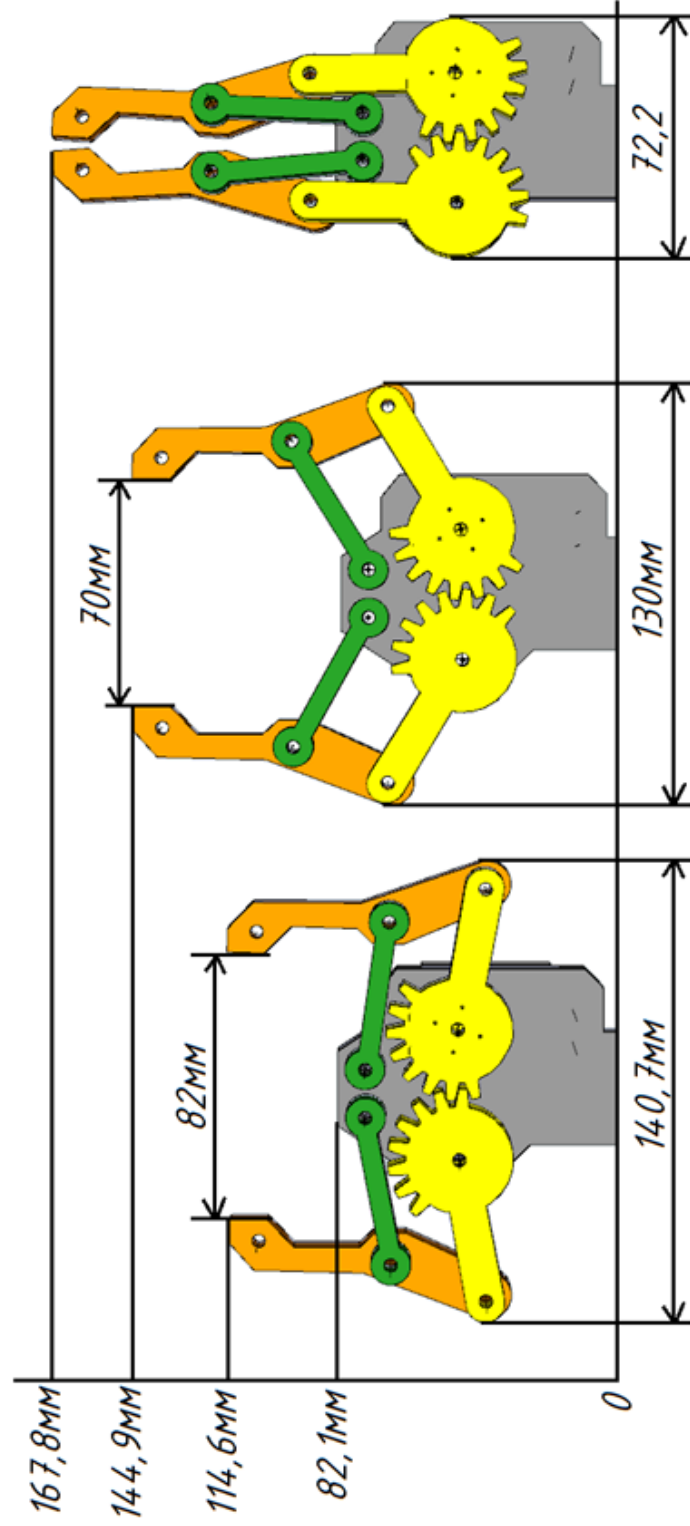
# Проектирование схвата манипулятора



Чертеж

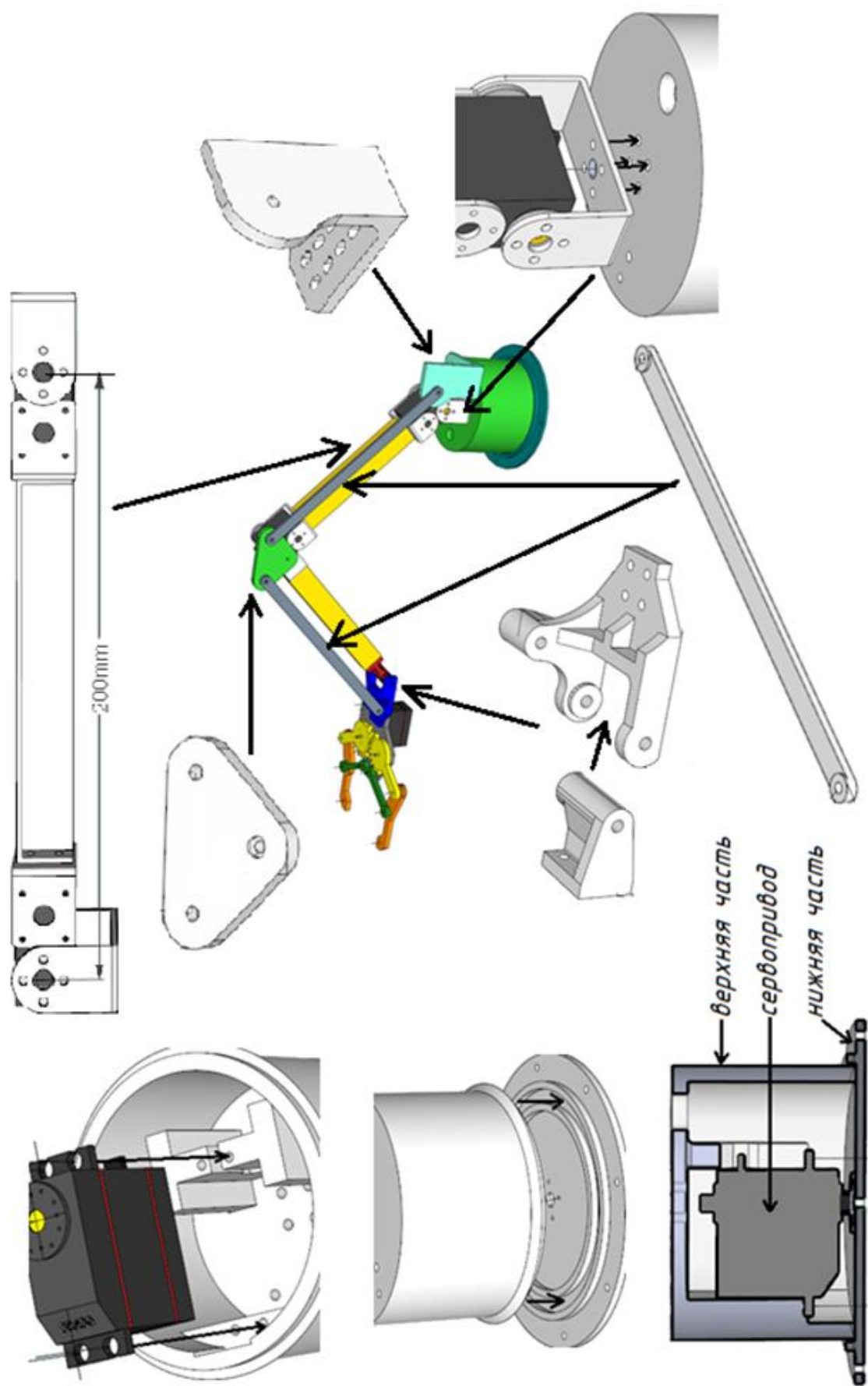


3D модель схвата



Схват при различных положениях

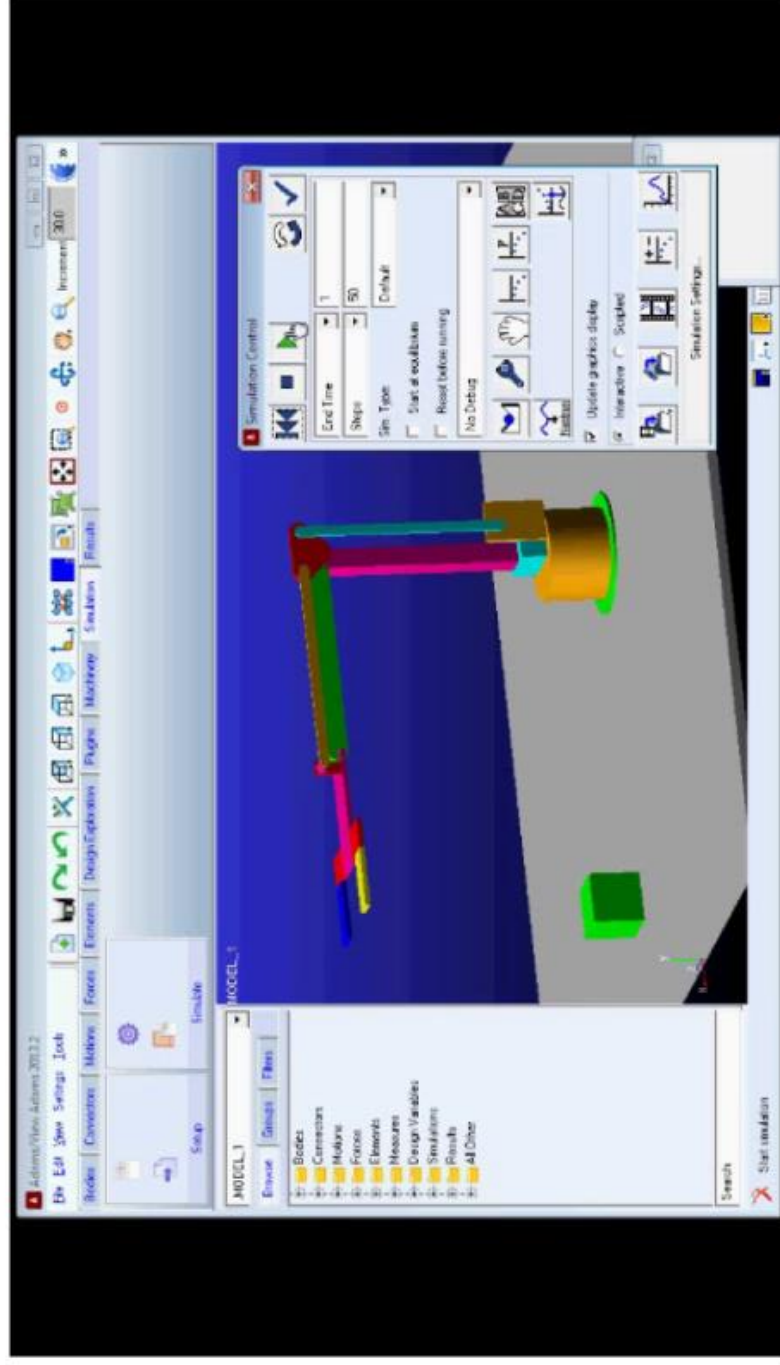
# Проектирование деталей манипулятора



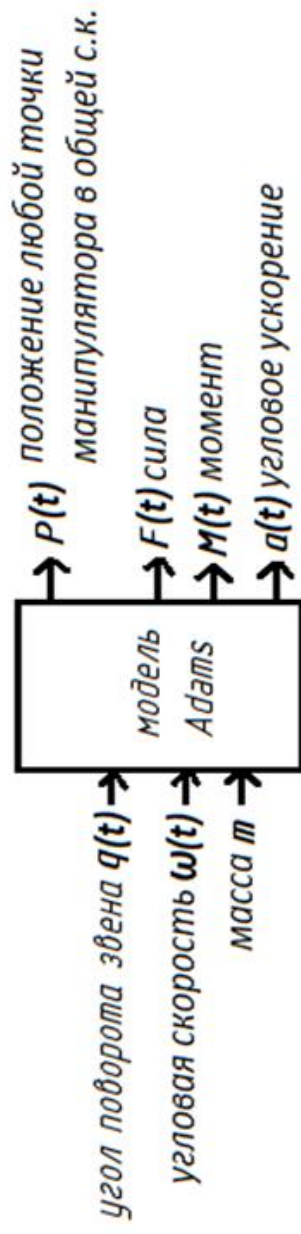
Модели деталей манипулятора

# Компьютерная динамическая модель манипулятора

Демонстрация  
работы  
компьютерной  
модели

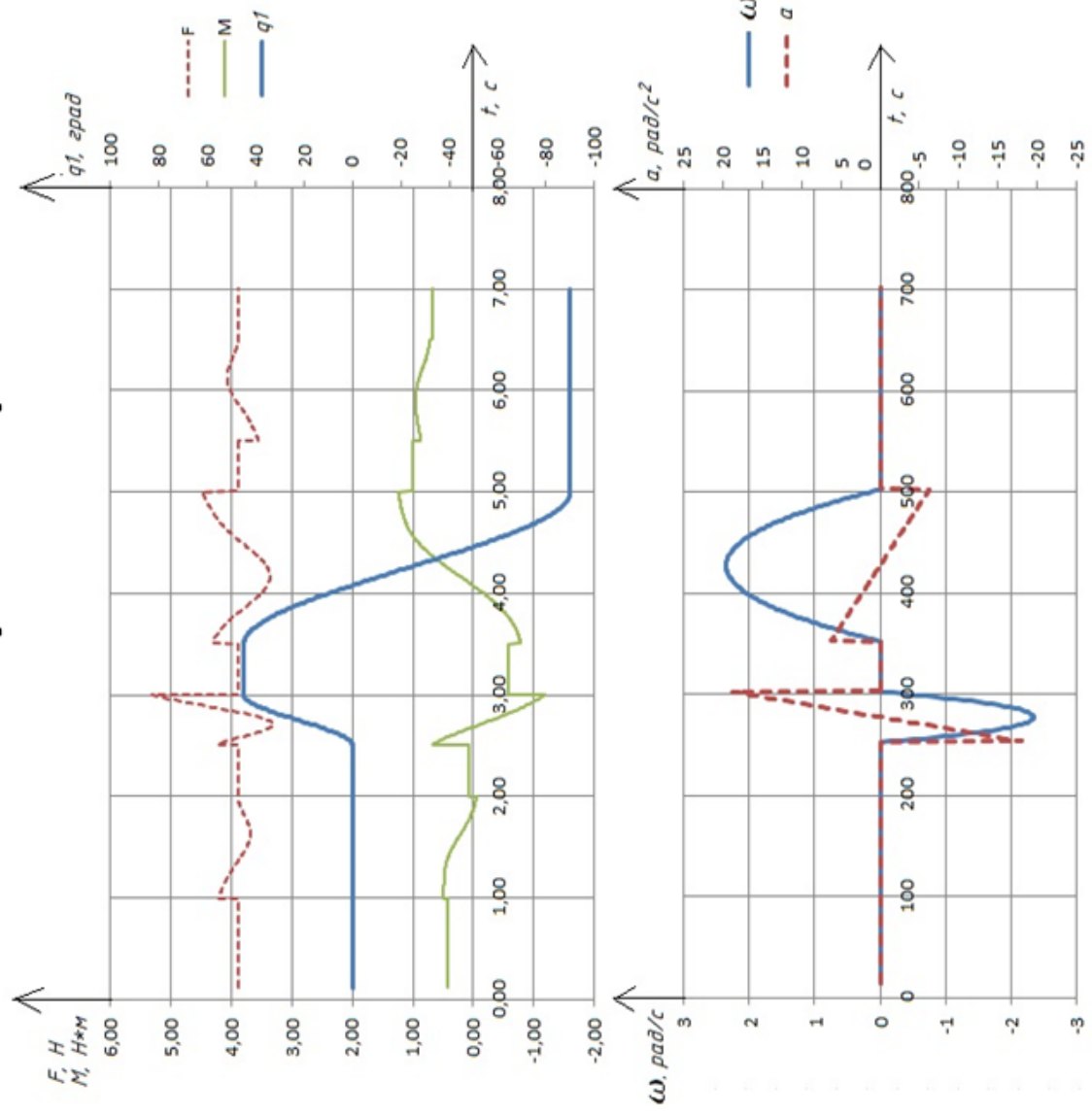


Входные и  
выходные  
данные в модели



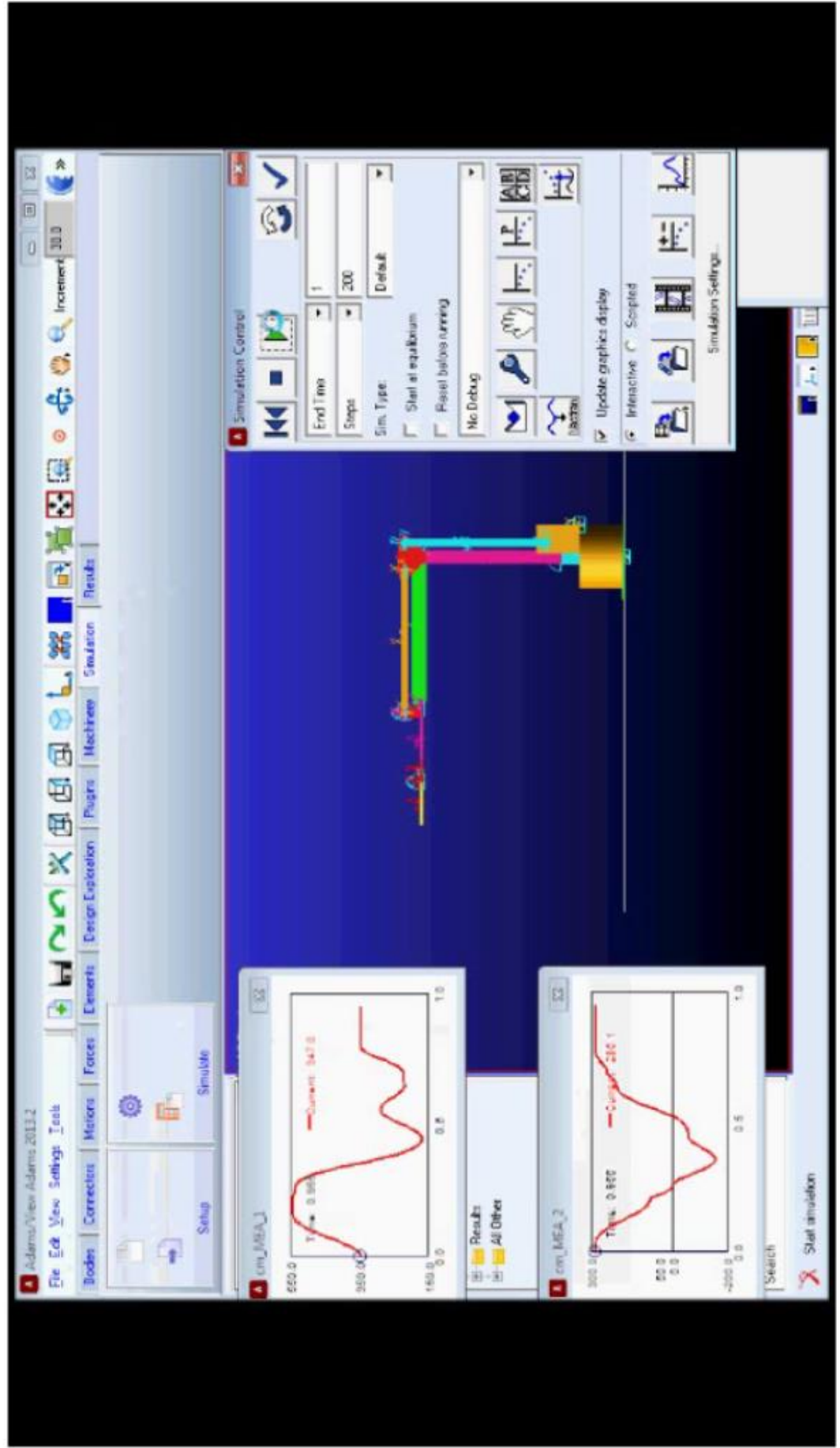
# Динамические характеристики манипулятора

Результат  
вычисления  
компьютерной  
модели для 1-го  
звена  
манипулятора

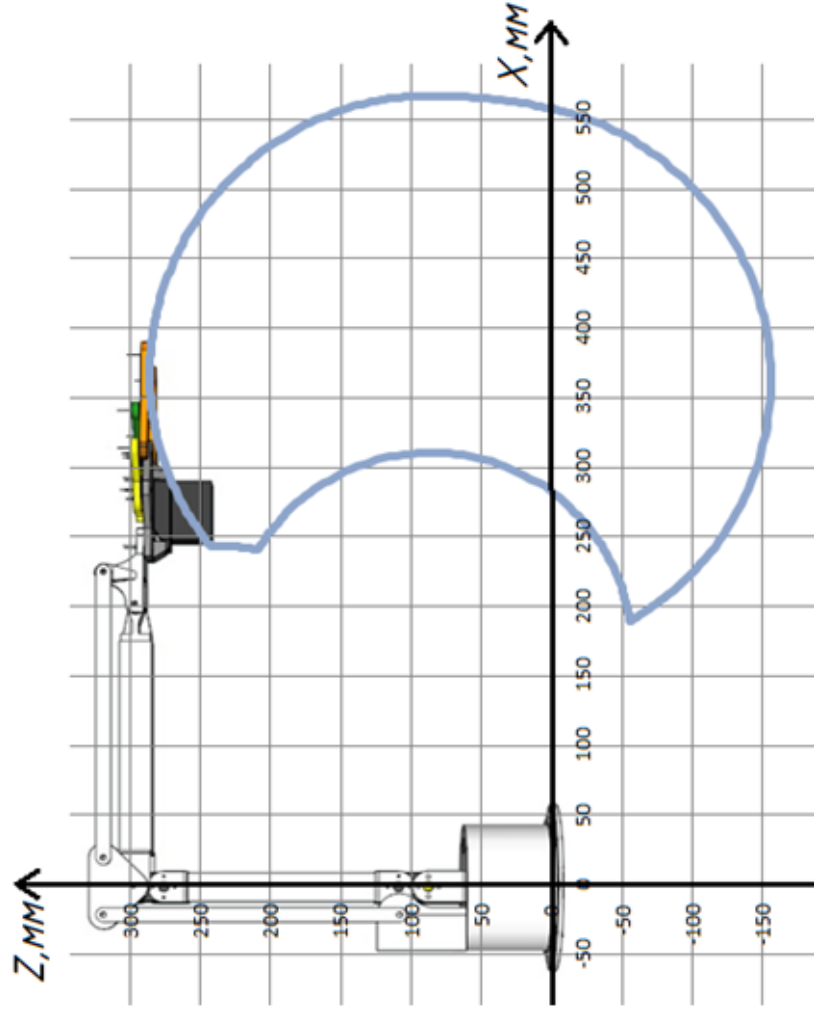




# Определение рабочей зоны манипулятора



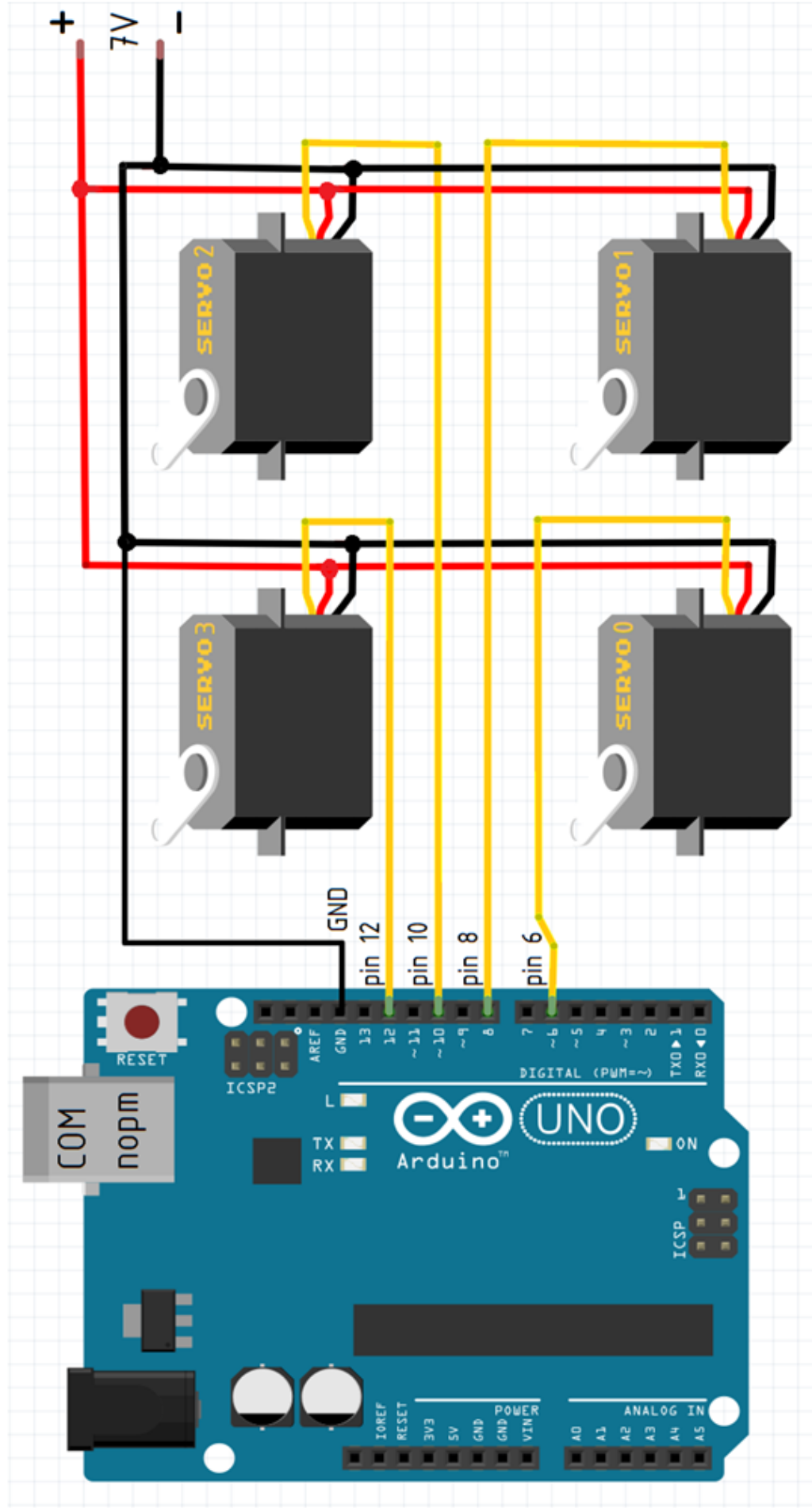
# Характеристики манипулятора



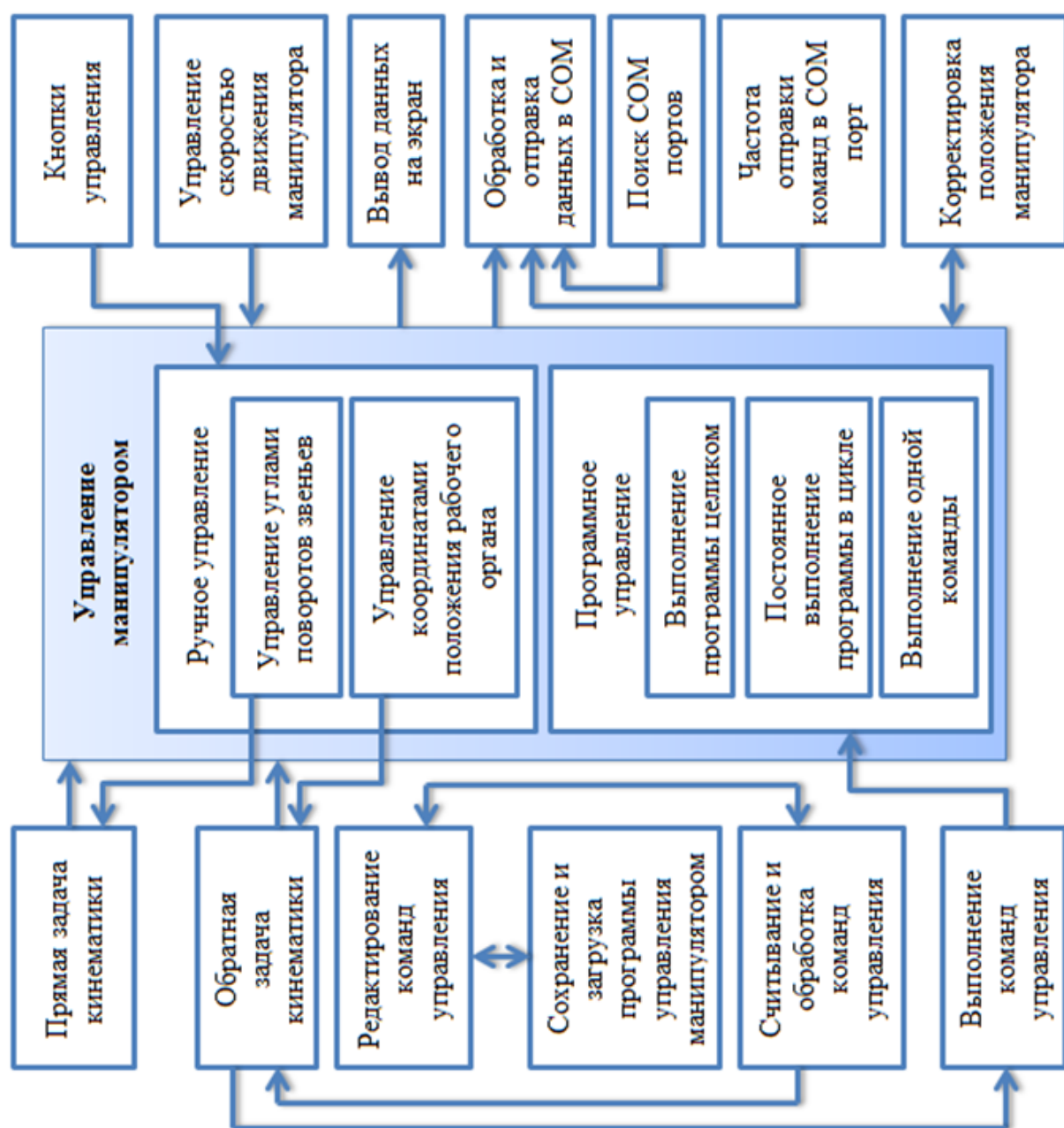
Повторяемость: 2мм  
Радиус действия: 570мм  
Диапазон перемещения: 180 град.  
Грузоподъёмность: 50 – 100 грамм  
Контролируемые оси: 3  
Вес: 775 грамм  
Монтажное положение: на полу

Рабочая зона

# Аппаратная реализация системы управления на основе Arduino



# Программа управления манипулятором



Упрощённая структура программы управления манипулятором



# Демонстрация работы манипулятора

