|  |  |
| --- | --- |
| Gerb-BMSTU_01 | Министерство науки и высшего образования Российской Федерации  Федеральное государственное бюджетное образовательное учреждение  высшего образования  «Московский государственный технический университет  имени Н.Э. Баумана  (национальный исследовательский университет)»  (МГТУ им. Н.Э. Баумана) |
| ФАКУЛЬТЕТ «СПЕЦИАЛЬНОЕ МАШИНОСТРОЕНИЕ»  КАФЕДРА «РОБОТОТЕХНИЧЕСКИЕ СИСТЕМЫ И МЕХАТРОНИКА»  РАСЧЕТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА  К ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЕ  НА ТЕМУ:  «Роботизированная система сортировки корреспонденции на почте»  Студент СМ7-84Б Д.З. Хан  (Подпись, дата)  Руководитель ВКР Е.А. Котов  (Подпись, дата)  Нормоконтролер В.А. Панков  (Подпись, дата) | |
| 2023 г. | |

РЕФЕРАТ

Расчетно-пояснительная записка 93 с., 59 рис., 2 табл., 12 источн., 3 прил.

Объектом исследования являются промышленный дельта робот для сортировки писем.

Цель работы – проектирование робота – манипулятора для сортировки писем с внедрением его в сортировочную линию АСЦ компании «Почта России».

Поставленная цель достигается путем исследования существующих решений сортировочных линий на базе робота-манипулятора, выявления их преимуществ и недостатков и создание робота, в соответствии с необходимой скоростью работы комплекса сортировки.

Были выделены следующие задачи, решаемые в рамках ВКР:

1. Разработка алгоритма работы РТК
2. Определение параметров манипулятора
3. Создание 3D-моделей звеньев манипулятора
4. Разработка системы управления приводами робота
5. Разработка системы управления манипулятора
6. Разработка алгоритмов определения положения письма
7. Моделирование работы системы

СОДЕРЖАНИЕ

[РЕФЕРАТ 2](#_Toc137949446)

[ВВЕДЕНИЕ 5](#_Toc137949447)

[1 Актуальность темы работы 9](#_Toc137949448)

[2 Анализ существующих решений 10](#_Toc137949449)

[3 Постановка задач проекта 13](#_Toc137949450)

[4 Технические требования 14](#_Toc137949451)

[5 Исследовательская часть 16](#_Toc137949452)

[5.1 Основные этапы работы системы 16](#_Toc137949453)

[5.2 Разработка структуры системы управления РТК 18](#_Toc137949454)

[5.3 Определение основных параметров манипулятора 20](#_Toc137949455)

[5.3.1 Выбор кинематической схемы 20](#_Toc137949456)

[5.3.2 Решение прямой и обратной задачи кинематики по положению 22](#_Toc137949457)

[5.3.3 Метрический синтез в соответствии с требуемой рабочей зоной 30](#_Toc137949458)

[5.3.4 Решение прямой и обратной задачи кинематики по скороси 36](#_Toc137949459)

[5.3.5 Определение максимального требуемого момента 38](#_Toc137949460)

[6. Конструкторская часть 40](#_Toc137949461)

[6.1 Энергетический расчет привода манипулятора 40](#_Toc137949462)

[6.2 Регулировочный расчет привода манипулятора 42](#_Toc137949463)

[6.2.1 Исходные данные 42](#_Toc137949464)

[6.2.2 Расчет контура тока 44](#_Toc137949465)

[6.2.3 Расчет контура скорости с внутренним контуром тока 46](#_Toc137949466)

[6.2.4 Расчет контура положения в позиционном режиме 51](#_Toc137949467)

[6.3 Создание моделей звеньев манипулятора 55](#_Toc137949468)

[7. Использование алгоритмов технического зрения 59](#_Toc137949469)

[8. Моделирование работы РТК 61](#_Toc137949470)

[ЗАКЛЮЧЕНИЕ 66](#_Toc137949471)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 67](#_Toc137949472)

[ПРИЛОЖЕНИЕ А 69](#_Toc137949473)

[ПРИЛОЖЕНИЕ Б 86](#_Toc137949474)

[ПРИЛОЖЕНИЕ В Графическая часть дипломного проекта 89](#_Toc137949475)

ВВЕДЕНИЕ

Несмотря на стремительное развитие технологий по передаче информации, почтовая корреспонденция все еще играет значительную роль в современной системе сообщений. По официальным данным предприятия «Почта России» на 2018 год количество бумажных отправлений составляет более 25 млн. писем [1]. Поэтому организация почтовых отправлений все еще остается одним из важных направлений деятельности фирмы, требующим большое количество времени и персонала. В настоящее время процесс доставки различных почтовых отправлений не может осуществляться без внедрения автоматизации на различных этапах. Примером современного центра для сортировки различных почтовых отправлений является Московский АСЦ (Рисунок 1).



Рисунок 1 – Московский АСЦ

Сортировочный центр является местом, в которое стекаются почтовые отправления со всех отделений близлежащих регионов, так как сортировка на местах является куда менее эффективной. Процесс сортировки бумажных писем в сортировочном центре состоит из нескольких этапов:

1. Разделение писем на группы на основе размера письма;
2. Формирование правильно ориентированных стопок писем;
3. Лицевание и штемпелевание писем;
4. Обработка писем на автоматической сортировочной машине;
5. Отправка отсортированных писем в места назначения.

На первом этапе бумажные отправления механическим способом отделяются друг от друга по размерам и распределяются в разные отделы для дальнейшей сортировки. Размеры писем и конвертов нормируются по ГОСТ Р 51506-99 Конверты почтовые. Технические требования. Методы контроля [2]. Габаритные размеры конвертов для внутренних почтовых отправлений приведены на Рисунке 2. Оформление лицевой стороны конверта для внутренних почтовых оформлений приведены на Рисунке 3.

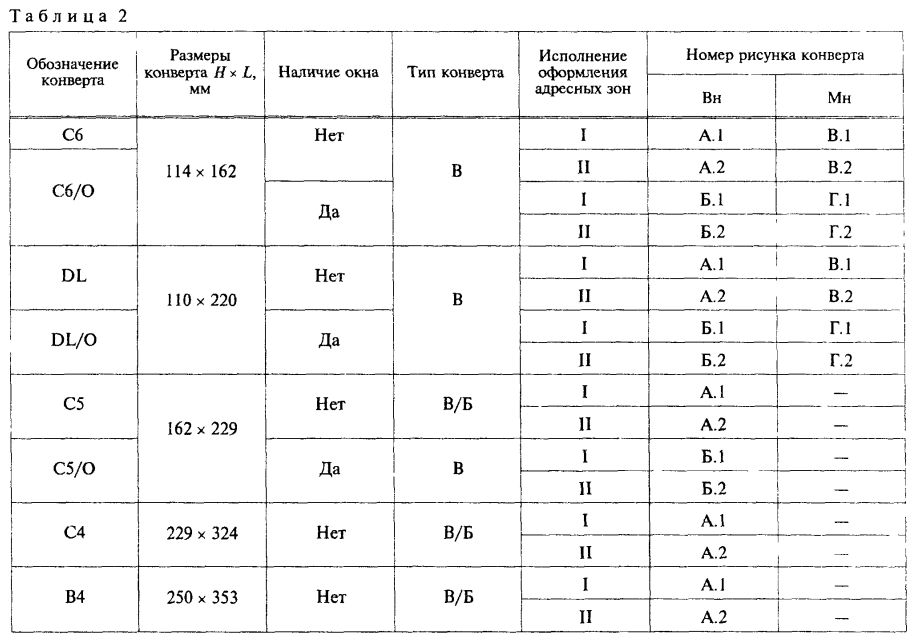


Рисунок 2 – Габаритные размеры писем для внутреннего сообщения

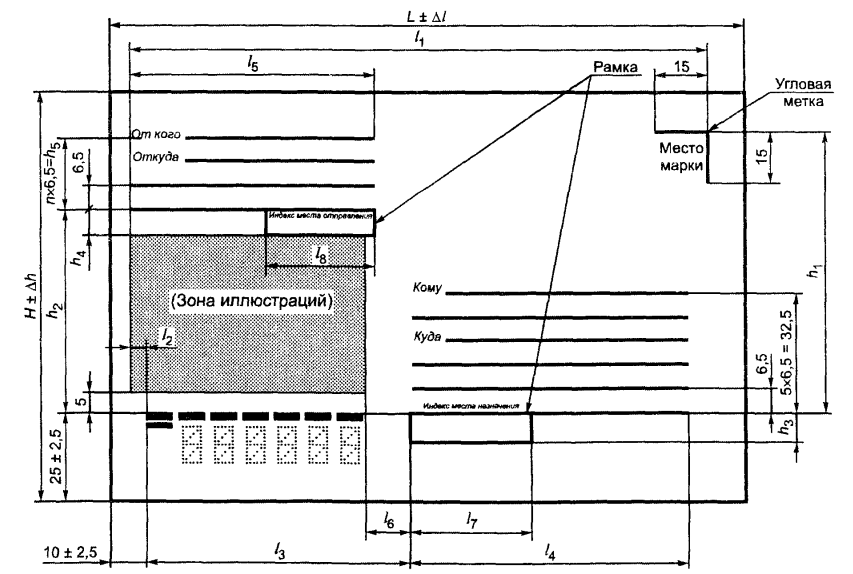
На втором этапе письма одного размера вручную ориентируются в одну сторону и формируются в стопки для дальнейшей сортировки. 

Рисунок 3 – Оформление лицевой стороны письма

На третьем этапе на специальных машинах происходит нанесение штемпелей и специальных меток для сортировки писем по адресам на специальном оборудовании.

На четвертом этапе стопки с письмами загружаются в сортировочную машину, которая способна за короткое время определить адрес назначения письма и отправить его в нужный выход. Распознавание писем происходит с помощью системы технического зрения. Внешний вид сортировочной машины приведен на Рисунке 4.



Рисунок 4 – Внешний вид машины для сортировки писем

Из всех этапов сортировки наиболее зависимым от человека является этап формирования стопок писем, поэтому именно этот этап подлежит рассмотрению и автоматизации в рамках данной работы.

# 1 **Актуальность** темы работы

Хотя в наше время бумажные письма практически исчезли из повседневной жизни, сотрудникам почтовых служб все равно приходится сталкиваться за год с обработкой миллионов различных писем, которые необходимо доставить по нужному адресу. Процесс сортировки посылок хорошо алгоритмизуется, он не состоит из особо сложных операций, которые может выполнить только человек, в основном это простые задачи типа перемещения, чтения адреса, нанесения необходимых марок и правильного позиционирования. Поэтому задачи, связанные с сортировкой посылок и, в частности, сортировкой писем, необходимо автоматизировать для увеличения производительности труда и освобождение рабочих от монотонной работы.

Все этапы сортировки писем, на которых введена автоматизация процесса, выполняются за довольно короткое время, в частности установка для распознавания адресов писем и дальнейшего их распределения способна за час обработать до 12 тыс. входящих писем. На основе характеристики сортирующей машины строится весь процесс сортировки писем на АСЦ, но для обеспечения такой скорости обработки входящих писем необходимо привлекать большое количество людей на этапе формирования правильно ориентированной стопки писем. Именно по этой причине в работе рассматривается вопрос автоматизации процесса формирования из потока неупорядоченных писем ориентированных стопок, которые могут без дальнейшей подготовки сразу быть направлены на машину для распознавания адресов назначения писем.

# 2 Анализ существующих решений

Задачи типа pick and place являются стандартными, широко данный задачи встречаются в пищевой промышленности, сборке электроники, пакетировании. Поэтому многие фирмы, производящие роботов, предоставляют готовые решения для операций похожего рода.

Одним из лидеров в сфере производства промышленных роботов является компания ABB. Одной из разработок данной компании является система для упаковки кондитерских изделий на основе камеры компьютерного зрения и 3-х подвижного робота IRB 360 Flexpicker (Рисунок 5).



Рисунок 5 – Робот ABB IRB 360 Flexpicker

Также свое решение для данного типа задач на основе камеры и дельта – робота имеет фирма Fanuc. На основе 6-осевого робота M-1iA реализована система сортировки конфет (Рисунок 6).

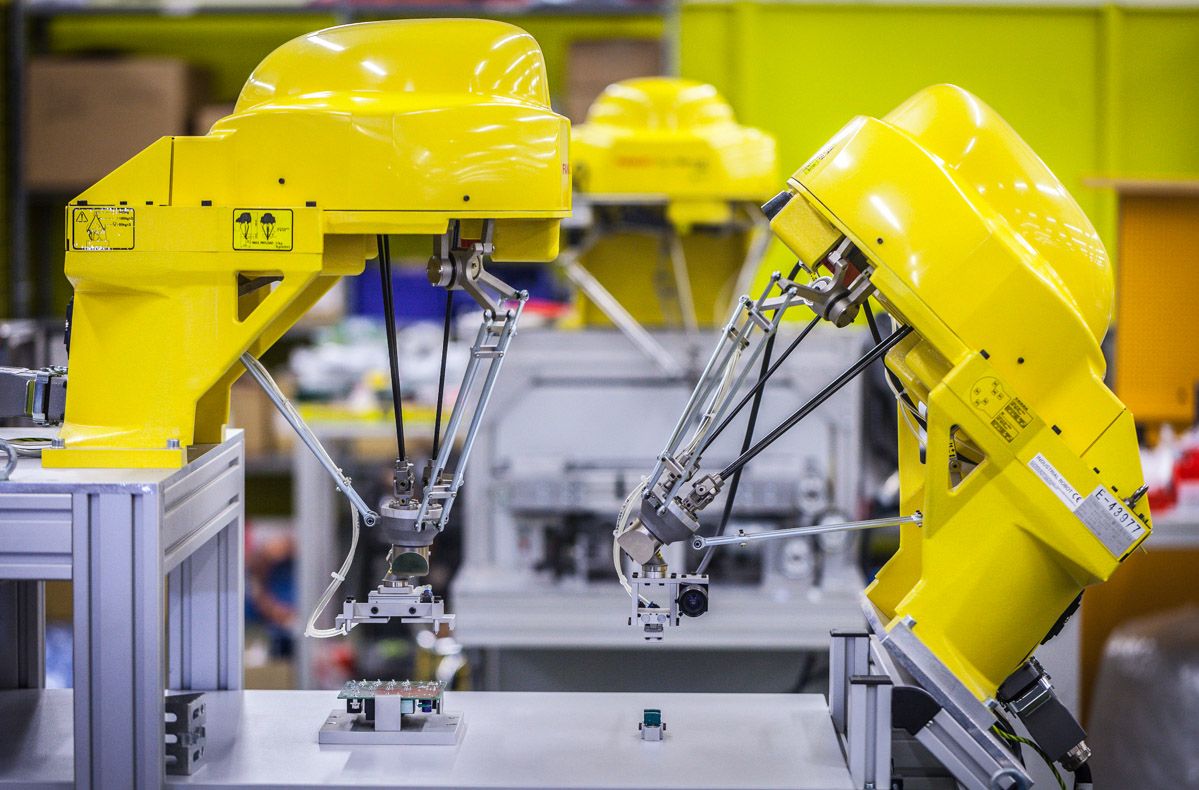


Рисунок 6 – Робот Fanuc M-1iA

Решение задачи по сортировке посылок было рассмотрено фирмой Comitas (Рисунок 7). Они предложили решение на основе группы мобильных роботов, которые одновременное перемещаются по рабочей области(столу), на которой расположены корзины для различных адресов отправления. На основе технического зрения и специальных меток на посылках система управления определяет адрес назначения посылки, оператор подносит посылку к камере, кладет посылку на свободную мобильную платформу, которая после подтверждения о получении посылки начинает движение. Данная система хорошо подстраивается под различные виды посылок, также она пригодна и для сортировки писем, но она не обеспечивает достаточной скорости обработки писем, поэтому решение заменять существующую систему обработки писем проект фирмы Comitas не является выгодным и практически оправданным в рамках поставленной задачи.

На основе рассмотренных решений можно сделать вывод, что для операций типа pick and place для объектов небольшой массы для увеличения скорости операций перемещения чаще всего используются дельта-роботы различной конфигурации, поэтому система сортировки писем будет строиться на основе группы дельта-роботов.



Рисунок 7 - Система сортировки посылок COMITAS

# 3 Постановка задач проекта

Целью проекта является проектирование РТК для сортировки писем, который может быть внедрен на АСЦ компании «Почта России».

Необходимость использования данного робота заключается в увеличении скорости сортировки писем и замене людского труда при сортировке.

Для достижения поставленной цели необходимо решить определенные задачи:

1. Разработка алгоритма работы РТК

2. Определение параметров манипулятора

3. Создание 3D-моделей звеньев манипулятора

4. Разработка системы управления приводами робота

5. Разработка системы управления манипулятора

6. Разработка алгоритмов определения положения письма

7. Моделирование работы системы

# 4 Технические требования

1. Размеры РТК не должны превышать:
   1. по длине –
   2. по высоте – ;
   3. по ширине – .
2. Количество писем, которое необходимо обработать за час одним роботом: не менее 3 тысячи писем (это значение связано с производительностью сортировочной машины)
3. Точность позиционирования схвата робота не менее 2 мм
4. Все расчеты вести для стандартных писем и конвертов для внутренних почтовых отправлений DL.
5. Рабочая зона манипулятора должна представлять собой комбинацию цилиндра и усеченного конуса. Размеры:
   1. Диаметр цилиндра ;
   2. Высота цилиндра ;
   3. Диаметр нижней окружности конуса ;
   4. Высота усеченного конуса ;
   5. Координата дна цилиндра рабочей зоны в системе координат, связанной с неподвижным основанием робота ;

В соответствии с требуемой скоростью обработки писем необходимо вычислить скорость рабочего органа и ускорение рабочего органа в зависимости от требуемой скорости сортировки писем.

Скорость сортировки писем вычислю в соответствии с производительностью сортировочной машины:

Количество циклов перемещения, которые робот должен совершить для перемещения писем и возврата в начальное положение .

Расстояние, которое робот должен пройти рабочий орган для перемещения письма с конвейерной ленты до ящика .

Тогда время, за которое робот выполнит один цикл перемещения:

Пусть участок равномерного движения робота длится 0.3 секунды, а разгон и торможение осуществляется 0.15 секунды.

Изменение скорости рабочего органа от времени представлено на Рисунке 10.

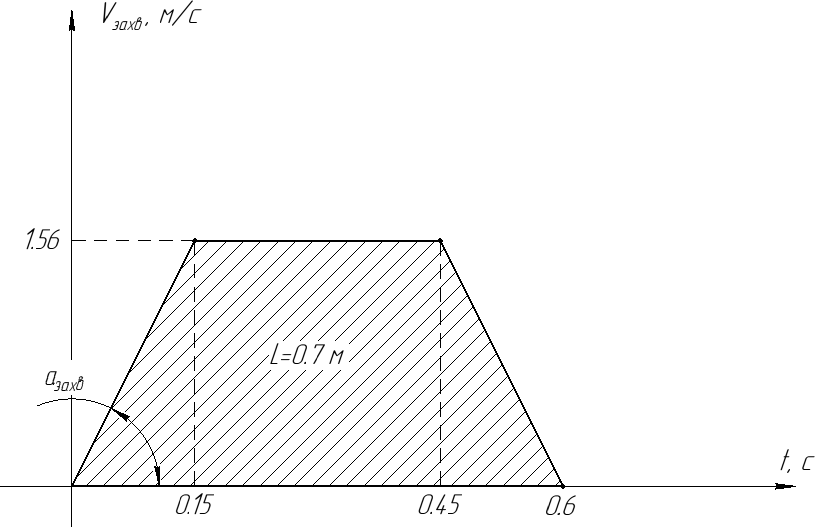
**

Рисунок 9 – Закон изменения скорости рабочего органа

Определю скорость захватного устройства с помощью площади трапеции:

Тогда ускорение можно найти по следующей формуле:

5 Исследовательская часть

## 5.1 Основные этапы работы системы

Результатом работы системы является формирование стопок писем, которые ориентированы одинаковым образом. Этапы работы системы можно разделить на следующие:

1. Разбиение исходной кучи писем на поток случайным образом ориентированных писем, уложенными в один слой – данный этап не рассматривается в рамках работы. Необходимость формирования однослойного потока писем объясняется дальнейшим распознанием писем камерой как прямоугольников белого цвета на контрастном фоне ленты. Одним из возможных способов разбиения кучи писем является использование конвейерных лент с разной скоростью вращения в сочетании с желобами для сдвига писем. Эти желоба могут крепиться над конвейерной лентой на определенной высоте для того, чтобы при движении ленты верхние уровни писем тормозились желобами и падали вниз (Рисунок 8).

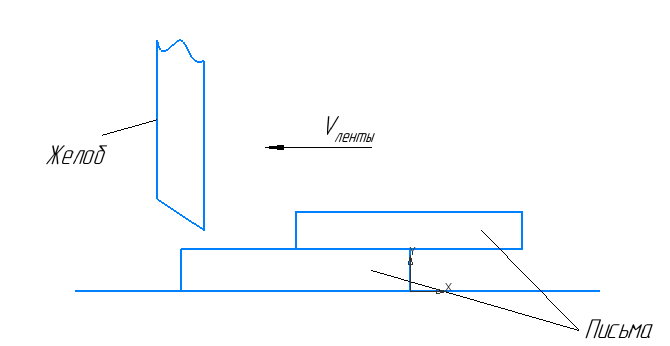


Рисунок 10 – Принцип формирования равномерного потока писем

1. Распознавание стороны письма и его ориентации с помощью камеры – с помощью камеры и библиотеки компьютерного зрения OpenCV возможно определить центр письма как центр белого прямоугольника на черном фоне с помощью камеры. Также с помощью обработки изображения возможно определить, какой стороной письмо расположено на ленте. С помощью камеры определяется угол, на который нужно повернуть конверт для его помещения в ящик с письмами.
2. Сортировка потока писем с помощью дельта-роботов – на робот приходят данные с камеры о положении центров писем, которые камера успела обработать. Эти данные приходят на робота, на привода робота подаются соответствующие сигналы управления для достижения заданной точки. Письмо захватывается роботом с помощью вакуумного захватного устройства. Все письма позиционируются одинаково, лицевой стороной вверх. Письма разделяются на 2 группы, лицевой стороной вверх и лицевой стороной вниз.
3. Письма, ориентированные лицевой стороной вверх, следуют на следующий этап сортировки, другие письма переворачиваются оператором вручную.

Работу системы можно представить в виде следующей блок-схемы (Рисунок 11):

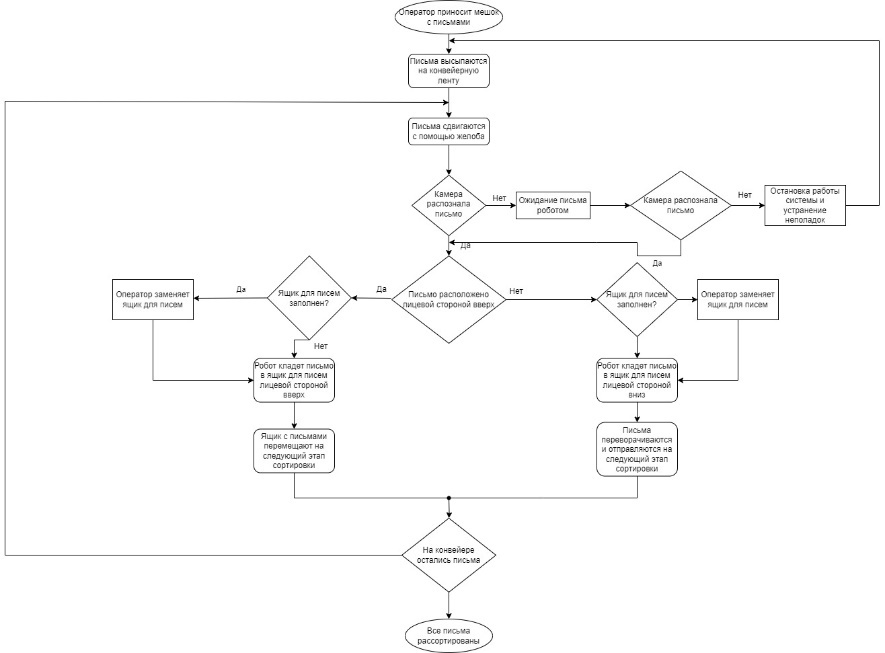


Рисунок 11 – Алгоритм работы системы

## 5.2 Разработка структуры системы управления РТК

Исходя из специфики робота, разделим систему на два уровня управления (Рисунок 11).

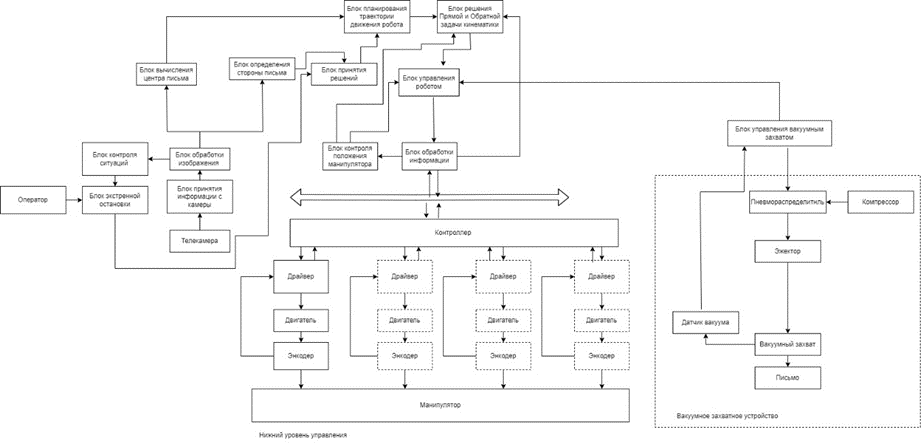
Верхний уровень рассчитан на выполнение задач, связанных со считыванием показаний с различных датчиков для локализации робота и на выполнение различных алгоритмов, а также для формирования управляющего сигнала для нижнего уровня

Уровень включает в себя:

1. Блок принятия решений, который определяет модель поведения робота в зависимости от ориентации;
2. Блок управления роботом, который получает с помощью камеры координаты центра письма и передает через шину управляющие сигналы на контроллеры приводов;
3. Блок управления вакуумным захватным устройством, который получает управляющий сигнал от блока управления роботом, также этот блок следит за контактом присоски захвата и письма;
4. Блок вычисления центра письма, который с помощью камеры определяет границы письма и его центр;

На нижнем уровне будут происходить обработка данных с датчиков двигателя и будет регулироваться система вакуумного захвата. Уровень включается в себя:

1. Контроллер, который посылает управляющие воздействия на драйвера приводов робота;
2. Энкодер для осуществления обратной связи по положению и скорости.
3. Эжектор для создания вакуума в области контакта присоски и письма

Рисунок 12 – Структура системы управления РТК

## 5.3 Определение основных параметров манипулятора

### 5.3.1 Выбор кинематической схемы

Из-за текущей ситуации в мире, многие промышленные манипуляционные роботы известных компаний крайне трудно найти на рынке, а цены на них сильно выросли. Также, оценивая скорость выполнения цикла перемещения дельта робота компании OMRON, мы можем понять, что для выполнения данной операции такая скорость перемещения является излишней, поэтому проектирование собственного манипулятора является более экономически целесообразным и практичным.

На основе аналогов решения похожих задач в промышленности была выбрана стандартная кинематическая схема дельта-робота с тремя степенями подвижности для перемещения каретки с рабочим органом манипулятора и отдельная вращательная степень подвижности для рабочего органа. Данная схема позволяет уменьшит массу подвижных звеньев за счет закрепления всех приводов к неподвижной верхней платформе робота. Схема робота приведена на Рисунке 12.

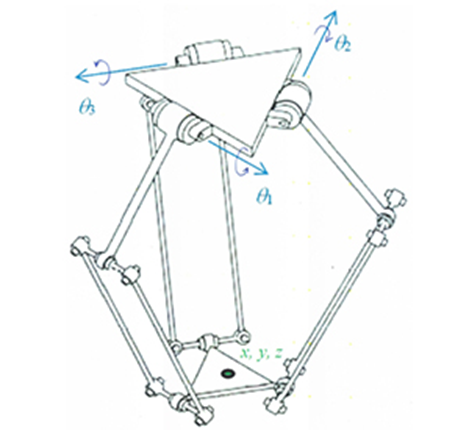


Рисунок 12 – Схема дельта робота

В качестве аналогов манипуляторов, представленных на рынке, были рассмотрены робот OMRON CR\_UGD4\_R и ABB IRB 360-3/1130 [4]. Характеристики манипуляторов приведены на Рисунке 13 [3]. Крупные фирмы предоставляют большой выбор дельта роботов, большая часть которых представляют собой именно трехрычажную схему, что подтверждает эффективность использования этой схемы для решения реальных задач.

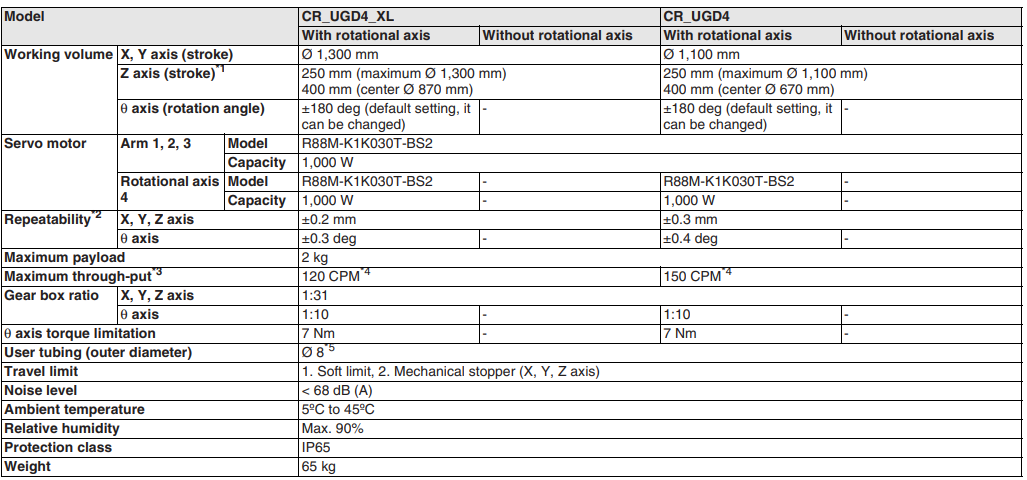


Рисунок 13 – Технические характеристики робота OMRON CR\_UGD4\_[[3]](#_Список_использованной_литературы)

Платформа для установки робота была спроектирована в рамках данной работы, внешний вид платформы представлен на Рисунке 14.

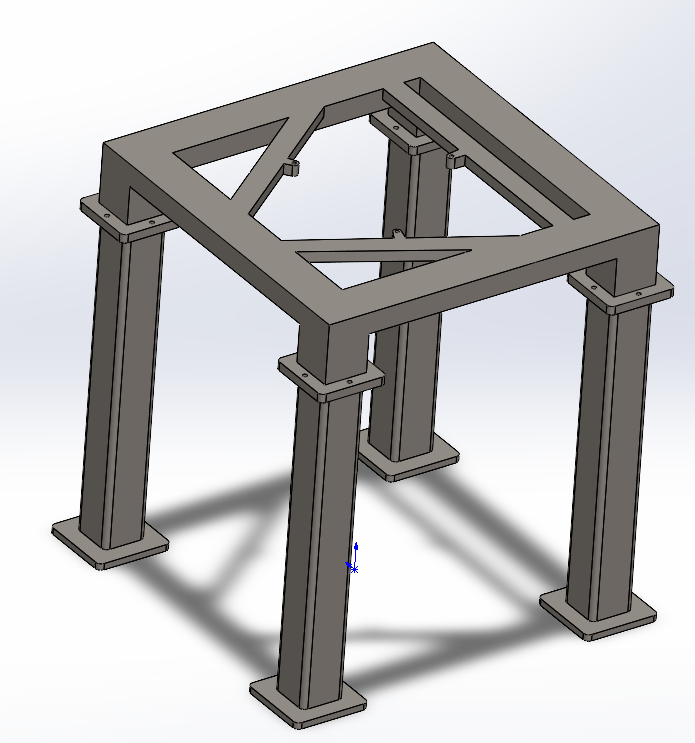


Рисунок 14 – Внешний вид платформы для установки робота

### 5.3.2 Решение прямой и обратной задачи кинематики по положению

Решением прямой задачи кинематики по положению является определение координат ориентации рабочего органа манипулятора по известному вектору обобщенных координат. Для решения прямой задачи необходимо определить матрицу перехода между неподвижной системой координат, которая связана с основанием робота, и подвижной системой координат, связанной с рабочим органом манипулятора. В общем виде однородная матрица преобразования между – ой и – ой системой координат примет вид:

В этой формуле – матрица поворота; – вектор переноса; – вектор, связанный с вектором центрального проектирования; – коэффициент масштабирования. С помощью однородной матрицы преобразования возможной найти координаты вектора – ой системы координат в – ой системе координат.

Решение прямой задачи по положению в данной работе было получено геометрическим методом на основе выбранной конфигурации робота. Расчетная схема представлена на Рисунке 15.[10]

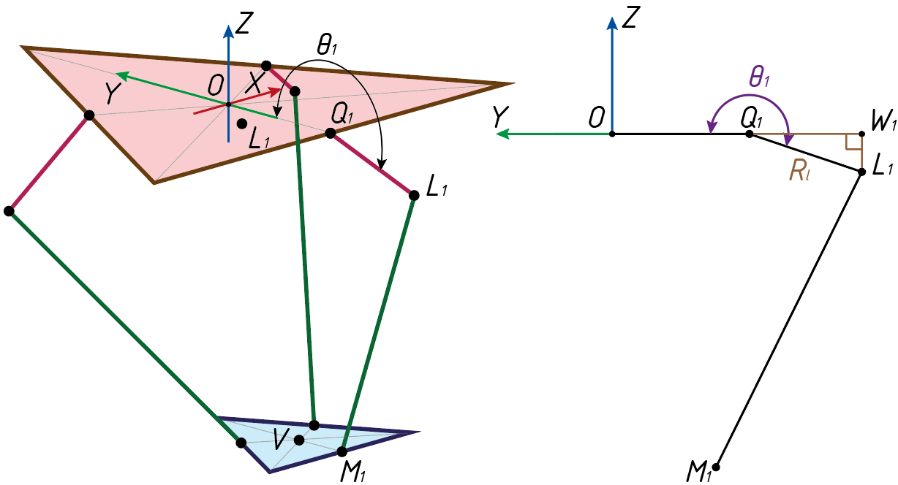


Рисунок 15 – Расчетная схема для решения ПЗК

Особенностью механизма дельта робота является параллельность подвижной платформы робота относительно верхней неподвижной платформы, поэтому для решения прямой задачи кинематики необходимо определить только координаты подвижной системы координат, связанной с нижней платформой.

Рассмотри плечо, лежащее в плоскости ZOY. Точка является точкой касания окружности, вписанной в треугольник неподвижной платформы робота, со стороной треугольника. Координату точки  по оси *Y* найдём с помощью известной для равностороннего треугольника формулы поиска радиуса вписанной окружности:

где F – длина стороны равностороннего треугольника.

Зная координаты точки можно определить координаты точки :

где – длина плеча робота.

Для поиска координат и оставшихся плеч воспользуюсь той же формулой:

Координаты ,и , вычислю с помощью систем координат, повернутых относительно базовой на 120° и 240° соответственно. В повернутых системах координат найду координаты и .

Точки *L*₁, *L₂* и *L₃* представляют собой трёхподвижные шарниры, и, если принять эти точки за центры вращения, то их концы – точки *M*₁, *M₂* и *M₃* описывают 3 сферы. Сдвинем каждую штангу параллельно плоскости *XOY* так, чтобы точки *M*₁, *M₂* и *M₃* оказались в точке *V*, что наглядно показано на Рисунке 16.

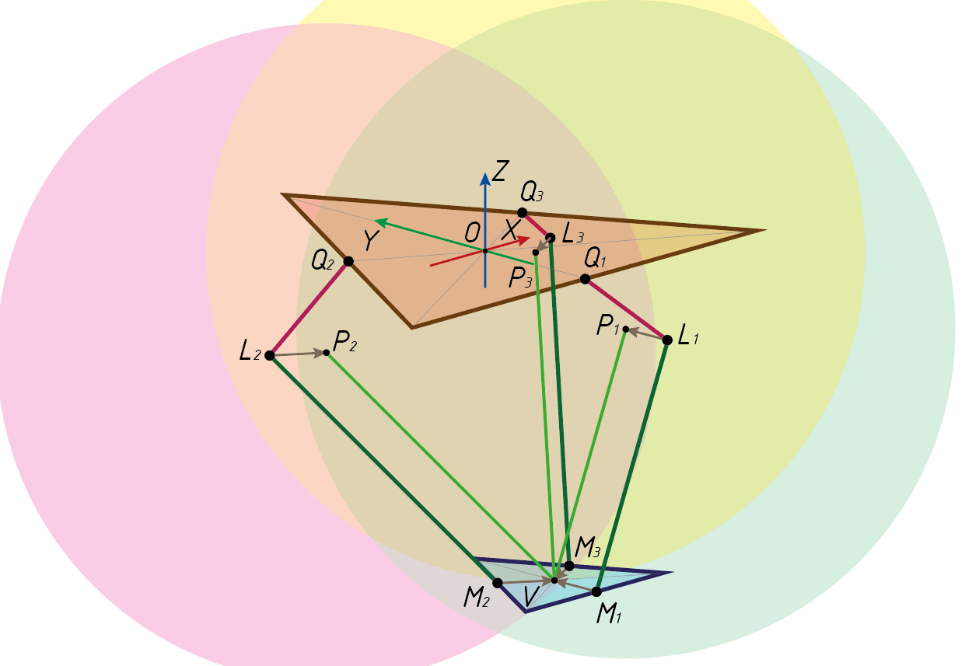


Рисунок 16 – Смещение звеньев механизма

Теперь рассматриваемые сферы (уже с центрами в точках *P*₁, *P₂* и *P₃*) пересекаются как раз в точке *V*. Для поиска координат новых центров сфер перейдём в плоскость *XOY* (рисунок ниже). Для поиска координаты точки *P*₁ достаточно сдвинуть координату по оси *Y* точки *L*₁ на  в большую сторону (f – сторона равностороннего треугольника подвижной платформы).

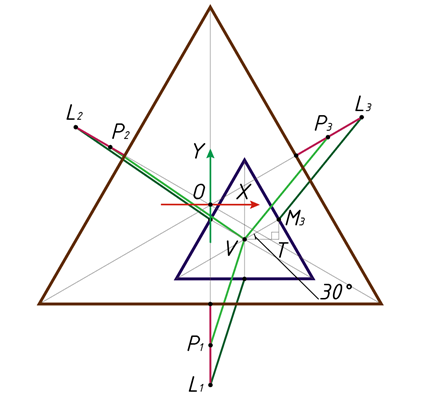


Рисунок 17 – Определение координат точки V

Координаты остальных точек легко найти, рассматривая прямоугольный треугольник *VM₃T*, гипотенуза которого равна так же. Из него координаты точек *P₂* и *P₃* определю по формулам:

После вышеизложенных расчётов мы можем найти необходимые координаты точки *V*, как меньшее по оси *Z* решение системы уравнений (21), представляющее собой нижнюю точку пересечения трёх сфер.

После преобразований и введения новых переменных получу решение системы (17):

По аналогии введу такие же обозначения и для других уравнений системы, тогда координаты точки V можно записать следующие выражения:

Найду координату ;

где

Из полученных корней выбираем наименьший и находим значение остальных координат.

Целью решения обратной задачи кинематики по положению заключается в определении значения обобщенных координат робота по известным значениям координат рабочего органа. В данной работе решение обратной задачи кинематики получено из геометрии механизма.

Начнём с рассмотрения плеча, лежащего в плоскости *YOZ*. Для определения требуемого угла  достаточно знать координаты точек *Q* и *L*, а вычислить угол можно по формуле:

Для того, чтобы найти координаты точки *L* проанализируем траектории движения концов рычага и штанги. Точка *L*, принадлежащая рычагу, описывает окружность в плоскости *YOZ* с центром в точке *Q*. Точка *M* является трёхподвижной парой, поэтому точка *L*, принадлежащая штанге описывает сферу с центром в точке *M*. Следовательно, для определения координаты точки *L* всего механизма нужно найти точку пересечения сферы и окружности. Таких точек пересечения две – *L* и *L₁*, нам же нужно выбрать точку, у которой координата по оси *Y* меньше. Соответственно, мы должны решить систему из двух уравнений:

где – длина рычага, – длина штанги.

Координату точки М по оси Y найду из уравнения:

Остальные координаты точки *M* будут равны соответствующим координатам точки *V*, что приведёт к итоговой системе:

Раскрыв скобки в первом и втором уравнениях и вычтя из первого уравнения второе, получим равенство, из которого линейным образом можно выразить  по формуле:

где

Тогда, подставив это в первое уравнение системы и приведя его к виду квадратного уравнения относительно можно составить систему для расчёта коэффициентов квадратного уравнения:

Далее найдём меньший корень квадратного уравнения по формуле ниже и подставим полученное решение в формулу (25), найдя тем самым неизвестную координату точки *L* по оси *Z*.

Для поиска двух оставшихся углов поворота рычагов воспользуемся следующим приёмом. Введём вспомогательную систему координат, повёрнутую на 120° вокруг оси *Z* относительно основной.

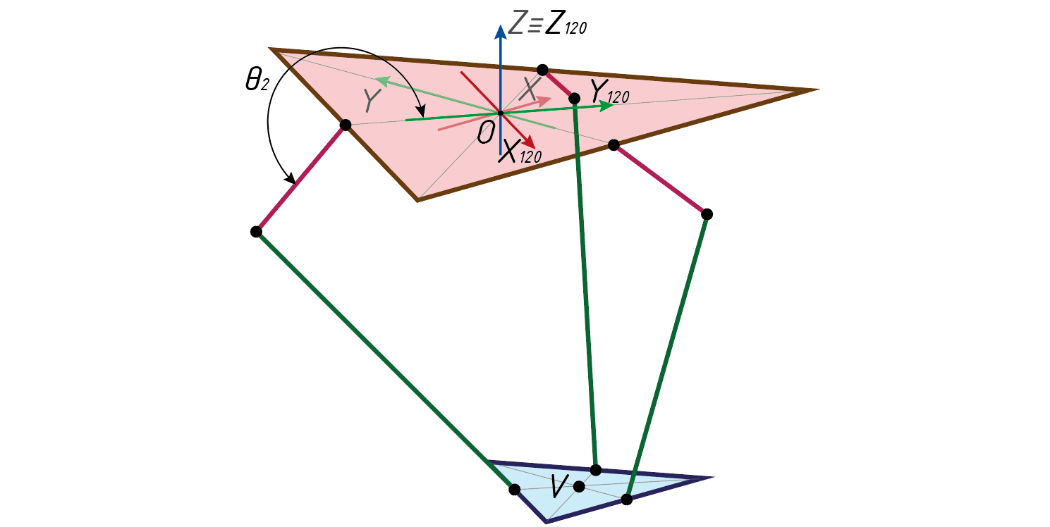
**

Рисунок 18 – Вспомогательная повернутая система координат

В новой системе координат задача для определения угла  сводится к задаче определения угла  с той лишь разницей, что координаты точки *V* нужно перевести из *XYZ* в  по известным формулам:

Аналогичным образом можно рассчитать угол поворота третьего рычага, введя систему координат, повёрнутую на 240°.  Заметим, что данные формулы справедливы лишь для случая, когда мы находим координаты точки в системе, повёрнутой по часовой стрелке относительно основной.

Реализация алгоритмов решения ПЗК и ОЗК с помощью программы MATLAB [11] приведена в Приложении А.

### 5.3.3 Метрический синтез в соответствии с требуемой рабочей зоной

Для того, чтобы реальная рабочая зона манипулятора включала в себя всю желаемую рабочую зону, необходимо провести метрический синтез звеньев разрабатываемого механизма. Для определения реальной рабочей зоны дельта робота воспользуюсь дискретным методом построения рабочей зоны.

Дискретный метод заключается в разбиении всего пространства на отдельные точки, решении обратной или прямой задачи кинематики в них и проверке этого решения на нарушение ограничений. Потом итерационно изменяется угол одного из рычагов на небольшой шаг. Далее изменяется угол другого рычага на также небольшой шаг и повторяется перебирание углов первого рычага. Таким образом, перебираются все возможные комбинации углов трёх рычагов. На каждой итерации осуществляется проверка, есть ли какие-то ограничения для перемещения каретки механизма в точку, соответствующую этим углам, и, если есть, то эта итерация прерывается и добавления получившейся точки в итоговый массив, описывающий рабочую зону, не происходит. В конечном счёте, для механизма можно найти набор точек, в которые рабочий орган может переместиться, а граничная поверхность, поострённая по крайним точкам этого набора, характеризует форму и размеры получившейся зоны. Но для обеспечения достижимости роботом желаемой точки необходимо также ряд ограничений, которые накладываются возможным положения рабочего органа.

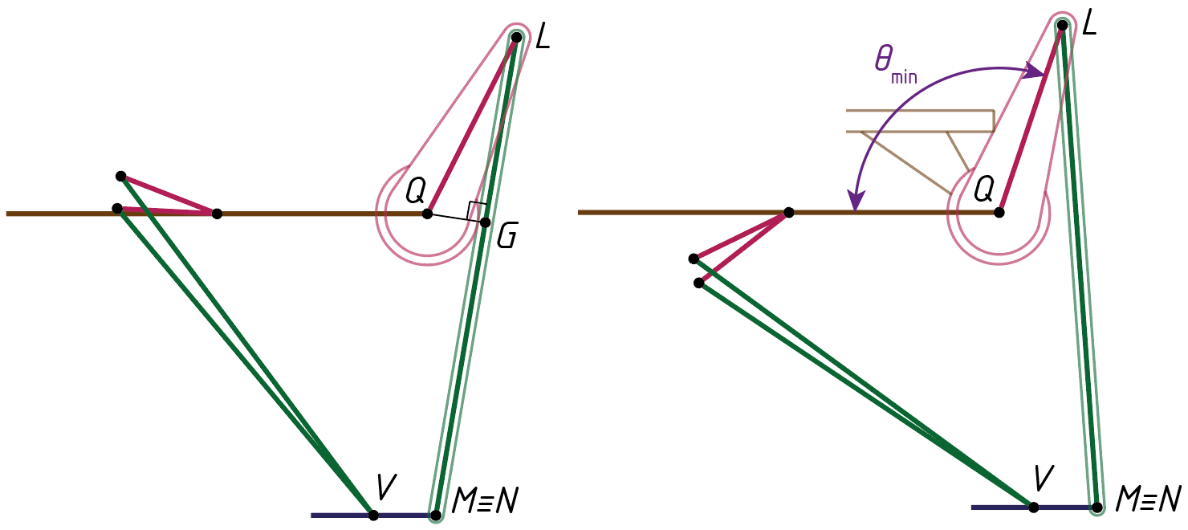


Рисунок 19 – Самопересечение звеньев манипулятора

Расстояние *QG* представляет собой сумму радиуса штанги и радиуса окружности (трубы), в которую можно вписать все приводные элементы (двигатель, редуктор, крепёж).

Угол  потребуется примерно оценить по предполагаемой конструкции основания и элементов крепления приводов. Этот угол в алгоритме построения легко учитывается – начну перебор углов рычагов с этого значения.

Не приблизилась ли штанга к приводу можно оценить по углу . С одной стороны, этот угол не должен быть меньше некоторого минимального значения , зависящего от *QG*. С другой стороны, этот угол не может быть больше 180°.

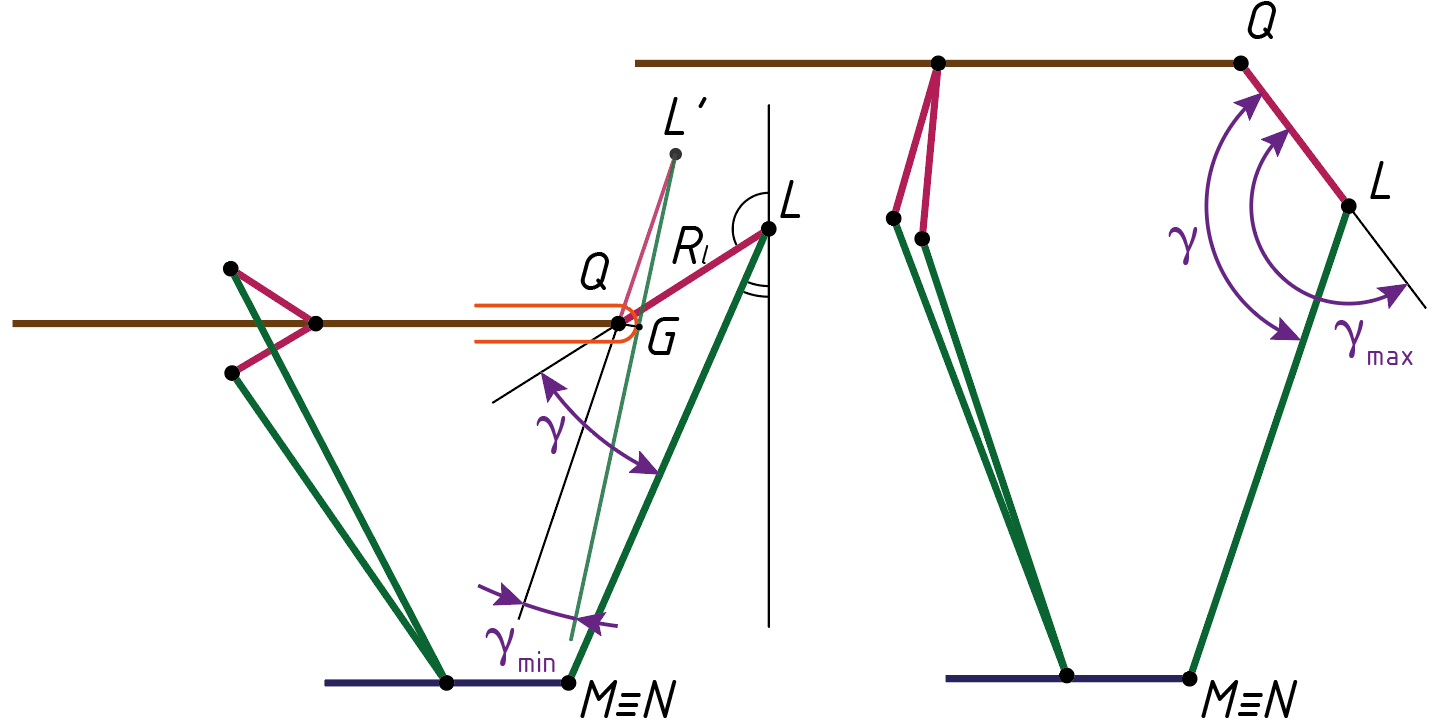


Рисунок 20 – Оценка самопересечения звеньев

Тогда искомый угол  найдем по формуле:

Необходимо применять в системе координат, в которой рычаг вращается в плоскости *YOZ*. Минимальный угол  определим из прямоугольного треугольника *QL'G*, в котором сторона *L'G* является касательной к окружности с центром в точке *Q* и радиусом *QG*. Зная также длину рычага найдем по формуле:

Следующее ограничение связано с трёхподвижными шаровыми шарнирами. Все они имеют предельный угол поворота как минимум вокруг двух осей. При использовании шаровых магнитных шарниров, зачастую, можно считать, что прочие используемые ограничения, не позволят шаровому шарниру дойти до крайнего положения. В случае использования других шарниров угол между штангой и рычагом будет ограничен максимальным углом поворота шара в обойме. Тогда, чтобы убедиться, что это ограничение не нарушается, нужно найти угол  между штангой и плоскостью вращения рычага и проверить, чтобы он не был больше, чем  для каждого плеча.

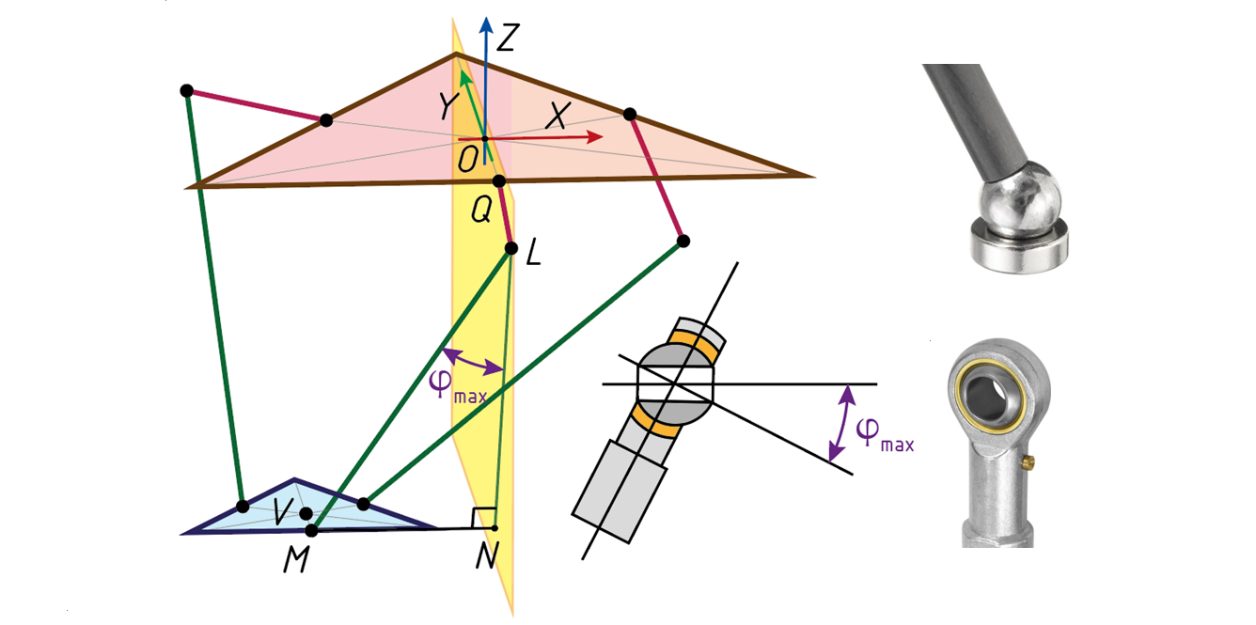


Рисунок 21 – Ограничения трехподвижного шарнира

Сторона LM известна и равна длине штанги , а сторона MN равна координате точки V по оси X. Тогда угол φ  можно найти по формуле:

Последним ограничением выделим допустимое значение угла давления, превышение которого может привести к заклиниванию и самоторможению. В исследуемом механизме можно выделить несколько углов давления, но мы остановимся на наложении ограничений на один из них – угол между касательной к траектории движения конца рычага, вдоль которой направлена скорость, и штангой, вдоль которой направлена сила. Наглядно угол давления показан υ на рисунке:

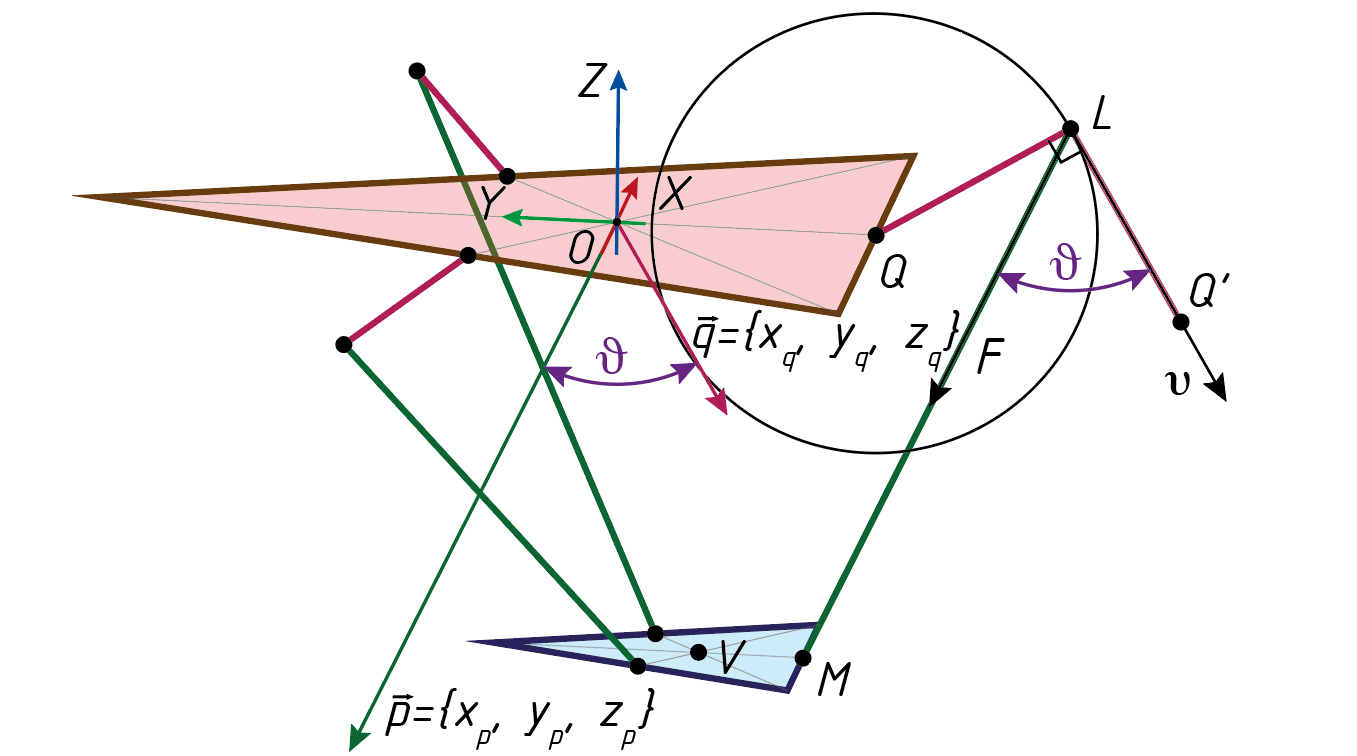
**

Рисунок 22 – Определение угла давления

Для его определения найдём координаты двух векторов  и . Если координаты первого вектора легко найти через координаты точек *M* и *L*, то для поиска второго вектора сначала нужно найти координаты точки *Q'*, повернув отрезок *QL* на 90° относительно точки *L*, чтобы получить касательную. В итоге, координаты векторов и  найдём по формулам:

Теперь угол давления найдём как угол между векторами  и  по формуле:

Метрический синтез определяется итерационным методом. Буду задавать размеры звеньев и запускать алгоритм, который будет проверять, подходит ли данная конфигурация для обслуживания дельта-роботом желаемой рабочей зоны. Визуально можно будет оценить, насколько реальная рабочая зона больше требуемой, алгоритм также будет выдавать и итоговые габариты получившегося устройства. Найдя подходящую конфигурацию, можно будет оценить, насколько робот отличается по размерным пропорциям от предложенных на рынке.

Итак, по приведённому выше алгоритму можно построить реальную рабочую зону. Тогда, имея все входные данные, можно построить желаемую зону и распределить по ней точки для анализа:

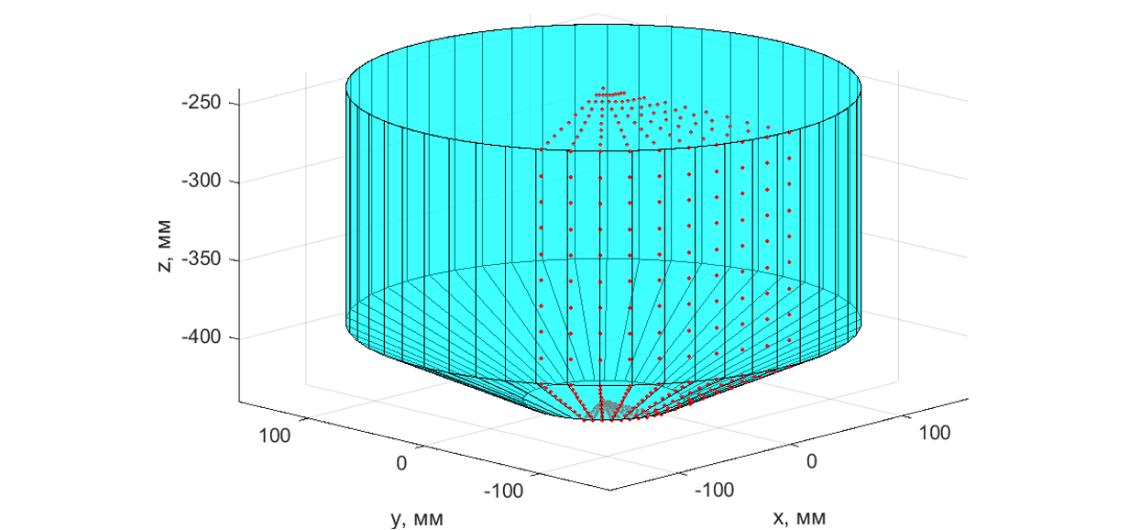


Рисунок 23 – Выбор точек для определения рабочей зоны

Точки для анализа распределяются по поверхности рабочей зоны. Теперь для каждой точки можно решить обратную задачу кинематики и проверить все условия, которые проверялись для точек рабочей зоны при построении последней. Если все точки этим условиям удовлетворят, то можно сделать вывод, что данная желаемая рабочая зона в реальную вписывается.

Можно заметить, что точки распределены не по всей поверхности, а по 1/6 её части. Это связанно с симметрией робота. При любом анализе можно рассматривать лишь 1/6 часть, отсчитываемую от плоскости *YOZ* в любую сторону на 60°. Реализация алгоритмов метрического синтеза в программе MATLAB представлена в Приложении А. Реальная и желаемая рабочая зона манипулятора изображена на Рисунке 24.

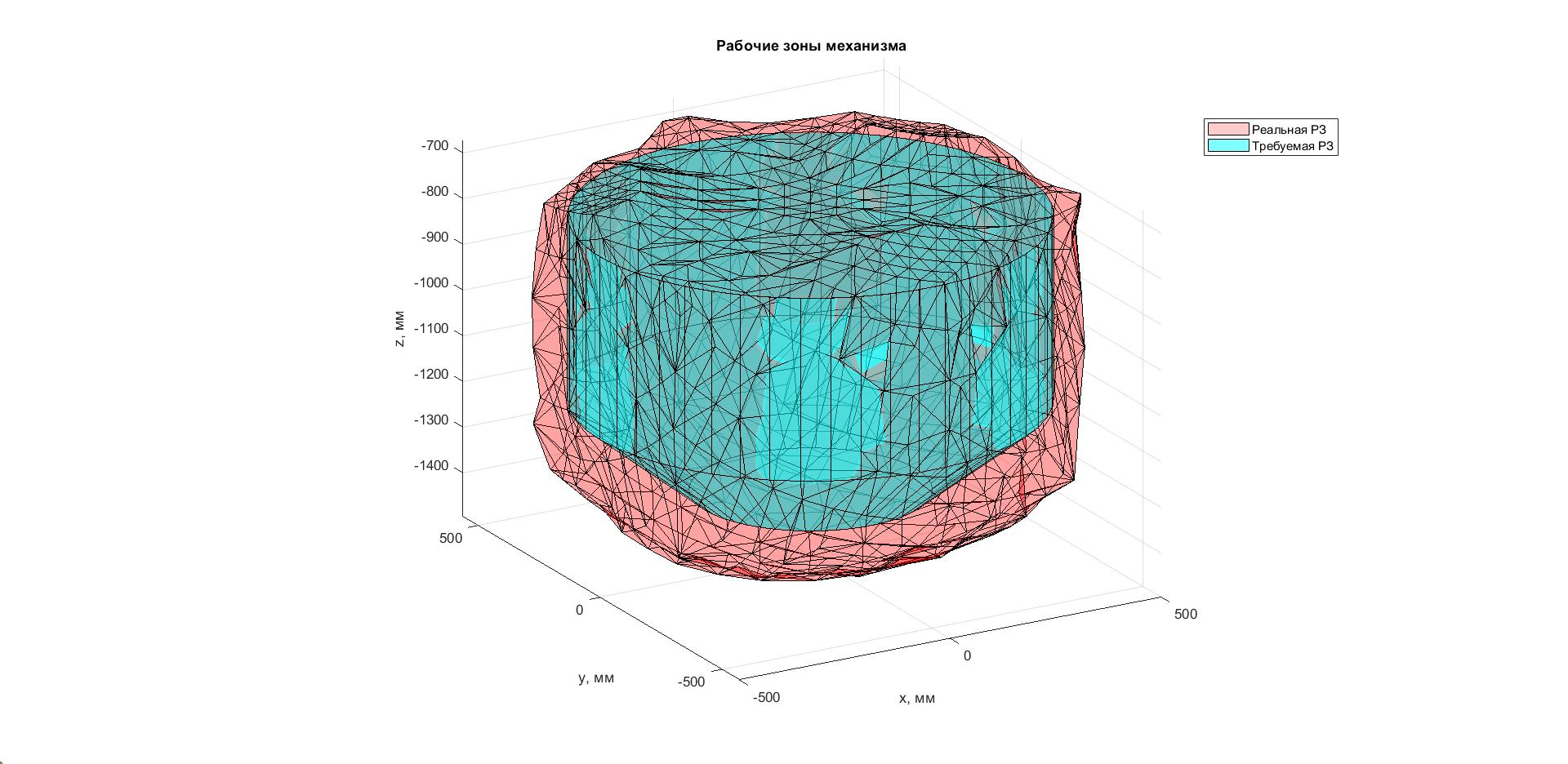
**

Рисунок 24 – Реальная и желаемая рабочая зона

Размеры механизма, полученные в результате метрического синтеза:

1. Длина стороны треугольника неподвижной платформы
2. Длина стороны треугольника нижней платформы
3. Длина рычагов
4. Длина штанги
5. Вынос основания

### 5.3.4 Решение прямой и обратной задачи кинематики по скорости

Целью решения прямой задачи кинематики по скорости является определение скорости движения рабочего органа относительно неподвижной системы координат по известным значениям производных обобщенных координат манипулятора и самих обобщенных координат.

Для решения прямой задачи необходимо записать уравнения связи манипулятора:

где - функция положения.

Записывая такие же уравнения для каждой степени подвижности, я могу составить матрицы:

С помощью этих матриц запишу матричное уравнение:

Решение прямой задачи по скорости можно найти по формуле:

где J – матрица Якоби для данного положения манипулятора.

Тогда решение обратной задачи кинематики по скорости можно записать в таком виде:

Алгоритмы решения прямой и обратной задачи по скорости реализованы в программе MATLAB, тексты программ приведены в Приложении Б.

Для подбора электродвигателя и редуктора необходимо определить максимальную требуемую скорость вращения плеча манипулятора, которая будет обеспечивать заданную максимальную скорость перемещения рабочего органа. Для обеспечения заданную скорость выполнения операции манипулятора скорость рабочего органа должна быть Для определения требуемой скорости вращения были реализованы алгоритмы в программе MATLAB. Функция genTP задает массив точек, находящийся внутри рабочей зоны манипулятора. Для каждой точке высчитывается набольшее значение угловой скорости плеч, при котором достигается заданная скорость движения рабочего органа, затем из массива полученных угловых скоростей выбирается наибольшее, что и является искомой величиной. Наибольшее значение угловой скорости рассчитывается по формуле:

где – i-ая вектор-строка обратной матрицы Якоби.

В результате вычисления я получил значение .

### 5.3.5 Определение максимального требуемого момента

Необходимые крутящие моменты, приложенные к плечам манипулятора, можно определить по следующей формуле [9]:

где – крутящие моменты первого, второго и третьего рычагов,

– составляющие приложенной к рабочему органу силы по соответствующим осям,

– матрица Якоби.

Для определения максимального приложенного момента необходимо рассчитать приведенную массу звеньев манипулятора. После создания 3D-моделей звеньев манипулятора и задания им материалов их масс-инерционные характеристики становятся известными, поэтому возможно произвести приведения масс подвижных звеньев к точке V.

Приведенную к точке V массу всего манипулятора можно записать в виде:

где – приведенная к точке V масса подвижной платформы вместе с захватываемым объектом и захватным устройством,

– приведенные массы рычагов с учетом штанг.

В общем случае приведенная к точке А масса звена выражается по формуле:

где – момент инерции i-го звена относительно мгновенной оси вращения,

– масса i-го звена,

– угловая скорость i-го звена относительно мгновенной оси вращения,

– линейная скорость мгновенной оси вращения i-го звена,

– линейная скорость точки приведения A.

Вычислю силу, которая заменяет инерционные нагрузки всех звеньев механизма:

Для подбора электродвигателя необходимо определить максимальный крутящий момент, поэтому и приведенная масса должна быть максимальной, что достигается при при заданном значении скорости рабочего органа . Поэтому при известной максимально скорости движения рабочего органа определяется максимальное значение крутящего момента.

Максимальное значение крутящего момента можно определить по формуле:

Используя эту формулу, я могу найти максимальное значение крутящего момента рычага для каждой точки заданного массива при известном значении максимального ускорения рабочего органа .

Реализация алгоритма поиска наибольшего крутящего момента были произведена в программе MATLAB, текст программы приведен в Приложении А. В результате вычислений наибольший крутящий момент в рабочей области манипулятора при ускорении рабочего органа , приведенном к рычагу моменте инерции нагрузки , массе подвижной платформы с массой письма и захватного устройства .

6. Конструкторская часть

## 6.1 Энергетический расчет привода манипулятора

Определение необходимой мощности электродвигателя:

Определение расчетного КПД привода:

Определение требуемой мощности двигателя с учетом КПД и коэффициента запаса:

Подобран двигатель Maxon EC-I 52 мощностью 250 Вт (Рисунок 25) и планетарный редуктор Maxon GP 52 C с передаточным отношением (Рисунок 26).

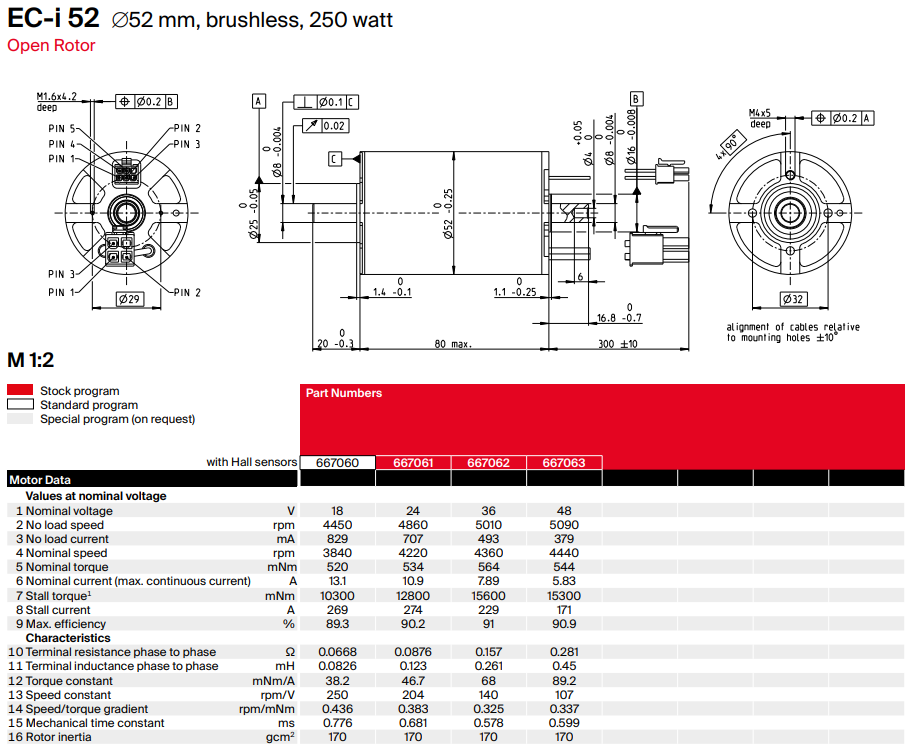
**

Рисунок 25 – Двигатель Maxon EC-I 52 [8]

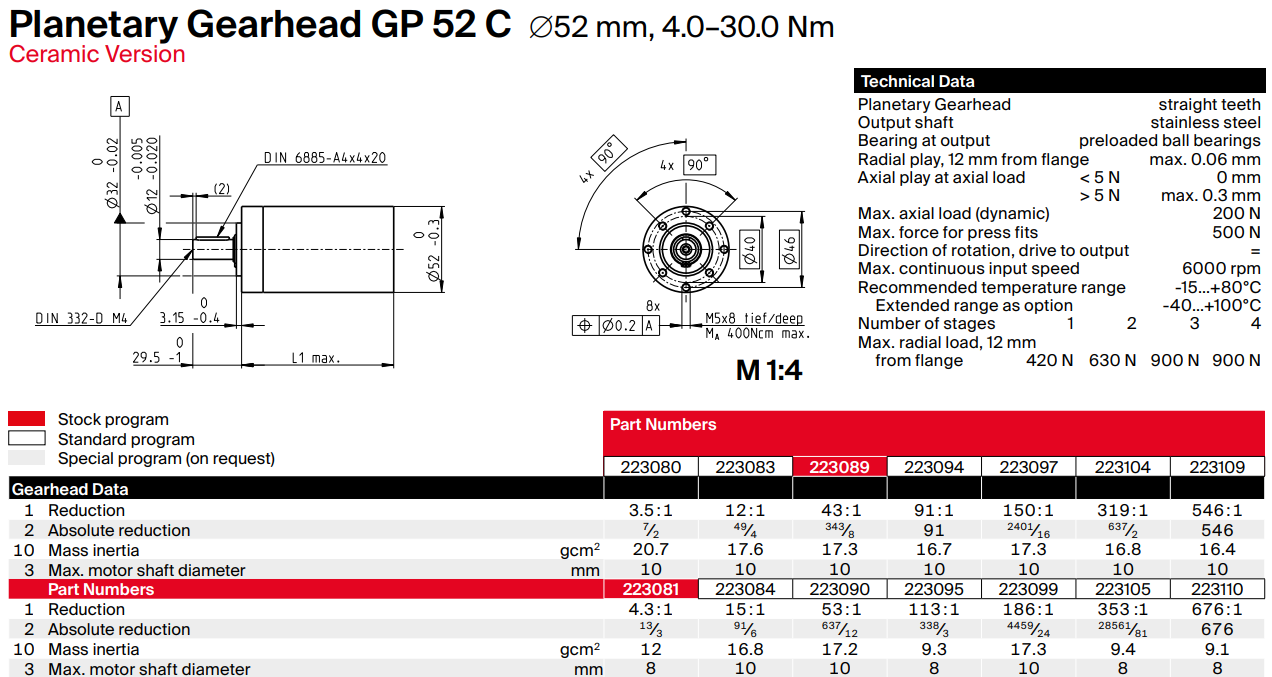
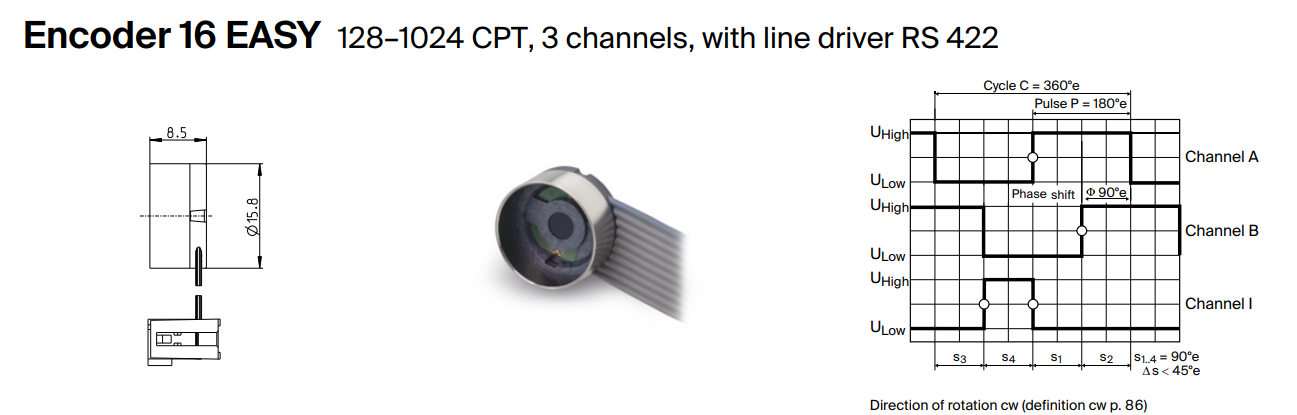
**

Рисунок 26 – Планетарный редуктор GP 52 C

Рисунок 27 – Энкодер 16 EASY

Приведу момент инерции нагрузки к валу электродвигателя:

Момент инерции ротора двигателя .

## 6.2 Регулировочный расчет привода манипулятора

### 6.2.1 Исходные данные

Определить вид и рассчитать параметры регуляторов электропривода робота по следующим исходным данным:

Таблица 2 – Исходные данные

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Тип  Эл. Двиг. | Rя  Ом | Тэ  с | Км  Н\*м/А | Кw  В\*с/рад | Jд  Кг\*м2 | J’н  Кг\*м2 | Ошибка  Емах  рад | Макс скорость  qр  рад/с | Макс  ускорение  qр рад/с2 | Передат  Число  Редуктора  I |
| Maxon EC I-45 | 0.07 | 0,0008 | 0.04 | 0.04 | 0,000017 | 0,00024 – 0,00028 | 0,05 | 3,27 | 21,48 | 91 |

Для упрощения расчетов принимаю:

* Крутизну характеристики датчика тока Кт = 2 В/A;
* Крутизну характеристики датчика скорости Кс = 1 В\*с/рад;
* Коэффициент усиления усилителя мощности Ку =5;
* Постоянную времени усилителя мощности Ту = 0,0005 с;

В процессе расчета необходимо выбрать вид и рассчитать параметры регуляторов тока, скорости и положения, обеспечивающие:

* Полосу пропускания контура скорости не менее 70 Гц;
* Показатель колебательности контура скорости Мск не выше 2;
* Показатель колебательности контура положения М не выше 1,3

Таблица 3 -

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Номинальный момент  Н\*м. | Ном.  Ток  А | Ном.  Скорость  Об/мин | Ном.  мощность  Вт | Ном.  Напряжение  В | Время разгона и торможения Тр, Тт  с | Время движ. с пост. Скоростью Тпс  с |
| 0.534 | 0.109 | 4220 | 250 | 24 | 0,15 | 0.3 |

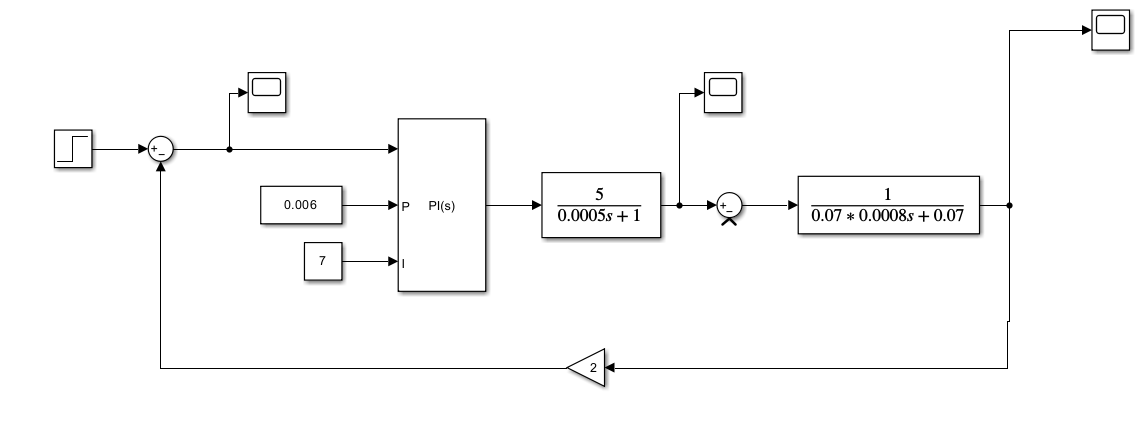
**Расчет параметров неизменяемой части привода:**

Суммарный момент инерции

Электромеханическая постоянная времени

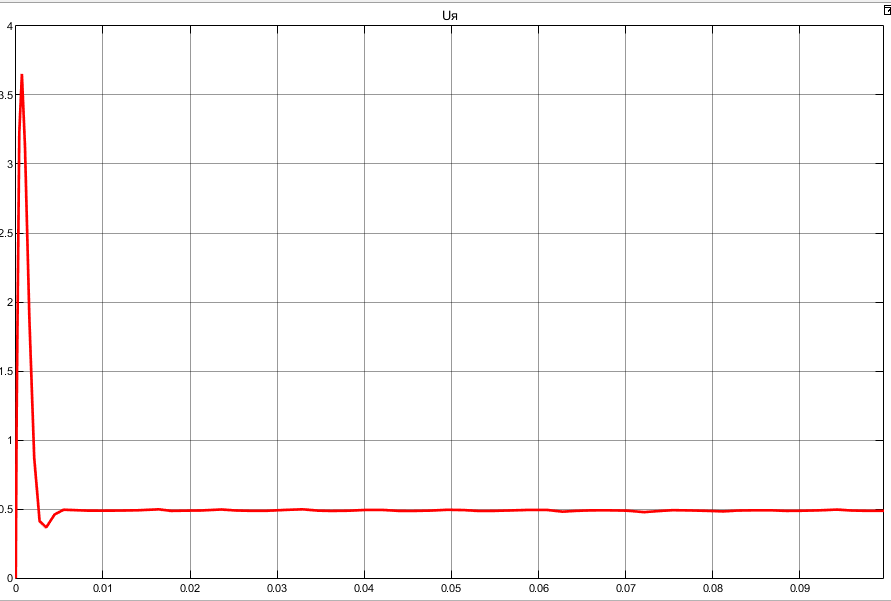
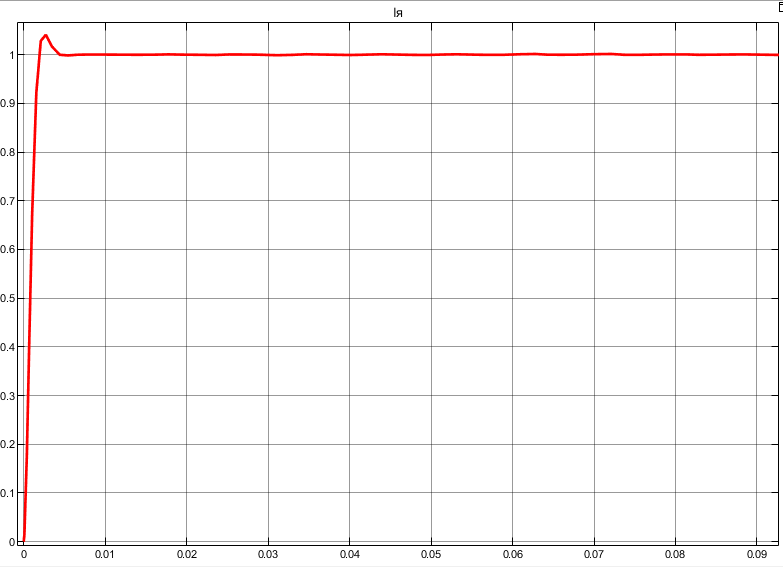
Полином неизменяемой части

### 6.2.2 Расчет контура тока

Рисунок 28 - Структурная схема контура тока

Требуется высокая точность, поэтому выбираем «ПИ»-регулятор тока. Контур тока рассчитывается таким образом. чтобы обеспечить перерегулирование Мт =1. Настройка на МО(модульный оптимум).

Результаты расчета контура тока:

а) б)

Рисунок 29. а) по **напряжениям**; б) по **току**. Отработка единичного сигнала в контуре тока в момент времени 0.1 сек.

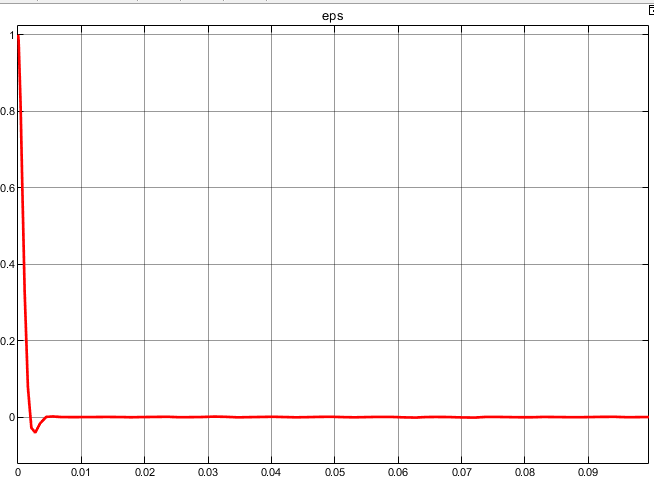


Рисунок 30. График **ошибки**. Отработка единичного сигнала в контуре тока в момент времени 0.1 сек.

### 6.2.3 Расчет контура скорости с внутренним контуром тока

* Требования к качеству переходного процесса Мск = 2 – настройка на СО (быстрая отработка возмущающего воздействия)
  + - Требования по быстродействию

Наличие интегральной составляющей в РС обеспечивает контуру скорости дополнительный астатизм по управлению и возмущающему воздействию. Лучше всего система с двумя «ПИ»-регуляторами с II порядками астатизма по управлению и I порядкой по возмущению. Выбранный выше ПИ-регулятор скорости обеспечивает независимость установившейся скорости от внешнего момента.

Выбираем «ПИ» регулятор контура скорости с внутренним ПИ регулятор тока. Для данной системы возможно только настройка на СО.

Динамические свойства контура скорости будут зависеть от текущего значения J (момента инерции).

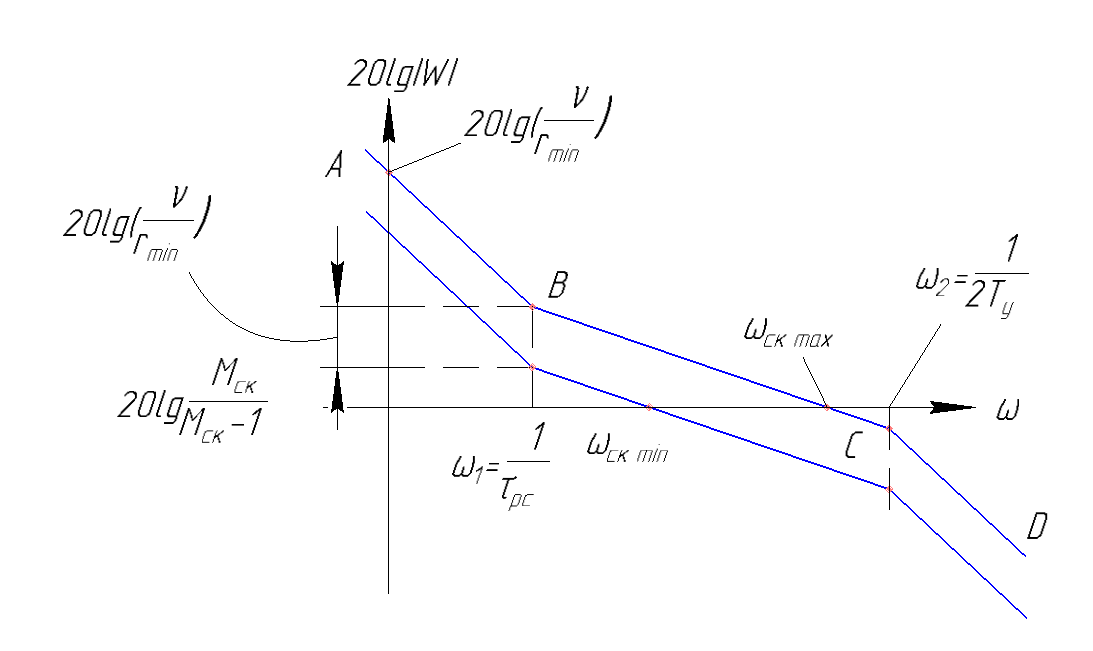


Рисунок 31. Анализ ЛАЧХ системы

Выберем так, чтобы

A(

Рассмотрим «AB»:

A(

Результаты расчета контура скорости:

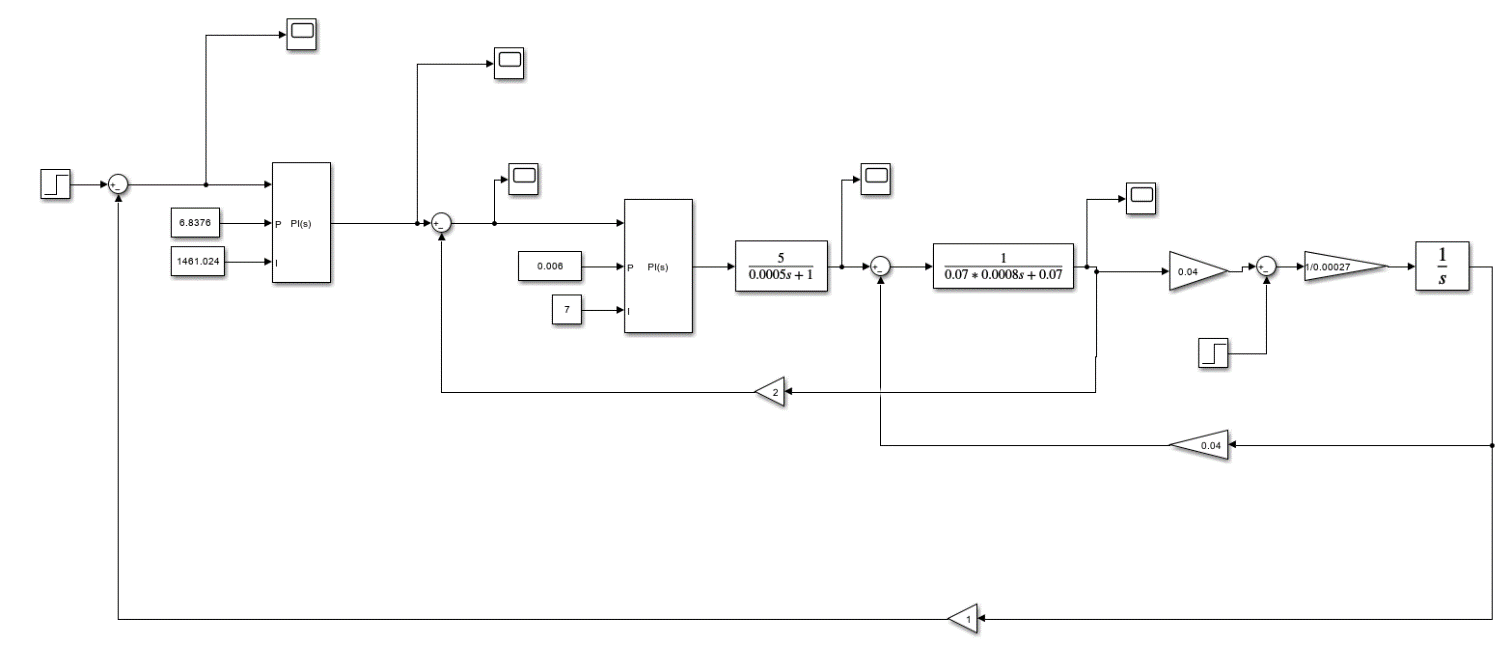
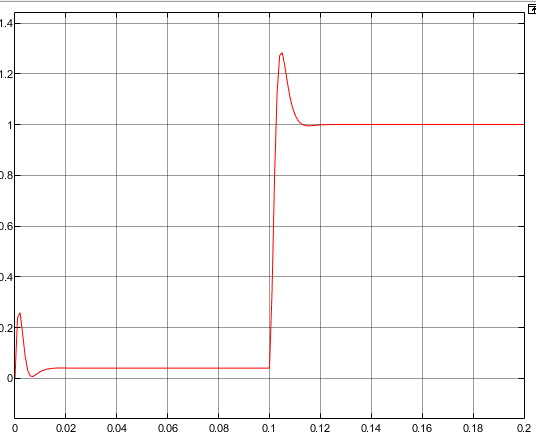


Рисунок 32 - Структурная схема контура скорости



а) б)

Рисунок 33. а) по **напряжениям**; б) по **току**. Отработка единичного сигнала в контуре скорости в момент времени 0 сек. и подача на вал двигателя момент в 0.1 сек.

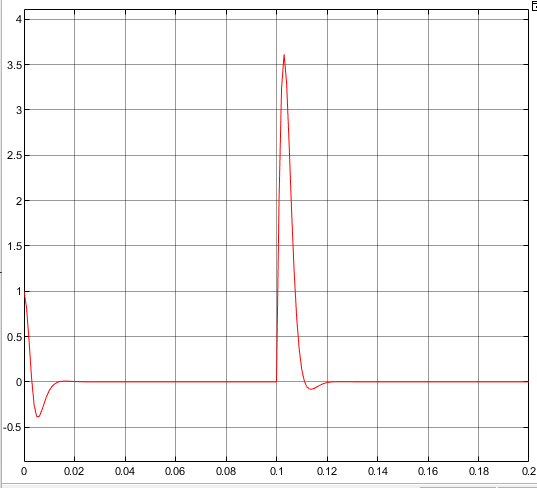


Рисунок 34. График **ошибки** для контура скорости

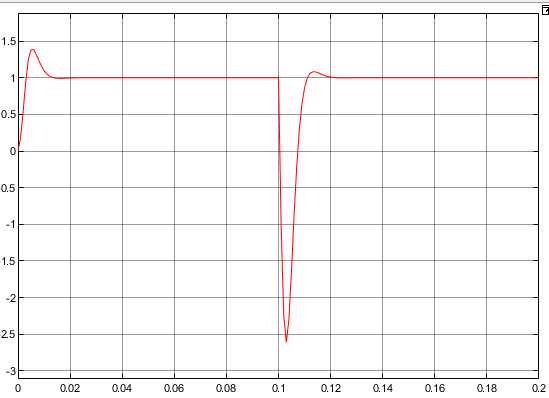
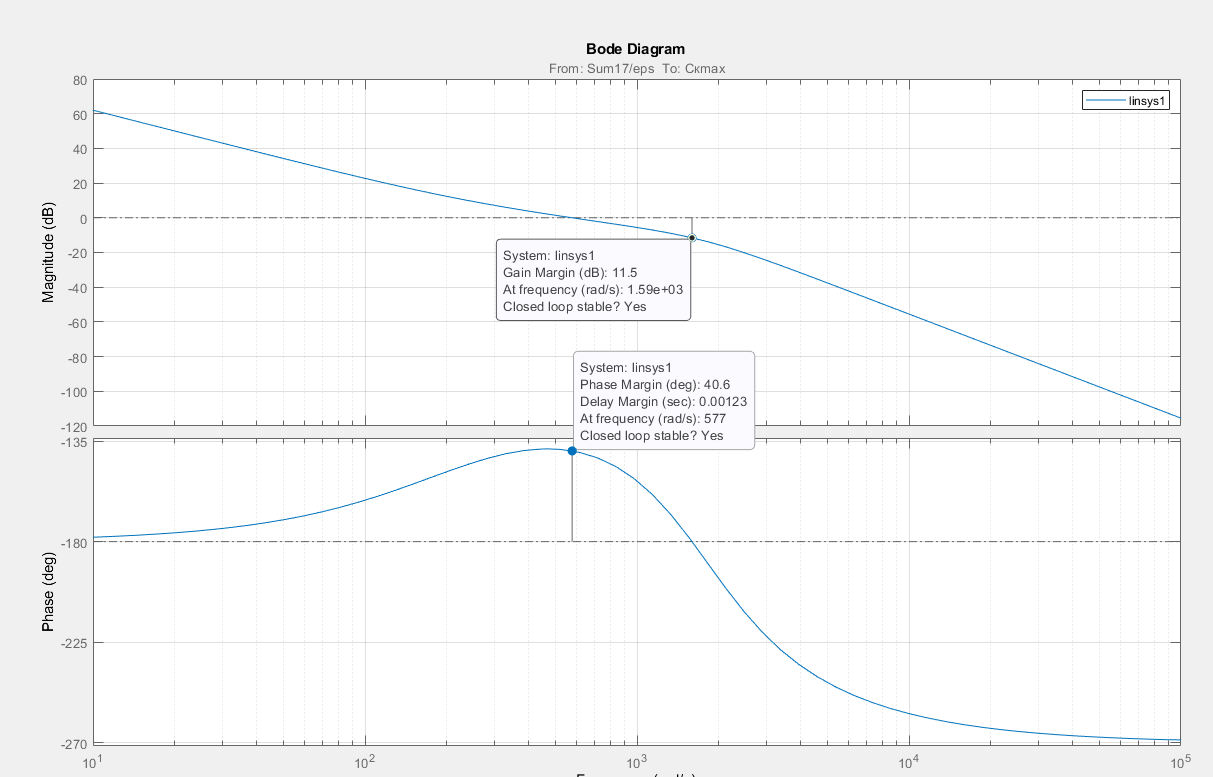


Рисунок 34. **Скорость** вала двигателя. Отработка единичного сигнала в контуре скорости в момент времени 0 сек. и подача на вала двигателя момент в 0.1 сек.

Рисунок 35. **ЛАФЧХ** для контура скорости

### 6.2.4 Расчет контура положения в позиционном режиме

Ввиду того что поэтому

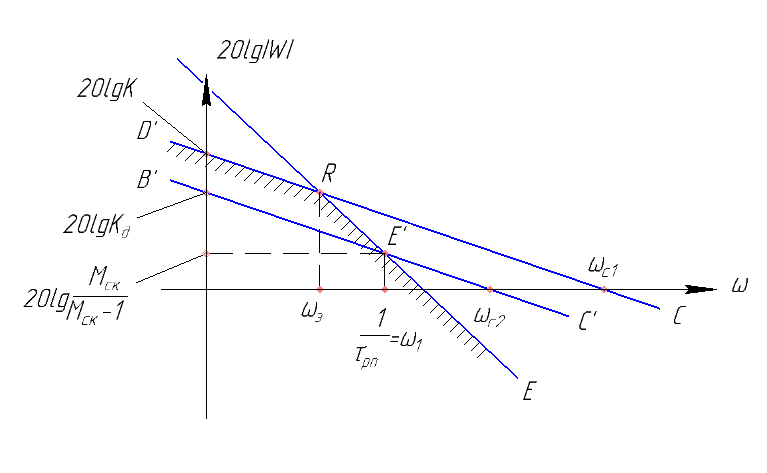


Рисунок 36. Анализа ЛАЧХ системы

Выбираем «П» регулятор

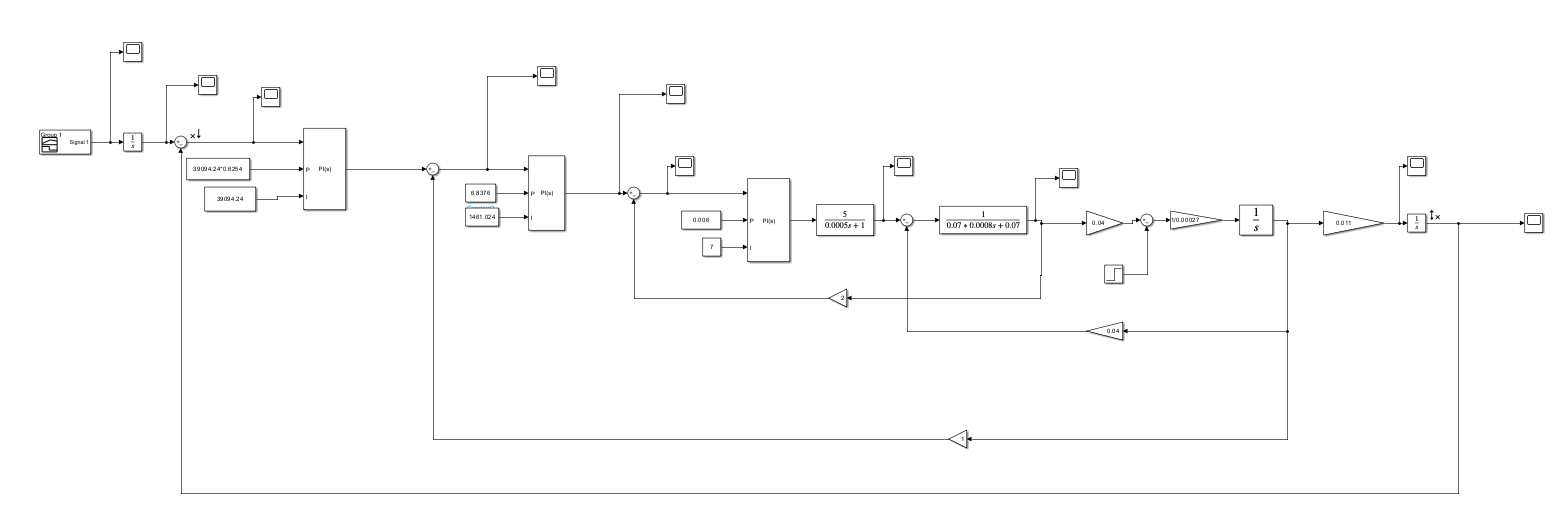
Для нормальной работы нужно чтобы: 4

Как видно условия не выполнено, поэтому выбираем «ПИ» регулятор контура положения

0

Что удовлетворяет требованиям задачи М<1.3

Результаты расчета контура положения:

Рисунок 37. Структурная схема контура положения с контуром скорости и тока

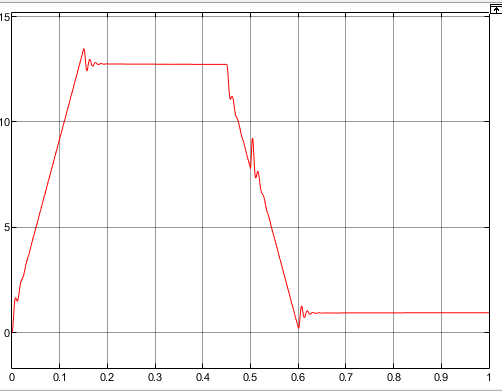


Рисунок 38. а) **Напряжения** в якоре. Отработка трапециевидного сигнала в контуре положения в момент времени и подача на вал двигателя момент в 0.5 сек.

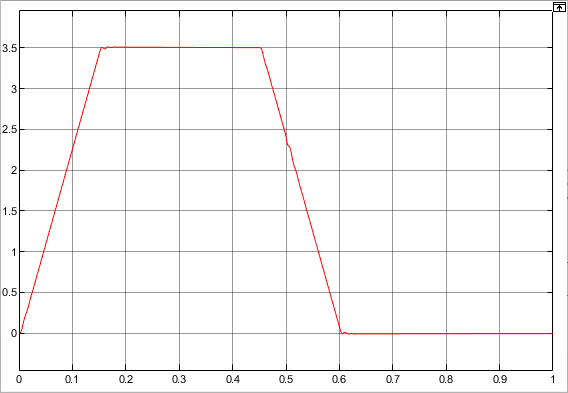


Рисунок 40. **Скорость** вала двигателя; Отработка трапециевидного сигнала в контуре положения в момент времени и подача на вал двигателя момент в 1 сек.

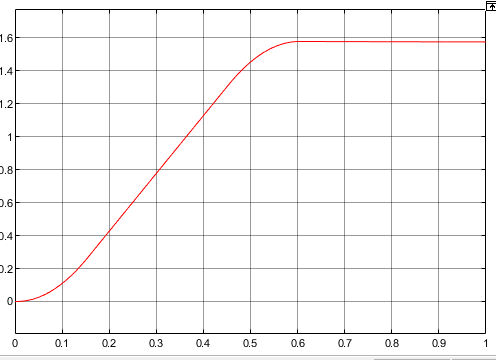


Рисунок 41. Отработка входного сигнала контура положения

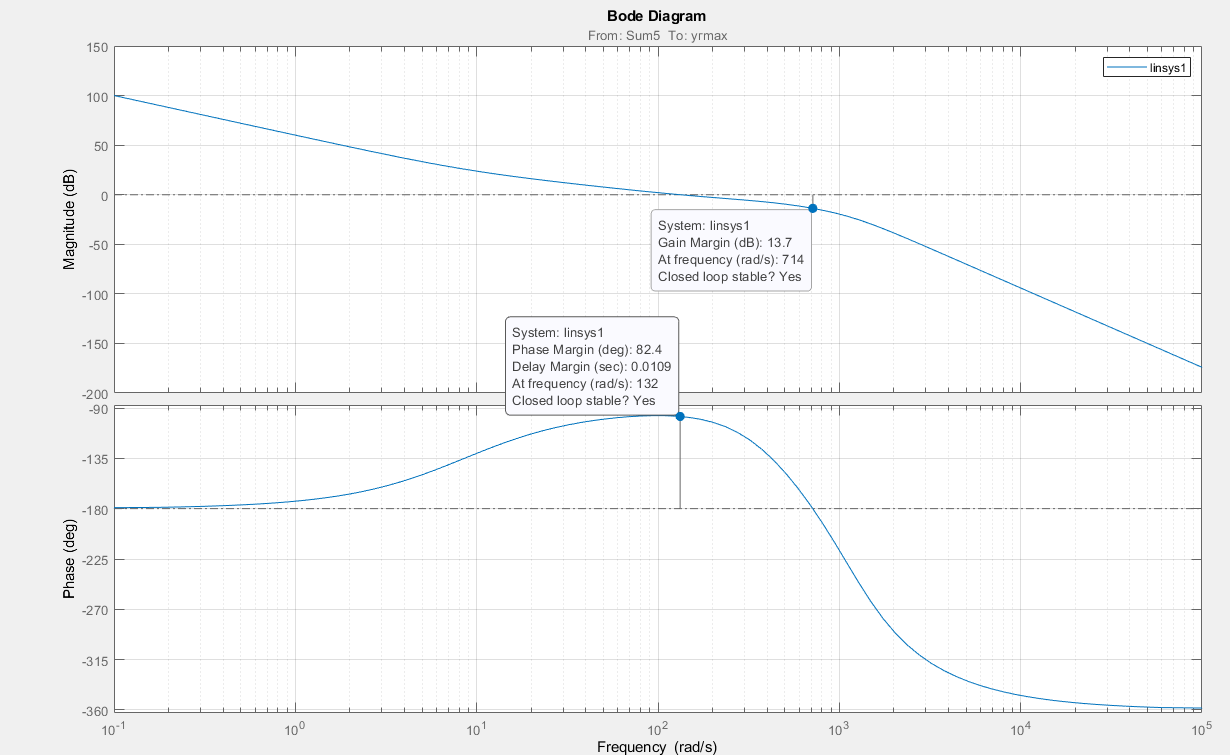


Рисунок 42. **ЛАФЧХ** для контура положения

## 6.3 Создание моделей звеньев манипулятора

Разработка 3D-моделей звеньев манипулятора была проведена в программе «КОМПАС-3D». Основные размеры звеньев были получены в результате метрического синтеза длин звеньев. Внешний вид деталей и сборочных единиц [12] приведены в Приложении В\

Для обеспечения быстрого и надежного захвата письма выберем вакуумный захват. Толщина письма не подлежит строгому нормированию, поэтому точно позиционировать схват для захвата детали без сторонних датчиков крайне трудно обеспечить. Поэтому для подтверждения захвата письма может служить сигнал с датчика вакуума, который встраивается в систему для контроля уровня вакуума между присоской схвата и захватываемым объектом. Этот сигнал может быть использован в системе управления роботом как сигнал для начала перемещения письма. Для проектирования вакуумного захвата необходимо рассчитать требуемый уровень вакуума для предотвращения отрыва схвата от поверхности объекта. Силу вакуумного притяжения можно определить по формуле[5]:

где – коэффициент, учитывающий возможное изменение атмосферного давления и свойств уплотнителя;

– давление воздуха внутри камеры присосов;

– эффективная площадь захвата;

В зависимости от расчетной модели расчетное значение силы удержания детали схватом можно вычислить по различным выражениям (Рисунок 43).

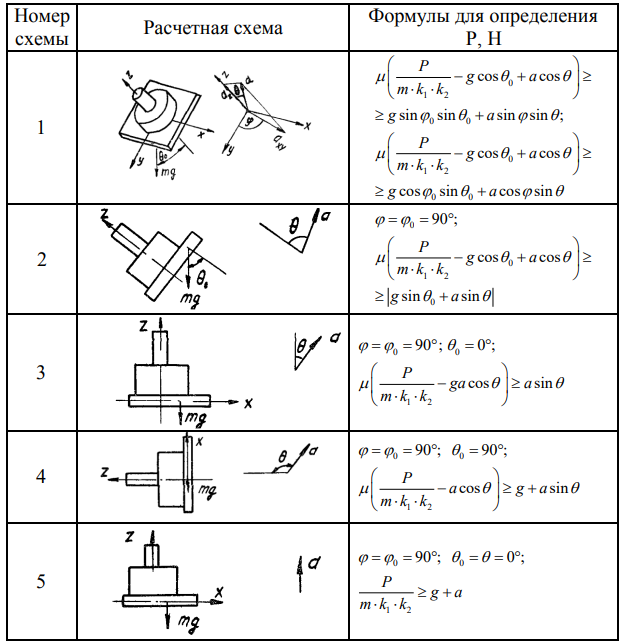


Рисунок 43 – Проверка удержания детали вакуумным схватом [6]

С помощью расчетной схемы №5 определю минимальное необходимое значение силы вакуумного притяжения:

где m – масса заготовки, g – ускорение свободного падения, а – ускорение захвата, Θ – угол между вектором ускорения и осью Z, – коэффициент запаса, коэффициент, учитывающий смещение точки приложения подъемной силы и центра тяжести заготовки.

Тогда минимальное значение силы вакуумного притяжения будет равно:

Тогда по формуле (53) найду значение остаточного давления

Для быстрого создания вакуума выберу генератор вакуума на основе эжектора.

Эжектор – устройство, в котором происходит передача кинетической энергии от одной среды, движущейся с большой скоростью, к другой. На вход в эжектор подается сжатый воздух, под действием которого в области между поверхностью объекта и схвата образуется вакуум. Схема работы эжектора представлена на Рисунке 44.

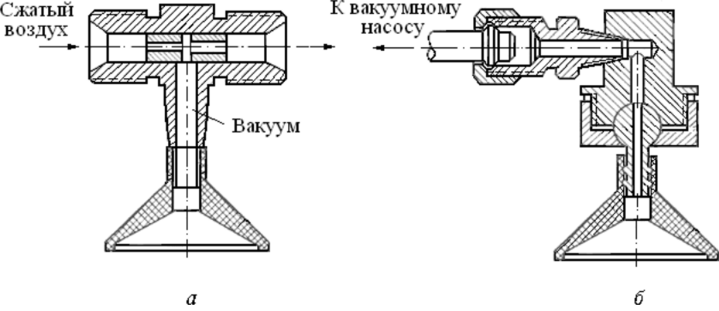


Рисунок 44 – Принцип работы эжектора

Для создания системы управления вакуумным захватом помимо эжектора и присоски необходимо подобрать распределитель, фильтр, глушитель, компрессор

Выбор пневмооборудования осуществлялся по каталогу фирмы SMC [7]. На основе предъявляемых к системе требований выберу эжектор, характеристика которого приведена на Рисунке 46.

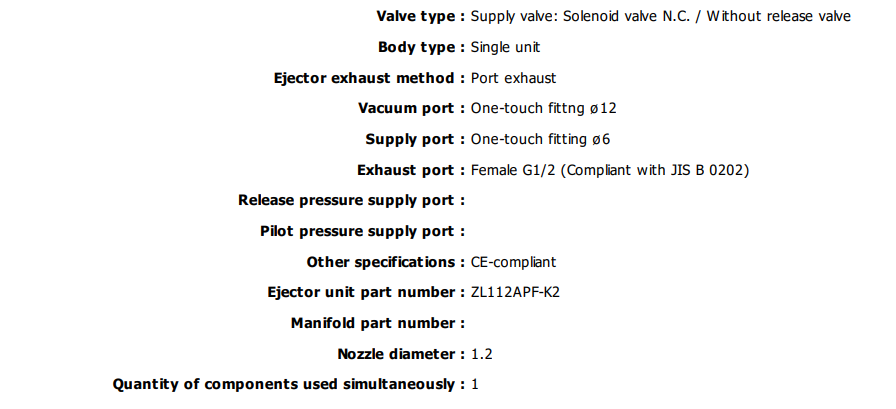


Рисунок 46 – Характеристики эжектора ZL112APF-K2

Результаты расчета времени создания необходимого уровня вакуума приведены на Рисунке 47.

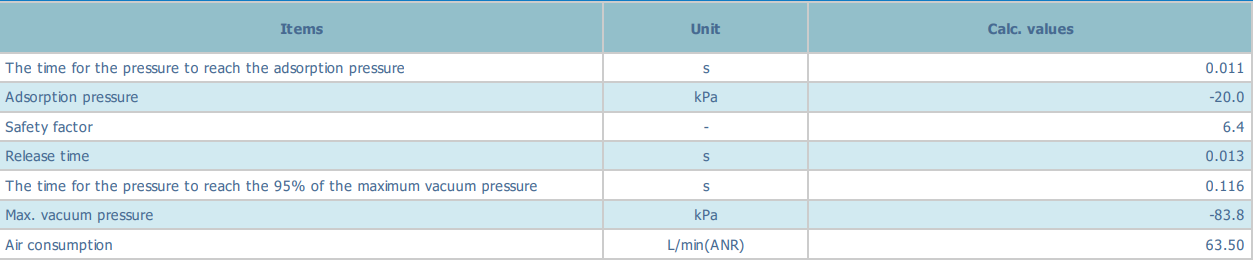


Рисунок 47 – Расчетное время создания вакуума

Внешний вид захватного устройства после соединения всех элементов вместе будет иметь вид, представленный на Рисунке 48.

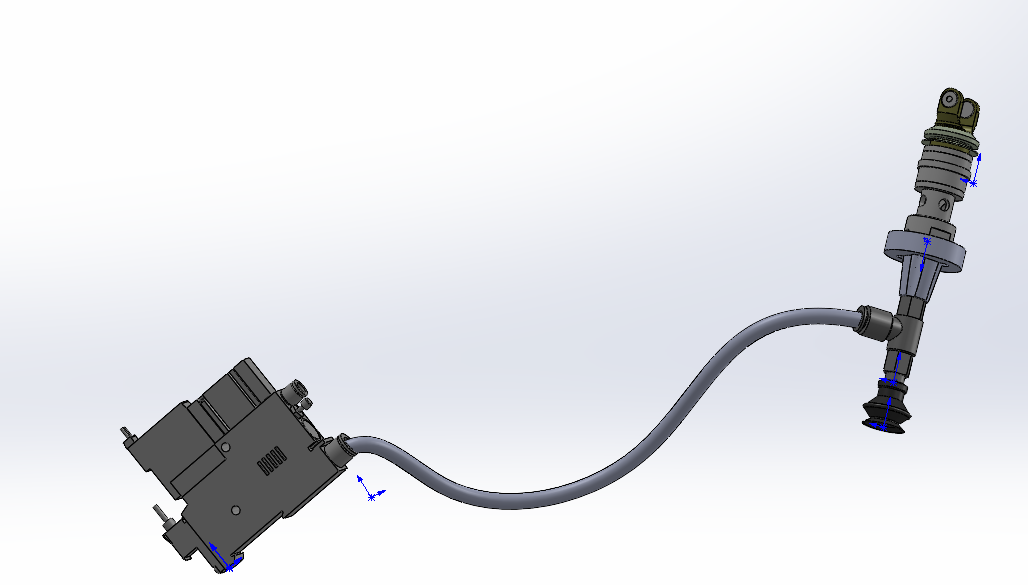


Рисунок 48 – Модель вакуумного захватного устройства вместе с эжектором

# 7. Использование алгоритмов технического зрения

Для работы системы необходимо определять центры писем и распознавать их ориентацию на конвейерной ленте с помощью библиотеки компьютерного зрения OpenCV. В OpenCV имеется функция, которая пытается найти прямоугольник максимального размера, который может вписаться в заданный замкнутый контур. Надо заметить, что эта функция не определяет является ли контур прямоугольным, она пытается вписать в него прямоугольник оптимальным способом.

Следующий шаг — разберемся с паразитными микроконтурами, которые мы обнаружили на объектах. Избавиться от них можно, вычислив площадь занимаемую этими контурами, а затем просто отсечь контуры с маленькой площадью.

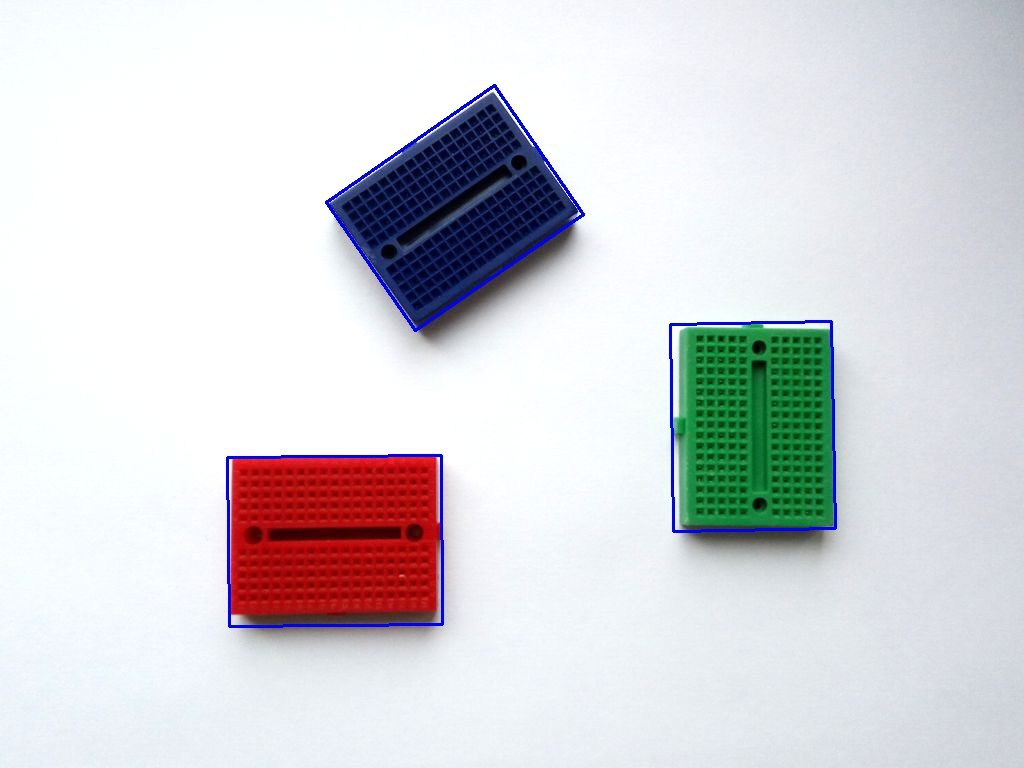


Рисунок 49 - Результат работы программы определения границ прямоугольников

Далее необходимо определить угол поворота контуров относительно горизонтали. Этот угол наклона можно найти чисто геометрически и выводить в видео-потоке. Для определения стороны письма необходимо найти микроконтуры, соответствующие опознавательным знакам на лицевой стороне конверта с письмом, в соответствии с ГОСТ. Код реализации алгоритма определения прямоугольников на изображении приведен в Приложении Б.

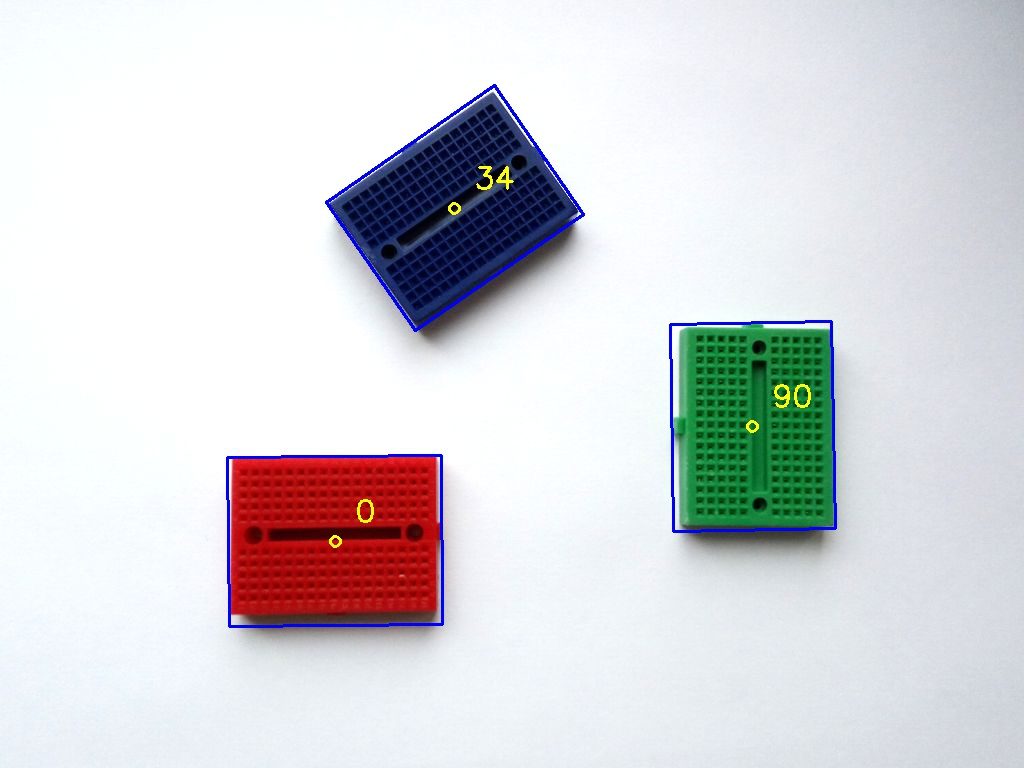


Рисунок 50 - Определение угла наклона прямоугольников

После определения положения цента письма в системе координат камеры необходимо получить координаты этой точки в мировой системе отсчета с помощью внутренней матрицы преобразования камеры. Для получения данной матрицы необходимо провести калибровку камеры. Реализация кода для калибровки камеры приведена в Приложении Б.

# 8. Моделирование работы РТК

Для проверки работоспособности алгоритмов управления необходимо провести моделирование работы системы в среде Simulink с помощью пакета Simscape Multibody и Robot System Toolbox.

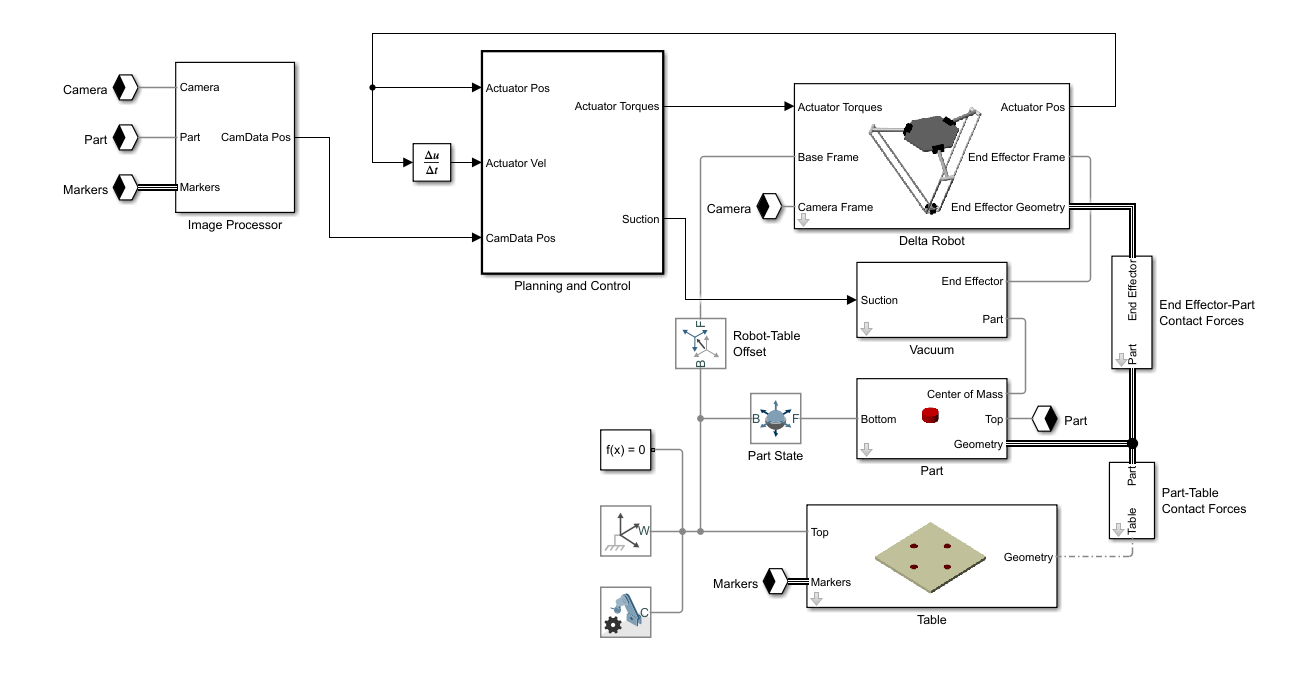


Рисунок 51 - Общий вид системы

Система состоит из основных функциональных блоков, каждый из которых решает

В данном блоке происходит обработка изображения с камеры и преобразование координат к мировой системе отсчета. Данный блок на вход принимает координаты на изображении для центров писем и текущее положение схвата, а на выходе – преобразованные значение центров писем и желаемое положение захватного устройства.

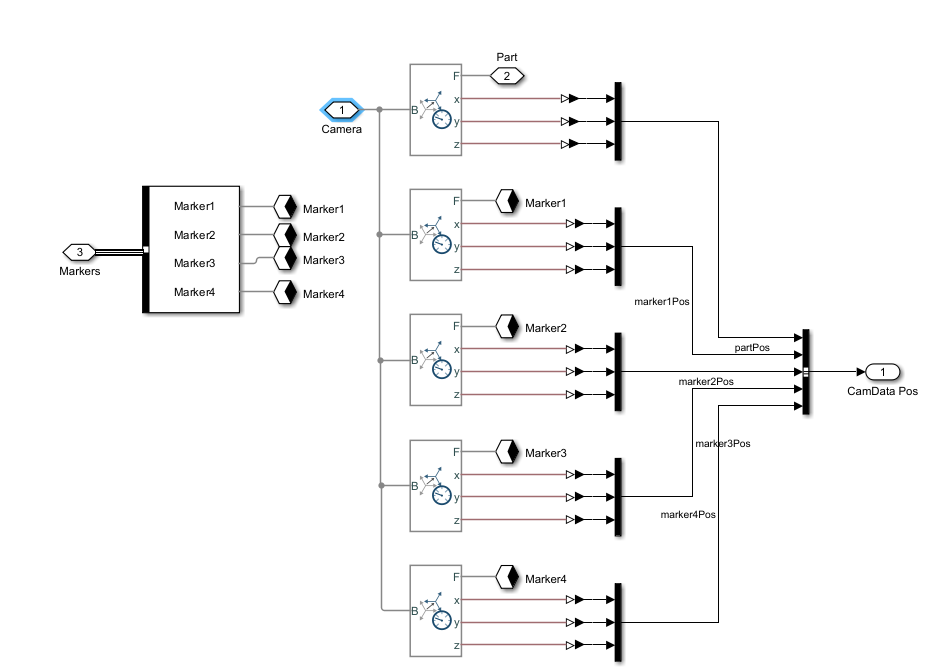


Рисунок 52 - Блок Image processor

В данном блоке происходит планирование траектории движения манипулятора между двумя заданными точками. Данная траектория определяется полиномом 3-ей степени с такими условиями, чтобы скорость и ускорение движения в начальной и конечной точках траектории были нулевыми. Также в планере маршрутов вводится временная зависимость при движении по траектории, которая апроксимируется полиномом пятой степени. Код, реализующий планер маршрутов, представлен в Приложении.

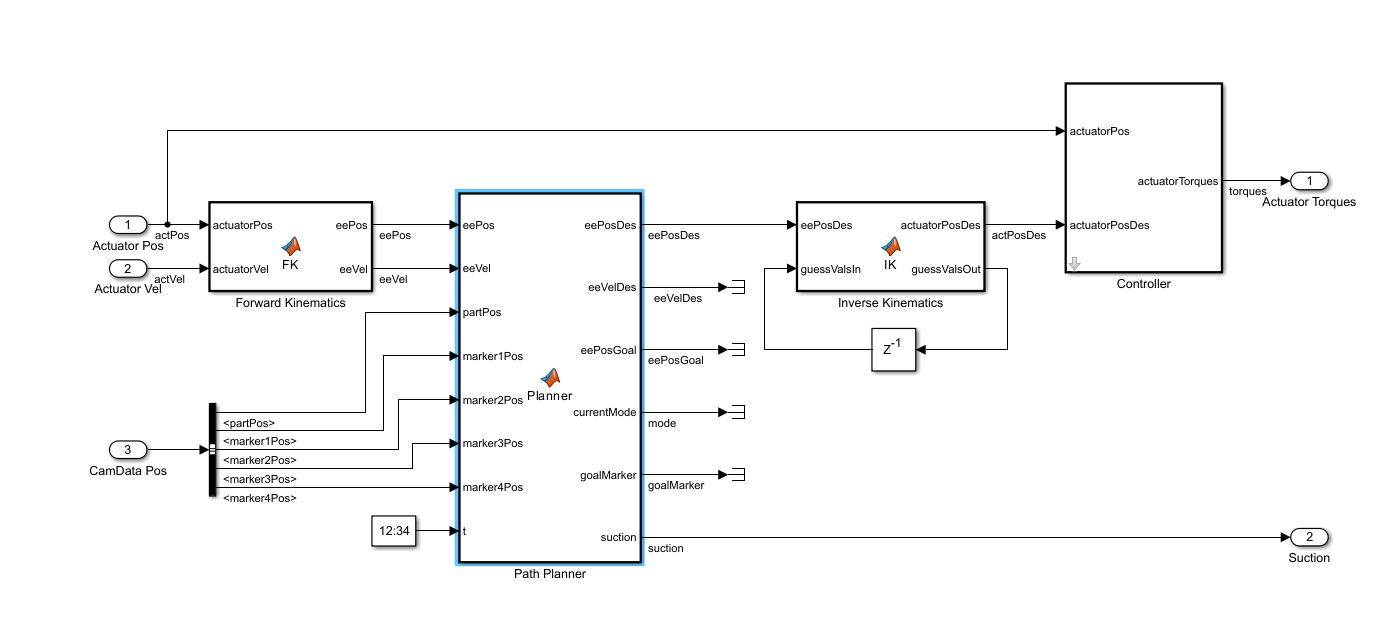


Рисунок 63 - Блок Planning and Control

На вход блока поступают текущие положение и скорость вращения двигателя, после чего эти данные поступают в блок решения прямой задачи кинематики. Далее полученные значения положения и скорости передаются в блок планирования траектории, после чего полученные значения поступают в блок решения обратной задачи кинематики.

В данном блоке вычисленные управляющие моменты передаются на степени подвижности манипулятора, реализованные с помощью инструментов пакета Simscape Multibody и звенья манипулятора, которые имеют массы и моменты инерции как у спроектированных звеньев. Верхняя платформа является неподвижной, нижняя подвижной, платформы в процессе движения всегда остаются параллельными друг другу.

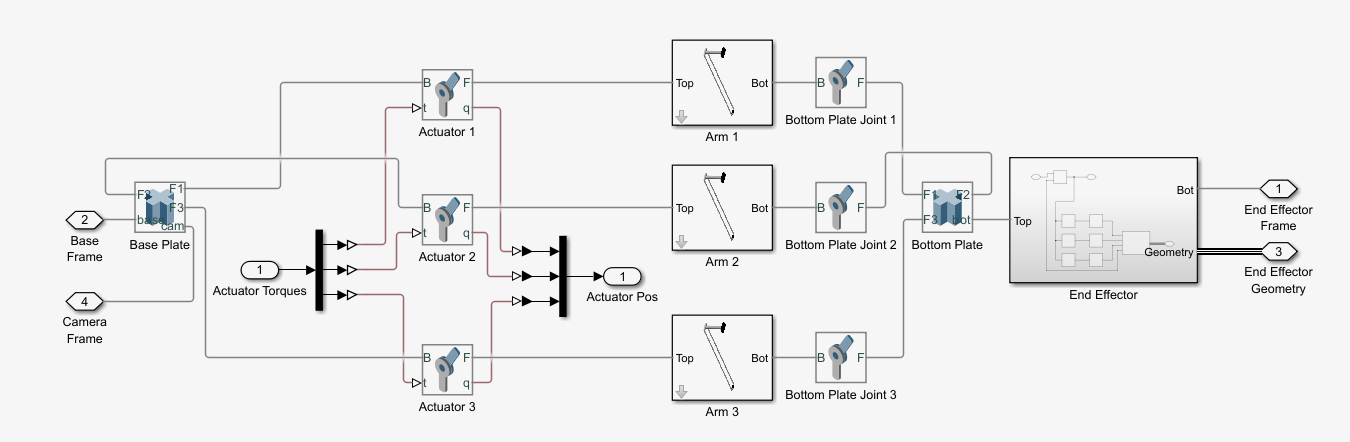


Рисунок 54 - Блок Delta Robot

В данном блоке задаются геометрические соотношения между перемещаемым телом и столом, на котором это тело расположено.

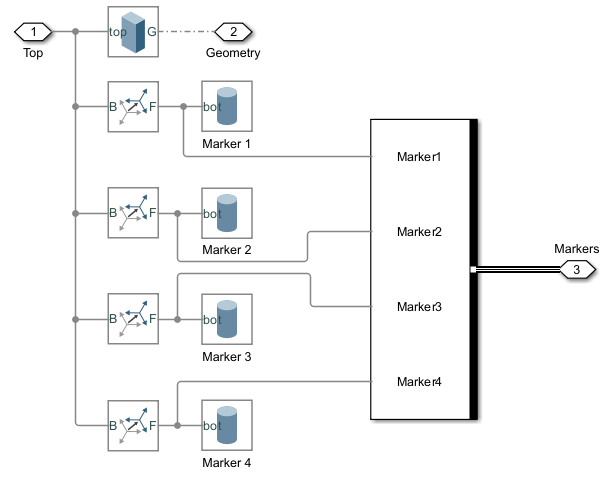


Рисунок 55 – Блок Table

В данном блоке задается геометрические параметры перемещаемого тела, его расположение в пространстве. Установленные закономерности геометрии преобразуются в вектор и передаются на дальнейшие этапы симуляции.

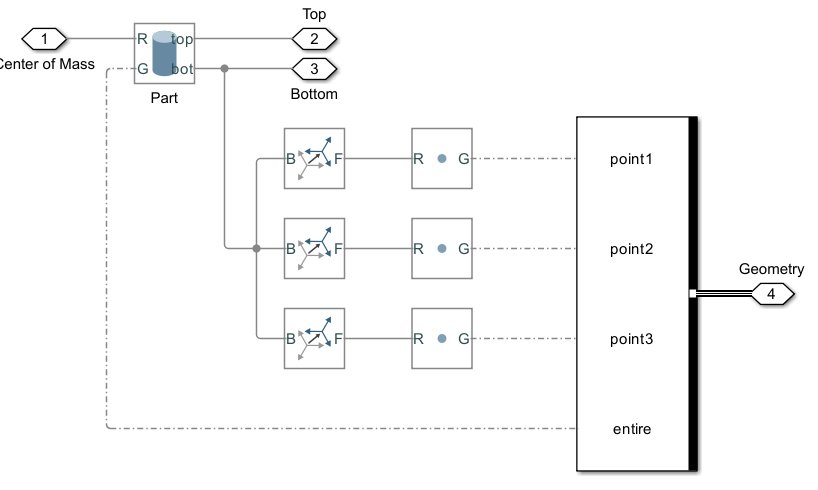


Рисунок 56 - Блок Part

В результате моделирования работы системы были получены графики изменения углов поворота степеней подвижности манипулятора, графики изменения управляющих моментов, которые вычисляются блоком Planning and Control для достижения требуемого положения манипулятора. Также была получена симуляция работы робота, которая позволяет визуально оценить правильность работы алгоритмов управления.

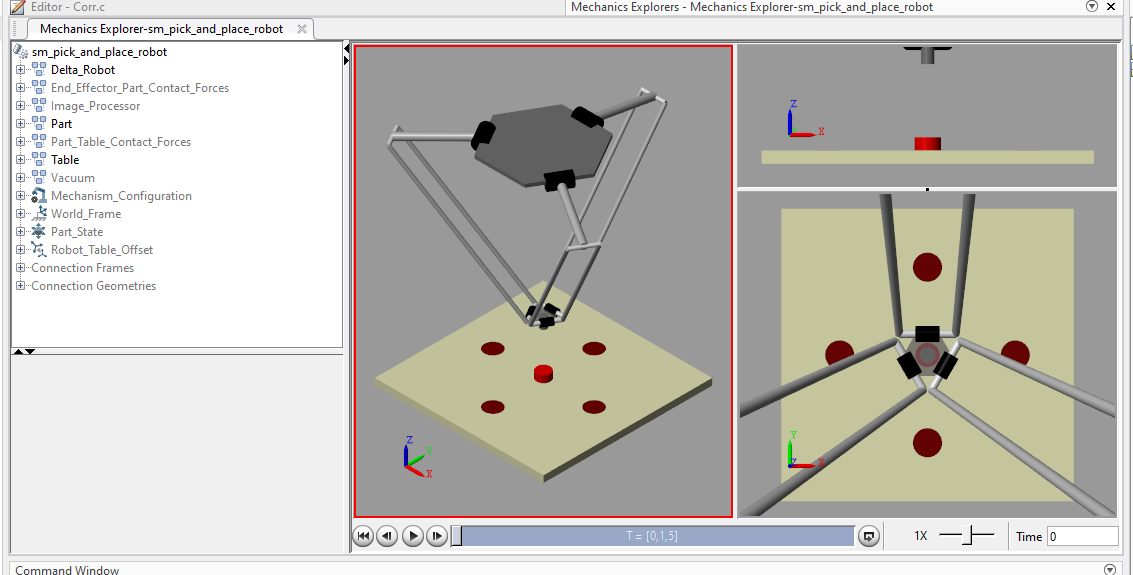


Рисунок 57 – Моделирование работы системы

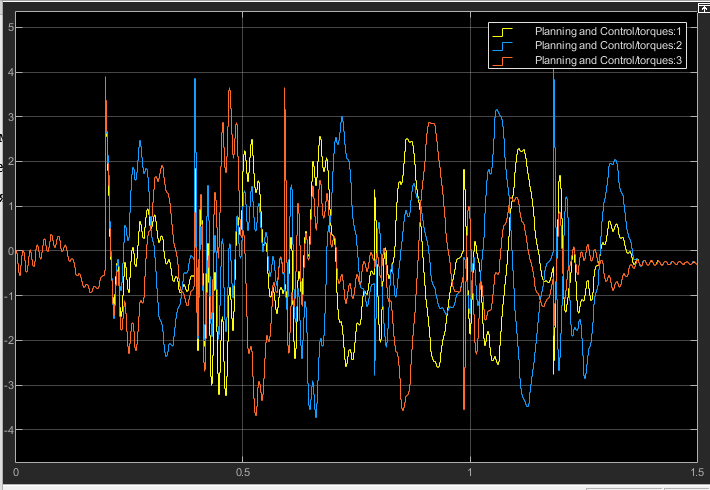


Рисунок 58 - Графики изменения управляющих моментов

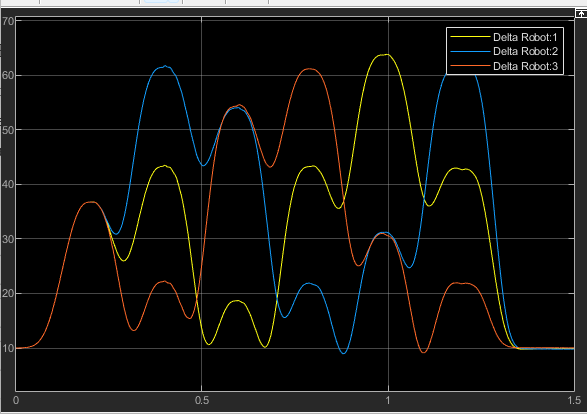


Рисунок 59 - Графики изменения обобщенных координат манипулятора

ЗАКЛЮЧЕНИЕ

В рамках научно-исследовательской работы была рассмотрена актуальность выявленной проблемы, проведен анализ ее существующих на данный момент решений, была рассмотрена структура управления системы, проведен метрический синтез звеньев дельта манипулятора, решены прямые и обратные задачи по положению и скорости для данной кинематической схемы манипулятора, определены необходимый крутящий момент и скорость вращения электродвигателя, подобраны электродвигатели по рассчитанным параметрам, спроектированы модели звеньев манипулятора, подобрано пневматическое захватное устройство. Для выполнения выпускной квалификационной работы необходимо решить следующие задачи:

1. Реализовать алгоритм определения координат центра письма при помощи камеры;
2. Определить необходимую скорость движения конвейерной ленты и подобрать соответствующий конвейер;
3. Разработать систему распознавания сторон письма при помощи камеры;
4. Разработать систему управления приводами робота;
5. Проверить разработанную систему управления приводами на устойчивость;

6. Провести моделирование работы системы в программе RoboDK;

7. Проверить, выполняются ли необходимые требования по быстродействию и точности работы системы.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Доклад о финансово-хозяйственной деятельности федерального государственного унитарного предприятия «Почта России» за 2018 год URL: https://www.pochta.ru/company/annual-reports.
2. ГОСТ Р 51506-99 Конверты почтовые. Технические требования. Методы контроля
3. Delta robot 3 axis + 1 rotation catalog URL: https://industrial.omron.eu/en/products/x-delta-3+1#all.
4. ABB IRB 360 Industrial Robotic Portfolio URL: https://new.abb.com/products/robotics/industrial-robots/irb-360.
5. Захватные устройства промышленных роботов. Учебное пособие / К.А. Украженко, Ю.В. Янчевский, А.А. Кулебякин, А.Ю. Торопов. – Ярославль : Изд-во ЯГТУ, 2007. - 83 с. ISBN 5-230-20645-
6. SMC Web Catalog vacuum equipment (Vacuum Generators/Vacuum Suction Cups/Other) URL: https://mssc.smcworld.com/vatsmss/?language=en.
7. Maxon Product Range 2022/23 URL: <https://online.flippingbook.com/view/1042987/296/>
8. Заметки о дельта-роботе URL: <https://habr.com/ru/post/580970/>
9. ОЦЕНКА БЛИЗОСТИ К ОСОБЫМ ПОЛОЖЕНИЯМ МЕХАНИЗМОВ ПАРАЛЛЕЛЬНОЙ СТРУКТУРЫ ПУТЕМ ДИФФЕРЕНЦИРОВАНИЯ УРАВНЕНИЙ СВЯЗИ/ П.А. Ларюшкин: УДК 621.01 DOI: 10.18698/0236-3941-2019-1-71-83.
10. Носова Н.Ю. Синтез механизмов параллельной структуры с кинематической развязкой / Н.Ю. Носова, В.А. Глазунов, С.В. Палочкин, А.Н. Терехова // Проблемы машиностроения и надѐжности машин. – 2014. – №5. – С. 34 – 40.

11. Документация MATLAB на русском языке. URL: https://docs.exponenta.ru/documentation-center.html (дата обращения 19.05 2023)

12. Дунаев П.Ф., Леликов О.П. «Конструирование узлов и деталей машин» / под ред. О.А. Ряховского – М.: Изд-во МГТУ им. Н. Э. Баумана, 2017.

ПРИЛОЖЕНИЕ А

**Текст программы «PZK»**

%Находит решение прямой задачи кинематически для дельта-робота

%Возвращает координаты точек L1, L2, L3, необходимые для построения робота и

%координаты точки V

function [L1, L2, L3, V] = PZK(Tetta1, Tetta2, Tetta3)

global R\_l R\_r VM OQ

%Расчёт координат концов рычагов

x\_L1 = 0;

y\_L1 = -OQ-R\_l\*cosd(Tetta1-180);

z\_L1 = -R\_l\*sind(Tetta1-180);

z\_L2 = -R\_l\*sind(Tetta2-180);

z\_L3 = -R\_l\*sind(Tetta3-180);

y\_L2S = -OQ-R\_l\*cosd(Tetta2-180);

y\_L3S = -OQ-R\_l\*cosd(Tetta3-180);

x\_L2 = y\_L2S\*sind(120);

y\_L2 = y\_L2S\*cosd(120);

x\_L3 = y\_L3S\*sind(240);

y\_L3 = y\_L3S\*cosd(240);

%Расчёт координат центров сфер (сдвинутых концов рычагов)

x\_P1 = x\_L1;

y\_P1 = y\_L1+VM;

z\_P1 = z\_L1;

x\_P2 = x\_L2+VM\*cosd(30);

y\_P2 = y\_L2-VM\*sind(30);

z\_P2 = z\_L2;

x\_P3 = x\_L3-VM\*cosd(30);

y\_P3 = y\_L3-VM\*sind(30);

z\_P3 = z\_L3;

%Расчёт точки пересечения сфер

w1 = R\_r^2-x\_P1^2-y\_P1^2-z\_P1^2;

w2 = R\_r^2-x\_P2^2-y\_P2^2-z\_P2^2;

w3 = R\_r^2-x\_P3^2-y\_P3^2-z\_P3^2;

a1 = x\_P2-x\_P1;

a2 = x\_P3-x\_P1;

b1 = y\_P2-y\_P1;

b2 = y\_P3-y\_P1;

c1 = z\_P2-z\_P1;

c2 = z\_P3-z\_P1;

d1 = (w1-w2)/2;

d2 = (w1-w3)/2;

e1 = ((b1\*c2 - b2\*c1)/(a1\*b2 - a2\*b1));

f1 = -(b1\*d2 - b2\*d1)/(a1\*b2 - a2\*b1);

e2 = -(a1\*c2 - a2\*c1)/(a1\*b2 - a2\*b1);

f2 = (a1\*d2 - a2\*d1)/(a1\*b2 - a2\*b1);

a\_KU = (e1^2 + e2^2 + 1);

b\_KU = (2\*e1\*(f1 - x\_P1) - 2\*z\_P1 + 2\*e2\*(f2 - y\_P1));

c\_KU = z\_P1^2 + (f1 - x\_P1)^2 + (f2 - y\_P1)^2 - R\_r^2;

z\_V = ((-b\_KU-sqrt(b\_KU^2-4\*a\_KU\*c\_KU)))/(2\*a\_KU);

x\_V = e1\*z\_V+f1;

y\_V = e2\*z\_V+f2;

L1 = [x\_L1, y\_L1, z\_L1];

L2 = [x\_L2, y\_L2, z\_L2];

L3 = [x\_L3, y\_L3, z\_L3];

V = [x\_V, y\_V, z\_V];

end

**Текст программы «OZK»**

%Находит решение обратной задачи кинематически для дельта-робота

function [Theta1, Theta2, Theta3] = OZK(X\_V, Y\_V, Z\_V)

%Расчёт координат точки V в системах координат, повёрнутых на 120° и 240°

%по часовой стрелке относительно основной

global cos120 sin120 cos240 sin240

X\_V\_120 = X\_V\*cos120 - Y\_V\*sin120;

Y\_V\_120 = X\_V\*sin120 + Y\_V\*cos120;

X\_V\_240 = X\_V\*cos240 - Y\_V\*sin240;

Y\_V\_240 = X\_V\*sin240 + Y\_V\*cos240;

%Расчёт углов поворота рычагов в соответствующих системах координат

Theta1 = Theta(X\_V, Y\_V, Z\_V);

Theta2 = Theta(X\_V\_120, Y\_V\_120, Z\_V);

Theta3 = Theta(X\_V\_240, Y\_V\_240, Z\_V);

End

**Текст программы «ZoneBuilder»**

%Функция построения реальной и желаемой рабочих областей

%дельта-робота

function [] = zoneBuilder()

global R\_l R\_r VM OQ cos120 sin120 cos240 sin240 %Размеры и константы

global minTheta QG phiMax varthetaMax %Ограничения

global WZ\_D WZ\_H WZ\_Z WZ\_a WZ\_d WZ\_h %Рабочая зона

global stepTheta %Шаг изменения углов

global minTheta\_real maxTheta\_real %Максимальный и минимальный углы рычагов

%при движении в реальной рабочей области

global minTheta\_wish maxTheta\_wish %Максимальный и минимальный углы рычагов

%при движении в желаемой рабочей области

global addSeg %Переменная, показывающая необходимость дополнительного

%сегмента в рабочей зоне

global status %Флаг свидетельствующий об удовлетворении условия нахождения

%желаемой рабочей зоны в реальной рабочей зоне

%поворота рычагов, которые могут быть с учёном ограничений

global Dg Hg hg %Габаритные размеры почившегося робота

maxTheta = 270; %Максимальный угол для перебора

%Задаём переменные максимальных и минимальных углов поворота рычагов,

%которые могут быть с учёном ограничений

minTheta\_real = maxTheta;

maxTheta\_real = minTheta;

minTheta\_wish = maxTheta;

maxTheta\_wish = minTheta;

%Подготовительные вычисления

gammaMin = asind(QG/R\_l);

gammaMax = 180;

%Создаём вектор с одной нулевой точкой (потом её удалим)

V\_RZ = [0, 0, 0];

%Создаём цикл перебора всех возможных комбинаций углов поворота рычагов

for Theta1 = minTheta:stepTheta:maxTheta

for Theta2 = minTheta:stepTheta:maxTheta

for Theta3 = minTheta:stepTheta:maxTheta

%Вызываем функцию, решающую ПЗК

[~,~,~,V] = PZK(Theta1, Theta2, Theta3);

X\_V = V(1); Y\_V = V(2); Z\_V = V(3);

%Вычисляем координаты точек в системах координат XOY

%X120Y120Z120 и X240Y240Z240

X\_Q1 = 0;

Y\_Q1 = -OQ;

Z\_Q1 = 0;

X\_L1 = 0;

Y\_L1 = -(OQ + R\_l\*sind(Theta1-90));

Z\_L1 = R\_l\*cosd(Theta1-90);

X\_M1 = X\_V;

Y\_M1 = Y\_V - VM;

Z\_M1 = Z\_V;

X\_V\_120 = X\_V\*cos120 - Y\_V\*sin120;

Y\_V\_120 = X\_V\*sin120 + Y\_V\*cos120;

Z\_V\_120 = Z\_V;

X\_Q2\_120 = 0;

Y\_Q2\_120 = -OQ;

Z\_Q2\_120 = 0;

X\_L2\_120 = 0;

Y\_L2\_120 = -(OQ + R\_l\*sind(Theta2-90));

Z\_L2\_120 = R\_l\*cosd(Theta2-90);

X\_M2\_120 = X\_V\_120;

Y\_M2\_120 = Y\_V\_120 - VM;

Z\_M2\_120 = Z\_V\_120;

X\_V\_240 = X\_V\*cos240 - Y\_V\*sin240;

Y\_V\_240 = X\_V\*sin240 + Y\_V\*cos240;

Z\_V\_240 = Z\_V;

X\_Q3\_240 = 0;

Y\_Q3\_240 = -OQ;

Z\_Q3\_240 = 0;

X\_L3\_240 = 0;

Y\_L3\_240 = -(OQ + R\_l\*sind(Theta3-90));

Z\_L3\_240 = R\_l\*cosd(Theta3-90);

X\_M3\_240 = X\_V\_240;

Y\_M3\_240 = Y\_V\_240 - VM;

Z\_M3\_240 = Z\_V\_240;

%Собираем это в вектора

Q1 = [X\_Q1, Y\_Q1, Z\_Q1];

L1 = [X\_L1, Y\_L1, Z\_L1];

M1 = [X\_M1, Y\_M1, Z\_M1];

V\_120 = [X\_V\_120, Y\_V\_120, Z\_V\_120];

Q2\_120 = [X\_Q2\_120, Y\_Q2\_120, Z\_Q2\_120];

L2\_120 = [X\_L2\_120, Y\_L2\_120, Z\_L2\_120];

M2\_120 = [X\_M2\_120, Y\_M2\_120, Z\_M2\_120];

V\_240 = [X\_V\_240, Y\_V\_240, Z\_V\_240];

Q3\_240 = [X\_Q3\_240, Y\_Q3\_240, Z\_Q3\_240];

L3\_240 = [X\_L3\_240, Y\_L3\_240, Z\_L3\_240];

M3\_240 = [X\_M3\_240, Y\_M3\_240, Z\_M3\_240];

%Вычисляем координаты точек N1 N2 N3

N1 = [L1(1); M1(2); V(3)];

N2\_120 = [L2\_120(1); M2\_120(2); V\_120(3)];

N3\_240 = [L3\_240(1); M3\_240(2); V\_240(3)];

%------Проверяем ограничения------%

% 1 Проверяем отсутствие пересечений

gamma1 = 180-acosd(-L1(3)/R\_l)-asind((N1(2)-L1(2))/R\_r);

gamma2 = 180-acosd(-L2\_120(3)/R\_l)-asind((N2\_120(2)-L2\_120(2))/R\_r);

gamma3 = 180-acosd(-L3\_240(3)/R\_l)-asind((N3\_240(2)-L3\_240(2))/R\_r);

if (gamma1 < gammaMax) && (gamma1 > gammaMin) && (gamma2 < gammaMax) && (gamma2 > gammaMin) && (gamma3 < gammaMax) && (gamma3 > gammaMin)

% 2 Проверяем, могут ли трёхподвижные шарниры позволить механизму принять такое положение

varphi1 = asind(abs(V(1))/R\_r);

varphi2 = asind(abs(V\_120(1))/R\_r);

varphi3 = asind(abs(V\_240(1))/R\_r);

if (varphi1<phiMax) && (varphi2<phiMax) && (varphi3<phiMax)

% 3 Проверяем, не превышают ли углы давления заданное

% максимальное значение

p1 = [M1(1)-L1(1); M1(2)-L1(2); M1(3)-L1(3)];

q1 = [0; L1(3)-Q1(3); Q1(2)-L1(2)];

vartheta1 = acosd(abs(p1(1)\*q1(1)+p1(2)\*q1(2)+p1(3)\*q1(3))/(sqrt(p1(1)^2+p1(2)^2+p1(3)^2)\*sqrt(q1(1)^2+q1(2)^2+q1(3)^2)));

p2 = [M2\_120(1)-L2\_120(1); M2\_120(2)-L2\_120(2); M2\_120(3)-L2\_120(3)];

q2 = [0; L2\_120(3)-Q2\_120(3); Q2\_120(2)-L2\_120(2)];

vartheta2 = acosd(abs(p2(1)\*q2(1)+p2(2)\*q2(2)+p2(3)\*q2(3))/(sqrt(p2(1)^2+p2(2)^2+p2(3)^2)\*sqrt(q2(1)^2+q2(2)^2+q2(3)^2)));

p3 = [M3\_240(1)-L3\_240(1); M3\_240(2)-L3\_240(2); M3\_240(3)-L3\_240(3)];

q3 = [0; L3\_240(3)-Q3\_240(3); Q3\_240(2)-L3\_240(2)];

vartheta3 = acosd(abs(p3(1)\*q3(1)+p3(2)\*q3(2)+p3(3)\*q3(3))/(sqrt(p3(1)^2+p3(2)^2+p3(3)^2)\*sqrt(q3(1)^2+q3(2)^2+q3(3)^2)));

if (vartheta1<varthetaMax) && (vartheta2<varthetaMax) && (vartheta3<varthetaMax)

%Если все условия выполнены - добавляем точку в вектор облака точек реальной РЗ

V\_RZ = cat(1, V\_RZ, V);

%Находим максимальный и минимальный углы поворота рычагов,

%которые могут быть с учёном ограничений

minTheta\_real = min([minTheta\_real, Theta1, Theta2, Theta3]);

maxTheta\_real = max([maxTheta\_real, Theta1, Theta2, Theta3]);

end

end

end

end

end

end

%Удаляем первую нулевую точку

V\_RZ(1,:) = [];

%Находим какие точки в нашем облаке являются граничными и создаём

%группы по 3 точки, которые будем соединять треугольниками (для

%образования поверхности)

[K] = boundary(V\_RZ);

%Находим точки для построения цилиндра

%Крышка

t = linspace(0, 2\*pi, 50);

Xo1 = (WZ\_D/2) \* cos(t);

Yo1 = (WZ\_D/2) \* sin(t);

Zo1 = ones(1, 50)\*(WZ\_Z+WZ\_H);

Zo2 = ones(1, 50)\*WZ\_Z;

%Цилиндрическая поверхность

[Xc,Yc,Zc] = cylinder((WZ\_D/2), 50);

Zc = Zc\*WZ\_H+WZ\_Z;

%Находим точки для построения параллелепипеда

Xp = [0 WZ\_a WZ\_a 0 0 WZ\_a WZ\_a 0];

Yp = [0 0 WZ\_a WZ\_a 0 0 WZ\_a WZ\_a] ;

Zp = Xp'\*Xp\*Yp'\*Yp/WZ\_a^2/WZ\_a^2\*WZ\_H/2 + WZ\_Z;

Xp = Xp - WZ\_a/2;

Yp = Yp - WZ\_a/2;

%Отрисовка рабочих зон

%Отображаем реальную РЗ

p1 = trisurf(K,V\_RZ(:,1),V\_RZ(:,2),V\_RZ(:,3),'Facecolor','red', 'FaceAlpha',0.2);

hold('on');

%Отображаем крышку и боковую поверхность цилиндра

p2 = patch(Xo1, Yo1, Zo1, 'c', 'FaceAlpha',0.5);

p3 = surf(Xc,Yc,Zc, 'FaceColor', 'c', 'FaceAlpha',0.5);

%Отображаем параллелепипед

p4 = surf(Xp, Yp, Zp, 'FaceColor', 'g', 'FaceAlpha', 0.2);

p4.Visible = 0; %Скрываем

%Отображаем дополнительные сегменты, если нужно

if addSeg == 0

%Дно цилиндра

p5 = patch(Xo1, Yo1, Zo2, 'c', 'FaceAlpha',0.5);

end

if addSeg == 1

%Дно усечённого конуса

t = linspace(0, 2\*pi, 50);

Xo2 = (WZ\_d/2) \* cos(t);

Yo2 = (WZ\_d/2) \* sin(t);

p6 = patch(Xo2, Yo2, Zo2-WZ\_h, 'c', 'FaceAlpha',0.5);

%Коническая поверхность

[Xco,Yco,Zco] = cylinder([(WZ\_d/2), (WZ\_D/2)], 50);

Zco = Zco\*WZ\_h+WZ\_Z-WZ\_h;

p7 = surf(Xco,Yco,Zco, 'FaceColor', 'c', 'FaceAlpha',0.5);

end

if addSeg == 2

%Сферическая поверхность

thetta = linspace(pi-asin((WZ\_D/2)/(WZ\_d/2)), pi, 50);

phi = linspace(0, 2\*pi, 50);

[THETTA, PHI] = meshgrid(thetta, phi);

Xsph = (WZ\_d/2)\*sin(THETTA).\*cos(PHI);

Ysph = (WZ\_d/2)\*sin(THETTA).\*sin(PHI);

Zsph = (WZ\_Z+(WZ\_d/2-WZ\_h))+(WZ\_d/2)\*cos(THETTA);

p8 = surf(Xsph,Ysph,Zsph, 'FaceColor', 'c', 'FaceAlpha',0.5);

end

%Настраиваем отображение

hold('off');

axis('equal');

xlabel('x, мм');

ylabel('y, мм');

zlabel('z, мм');

title('Рабочие зоны механизма');

%legend('Реальная РЗ', 'Требуемая РЗ');

view([-30 , 20]);

grid('on');

%Проверяем, подходит ли данная конфигурация робота под желаемую рабочую

%зону

%Формируем массив точек для проверки

%Точки крышки цилиндра

[t, r] = meshgrid(linspace(-pi/2, -pi/2-pi/3, 10), linspace(0, WZ\_D/2,10));

[Xtc, Ytc] = pol2cart(t,r);

Ztc = ones(10, 10)\*(WZ\_Z+WZ\_H);

%Точки боковой поверхности цилиндра

[t,r] = meshgrid(linspace(-pi/2, -pi/2-pi/3, 10), WZ\_D/2);

[Xmv, Ymv] = pol2cart(t,r);

mv = linspace(WZ\_Z, WZ\_Z+WZ\_H, 10);

for k = 2:9

Xss((k-1)\*10-9:(k-1)\*10) = Xmv;

Yss((k-1)\*10-9:(k-1)\*10) = Ymv;

Zss((k-1)\*10-9:(k-1)\*10) = ones(1, 10)\*mv(k);

end

if addSeg == 0

%Точки дна цилиндра

[t, r] = meshgrid(linspace(-pi/2, -pi/2-pi/3, 10), linspace(0, WZ\_D/2,10));

[Xbc, Ybc] = pol2cart(t,r);

Zbc = ones(10, 10)\*(WZ\_Z);

%Формируем итоговый вектор точек

PointsToBeChecked\_X = [reshape(Xtc',1,[]), reshape(Xss',1,[]), reshape(Xbc',1,[])];

PointsToBeChecked\_Y = [reshape(Ytc',1,[]), reshape(Yss',1,[]), reshape(Ybc',1,[])];

PointsToBeChecked\_Z = [reshape(Ztc',1,[]), reshape(Zss',1,[]), reshape(Zbc',1,[])];

end

if addSeg == 1

%Точки дна конуса

[t, r] = meshgrid(linspace(-pi/2, -pi/2-pi/3, 10), linspace(0, WZ\_d/2,10));

[Xbco, Ybco] = pol2cart(t,r);

Zbco = ones(10, 10)\*(WZ\_Z-WZ\_h);

%Точки боковой поверхности конуса

mv = linspace(WZ\_Z-WZ\_h, WZ\_Z, 10);

mvD = linspace(WZ\_d, WZ\_D, 10);

for k = 2:10

[t,r] = meshgrid(linspace(-pi/2, -pi/2-pi/3, 10), mvD(k)/2);

[Xmv, Ymv] = pol2cart(t,r);

Xssco((k-1)\*10-9:(k-1)\*10) = Xmv;

Yssco((k-1)\*10-9:(k-1)\*10) = Ymv;

Zssco((k-1)\*10-9:(k-1)\*10) = ones(1, 10)\*mv(k);

end

%Формируем итоговый вектор точек

PointsToBeChecked\_X = [reshape(Xtc',1,[]), reshape(Xss',1,[]), reshape(Xbco',1,[]), reshape(Xssco',1,[])];

PointsToBeChecked\_Y = [reshape(Ytc',1,[]), reshape(Yss',1,[]), reshape(Ybco',1,[]), reshape(Yssco',1,[])];

PointsToBeChecked\_Z = [reshape(Ztc',1,[]), reshape(Zss',1,[]), reshape(Zbco',1,[]), reshape(Zssco',1,[])];

end

if addSeg == 2

%Точки сферы

thetta = linspace(pi-asin((WZ\_D/2)/(WZ\_d/2)), pi, 10);

phi = linspace(-pi/2, -pi/2-pi/3, 10);

[THETTA, PHI] = meshgrid(thetta, phi);

Xsph = (WZ\_d/2)\*sin(THETTA).\*cos(PHI);

Ysph = (WZ\_d/2)\*sin(THETTA).\*sin(PHI);

Zsph = (WZ\_Z+(WZ\_d/2-WZ\_h))+(WZ\_d/2)\*cos(THETTA);

%Формируем итоговый вектор точек

PointsToBeChecked\_X = [reshape(Xtc',1,[]), reshape(Xss',1,[]), reshape(Xsph',1,[])];

PointsToBeChecked\_Y = [reshape(Ytc',1,[]), reshape(Yss',1,[]), reshape(Ysph',1,[])];

PointsToBeChecked\_Z = [reshape(Ztc',1,[]), reshape(Zss',1,[]), reshape(Zsph',1,[])];

end

% %Показываем точки для анализа

% hold('on');

% plot3(PointsToBeChecked\_X,PointsToBeChecked\_Y, PointsToBeChecked\_Z, '.r');

% hold('off');

%Проверяем каждую точку на условие нахождения её в реальной рабочей зоне

%Задаём начальный вектор статуса (ни одна точка не подходит)

stt(1:size(PointsToBeChecked\_X, 2)) = 0;

for k=1:size(PointsToBeChecked\_X, 2)

%Вызываем функцию, решающую ПЗК

vectTheta = [0, 0, 0];

[vectTheta(1),vectTheta(2), vectTheta(3)] = OZK(PointsToBeChecked\_X(k), PointsToBeChecked\_Y(k), PointsToBeChecked\_Z(k));

Theta1 = vectTheta(1); Theta2 = vectTheta(2); Theta3 = vectTheta(3);

X\_V = PointsToBeChecked\_X(k); Y\_V = PointsToBeChecked\_Y(k); Z\_V = PointsToBeChecked\_Z(k);

if imag(vectTheta) == [0, 0, 0]

%Вычисляем координаты точек в системах координат XOY

%X120Y120Z120 и X240Y240Z240

X\_Q1 = 0;

Y\_Q1 = -OQ;

Z\_Q1 = 0;

X\_L1 = 0;

Y\_L1 = -(OQ + R\_l\*sind(Theta1-90));

Z\_L1 = R\_l\*cosd(Theta1-90);

X\_M1 = X\_V;

Y\_M1 = Y\_V - VM;

Z\_M1 = Z\_V;

X\_V\_120 = X\_V\*cos120 - Y\_V\*sin120;

Y\_V\_120 = X\_V\*sin120 + Y\_V\*cos120;

Z\_V\_120 = Z\_V;

X\_Q2\_120 = 0;

Y\_Q2\_120 = -OQ;

Z\_Q2\_120 = 0;

X\_L2\_120 = 0;

Y\_L2\_120 = -(OQ + R\_l\*sind(Theta2-90));

Z\_L2\_120 = R\_l\*cosd(Theta2-90);

X\_M2\_120 = X\_V\_120;

Y\_M2\_120 = Y\_V\_120 - VM;

Z\_M2\_120 = Z\_V\_120;

X\_V\_240 = X\_V\*cos240 - Y\_V\*sin240;

Y\_V\_240 = X\_V\*sin240 + Y\_V\*cos240;

Z\_V\_240 = Z\_V;

X\_Q3\_240 = 0;

Y\_Q3\_240 = -OQ;

Z\_Q3\_240 = 0;

X\_L3\_240 = 0;

Y\_L3\_240 = -(OQ + R\_l\*sind(Theta3-90));

Z\_L3\_240 = R\_l\*cosd(Theta3-90);

X\_M3\_240 = X\_V\_240;

Y\_M3\_240 = Y\_V\_240 - VM;

Z\_M3\_240 = Z\_V\_240;

%Собираем это в вектора

Q1 = [X\_Q1, Y\_Q1, Z\_Q1];

L1 = [X\_L1, Y\_L1, Z\_L1];

M1 = [X\_M1, Y\_M1, Z\_M1];

V\_120 = [X\_V\_120, Y\_V\_120, Z\_V\_120];

Q2\_120 = [X\_Q2\_120, Y\_Q2\_120, Z\_Q2\_120];

L2\_120 = [X\_L2\_120, Y\_L2\_120, Z\_L2\_120];

M2\_120 = [X\_M2\_120, Y\_M2\_120, Z\_M2\_120];

V\_240 = [X\_V\_240, Y\_V\_240, Z\_V\_240];

Q3\_240 = [X\_Q3\_240, Y\_Q3\_240, Z\_Q3\_240];

L3\_240 = [X\_L3\_240, Y\_L3\_240, Z\_L3\_240];

M3\_240 = [X\_M3\_240, Y\_M3\_240, Z\_M3\_240];

%Вычисляем координаты точек N1 N2 N3

N1 = [L1(1); M1(2); V(3)];

N2\_120 = [L2\_120(1); M2\_120(2); V\_120(3)];

N3\_240 = [L3\_240(1); M3\_240(2); V\_240(3)];

%------Проверяем ограничения------%

% 1 Проверяем отсутствие пересечений

gamma1 = 180-acosd(-L1(3)/R\_l)-asind((N1(2)-L1(2))/R\_r);

gamma2 = 180-acosd(-L2\_120(3)/R\_l)-asind((N2\_120(2)-L2\_120(2))/R\_r);

gamma3 = 180-acosd(-L3\_240(3)/R\_l)-asind((N3\_240(2)-L3\_240(2))/R\_r);

if (gamma1 < gammaMax) && (gamma1 > gammaMin) && (gamma2 < gammaMax) && (gamma2 > gammaMin) && (gamma3 < gammaMax) && (gamma3 > gammaMin)

% 2 Проверяем, могут ли трёхподвижные шарниры позволить механизму принять

% такое положение

varphi1 = asind(abs(V(1))/R\_r);

varphi2 = asind(abs(V\_120(1))/R\_r);

varphi3 = asind(abs(V\_240(1))/R\_r);

if (varphi1<phiMax) && (varphi2<phiMax) && (varphi3<phiMax)

% 3 Проверяем, не превышают ли углы давления заданное

% максимальное значение

p1 = [M1(1)-L1(1); M1(2)-L1(2); M1(3)-L1(3)];

q1 = [0; L1(3)-Q1(3); Q1(2)-L1(2)];

vartheta1 = acosd(abs(p1(1)\*q1(1)+p1(2)\*q1(2)+p1(3)\*q1(3))/(sqrt(p1(1)^2+p1(2)^2+p1(3)^2)\*sqrt(q1(1)^2+q1(2)^2+q1(3)^2)));

p2 = [M2\_120(1)-L2\_120(1); M2\_120(2)-L2\_120(2); M2\_120(3)-L2\_120(3)];

q2 = [0; L2\_120(3)-Q2\_120(3); Q2\_120(2)-L2\_120(2)];

vartheta2 = acosd(abs(p2(1)\*q2(1)+p2(2)\*q2(2)+p2(3)\*q2(3))/(sqrt(p2(1)^2+p2(2)^2+p2(3)^2)\*sqrt(q2(1)^2+q2(2)^2+q2(3)^2)));

p3 = [M3\_240(1)-L3\_240(1); M3\_240(2)-L3\_240(2); M3\_240(3)-L3\_240(3)];

q3 = [0; L3\_240(3)-Q3\_240(3); Q3\_240(2)-L3\_240(2)];

vartheta3 = acosd(abs(p3(1)\*q3(1)+p3(2)\*q3(2)+p3(3)\*q3(3))/(sqrt(p3(1)^2+p3(2)^2+p3(3)^2)\*sqrt(q3(1)^2+q3(2)^2+q3(3)^2)));

if (vartheta1<varthetaMax) && (vartheta2<varthetaMax) && (vartheta3<varthetaMax)

%Если все условия выполнены - добавляем точку в

%вектор облака точек реальной РЗ

stt(k) = 1;

%Находим максимальный и минимальный углы поворота рычагов,

%которые могут быть с учёном ограничений

minTheta\_wish = min([minTheta\_wish, Theta1, Theta2, Theta3]);

maxTheta\_wish = max([maxTheta\_wish, Theta1, Theta2, Theta3]);

end

end

end

end

end

if min(stt) == 1

status = 1;

Dg = (OQ+R\_l)\*2;

hg = R\_l\*sind(180-minTheta\_wish);

if hg < 0

hg = 0;

end

Hg = hg - WZ\_Z;

else

status = 0;

end

end

**Текст программы «Jacobian»**

%Задаём символьные переменные

syms Tetta\_1 Tetta\_2 Tetta\_3 X\_V Y\_V Z\_V

syms OQ VM R\_l R\_r F f X\_0 Y\_0 Z\_0

syms omega\_1(t) v\_VX(t) v\_VY(t) v\_VZ(t)

syms epsilon\_1 a\_VX a\_VY a\_VZ

syms v\_VX v\_VY v\_VZ

%Записываем решение обратной кинематической задачи в аналитическом виде

NL = sqrt(R\_r^2-X\_V^2);

y\_M = -VM + Y\_V;

y\_Q = -OQ;

const\_1 = y\_M - y\_Q;

NQ = sqrt(const\_1^2 + Z\_V^2);

Eq1 = Tetta\_1 == 2\*pi - acos((R\_l^2 + NQ^2 - NL^2)/(2\*R\_l\*NQ))...

- acos(const\_1/NQ);

%Формируем уравнение связи

F\_1 = 2\*pi - acos((R\_l^2 + NQ^2 - NL^2)/(2\*R\_l\*NQ))...

- acos(const\_1/NQ) - Tetta\_1;

% F\_1 = - (R\_l^2 + NQ^2 - NL^2)/(2\*R\_l\*NQ)...

% - const\_1/NQ+cos(2\*pi - Tetta\_1);

%Проделываем те же операции для 2-го плеча

X\_V\_120 = X\_V\*cos(2\*pi/3) - Y\_V\*sin(2\*pi/3);

Y\_V\_120 = X\_V\*sin(2\*pi/3) + Y\_V\*cos(2\*pi/3);

Z\_V\_120 = Z\_V;

NL = sqrt(R\_r^2-X\_V\_120^2);

y\_M = -VM + Y\_V\_120;

y\_Q = -OQ;

const\_1 = y\_M - y\_Q;

NQ = sqrt(const\_1^2 + Z\_V\_120^2);

Eq2 = Tetta\_2 == 2\*pi - acos((R\_l^2 + NQ^2 - NL^2)/(2\*R\_l\*NQ))...

- acos(const\_1/NQ);

F\_2 = 2\*pi - acos((R\_l^2 + NQ^2 - NL^2)/(2\*R\_l\*NQ))...

- acos(const\_1/NQ) - Tetta\_2;

% F\_2 = - (R\_l^2 + NQ^2 - NL^2)/(2\*R\_l\*NQ)...

% - const\_1/NQ + cos(2\*pi - Tetta\_2);

%Для 3-го плеча

X\_V\_240 = X\_V\*cos(4\*pi/3) - Y\_V\*sin(4\*pi/3);

Y\_V\_240 = X\_V\*sin(4\*pi/3) + Y\_V\*cos(4\*pi/3);

Z\_V\_240 = Z\_V;

NL = sqrt(R\_r^2-X\_V\_240^2);

y\_M = -VM + Y\_V\_240;

y\_Q = -OQ;

const\_1 = y\_M - y\_Q;

NQ = sqrt(const\_1^2 + Z\_V\_240^2);

Eq3 = Tetta\_3 == 2\*pi - acos((R\_l^2 + NQ^2 - NL^2)/(2\*R\_l\*NQ))...

- acos(const\_1/NQ);

F\_3 = 2\*pi - acos((R\_l^2 + NQ^2 - NL^2)/(2\*R\_l\*NQ))...

- acos(const\_1/NQ) - Tetta\_3;

% F\_3 = - (R\_l^2 + NQ^2 - NL^2)/(2\*R\_l\*NQ)...

% - const\_1/NQ + cos(2\*pi - Tetta\_3);

%Составляем матрицы А и В

A = [diff(F\_1, X\_V), diff(F\_1, Y\_V), diff(F\_1, Z\_V);

diff(F\_2, X\_V), diff(F\_2, Y\_V), diff(F\_2, Z\_V);

diff(F\_3, X\_V), diff(F\_3, Y\_V), diff(F\_3, Z\_V)];

B = [diff(F\_1, Tetta\_1), 0, 0;

0, diff(F\_2, Tetta\_2), 0;

0, 0, diff(F\_3, Tetta\_3)];

J = -A\*(inv(B))

**Текст программы «Do\_Velocity»**

%Задаём начальное значение

wmax = 0;

F = 600; %Длина стороны треугольника основания

f = 200; %Длина стороны треугольника платформы

%Задаём входные данные

vmax = 1560;

VM = f\*sqrt(3)/6;

OQ = F\*sqrt(3)/6;

R\_l = 630; %Длина рычагов (по осям)

R\_r = 1000; %Длина штанг (по осям)

cos120 = cosd(120);

sin120 = sind(120);

cos240 = cosd(240);

sin240 = sind(240);

[TestPointsX, TestPointsY, TestPointsZ] = genTP();

%Запускаем цикл проверки в каждой точке

for k = 1:size(TestPointsX, 2)

J =

))];

w\_max\_TP = [norm(J(1, :))\*vmax, norm(J(2, :))\*vmax, norm(J(3, :))\*vmax];

%Находим максимум скорости рычагов

wmax = max([abs(w\_max\_TP), wmax]);

end

n = 30\*wmax/pi

**Текст программы «Do\_Force»**

global vmax %Максимальная требуемая скорость исполнительного звена

global Jr mk %Массоинерционных характеристики

global amax %Максимальное требуемое ускорение рабочего органа

%Задаём начальное значение

%Добавляем глобальные переменные входных и выходных данных

global WZ\_D WZ\_H WZ\_Z WZ\_d WZ\_h %Рабочая зона

global addSeg %Переменная, показывающая необходимость дополнительного

%сегмента в рабочей зоне

global dotDensity %Плотность точек

%Плотность точек анализа

dotDensity = 30;

%Параметры рабочей зоны

addSeg = 1; %Добавляемый к рабочей зоне сегмент. 0 - ничего; 1 - усечённый

%конус; %2 - часть сферы

WZ\_D = 1000; %Диаметр цилиндра рабочей зоны

WZ\_H = 400; %Высота цилиндра рабочей зоны

WZ\_Z = -1150; %Координаты дна цилиндра рабочей зоны

WZ\_a = WZ\_D\*sqrt(2)/2; %Сторона вписанного в основание цилиндра квадрата

WZ\_d = 500; %Диаметр нижней окружности усечённого конуса

WZ\_h = 200; %Высота добавляемой к РЗ части

%Максимальная требуемая скорость рабочего органа

vmax = 1560;

%Максимальное требуемое от рабочего органа ускорение

amax = 10400;

%Момент инерции рычага

Jr = 200000;

%Масса каретки

mk = 2;

%Размеры звеньев

F = 600; %Длина стороны треугольника основания

f = 200; %Длина стороны треугольника платформы

R\_l = 630; %Длина рычагов (по осям)

R\_r = 1000; %Длина штанг (по осям)

%Константы

cos120 = cosd(120);

sin120 = sind(120);

cos240 = cosd(240);

sin240 = sind(240);

VM = f\*sqrt(3)/6;

OQ = F\*sqrt(3)/6;

%Генерируем массив точек для анадлиза

[TestPointsX, TestPointsY, TestPointsZ] = genTP();

%Задаём начальные значения

Mmax = 0;

Mmin = 1e20;

%Запускаем цикл проверки в каждой точке

for k = 1:size(TestPointsX, 2)

%Находим максимальные скорости входных звеньев

w\_max\_TP = [norm(J(1, :))\*vmax, norm(J(2, :))\*vmax, norm(J(3, :))\*vmax];

%Вычисляем приведённую массу в точке V

m\_pr = mk + Jr\*(w\_max\_TP(1)/vmax)^2 + Jr\*(w\_max\_TP(2)/vmax)^2 + Jr\*(w\_max\_TP(3)/vmax)^2;

%Вычисляем силу, действующую на механизм

F = m\_pr\*amax\*(10^(-3));

J = transpose(inv(J));

M\_max\_TP = [norm(J(1, :))\*F, norm(J(2, :))\*F, norm(J(3, :))\*F];

%Находим максимальный момент среди рычагов

Mmax = max([abs(M\_max\_TP), Mmax]);

Mmin = min([abs(M\_max\_TP), Mmin]);

end

%Переводим результат в H•м

maxM = Mmax\*10^(-3)

%Вычисляем неравномерность максимального крутящего момента по рабочей зоне

uneven = Mmax/Mmin

ПРИЛОЖЕНИЕ Б

**Текст программы «Calibration»**

**import** matplotlib.pyplot **as** plt

**import** cv2 **as** cv

**import** numpy **as** np

**import** glob

*# termination criteria*

criteria **=** (cv**.**TERM\_CRITERIA\_EPS **+** cv**.**TERM\_CRITERIA\_MAX\_ITER, 30, 0.001)

*# prepare object points, like (0,0,0), (1,0,0), (2,0,0) ....,(6,5,0)*

objp **=** np**.**zeros((9**\***6, 3), np**.**float32)

objp[:,:2] **=** np**.**mgrid[0:9, 0:6]**.**T**.**reshape(**-**1,2)

*# Arrays to store object points and image points from all the images.*

objpoints **=** [] *# 3d point in real world space*

imgpoints **=** [] *# 2d points in image plane.*

images **=** glob**.**glob('image/\*.png')

plt**.**figure(figsize**=**(15, 60))

**for** i, fname **in** enumerate(images):

img **=** cv**.**imread(fname)

gray **=** cv**.**cvtColor(img, cv**.**COLOR\_BGR2GRAY)

*# Find the chess board corners*

ret, corners **=** cv**.**findChessboardCorners(gray, (9, 6), **None**)

*# If found, add object points, image points (after refining them)*

**if** ret **==** **True**:

objpoints**.**append(objp)

corners2 **=** cv**.**cornerSubPix(gray, corners, (11,11), (**-**1,**-**1), criteria)

imgpoints**.**append(corners2)

*# Draw and display the corners*

cv**.**drawChessboardCorners(img, (9, 6), corners2, ret)

img **=** cv**.**cvtColor(img, cv**.**COLOR\_BGR2RGB)

plt**.**subplot(int(np**.**ceil(len(images) **/** 2)), 2, i **+** 1)

plt**.**imshow(img)

plt**.**show()

ret, mtx, dist, rvecs, tvecs **=** cv**.**calibrateCamera(objpoints, imgpoints, gray**.**shape[::**-**1], **None**, **None**)

print(mtx, dist, rvecs, tvecs, sep**=**'\n\n')

**Текст программы «Find\_countours»**

import sys

import numpy as np

import cv2 as cv

import math

hsv\_min = np.array((0, 54, 5), np.uint8)

hsv\_max = np.array((187, 255, 253), np.uint8)

color\_blue = (255,0,0)

color\_yellow = (0,255,255)

if \_\_name\_\_ == '\_\_main\_\_':

fn = 'image2.jpg' # имя файла, который будем анализировать

img = cv.imread(fn)

hsv = cv.cvtColor( img, cv.COLOR\_BGR2HSV ) # меняем цветовую модель с BGR на HSV

thresh = cv.inRange( hsv, hsv\_min, hsv\_max ) # применяем цветовой фильтр

\_, contours0, hierarchy = cv.findContours( thresh.copy(), cv.RETR\_TREE, cv.CHAIN\_APPROX\_SIMPLE)

# перебираем все найденные контуры в цикле

for cnt in contours0:

rect = cv.minAreaRect(cnt) # пытаемся вписать прямоугольник

box = cv.boxPoints(rect) # поиск четырех вершин прямоугольника

box = np.int0(box) # округление координат

center = (int(rect[0][0]),int(rect[0][1]))

area = int(rect[1][0]\*rect[1][1]) # вычисление площади

# вычисление координат двух векторов, являющихся сторонам прямоугольника

edge1 = np.int0((box[1][0] - box[0][0],box[1][1] - box[0][1]))

edge2 = np.int0((box[2][0] - box[1][0], box[2][1] - box[1][1]))

# выясняем какой вектор больше

usedEdge = edge1

if cv.norm(edge2) > cv.norm(edge1):

usedEdge = edge2

reference = (1,0) # горизонтальный вектор, задающий горизонт

# вычисляем угол между самой длинной стороной прямоугольника и горизонтом

angle = 180.0/math.pi \* math.acos((reference[0]\*usedEdge[0] + reference[1]\*usedEdge[1]) / (cv.norm(reference) \*cv.norm(usedEdge)))

if area > 500:

cv.drawContours(img,[box],0,(255,0,0),2) # рисуем прямоугольник

cv.circle(img, center, 5, color\_yellow, 2) # рисуем маленький кружок в центре прямоугольника

# выводим в кадр величину угла наклона

cv.putText(img, "%d" % int(angle), (center[0]+20, center[1]-20),

cv.FONT\_HERSHEY\_SIMPLEX, 1, color\_yellow, 2)

cv.imshow('contours', img)

cv.waitKey()

cv.destroyAllWindows()

ПРИЛОЖЕНИЕ В Графическая часть дипломного проекта

В графическую часть дипломного проекта входят:

-Сборочный чертеж верхней части дельта-робота,

- Габаритный чертеж манипулятора,

- Принципиальная схема системы вакуумного захватного усьройства,

- Спецификация к сборочному чертежу.

