|  |  |
| --- | --- |
| 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:  «Двурукий манипулятор»  Студент группы СМ7-83Б И. В. Горский  (Подпись, дата)  Руководитель ВКР С. В. Калиниченко  (Подпись, дата)  Консультант …  (Подпись, дата)  Консультант …  (Подпись, дата)  Нормоконтролер …  (Подпись, дата) | |
| 2024 г. | |

# **РЕФЕРАТ**

Расчетно-пояснительная записка 62 с., 45 рис., 5 табл., 5 источников, 1 прил.

ДВУРУКИЙ МАНИПУЛЯТОР, ПРИВОД ВРАЩЕНИЯ СХВАТА, СИСТЕМА ТЕХНИЧЕСКОГО ЗРЕНИЯ.

Объектом разработки является двурукий манипулятор для сортировки почтовых отправлений.

Цель работы — разработать двурукий манипулятор, акцентируя внимание на руке для тяжёлых грузов (бандеролей).

Поставленная цель достигается за счет разработки функциональной схемы системы управления РТК, разработки системы технического зрения (подбора компонентов, создания алгоритмов) и построения трёхмерных моделей рук в программе КОМПАС-3Д.

**СОДЕРЖАНИЕ**

[**РЕФЕРАТ** 2](#_Toc169565210)

[**ВВЕДЕНИЕ** 5](#_Toc169565211)

[**1. ОБЗОР ПРОЕКТА** 6](#_Toc169565212)

[**1.1 Актуальность, объект и предметы исследования** 6](#_Toc169565213)

[**1.2 Анализ существующих сортировочных двуруких манипуляторов** 7](#_Toc169565214)

[**2. ТЕОРЕТИЧЕСКОЕ ПРОЕКТИРОВАНИЕ** 9](#_Toc169565215)

[**2.1 Постановка задачи** 9](#_Toc169565216)

[**2.2 Технические требования** 9](#_Toc169565217)

[**2.3 Разработка функциональной схемы управления роботом** 10](#_Toc169565218)

[**2.4 Первичное проектирование манипулятора** 11](#_Toc169565219)

[**2.5 Разработка структурной схемы робота** 12](#_Toc169565220)

[**3. ПРОЕКТИРОВАНИЕ ПРИВОДА ВРАЩАТЕЛЬНОЙ СТЕПЕНИ ПОДВИЖНОСТИ** 13](#_Toc169565221)

[**3.1 Определение ЭД** 13](#_Toc169565222)

[**3.2 Поверочный расчёт вала** 15](#_Toc169565223)

[**3.3 Расчёт подшипников качения** 18](#_Toc169565224)

[**3.4 Расчёт подшипников качения на статическую грузоподъёмность** 19](#_Toc169565225)

[**3.5 Расчёт подшипников качения на выносливость** 19](#_Toc169565226)

[**3.6 Расчёт шпоночных соединений** 20](#_Toc169565227)

[**4. СБОРКА ПРИВОДА, МОДЕЛЬ РУКИ И ПОВОРОТНЫЙ МЕХАНИЗМ БАЗЫ** 23](#_Toc169565228)

[**5. ВЫБОР КОМПОНЕНТОВ ИСПОЛНИТЕЛЬНЫХ СИСТЕМ** 26](#_Toc169565229)

[**5.1 Анализ и выбор компонентов для системы управления** 26](#_Toc169565230)

[**5.2 Анализ и выбор компонентов для схвата манипулятора** 29](#_Toc169565231)

[**5.3 Анализ и выбор компонентов для системы технического зрения** 30](#_Toc169565232)

[**6.** **СИНТЕЗ БЛОКА ЛОГИЧЕСКОГО УПРАВЛЕНИЯ ОБОРУДОВАНИЕМ РТК И СТРУКТУРОЙ СХЕМЫ** 33](#_Toc169565233)

[**7.** **ПОСТРОЕНИЕ УПРАВЛЯЮЩЕГО КОНЕЧНОГО АВТОМАТА ДЛЯ РТК, ПРЕДНАЗНАЧЕННОГО ДЛЯ СОРТИРОВКИ ПОЧТОВЫХ ОТПРАВЛЕНИЙ** 36](#_Toc169565234)

[**7.1 Конечные автоматы компонентов системы** 37](#_Toc169565235)

[**7.2 Управляющий автомат и сеть Петри** 40](#_Toc169565236)

[**8.** **РЕГУЛИРОВОЧНЫЙ РАСЧЁТ ПРИВОДА ВРАЩЕНИЯ СХВАТА РУКИ МАНИПУЛЯТОРА** 42](#_Toc169565237)

[**8.1 Расчет контура тока** 47](#_Toc169565238)

[**8.2 расчет контура скорости с контуром тока** 49](#_Toc169565239)

[**8.3 Расчёт контура положения в контурном режиме** 55](#_Toc169565240)

[**ЗАКЛЮЧЕНИЕ** 61](#_Toc169565241)

[**СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ** 62](#_Toc169565242)

[**ПРИЛОЖЕНИЕ А** 63](#_Toc169565243)

[**ПРИЛОЖЕНИЕ Б** 68](#_Toc169565244)

# **ВВЕДЕНИЕ**

Целью выпускной квалификационной работы является разработка системы управления двурукого манипулятора для улучшения качества сортировки различных объектов, которые могут поступать на пункт сортировки.

Работа включает в себя проектирование привода вращения схвата руки манипулятора: выбор электродвигателя через энергетический расчёт, проверка на прочность компонентов привода, проведение регулировочного расчёта привода с целью подобрать коэффициенты регуляторов для его контуров; создание реалистичной трёхмерной модели руки манипулятора для работы с тяжёлыми грузами, имеющими вес от 5 кг до 20 кг; разработку узла поворота баз рук манипулятора; создание конечного автомата для РТК и сети Петри с целью наглядно показать алгоритм работы системы.

Основные положения: в процессе работы должна проводиться процедура передачи управления между рукой для писем и рукой для бандеролей; система технического зрения должна определять положение и тип отправления; руки должны выполнять операции за определённое время с целью повышения скорости работы РТК.

# **1. ОБЗОР ПРОЕКТА**

# **1.1 Актуальность, объект и предметы исследования**

**Актуальность** данной выпускной квалификационной работы заключается в наличии неавтоматизированных этапов сортировки почтовых отправлений на складах маркетплейсов или пунктов сортировки на почте. Проектируемый манипулятор позволит автоматизировать данные процессы, повысив надёжность и увеличив скорость сортировки. Разрабатываемый манипулятор не требует наличия оператора, что позволяет ему работать круглосуточно.

**Объект исследования** состоит из следующих частей:

* Проектирование и конструирование роботов
* Система управления электропривода РТС

**Предметом исследования** является проектирование и система управления манипулятором без участия оператора.

## **1.2 Анализ существующих сортировочных двуруких манипуляторов**

На сегодняшний день таких систем, которые бы использовали двурукие манипуляторы не для работы обеими руками с одним объектом, а использовали бы руки по отдельности для захвата соответствующих им предметов не так много; для крупных объектов, таких как почтовые отправления, таких решений не существует в принципе.

**Rethink Robotics Sawyer** – это разработанный в Соединённых Штатах Америки манипулятор, использующий две руки. Данный робот является коллаборативным, то есть он может работать с человеком безопасно, не нанося ему травмы и увечья. У робота есть камера и сенсоры, которые помогают роботу взаимодействовать с окружающим миром и с человеком. Робот программируется человеком при помощи ручного перемещения звеньев манипулятора (датчики на приводах запоминают свои положения в процессе, что позволяет воспроизводить движение в автоматическом режиме).

**Shadow Dexterous Hand** – этот робот разработан в Великобритании коммерческой компанией. Кисть руки робота имеет пять пальцев; сама же рука имеет 6 степеней подвижности, что обеспечивает возможность занятия конечным органом любого желаемого положения. Большое количество сенсоров открывает возможности для тонкой настройки системы управления роботом.

Изображение выглядит как автомат, игрушка, машина, робот

Автоматически созданное описаниеИзображение выглядит как свет

Автоматически созданное описание

Рисунок 1 — Двурукие манипуляторы Sawyer и Dexterous Hand

**ABB YuMi –** двурукий манипулятор, разработанный в Японии. Робот является коллаборативным, то есть он может работать с человеком, не причиняя ему вреда. Оператор может программировать манипулятор, вручную перемещая звенья манипулятора. Датчики угла запомнят свои траектории и смогут воспроизвести их в автоматическом режиме позже. Существует модификация данного робота, способная выполнять задачу сортировки различных объектов, которые могут быть захвачены только одним типом захватного устройства, что вынуждает использовать СТЗ для выбора руки для каждого объекта.

**Декстер** – двурукий манипулятор, разработанный в Соединённых Штатах Америки для Международной Космической Станции. Он представляет из себя туловище без головы, которое имеет две руки. На каждой руке расположен свой набор различных конечных органов, использующихся для соответствующих им задач. Робот может работать двумя руками с одним объектом или с двумя разными.

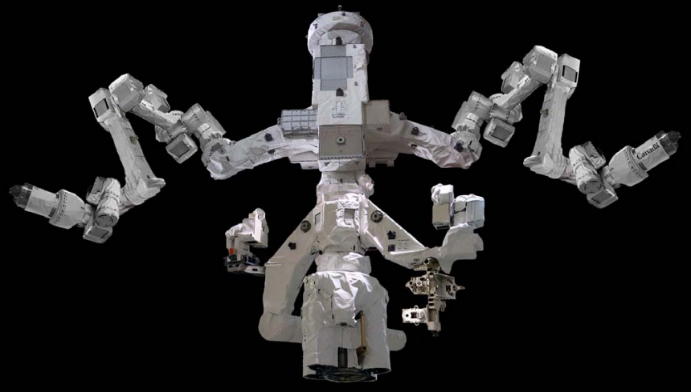
****

Рис. 2. Двурукие манипуляторы Декстер и YuMi

# **2. ТЕОРЕТИЧЕСКОЕ ПРОЕКТИРОВАНИЕ**

## **2.1 Постановка задачи**

В ВКРБ спроектирован РТК, манипулятор, как его часть, с акцентом на руке для переноса тяжёлых грузов в виде коробок (бандеролей). Выполнение данной работы позволит студенту углубить свои знания в таких дисциплинах как: Моделирование и исследование РТС, Управление роботами, Информационные устройства робототехнических систем.

Для достижения цели в работе поставлены следующие задачи:

* На основе обзора проекта разработать алгоритм работы РТК
* Выбрать компоненты самой руки и её периферии
* Разработать структурную схему системы управления
* Разработать модель манипулятора
* Разработать привод вращения схвата руки манипулятора для бандеролей, его сборочный чертёж и спецификацию к нему
* Разработать сеть петри для РТК
* Рассмотреть алгоритм работы СТЗ
* Решить прямую и обратную задачи по положению

## **2.2 Технические требования**

Проектируемая рука манипулятора должна иметь возможность поднимать и перемещать коробку (бандероль) весом от 5 до 20 кг с ограничениями по габаритам: до 60см в длину; сумма трёх измерений до 90см. Расстояние от центра робота до крайней точки, где может располагаться бандероль на конвейере - до 1.1м. Максимальная высота, на которую может быть поднята бандероль в процессе работы руки - 0.2м. Радиус сферы рабочей зоны руки манипулятора должен составлять 1.3м. Захватное устройство руки должно вращаться на 180 градусов для обеспечения необходимых условий захвата.

## **2.3 Разработка функциональной схемы управления роботом**

Система управления данного робототехнического комплекса состоит из двух уровней:

* **Верхний уровень:** Этот уровень состоит из системы технического зрения (камеры, устройства считывания штрих-кодов, фотодатчика) и управляющего компьютера, которые обмениваются данными посредством USB – кабелей (за исключением фотодатчика – с него сигнал идёт по кабелю Ethernet).
* **Нижний уровень:** Данный уровень включает в себя конвейер (управление им осуществляется по скорости), приводы непосредственно рук манипулятора, управляющие клапана пневмосистемы, от которых зависит работа сцепной муфты и захватных устройств рук для писем и бандеролей.

Изображение выглядит как текст, снимок экрана, диаграмма, Параллельный

Автоматически созданное описание

Рис. 3. Функциональная схема системы управления

## **2.4 Первичное проектирование манипулятора**

В проекте используется схема с общим приводом для двух баз (переключение будет производиться посредством сцепной муфты). Рука для коробок (бандеролей) имеет 5 степеней подвижности, включая базу. Отличительной особенностью этой руки является привод вращения схвата манипулятора – его нет на руке для писем за ненадобностью.

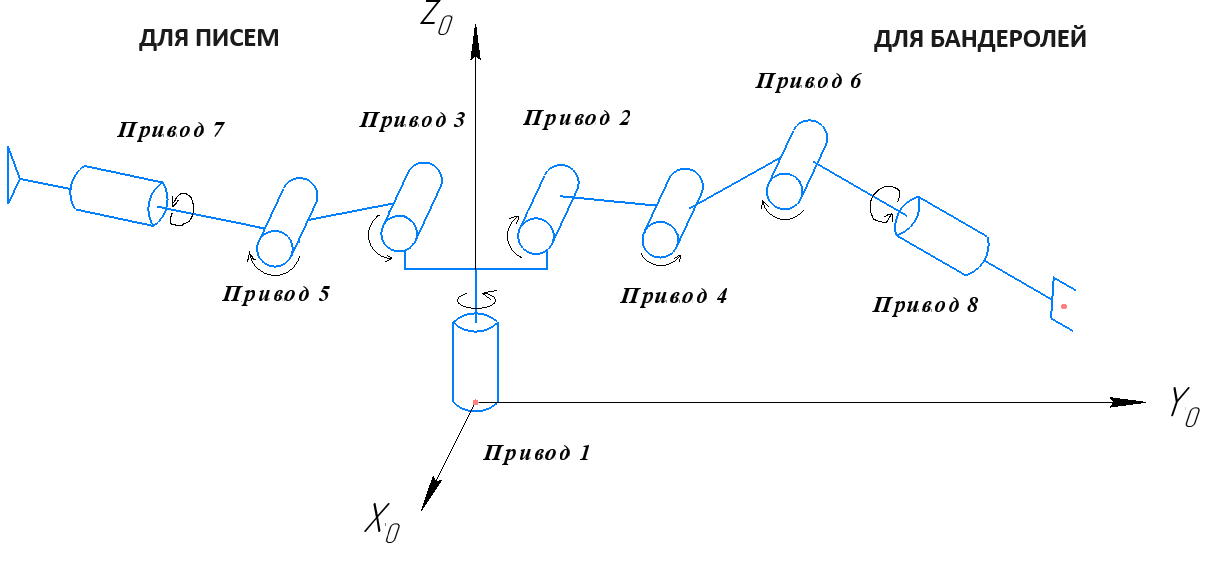


Рис.4. Упрощённая схема структуры манипулятора

Для звеньев рук важным фактором при выборе электродвигателей является мощность – рукам придётся переносить значительный вес (масса коробки – до 20 кг), а также нельзя забывать, что сами звенья манипулятора имеют большие значения масс и моментов инерции. Помимо этого, приводы степеней подвижности обеих рук манипулятора должны иметь широкие возможности по управлению и настройке. Этим требованиям отвечают бесколлекторные двигатели постоянного тока (БДПТ). Достоинствами БДПТ являются высокий крутящий момент и лёгкость в управлении, необходимые манипулятору, выполняющему заявленные в данной работе задачи. БДПТ требуется преобразователь тока (из переменного тока сети в постоянный ток на двигателе). Данный недостаток не доставляет неудобств, т.к. можно установить один преобразователь на все приводы манипулятора. В итоге, для поставленной задачи лучше всего подходят бесколлекторные двигатели постоянного тока.

## **2.5 Разработка структурной схемы робота**

**Рассматриваемый робот** представляет из себя систему из двух рук, их базы запитываются от общего привода. Все сочленения в руке вращательные. Выбор в пользу вращательных степеней подвижности был сделан потому, что поступательные степени сложнее в реализации – для них нужны либо передачи рейка - шестерня либо передачи винт-гайка. Целесообразнее использовать мотор-редукторы для обеспечения подъёма и переноса бандеролей, которые являются грузом с существенной массой (до 20 кг), что накладывает дополнительные требования по прочности для компонентов поступательной кинематической пары.

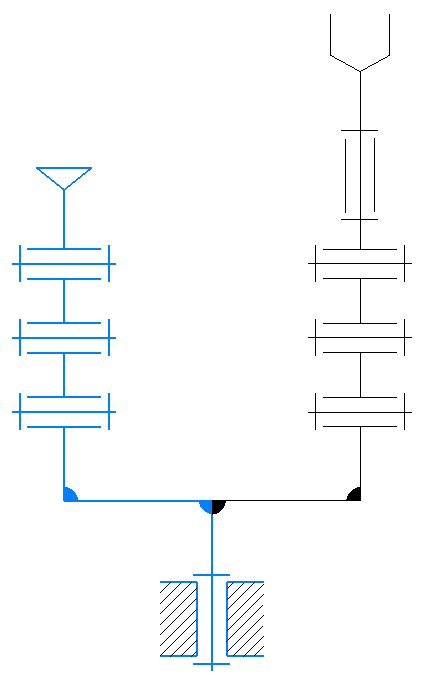


Рис. 5. Структурная схема манипулятора

# **3. ПРОЕКТИРОВАНИЕ ПРИВОДА ВРАЩАТЕЛЬНОЙ СТЕПЕНИ ПОДВИЖНОСТИ**

# **3.1 Определение ЭД**

Механическая структура руки манипулятора представляет из себя множество звеньев, соединённых вращательными сочленениями. На каждый привод действуют моменты сил тяжести и сил инерции, кроме привода вращения схвата манипулятора – момент сил инерции нагрузки для него крайне мал, а сила тяжести не создаёт момента вовсе. Таким образом, вся работа привода сводится к подаче нужного для достижения должного ускорения момента без учёта каких-либо прочих факторов. Однако, к данному приводу предъявляются особые требования по массе и габаритам – это последний привод руки для бандеролей, его масса и момент инерции значительно влияют на необходимые моменты для предыдущих приводов.

Таким образом, выбираем для рассмотрения привод вращения схвата руки для бандеролей. Привод должен поворачивать захватное устройство за ограниченное время на заданный угол, таким образом, он является следящим и на него накладываются следующие требования:

* Высокий пусковой момент
* Мягкая характеристика
* Длительный срок работы

Под данные критерии подходят ДПТ и асинхронный двигатель. Недостатком ДПТ является использование щеточно-коллекторного узла, что решается заменой его на БДПТ. В БДПТ используются дорогие и хрупкие постоянные магниты, помимо прочего, они имеют меньшую по сравнению с ДПТ удельную мощность, однако более простую систему управления и малые габариты. В соответствии с указанными преимуществами и недостатками выбранный тип ЭД - БДПТ

Мощность ЭД определяется по формуле:

Изображение выглядит как Шрифт, линия, снимок экрана, число

Автоматически созданное описание

где

P*н*- мощность нагрузки на выходном валу;

*𝜂* - КПД цепи двигатель - нагрузка;

K*з* - коэффициент запаса;

*Мн* - момент нагрузки на выходном валу (он равен 40 Нм);

ω - угловая скорость на выходном валу (она равна 8 об/мин);

Входная мощность передачи с учётом КПД:

Изображение выглядит как текст, Шрифт, линия, число

Автоматически созданное описание

В силу большого момента на выходном валу выбор редуктора был сильно сужен. Из предложенных вариантов был подобран модельный ряд редукторов Maxon GP 42 С, у которых максимальный момент на выходном валу достигает 15 Нм. Для данного ряда редукторов рекомендовался двигатель EC-max 30 c номинальной мощностью N = 60 Вт и характеристиками: nдв = 8040 об/мин; Mдв = 60 мН\*м. В силу того, что N > N*Пдв*, получаем, что компоненты подобраны верно.

Общее передаточное отношение зубчатой передачи и редуктора должно быть:

uобщ = 1005

Выбираем редуктор Maxon GP 42 C с передаточным отношением

uр = 546.

Тогда, передаточное отношение зубчатой передачи будет:

uзп = 1,84

*Итог :*

Двигатель: Maxon EC-max 30

* Мощность: N = 60 Вт;
* Номинальный момент: Mном = 60 мН\*м.
* Номинальная скорость: nном = 8040 об/мин;

Редуктор: Maxon GP 42 C

* Передаточное отношение: u = 546;
* Длительный / Допустимый момент - 15 / 22,5 Н\*м.
* КПД: η= 0,70;

# **3.2 Поверочный расчёт вала**

Обозначим за R1 и R2 реакции в подшипниках; Ra - реакция в опорах на растяжение; Т - внешний момент, приложенный к валу; Tz = Т = 60 Нм - момент на зубчатом колесе; Fz = √(13032+5022) = 1400 Н - сила на зубчатом колесе[1]. Силы Fr = Fa = 1000 Н - изгибающая и растягивающая силы (не указанны в тех. задании, поэтому подобраны примерно). Рассмотрим случай, когда Fr и Fz сонаправлены:

Изображение выглядит как текст, диаграмма, линия, Параллельный

Автоматически созданное описание

Рис.6. Приложение сил к выходному валу

Рассчитаем реакции в подшипниках:

44\*R2 + 75,5\*Fz + 99\*Fr  = 0 → R2 = 4652 Н

R1 + R2 + Fz + Fr  = 0 → R1 = 2252 Н

Построим эпюры и найдем самое нагруженное сечение:

Изображение выглядит как текст, линия, диаграмма, Шрифт

Автоматически созданное описание

Рис.7. Эпюры сил и моментов

Как видно из рисунка 7, самое нагруженное сечение - II. Рассчитаем в нем напряжения:

Wx = π32∗(0.0353−0.0203)=3.424∗10-6 *м3*

σи = 205 *Н/м* Wx= 59,9 *МПа*

А = π4∗(0.0352−0.0202)=6.48∗10-4 м2

σр = 1,5 *МПа*

σэкв = √ σ2*и* + σ2*р*=59,9 *МПа*

Запас будет:  
s=σ*в/*σ*экв*=24559,9=4.1, где σв = 245 МПа - предел выносливости алюминия.

*Итог:* Запас прочности спроектированного вала составляет s = 4,1.

# **3.3 Расчёт подшипников качения**

Расчёт подшипников качения бывает двух видов: расчёт на заданный ресурс и расчёт на статическую прочность.

Для расчёта подшипника необходимо знать реакцию в подшипнике. В рассматриваемом случае на тихоходном валу установлены два шариковых радиально-упорных подшипника и реакции в них следующие: 2252 Н и 4652 Н. Модели подшипников одинаковые, поэтому достаточно провести расчёт для более нагруженного, т.е. реакцию на подшипнике принимаем за 4652 Н [2].

Исходные данные для расчёта на выносливость:

Модель: 46107 ГОСТ 831-75.

Расшифровка: шариковый радиально-упорный подшипник, внешний диаметр 52 мм, внутренний – 35 мм, ширина подшипника 14 мм

Статическая грузоподъёмность *С*0=12900 *Н*

Динамическая грузоподъёмность *С*=18100 *Н*.

Радиальная сила Fr=4652 *Н*.

Осевая сила Fa=1000 *Н*.

## **3.4 Расчёт подшипников качения на статическую грузоподъёмность**

При расчёте подшипника качения на статическую прочность проверяется условие сможет ли подшипник выдержать статическую нагрузку. Расчёт будет проводиться согласно [3].

Por=max{X0∗Fr+Y0∗Fa,F*пик*r},

P0r - эквивалентная статическая радиальная сила, действующая на подшипник,

Fr, Fa– радиальная и осевая статическая сила,

X0, Y0– коэффициенты, учитывающие влияние радиальной и осевой силы на разрушения подшипника при статической нагрузке.

Por=max{X0∗F*пик*r+Y0∗F*пик*a,F*пик*r}=max{0,6∗4652 +0,37∗1000, 4652 }=4652 *Н*.

Условие статической прочности подшипника:

Por ≤ C0r = C0*,*

4652 ≤ 12900*,*

*Итог:* Условие выполнено, а значит подшипник выдерживает статическую нагрузку при пиковых значениях усилий.

## **3.5 Расчёт подшипников качения на выносливость**

Расчёт подшипников качения на выносливость подразумевается в проверке возможности реализации подшипником заданного ресурса с некоторой вероятностью выхода подшипника из строя, за вероятность выхода из строя берут число меньше или равно 10%. Ресурс – число оборотов, при истечении которого не происходит усталостного повреждения колец или тел качения, одного кольца подшипника относительно другого при заданном усилии[4].

Эквивалентная нагрузка:

Pr=(X∗V∗Fr+Y∗Fa)∗K*Б*∗K*Т* ,

K*Б*=1,5; K*Т*=1,

т.к. температура подшипника <100℃,

V=1; Fr=4652 *Н*; Fa=1000*Н*.

X, Y зависят от параметра e, однако в нашем случае они и так определены:

X = 0,41; Y = 0,87

Тогда, P=(0,41∗1∗4652+0,87∗1000)∗1,5∗1=4164,75 *Н*.

Ресурс подшипника при 90 % успехе при заданной номинальной нагрузке:

L10ah=a1∗a23∗10660∗n∗(*С*P)m

m– параметр кривой усталости; для шарикоподшипников m = 3

a1- коэффициент долговечности;

a1= 1

a23- коэффициент, характеризующий совместное влияние на долговечность особых свойств металла деталей подшипника и условий его эксплуатации; из [1]

a23= 0,75

L10ah=1∗0,75∗10660∗3,5∗(181004652)3=210358 *ч*.

*Итог :* Полученное значение значительно больше заданного ожидаемого ресурса в 3000 часов, а это значит, что подшипники выдержат необходимый срок службы с вероятностью 99,7%.

# **3.6 Расчёт шпоночных соединений**

Для передачи крутящего момента с зубчатого колеса тихоходного вала на тихоходный вал используется шпоночное соединение с призматической шпонкой с закругленными торцами ГОСТ 23360-78. Данный расчёт является проектным, необходимо определить длину шпонки. Расчёт проводился согласно.

Изображение выглядит как зарисовка, диаграмма, рисунок, Технический чертеж

Автоматически созданное описание

Рис. 8. Схема для расчёта напряжений смятия в шпоночном соединении

Напряжения, сминающие шпонку, не должны превышать допустимой нагрузки:

σ*см*=2∗T*пикт*d∗l*раб*∗k≤[σ*см*]

Согласно ГОСТ 23360-78 шпонки делают из стали 45Х. Для этих сталей допустимое напряжение смятия:

[σ*см*] = 360 *МПа𝜎см = 360 МПа*

.Расчитаем длины шпонок, как:

[l*раб*⎤⎦⎥=2∗T*пикт*∗1000d∗[σ*см*]∗k

Расчёт шпоночного соединения на валу ведущего колеса:

Параметры для данного шпоночного соединения:

d = 12 мм

b = 4 мм,

k = 2 мм,

h = 4 мм,

T*пикт*=57,7

Н\*м – пиковый крутящий момент на тихоходном валу.

[l*раб*⎤⎦⎥=2∗T*пикт*∗1000d∗[σ*см*]∗k=2∗57,7∗10000,014∗360∗106∗0,002≥11 *мм*

[L]=[l*раб*]+b=11+5=16 *мм*

L ≥[L]

Минимальное значение L для данного диаметра вала 18 мм согласно ГОСТ 23360-78 – 18 мм.

*Итог :* Примем значение L=18 *мм*.

Расчёт шпоночного соединения на валу ведомого колеса:

Параметры для данного шпоночного соединения:

d = 35 мм

b = 10 мм,

k = 3 мм,

h = 8 мм,

T*пикт*=111,8 Н\*м – пиковый крутящий момент на тихоходном валу.

[l*раб*⎤⎦⎥=2∗T*пикт*∗1000d∗[σ*см*]∗k=2∗111,8∗10000,035∗360∗106∗0,003≥5,9 *мм*

[L]=[l*раб*]+b=5,9+10=15,9 *мм.*

L ≥[L]*.*

Минимальное значение L для данного диаметра вала 35 мм согласно ГОСТ 23360-78 – 22 мм.

*Итог :* Примем значение L=22 *мм*.

# **4. СБОРКА ПРИВОДА, МОДЕЛЬ РУКИ И ПОВОРОТНЫЙ МЕХАНИЗМ БАЗЫ**

В рамках проекта была разработана сборка привода. Для осуществления сборки спроектирован корпус, в который вставляется мотор-редуктор и выходной вал с фланцем. Привод имеет фланец на выходном валу, имеющий отверстия для соединения со схватом манипулятора. Также в корпусе привода есть отверстия, позволяющие закрепить его на выходном валу привода предыдущей степени подвижности. Привод имеет механические ограничения по углу поворота, реализованные при помощи канавки на колесе и шлица. Сборочный чертёж привода представлен в Приложении Б. Рисунок сборки привода и схвата манипулятора приведён ниже:

Изображение выглядит как машина

Автоматически созданное описание

Рис. 9. Сборка привода вращения схвата манипулятора

Разработана трёхмерная модель руки для бандеролей. При создании модели учитывались массы и инерции звеньев: каждый последующий привод легче предыдущего, так как ему нужно сопротивляться меньшему моменту сил тяжести и работать на меньшую нагрузку. Привод качания схвата расположен в начале своего звена с целью уменьшить приведённые к валам предыдущих приводов моменты инерций данного звена.

Изображение выглядит как цилиндр, желтый

Автоматически созданное описание

Рис. 10. Левая рука манипулятора (вид 1)

Изображение выглядит как желтый, кран

Автоматически созданное описание

Рис. 11. Левая рука манипулятора (вид 2)

В процессе работы был разработан поворотный механизм для баз двух манипуляторов. Данный механизм работает следующим образом: двигатель передаёт крутящий момент на вал муфты посредством цилиндрической передачи; если управляющий компьютер выбрал базу, то на соответствующую пневмомембрану подаётся давление, и шестерня конической передачи соединяется с колесом (происходит сцепление), и крутящий момент начинает передаваться на базу. Далее момент конической передачей передаётся наверх. Когда выбранная управляющим компьютером рука манипулятора завершит работу, давление перестанет подаваться на мембрану и она вернётся в исходное положение при помощи пружины. Сцепная муфта обладает рядом преимуществ перед фрикционной, главным из которых является постоянство передаточного отношения.

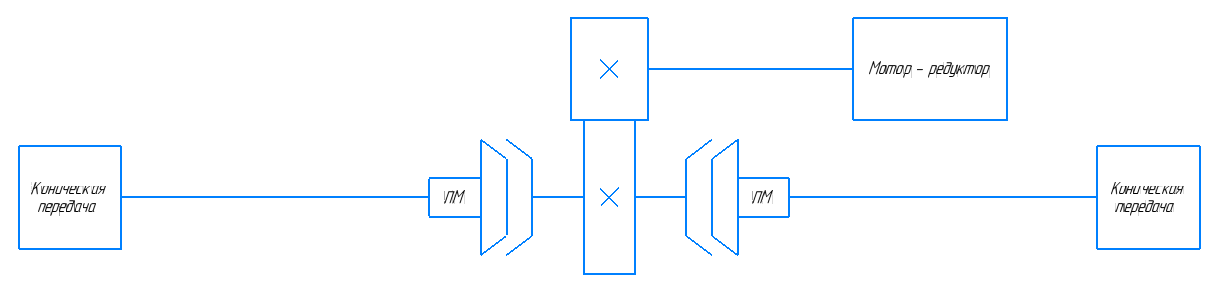


Рис. 12. Схема механизма переключения между валами

# **5. ВЫБОР КОМПОНЕНТОВ ИСПОЛНИТЕЛЬНЫХ СИСТЕМ**

## **5.1 Анализ и выбор компонентов для системы управления**

Для данной работы был выбран одноплатный компьютер Raspberry Pi 4 Model B. Model B является первым поколением плат Raspberry Pi, оно обладает хорошей скоростью работы, хоть и стоит дороже последующих моделей и обладает меньшим объёмом оперативной памяти. Данная модель обладает SoC BCM2835, памятью 512 Мбайт, стандартным формфактором, возможностью подключения к сети Ethernet, 26-контактным GPIO. Беспроводная связь отсутствует. Model B обладает 64-битным четырёхъядерным процессором ARM Cortex-A72 (тактовая частота такого процессора – 1,5 ГГц); оперативная память составляет 8 ГБ. Стоит уточнить, что Raspberry Pi 4 Model B имеет встроенный Wi-Fi адаптер 802.11ac, Bluetooth 5, отсутствием ограничения на пропускную способность сети Ethernet. Имеются по два порта USB 2.0 и по два для 3.0. В данной версии отказались от большого количества ОЗУ и переработали печатную плату. Недостатком платы является необходимость питания только пятью вольтами, прочие напряжения не подходят.

В итоге, лучшим решением для поставленных задач можно считать Raspberry Pi 4 Model B, обладающую необходимыми вычислительными мощностями, памятью и скоростью выполнения команд.

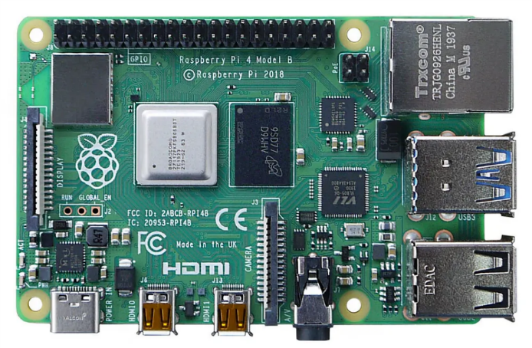


Рисунок 13 - Плата Raspberry Pi 4 Model B

**Характеристики Raspberry Pi 4 Model B**

* Чип: Broadcom BCM2711 (CPU, GPU, DSP, and RAM)
* Процессор (CPU): 64-битный 4-ядерный ARMv8 Cortex-A72 с тактовой частотой 1,5 ГГц
* Видеопроцессор (GPU): VideoCore IV 3D с тактовой частотой 500 МГц
* Память (RAM): 8 Гб LPDDR4 с тактовой частотой 1600 МГц
* Разъемы:

аудио/видео цифровой: micro-HDMI - 2 шт;

аудио/видео аналоговый: 3,5 мм jack (4 pin);

USB порт: USB 2.0 - 2 шт;

USB порт: USB 3.0 - 2 шт;

накопитель: microSD;

сеть: 10/100/1000 Ethernet;

порт камеры: Raspberry Pi camera port (2-lane MIPI CSI)

* Интерфейс ввода-вывода (GPIO 40 pin):

Up to 6x UART;

Up to 6x I2C;

Up to 5x SPI;

1x SDIO interface;

1x DPI (Parallel RGB Display);

1x PCM;

Up to 2x PWM channels;

Up to 3x GPCLK outputs

* Беспроводные интерфейсы:

WiFi: двухполосный 2,4 ГГц and 5 ГГц IEEE 802.11.b/g/n/ac

Bluetooth: 5.0 Classic и Low Energy (BLE)

* питание: USB-C 5 в
* Размер (ДхШхВ): 85 x 56 x 17 мм
* Вес: 70 гр

Управление рукой манипулятора осуществляется микроконтроллером, который должен иметь возможность получать и обрабатывать данные с датчиков приводов (решать обратную задачу кинематики), а также выдавать управляющие сигналы на приводы (предварительно решая прямую задачу кинематики – строя требуемые траектории для приводов исходя из требуемой траектории для захватного устройства). В результате анализа рынка микроконтроллеров был сделан выбор в пользу платы Arduino uno R3. Она является аппаратно-программным комплексом с открытой архитектурой для решения различных задач робототехнической направленности. На данной плате установлены порт USB 3.0, поддерживающий возможность питания этой платы, непосредственно сам разъём питания, стабилизаторы напряжения на 3.3 и на 5 В, контроллер ATmega328P с тактовой частотой 16 МГц, светодиодные индикаторы.

**Характеристики Arduino uno R3**

* Микроконтроллер - ATmega328P
* Микроконтроллер: ATmega328P
* Диапазон допустимого напряжения питания: 5-20 В
* Рекомендуемое напряжение питания: 7-12 В
* Количество цифровых вводов/выводов: 14
* ШИМ: 6 цифровых пинов могут быть использованы как выводы
* Количество аналоговых выводов: 6
* Максимальная сила тока: 40 mAh с одного вывода и 500 mAh со всех выводов.
* Flash память: 32 кб
* SRAM: 2 кб
* EEPROM: 1 кб
* Тактовая частота: 16 МГц
* LED\_BUILTIN: Встроенный светодиод - 13

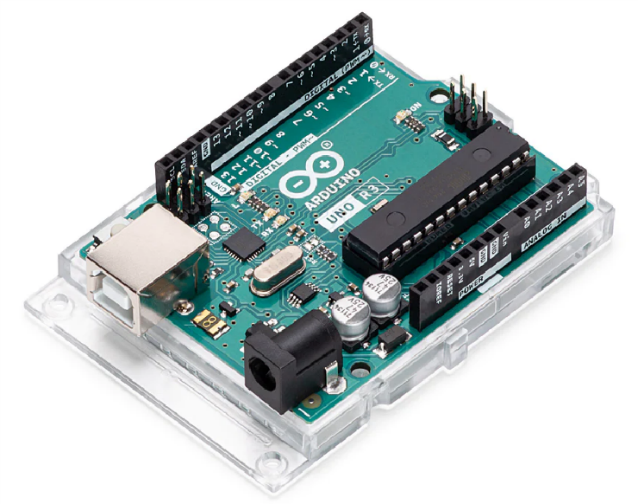


Рис.14. Arduino uno R3

## **5.2 Анализ и выбор компонентов для схвата манипулятора**

Для схвата манипулятора были выбраны две вакуумные сильфонные присоски Festo ESV-20-BN, так как у них сравнительно низкий диаметр (и, как следствие, этими присосками можно захватить коробки различных габаритов), высокая грузоподъёмность и удобный тип крепления.

**Технические характеристики:**

* Компенсатор высоты для держателя вакуумного захвата - 10 мм
* Минимальный радиус заготовки - 40 мм
* Тип конструкции - Балонный пневмоцилиндр
* Диаметр присоски - 50 мм
* Объем присоски - 17,6 см³
* Форма вакуумного захвата - Круглая, 1.5 гофра
* Усилие отрыва при 70% вакуума - 100 Н
* Номинальное рабочее давление -0,7 бар
* Усилие удержания при номинальном рабочем давлении - 100 Н
* Тип крепления - Через канал вакуума
* Вес - 16,2 г

Изображение выглядит как круг, серебряный, керамический

Автоматически созданное описание

Рис.13 Присоска вакуумная сильфонная Festo ESV-20-BN

## **5.3 Анализ и выбор компонентов для системы технического зрения**

Исходя из требований по углу обзора камеры, а также принимая во внимание её задачи (она должна выдавать картинку такого разрешения, чтобы управляющий компьютер после её обработки смог вычислить координаты центра отправления и угол его поворота) была выбрана камера FQ-M. Данная камера обладает достаточным разрешением для того, чтобы управляющий компьютер по данным с этой камеры смог определить координаты центра бандероли и ориентацию этой бандероли. Камера обладает частотой кадров в 60, что является избыточным для рассматриваемой системы. Тем не менее, это может оказаться полезным в случае непрекращающегося движения посылки по конвейеру.

Характеристики:

* Частота кадров: 60 кадров/с
* Разрешение: 752х480
* Датчик изображения: КМОП



Рис.16. Видеокамера FQ-M

Подберём фотодатчик, который будет отслеживать наличие или отсутствие отправления на определённом участке конвейера. Из поставленной задачи получает следующие требования: фотодатчик должен иметь дальность работы большую, чем ширина конвейера (600 мм), а также высокое быстродействие – от него зависит скорость реагирования всей системы на появление отправления. Принимая во внимание все вышеописанные требования, выберем фотодатчик Lanbao Pss-bc100dpb-e2 07-00038906. Данный датчик не обладает USB интерфейсом, что означает, что данные на управляющий компьютер будут поступать через кабель Ethernet.

Характеристики:

* Максимальный ток: 0.2 А
* Время отклика: 1 мс
* Выходной сигнал: PNP
* Расстояние срабатывания: 1000 мм
* Монтаж: встраиваемый
* Напряжение: 24 В
* Материал корпуса: АБС-пластик



Рис.17. фотодатчик Lanbao Pss-bc100dpb-e2 07-00038906

Для определения параметров отправления (тип отправления, масса и габариты) используется сканер штрих-кодов. Данный сканер должен иметь возможность распознать штрих-код на расстоянии до 500 мм и под углом до 45 градусов. Выбран сканер Zebra DS3608-ER DS3608-ER3UF202ZVW как наиболее подходящий.

Характеристики:

* Защита от пыли и влаги: IP65
* Габариты оборудования, см: 18,5 x 7,6 x 13,2
* Вид сканера: Проводной, Промышленный
* Вес оборудования, кг: 0.334
* Входное напряжение: 4,5-12,6 В
* Чтение штрих-кодов: 1D/2D



Рисунок 18 - Сканер штрих-кода Zebra DS3608-ER DS3608-ER3UF202ZVW

# **СИНТЕЗ БЛОКА ЛОГИЧЕСКОГО УПРАВЛЕНИЯ ОБОРУДОВАНИЕМ РТК И СТРУКТУРОЙ СХЕМЫ**

Система включает себя следующие компоненты: три конвейера: входной (1), для писем (2), для бандеролей (3), фотодатчик (4), видеокамер (5), сканер штрихкода (6) и манипулятора, имеющего две руки на разных вращающихся базах (7) (одна рука для писем, другая для бандеролей).

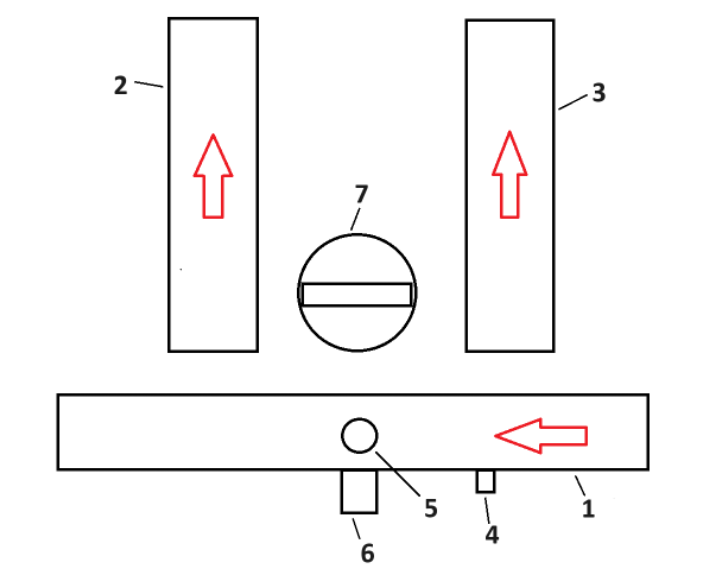


Рис. 19. Схема рабочей станции

Далее будет рассматриваться алгоритм управления в случае, когда переносимым отправлением является бандероль. Аналогичный случай для писем рассмотрен не будет для избежания излишних повторений.

Алгоритм работы:

1. Различные отправления идут по общему конвейеру.

2. В случае прохождения через фотодатчик отпраления конвейер доводит его до сканера штрих-кодов и останавливается (то есть конвейер отрабатывает некоторый сигнал по скорости, такой, что перемещение конвейера будет таковым, что расстояние отправления до обеих рук манипулятора будет приблизительно одинаковым).

3. При помощи сканера штрих-кодов получается информация об почтовом отправлении.

4. В случае, если оказалось, что рассматриваемое отправление является бандеролью, то захватное устройство выводится на позицию над коробкой, параллельно поворачиваясь в нужную для захвата коробки ориентацию, опускается, захватывает бандероль и подымает её, затем перемещаясь к конвейеру для бандеролей.

5. Если конвейер для коробок свободен, то на него ставится бандероль; если нет, то рука входит в режим ожидания (цикл).

6. Если на входном конвейере остались отправления, то производится возврат к пункту 2. Иначе - к пункту 1.

Алгоритм работы РТК спроектирован так, что исключены все возможности прекращения его работы – все отправления, рано или поздно, будут отсортированы. К недостаткам алгоритма можно отнести вход руки в цикл ожидания в отсутствие места для соответствующего отправления. В остальном же система обладает высоким быстродействием и потенциалом для модернизации – алгоритм может быть адаптирован для случая одновременной работы привода вращения баз на обе базы манипулятора – в таком случае обе руки будут поворачиваться одновременно.

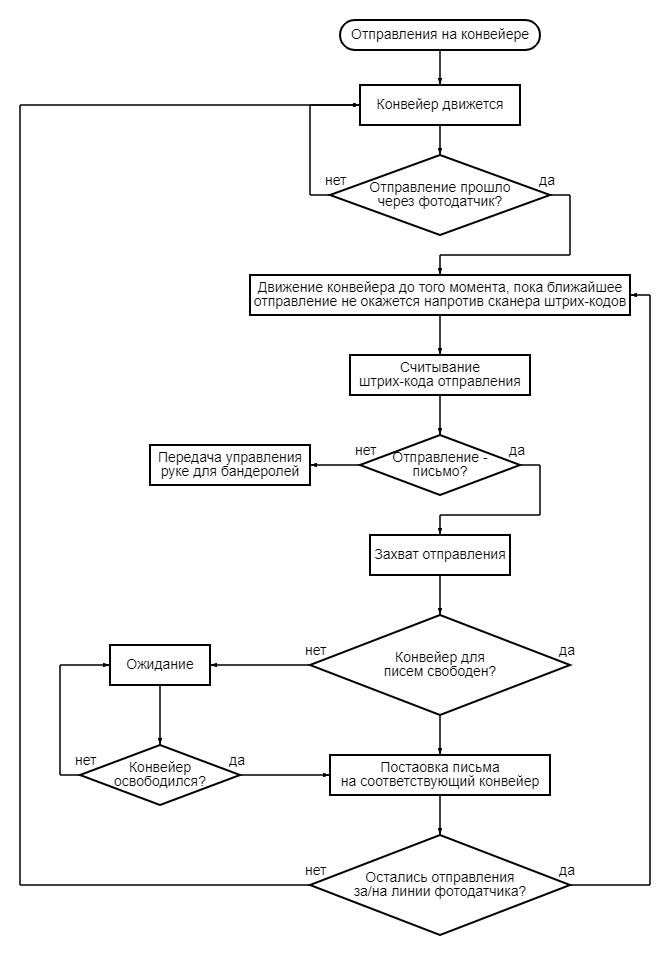


Рисунок 20 - Алгоритм работы системы

# **ПОСТРОЕНИЕ УПРАВЛЯЮЩЕГО КОНЕЧНОГО АВТОМАТА ДЛЯ РТК, ПРЕДНАЗНАЧЕННОГО ДЛЯ СОРТИРОВКИ ПОЧТОВЫХ ОТПРАВЛЕНИЙ**

**Имеется:**

* Манипуляционный робот для писем
* Манипуляционный робот для бандеролей
* СТЗ для проверки наличия почтовых отправлений на главном конвейере

Алгоритм работы системы можно представить следующим образом:

С определённой частотой опрашивается фотодатчик, установленный на конвейере. Если фотодатчик выдаёт управляющему компьютеру сигнал о наличии на конвейере почтового отправления, то управляющий компьютер посылает на конвейер сигнал по скорости, он отрабатывается и в итоге отправление останавливается в таком положении, что оно примерно равноудалено от обеих рук манипулятора. Помимо этого, сканер штрих-кодов считывает информацию об отправлении. Если отправление – бандероль, то управляющий компьютер отдаёт сигнал микроконтроллеру руки для бандеролей о том, что тот может начать управление. Определяется центр бандероли и её ориентация (вычисления производит управляющий компьютер). По полученным данным строится желаемая траектория схвата, из которой получаются желаемые траектории приводов. Привода начинают отработку движения по задаваемой траектории. В результате захватное устройство с захваченной коробкой приходит в конечную точку на конвейере для коробок.

Для описания объектов управления в работе спроектированы конечные автоматы.

## **7.1 Конечные автоматы компонентов системы**

Рука манипулятора для писем:

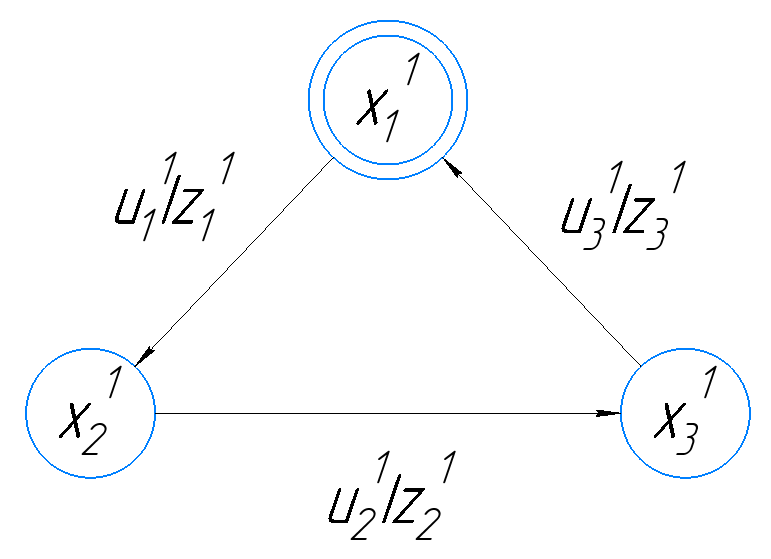
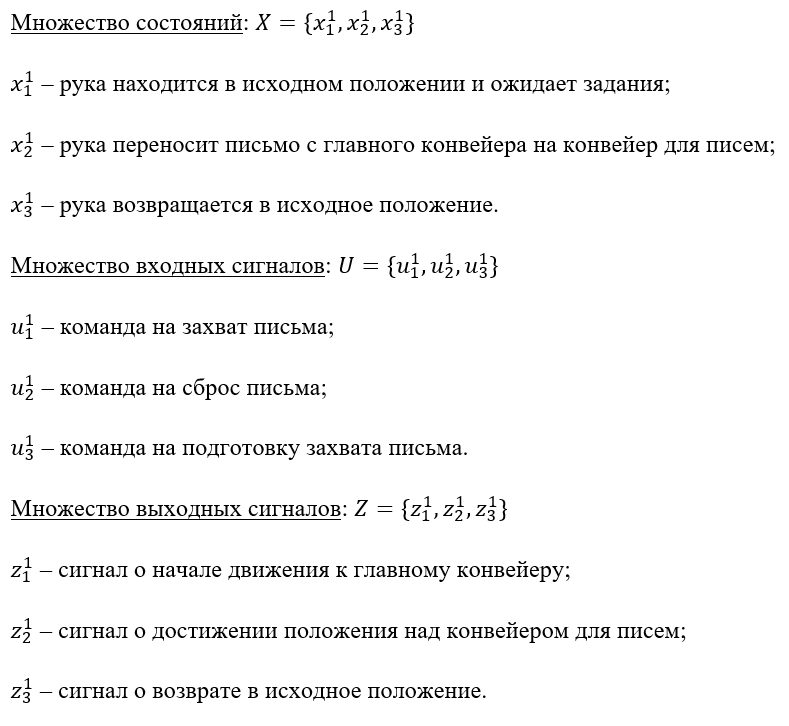


Рис.21. Модель манипуляционного робота для писем

****

Рука манипулятора для бандеролей:

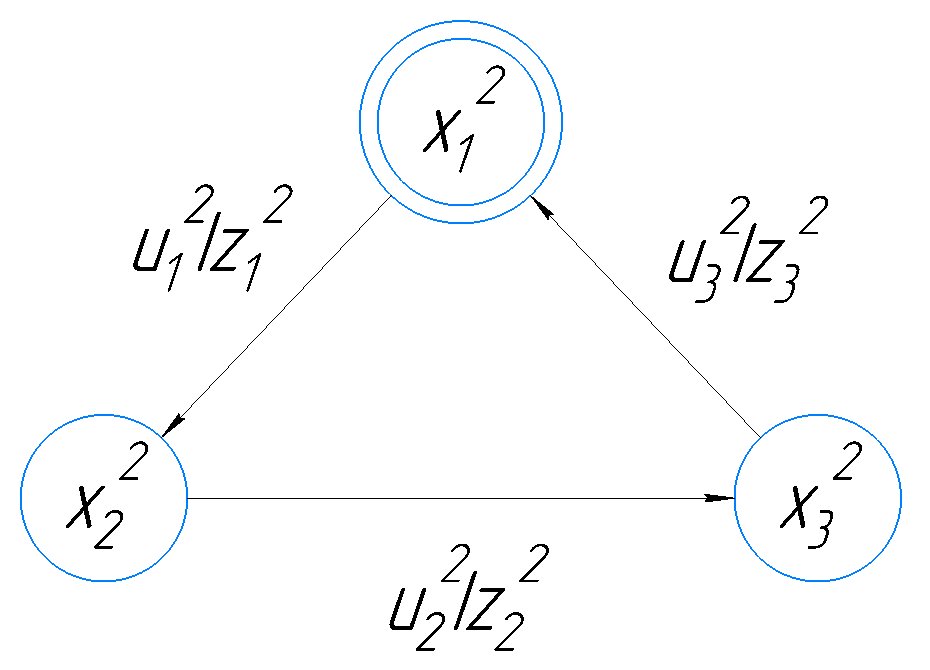
****

Рис. 22. Модель манипуляционного робота для бандеролей

**Изображение выглядит как текст, снимок экрана, Шрифт, документ

Автоматически созданное описание**

СТЗ для проверки наличия почтового отправления на конвейере:

****

Рис. 23. Модель СТЗ для проверки наличия почтового отправления

Изображение выглядит как текст, снимок экрана, Шрифт, документ

Автоматически созданное описание

– на главном конвейере присутствует предмет;

– значение массы переданы в управляющий компьютер;

– размеры предмета и его ориентация переданы в управляющий компьютер.

## **7.2 Управляющий автомат и сеть Петри**

В процессе работы над проектом были разработаны управляющий автомат и сеть Петри для рассматриваемого робототехнического комплекса. В качестве основного элемента управления будем использовать одну из разновидностей автомата с переменной структурой. Сеть Петри описывает алгоритм работы РТК. На схеме наглядно видно ветвление при выборе между рукой для писем и рукой для бандеролей

Структура управляющего автомата УА.

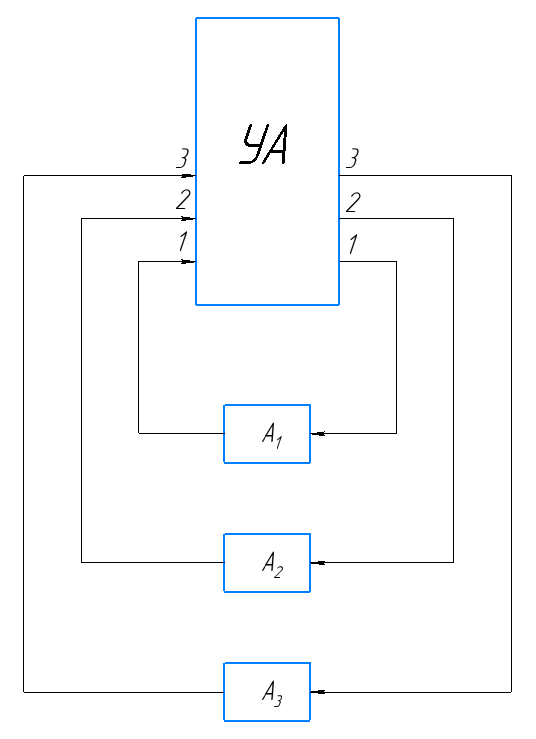


Рис. 24. Структура управляющего автомата УА

СТЗ для проверки наличия почтового отправления;

– модель манипуляционного робота для писем;

– модель манипуляционного робота для бандеролей.

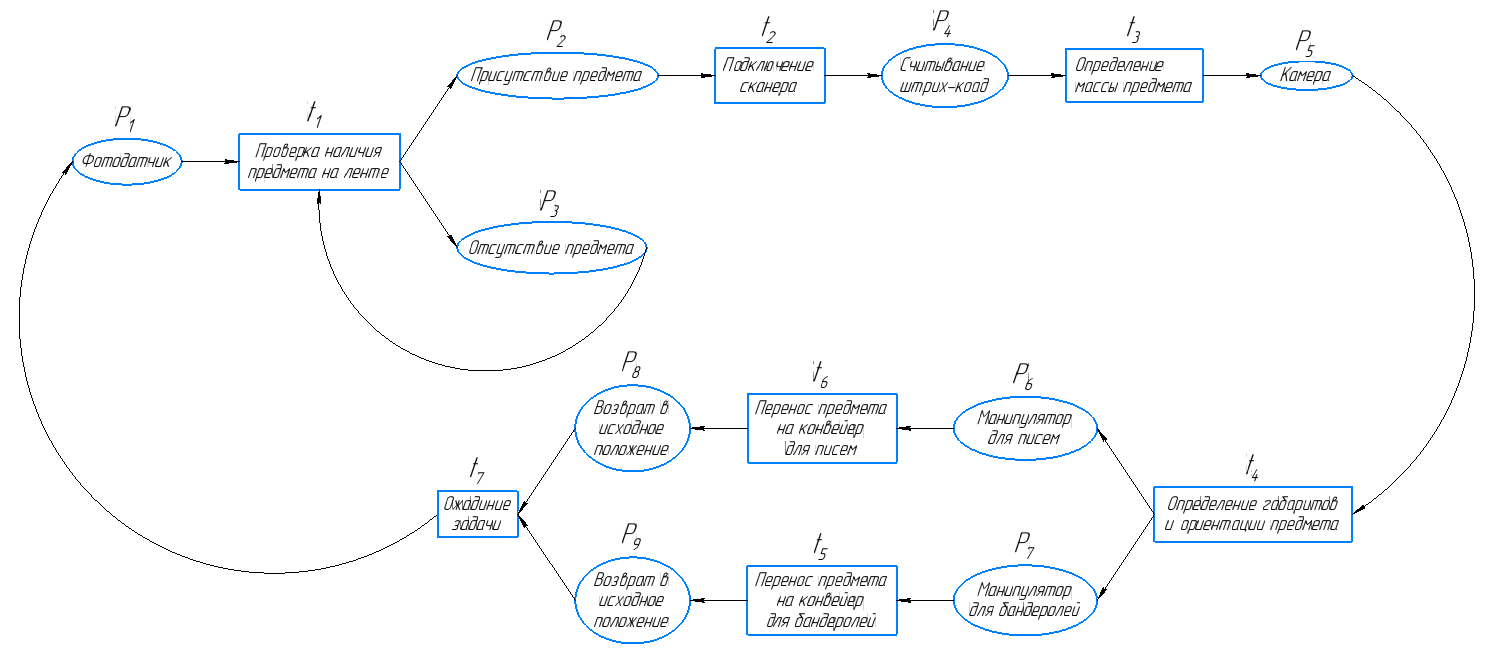


Рисунок 25 - Сеть Петри для всей системы

# **РЕГУЛИРОВОЧНЫЙ РАСЧЁТ ПРИВОДА ВРАЩЕНИЯ СХВАТА РУКИ МАНИПУЛЯТОРА**

Регулировочный расчёт привода проводится с целью подбора регуляторов по положению, скорости и току. Регуляторы позволяют улучшить отработку приводом входного воздействия: регулятор может сделать неустойчивую систему устойчивой, ускорить устранение ошибки или предотвратить перерегулирование. Для рассматриваемого привода вращения схвата руки манипулятора рассматривается схема с тремя контурами. Параметры привода взяты из каталога maxon. Структурная схема привода представлена на Рис. 26.

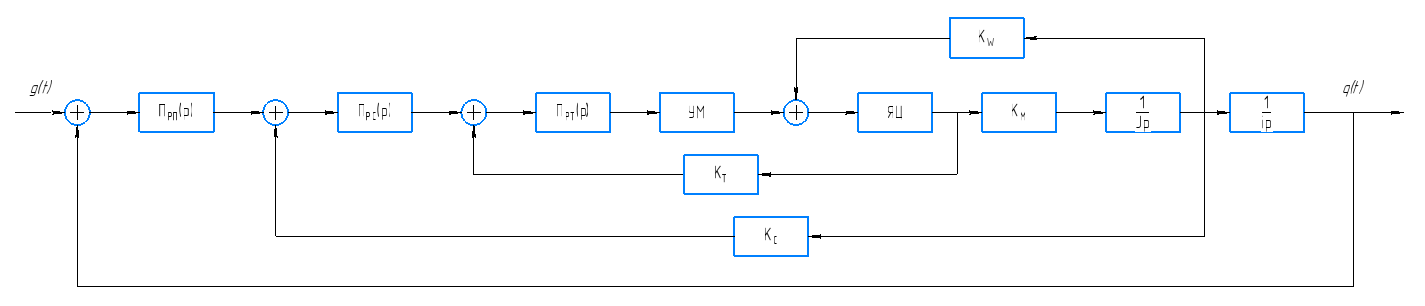


Рис. 26. Функциональная схема привода

По каталогу Maxon для двигателя Maxon EC-max 30 модель 272763, рекомендуется контроллер ESCON Module 50/5Изображение выглядит как текст, снимок экрана, Параллельный, линия

Автоматически созданное описание

Рис. 27 – ESCON Module 50/5

Задача состоит в том, чтобы определить вид регуляторов электропривода вращения схвата руки манипулятора для бандеролей и рассчитать параметры этих регуляторов. Расчёт был проведён для выбранного ранее в результате энергетического расчёта привода электропривод EC-max 30 60 Вт от производителя Maxon Motors.

Изображение выглядит как текст, диаграмма, снимок экрана, Параллельный

Автоматически созданное описание

Рис. 28 – Паспортные данные EC-max 30

Изображение выглядит как текст, снимок экрана, диаграмма, Параллельный

Автоматически созданное описание

Рис. 29 – Паспортные данные GP 42 C

**Основные параметры:**

* Сопротивление цепи ротора Rя – 1.27 Ом
* Индуктивность цепи ротора Lя – 0.000143 Гн
* Моментная постоянная Км – 0.0243 Н\*м/А
* Скоростная постоянная Кw – 0.0243 рад/с/В
* Момент инерции Jд – 2.19\*10-6 Кг\*
* Момент нагрузки J’н – 8\*10-6 – 2.08\*10-5 Кг\*
* Время разгона и торможения Тр, Тт – 0.5 с
* Время движ. с пост. Скоростью Тпс – 2 с
* Крутизну характеристики датчика тока Кт - 1 В/A
* Крутизну характеристики датчика скорости Кс - 1 В\*с/рад
* Коэффициент усиления усилителя мощности Ку – 10
* Постоянную времени усилителя мощности Ту - 0,0004 с

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

* Независимость установившейся скорости привода от внешнего момента;
* 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;

Изображение выглядит как диаграмма, линия, Технический чертеж, Прямоугольник

Автоматически созданное описание

Рис. 30 – Генератор траектории

Расчет параметров неизменяемой части привода:

* Суммарный момент инерции
* Электромеханическая постоянная времени

Усилитель мощности (УМ)

Якорная цепь (ЯЦ)

### **8.1 Расчет контура тока**

При управлении двигателем особое внимание нужно уделять точности работы контура тока, ведь от его работы зависят все остальные контуры рассматриваемой модели привода. В результате расчёта нужно обеспечить перерегулирование Мт =1. Настройка производится на модульный оптимум (МО).

Общий коэффициент усиления:

Для компенсации большей постоянной времени примем:

Коэффициент усиления регулятора тока РТ

Результаты расчета контура тока:

Изображение выглядит как диаграмма, План, Технический чертеж, зарисовка

Автоматически созданное описание

Рисунок 31 – Структурная схема контура тока

Изображение выглядит как текст, График, линия, диаграмма

Автоматически созданное описание Изображение выглядит как текст, График, линия, снимок экрана

Автоматически созданное описание

а) б)

Рис. 32 – а) по **напряжениям**; б) по **току**. При

Изображение выглядит как текст, линия, График, диаграмма

Автоматически созданное описание

Рис. 33 – График **ошибки**. При

### **8.2 расчет контура скорости с контуром тока**

* Требования к качеству переходного процесса

Мск = 2 – настройка на СО (быстрая отработка возмущающего воздействия)

* Требования по быстродействию

Дополнительный астатизм по управлению и возмущающему воздействию контура скорости обеспечим введением интегральной составляющей в регулятор скорости. Лучше остальных себя показывают системы, в которых присутствуют ПИ регулятор тока и ПИ регулятор скорости. В этих системах обеспечивается второй порядок астатизма по управлению и первый порядок астатизма по возмущающему воздействию. В случае с выбранным выше ПИ регулятором скорости получаем, что установившаяся скорость не зависит от внешнего момента.

В итоге были выбраны ПИ регулятор контура скорости и ПИ регулятор контура тока. Единственная возможная настройка данной системы – настройка на СО.

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

Изображение выглядит как диаграмма, линия, текст, Шрифт

Автоматически созданное описание

Рис. 34 – Анализ ЛАЧХ системы

Выберем так, чтобы

**Рассмотрим «ВС»:**

A(

* Постоянная времени регулятора скорости РС

**Рассмотрим «AB»:**

* Коэффициент усиления регулятора скорости РС
* Результаты расчета контура скорости:

Изображение выглядит как диаграмма, Технический чертеж, План, схематичный

Автоматически созданное описание

Рис. 35 – Структурная схема контура скорости с контуром тока

Изображение выглядит как текст, линия, График, диаграмма

Автоматически созданное описание Изображение выглядит как текст, График, линия, диаграмма

Автоматически созданное описание

а) б)

Рис. 36 – а) по **напряжениям**; б) по **току**. При

Изображение выглядит как текст, График, линия, снимок экрана

Автоматически созданное описание

Рис. 37 – График **ошибки** для контура скорости. При

Изображение выглядит как текст, График, линия, диаграмма

Автоматически созданное описание

Рис. 38 – **Скорость** вала двигателя. При

Изображение выглядит как текст, линия, диаграмма, График

Автоматически созданное описание

Рис. 39 – **ЛАФЧХ** для контура скорости

### **8.3 Расчёт контура положения в контурном режиме**

Исходные данные: 𝐸𝑚𝑎𝑥 = 0.01 рад, 𝑞𝑝′ = 1.54 рад/с, 𝑞𝑝′′ = 9.14 рад/с2.

Ввиду того что поэтому

Изображение выглядит как линия, диаграмма, График, текст

Автоматически созданное описание

Рис. 40 – Анализ ЛАЧХ системы

Выбираем «П» регулятор

Для нормальной работы нужно чтобы: 4

Как видно, условие не удовлетворено, поэтому выбираем «ПИ» регулятор контура положения:

Коэффициент усиления регулятора положения РП

Постоянная времени регулятора положения РП

Результаты расчета контура положения:

Изображение выглядит как диаграмма, текст, Технический чертеж, План

Автоматически созданное описание

Рис. 19 – Структурная схема контура положения с контуром скорости и тока

Изображение выглядит как текст, диаграмма, График, линия

Автоматически созданное описание

Рис. 41 – **Напряжения** в якоре при

Изображение выглядит как текст, График, диаграмма, линия

Автоматически созданное описание

Рис. 42 – **Ток** в якоре при

Изображение выглядит как текст, линия, диаграмма, График

Автоматически созданное описание

Рис. 43 – **Скорость** вала двигателя. Отработка трапециевидного сигнала в контуре положения при

Изображение выглядит как текст, График, линия, диаграмма

Автоматически созданное описание

Рис. 23 – График **ошибки**

Изображение выглядит как текст, График, линия, диаграмма

Автоматически созданное описание

Рис. 44 – Отработка входного сигнала контура положения

Изображение выглядит как текст, линия, диаграмма, График

Автоматически созданное описание

Рис. 45 – **ЛАФЧХ** для контура положения

# **ЗАКЛЮЧЕНИЕ**

В ходе проделанной работы были получены следующие результаты:

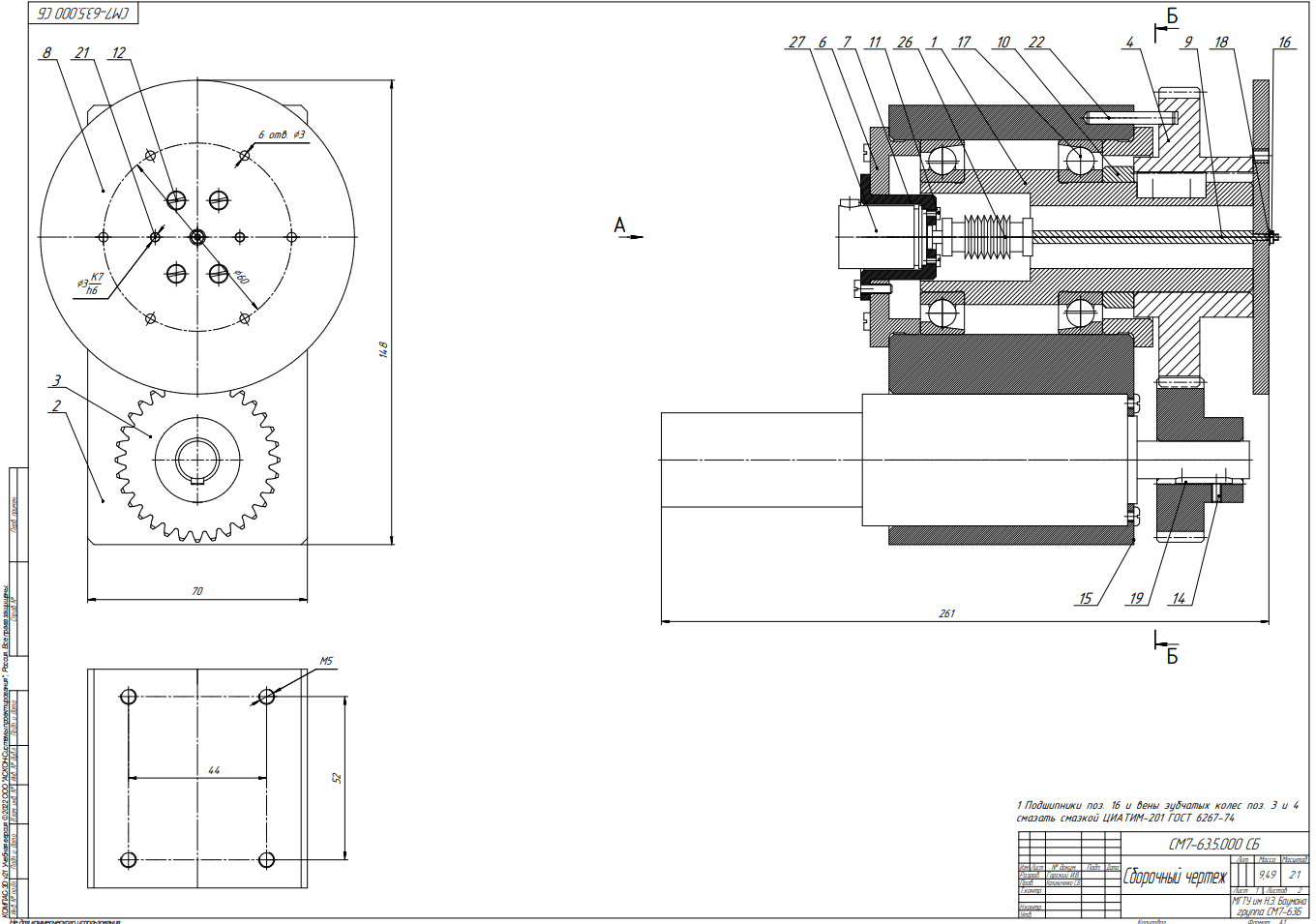
1. Проведен обзор проекта на актуальность исследование и анализ существующих двуруких манипуляторов для сортировки;
2. Разработана функциональная схема управления роботом;
3. Разработана структурная схема руки;
4. Определенны компоненты исполнительных систем;
5. Разработан привод вращения захватного устройства руки манипулятора;
6. Сконструирована реалистичная модель правой руки манипулятора для тяжёлых почтовых отправлений;
7. Разработана схема поворотного механизма базы манипулятора;
8. Построен управляющий конечный автомат РТК и сеть Петри для всей системы;
9. Проведён регулировочный расчёт контуров привода вращения захватного устройства руки манипулятора.

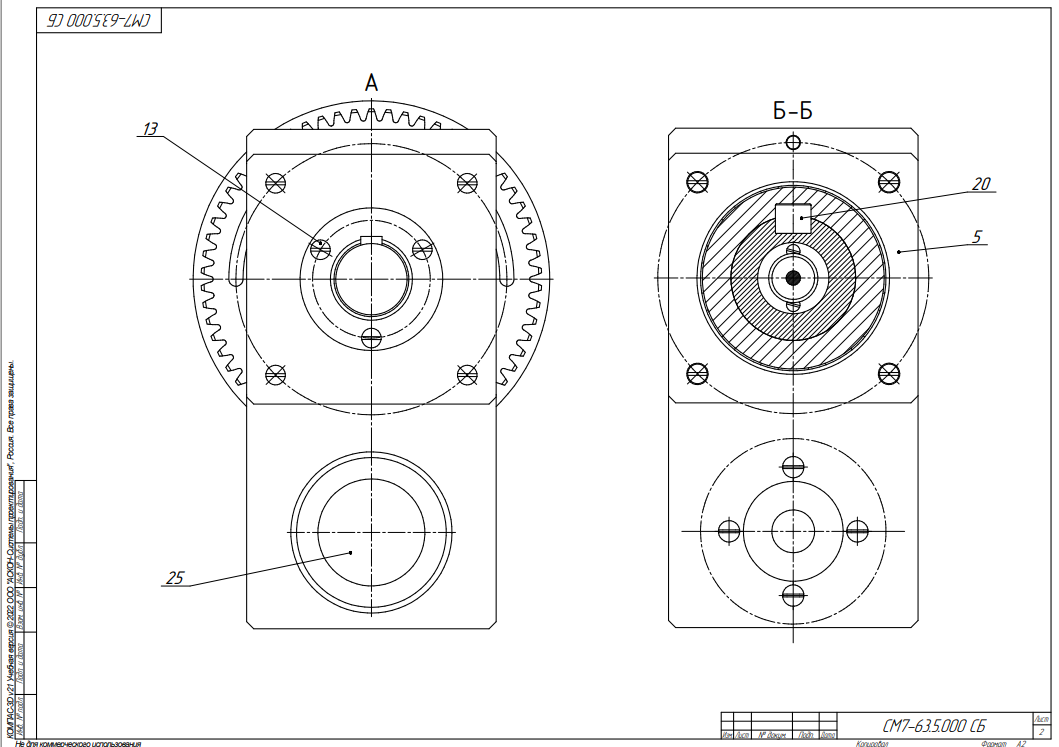
# **СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

1. Дунаев, П. Ф. Конструирование узлов и деталей машин [Текст]: Учеб. пособие для студ. техн. спец. вузов/ П. Ф. Дунаев, О. П. Леликов — Изд. 8-е перераб. и доп. – М.: Издательский центр «Академия», 2004., – 496 с. ISBN 5-7695-1041-2
2. Байков, Б. А. Детали машин: Атлас конструкций [Изоматериал]: Учеб. пособие для студентов машиностроительных специальностей вузов. В 2-х ч. Ч. 1 / Б. А. Байков, В. Н. Богачев, А. В. Буланже и др.: под общ. ред. д-ра техн. наук проф. Д. Н. Решетова. — Изд. 5-е, перераб. и доп. М.: Машиностроение, 1992. — 352 с: ил. ISBN 5-217-01507-1.
3. Леликов О. П. Основы расчета и проектирования деталей и узлов машин. Конспект лекций по курсу «Детали машин». Изд. 3-е переработанное и доп. — М.: Машиностроение, 2007., — 464 с. ISBN 978-5-217-03390-4.
4. Детали машин: учебник для вузов /Л.А.Андриенко, Б.А.Байков, И.К.Ганулич и др.; под. ред. О.А.Ряховского. 2-е изд. М.: Изд-во МГТУ им. Н.Э.Баумана, 2004. 520 с.
5. Перель Л.Я., Филатов А.А. Подшипники качения. Справочник, 2-е издание, - М.; Машиностроение, 1992., — 347 с. ISBN 945-5-217-030-3.

# **ПРИЛОЖЕНИЕ А**

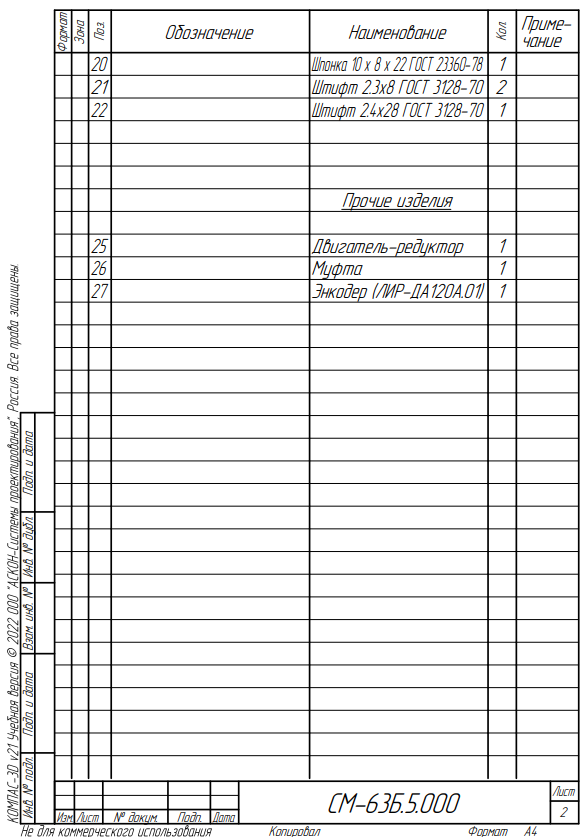
В приложении представлены сборочный чертёж привода и спецификация.





Изображение выглядит как текст, Параллельный, чек, документ

Автоматически созданное описание



# **ПРИЛОЖЕНИЕ Б**

В приложении представлены коды различных программ, написанных в результате выполнения работы.

Функции, использованные при моделировании совместной работы приводов:

function [q1, q2, q3] = OZP1(x, y)

l1 = 0.8;

l2 = 0.7;

l3 = 0.1;

h1 = x^2 + (y + l3)^2;

alpha = atan((y + l3) / x);

q2 = -acos((h1 - l1^2 - l2^2) / (2 \* l1 \* l2));

q1 = acos(abs(l2^2 - h1 - l1^2) / (2 \* sqrt(h1) \* l1)) + alpha;

q3 = -pi/2 - q1 - q2;

end

function [J1, J2, J3] = Inertia(q1, q2, q3)

l1 = 0.8;

l2 = 0.7;

l3 = 0.1;

m1 = 30;

m2 = 20;

m3 = 25;

x1 = cos(q1)\*l1;

y1 = sin(q1)\*l1;

x2 = x1 + cos(q1+q2)\*l2;

y2 = y1 + sin(q1+q2)\*l2;

x3 = x2 + cos(q1+q2+q3)\*l3;

y3 = y2 + sin(q1+q2+q3)\*l3;

I1 = (1/12 \* m1 \* (l1)^2) + 1/4 \* m1 \* (l1)^2;

I2 = (1/12 \* m2 \* (l2)^2) + m2 \* ((cos(q1) \* l1 + cos(q1 + q2) \* l2 / 2)^2 + ...

(sin(q1) \* l1 + sin(q1 + q2) \* l2 / 2)^2);

I3 = (1/12 \* m3 \* (l3)^2) + m3 \* (((x2 + x3) / 2)^2 + ((y2 + y3) / 2)^2);

J1 = I1 + I2 + I3;

I4 = (1/12 \* m2 \* (l2)^2) + m2 \* (((-x1 + x2) / 2)^2 + ((-y1 + y2) / 2)^2); % I2

I5 = (1/12 \* m3 \* (l3)^2) + m3 \* (((x2 + x3) / 2 - x1)^2 + ((y2 + y3) / 2 - y1)^2); % I3

J2 = I4 + I5;

I6 = (1/12 \* m3 \* (l3)^2) + m3 \* (((x2 + x3) / 2 - x2)^2 + ((y2 + y3) / 2 - y2)^2); % I3

J3 = I6;

function [Mg1, Mg2, Mg3] = Gravitation(q1, q2, q3)

g = 9.81;

l1 = 0.8;

l2 = 0.7;

l3 = 0.1;

m1 = 30;

m2 = 20;

m3 = 25;

x1 = cos(q1)\*l1;

x2 = x1 + cos(q1+q2)\*l2;

x3 = x2 + cos(q1+q2+q3)\*l3;

G1 = -m1\*g\*(x1)/2;

G2 = -m2\*g\*(x1+x2)/2;

G3 = -m3\*g\*(x2+x3)/2;

Mg1 = G1 + G2 + G3;

G4 = -m2\*g\*((x1+x2)/2 -x1);

G5 = -m3\*g\*((x2+x3)/2 - x1);

Mg2 = G4 + G5;

G6 = -m3\*g\*(x3-x2)/2;

Mg3 = G6;

end

function [Mf1, Mf2, Mf3, Mf4] = InertialM(X, Y, w0, w1, w2, w3, e1, e2, e3, holdingBox)

l1 = 0.8;

l2 = 0.7;

l3 = 0.1;

m1 = 30;

m2 = 20;

m3 = 25;

Ax = X(2);

Ay = Y(2);

Bx = X(3);

By = Y(3);

Cx = X(4);

Cy = Y(4);

Xc = [(Ax) / 2, (Ax + Bx) / 2, (Bx + Cx) / 2];

Yc = [(Ay) / 2, (Ay + By) / 2, (By + Cy) / 2];

vA = w1\*l1;

vAx = -vA\*Ay/l1;

vAy = vA\*Ax/l1;

vBA = (w1+w2)\*l2;

vBAx = -vBA\*(By - Ay)/l2;

vBAy = vBA\*(Bx - Ax)/l2;

vBx = vAx + vBAx;

vBy = vAy + vBAy;

vCB = (w1+w2+w3)\*l3;

vCBx = -vCB\*(Cy - By)/l3;

vCBy = vCB\*(Cx - Bx)/l3;

vCx = vBx + vCBx;

vCy = vBy + vCBy;

aAt = e1\*l1;

aAn = (vA^2)/l1;

aAx = -aAt\*(Ay)/l1 - aAn\*(Ax)/l1;

aAy = aAt\*(Ax)/l1 - aAn\*(Ay)/l1;

aBAt = (e2+e1)\*l2;

aBAn = ((w2+w1)^2)\*l2;

aBAx = -aBAt\*(By - Ay)/l2 - aBAn\*(Bx - Ax)/l2;

aBAy = aBAt\*(Bx - Ax)/l2 - aBAn\*(By - Ay)/l2;

aBk = abs(2\*w1\*vBA);

aBkx = -aBk \* vBAy \* sign(w1) / abs(vBA);

aBky = aBk \* vBAx \* sign(w1) / abs(vBA);

aBx = aAx + aBAx; %+ aBkx;

aBy = aAy + aBAy; %+ aBky;

aCBt = (e1+e2+e3)\*l3;

aCBn = ((w1+w2+w3))\*l3;

aCBx = -aCBt\*(Cy - By)/l3 - aCBn\*(Cx - Bx)/l3;

aCBy = aCBt\*(Cx - Bx)/l3 - aCBn\*(Cy - By)/l3;

aCk = abs(2\*(w1+w2)\*vCB);

aCkx = -aCk \* vCBy \* sign(w2) / abs(vCB);

aCky = aCk \* vCBx \* sign(w2) / abs(vCB);

aCx = aBx + aCBx;% + aCkx;

aCy = aBy + aCBy;% + aCky;

aC1x = (aAx)/2;

aC1y = (aAy)/2;

aC2x = (aAx + aBx)/2;

aC2y = (aAy + aBy)/2;

aC3x = (aCx + aBx)/2;

aC3y = (aCy + aBy)/2;

acx = -Xc\*(w0^2);

Fcx = -acx.\*[m1, m2, m3];

H1 = Yc;

H2 = Yc - Ay;

H3 = Yc(3) - By;

Mf1 = - Fcx(1)\*H1(1) - Fcx(2)\*H1(2) - Fcx(3)\*H1(3);

Mf2 = - Fcx(2)\*H2(2) - Fcx(3)\*H2(3);

Mf3 = - Fcx(3)\*H3;

if(holdingBox)

Mf4 = 20\*Cx\*(w0^2)\*0.1;

else

Mf4 = 0;

end

function [X, Y] = PZP1(q1, q2, q3)

l1 = 0.8;

l2 = 0.7;

l3 = 0.1;

X = [0, cos(q1)\*l1, cos(q1)\*l1 + cos(q1 + q2)\*l2, cos(q1)\*l1 + cos(q1 + q2)\*l2 + cos(q1 + q2 + q3)\*l3];

Y = [0, sin(q1)\*l1, sin(q1)\*l1 + sin(q1 + q2)\*l2, sin(q1)\*l1 + sin(q1 + q2)\*l2 + sin(q1 + q2 + q3)\*l3];

%grid on;

%plot([-2, 2], [2, 2]);

%plot([-2, 2], [-2, -2]);

%plot(X, Y);

%hold off;

end

Реализация алгоритма обработки изображения на языке С++:

#include "opencv2/imgcodecs.hpp"

#include "opencv2/highgui.hpp"

#include "opencv2/imgproc.hpp"

#include <stdlib.h>

#include <stdio.h>

#include <Math.h>

#include <iostream>

using namespace cv;

using namespace std;

Mat src\_gray;

int thresh = 100;

RNG rng(12345);

void thresh\_callback(int, void\*);

int main(int argc, char\*\* argv)

{

CommandLineParser parser(argc, argv, "{@input | C:/amyfiles/BMSTU/8 semestr/diplom/img/RB1.jpg | input image}");

Mat src = imread(samples::findFile(parser.get<String>("@input")), 1);

if (src.empty())

{

cout << "Could not open or find the image!\n" << endl;

cout << "Usage: " << argv[0] << " <Input image>" << endl;

return -1;

}

cvtColor(src, src\_gray, COLOR\_BGR2GRAY);

blur(src\_gray, src\_gray, Size(3, 3));

const char\* source\_window = "Source";

namedWindow(source\_window);

imshow(source\_window, src);

const int max\_thresh = 255;

createTrackbar("Canny thresh:", source\_window, &thresh, max\_thresh, thresh\_callback);

thresh\_callback(0, 0);

waitKey();

return 0;

}

double distance(int a, int b, int i, int j)

{

return sqrt(pow(a - i, 2) + pow(b - j, 2));

}

double pi = 3.1415926535897;

double GetDistanceToSegment(double cx, double cy, double ax, double ay, double bx, double by)

{

// cx, cy - координаты точки

// ax, ay, bx, by - координаты прямой

// вычисление длин отрезков

double segmAC = sqrt((cx - ax) \* (cx - ax) + (cy - ay) \* (cy - ay));

double segmBC = sqrt((cx - bx) \* (cx - bx) + (cy - by) \* (cy - by));

double segmAB = sqrt((ax - bx) \* (ax - bx) + (ay - by) \* (ay - by));

// вычисление косинусов

double cosA = (segmAC \* segmAC + segmAB \* segmAB - segmBC \* segmBC) / (2 \* segmAC \* segmAB);

double cosB = (segmBC \* segmBC + segmAB \* segmAB - segmAC \* segmAC) / (2 \* segmBC \* segmAB);

double cosC = (segmBC \* segmBC + segmAC \* segmAC - segmAB \* segmAB) / (2 \* segmBC \* segmAC);

// вычисление углов

double angleA = acos(cosA) \* 180 / pi;

double angleB = acos(cosB) \* 180 / pi;

double angleC = acos(cosC) \* 180 / pi;

// если угол А или B больше 90 градусов то,

// это значит что - нельзя провести перпендикуляр к отрезку AB

if (angleA > 90.00 || angleB > 90.00)

{

return min(segmAC, segmBC);

}

else

{

return abs((by - ay) \* cx - (bx - ax) \* cy + bx \* ay - by \* ax) /

sqrt((by - ay) \* (by - ay) + (bx - ax) \* (bx - ax));

}

}

void thresh\_callback(int, void\*)

{

Mat canny\_output;

Canny(src\_gray, canny\_output, thresh, thresh \* 2);

vector<vector<Point> > contours;

vector<Vec4i> hierarchy;

findContours(canny\_output, contours, hierarchy, RETR\_TREE, CHAIN\_APPROX\_SIMPLE);

size\_t biggestId = 0;

size\_t biggestSize = 0;

for (size\_t j = 0; j < contours.size(); j++)

{

if (contours[j].capacity() > biggestSize)

{

biggestSize = contours[j].capacity();

biggestId = j;

}

}

Mat drawing = Mat::zeros(canny\_output.size(), CV\_8UC3);

/\*for (size\_t i = 0; i < contours.size(); i++)

{

Scalar color = Scalar(rng.uniform(0, 256), rng.uniform(0, 256), rng.uniform(0, 256));

drawContours(drawing, contours, (int)i, color, 2, LINE\_8, hierarchy, 0);

cout << color << endl;

}

\*/

Scalar color = Scalar(200, 200, 200);

drawContours(drawing, contours, (int)biggestId, color, 2, LINE\_8, hierarchy, 0);

int Xcount = 0;

int Xvalue = 0;

int Ycount = 0;

int Yvalue = 0;

for (int i = 0; i < drawing.rows; i++)

{

for (int j = 0; j < drawing.cols; j++)

{

if (drawing.at<Vec3b>(i, j)[0] == 200)

{

Xcount++;

Xvalue += i;

Ycount++;

Yvalue += j;

}

}

}

int centerX = Xvalue/Xcount;

int centerY = Yvalue / Ycount;

int maxLens[4] = {0, 0, 0, 0};

int corners[4][2] = { 0, 0, 0, 0, 0, 0, 0, 0 };

//corner 1

for (int i = 1; i < drawing.rows; i++)

{

for (int j = 1; j < drawing.cols; j++)

{

if (drawing.at<Vec3b>(i, j)[0] == 200)

{

if (distance(i, j, centerX, centerY) > maxLens[0])

{

corners[0][0] = i;

corners[0][1] = j;

maxLens[0] = distance(i, j, centerX, centerY);

}

}

}

}

//corner 2

for (int i = 1; i < drawing.rows; i++)

{

for (int j = 1; j < drawing.cols; j++)

{

if (drawing.at<Vec3b>(i, j)[0] == 200)

{

if (distance(i, j, centerX, centerY) > maxLens[1])

{

if (distance(i, j, corners[0][0], corners[0][1]) > 50)

{

corners[1][0] = i;

corners[1][1] = j;

maxLens[1] = distance(i, j, centerX, centerY);

}

}

}

}

}

//corner 3

for (int i = 1; i < drawing.rows; i++)

{

for (int j = 1; j < drawing.cols; j++)

{

if (drawing.at<Vec3b>(i, j)[0] == 200)

{

if (distance(i, j, centerX, centerY) > maxLens[2])

{

if (distance(i, j, corners[0][0], corners[0][1]) > 50 &&

distance(i, j, corners[1][0], corners[1][1]) > 50)

{

corners[2][0] = i;

corners[2][1] = j;

maxLens[2] = distance(i, j, centerX, centerY);

}

}

}

}

}

//corner 4

for (int i = 1; i < drawing.rows; i++)

{

for (int j = 1; j < drawing.cols; j++)

{

if (drawing.at<Vec3b>(i, j)[0] == 200)

{

if (distance(i, j, centerX, centerY) > maxLens[3])

{

if (distance(i, j, corners[0][0], corners[0][1]) > 50 &&

distance(i, j, corners[1][0], corners[1][1]) > 50 &&

distance(i, j, corners[2][0], corners[2][1]) > 50)

{

corners[3][0] = i;

corners[3][1] = j;

maxLens[3] = distance(i, j, centerX, centerY);

}

}

}

}

}

int closest;

// closest to 0 corner

for (int k = 1; k < 4; k++)

{

int d1 = distance(corners[0][0], corners[0][1], corners[1][0], corners[1][1]);

int d2 = distance(corners[0][0], corners[0][1], corners[2][0], corners[2][1]);

int d3 = distance(corners[0][0], corners[0][1], corners[3][0], corners[3][1]);

if (d1 <= d2 && d1 <= d3)

closest = 1;

else if (d2 <= d1 && d2 <= d3)

closest = 2;

else if (d3 <= d1 && d3 <= d2)

closest = 3;

}

double dirVector[2];

if (corners[0][1] < corners[closest][1])

{

dirVector[0] = corners[closest][0] - corners[0][0];

dirVector[1] = corners[closest][1] - corners[0][1];

}

else

{

dirVector[0] = -corners[closest][0] + corners[0][0];

dirVector[1] = -corners[closest][1] + corners[0][1];

}

// normalization

int vecLen = sqrt(pow(dirVector[0], 2) + pow(dirVector[1], 2));

dirVector[0] = dirVector[0] / vecLen;

dirVector[1] = dirVector[1] / vecLen;

dirVector[0] = dirVector[0] \* 50;

dirVector[1] = dirVector[1] \* 50;

// drawing direction

for (int i = 0; i < drawing.rows; i++)

{

for (int j = 0; j < drawing.cols; j++)

{

if (GetDistanceToSegment(i, j, centerX, centerY, dirVector[0] + centerX, dirVector[1] + centerY) < 3)

{

drawing.at<Vec3b>(i, j)[1] = 20;

drawing.at<Vec3b>(i, j)[0] = 20;

drawing.at<Vec3b>(i, j)[2] = 220;

}

}

}

// drawing corners

for (int i = 0; i < drawing.rows; i++)

{

for (int j = 0; j < drawing.cols; j++)

{

if (distance (i, j, corners[0][0], corners[0][1]) < 10 ||

distance(i, j, corners[1][0], corners[1][1]) < 10 ||

distance(i, j, corners[2][0], corners[2][1]) < 10 ||

distance(i, j, corners[3][0], corners[3][1]) < 10)

{

drawing.at<Vec3b>(i, j)[1] = 20;

drawing.at<Vec3b>(i, j)[0] = 20;

drawing.at<Vec3b>(i, j)[2] = 220;

}

}

}

// drawing center

for (int i = 0; i < drawing.rows; i++)

{

for (int j = 0; j < drawing.cols; j++)

{

if (i == centerX || j == centerY)

{

drawing.at<Vec3b>(i, j)[0] = 200;

}

}

}

for (int i = 0; i < drawing.rows; i++)

{

for (int j = 0; j < drawing.cols; j++)

{

if (drawing.at<Vec3b>(i, j)[1] == 200 &&

drawing.at<Vec3b>(i, j)[0] == 200 &&

drawing.at<Vec3b>(i, j)[2] == 200)

{

drawing.at<Vec3b>(i, j)[1] = 20;

drawing.at<Vec3b>(i, j)[0] = 20;

drawing.at<Vec3b>(i, j)[2] = 20;

}

if (drawing.at<Vec3b>(i, j)[1] == 0 &&

drawing.at<Vec3b>(i, j)[0] == 0 &&

drawing.at<Vec3b>(i, j)[2] == 0)

{

drawing.at<Vec3b>(i, j)[1] = 255;

drawing.at<Vec3b>(i, j)[0] = 255;

drawing.at<Vec3b>(i, j)[2] = 255;

}

}

}

cout << corners[0][0] << endl;

cout << corners[0][1] << endl;

cout << centerX << endl;

cout << centerY << endl;

imshow("Contours", drawing);

}