



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 «Международных образовательных программ»

КАФЕДРА «Робототехнические системы и мехатроника»

РАСЧЕТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА
К ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЕ
НА ТЕМУ:

«Применение нейронных сетей для обучения манипуляционно-
го робота операции захвата объектов»

Студентка группы СМ7И-42М	_____	М. Жамал
	(Подпись, дата)	
Руководитель ВКР	_____	А. С. Ющенко
	(Подпись, дата)	
Консультант по	_____	...
	(Подпись, дата)	
Консультант по	_____	...
	(Подпись, дата)	
Нормоконтролер	_____	...
	(Подпись, дата)	

2020 г.

РЕФЕРАТ

Расчетно-пояснительная записка 73 с., 39 рис., 1 табл., 26 источников, 1 прил.

НЕЙРОННЫЕ СЕТИ, РОБОТИЗИРОВАННЫЙ ЗАХВАТ, РОБОТИЗИРОВАННОЕ УПРАВЛЕНИЕ, КОНВОЛЮЦИОННЫЕ НЕЙРОСЕТИ, ПРОМЫШЛЕННЫЙ МАНИПУЛЯТОР, ИСКУССТВЕННЫЙ ИНТЕЛЛЕКТ, УПРАВЛЕНИЕ МАНИПУЛЯТОРОМ

Объектом разработки является манипулятором ABB IRB 14000 (YuMi)[1].

Цель работы — создание и проектирование системы упаковки набора инструментов манипулятором с помощью искусственных нейронных сетей и изучение возможности и эффективности применения нейронных сетей для обучения манипуляционного робота операции захвата объектов.

СОДЕРЖАНИЕ

РЕФЕРАТ	2
ВВЕДЕНИЕ.....	4
1 Обзор литературы	7
2 Постановка задачи	11
3 Функциональная схема задачи	16
4 Система обнаружения захвата	18
4.1 Предварительная обработка изображения глубины.....	18
4.2 Построение нейросети и её структуры.....	19
4.3 Подготовка обучающего набора данных.....	20
4.4 Экспериментальное исследование и результаты.....	23
5 Система распознавания объекта.....	28
6 Переход в 3D координаты манипулятора.....	29
7 Планирование траектории конечного эффектора манипулятора.....	32
8 Кинематический и динамический анализ манипулятора	37
8.1 Кинематический анализ манипулятора	37
8.2 Динамический анализ манипулятора.....	45
9 Система управления манипулятором и схватом.....	53
10 Моделирование и симуляция системы	63
ЗАКЛЮЧЕНИЕ	69
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	71
ПРИЛОЖЕНИЕ А Графическая часть выпускной квалификационной работы.....	74

ВВЕДЕНИЕ

Манипуляторы являются основным членом в современной промышленности, особенно на автоматизированных заводах, где манипуляторы заняты в операциях пайки и сварки; сборки и разборки; упаковки на последнем шаге производства; сортировки объектов, например, на мусороперерабатывающих заводах (Рисунка 1.а) и т.д.

Требования задач манипулятора отличаются в зависимости от применения. Некоторые задачи манипуляторов как сборка и разборка (Рисунка 1.б) и упаковка (Рисунка 1.с) включают операцию захвата, которая является основной задачей, выполняемой роботом при манипулировании изделия.



Рисунок 1 — применении манипуляторов в промышленности (а), (б) и (с).

До 2004 года, обычно использовали аналитические подходы к решению задачи захвата, но с тех пор появился новый эмпирический подход, это - Машинное Обучение. Машинное обучение зависит от визуальной информацией об объекте и заключается в обучении искусственной нейронной сети правильного положения захвата с помощью огромного набора данных. Машинное обучение стал активной исследовательской области, и в последнее время в ней проводилось много исследований [2],[3],[4],[5]. Обычно исследуют структуру нейронной сети, выборку обучающего набора данных, вид данных входящих на нейросети и метода обучения. Исследование направляется на повышение точности захвата или на повышение скорости и применимости метода в реальном времени или на поиск баланса между этими двумя критериями для достижения желаемой задачи, которая включает операцию захвата.

Исследование применения нейронных сетей для обучения манипулятора операции захвата является очень важным для автоматизации процесса захвата, которая позволяет избавить людей от монотонных и простых операций при процессе производстве, и существенно увеличивает скорость, точность и качество работ и может открываться новые применения для манипулятора, например, применение манипулятора для упаковки различных объектов одного заказа на автоматизированном складе типа Amazon warehouse, где формы и размеры объектов неизвестны за ранее.

Мы решили работать над темой применения нейронных сетей для обучения манипуляционного робота операции захвата объектов, на примере применения «Система упаковки набора инструментов манипулятором на инструментальном заводе», где мы представляем метод применения нейронной сети для распознавания инструментов и обучения правильного положения захвата инструментов с использованием метода обучения “**Transfer learning**”[6], который используется в первый раз, для такого применения «Система упаковки набора инструментов манипулятором». Transfer learning — это метод обучения нейросетей, когда знания нейросети, которая была обучена на одной задаче, переносятся на другую задачу.

Данная система выполняет следующие этапы: Распознавание объектов(инструментов), расположенных в произвольной ориентации в рабочем пространстве не конвейере; Захват объектов манипулятором; Перенос объекта в другое место над коробкой; Определение требуемого положения на основе класса распознанного объекта и размещение его в этом положении на новом месте.

Система упаковки набора инструментов манипулятором с помощью искусственных нейронных сетей, может действовать на заводах инструментов, где операция упаковки является необходимой. Предложенный метод может быть применен ко любой задаче упаковки на автоматизированных заводах, так как в каждой производственной линии, последний шаг является упаковкой. Метод может использоваться ещё и для других применений, как упаковка заказов

медикаментов в электронной аптеки и упаковка заказов на автоматизированных складах как Amazon warehouse.

1 Обзор литературы

Роботизированное захватывание было активным предметом исследований в течение десятилетий [4],[5],[7], на алгоритмы синтеза захвата было потрачено много усилий. Опираясь на данные обзорной статьи [8],[9], методы процесса захвата можно разделить на аналитические и эмпирические (управления данными).

Аналитические методы:

Аналитические методы зависят от анализа механики захвата и контактных взаимодействий «захват – объект». Поскольку целью захвата является удержание объекта, захват должен иметь следующие основные свойства [10]: сопротивление возмущениям, ловкость, равновесие и устойчивость. Задача захвата обычно представляется как задача ограниченной оптимизации по критериям, которые измеряют одно или несколько предыдущих свойств.

Аналитический метод основан на использовании упрощенных контактных моделей. При моделировании доступны точные модели кинематики захвата и объекта. Vicchi и Kumar [11] сформулировали проблему нахождения точной и удобной модели соответствия контактов. Это необходимо для анализа статически неопределенных захватов, в которых не все внутренние силы можно контролировать, как, например, в случае использования неполноприводного схвата (захватного устройства), когда число контролируемых степеней свободы (DOF) меньше числа контактных сил. Prattichizzo [12] моделировал такую систему путем представления набора пружин на контактах и сочленениях.

Селиверстова Е. В. [13] рассмотрела алгоритм выбора способа захвата деформируемого объекта с учетом минимизации контактных сил, превратив задачу захвата в задачу оптимизации (минимизации) сил контакта между объектом и губками захвата, (Рисунок. 2), где доступны точные геометрические и физические модели объекта.



Рисунок 2 — Моделирование контактного взаимодействия при захвате

На практике, робототехнической системе присущи систематические и случайные погрешности, которые обусловлены шумом датчиков, а также неточными моделями кинематики и динамики робота, датчиков или объекта. Таким образом, относительное положение объекта и захвата может быть известно только приблизительно, что затрудняет точное размещение губок захвата. Аналитический метод не обладает достаточной робастностью, так как каждый анализ направлен на определенную модель или тип объектов и не способен справиться с новыми неизвестными объектами. Vicchi и Kumar [11] показывали отсутствие подходов к синтезу захватов, которые устойчивы к ошибкам позиционирования.

Эмпирические методы:

Эмпирические методы начали распространяться с 2004 года с появлением нового программного обеспечения, например Graspit! [14]. ПО предназначено для симуляции захвата и окружающей среды с возможности контролировать среду и проводить огромное количество экспериментов эффективно без доступа к дорогостоящему оборудованию робототехники, что также добавило бы большую сложность процессу оценки. Всё это позволило разработать и оценить новые способы захвата.

Эмпирические методы обычно используют машинное обучение для разработки моделей, которые отображают показания роботизированных датчиков непосредственно на метки успеха от людей [15] или метки успеха в физических испытаниях (способность поднимать объект) [16]. В отличие от аналитических, эмпирические методы зависят от представления объекта и перцептивной обра-

ботки, например, извлечение особенности, сходство матриц, распознавание или классификация объектов и оценка положения. Полученные данные затем используются для извлечения захватов из какой-либо базы данных или для выбора и их оценки по сравнению с существующим опытом.

Глубокое обучение в последнее время значительно продвинулось вперед в применении компьютерного зрения и понимания сцен, поэтому наблюдается большая тенденция к осуществлению методов глубокого обучения для решения задач роботизированного захватывания. Kumra и Kanan [17] рассмотрели задачу обнаружения наилучшего способа захвата, используя глубокое обучение с учетом RGB изображений и глубины объектов. Они предложили мультимодальную модель, которая использует две глубокие конволюционные (сверточные) нейронные сети (ResNet), каждая из которых состоит из 50 Residual слоев (Рисунок. 3). Одна сеть используется для извлечения особенностей из RGB-изображения, а вторая – для извлечения особенностей из изображения-глубины, а затем использует мелкую конволюционную нейронную сеть для определения конфигурации захвата для интересующего объекта. Их мультимодальная модель достигла точности 89,21% на стандартном наборе данных Cornell [18] и работает в реальном масштабе времени.

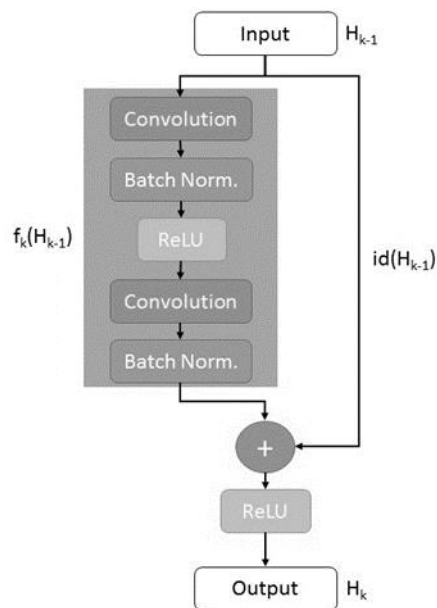


Рисунок 3 — Пример Residual блок в ResNet[16]

Mahler и другие [19] обучили модель конволюционной нейронной сети качества захвата Grasp Quality Convolutional Neural Network (GQ-CNN), которая быстро предсказывает вероятность успеха захватов. На вход нейросети подается изображение глубины объекта с несколькими вариантами захвата. Захваты задаются как плоское положение с углом и глубиной захвата относительно датчика RGB-D. После обучения на наборе данных Dex-Net 2.0 [18], сеть достигла успеха 93% на восьми известных объектах за 0,8 с. Авторы указали одну из проблем, связанную с выбором захвата из фиксированного набора вариантов. Она заключается в том, что все эти варианты могут иметь низкую вероятность успеха. Решить проблему можно методом перекрестной энтропии.

Можно заметить, что исследование [17], [19] больше фокусируется на точности сети, а не на применении нейронной сети в реальном времени. Нейросети [19] надо 0,8 с, что является не малым времени и может трудиться применения нейросети в реальном. А нейросеть в [17] состоит от большого номера слоев, и авторы не представили время работа нейросети. Наша работа направлена на увеличение скорости и применимости метода в реальном времени и на использование метода обучения “**Transfer learning**”[6], который используется в первый раз, для достижения задачи захвата при выполнении желаемой задачи упаковки набора инструментов.

2 Постановка задачи

Операция упаковки набора инструментов относится к типу операции «pick and place», на котором робот выполняет следующие этапы:

- Распознавание объектов(инструментов), расположенных в произвольной ориентации в рабочем пространстве не конвейере;
- Захват объектов манипулятором;
- Перенос объекта в другое место над коробкой (Рисунок. 4);
- Определение требуемого положения на основе класса распознанного объекта и размещение его в этом положении на новом месте.

На данной работе, нейронная сеть используется для распознавания инструмента и определения правильного положения его захвата.



Рисунок 4 — Набор инструментов

Для реализации цели исследования были поставлены некоторые **задачи**:

1. Разработка системы распознавания объекта и обнаружения правильного положения захвата с помощью нейронной сети.
 - a. Построение нейросети и выбор её структуры.
 - b. Подготовка обучающего набора.
 - c. Обучение нейросети с помощью метода Transfer learning.
 - d. Оценить точность нейросеть.
2. Кинематический и динамический анализ манипулятора.
3. Планирование траектории движения манипулятора при выполнении задачи.

4. Разработка системы управления манипулятора для выполнения задачи упаковки и настройка (tuning) её параметры с помощью MATLAB.
5. Моделирование системы упаковки манипуляционным роботом в пакете ROS.

Описание задачи: Объекты поступают на конвейере в произвольном порядке. Шаговый конвейер подвигается на шаг, манипулятор распознает объект, правильно его захватывает и складывает в коробку.

Для иллюстрации работы системы, были выбраны и проектированы CAD-модели для инструментов шесть объектов: отвёртка, молоток, ключ рожковый, плоскогубцы, изоляционная лента, строительный нож. Предметы имеют трехмерную форму и весят 50 – 300 г. На приложении А, показаны эскизы объектов и коробки, все размеры указаны в мм.

Для выполнения задачи был выбран робот ABB IRB 14000 (YuMi) [1] (Рисунок. 5), так как он обладает рядом преимуществ. Особенность робота заключается в том, что он является коллаборативным. Робот оснащен двумя руками, каждая из которых имеет 7 степеней свободы, что обеспечивает большую ловкость и универсальность. Полезная нагрузка робота 0,5 кг, что считается достаточным для удерживания инструментов.



Рисунок 5 — Робот ABB YuMi. Рисунок от [1]

В соответствии с [20], захват будет определяться прямоугольником (Рисунок. 6) в плоскости изображения четырьмя параметрами:

u, v – координаты центра прямоугольника в изображении (в пикселях).
 φ – угол поворота на относительно горизонтальной линии.
 w – Ширина захвата.

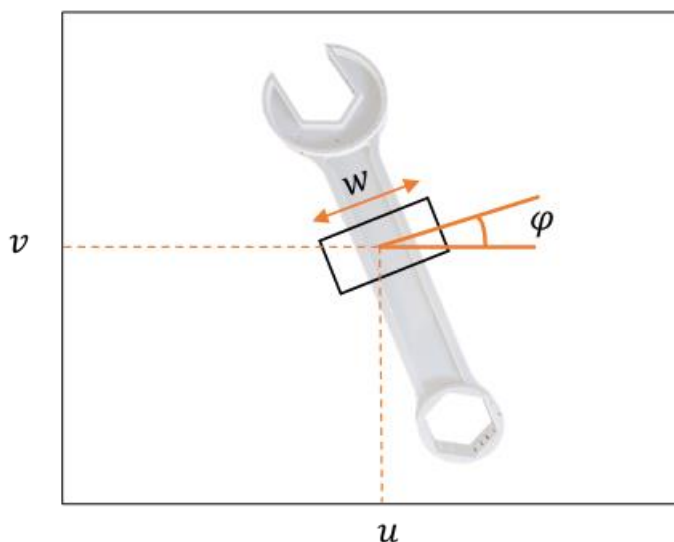


Рисунок 6 — Обозначение захвата прямоугольником

Описание окружающей среды работы:

При разработке роботизированной системы, необходимо детально описать среду, в которой она будет работать. Эти детали помогают уточнить выбор устройств в системе. Например, для камеры нужно удовлетворить требованиям по разрешению, а для манипулятора обеспечить подходящее рабочее пространство.

Окружающая среда включает в себя конвейер, который подает объекты и коробку, в которую манипулятор укладывает инструменты. Робот должен располагаться в рабочем пространстве таким образом, чтобы ему были доступны и объекты на конвейере, и коробка.

Если робот будет располагаться на уровне конвейера (Рисунок. 7), то рабочее пространство манипулятора охватит ограниченное пространство эффективной поверхности конвейера и коробки. Поэтому целесообразно расположить манипулятор на 20 см ниже уровня конвейера, как показано на рисунке 8, так манипулятору будут доступны и объекты на конвейере, и коробка.

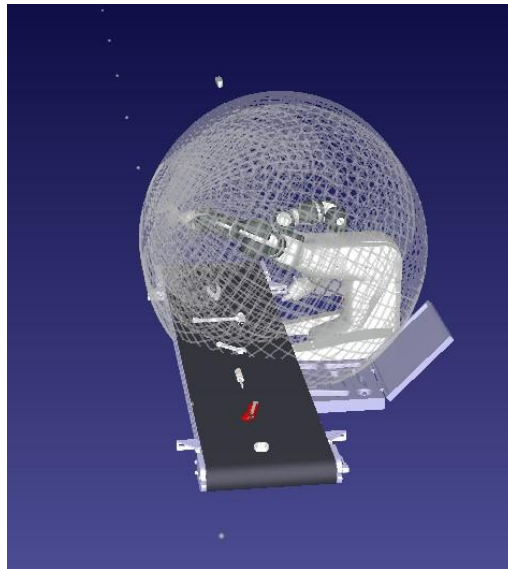


Рисунок 7 — Манипулятор YuMi на уровне конвейера.

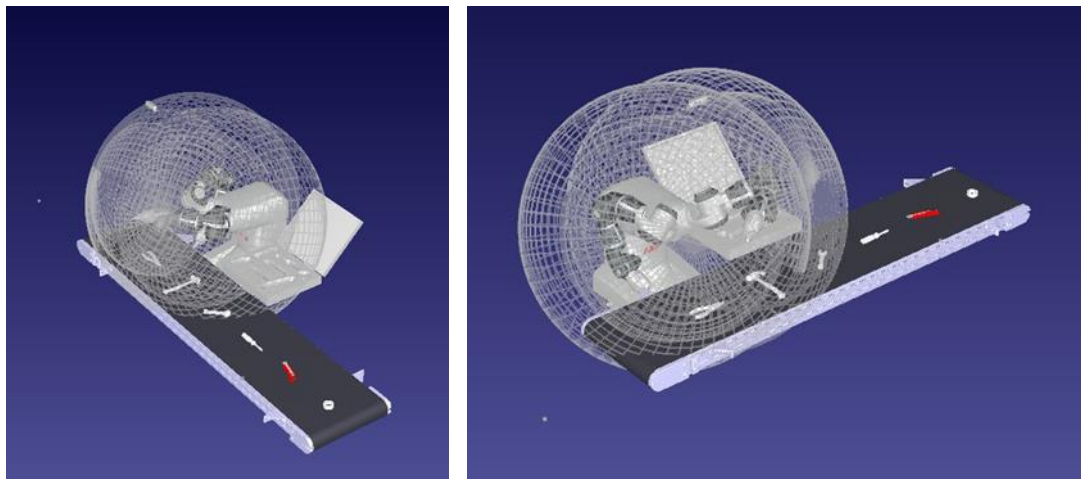


Рисунок 8 — Рабочее пространство манипулятора с двух точек зрения.

Для получения изображения глубины объекта используется камера kinect. Камера должна располагаться над объектом не меньше, чем 0,7 м, для того, чтобы объект попал в поле видимости. На рисунке 9, показан угол обзора камеры.

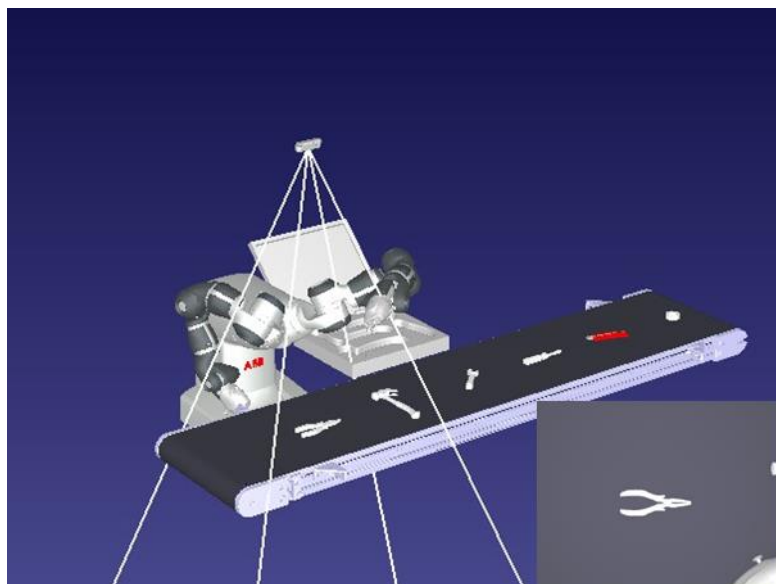


Рисунок 9 — Угол обзора камеры

Предположим, что захват всегда будет вертикальным относительно плоскости стола. Дополнительными ограничениями управления в зависимости от нагрузки, как видно из диаграммы (Рисунок. 10), являются ограничения на положения конечного эффектора (схвата) по осям x и z .

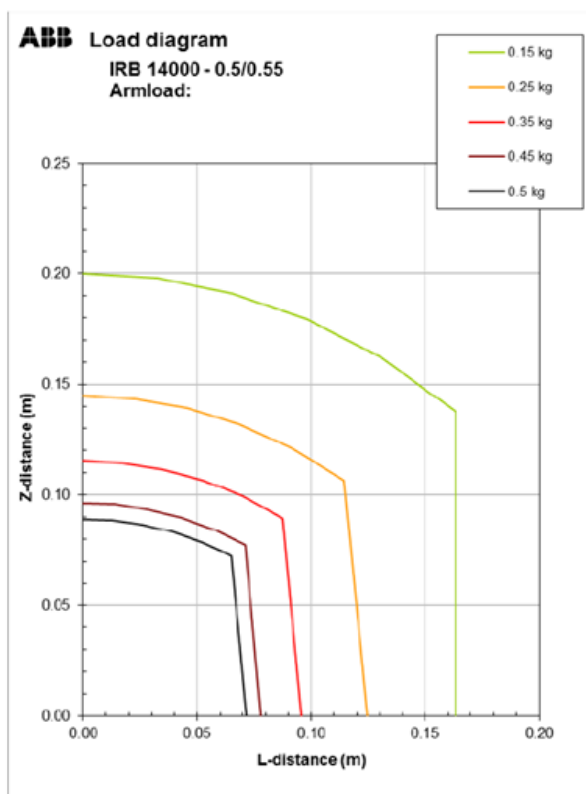


Рисунок 10 — load diagram Robot IRB 14000 (YuMi)

3 Функциональная схема задачи

Функциональная схема системы управления манипулятором выполняющей операцию упаковки, показана на Рисунке 11. Процесс начинается с получения изображения глубины объекта с камеры после одного шага конвейера. Далее происходит его предварительная обработка (удаления недопустимых значений и нормализации). Обработанное изображение подается на вход двух нейросетей, чтобы определить параметры лучшего захвата и распознать объект (класс объекта). На выход первой нейросети будет лучшая поза захвата в координатах изображения глубины, обозначающая координатами пиксела лучшего захвата (u, v) , и углом поворота $\tilde{\varphi}$, которые представлены ранее. Далее происходит переход в 3D координаты манипулятора используя матрица перехода, и получим вектор координат конечного эффектора (схват), нужные для выполнения лучшего захвата: $[x \ y \ z \ 0 \ \pi \ \varphi]$. Углы поворота около x и y выбираются таким образом, чтобы обеспечить вертикальный захват на поверхности конвейера. На выход второй нейросети будет класс объекта, обозначенный номером с 0 до 5. Далее, класс объекта и вектор координат его лучшего захвата подаются на вход блока планирования траектории конечного эффектора манипулятора, на выход которого будет вектор траектории конечного эффектора $P(t)$, по которой конечный эффектор проходит, захватит инструмента и переносит его в его нужное место и положение в коробкой. Далее, используя обратной кинематики манипулятора превратится вектор траектории $P(t)$ в вектор обобщенных координат $q(t)$, который нужен для управления манипулятором. Далее через системы управления манипулятором и схватом получим нужные команды для контролера манипулятора чтобы выполнить задачи упаковкой.

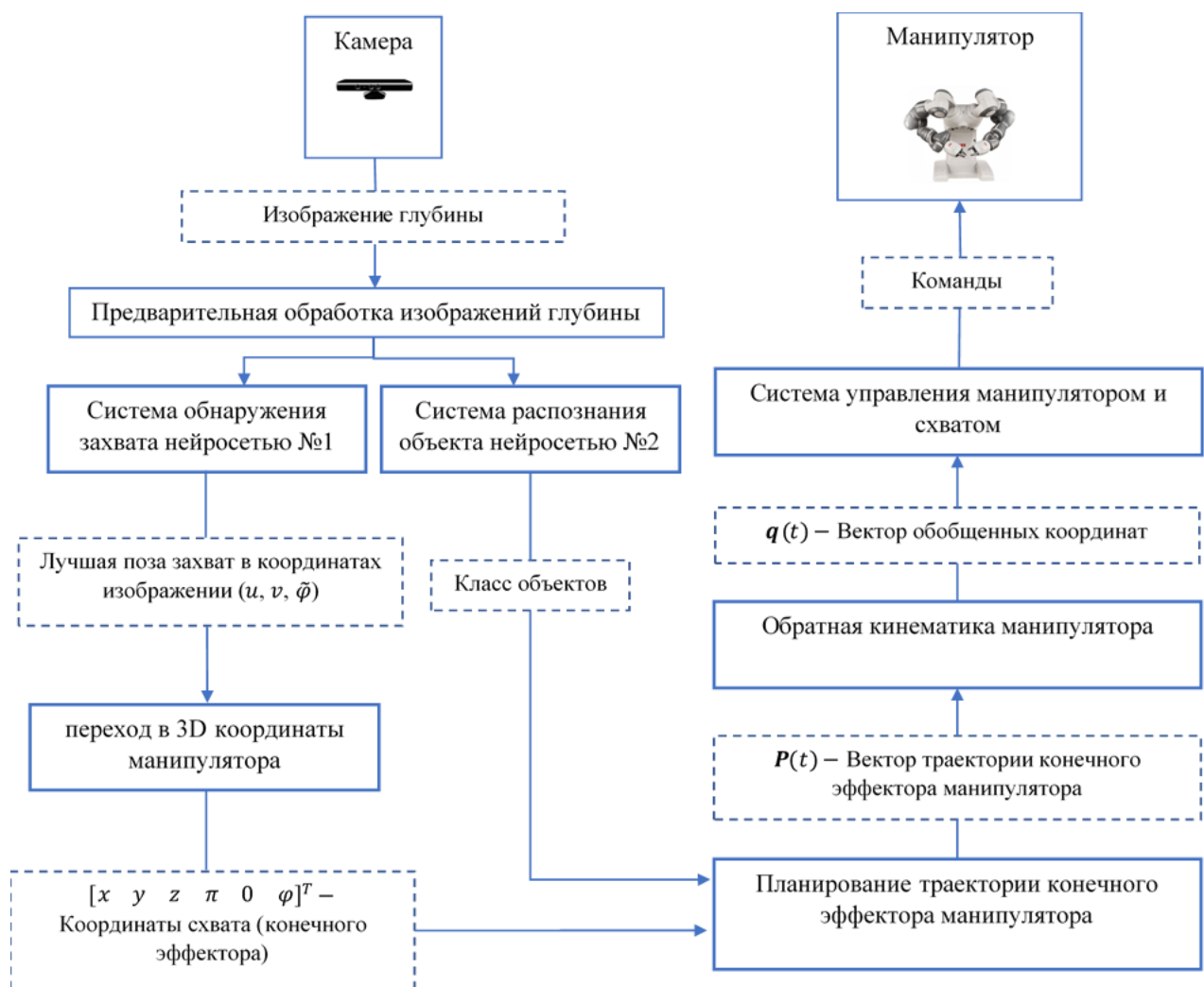


Рисунок 11 — Функциональная схема системы управления манипулятором для выполнения задачи захвата

В следующих разделах будет подробное описание каждой системы схемы.

4 Система обнаружения захвата

Конволюционная нейронная сеть была использована для обнаружения правильного положения захвата. Для обучения нейросеть используется метод «Transfer learning»[6], при котором, знания нейросети, которая была обучена на одной задаче, переносятся на другую задачу, и в нашем случае знания нейросети которая будет обучена на общей задаче «найти правильного положения захвата общих объектов» переносятся на нашу задачу для нахождения правильного положения захвата набор инструментов.

Используя, метод обучения «Transfer learning» нейросеть обучается определения положения захвата инструментов в два этапа:

1. **Предобучение**, где сеть обучилась определить правильного положения захвата на большом наборе данных с разнообразием классов объектов;
2. **Fine-tuning**, когда предобученная модель нейросеть дообучилась на данных шестых объектов(инструментов) целевой задачи.

Этап, предобучение нейросети является важным и основным и включает: построение нейросети, выбор её структуры, подготовка обучающего набора, и оценить нейросеть после обучения.

На следующих разделах, описываются предварительная обработка изображения глубины, построение нейросети, выбор её структуры, подготовка обучающего набора данных, и экспериментальное исследование и результаты после выполнения двух этапы обучения.

4.1 Предварительная обработка изображения глубины

Дополнительное добавление шума не является необходимым, так как набор данных, снимаемый с камеры, уже содержит реальные шумы датчика. Сначала, изображения глубины заполняются с помощью OpenCV чтобы удалить недопустимые значения и заполнить пробелы. После этого 3-D изображения нормализуются (вычитается среднее значение от изображения глубины), чтобы обеспечить инвариантность по глубине (Рисунок 12).

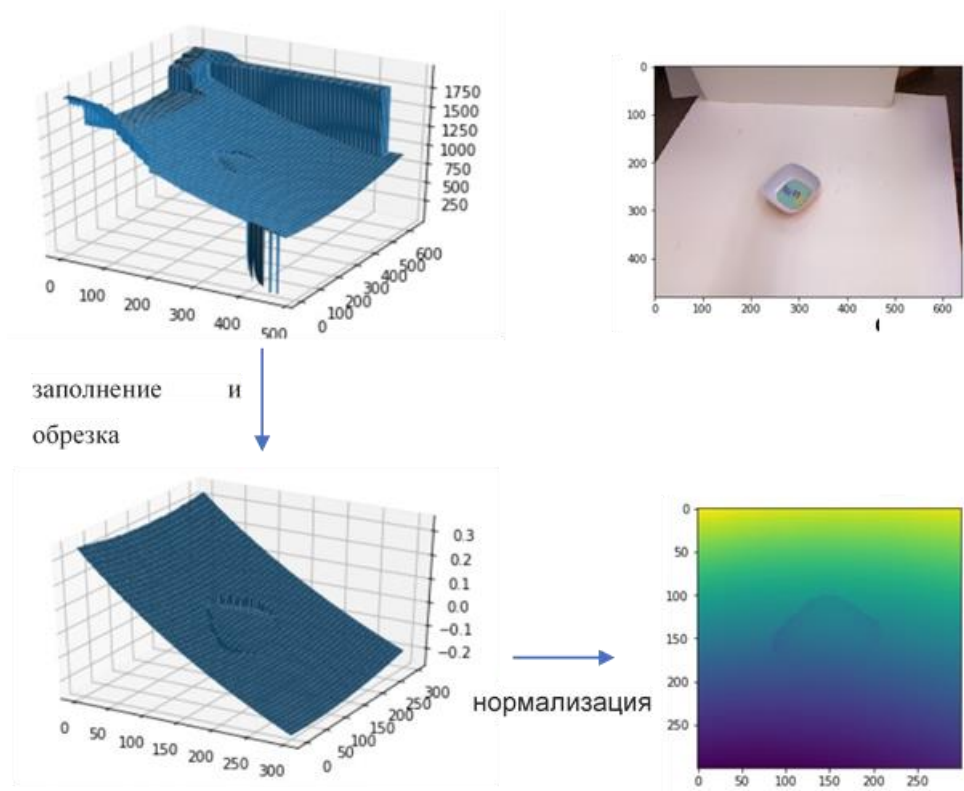


Рисунок 12 — Обработка изображения глубины.

4.2 Построение нейросети и её структуры

Нейросеть построится таким образом, что на входе нейросети поступает изображение глубины, а на выходе, три изображения, показывающие качество захвата, угол ориентации захвата и изображение, обозначающее ширину захвата (Рисунок 13). Качество захвата определяется интенсивностью красного цвета. Предпочтительное место захвата определяется по наивысшему значению интенсивности, имеющему 256 градаций.

Такая типа выходные данные(изображение) был выбран, потому что содержит качества захвата и нужную ориентацию для всех пикселах в изображении что позволяет выбрать второй локальный максимум в изображении качества если при выборе первого максимума захват не удалось.

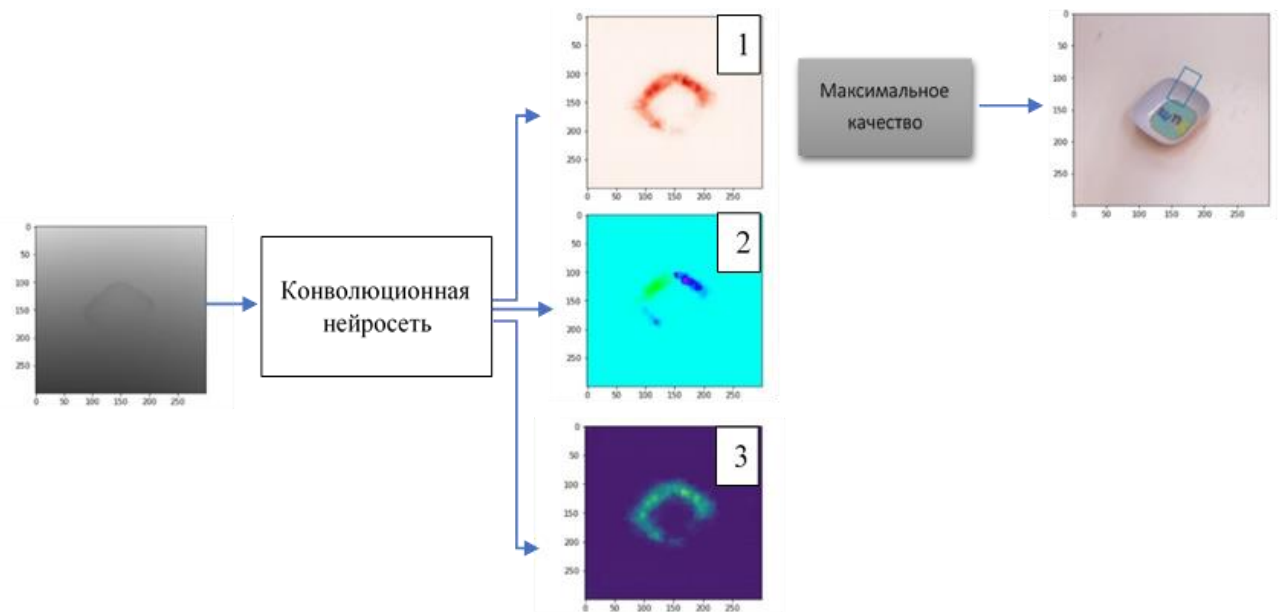


Рисунок 13 — схема системы обнаружения захвата. 1) изображение, обозначающее качество захвата, 2) изображение угла ориентации захвата, 3) изображение ширины захвата.

Тип слоев нейросети был выбран как свёрточных слой (конволюционный слой), поскольку он доказал высокую эффективность и способность к обучению, когда нейронные сети обучаются на изображениях.

Структура нейросети была основана на предложении [21], где нейросеть состоит из шести слоев, три конволюционных и три транспонированных по отношению к конволюционным. (Рисунок. 14).

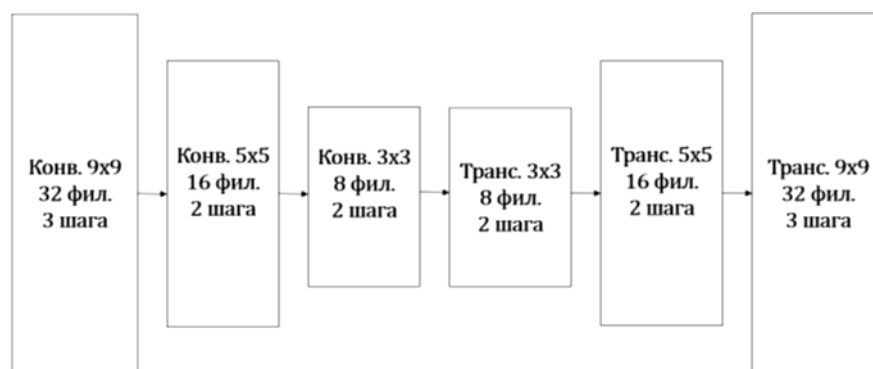


Рисунок 14 — Структура нейросеть.

4.3 Подготовка обучающего набора данных

Существует много наборов данных для обучения нейронных сетей захвата, но наиболее распространенным является набор данных захвата Cornell (Ри-

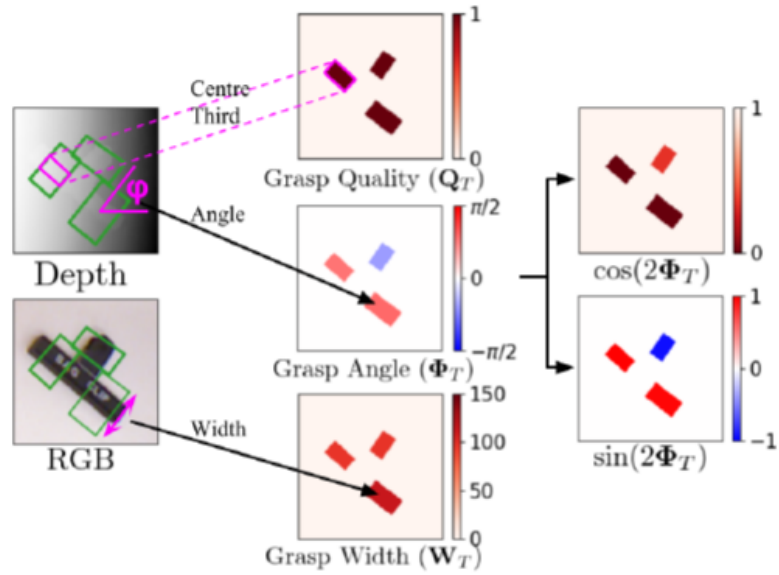


Рисунок 16 — Подготовка обучающего набора.

Из базы данных были взяты RGB изображение, 3-D изображение (depth image). Способы захвата, отмеченные прямоугольниками для данного объекта, показаны в левой части на рисунке 16.

Далее вычисляются параметры, определяющие качество захвата, угол ориентации захвата и его ширина (средняя часть Рисунок 16).

Качество захвата (Grasp Quality): Q^T

Для определения качества захвата Q^T (Grasp Quality) всем пикселям, соответствующим области пересечения прямоугольников захвата с объектом, присваивается значение 1, а остальным пикселям присваивается значение 0.

Угол захвата (Grasp angle): $\tilde{\phi}^T$

Угол ориентации захвата $\tilde{\phi}^T$ (Grasp Angle), определяется как угол поворота прямоугольника захвата относительно горизонтальной оси и принимает значения в диапазоне $\left[-\frac{\pi}{2}, +\frac{\pi}{2}\right]$. Для устранения неоднозначности, которая возникает, когда угол изменяется на $\pm\frac{\pi}{2}$, каждый угол определяется двумя значениями: $\sin(2\tilde{\phi}^T)$, $\cos(2\tilde{\phi}^T)$ (см. правую часть Рисунок. 16).

Ширина захвата (Grasp Width): W^T

При вычислении ширины захвата W^T (Grasp Width) ширине присваивается значение длины перпендикуляра к стороне соответствующего прямоуголь-

ника. Поскольку положение каждого объекта заранее неизвестно, то при обучении нейросети каждое изображение было повернуто на десять разных углов. Таким образом, обучающая выборка увеличилась в десять раз по сравнению с [18].

4.4 Экспериментальное исследование и результаты

Сеть была обучена на 8280 изображений глубины для 224 объектов (80% от набор данных), и после проверки сеть достигла 81,04% успеха захвата на наборе новых объектов (20% от набор данных).

На рисунке 17 показаны примеры успешных захватов. Для каждого из объектов на первом изображении зелеными прямоугольниками показаны возможные захваты, взятые из базы данных cornell. Красным прямоугольником отмечен захват, полученный нейросетью. Он также показан также на втором изображении, которое иллюстрирует изображение глубины, поступающее в нейросеть. Отдельно определяются качество захвата и угол его ориентации (третье и четвертое изображения). Качество захвата определяется интенсивностью красного цвета. Лучшее место захвата определяется по наиболее высокому значению интенсивности, имеющему 256 градаций. Таким образом, можно получить другое возможное место захвата, соответствующее второму локальному максимуму в изображении качества. в изображении качества.

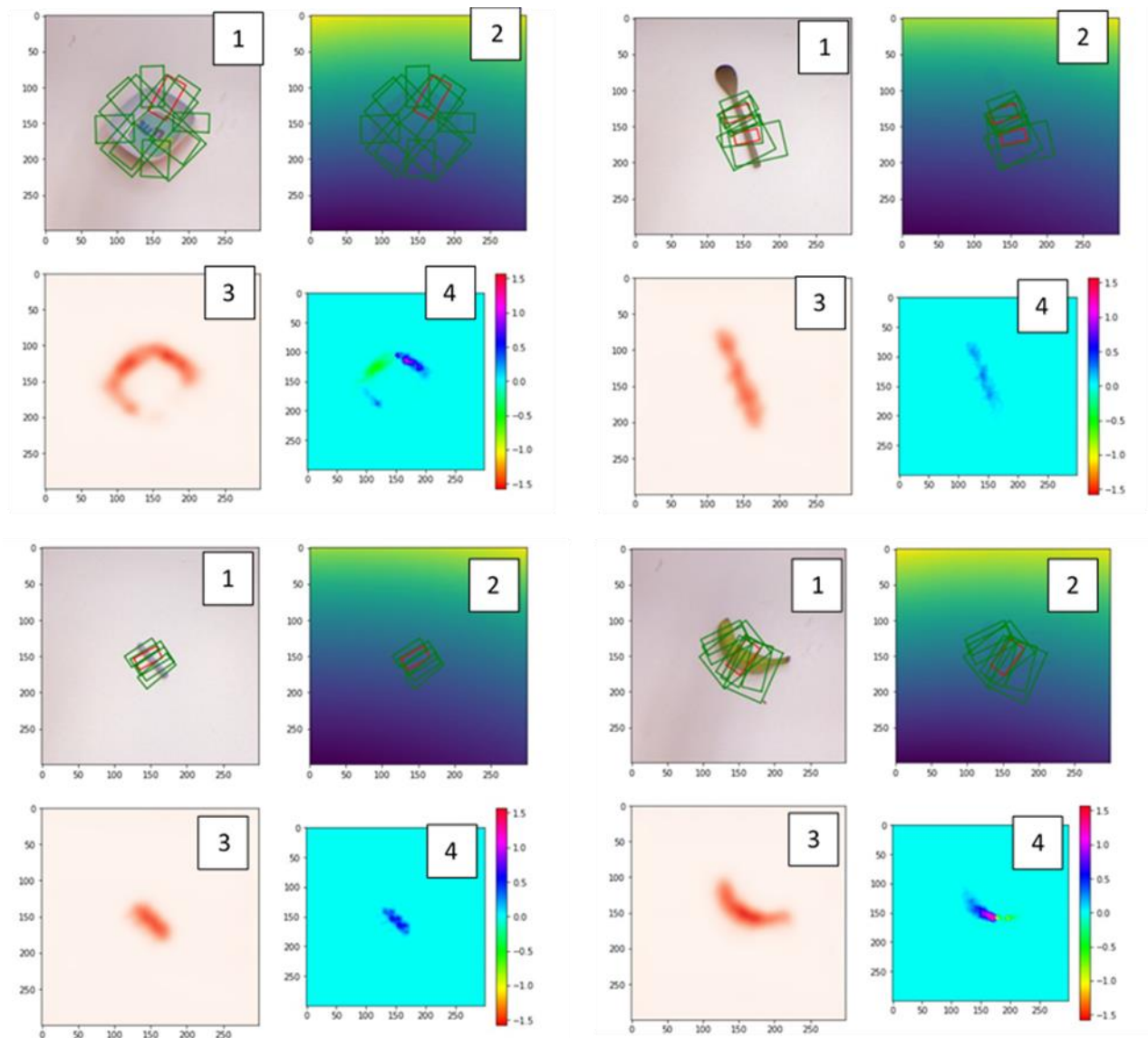


Рисунок 17 — Успешные решения захвата объектов. Для каждого объекта, показано 4 изображения: (1)- изображение объекта с зелёными прямоугольниками для возможных захватов и с красным прямоугольником для захвата, полученного нейросетью; (2) - изображение глубины объект.

Примеры неудачных захватов приведены на рисунке 18. Здесь приняты те же обозначения, что и на рисунке 17. Можно предположить, что неудачные решения нейросети обусловлены как сложной конфигурацией самих объектов (левое изображение), так и их неудачной ориентацией по отношению к оптической системе наблюдения (правое изображение).

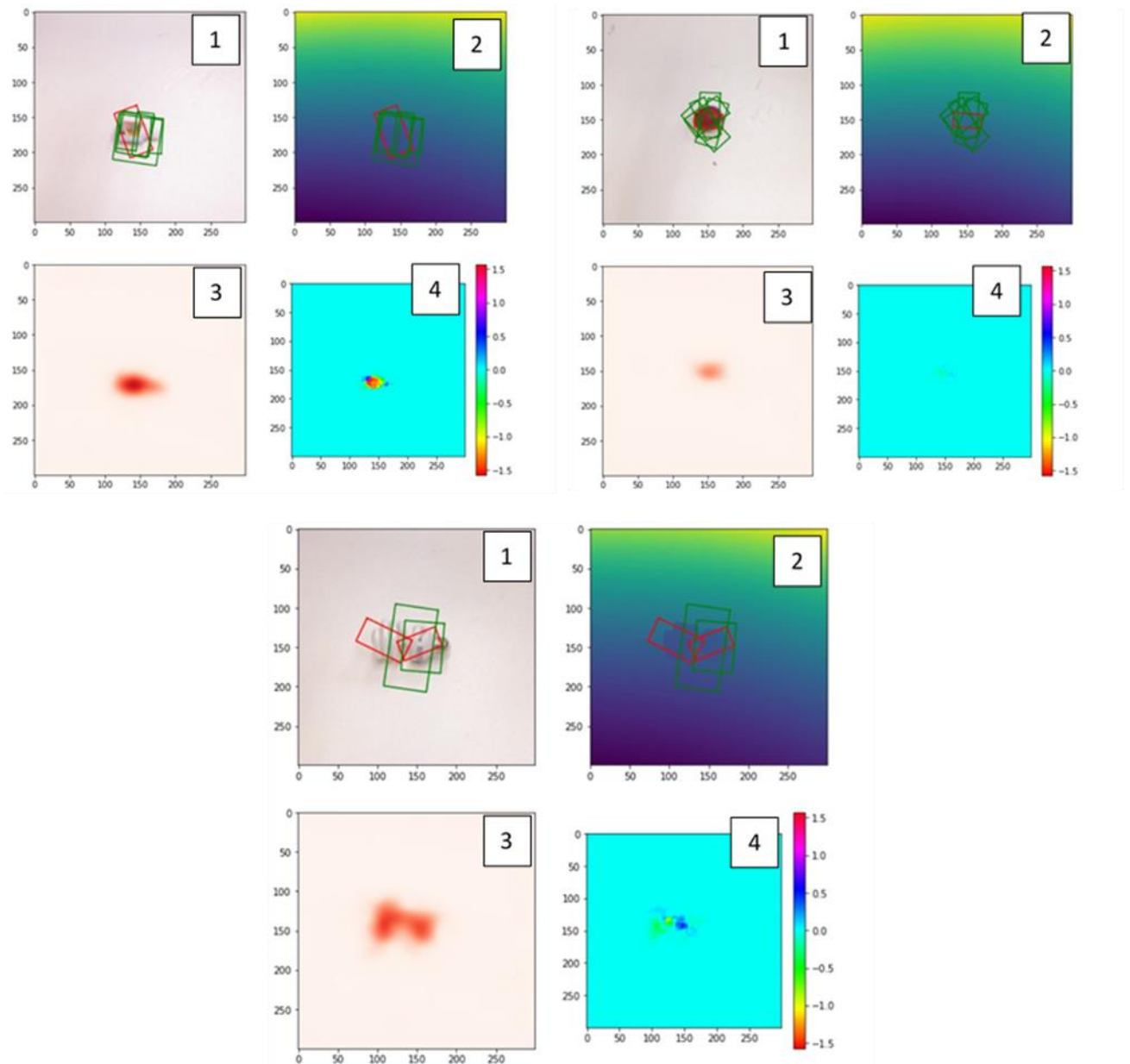


Рисунок 18 — Примеры неудачных решений захвата объектов, полученных сетью.

По результатам проведенного исследования можно заключить, что построенная нейросеть достаточно хорошо обучалась правильным положениям захвата. В меньшей мере процесс обучения был удачным по отношению к углам ориентации и к ширине захвата. Трудности возникают и в особом случае, когда угол равен $+\frac{\pi}{2}$ или $-\frac{\pi}{2}$ и возникает ошибка в определении знака угла. Процесс обучения был неудачным по отношению к ширине, когда полученная нейросетью ширина была близкой к ширине объекта, но немного меньше. Отмеченные неправильные ширины можно объяснить особыми требованиями по

давлению, которые нужны, чтобы удерживать каждый объект, который нейросеть не смогла обучить.

Для решения проблемы неточности ширины, полученной нейросетью, было добавлено управление шириной захвата с обратной связью по силе реакции.

Второй этап метода обучения Transfer Learning «Fine-tuning»:

На этом этапе происходит дообучение предобученной модель нейросеть на изображениях глубины данных шестых объектов(инструментов) целевой задачи.

После дообучения нейросети на шесть объектов – инструменты, используемые в задаче, сеть достигала 97% успеха захвата. Что является достаточно хорошим для выполнения задачи упаковки набора инструментов.

На рисунке 19 показаны способы захвата инструментов, полученных нейросетью. Для каждого из объектов на первом изображении показаны изображения глубины, на котором красным прямоугольником показан захват, полученный нейросетью. На втором, третьем и четвёртом изображении, показаны качество захвата, угол его ориентации и ширина захвата.

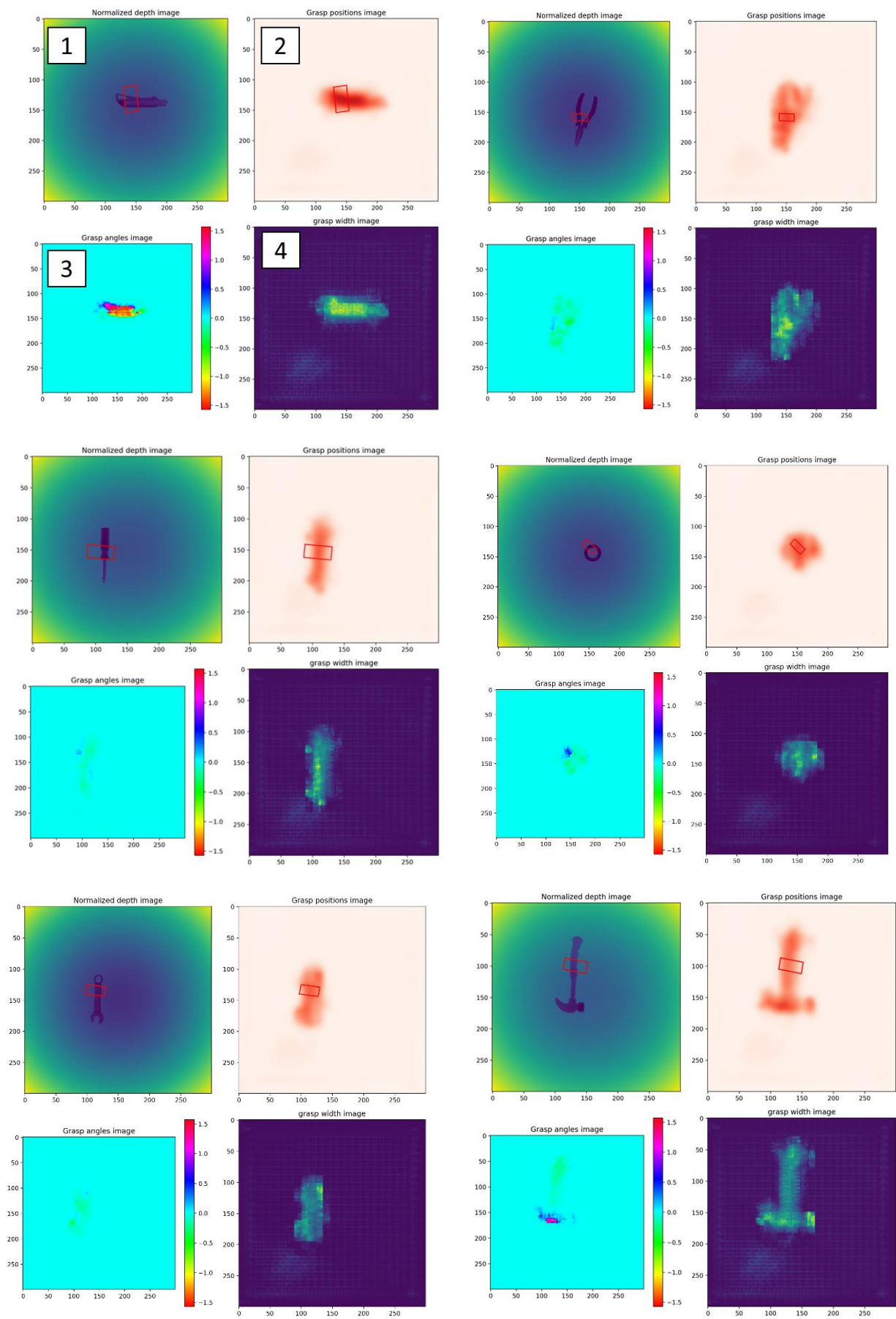


Рисунок 19 — Успешные решения захвата, полученные сетью

5 Система распознавания объекта

Конволюционная нейронная сеть была использована для распознавания объекта. Нейросеть обучилась классификации 6 классов для 6 объектов. Каждый класс обозначается номером с 0 до 5:

- 0 – класс отвёртка
- 1 – класс молоток
- 2 – класс ключ рожковый
- 3 – класс плоскогубцы
- 4 – класс строительный нож
- 5 – класс изоляционная лента

На Рисунок. 20 показано примерные вход и выход системы распознавания объекта для одного из инструментов, где сеть классифицирует объект на класс 4 после аппроксимации выходного сигнала от 4,1.

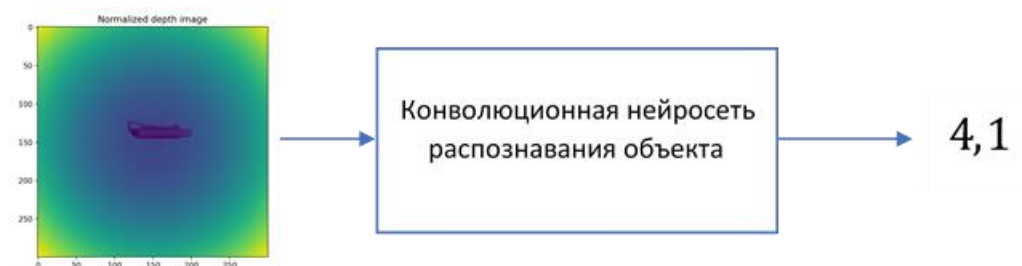


Рисунок 20 — Структура системы распознавания объектов.

Структура нейросети состоит из трёх конволюционных слоев. Нейросеть распознавания объекта достигала 98% успеха классификации после обучения на изображении глубины инструментов с различными углами.

6 Переход в 3D координаты манипулятора

После получения изображения качества, ориентации и ширины захвата, и определение позиции захвата в координатах изображения соответствующей самого высокого качества захвата, можно тогда получить с изображения глубины камеры, 3D координаты захвата $(x_{camera}, y_{camera}, z_{camera})$, но они будут обозначенные в системе координат камеры, поэтому необходимо переходить в 3D координаты манипулятора для формирования траектории движения конечного эффектора манипулятора.

На рисунке 21 показана система координат камеры и мировая система координат манипулятора:

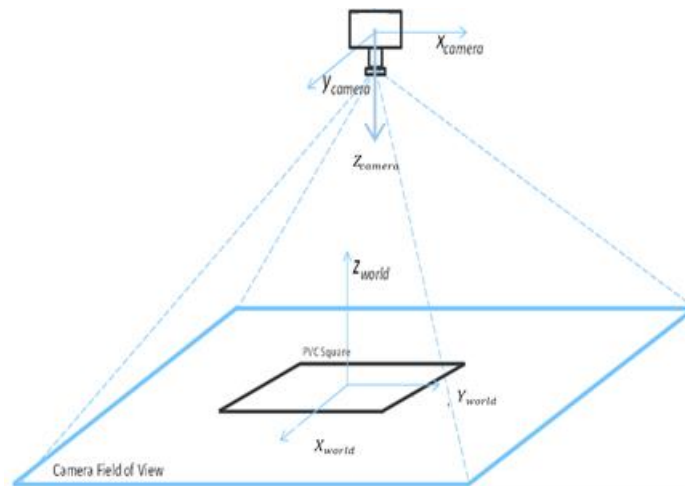


Рисунок 21 — Системы координат камеры и робота.

Опираясь на допущение, что захват всегда будет вертикальным относительно плоскости стола (плоскость x - y), вектор координат конечного эффектора (схват), нужный для выполнения лучшего захвата будет с видом: $[x \ y \ z \ 0 \ \pi \ \varphi]^T$ где x, y, z – координаты центра захватного устройства, и φ – угол поворота захватного устройства вокруг оси z . Будем считать, что $\varphi \approx \tilde{\varphi}$, потому что оси Z в обеих системах являются вертикальными.

Теперь надо найти координаты точки захвата $p_g = [x \ y \ z]$ в системе координат манипулятора. Узнав, позицию и ориентацию камеры в системе координат ма-

нипулятора, можно вернуть захват с 2D-системы координат камеры в 3D-систему координат робота, используя следующую формулу:

$$p_g = T_{WC}(\tilde{p}_g) \quad (1)$$

где T_{WC} – матрица перехода с системы координат камеры в систему координат робота, которая определяется следующим:

$$T_{WC} = \begin{pmatrix} R_{WC} & d_{WC} \\ 0 & 1 \end{pmatrix} \quad (2)$$

где R_{WC} – матрица поворота системы координат и d_{WC} – вектор перемещения центра системы координат.

Для перехода из системы координат камеры в мировую систему координат, надо повернуть на 180 градусов вокруг оси X, затем на -90 градусов вокруг оси Z (Рисунок 21).

Определение матриц перехода:

$$R_{180^\circ, X} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(\pi) & -\sin(\pi) \\ 0 & \sin(\pi) & \cos(\pi) \end{pmatrix} \quad (3)$$

$$R_{-90^\circ, Z} = \begin{pmatrix} \cos(-\frac{\pi}{2}) & -\sin(-\frac{\pi}{2}) & 0 \\ \sin(-\frac{\pi}{2}) & \cos(-\frac{\pi}{2}) & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (4)$$

$$R_{WC} = R_{180^\circ, X} \cdot R_{-90^\circ, Z} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & -1 \end{bmatrix} \quad (5)$$

Перемещение центра системы координат камеры:

$$d_{WC} = \begin{bmatrix} d_x \\ d_y \\ d_z \end{bmatrix} = \begin{bmatrix} -0,2 \\ 0 \\ -0,7 \end{bmatrix} \quad (6)$$

где d_x – перемещение по оси X,

d_y – перемещение по оси Y,

d_z – перемещение по оси Z.

Получим следующая матрица перехода:

$$T_{WC} = \begin{bmatrix} 0 & 1 & 0 & -0,2 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & -0,7 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (7)$$

7 Планирование траектории концевой эффектора манипулятора

Для планирования траектории концевой эффектора манипулятора, сначала составляется план действий (задач), а затем формируется его траектория.

Планирование действий:

После каждого шага конвейера, один объект (инструмент) будет в поле видимости камеры. После определения способа захвата объекта (нужная позиция и ориентация концевой эффектора), будет применена серия действий, состоящая из восьми состояний (Рисунок 22). Первое состояние «Move to Grasp Position & Open Gripper» заключается в том, что концевой эффектор достигает нужных позиций и ориентации для захвата и открывает схват. Второе – «Close Gripper», при котором схват закрывается по алгоритму, описанному ранее. При третьем состоянии «Move to Grasp Departure Position» концевой эффектор перейдет к положению отправления. Четвертая позиция «Move to Place Approach Position» заключается в переходе концевой эффектора к определённому положению инструмента над коробкой. При пятой – «Move to Place Position» концевой эффектор переходит к коробке, чтобы уложить объект (инструмент). Шестое состояние «Open Gripper» – укладывает объект, открывая схват. Седьмая – «Move to Place Departure Position» при которой концевой эффектор переходит к положению отправления от позиции уложения. При восьмой позиции «Move to Grasp Approach Position» концевой эффектор перейдет к положению над объектом чтобы подготовиться к следующему захвату. Таким образом, манипулятор готов к очередному выполнению цикла.

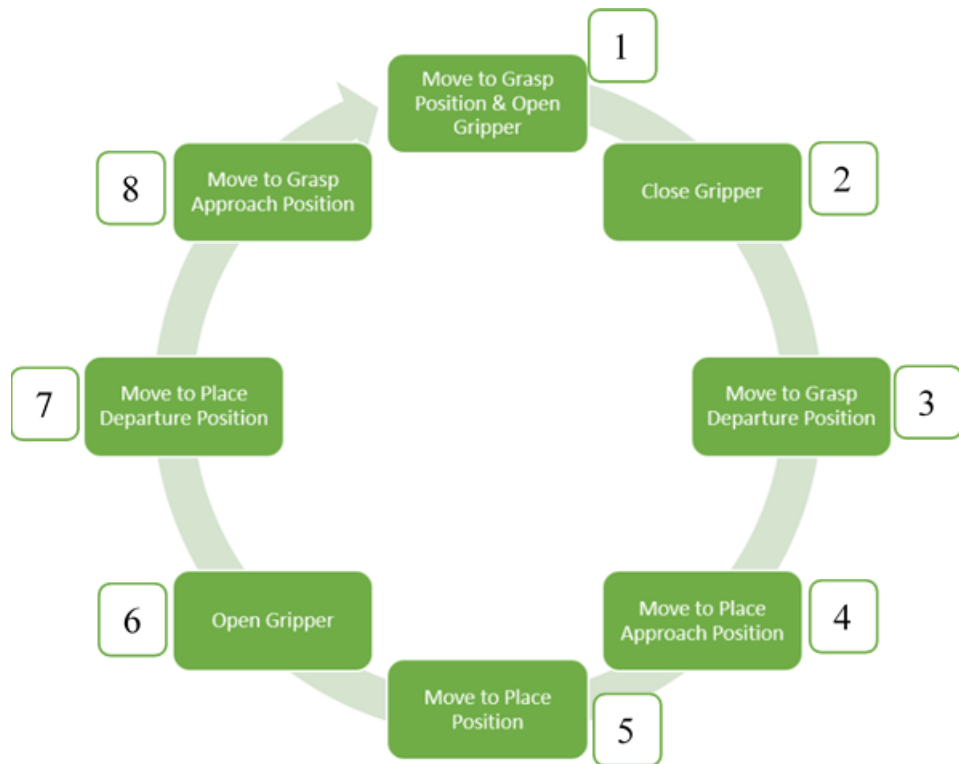


Рисунок 22 — Серия действий для выполнения процесса захвата объекта.

На каждом действии, в том числе во время перехода на вход подается скорость с трапецеидальным профилем. Необходимость использования сигнала такого вида будет объяснено в следующей главе.

Формирование траектории:

Предположим, что искомая траектория движения манипулятора определена в пространстве углов сочленений с помощью обратной кинематики. Невозможно сохранить всю траекторию движения в памяти компьютера, поэтому обычно в памяти хранится последовательность точек $p_i(t_k)$ для каждой совместной переменной i , представляющей искомые значения этой переменной в дискретные моменты времени t_k .

Предположим, что точки равномерно распределены во времени, определим период дискретизации следующим образом:

$$T = t_{k+1} - t_k \tag{8}$$

На каждом промежутке времени $[t_k, t_{k+1}]$, заданное ускорение будет постоянным, так как меньшие постоянные ускорения приводят к насыщению привода. Профиль скорости получается при интегрировании ускорения, а профиль позиции получается при интегрировании скорости. На рисунке 23 (а), показан профиль постоянного ускорения, на рисунке 23 (b) и (с) показаны соответствующие ему профили скорости и позиции (угол сочленения).

Схема управления роботом требует непрерывно заданной траектории движения. Для преобразования таблицы сквозных точек $p_i(t_k)$ в непрерывную искомую траекторию $p_d(t)$ используется кубическая полиномиальная интерполяция [23], которой можно описать искомую траекторию $p_{di}(t)$ для сочленения i , показанную на рисунке 23 (с) в виде:

$$p_{di}(t) = \begin{cases} a_i + (t - t_k)b_i + (t - t_k)^2 c_i, & t_k \leq t < t_k + t_b \\ d_i + v_i t, & t_k + t_b \leq t < t_{k+1} - t_b \\ e_i + (t - t_{k+1})f_i + (t - t_{k+1})^2 g_i, & t_{k+1} - t_b \leq t < t_{k+1} \end{cases} \quad (9)$$

где $a_i, b_i, c_i, d_i, e_i, f_i, g_i$ – постоянные полиномиальной интерполяции, вычисляющиеся из граничных условий, v_i, t_b – параметров проектирования.

Коэффициент v_i можно интерпретировать как максимальную скорость, допустимую для переменной сочленения i , чтобы достигнуть конечное положение захвата.

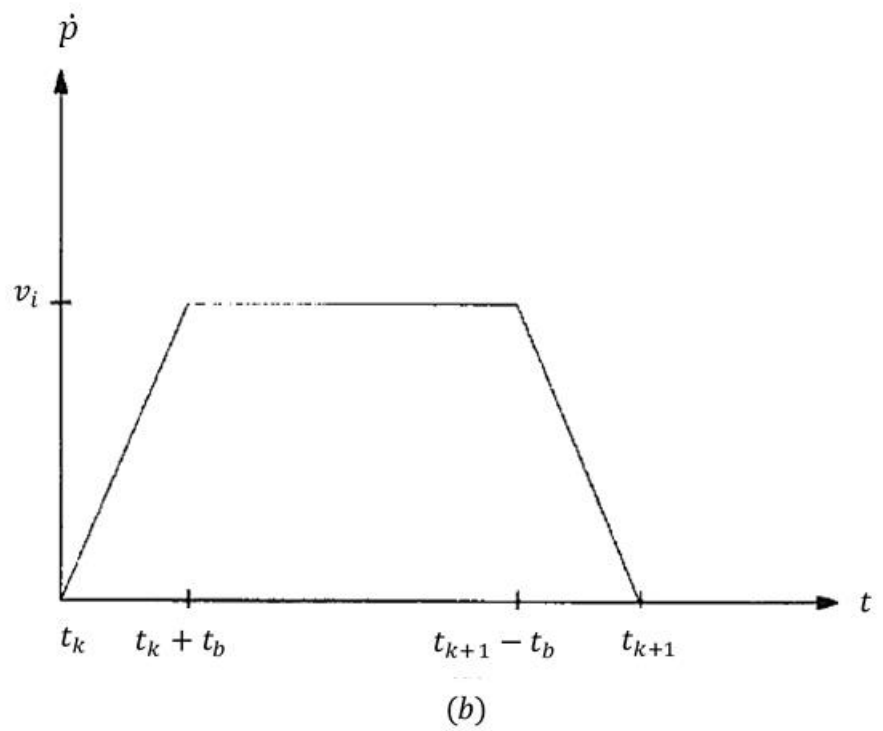
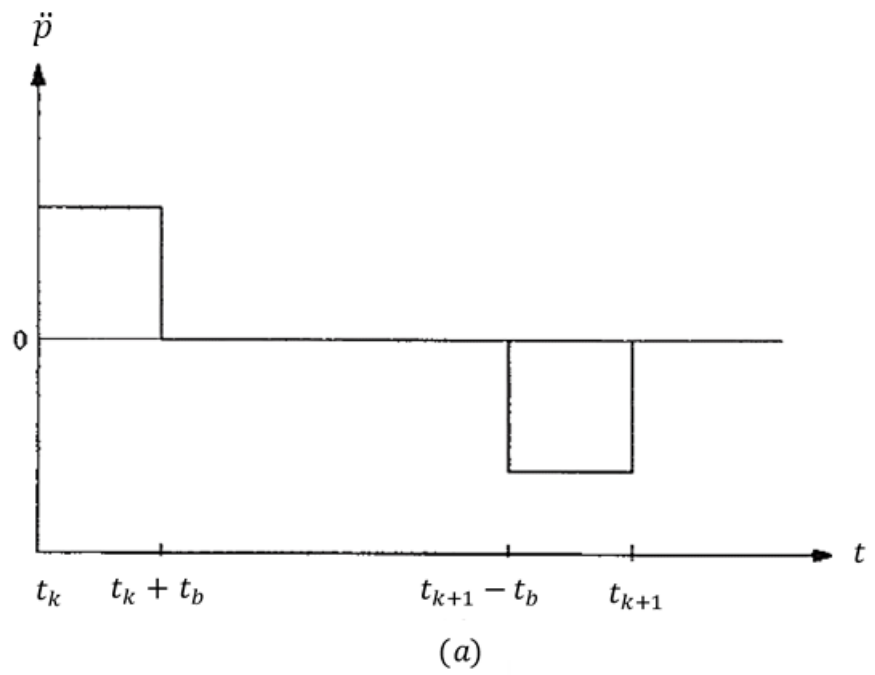


Рисунок 23 — (a) профиль постоянного ускорения, (b) профиль скорости.

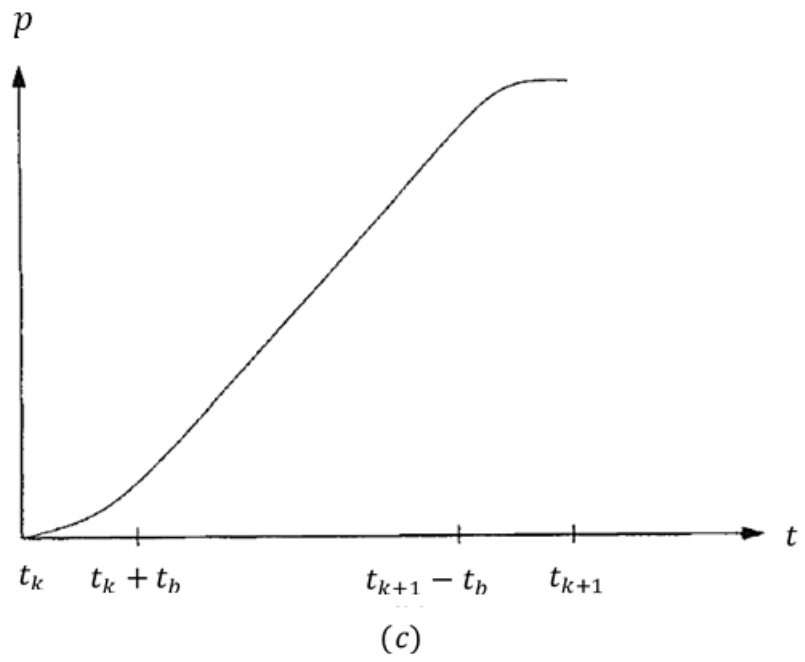


Рисунок 23 — (с) профиль позиции (угол поворота) сочленения.

8 Кинематический и динамический анализ манипулятора

После получения нужную траекторию для выполнения задачи упаковки происходит управления манипулятора для достижения траектории, на этом разделе кинематический и динамический анализ манипулятора были изучены чтобы на следующем этапе построить систему управления манипулятора.

Робот ABB IRB 14000 (YuMi) имеет две руки, каждая из которых имеет семь степеней свободы. Когда робот-манипулятор имеет более шести степеней свободы (три степени для перемещения вдоль X, Y, Z и три для вращения вокруг X, Y, Z), он считается избыточным.

Избыточные манипуляторы широко используются, так как обладают большей ловкостью и универсальностью, чем не избыточные. В избыточных манипуляторах число степеней подвижности превышает число степеней свободы рабочего эффектора манипулятора, что приводит к возможности генерировать бесконечное число решений.

8.1 Кинематический анализ манипулятора

Кинематическая схема манипуляционного робота представлена на рисунке 24. Из рисунка видно, что этот механизм имеет семь вращательных степеней свободы, как показано на рисунке 25.

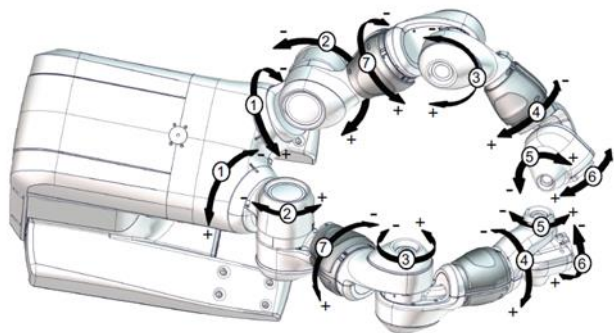


Рисунок 24 — Робот ABB Irb 14000 YuMi.

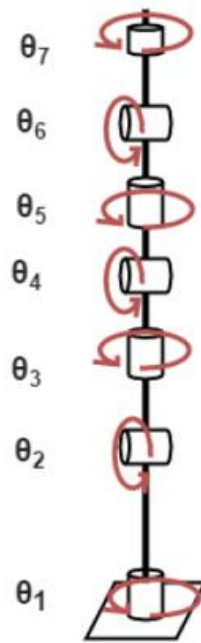


Рисунок 25 — Степеней подвижности робота.

Первые три сочленения, принимаем как виртуальное сферическое сочленение, так как их оси пересекаются в одной точке. Аналогично с последними тремя сочленениями. Между этими двумя виртуальными сферическими сочленениями находится еще одно вращательное.

Схема манипулятора похожа на человеческую руку, где (1), (2) и (3) – это плечевые сочленения (shoulder); (4) – локтевое сочленение (elbow); где (5), (6) и (7) – лучезапястное сочленение (wrist). (Рисунок 26).

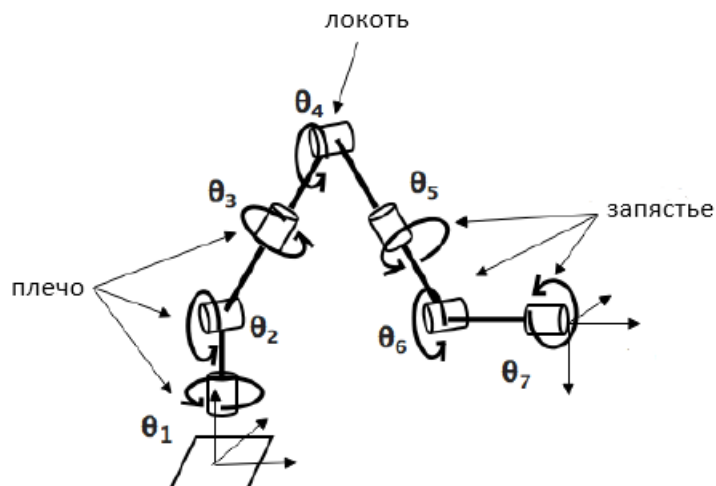


Рисунок 26 — Сочленения робота.

Прямая кинематика:

По прямой задаче вычисляем положения и ориентацию рабочего эффектора манипулятора, а также заданные углы поворота его звеньев.

Вывод уравнений задачи прямой кинематики не представляет собой сложной задачи, в отличии вывода уравнений обратной кинематики. Следовательно, всегда есть решение прямой кинематической задачи. Метод Денавита-Хартенберга [22] является наиболее распространенным методом описания кинематики манипулятора, который приводит к уравнениям прямой кинематики.

По этому методу, матрица преобразования определяется:

$${}^0T_7 = {}^0T_1 {}^1T_2 {}^2T_3 {}^3T_4 {}^4T_5 {}^5T_6 {}^6T_7 \quad (10)$$

где ${}^{i-1}T_i$ матрица преобразования с звена $i - 1$ до звена i , которая может быть разбита на две подматрицы, ${}^{i-1}R_i$ - матрица поворота и ${}^{i-1}p_i$ - вектор координата.

$${}^{i-1}T_i = \begin{bmatrix} {}^{i-1}R_i & {}^{i-1}p_i \\ 0 & 1 \end{bmatrix} \quad (11)$$

Матрицы поворота вокруг X и Z, (R_x, R_z) определяются:

$$R_x = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\alpha_i) & -\sin(\alpha_i) \\ 0 & \sin(\alpha_i) & \cos(\alpha_i) \end{bmatrix} \quad (12)$$

$$R_z = \begin{bmatrix} \cos(\theta_i) & -\sin(\theta_i) & 0 \\ \sin(\theta_i) & \cos(\theta_i) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (13)$$

Формирование системы координат:

Каждая система координат формируется на основе следующих трех правил:

- 1) ось z_{i-1} направлена вдоль оси i -го сочленения;
- 2) ось x_i перпендикулярна оси z_{i-1} и направлена от нее;
- 3) ось y_i дополняет оси x_i, z_i до правой декартовой системы координат.

Построим системы координат каждого звена по методу Денавита-Хартенберга (Рисунок 27).

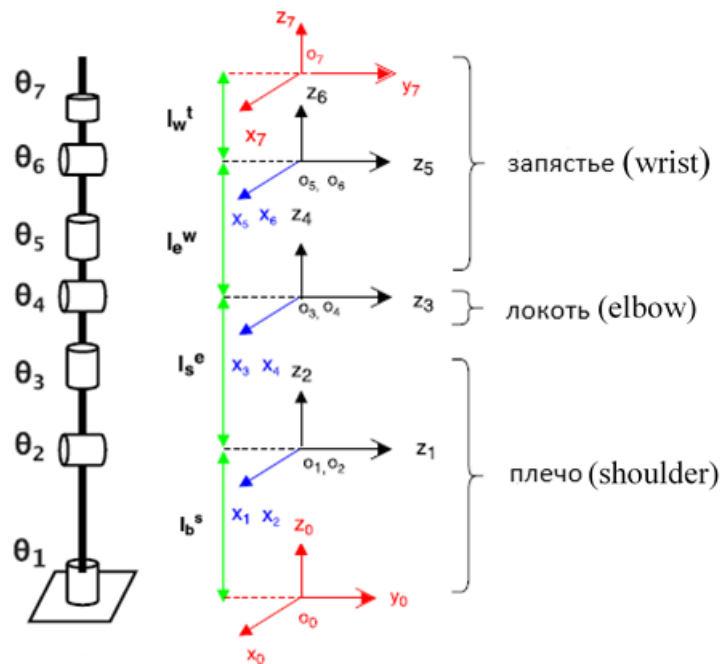


Рисунок 27 — Системы координат.

Уравнения прямой кинематики:

После формирования систем координат звеньев манипулятора, определим параметры звеньев и сочленений для каждого i ($i = 1 \dots 7$) (таблица. 1).

Таблица 1 – параметры звеньев и сочленений

Joint	Link angle (rad)	Link twist α_i (rad)	Link length a_i (m)	Link offset d_i (m)
1	θ_1	$-\frac{\pi}{2}$	0	$b l_s$
2	θ_2	$\frac{\pi}{2}$	0	0

3	θ_3	$-\frac{\pi}{2}$	0	${}^s l_e$
4	θ_4	$\frac{\pi}{2}$	0	0
5	θ_5	$-\frac{\pi}{2}$	0	${}^e l_w$
6	θ_6	$\frac{\pi}{2}$	0	0
7	θ_7	0	0	${}^w l_t$

где ${}^b l_s$ это длина от основания до плеча, ${}^s l_e$ длина от плеча до локтя, ${}^e l_w$ это длина от локтя до запястья, ${}^w l_t$ это длина от запястья до рабочего органа (схвата) (Рисунок 26).

Вычисляем матрицу преобразования:

$${}^0 T_1 = \begin{bmatrix} C_1 & 0 & -S_1 & 0 \\ S_1 & 0 & C_1 & 0 \\ 0 & -1 & 0 & {}^b l_s \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (14)$$

$${}^1 T_2 = \begin{bmatrix} C_2 & 0 & S_2 & 0 \\ S_2 & 0 & -C_2 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (15)$$

$${}^2 T_3 = \begin{bmatrix} C_3 & 0 & -S_3 & 0 \\ S_3 & 0 & C_3 & 0 \\ 0 & -1 & 0 & {}^s l_e \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (16)$$

$${}^3 T_4 = \begin{bmatrix} C_4 & 0 & S_4 & 0 \\ S_4 & 0 & -C_4 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (17)$$

$${}^4T_5 = \begin{bmatrix} C_5 & 0 & -S_5 & 0 \\ S_5 & 0 & C_5 & 0 \\ 0 & -1 & 0 & e l_w \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (18)$$

$${}^5T_6 = \begin{bmatrix} C_6 & 0 & S_6 & 0 \\ S_6 & 0 & -C_6 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (19)$$

$${}^6T_7 = \begin{bmatrix} C_7 & -S_7 & 0 & 0 \\ S_7 & C_7 & 0 & 0 \\ 0 & 0 & 1 & w l_t \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (20)$$

$${}^0T_7 = {}^0T_1 {}^1T_2 {}^2T_3 {}^3T_4 {}^4T_5 {}^5T_6 {}^6T_7 = \begin{bmatrix} R_{11} & R_{12} & R_{13} & p_x \\ R_{21} & R_{22} & R_{23} & p_y \\ R_{31} & R_{32} & R_{33} & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (21)$$

$$\begin{aligned} R_{11} = & C_6 C_7 (C_4 C_5 (C_1 C_2 C_3 - S_1 S_3 - C_1 S_2) - C_1 S_2 S_4 C_5 \\ & - S_5 (C_1 C_2 S_3 + S_1 C_3)) \\ & + S_6 C_7 (S_4 (C_1 C_2 C_3 - S_1 S_3 - C_1 S_2) - C_1 S_2 C_4) \\ & + S_7 (-C_4 S_5 (C_1 C_2 C_3 - S_1 S_3 - C_1 S_2) + C_1 S_2 S_4 S_5 \\ & - C_5 (C_1 C_2 S_3 + S_1 C_3)) \end{aligned} \quad (22)$$

$$\begin{aligned} R_{12} = & C_6 S_7 (C_1 S_2 S_4 C_5 \\ & + S_5 (C_1 C_2 S_3 + S_1 C_3) - C_4 C_5 (C_1 C_2 C_3 - S_1 S_3 \\ & - C_1 S_2)) \\ & - S_6 S_7 (S_4 (C_1 C_2 C_3 - S_1 S_3 - C_1 S_2) - C_1 S_2 C_4) \\ & + C_7 (-C_4 S_5 (C_1 C_2 C_3 - S_1 S_3 - C_1 S_2) + C_1 S_2 S_4 S_5 \\ & - C_5 (C_1 C_2 S_3 + S_1 C_3)) \end{aligned} \quad (23)$$

$$\begin{aligned}
R_{13} = & S_6(C_4C_5(C_1C_2C_3 - S_1S_3 - C_1S_2) - C_1S_2S_4C_5 \\
& - S_5(C_1C_2S_3 + S_1C_3)) \\
& - C_6(S_4(C_1C_2C_3 - S_1S_3 - C_1S_2) - C_1S_2C_4)
\end{aligned} \tag{24}$$

$$\begin{aligned}
R_{21} = & C_6C_7(C_4C_5(S_1C_2C_3 + C_1S_3) - C_5S_1S_2S_4 \\
& + S_5(C_1C_3 - S_1S_3C_2)) \\
& + S_6C_7(S_4(S_1C_2C_3 + C_1S_3) - S_1S_2C_4) \\
& + S_7(C_4S_5(S_1C_2C_3 + C_1S_3) + S_1S_2S_4S_5 \\
& + C_5(C_1C_3 - S_1S_3C_2))
\end{aligned} \tag{25}$$

$$\begin{aligned}
R_{22} = & -C_6S_7(C_4C_5(S_1C_2C_3 + C_1S_3) - C_5S_1S_2S_4 \\
& + S_5(C_1C_3 - S_1S_3C_2)) - S_6S_7(S_4(S_1C_2C_3 + C_1S_3) \\
& - S_1S_2C_4) \\
& + C_7(C_4S_5(S_1C_2C_3 + C_1S_3) + S_1S_2S_4S_5 \\
& + C_5(C_1C_3 - S_1S_3C_2))
\end{aligned} \tag{26}$$

$$\begin{aligned}
R_{23} = & S_6(C_4C_5(S_1C_2C_3 + C_1S_3) - C_5S_1S_2S_4 \\
& + S_5(C_1C_3 - S_1S_3C_2)) \\
& - C_6(S_4(S_1C_2C_3 + C_1S_3) - S_1S_2C_4)
\end{aligned} \tag{27}$$

$$\begin{aligned}
R_{31} = & C_6C_7(S_2S_3S_5 - S_2C_3C_4C_5 - C_2S_4C_5) \\
& + S_6C_7(-S_2C_3S_4 - C_2C_4) \\
& + S_7(S_3C_3C_4S_5 + C_2S_4S_5 + C_5S_2S_3)
\end{aligned} \tag{28}$$

$$\begin{aligned}
R_{32} = & -C_6S_7(S_2S_3S_5 - S_2C_3C_4C_5 - C_2S_4C_5) \\
& + S_6S_7(S_2C_3S_4 + C_2C_4) \\
& + C_7(S_3C_3C_4S_5 + C_2S_4S_5 + C_5S_2S_3)
\end{aligned} \tag{29}$$

$$R_{33} = S_6(S_2S_3S_5 - S_2C_3C_4C_5 - C_2S_4C_5) + C_6(S_2C_3S_4 + C_2C_4) \quad (30)$$

$$\begin{aligned} p_x = & {}^s l_e C_1 S_2 - {}^w l_t (S_4(S_1 S_3 - C_1 C_2 C_3) - C_1 C_4 S_2) \\ & - [{}^w l_t (C_6(S_4(S_1 S_3 - C_1 C_2 C_3) - C_1 C_4 S_2) \\ & + S_6(C_5(C_4(S_1 S_3 - C_1 C_2 C_3) + C_1 S_2 S_4) \\ & + S_5(C_3 S_1 + C_1 C_2 S_3)))] \end{aligned} \quad (31)$$

$$\begin{aligned} p_y = & [{}^w l_t (C_6 (S_4 (C_1 S_3 + C_2 C_3 S_1 + C_4 S_1 S_2) \\ & + S_6 (C_5 (C_4 (C_1 S_3 + C_2 C_3 S_1) - S_1 S_2 S_4) \\ & + S_5 (C_1 C_3 - C_2 S_1 S_3))))] {}^e l_w (S_4 (C_1 S_3 + C_2 C_3 S_1) \\ & + C_4 S_1 S_2) + {}^s l_e S_1 S_2 \end{aligned} \quad (32)$$

$$\begin{aligned} p_z = & {}^b l_s + {}^e l_w (C_2 C_4 - C_3 S_2 S_4) \\ & - [({}^w l_t (S_6 (C_5 (C_2 S_4 + C_3 C_4 S_2) - S_2 S_3 S_5) \\ & - C_6 (C_2 C_4 - C_3 S_2 S_4)) + {}^s l_e C_2] \end{aligned} \quad (33)$$

Обратная (Инверсная) кинематика:

Обратная кинематика — это процесс определения углов в сочленении манипулятора для достижения необходимой ориентации рабочего органа (схвата).

Из-за того, что робот избыточный (имеет 7 степеней свободы), при решении инверсной матрицы переводов появится бесконечное число решений.

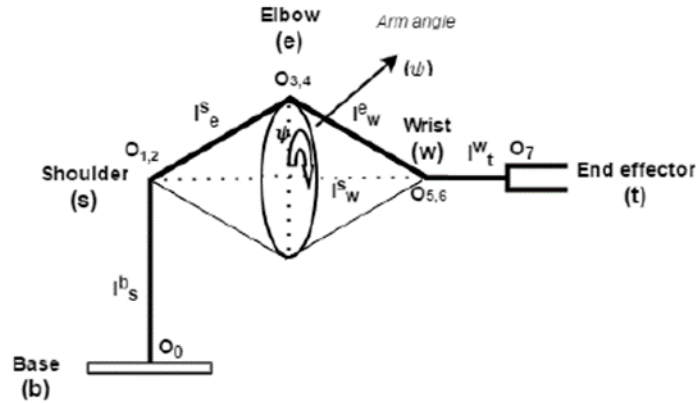


Рисунок 28 — Системы координат.

На рисунке 28 видно, что, когда хват в фиксированном положении, положение запястья также будет фиксировано. Пока хват фиксирован, можно видеть, что положение локтя может свободно вращаться вокруг оси, соединяющей плечо и запястье. Так возникает самопроизвольное движение в этой модели манипулятора, что приводит к бесконечному числу решений.

Параметризация избыточности решается путем наложения дополнительного условия на угол плеча между опорной плоскостью и плоскостью плеча. Полной используемой метод решения можно найти в литературе [23].

После решения обратной кинематики манипулятора, получить 7 углов сочленения $q_i, i = 1, \dots, 7$ соответствующие каждой точке в траектории.

8.2 Динамический анализ манипулятора

Для целей проектирования управления необходимо иметь математическую модель, раскрывающую динамическое поведение системы. Подход к динамической модели состоит в том, чтобы получить кинетическую и потенциальную энергию манипулятора, а затем использовать уравнения движения Лагранжа [24].

$$\frac{d}{dt} \frac{\partial L}{\partial \dot{q}} - \frac{\partial L}{\partial q} = \tau \quad (34)$$

где q – n -мерный вектор переменных сочленений q_i состоящий из углов сочленений θ_i ,

τ – n-мерный вектор обобщенных крутящих моментов,

L – Лагранж – разность кинетической и потенциальной энергий.

В нашем случае у одной руки робота Yumi 7 ABB 7 степеней свободы, n=7.

$$L = K - P \quad (35)$$

Динамика манипулятора находится в стандартном виде [25]:

$$M(q)\ddot{q} + V(q, \dot{q}) + G(q) = \tau \quad (36)$$

где $M(q)$ – матрица инерции,

$V(q, \dot{q})$ – центробежный вектор (Кориолиса),

$G(q)$ вектора силы тяжести.

Кинетическая Энергия манипулятора

Дана точка на звене i с координатами ${}^i r = [x \ y \ z \ 1]^T$ относительно системы координат i прикрепленного к звену. Координатами точки в базовой системе координат являются:

$$r = T_i \cdot {}^i r \quad (37)$$

где T_i - 4×4 матрица однородного преобразования с базовой системы координат в систему координат i -ой звено, которая является функцией от переменных q_1, q_2, \dots, q_i .

Следовательно, скорость точки в базовых координатах равна

$$v = \frac{dr}{dt} = \sum_{j=1}^i \left[\frac{\partial T_i}{\partial q_j} \dot{q}_j \right] {}^i r \quad (38)$$

Поскольку $\frac{\partial T_i}{\partial q_j} = 0, j > i$, мы можем заменить верхний предел суммирования на число звеньев n . Матрицы T_i известны поэтому вычислим матрицы $\frac{\partial T_i}{\partial q_j}$ (4×4).

Кинетическая энергия бесконечно малой массы dm при ${}^i r$, которая имеет скорость $v = [v_x \ v_y \ v_z]^T$ равна:

$$\begin{aligned} dK_i &= \frac{1}{2} (v_x^2 + v_y^2 + v_z^2) dm \\ &= \frac{1}{2} \text{trace}(v^T v) dm = \frac{1}{2} \text{trace}(v v^T) dm \\ &= \frac{1}{2} \text{trace} \left[\sum_{j=1}^n \sum_{k=1}^n \frac{\partial T_i}{\partial q_j} ({}^i r \ {}^i r^T dm) \frac{\partial T_i^T}{\partial q_k} \dot{q}_j \dot{q}_k \right] dm \end{aligned} \quad (39)$$

где trace матрицы – сумма элементов на главной диагонали матрицы.

Таким образом, полная кинетическая энергия для звена i задается формулой:

$$K_i = \int_{link \ i} dK_i \quad (40)$$

Подставляя dK_i из (39), мы можем переместить интеграл внутрь суммирования. Затем, определив псевдо-инерционную матрицу 4×4 для звена i как

$$I_i \equiv \int_{link\ i} {}^i r\ {}^i r^T dm \quad (41)$$

$$I_i = \begin{bmatrix} \int x^2 dm & \int yx dm & \int zx dm & \int x dm \\ \int xy dm & \int y^2 dm & \int zy dm & \int y dm \\ \int xz dm & \int yz dm & \int z^2 dm & \int z dm \\ \int x dm & \int y dm & \int z dm & \int dm \end{bmatrix} \quad (42)$$

где интегралы берутся по объему звена i - Это матрица постоянных, которая вычисляется один раз для каждого звена. Это зависит от геометрии и распределения массы звена i . Моменты инерции для i -го звена:

$$\begin{cases} I_{xx} = \int (y^2 + z^2) dm \\ I_{yy} = \int (x^2 + z^2) dm \\ I_{zz} = \int (x^2 + y^2) dm \end{cases} \quad (43)$$

Центробежные моменты инерции

$$\begin{cases} I_{xy} = \int xy dm \\ I_{xz} = \int xz dm \\ I_{yz} = \int yz dm \end{cases} \quad (44)$$

Первые моменты:

$$\begin{cases} m\bar{x} = \int x dm \\ m\bar{y} = \int y dm \\ m\bar{z} = \int z dm \end{cases} \quad (45)$$

где m общая масса звена i ,

Координаты ${}^i\bar{r}$ в системе координат i центра тяжести звена i мы можем записать как:

$${}^i\bar{r} = [\bar{x} \quad \bar{y} \quad \bar{z} \quad 1]^T \quad (46)$$

$$I_i = \begin{bmatrix} \frac{-I_{xx} + I_{yy} + I_{zz}}{2} & I_{xy} & I_{xz} & m\bar{x} \\ I_{xy} & \frac{I_{xx} - I_{yy} + I_{zz}}{2} & I_{yz} & m\bar{y} \\ I_{xz} & I_{yz} & \frac{I_{xx} + I_{yy} - I_{zz}}{2} & m\bar{z} \\ m\bar{x} & m\bar{y} & m\bar{z} & m \end{bmatrix} \quad (47)$$

Эти значения указаны в таблице в спецификации производителя манипулятора.

Возвращаясь теперь к нашему развитию, общая кинетическая энергия руки может быть записана как

$$K = \sum_{i=1}^n K_i = \frac{1}{2} \sum_{i=1}^n \text{trace} \left[\sum_{j=1}^n \sum_{k=1}^n \frac{\partial T_i}{\partial q_j} I_i \frac{\partial T_i^T}{\partial q_k} \dot{q}_j \dot{q}_k \right] \quad (48)$$

Поскольку след (trace) суммы матриц является суммой отдельных следов, мы можем обмениваться суммированием и оператором трассировки, чтобы получить:

$$K = \frac{1}{2} \sum_{j=1}^n \sum_{k=1}^n m_{jk}(q) \dot{q}_j \dot{q}_k \quad (49)$$

Или

$$K = \frac{1}{2} \dot{q}^T M(q) \dot{q} \quad (50)$$

где $n \times n$ матрица инерции $M(q)$ имеет элементы, определенные как

$$m_{jk}(q) = \sum_{i=1}^n \text{trace} \left[\frac{\partial T_i}{\partial q_j} I_i \frac{\partial T_i^T}{\partial q_k} \right] \quad (51)$$

Потенциальная Энергия манипулятора

Если звено i имеет массу m_i и центр тяжести, ${}^i\bar{r}$ выраженный в системе координат i , то потенциальная энергия звена задается по формуле:

$$P_i = -m_i g^T T_i {}^i\bar{r} \quad (52)$$

где вектор гравитации выражается в базовых координатах как:

$$g = [0 \quad 0 \quad -9.8062 \quad 0]^T \quad (53)$$

Таким образом, полная потенциальная энергия манипулятора равна

$$P = - \sum_{i=1}^n m_i g^T T_i {}^i\bar{r} \quad (54)$$

Заметив, что $m_i {}^i\bar{r}$ это последний столбец звена i псевдо-инерционной матрицы T_i , мы можем записать

$$P(q) = - \sum_{i=1}^n q^T T_i(q) I_i e_4 \quad (55)$$

где e_4 последний столбец матрицы идентичности 4×4 (т. е. $e_4 = [0 \quad 0 \quad 0 \quad 1]^T$). Уравнение Лагранжа. Лагранжиан манипулятора это

$$L(q, \dot{q}) = K(q, \dot{q}) - P(q) = \frac{1}{2} \dot{q}^T M(q) \dot{q} - P(q) \quad (56)$$

Теперь члены, необходимые в уравнении Лагранжа

$$\left\{ \begin{array}{l} \frac{\partial L}{\partial \dot{q}} = \frac{\partial K}{\partial \dot{q}} = M(q) \dot{q} \\ \frac{d}{dt} \frac{\partial L}{\partial \dot{q}} = M(q) \ddot{q} + \dot{M}(q) \dot{q} \\ \frac{\partial L}{\partial q} = \frac{1}{2} \frac{\partial}{\partial q} (\dot{q}^T M(q) \dot{q}) - \frac{\partial P(q)}{\partial q} \end{array} \right. \quad (57)$$

Поэтому динамическое уравнение манипулятора имеет вид:

$$M(q) \ddot{q} + \dot{M}(q) \dot{q} - \frac{1}{2} \frac{\partial}{\partial q} (\dot{q}^T M(q) \dot{q}) + \frac{\partial P(q)}{\partial q} = \tau \quad (58)$$

Определение кориолисова / центростремительного вектора:

$$V(q, \dot{q}) = \dot{M}(q) \dot{q} - \frac{1}{2} \frac{\partial}{\partial q} (\dot{q}^T M(q) \dot{q}) = \dot{M} \dot{q} - \frac{\partial K}{\partial q} \quad (59)$$

Вектор гравитации:

$$G(q) = \frac{\partial P(q)}{\partial q} \quad (60)$$

Можно записать:

$$M(q) \ddot{q} + V(q, \dot{q}) + G(q) = \tau \quad (61)$$

На самом деле рука робота всегда подвержена трению и возмущениям. Поэтому можно обобщить полученную модель руки, записав динамику манипулятора в виде

$$M(q) \ddot{q} + V(q, \dot{q}) + F_v \dot{q} + F_d(\dot{q}) + G(q) + \tau_d = \tau \quad (62)$$

где F_v - матрица коэффициентов вязкого трения,

F_d - динамический член трения.

Также добавляется возмущение τ_d , которое может представлять, например, любую неточно смоделированную динамику.

Можем написать:

$$M(q)\ddot{q} + N(q, \dot{q}) + \tau_d = \tau \quad (63)$$

где

$$N(q, \dot{q}) \equiv V(q, \dot{q}) + F(\dot{q}) + G(q) \quad (64)$$

$q(t) \in R^n$ переменные сочленений,

$\tau(t)$ контрольный крутящий момент,

$\tau_d(t)$ возмущение.

9 Система управления манипулятором и схватом

Управление манипулятором осуществляется крутящим моментом. Управление реализуется в виде прямого управления (Inner feedforward loop) и внешнего контура (PD регулятор) согласно с [25].

Прямое управление (Inner feedforward loop):

Допустим существует желаемая траектория движения $q_d(t)$. Чтобы обеспечить отслеживание траектории переменных каждого звена, выходная ошибка или ошибка отслеживания определяется следующим образом:

$$e(t) = q_d(t) - q(t) \quad (65)$$

Дважды продифференцируем:

$$\begin{aligned} \dot{e} &= \dot{q}_d - \dot{q} \\ \ddot{e} &= \ddot{q}_d - \ddot{q} \end{aligned} \quad (66)$$

Подставив \ddot{q} из (63) в (66):

$$\ddot{e} = \ddot{q}_d + M^{-1}(N + \tau_d - \tau) \quad (67)$$

Определение управляющего воздействия:

$$u = \ddot{q}_d + M^{-1}(N - \tau) \quad (68)$$

Функция возмущения

$$\omega = M^{-1}\tau_d \quad (69)$$

Определим состояние $x(t) \in R^{2n}$

$$x = \begin{bmatrix} e \\ \dot{e} \end{bmatrix} \quad (70)$$

Запишем динамику ошибок слежения:

$$\frac{d}{dt} \begin{bmatrix} e \\ \dot{e} \end{bmatrix} = \begin{bmatrix} 0 & I \\ 0 & 0 \end{bmatrix} \begin{bmatrix} e \\ \dot{e} \end{bmatrix} + \begin{bmatrix} 0 \\ I \end{bmatrix} u + \begin{bmatrix} 0 \\ I \end{bmatrix} \omega \quad (71)$$

Из (68) получено:

$$\tau = M(\ddot{q}_d - u) + N \quad (72)$$

Уравнение (72) является законом управления крутящим моментом.

Подставив (72) в (63):

$$M\ddot{q} + N + \tau_d = M(\ddot{q}_d - u) + N \quad (73)$$

Или

$$\ddot{e} = u + M^{-1}\tau_d \quad (74)$$

Нелинейное преобразование (68) превратило сложную нелинейную задачу управления в простую задачу проектирования линейной системы, состоящей из n разобщенных подсистем, каждая из которых подчиняется законам Ньютона.

Полученная схема управления представлена на рисунке 29.

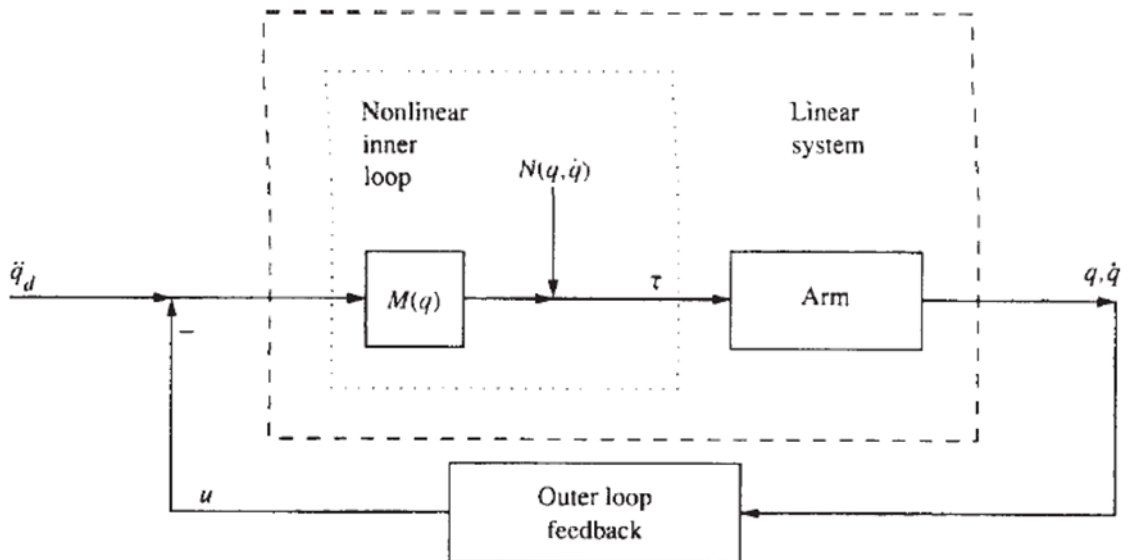


Рисунок 29 — Схема системы управления

Внешний контур представляет собой контур обратной связи. Выберем динамический компенсатор $H(s)$.

$$U(s) = H(s)E(s) \quad (75)$$

Из (74), система ошибок замкнутого контура имеет передаточную функцию:

$$T(s) = s^2I - H(s) \quad (76)$$

где $\tau(t)$ вычисляется путем замены $\ddot{q}_d - u$ на \ddot{q} .

Синтез внешнего контура (PD регулятор):

Одним из способов выбора вспомогательного управляющего сигнала $u(t)$ является пропорционально-дифференцирующий регулятор (PD).

$$u = -K_v \dot{e} - K_p e \quad (77)$$

Тогда общее входное воздействие на манипулятор станет:

$$\tau = M(q)(\ddot{q}_d + K_v \dot{e} + K_p e) + N(q, \dot{q}) \quad (78)$$

Динамика ошибок замкнутого контура:

$$\ddot{e} + K_v \dot{e} + K_p e = \omega \quad (79)$$

Или в форме пространства состояний:

$$\frac{d}{dt} \begin{bmatrix} e \\ \dot{e} \end{bmatrix} = \begin{bmatrix} 0 & I \\ -K_p & -K_v \end{bmatrix} \begin{bmatrix} e \\ \dot{e} \end{bmatrix} + \begin{bmatrix} 0 \\ I \end{bmatrix} \omega \quad (80)$$

Характеристический полином замкнутого контура имеет вид

$$\Delta_c(s) = |s^2I + K_v s + K_p| \quad (81)$$

Выберем постоянные PD. Обычно берется диагональ матриц постоянных $n \times n$ так, что

$$K_v = \text{diag}\{k_{v_i}\}, \quad K_p = \text{diag}\{k_{p_i}\} \quad (82)$$

$$\Delta_c(s) = \prod_{i=1}^n (s^2 + k_{v_i}s + k_{p_i}) \quad (83)$$

Система ошибок асимптотически устойчива, когда все k_{v_i} и k_{p_i} положительны. Стандартная форма характеристического полинома второго порядка имеет вид

$$p(s) = s^2 + 2\xi\omega_n s + \omega_n^2 \quad (84)$$

где ξ коэффициент демпфирования,

ω_n собственная частота для ошибки сочленения i .

Таким образом, желаемая производительность в каждом компоненте ошибки $e(t)$ может быть достигнута путем выбора коэффициентов усиления PD в качестве

$$k_{p_i} = \omega_n^2, \quad k_{v_i} = 2\xi\omega_n \quad (85)$$

Работу нежелательно превышать характеристики, так как желаемая траектория заканчивается на поверхности заготовки. Поэтому коэффициенты усиления PD выбираются для критического демпфирования $\xi = 1$, следовательно

$$k_{v_i} = 2\sqrt{k_{p_i}} \quad , \quad k_{p_i} = \frac{k_{v_i}^2}{4} \quad (86)$$

На рисунке 29 системы управления с регулятором PD.

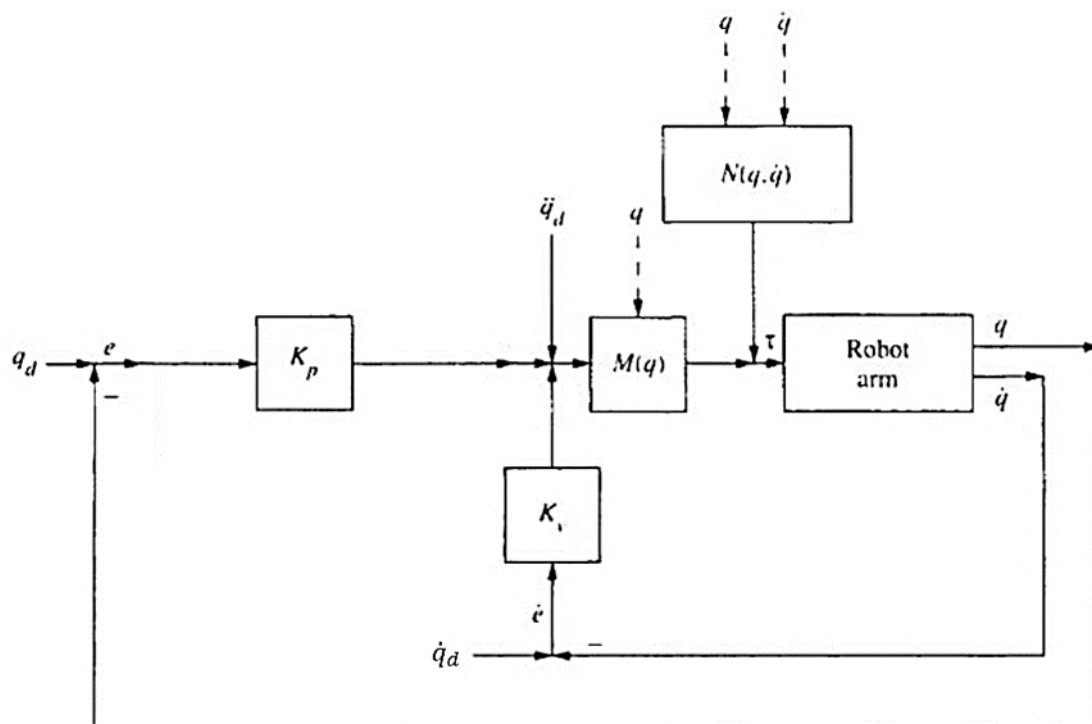


Рисунок 30 — Схема системы управления

Настройка(tuning) параметры системы управления манипулятора:

Настройка параметры системы управления манипулятора происходит с помощью MATLAB (Simulink) из-за некоторых преимуществ, которыми обладает Simulink. В Simulink можно моделировать нелинейные системы, с возможностью легко настроить параметры ПД регулятора. Кроме того, библиотека «Simscape Multibody» в Simulink позволяет моделировать многомодульные механические системы, например, манипулятор. В пакете есть модель сочленения с возможностью параметризации по многим параметрам, позволяющая сделать модель настолько близкой к реальной, насколько это возможно.

Согласно структурной схеме системы управления на рисунке. 30, в MATLAB была построена блок-схема, показанная на рисунке 31.

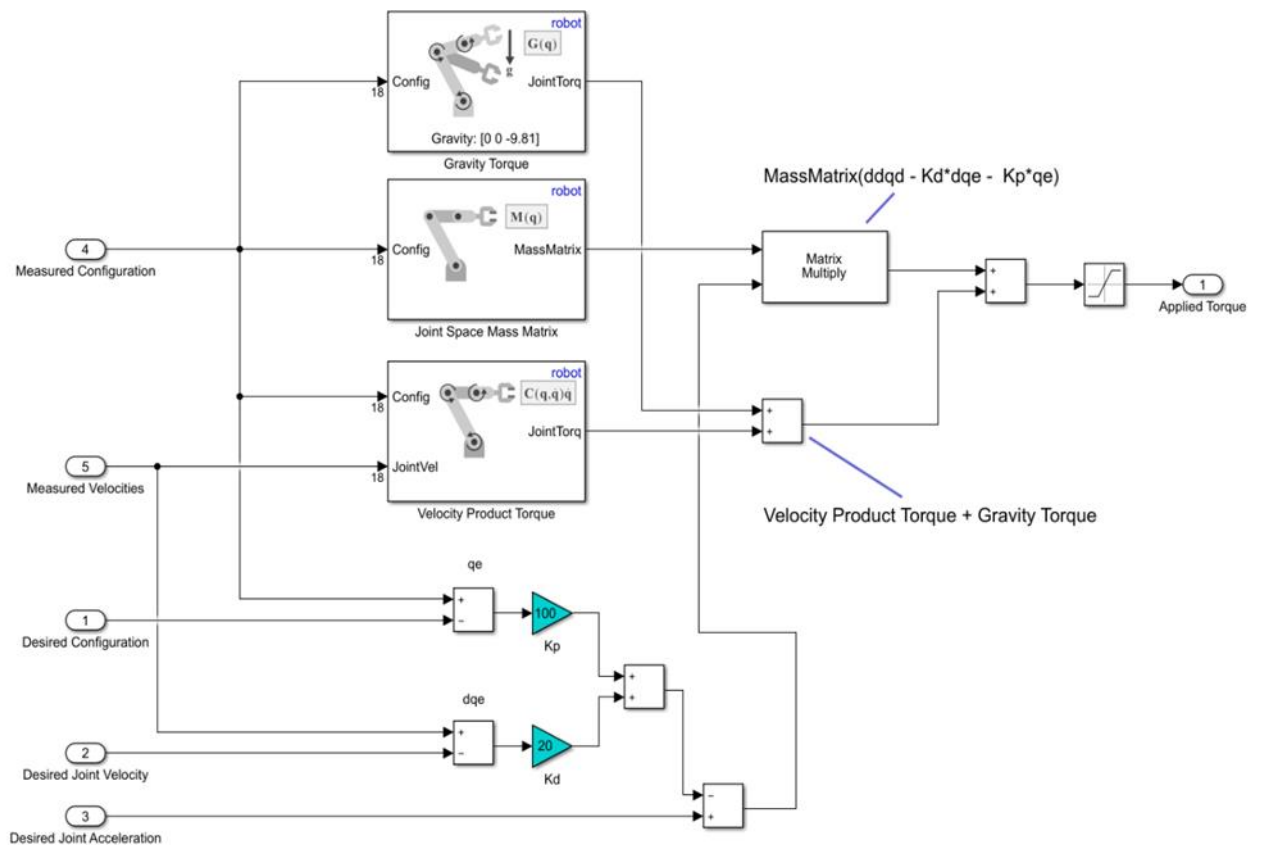


Рисунок 31 — блок-схема системы управления в MATLAB.

После построения модели манипулятора и системы управления в Simulink, были настроены коэффициенты ПД регулятора с помощью MATLAB с соблюдением требований по устойчивости, качеству переходного процесса и точности:

$$k_{v_i} = 20, k_{p_i} = 100$$

Системе крайне нежелательно иметь большое перерегулирование, так как конечной эффектор должен достигать позиции захвата без столкновений с поверхностью стола или с другими частями объекта. Важно, чтобы точность по положению была максимальной настолько, насколько это возможно. Из тех же соображений скорость конечного эффектора должна быть нулевой при достижении позиции захвата.

Результаты:

При подаче на вход постоянных необходимых углов последних четырех сочленений (под номером 3, 4, 5, 6) ошибка по положению не должно быть более 0.00015 рад, что является допустимым (Рисунок. 32). При подаче на вход

желаемой нулевой скорости, ошибка по скорости быстро пошла к нулю (Рисунок. 33).

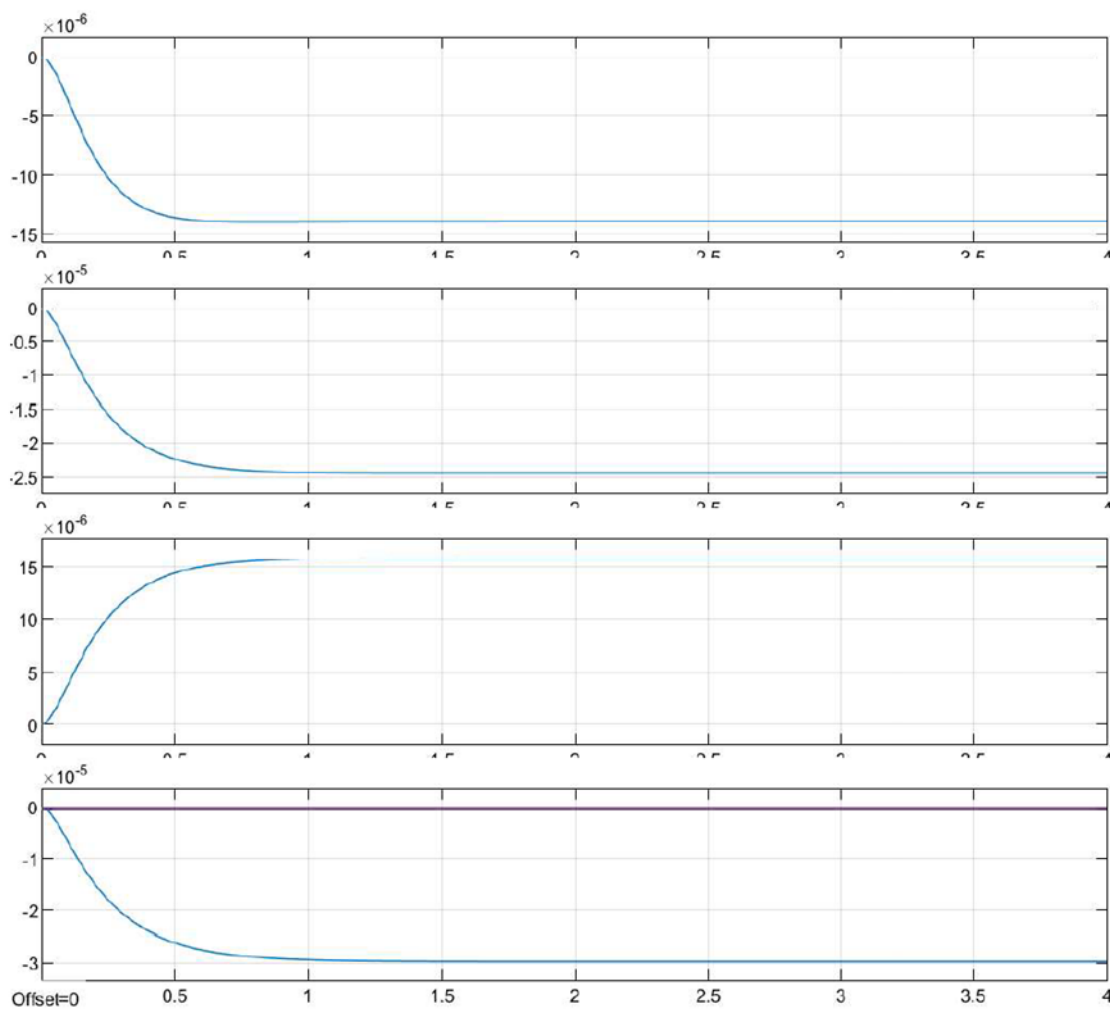


Рисунок 32 — Переходный процесс по положению сочленений 3,4,5,6.

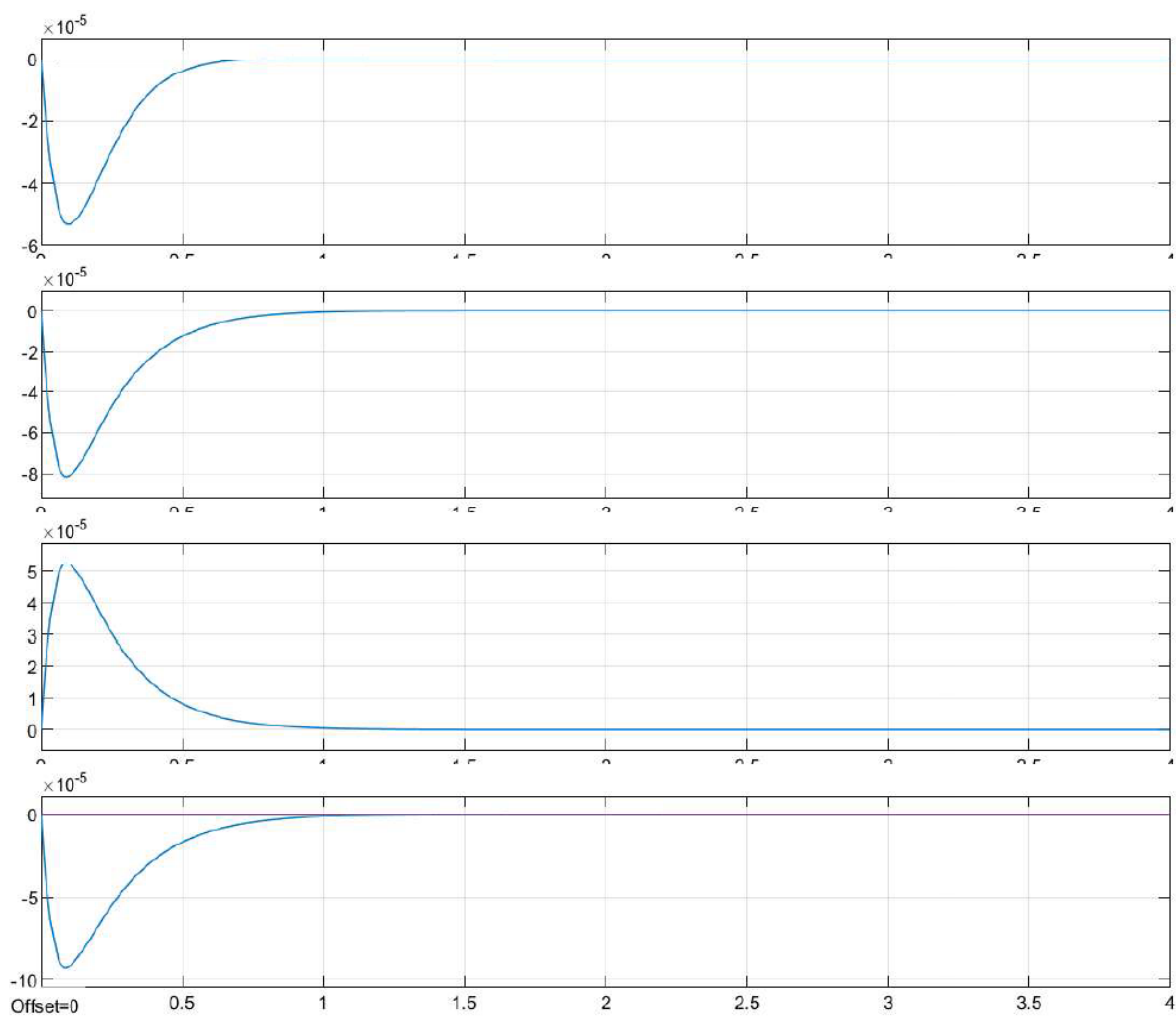


Рисунок 33 — Переходный процесс по скорости сочленении 3,4,5,6.

Также был исследован переходный процесс управляющих крутящих моментов сочленений 3,4,5,6, используя предложенную систему (Рисунок 34).

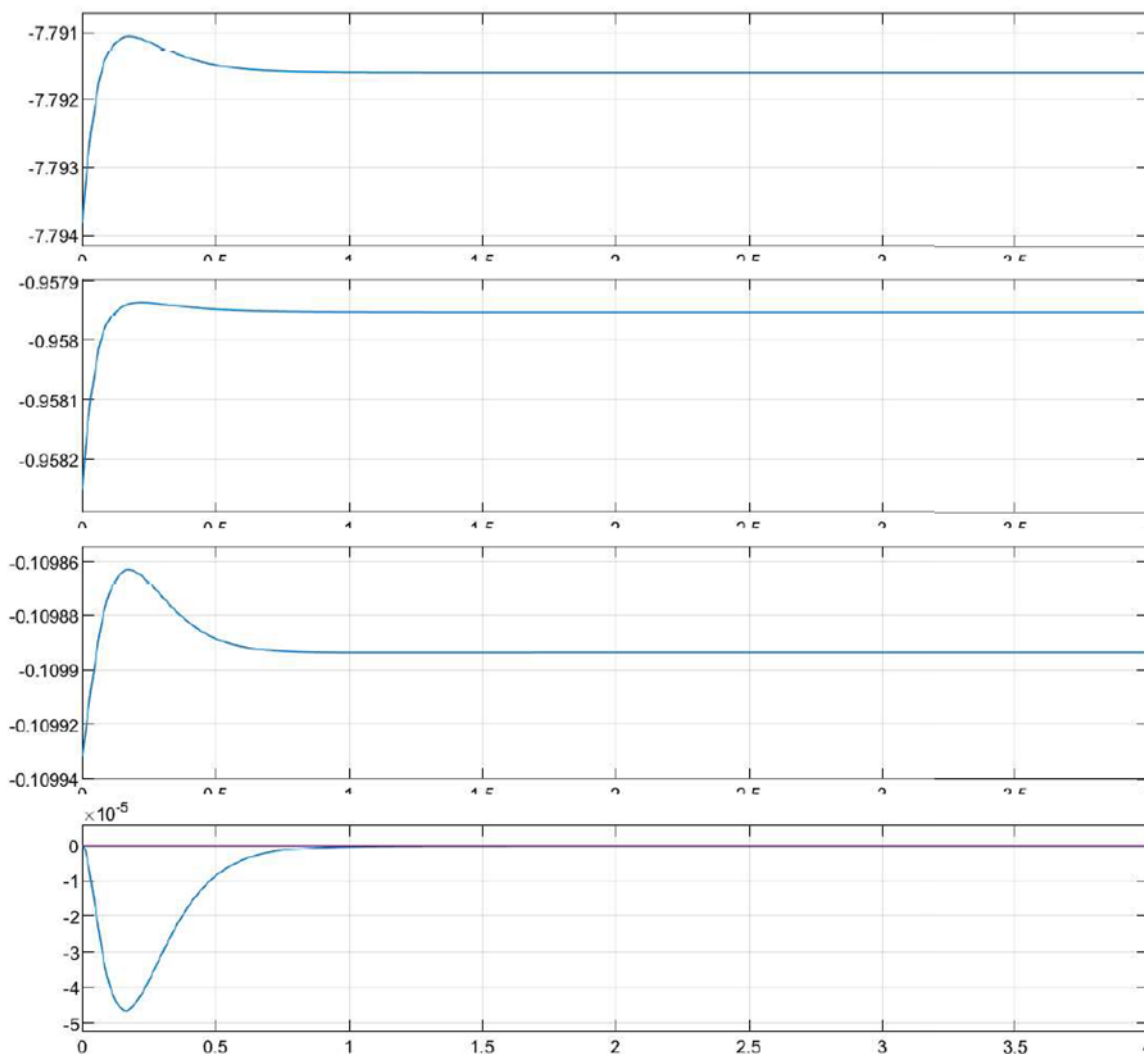


Рисунок 34 — Переходный процесс управляющих крутящих моментов сочленений.

Система управление схватом:

После достижения точки захвата, необходимо управлять схватом, чтобы захватить объект достаточно хорошо, удержать, манипулировать и уложить в коробку.

Из-за неточности полученной ширины захвата нейросетей используют отдельную систему управления схватом с обратной связью по силе реакции.

Управление схватом, расширяет вектор управляющих моментов на:

$$\tau = [\tau_1 \quad \tau_2 \quad \tau_7 \quad \tau_3 \quad \tau_4 \quad \tau_5 \quad \tau_6 \quad F_1 \quad F_2]^T \quad (87)$$

где F_1 – сила, приложенная к первой губке захвата,

F_2 – сила, приложенная ко второй губке захвата.

Объект считается захваченным, когда, захват крепко сжат. Для этого, сила равная $10 N$ прикладывается к каждой губке захвата, и останавливается только тогда, когда обе две силы реакции $F_{reaction 1}, F_{reaction 2}$ достигают определённого ограничения $2 N$, которое считается достаточным для удержания каждого из шести объектов.

Система выполняет алгоритм, записанный в псевдокоде на приложении А.

Выводы: после того как устойчивая система управления манипулятора и захвата были построены и иллюстрированы, можно моделировать полную систему в пакете ROS.

10 Моделирование и симуляция системы

После формирования всех подсистем, необходимо проверить систему на симуляции. Для моделирования и симуляции были использованы среды и ROS и Gazebo и программный язык Python.

Для моделирования и симуляции системы в целом необходимо использовать среду, которая позволяет всем подсистемам легко общаться друг с другом, и позволять выполнять аккуратную физическую симуляцию. Таким образом для симуляции была выбрана среда ROS и Gazebo. Программирование велось на языке Python.

Кинематическая цепь робота ABB (YuMi) описывается форматом URDF (Unified Robot Description Format), который содержит описание всех звеньев и сочленений, определяющих предка (parent) и потомка (child) для каждого сочленения (рисунка 35).

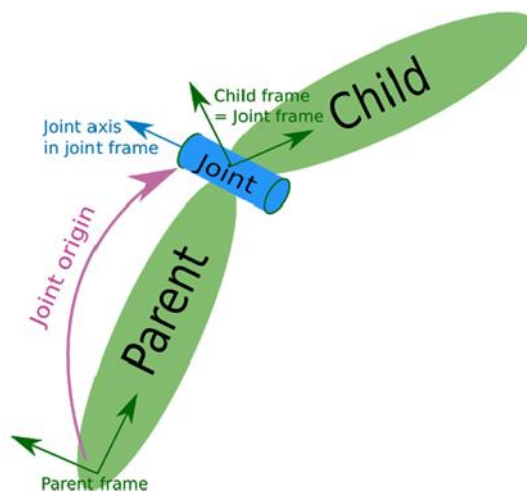


Рисунок 35 — Описание parent и child для каждого сочленения

В среде Gazebo установлена надстройка «Kinect camera» для симуляции модели камеры и получения от нее изображения глубины. С камеры получают следующие темы (topics) и сообщения (messages), которые содержат необработанные изображения глубины:

/camera/depth/camera_info (camera parameters for calibration and rectifying the image)

/camera/depth/image_raw (raw depth image)

/camera/depth/points (equivalent point cloud)

Камера публикует по своей теме сообщения, содержащие строки глубины изображения(рисунок 36).

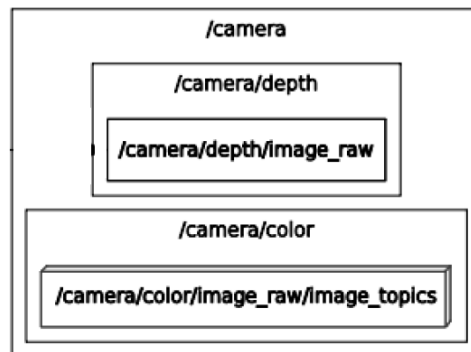


Рисунок 36 — Темы, полученные от камеры.

Для управления манипулятором в Gazebo необходимо определить и затем запустить набор контроллеров сочленений (конфигурации контроллеров загружаются из файла YAML). Контроллеры сочленений включают в себя `arm_controllers` и `gripper_controllers`. Созданный пакет YUMI запускает контроллеры сочленений и захвата и публикует состояния сочленений манипулятора и изображения глубины с камеры Kinect.

Для планирования движения и проверки на столкновения, используется узел `move_group` от MoveIt toolbox, которое обеспечивает генерацию траекторий больших степеней свободы для избегания препятствий и решения обратной задачи кинематики численными методами, используя «The KDL Kinematics Plugin». В MoveIt, определяем ПД регулятор.

На рисунке 37, можно увидеть узлы (nodes) и темы(topics) разработанного пакета системы в среде ROS.

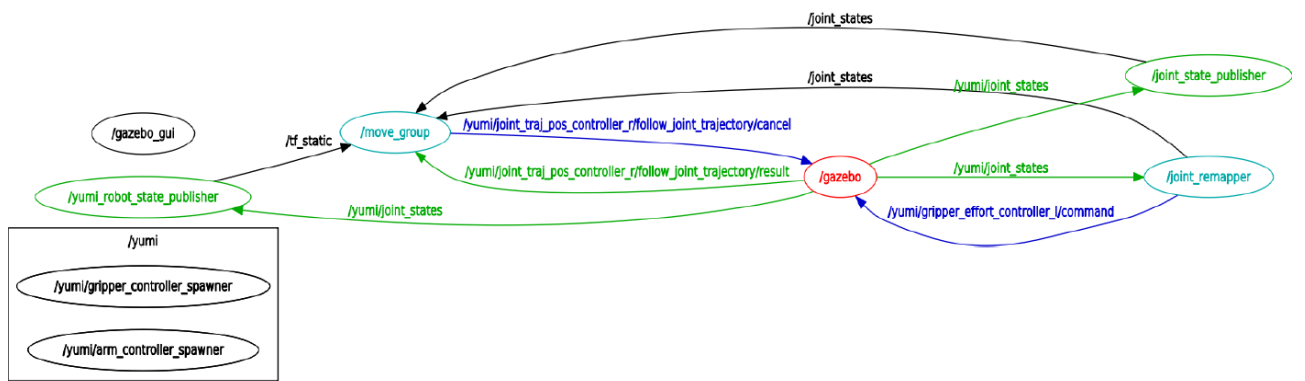


Рисунок 37 — узлы (nodes) и темы(topics) системы в среде ROS.

Основные узлы и темы ROS: (Рисунок 37)

Узел **joint_state_publisher**, используется для публикации состояния сочленении на теме `/joint_states` в определенной частоте;

Узел **gazebo**, получает команды управления сочленениями и захватом от узла `move_group` и публикует состояния сочленений манипулятора и изображения глубины от камеры;

Узел **move_group**, генерируемый Moveit toolbox. Он включает в себя обратную кинематику манипулятора и обеспечивает решение для достижения желаемой позиции и скорости, избегая столкновений и учитывая ограничения робота;

Тема `/tf` используется для отслеживания всех 3D-координатных систем в роботизированной системе, которые изменяются с течением времени: таких как система манипулятора (robot frame), базовая система (base frame) и система захвата (gripper frame) и публикует состояния сочленений манипулятора.

Более подробное схемы тем и узлов можно увидеть на рисунке 38.

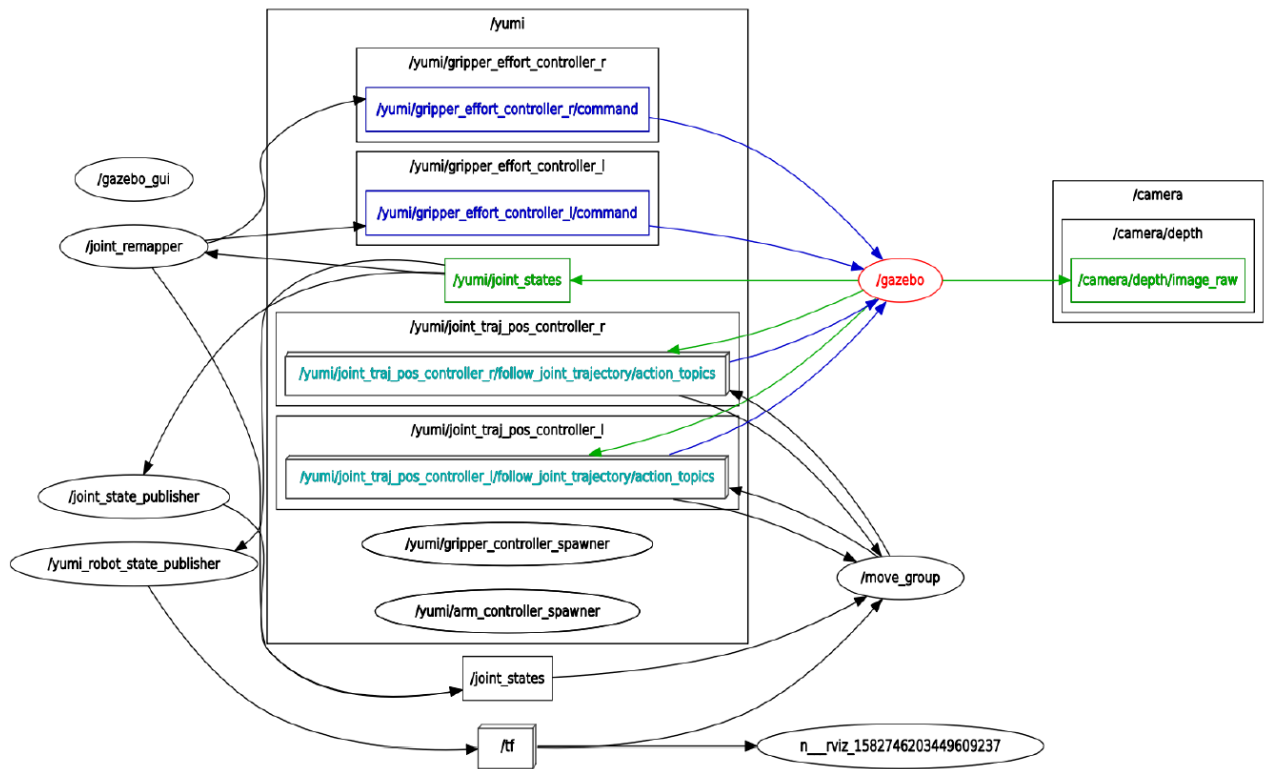


Рисунок 38 — подробная схема системы в среде ROS.

Разработанный пакет для симуляции системы упаковки с помощью нейросети можно найти на сайте Github по ссылке [26].

На рисунке 39, показана симуляция системы при выполнении задачи упаковки используя разработанного пакета в ROS, где сначала снимается изображение глубины, камерой и происходит определение правильной позиции захвата объекта с помощью нейросети (рисунок 39.а), далее происходит захвата объекта и перенос его в ножное место над коробкой (рисунок 39. b,c), далее схват открывается чтобы уложить объект на коробке (рисунок 39.d).

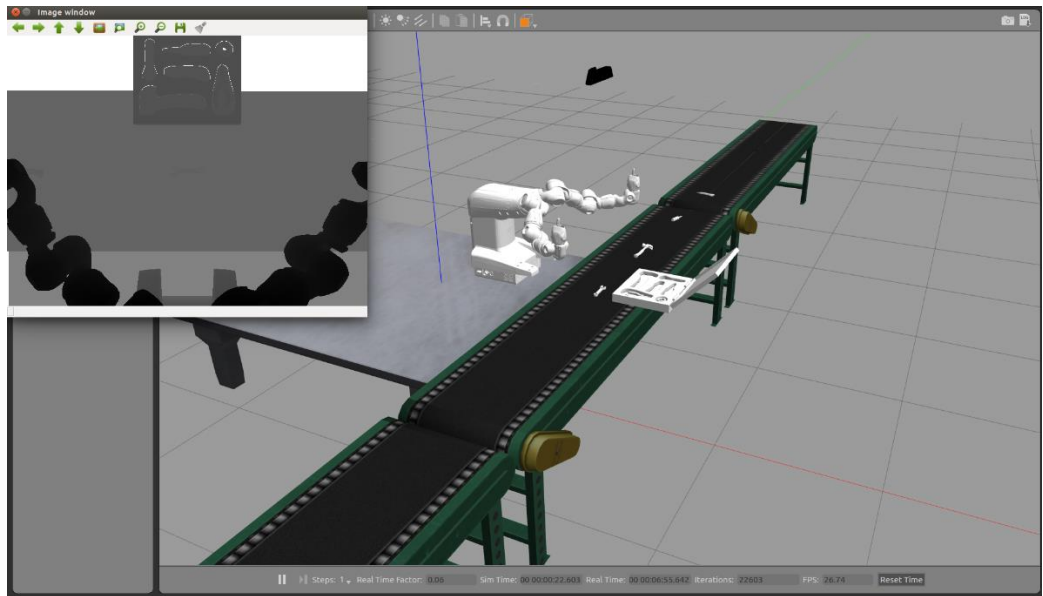


Рисунок 39 — (а) Симуляция системы упаковки, изображение глубины.

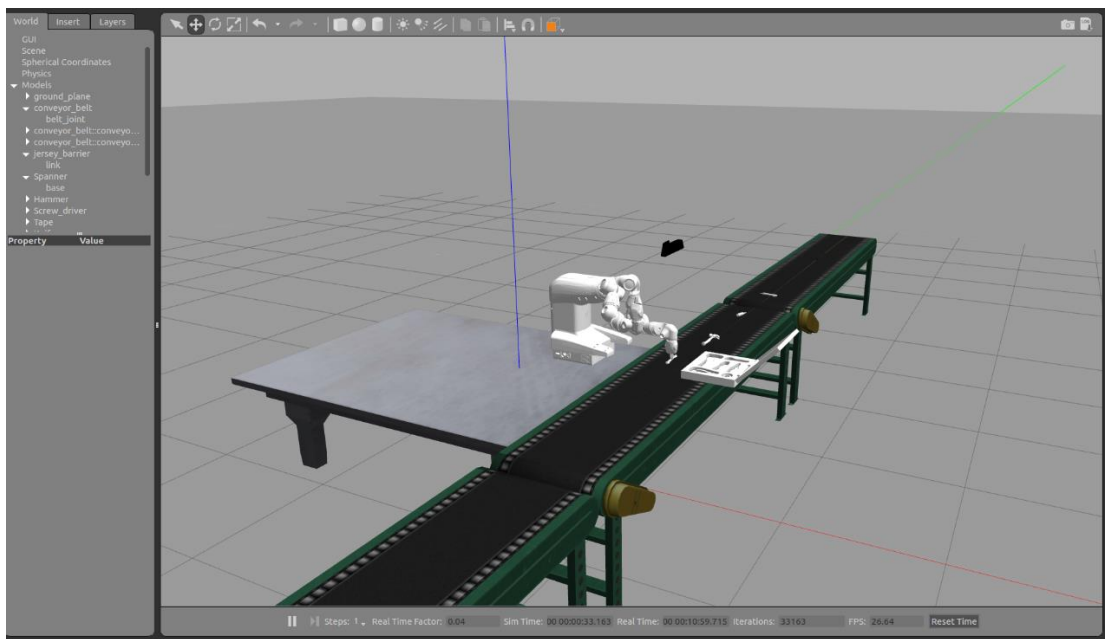


Рисунок 39 — (b) Симуляция системы упаковки; захват объекта.

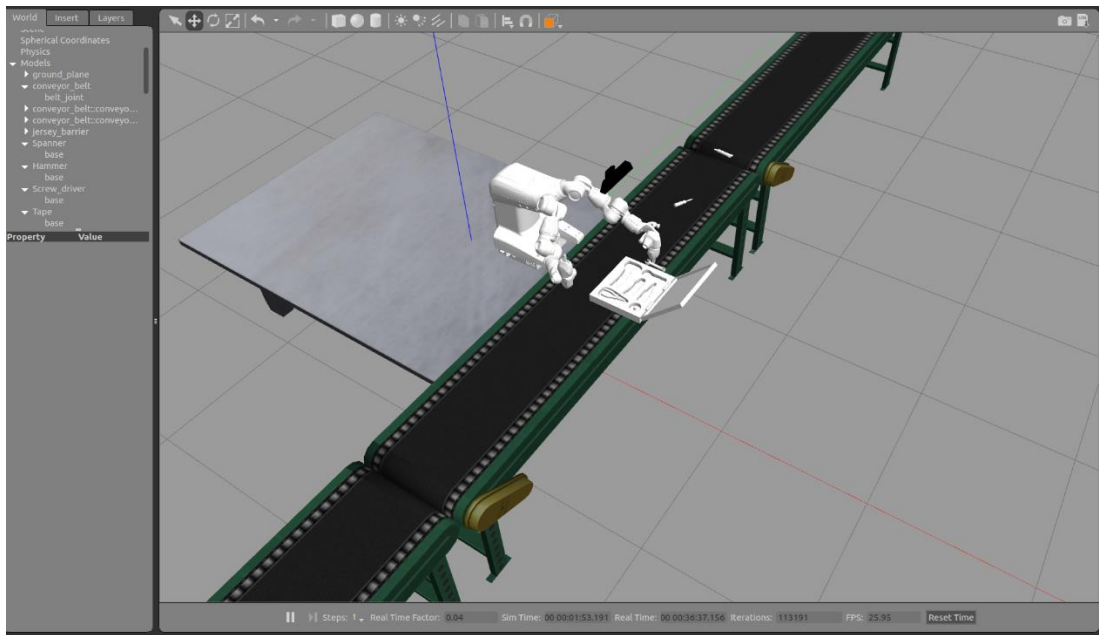


Рисунок 39 — (с) Симуляция системы упаковки; перенос объекта в ножное место.

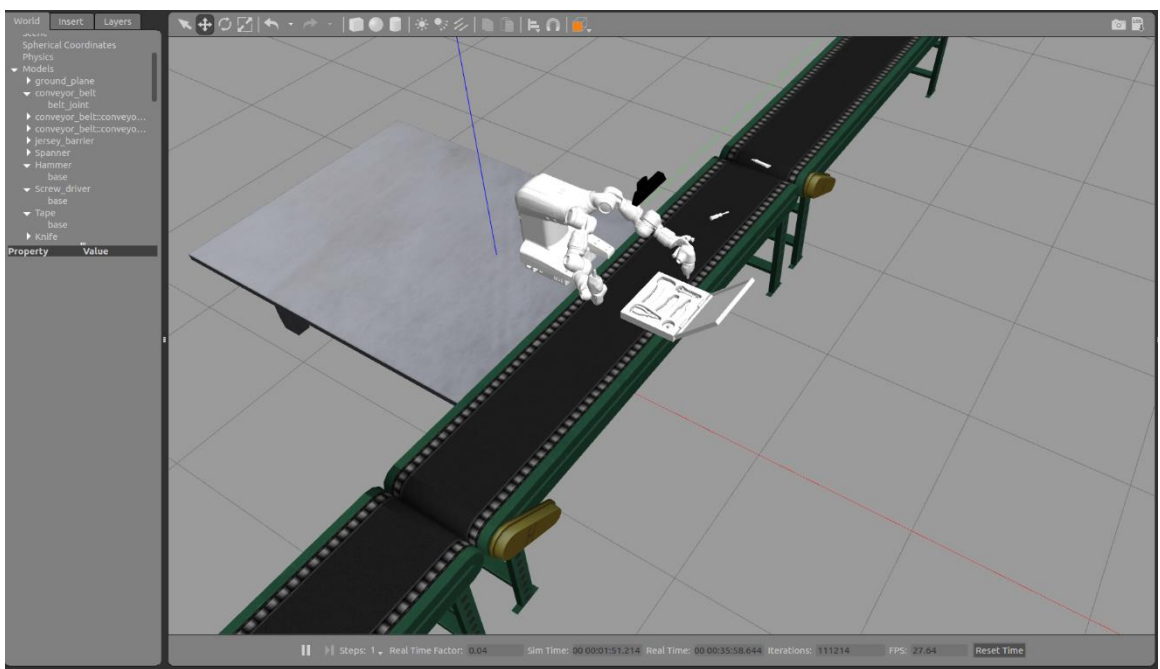


Рисунок 39 — (d) Симуляция системы упаковки; уложение объекта в ножное место.

ЗАКЛЮЧЕНИЕ

В настоящей выпускной квалификационной работе магистра было рассмотрено теоретическое проектирование процесса упаковки набора инструментов манипулятором с помощью искусственных нейронных сетей и его практическая реализация.

Были построены две нейронных сети, одна – для обнаружения правильной позиции захвата инструмента, вторая – для распознавания инструмента.

Нейросеть обнаружения захвата сначала обучалась на изображениях глубины большого набора общих объектов, а затем переобучилась на наборе шести инструментов задачи. Нейронная сеть распознавания инструментов обучилась классификации инструментов на шести классах объектов.

Система управления манипулятором по крутящему моменту, была изучена и проектирована теоретически, затем проверена на симуляции.

После выполнения задачи, можно заключить следующее:

— можно использовать предлагаемую нейросеть для обнаружения правильного положения захвата и его ориентации: построенная нейросеть достаточно хорошо обучилась правильным положениям захвата. В меньшей мере процесс обучения был удачным по отношению к углам ориентации в особом случае, при угле равном $+\frac{\pi}{2}$ или $-\frac{\pi}{2}$;

— не рекомендуется использовать нейросеть для обучения ширине захвата. Нейросеть неудачно обучилась правильной ширине захвата. Неправильные ширины можно объяснить особыми требованиями ширины захвата, для удерживания каждого объекта;

— проблема определения ширины захвата решается дополнительным управлением шириной захвата с обратной связью по силе реакции;

— предложенная система управления манипулятором, крутящим моментом, можно использовать для управления манипулятором при выполнении задачи упаковки набора инструментов.

В связи с аппроксимацией и упрощением, которые делаются в средах симуляции (Gazebo, ROS и т.д.), предлагается для дальнейшей работы практически реализовать систему на реальном манипуляторе ABB IRB 14000 (Yumi), чтобы обеспечить эффективность работы системы в действительной более-сложной физике.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Abb. YuMi®-IRB 14000 Overview.
2. Saxena A., Driemeyer J., Ng A.Y. Robotic grasping of novel objects using vision // The International Journal of Robotics Research. Sage Publications Sage UK: London, England, 2008. Vol. 27, № 2. P. 157–173.
3. Lenz I., Lee H., Saxena A. Deep learning for detecting robotic grasps // The International Journal of Robotics Research. SAGE Publications Sage UK: London, England, 2015. Vol. 34, № 4–5. P. 705–724.
4. Ciocarlie M. et al. Towards reliable grasping and manipulation in household environments // Experimental Robotics. Springer, 2014. P. 241–252.
5. Jiang Y., Moseson S., Saxena A. Efficient grasping from rgb-d images: Learning using a new rectangle representation // 2011 IEEE International conference on robotics and automation. IEEE, 2011. P. 3304–3311.
6. Torrey L., Shavlik J. Transfer learning // Handbook of research on machine learning applications and trends: algorithms, methods, and techniques. IGI Global, 2010. P. 242–264.
7. Redmon J., Angelova A. Real-time grasp detection using convolutional neural networks // 2015 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2015. P. 1316–1322.
8. Sahbani A., El-Khoury S., Bidaud P. An overview of 3D object grasp synthesis algorithms // Robotics and Autonomous Systems. Elsevier, 2012. Vol. 60, № 3. P. 326–336.
9. Bohg J. et al. Data-driven grasp synthesis—a survey // IEEE Transactions on Robotics. IEEE, 2013. Vol. 30, № 2. P. 289–309.
10. Лесков А.Г., Бажинова К.В., Селиверстова Е.В. МЕТОДЫ ОПРЕДЕЛЕНИЯ КАЧЕСТВА ЗАХВАТА ОБЪЕКТА.
11. Bicchi A., Kumar V. Robotic grasping and contact: A review // Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Ro-

- botics and Automation. Symposia Proceedings (Cat. No. 00CH37065). IEEE, 2000. Vol. 1. P. 348–353.
12. Prattichizzo D. et al. On the manipulability ellipsoids of underactuated robotic hands with compliance // *Robotics and Autonomous Systems*. Elsevier, 2012. Vol. 60, № 3. P. 337–346.
 13. Лесков А.Г., Селиверстова Е.В. Алгоритм планирования и выбора способа захвата деформируемого объекта многопалым хватным устройством манипуляционного робота // *Мехатроника, автоматизация, управление. Общество с ограниченной ответственностью Издательство Новые технологии*, 2017. Vol. 18, № 11. P. 739–744.
 14. Miller A.T., Allen P.K. Graspit! a versatile simulator for robotic grasping // *IEEE Robotics & Automation Magazine*. IEEE, 2004. Vol. 11, № 4. P. 110–122.
 15. Balasubramanian R. et al. Physical human interactive guidance: Identifying grasping principles from human-planned grasps // *IEEE Transactions on Robotics*. IEEE, 2012. Vol. 28, № 4. P. 899–910.
 16. Pinto L., Gupta A. Supersizing self-supervision: Learning to grasp from 50k tries and 700 robot hours // *2016 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2016. P. 3406–3413.
 17. Kumra S., Kanan C. *Robotic Grasp Detection using Deep Convolutional Neural Networks*. 2017.
 18. Cornell University. Robot Learning Lab. Learning to Grasp [Electronic resource]. 2017. URL: http://pr.cs.cornell.edu/grasping/rect_data/data.php.
 19. Mahler J. et al. Dex-Net 2.0: Deep Learning to Plan Robust Grasps with Synthetic Point Clouds and Analytic Grasp Metrics. 2017.
 20. Bicchi A. et al. 2011 IEEE International Conference on Robotics and Automation (ICRA) ; 9-13 May 2011, Shanghai, China.
 21. Morrison D., Corke P., Leitner J. Closing the loop for robotic grasping: A real-time, generative grasp synthesis approach // *arXiv preprint arXiv:1804.05172*. 2018.

22. Pujol J. The solution of nonlinear inverse problems and the Levenberg-Marquardt method // Geophysics. Society of Exploration Geophysicists, 2007. Vol. 72, № 4. P. W1–W16.
23. Mohammed S. Kinematic Motion Planning for a 7-Axis Robotic Arm (LWA70 by Schunk). 2016.
24. Gu Y., Loh N.K. Dynamic model for industrial robots based on a compact Lagrangian formulation // 1985 24th IEEE Conference on Decision and Control. IEEE, 1985. P. 1497–1501.
25. Lewis F.L., Dawson D.M., Abdallah C.T. Robot manipulator control: theory and practice. CRC Press, 2003.
26. Mais Jamal. YuMi-control package [Electronic resource]. URL: <https://github.com/MaisJamal/YuMi-control>.

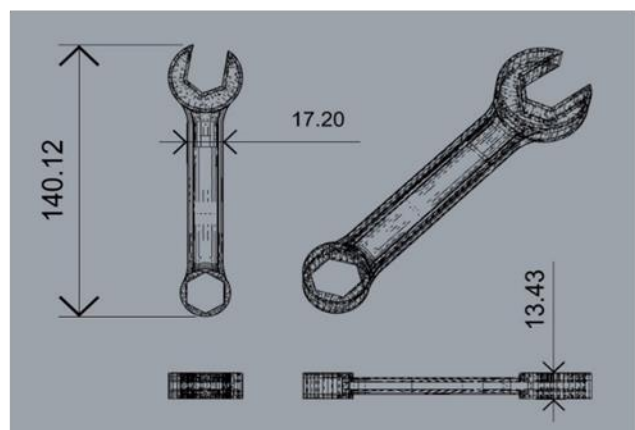
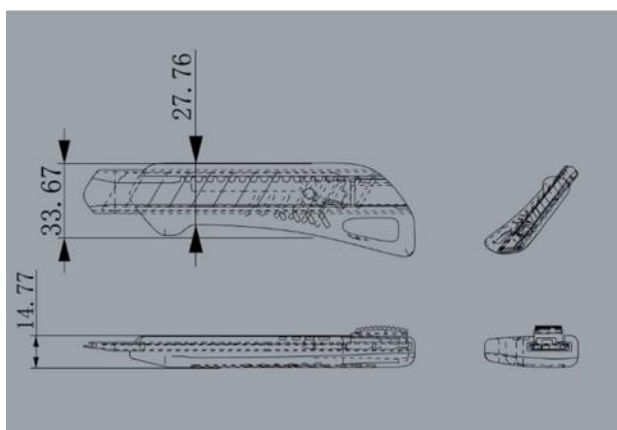
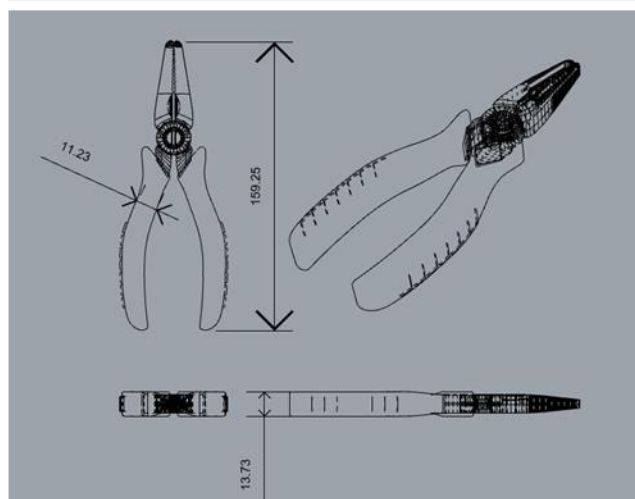
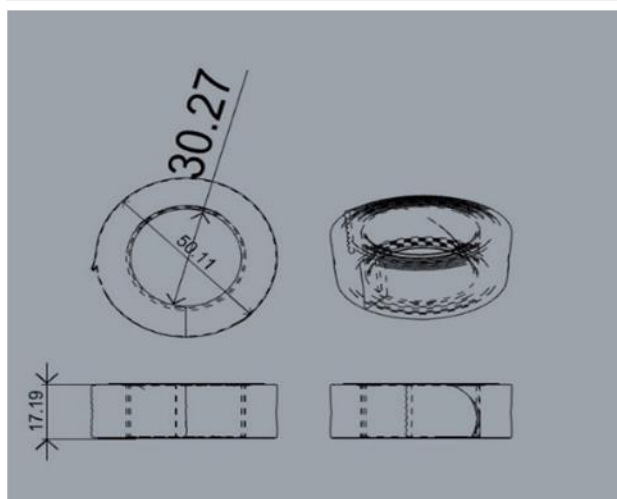
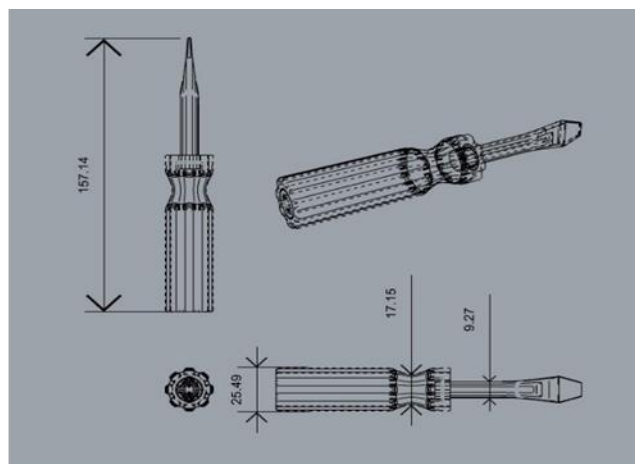
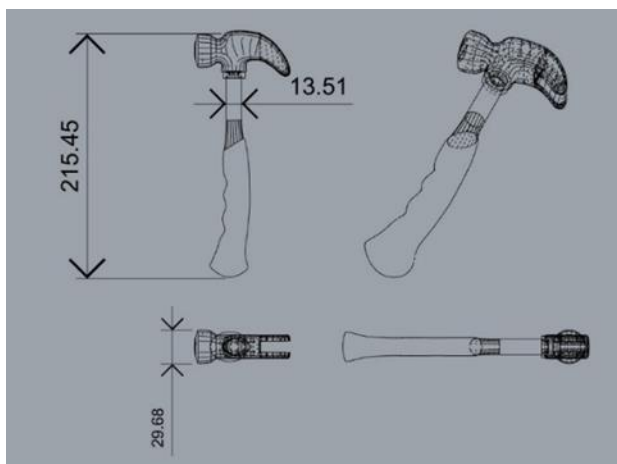
ПРИЛОЖЕНИЕ А

Графическая часть выпускной квалификационной работы

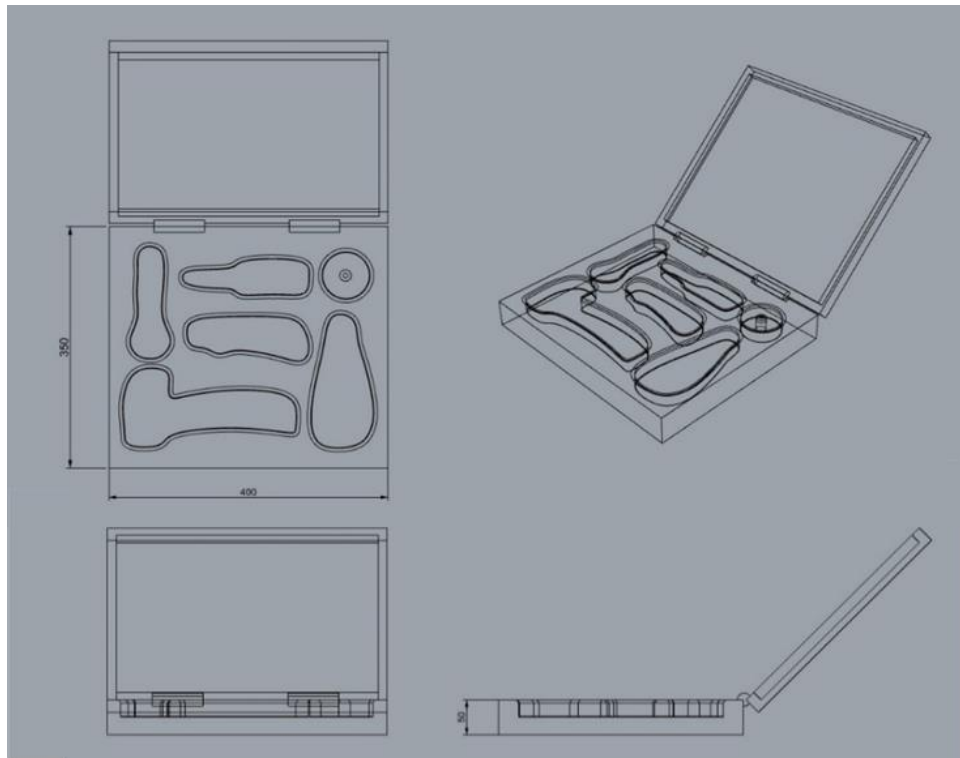
В графическую часть выпускной квалификационной работы входят:

- чертеж(эскизы) объектов: отвёртка, строительный нож, плоскогубцы, молоток, ключ рожковый и изоляционная лента;
- чертежа коробки;
- алгоритм управления схвата;

Эскизы объектов: отвёртка, строительный нож, плоскогубцы, молоток, ключ рожковый и изоляционная лента (в мм):



Чертежа коробки. (в миллиметрах):



Алгоритм управления схвата:

```
1  if is_gripper_open = True do:
2      while  $F_{reaction\ 1} < 2$  or  $F_{reaction\ 2} < 2$  do:
3           $F_1 = 10$ 
4           $F_2 = 10$ 
5      end while
6      is_gripper_open = False
7  end if
```