

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ
УНИВЕРСИТЕТ ИМ. В.Г. ШУХОВА

На правах рукописи



Волошкин Артём Александрович

**МЕТОДЫ ПРОЕКТИРОВАНИЯ И ОПТИМИЗАЦИИ АВТОНОМНОЙ
РОБОТОТЕХНИЧЕСКОЙ СИСТЕМЫ ДЛЯ СБОРА ФРУКТОВ**

Специальность: 2.5.4 – Роботы, мехатроника и робототехнические системы

Диссертация

на соискание учёной степени кандидата технических наук

Научный руководитель:
д-р. техн. наук, профессор
Рыбак Лариса Александровна

Белгород 2025

ОГЛАВЛЕНИЕ

Введение.....	4
ГЛАВА 1. АНАЛИЗ СОСТОЯНИЯ ИССЛЕДОВАНИЙ.....	16
1.1 Анализ состояния исследований в области робототехнических систем для сбора фруктов.....	16
1.1.1 Робототехнические системы для массового сбора фруктов.....	17
1.1.2 Робототехнические системы для штучного сбора фруктов	20
1.1.3 Захватные устройства робототехнических систем для сбора фруктов	26
1.2 Анализ состояния исследований процесса сбора яблок	32
1.3 Требования к разрабатываемой робототехнической системе	38
1.4 Анализ исследований в области численного моделирования и оптимизации параметров робототехнических систем.....	40
1.5 Комбинированный метод проектирования робототехнической системы для сбора фруктов	46
Выводы по главе.....	54
ГЛАВА 2. МАТЕМАТИЧЕСКОЕ МОДЕЛИРОВАНИЕ робототехнической СИСТЕМЫ.....	56
2.1 Математическая модель кинематики робота-трипода с центральной пассивной кинематической цепью и телескопическим звеном	56
2.2 Результаты моделирования построения рабочей области робота-трипода с учётом особых положений и пересечений звеньев	65
2.3 Алгоритм оптимизации геометрических параметров робота-трипода	67
2.4 Алгоритм технического зрения для локализации деревьев и распознавания фруктов.....	73
2.5 Математическая модель захватного устройства.....	85
2.6 Оптимизация параметров захватного устройства.....	91
Выводы по главе.....	93
ГЛАВА 3. РАЗРАБОТКА ЭЛЕКТРОННО-ЦИФРОВОЙ МОДЕЛИ И ИМИТАЦИОННОЕ МОДЕЛИРОВАНИЕ	96

3.1 Имитационное моделирование робота-трипода с телескопическим звеном в динамической среде	96
3.2 Разработка цифрового двойника робота-трипода.....	106
3.3 Топологическая оптимизация конструктивных элементов робота-трипода ..	107
3.4 Имитационная модель устройства для складирования фруктов.....	114
3.5 Имитационное моделирование робототехнической системы с захватным устройством	116
Выводы по главе.....	122
ГЛАВА 4. ЭКСПЕРИМЕНТАЛЬНЫЕ ИССЛЕДОВАНИЯ.....	123
4.1 Выбор технических средств и разработка механической части экспериментального образца робототехнической системы	123
4.1.1 Конструкция захватного устройства.....	123
4.1.2 Экспериментальный образец робота-трипода с телескопическим звеном	125
4.2 Система управления и программное обеспечение	126
4.2.1 Система управления мобильной колесной платформой.....	126
4.2.2 Система управления движением робота-трипода	132
4.3 Проведение экспериментальных исследований.....	137
Выводы по главе.....	141
ЗАКЛЮЧЕНИЕ	143
СПИСОК ЛИТЕРАТУРЫ.....	146
ПРИЛОЖЕНИЕ 1	160
ПРИЛОЖЕНИЕ 2	172
ПРИЛОЖЕНИЕ 3	174
ПРИЛОЖЕНИЕ 4	175
ПРИЛОЖЕНИЕ 5	176

ВВЕДЕНИЕ

Актуальность темы исследования

В последние годы наблюдается резкое увеличение спроса на интеллектуальных роботов, способных выполнять сложные задачи без вмешательства человека практически во всех отраслях сельского хозяйства. Автоматизация сельскохозяйственных процессов, в том числе сбора фруктов, становится одним из перспективных направлений сельского хозяйства. Рост мирового спроса на продовольствие, удорожание ручного труда и стремление повысить эффективность производства заставляют сельхозпредприятия искать инновационные решения. Робототехнические системы для сбора фруктов позволяют не только сократить затраты на ручной труд, но и минимизировать брак собранных фруктов. Они так же, как и люди могут работать в сложных условиях пересечённой местности, ограниченной видимости, но способны это делать вне зависимости от трудового распорядка, что делает их незаменимыми в современном аграрном секторе. Известны разработки российских учёных, таких как Измайлов А.Ю., Хорт Д.О., Солёный С.В. и др., посвящённые разработке автономных роботов для сбора урожая, выявления и удаления сорняков, распыления химикатов и других сельскохозяйственных задач. Однако, при этом существует ряд проблем, связанных с повреждением фруктов, невысокой производительностью, маневренностью, что ограничивает их широкое применение. Актуальность решения указанной проблемы связана с созданием новых типов автономных робототехнических систем (РТС) для сбора фруктов, которые не оказывают негативного влияния на дерево и собираемые фрукты, обладают высокой степенью маневренности и автономности. Существенный вклад в развитие теории робототехнических систем различного назначения внесли ученые: Глазунов В.А., Каляев И.А., Болотник Н.Н., Брискин Е.С., Яцун С.Ф., Пшихопов В.Х., Жога В.В., и др. При разработке подобных систем возникает ряд проблем, связанных с неравномерностью и сложной геометрией фруктов, которая варьируется по размерам и форме, что затрудняет их точный захват без повреждений. Роботы

также должны работать в условиях неструктурированной среды, что приводит к необходимости разрабатывать адаптивные системы. В полевых условиях на работу сенсоров влияют условия внешней среды: дождь, ветер и изменения освещённости, что снижает эффективность распознавания объектов. Требования к системе управления зависят от способа отделения фруктов, например, отрывание, требует меньшей вычислительной мощности, чем отрезание, для которого необходимо определять положение плодоножки. Выбор конструкции захватного устройства также оказывает влияние на всю компоновку РТС. Большинство разработчиков подобных систем сосредоточены на отдельных компонентах системы — захвате, подвижной платформе или манипуляторе, что ограничивает целостное решение задачи автоматизации сбора фруктов. Если подходить к разработке РТС комплексно, становится очевидным, что эффективность одного компонента РТС может быть значительно снижена другим, разработка которого не выполнена должным образом. В условиях стремительного развития робототехники повышение производительности проектирования роботов обеспечивается за счёт использования новейших методов цифрового проектирования и оптимизации. Для этого часто используют эволюционные, генетические и другие оптимизационные алгоритмы, которые позволяют значительно улучшить параметры конструкций, сокращая вес, повышая жёсткость и снижая затраты на производство. Это особенно важно для автономных РТС, предназначенных для работы в неструктурированной среде в течение длительного времени, где требуется максимальная производительность. Также важно обеспечить транспортировку собранных фруктов без повреждений, включая процесс перемещения фрукта в контейнер, последующие складирования и доставку в пункт разгрузки для последующего хранения. При этом задача бережной транспортировки и складирования фруктов крайне мало освещена в научных работах и является серьёзной проблемой для автономного процесса сбора фруктов.

Степень разработанности темы исследования

Анализ широкого спектра литературных источников, включая исследования в области робототехнических систем для сбора фруктов, продемонстрировал значительный прогресс в разработке прототипов роботов, способных работать с производительностью, сравнимой с человеком. Современные системы используют технологии технического зрения и алгоритмы искусственного интеллекта для определения степени спелости фруктов и их точного расположения в условиях неструктурированной среды.

Интеграция алгоритмов искусственного интеллекта позволяет эффективно распознавать объекты и осуществлять точную навигацию к фруктам. Однако результаты анализа показывают, что топология и качество конструкции механической части роботизированных систем остаются существенным ограничением для реализации полного потенциала управляющих программ. Для достижения высокой скорости сбора современные прототипы часто оснащаются несколькими манипуляторами последовательной структуры, каждый из которых работает с относительно низкой скоростью, что компенсируется их одновременной работой.

Особое внимание в исследованиях уделено недостаткам захватных устройств. Как правило, их проектирование ведётся изолированно от общей архитектуры системы, что приводит к снижению её общей эффективности. Многие захваты разрабатываются без учёта дальнейшего технологического процесса, что может приводить к повреждению фруктов. Кроме того, выбор материалов и конфигураций часто не соответствует требованиям к долговечности и адаптивности устройств. Жёсткие захваты имеют низкую адаптивность к фруктам разного размера и формы, тогда как мягкие захваты, хоть и снижают риск повреждения, часто обладают недостаточной силой удержания.

Эти проблемы также связаны со сложностью интеграции захватов с манипулятором и системой управления роботом. Несоответствие параметров между компонентами приводит к снижению точности и скорости работы, что

ограничивает эффективность системы в целом. Таким образом, для повышения производительности роботизированных систем необходимо уделять больше внимания комплексному проектированию хватных устройств, манипулятора и архитектуры управляющей системы. Кроме того, серийное производство робототехнических систем для сбора фруктов отсутствует, а большинство известных решений находятся на стадии прототипирования, что указывает на необходимость дальнейших исследований и разработки новых высокоэффективных решений.

Выделены ключевые характеристики автономных РТС для сбора фруктов, к которым относятся: длительность автономной работы, скорость сбора фруктов, их сохранность и товарный вид. Исследование конструктивных и схемно-технических решений хватных устройств для сбора фруктов показало, что большинство из них основаны на пальцевых захватах, которые вынуждают после каждого отделения фрукта выполнять операцию складирования, а для позиционирования хватного устройства, зачастую, используются механизмы последовательной структуры, обладающие относительно низкой скоростью движения. Несмотря на то, что исследуемые системы могут обеспечивать высокие показатели отдельных характеристик сбор урожая фруктов, они также имеют некоторые проблемы:

- первая проблема – современные РТС для сбора фруктов чаще всего используют традиционные механизмы последовательной структуры, обеспечение жёсткости и точности которых затруднительно без увеличения их металлоёмкости. Кроме того, механизмы последовательной структуры с большой рабочей областью могут привести к опрокидыванию подвижного автономного колёсного робота;

- вторая проблема – хватное устройство без внутренней полости или встроенного конвейера для транспортировки фруктов в контейнер имеет низкую скорость сбора и требует более сложный алгоритм управления, учитывающий складирование фруктов, при этом, не гарантируя сохранность;

- третья проблема – существующие устройства транспортировки фруктов от захватного устройства и до контейнера фруктов имеют либо сложное устройство, либо не обеспечивают сохранность фруктов;

- четвертая проблема – в большей части исследований не рассмотрена задача формализации начальной и средней стадий проектирования роботизированных систем, а выбор геометрических параметров не обоснован должным образом и не оптимизирован.

Создание новых эффективных моделей, методов моделирования, оптимизации и проектирования, а также полномасштабного экспериментального образца автономной РТС для сбора фруктов, являются актуальными задачами исследования.

Объект исследования — автономная робототехническая система для сбора фруктов, включающая колесную мобильную платформу и робот-трипод с центральной пассивной кинематической цепью, телескопическим звеном и захватным устройством.

Предмет исследования — методы и алгоритмы моделирования, оптимизации геометрических и конструктивных параметров, топологии, а также формализации всех стадий проектирования автономной робототехнической системы.

Цель работы

Повышение автономности и расширение функциональных возможностей робототехнических систем для сбора фруктов за счёт совершенствования метода проектирования на основе формализации всех его стадий, оптимизации параметров, топологии и машинного обучения.

Для достижения поставленной цели в процессе работы решались следующие **задачи**:

1. Обоснование структуры робототехнической системы для сбора фруктов на основе колесной мобильной платформы и робота-трипода с центральной

пассивной кинематической цепью и телескопическим звеном с захватным устройством.

2. Разработка математической модели робототехнической системы, описывающей зависимость положения выходного звена робота-трипода с пассивной центральной кинематической цепью от углов ориентации его рабочей платформы и выдвижения установленного на ней телескопического звена.

3. Синтез эвристического алгоритма многокритериальной оптимизации параметров робота-трипода с пассивной центральной кинематической цепью на основе комбинации однокритериальных эволюционных алгоритмов с переменными коэффициентами важности с целью расширения диапазонов ориентации робота-трипода и минимизации размеров конструкции.

4. Синтез двухэтапного алгоритма технического зрения для локализации деревьев, а также точного распознавания и захвата фруктов на основе интеграции сверточных нейронных сетей и преобразования Хафа.

5. Разработка комбинированного метода проектирования робототехнической системы на основе формализации всех этапов проектирования и топологической оптимизации распределения материала в конструктивных элементах.

Научная новизна работы

1. Предложена модульная структура робототехнической системы для сбора фруктов, выполненная на базе мобильной колёсной платформы и робота-трипода, в котором для исключения избыточных степеней свободы и повышения управляемости использована пассивная центральная кинематическая цепь с выдвижным телескопическим звеном, а также захватное устройство с гибкой трубой для транспортировки и складирования фруктов; предложена структура и параметризованная модель специального захватного устройства, позволяющего комбинировать методы отрывания и скручивания фруктов за счёт наличия на корпусе винтовой поверхности, по которой перемещается внутренняя труба, имеющая на одном торце пальцы, обеспечивающие захват и отделение фруктов.

2. Разработана математическая модель кинематики робототехнической системы, описывающая зависимость положения телескопического звена с захватным устройством от углов ориентации подвижной платформы робота-трипода, учитывающая положения шарниров крепления кинематических цепей робота-трипода в виде систем нелинейных уравнений, определяющих ограничения на множество достижимых положений и ориентаций выходного звена и допустимых диапазонов движения линейных приводов.

3. Синтезирован эвристический алгоритм многокритериальной оптимизации параметров робота-трипода, на первом этапе которого выполняется нормализация критериев, а на втором этапе - итеративная процедура оптимизации на основе однокритериальных эволюционных алгоритмов с использованием свёртки критериев с переменными коэффициентами важности. Использование в качестве критериев компактности конструкции и критерия, зависящего от количества недостижимых ориентаций платформы при заданных ограничениях на диапазоны приводных штанг, позволяет расширить функциональные и эксплуатационные характеристики.

4. Синтезирован двухэтапный алгоритм технического зрения для локализации деревьев и распознавания фруктов на основе интеграции нейросетевых алгоритмов и преобразования Хафа; на первом этапе алгоритма выполняется формирование списка заданий на основе идентификации крон деревьев, а на втором - актуализация списка заданий на основе идентификации и сегментации плодов с учётом их спелости, что позволяет на основе визуальных данных комбинировать автономную навигацию и идентификацию координат объектов.

5. Разработан комбинированный метод проектирования робототехнической системы для сбора фруктов, включающий формализацию всех стадий проектирования: создание математических и электронно-цифровых моделей, цифровых двойников, топологическую оптимизацию распределения материала в конструктивных элементах с использованием CAD/CAE-систем, позволяющий на

основе полной динамической и имитационной моделей получить детализированный цифровой двойник робототехнической системы и ее рациональную конструкцию.

Теоретическая и практическая значимость работы

Теоретическая значимость заключается в развитии научных основ математического и имитационного моделирования, проектирования, включая новые схемно-технические решения, модели, алгоритмы технического зрения и машинного обучения, а также высокопроизводительные методы многокритериальной оптимизации геометрических и конструктивных параметров робототехнических систем на базе робота-трипода с дополнительной кинематической цепью.

Практическая значимость состоит в следующем:

- реализованы в виде прикладных программ практические эвристические алгоритмы многокритериальной оптимизации параметров робота-трипода с пассивной центральной кинематической цепью, а также оптимизации параметров захватного устройства для обеспечения компактности конструкции и маневренности робота;

- предложена и отработана методика экспериментальной проверки синтезированного SLAM алгоритма одновременной локализации и построения трехмерной карты местности, а также алгоритма распознавания и захвата плодов для автоматизированного сбора плодов, обеспечивающих автономную навигацию в пространстве;

- разработан метод проектирования робототехнических систем, включающий формализацию всех стадий проектирования: создание математических и электронно-цифровых моделей, цифровых двойников и прототипов с использованием CAD/CAE-систем, методы топологической оптимизации распределения материала в конструктивных элементах, позволяющий повысить точность и достоверность расчетов, а также производительность;

- разработана и апробирована конструкция специального захватного устройства, позволяющего комбинировать методы отрывания и скручивания фруктов, обеспечивая надежный захват и отделение их без повреждений;

- с использованием предложенных в работе методов разработана и апробирована в лабораторных и реальных условиях плодового сада конструкция полномасштабного экспериментального образца автономной робототехнической системы для сбора фруктов, позволяющая обеспечить надежный захват и отделение фруктов без повреждений, а также складирование и транспортировку.

Результаты исследований создадут предпосылки для создания и внедрения новых типов робототехнических систем в агропромышленный комплекс для автоматизации полного цикла процесса сбора урожая фруктов, включая транспортно-технологические, погрузочно-разгрузочные, складские работы при уборке урожая, позволит повысить производительность труда, улучшить качество собранных фруктов.

Результаты исследований имеют потенциал практического использования в практике отечественной сельскохозяйственной отрасли, тепличных, фермерских хозяйств и питомников, занимающихся возделыванием плодовых культур. Практическое внедрение результатов позволит расширить возможности оптимизации логистических процессов, автономной навигации и управляемости.

Методология и методы исследования

Поставленные задачи решаются с применением методов теоретической механики, математического моделирования, многокритериальной оптимизации, векторной алгебры и вычислительной математики. Для реализации синтезированных алгоритмов разработаны программные модули на языке программирования C++ с использованием библиотеки параллельных вычислений OpenMP, MATLAB. Для визуализации результатов моделирования применено программное обеспечение Blender, которое позволило создать уникальные и наглядные 3D-изображения. Для разработки электронно-цифровых моделей и

имитационного моделирования использован программный комплекс Siemens PLM Software NX.

Положения, выносимые на защиту:

1. Структура робототехнической системы для сбора фруктов, выполненная на базе мобильной колёсной платформы и робота-трипода, в котором для исключения избыточных степеней свободы и повышения управляемости использована пассивная центральная кинематическая цепь с выдвижным телескопическим звеном, а также захватное устройство с гибкой трубой для транспортировки и складирования фруктов; структура и параметризованная модель захватного устройства, позволяющего комбинировать способы отрывания и скручивания фруктов, а также автоматическое перемещение в контейнер.

2. Математическая модель кинематики робототехнической системы, описывающая зависимость положения телескопического звена с захватным устройством от углов ориентации подвижной платформы робота-трипода, учитывающая положения шарниров крепления кинематических цепей робота-трипода к платформам в виде систем нелинейных уравнений, определяющих ограничения на множество достижимых положений и ориентаций выходного звена, а также допустимых диапазонов движения линейных приводов.

3. Алгоритм многокритериальной оптимизации параметров робота-трипода с пассивной кинематической цепью, на первом этапе которого выполняется нормализация критериев, а на втором этапе - итеративная процедура оптимизации на основе однокритериальных эволюционных алгоритмов с использованием свёртки критериев с переменными коэффициентами важности.

4. Двухэтапный алгоритм технического зрения для локализации деревьев и распознавания фруктов на основе интеграции нейросетевых алгоритмов и преобразования Хафа; на первом этапе алгоритма выполняется формирование списка заданий на основе идентификации крон деревьев, а на втором - актуализация списка заданий на основе идентификации и сегментации плодов с учётом их спелости;

5. Метод комбинированного проектирования робототехнической системы на основе формализации всех стадий проектирования, включая создание математических и электронно-цифровых моделей, цифровых двойников, топологическую оптимизацию распределения материала в конструктивных элементах с использованием CAD/CAE-систем.

Степень достоверности полученных результатов

Достоверность результатов обеспечивается строгими математическими выводами, согласованностью с опубликованными результатами научных исследований других авторов, подтверждается результатами компьютерного моделирования (с проверкой применённых моделей и алгоритмов на задачах, имеющих аналитическое решение), лабораторными и натурными испытаниями разработанного экспериментального образца автономной робототехнической системы для сбора фруктов. Результаты диссертации докладывались и обсуждались на российских и международных научных конференциях, получили положительный отзыв научной общественности, а также предприятий агропромышленного комплекса.

Реализация результатов исследования

Спроектирован и изготовлен экспериментальный образец автономной робототехнической системы сбора фруктов, который успешно прошёл экспериментальные испытания на базе научной лаборатории мехатроники и робототехники Научно-исследовательского института робототехники и систем управления (НИИ РТиСУ) БГТУ им. В.Г. Шухова, а также натурные испытания в условиях садоводческого хозяйства.

Работа выполнена при финансовой поддержке Государственного задания Минобрнауки РФ, проект № FZWN-2020–0017.

Теоретические результаты диссертационной работы используются в учебном процессе в БГТУ им. В.Г. Шухова при подготовке бакалавров и магистров по направлению подготовки 15.03.05 «Конструкторско-технологическое обеспечение машиностроительных производств» - направленность «Технология

машиностроения» и направлению подготовки 15.04.06 «Мехатроника и робототехника» - направленность «Робототехника и искусственный интеллект»

Полученные в диссертационной работе результаты внедрены и используются в проектно-конструкторской и производственной деятельности ООО «РобоКомпонент» (г. Москва)

Апробация результатов работы

Основные результаты представлены на следующих конференциях:

– XIV Международная конференция «Оптимизация и Приложения» OPTIMA, (Петровац, Черногория, 2023);

– VI, Международная конференция «Достижения в робототехнике» AIR:2023 (г. Пенджаб, Индия, 2023);

– Международная конференция IFToMM по механизмам, передачам и приложениям MeTrApp (Пуатье, Франция, 2023)

– 15-я Международная конференция «Интеллектуальные системы» INTELS'2022 (г. Москва, Россия).

– XXXIV Международная инновационная конференция молодых учёных и студентов МИКМУС (г. Москва, ИМАШ им. А.А. Благонравова РАН, 2022);

Публикации

По материалам диссертации опубликовано 22 научных работы, из них 7 в рецензируемых научных изданиях, входящих в перечень ВАК и 15 в научных изданиях, входящих в Scopus и Web of Science. Получен 1 патент РФ на изобретение и 4 свидетельства о государственной регистрации программ для ЭВМ.

Структура диссертации

Диссертационная работа состоит из введения, четырёх глав, заключения и списка литературы. Общий объем диссертации 176 страниц. Диссертационная работа содержит 10 таблиц, 82 рисунка и 5 приложений. Список литературы включает 128 источников.

ГЛАВА 1. АНАЛИЗ СОСТОЯНИЯ ИССЛЕДОВАНИЙ

В главе выполнен обзор и анализ существующих робототехнических систем для сбора фруктов. Проанализированы преимущества и недостатки существующих устройств и способов сбора фруктов. Выявлены требования к разрабатываемой автономной робототехнической системе для сбора фруктов. Выполнен анализ исследований в области моделирования и оптимизации, которые могут применяться для задач оптимизации конструктивных и геометрических параметров роботов в процессе проектирования.

1.1 Анализ состояния исследований в области робототехнических систем для сбора фруктов

В связи с резким увеличением численности населения сельскохозяйственная промышленность находится под давлением необходимости увеличения посевных площадей и производства, что увеличивает цену на и так дорожающий ручной труд [1-3]. Разработка интеллектуальных роботов, способных выполнять сложные задачи без какого-либо вмешательства человека могут стать залогом сохранения экономического развития передовых стран. Одной из ключевых отраслей экономики является аграрный сектор. Ещё в 1980-х годах американские учёные С. Schertz и D. Brown предложили использовать роботов для сбора фруктов. В последствии появлялось все больше и больше результатов схожих исследований [4]. На основе анализа научных публикаций в области автоматизации сбора плодовых культур установлено, что современные исследования сосредоточены на двух ключевых направлениях. Во-первых, осуществляется модернизация и адаптация существующих средств механизации, включая совершенствование конструктивных элементов и алгоритмов управления. Во-вторых, ведётся активная разработка роботизированных систем, включая внедрение искусственного интеллекта и технического зрения для обеспечения точности и адаптивности процессов сбора урожая. Эти подходы направлены на повышение эффективности

агропромышленного комплекса за счёт снижения зависимости от ручного труда и минимизации потерь урожая [5-7]. Актуальность проблемы автоматизации сбора урожая плодовых культур подтверждается не только локальными исследованиями, но и глобальными тенденциями в агропромышленном секторе. Как показывает анализ работ [8-12], в течение последнего десятилетия научное сообщество активно разрабатывает РТС для сбора фруктов, которые классифицируются по двум ключевым категориям:

1. Для деликатного сбора фруктов, где основным фактором является сохранить фрукт без повреждений.
2. Для массового сбора фруктов, где основным фактором является скорость и объёмы собираемых фруктов.

1.1.1 Робототехнические системы для массового сбора фруктов

Массовый сбор фруктов, как правило, происходит при помощи виброустановки, которая воздействует на ствол дерева, стряхивая фрукты в предварительно разложенный вокруг ствола каркас-сетку. Одна из таких установок представлена в работе [13], где представлен анализ эффективности трёх типов оборудования, разработанного для механизации сбора урожая (рисунок 1.1, а, б, в). Исследование подчёркивает, что, несмотря на повреждение плодов, оборудование обеспечивает высокую манёвренность, минимальный вред для деревьев и экономическую эффективность, что позволило одобрить его как садоводческим предприятиям, так и экологическим организациям. Таким образом, технология успешно решает задачу сбора фруктов на улицах города, сочетая технические преимущества с экологической безопасностью, даже при условии, что сохранность плодов не является приоритетом.



а)

б)



в)

Рисунок 1.1 - Системы для сбора фруктов с вибрационным устройством и зонтом: а) Т-образный захват-вибратор; б) система «Esturipa» с зонтиком; в) Система «СМТА».

В статье [14] рассматривается проблема неэффективности традиционных методов вибрационного сбора плодов камелии, при которых высока доля пропущенных плодов и повреждённых цветочных почек. Авторы отмечают, что классические технологии, воздействующие исключительно на ствол, не учитывают архитектуру кроны камелии, что снижает качество уборки. Для решения этой задачи крона камелии разделена на две зоны — верхнюю и нижнюю — по максимальному диаметру. На основе анализа биомеханических свойств растения (упругость ветвей, распределение плодов, масса почек) разработана специализированная вибрационная система, которая воздействует на две зоны

(рисунок 1.2, а). Эксперименты показали, что зонированное воздействие, при помощи дополнительного захватного устройства для кроны (рисунок 1.2, б), снизило долю пропущенных плодов на 30–40%, а повреждение цветочных почек — на 25% по сравнению с традиционными методами.



Рисунок 1.2 - Системы для сбора плодов камелии; а) общий вид системы; б) 3D изображения захватного устройства и вибратора.

В работе [15] представлен комплекс для сбора плодов с оливковых деревьев. Основное отличие заключается в том, что воздействие происходит непосредственно на крону дерева, что оказывает меньшее воздействие на корневую систему и ствол дерева, но повреждая листья и ветви. Описанный метод сбора отлично подходит для низкорастущих деревьев и кустарников с тонким стволом, однако, это не применимо к высоким деревьям.

Существуют установки, которые направлены на естественное стимулирование падения фруктов за счет потока воздуха или воды. В работе [16] представлена система, которая обеспечивает сбор фруктов путём всасывания их в загрузочную воронку (рисунок 1.3). Главный недостаток уборочной системы с воздушным всасыванием является возможность сбора лёгких фруктов, при увеличении силы всасывания в загрузочную воронку начнёт попадать фракция, не предусмотренная для сбора. В настоящее время установки такого типа не используются.



Рисунок 1.3 - Общий вид уборочной системы с воздушным всасыванием

Способы сбора, основанные на механической вибрации, чаще всего подходят для фруктов с плотной кожурой или фруктов, не требующих сохранения товарного вида, например, для переработки или утилизации как в случае с декоративными деревьями в городских условиях. Таким образом, современные методы массового сбора плодов эффективны в узкоспециализированных задачах, но требуют дальнейшей адаптации к конкретным видам культур и рыночным требованиям. Их внедрение оправдано в условиях, где скорость и экономия ресурсов превалируют над необходимостью сохранения целостности плодов.

1.1.2 Робототехнические системы для штучного сбора фруктов

Штучный сбор фруктов должен осуществляться исключительно в автономном режиме, при этом ключевым фактором является скорость сбора. При низкой скорости сбора использование роботизированных средств перестаёт быть оправданным и не несёт экономической выгоды. Для реализации штучного сбора фруктов известны устройства с различными типами захватов, манипуляторов, подвижных оснований и систем обнаружения фруктов. Штучный сбор является более сложным с точки зрения управления и навигации и требует адаптации для конкретного фрукта [17]. Развитие роботизированных систем для штучного сбора плодов неразрывно связано с эволюцией технического зрения, ставшей фундаментом для автоматизации сельскохозяйственных процессов. Однако, внедрение систем технического зрения привело к новым вызовам: от калибровки систем в неструктурированных средах до баланса между скоростью и точностью. В работе [18] рассматриваются ключевые этапы развития систем сбора фруктов,

анализируются инновационные решения и выделяются перспективы, где синтез технического зрения, адаптивных манипуляторов и умных стратегий управления открывает путь к созданию универсальных роботов, способных работать в условиях реального сельского хозяйства.

В статье [19] представлены проектирование и разработка РТС с пневматическим мягким роботизированным захватным устройством, созданным для автоматизации сбора яблок (рисунок 1.4, а). РТС показала 67% эффективность отделения плодов при среднем времени захвата 7,3 секунды. Исследование проводилось без предварительного удаления проблемных яблок (например, с плотной плодоножкой), что позволило оценить реальные возможности системы и выявить необходимость прореживания для повышения доли плодов, доступных для автоматизированного сбора (рисунок 1.4, б).



Рисунок 1.4 - Системы для сбора яблок; а) общий вид системы; б) захватное устройство

Устройство и способ наполнения контейнеров для фруктов являются критически важными элементами в цепочке процесса сбора урожая, обеспечивая бережную транспортировку и равномерное распределение фруктов [20]. Совершенствование этих устройств отражает стремление минимизировать повреждения плодов и повысить эффективность логистики. Ранние модели, основанные на конвейерах с возвратно-поступательными пластинами (рисунок 1.5, а), постепенно уступили место современным решениям, таким как мягкие накладки и поролоновые ролики (рисунок 1.5, б), снижающие повреждения фруктов при

загрузке в контейнер. В послеуборочной обработке аналогичный прогресс привел к замене наклонных конвейеров на откидные лотки, вакуумные присоски, а также системы с цилиндрическими щётками и поворотными лопастями, которые аккуратно ориентируют плоды и предотвращают их деформацию (рисунок 1.5, в). Несмотря на разнообразие разработок, лишь немногие контейнеры адаптированы к сложным полевым условиям, где ограничения по габаритам, мобильности и устойчивости к внешним факторам (пыль, вибрация) остаются ключевыми вызовами. Даже при наличии автоматических систем управления и датчиков большинство устройств всё ещё требуют участия операторов, особенно при корректировке распределения плодов или устранении заторов. К текущим проблемам относятся низкая производительность, неравномерная загрузка контейнеров, сложность обслуживания и крупные габариты оборудования. На основании обзора сделан вывод, что будущие разработки должны фокусироваться на создании компактных, полностью автоматизированных и интеллектуальных систем, способных адаптироваться к различным сортам фруктов и условиям эксплуатации.

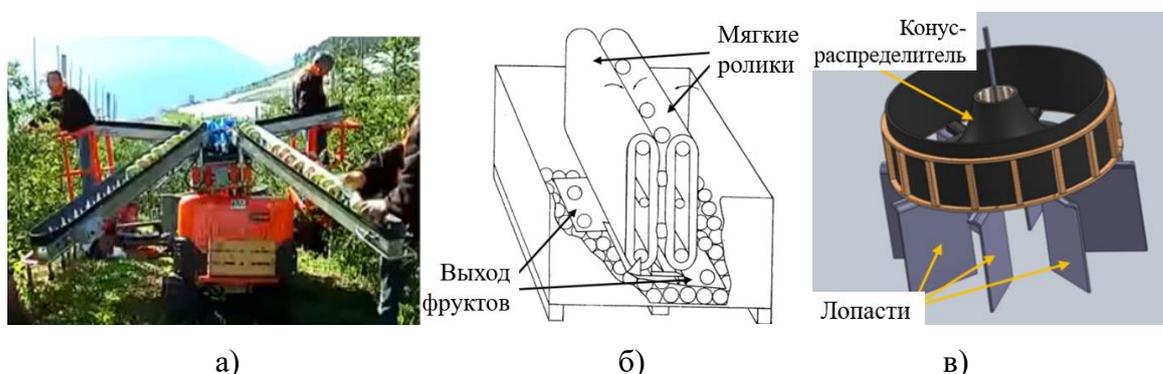


Рисунок 1.5 - Системы для складирования фруктов: а) пластинчатая система складирования; б) система с мягким роликом; в) поворотные лопасти.

Компания «Advanced.farm» предложила использовать 6 роботизированных рук, машинное зрение и присоску для аккуратного сбора яблок без необходимости присутствия человека. Предложенная компоновка позволяет собирать около 30 яблок в минуту, а от идентификации до сбора фруктов проходит всего 1,5 секунды. Благодаря спутниковой навигации в сочетании с техническим зрением система

может точно идентифицировать спелые фрукты на дереве (рисунок 1.6, а, б). Основным недостатком системы с шестью манипуляторами является то, что все манипуляторы закреплены на одном основании, в результате чего сложно обеспечить непрерывную работу для всех манипуляторов одновременно. Высокая цена каждого манипулятора также играет весомую роль при повсеместной эксплуатации устройств для сбора фруктов, так как при поломке прекращают работу все манипуляторы.



а)

б)

Рисунок 1.6 - Робот-сборщик яблок с шестью манипуляторами: а) общий вид системы; б) захватное устройство.

В работе [22] представлен робот для сбора фруктов, который состоит из гусеничного автономного основания, манипулятора последовательной структуры и захватного устройства. Для ускорения процесса сбора фруктов на манипуляторе предусмотрена гофрированная труба, в которую помещаются сорванные плоды (рисунок 1.7). Это техническое решение позволяет значительно ускорить процесс сбора, но при этом осложняет процесс складирования, так как без дополнительных механизмов падающие фрукты повреждаются. Недостатком этого устройства является использование гусениц в качестве ходовой части подвижного основания, это значительно увеличивает энергопотребление системы и снижает горизонтальную устойчивость, при этом достоинство гусеничной ходовой в виде повышенной проходимости не является существенным в условиях промышленных садов.



Рисунок 1.7 - Робот для сбора яблок на гусеничной основе

Одним из перспективных направлений в области автоматизированного сбора урожая является использование беспилотных летательных аппаратов (БПЛА). В частности, израильская компания Tevel разработала систему для сбора яблок, в которой БПЛА оснащён роботизированной рукой [23] (рисунок 1.8, а, б). В процессе работы дрон обнаруживает спелые плоды, подлетает к дереву, захватывает фрукты вращательным движением и переносит их в устройство для сбора. Данное решение позволяет автоматизировать процесс сбора, однако имеет ряд ограничений, связанных со сложностью цикла работы: требуется время на определение положения фрукта, расчёт траектории с учётом особенностей управления БПЛА, а также на складирование плодов. Кроме того, эффективность технологии зависит от погодных условий и качества подрезки деревьев, что предъявляет дополнительные требования к организации сада.



а)



б)

Рисунок 1.8 - РТС для сбора яблок от компании Tevel: а) БПЛА;

б) РТС с манипуляторами в виде БПЛА.

Еще одним направлением в автоматизированном сборе урожая являются малоразмерные мобильные робототехнические системы, работающие в группе и охватывающие небольшие участки. В исследовании [24] предложена тележка с установленным манипулятором и мягким захватным устройством, осуществляющая сбор фруктов на основе данных с камеры технического зрения (рисунок 1.9). После распознавания плодов манипулятор захватывает их и складывает в контейнер. К преимуществам такого подхода относятся простота конструкции и возможность кооперативной работы нескольких устройств. Однако основным недостатком является низкая скорость сбора, поскольку каждый фрукт обрабатывается индивидуально. В отличие от систем с конвейерными или трубчатыми захватами, представленный манипулятор последовательной структуры обладает избыточными степенями свободы, низкой скоростью и высоким энергопотреблением, что ограничивает его эффективность.



Рисунок 1.9 - Малоразмерный РТС для сбора яблок с мягким захватом

Проведенный обзор состояния исследования выявил следующие основные недостатки известных РТС, к которым можно отнести:

1. Манипуляторы с избыточными степенями свободы, в том числе последовательной структуры характеризуются низкой скоростью, высоким энергопотреблением и сложностью управления.

2. Гусеничные платформы увеличивают энергозатраты и снижают устойчивость по сравнению с колесными, особенно в условиях промышленных садов.

3. Эффективность эксплуатации беспилотных летательных аппаратов существенно зависит от погодных условий и своевременной, качественной подрезки кроны деревьев. При отсутствии благоприятных условий и надлежащего ухода за деревьями эффективность применения БПЛА значительно снижается.

4. Системы с несколькими манипуляторами на одном основании имеют высокую стоимость и сложность обеспечения одновременной работы всех манипуляторов.

5. Контейнеры для сбора фруктов часто не адаптированы к полевым условиям и часто допускают повреждения фруктов при ударах.

Компоновка существующих РТС для сбора фруктов может иметь различную эффективность, однако зачастую ключевое значение в скорости и качестве выполнения операции сбора фруктов имеет захватное устройство.

1.1.3 Захватные устройства робототехнических систем для сбора фруктов

Захватные устройства (ЗУ) играют ключевую роль в процессе автоматизированного сбора фруктов, обеспечивая надежную фиксацию плодов и их отделение от растения. Эффективность работы таких устройств напрямую влияет на качество собранного урожая, минимизацию повреждений и общую производительность РТС. В условиях растущего спроса на автоматизацию сельского хозяйства исследования в области разработки оптимальных ЗУ приобретают особую актуальность. Во всем мире активно изучаются перспективные подходы к проектированию захватных устройств, учитывающих биологические особенности различных культур [25]. Например, для сбора яблок проведены детальные исследования, позволившие определить необходимую силу удержания плода и наиболее эффективные способы его отделения от плодоножки [26]. Эти данные легли в основу разработки устройств, адаптированных под

специфику конкретных культур. Таким образом, анализ современных решений в области захватных устройств позволяет выявить ключевые тенденции, технические вызовы и перспективы создания универсальных и высокопроизводительных систем для сбора урожая.

Наиболее часто используемые: механические, пневматические, мягкие, многозвенные и захваты на основе гранул. Механические захватные устройства имеют жёсткую структуру и, как правило, содержат датчики обратной связи, по информации с которых определяется сила сжатия фрукта для предотвращения повреждений [27, 28]. Основным недостатком механических захватных устройств является жёсткость пальцев. Для устранения этого недостатка используют мягкие губки, которые защищают кожуру от повреждений (рисунок 1.10, а, б). Механические захватные устройства менее всего приспособлены для деликатного сбора фруктов и требуют увеличения площади контакта с поверхностью плода.

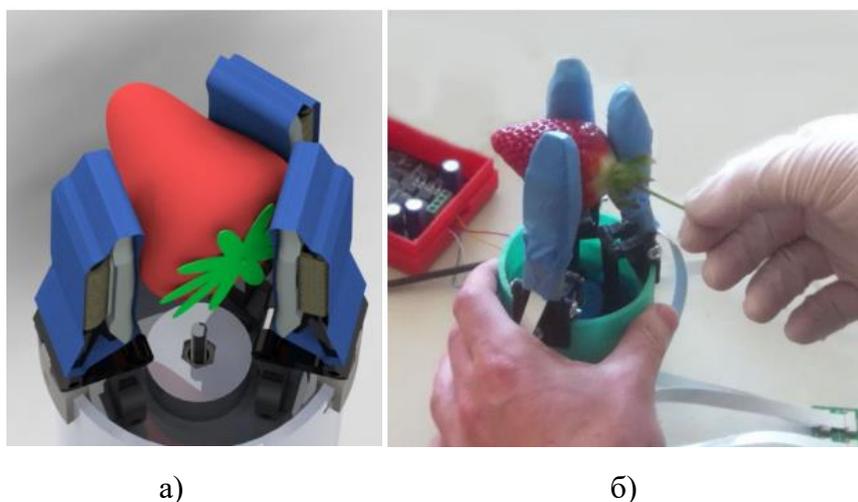


Рисунок 1.10 - Механическое захватное устройство с датчиками обратной связи: а) 3D модель; б) прототип.

Вакуумные захватные устройства имеют простое устройство и чаще всего состоят из присоски (захвата Бернулли) и генератора вакуума, которые соединены вакуумной магистралью. Присоска, контактирующая с поверхностью фрукта, не имеет твёрдых элементов способных повредить фрукт, что является значительным плюсом для сбора урожая без повреждений. Недостатком вакуумных захватных устройств является сложность адаптации к криволинейной поверхности фрукта для

обеспечения герметичного захвата. В работе [29] представлено захватное устройство, которое может осуществлять захват различных объектов криволинейной формы, в том числе и яблок (рисунок 1.11). Недостатком вакуумных захватных устройств является способ отделения фрукта путём отрывания. Так как отламывание и скручивание в значительной степени увеличивают шанс разгерметизации захвата. Этот недостаток можно устранить, однако в этом случае и вырастет энергопотребление компрессора, что в рамках мобильной робототехники недопустимо.

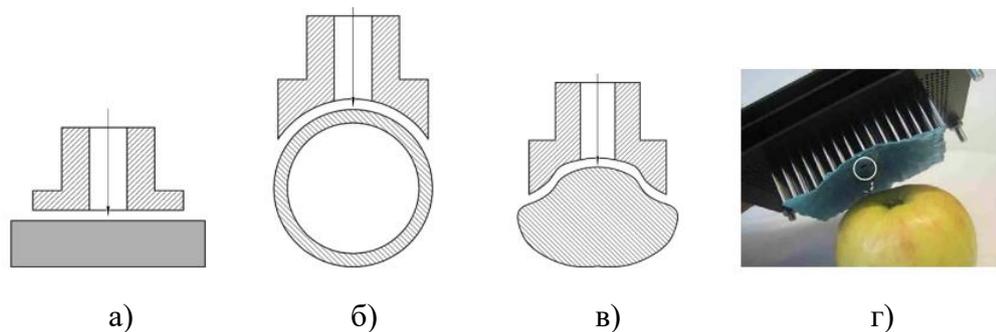


Рисунок 1.11 - Вакуумное захватное устройство для яблок: а) применение на плоской поверхности; б) применение на цилиндрической поверхности; в) применение на криволинейной поверхности; г) применение для захвата фрукта.

Мягкие захваты более эффективны для работы с хрупкими объектами чем механические, но для сбора фруктов требуются специализированные конструкции. Современные технологии многокомпонентной печати и компьютерного проектирования позволяют создавать высокоэффективные решения. В статье [30] представлена структура проектирования, объединяющая генеративный дизайн и оптимизацию топологии (рисунок 1.12, а, б). Исследование сосредоточено на мягких захватах из нескольких материалов, приводимых вакуумом, которые демонстрируют режимы захвата (например, сжатие и заклинивание) без явного программирования (рисунок 1.12, в). Авторы подчеркивают, что предложенный подход, сочетающий генеративный дизайн, оптимизацию и автоматизированную оценку представляет новую тенденцию разработки адаптивных захватов.

В статье [31, 32] предлагается алгоритм оптимизации, который генерирует разнообразные конструкции захватов для эффективного удержания фруктов.

Исследуется пространство проектирования из 28 параметров, включая редко изученные конфигурации расположения пальцев. Для точной оценки характеристик захватов используется моделирование методом конечных элементов (FEM), учитывающее контактные взаимодействия. Предложенный подход позволяет преодолеть разрыв между вычислительным исследованием сложных пространств проектирования и практическим захватом фруктов с использованием простых схем управления, что открывает новые перспективы для развития мягкой робототехники. Основным недостатком мягких захватных устройств является сложность разработки и изготовления. На данный момент нет решений, которые позволили бы собирать фрукты непрерывно. Мягкие захватные устройства уступают в скорости работы, а потребность в компрессоре значительно увеличивает энергопотребление по сравнению с механическими ЗУ.

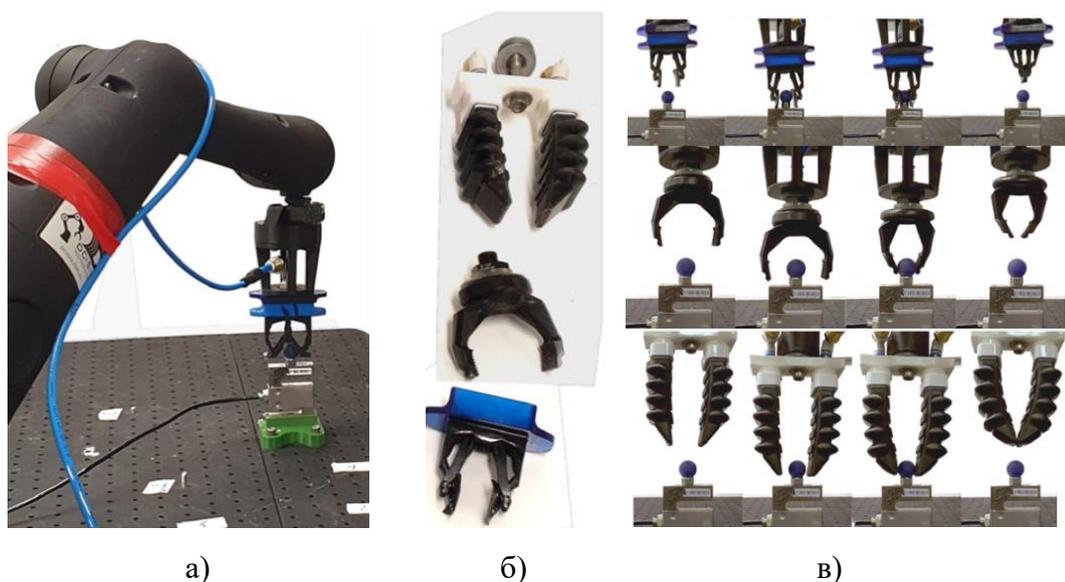
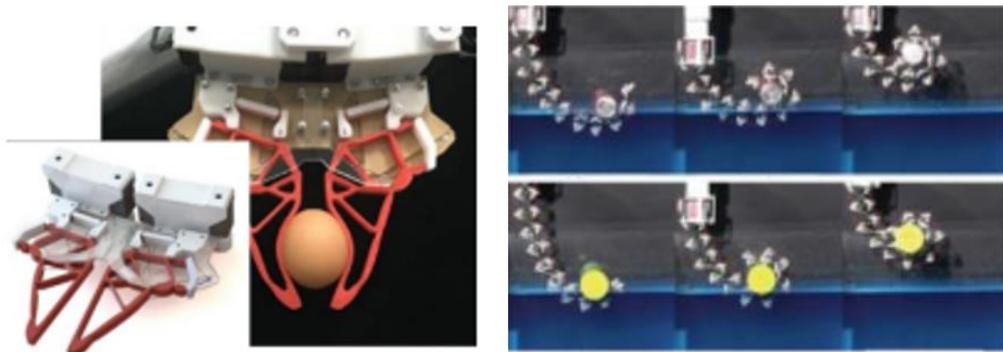


Рисунок 1.12 - Мягкие ЗУ с пневматическим приводом: а) пример использования на манипуляторе; б) виды мягких ЗУ; в) кадры функционирования мягких ЗУ.

Для приведения в движение мягких захватных устройств также используются электроприводы. Такие захватные устройства имеют многозвенную структуру [33], которая может состоять как из гибких сегментов (рисунок 1.13, а), так и комбинироваться с жёсткими сегментами. Этот механизм использует непрерывную структуру для рассеивания и поглощения кинетической энергии быстро движущихся объектов в любом направлении, а также обеспечивает надёжную

фиксацию объекта без необходимости точного позиционирования. Каждый рычаг оснащен проволоочной приводной системой, которая позволяет регулировать как сгибание, так и жесткость. Созданный прототип робота демонстрирует высокую адаптивность механизма: он успешно справляется с захватом объектов разных размеров, форм и под разными углами (рисунок 1.13, б). Предложенный подход доказывает свою универсальность и потенциал для применения в задачах, требующих гибкости и надежности захвата.



а)

б)

Рисунок 1.13. Многозвенные захватные устройства: а) на основе гибких сегментов; б) комбинированные.

Основным недостатком многозвенных ЗУ является сравнительно низкая скорость работы, что делает такие системы менее эффективными для фруктов, требующих быстрого сбора. В текущем состоянии технологий многозвенные захватные устройства не всегда способны обеспечить необходимую производительность, что ограничивает их применимость в условиях массового сбора урожая.

Большинство современных решений основаны на многопальцевых системах, которые, несмотря на свою гибкость, сталкиваются с рядом сложностей: необходимостью управления множеством суставов, измерения силы для предотвращения повреждений объектов и высокими вычислительными затратами на определение усилий захвата. В статьях [34, 35] представлен альтернативный способ захвата объектов, основанный на использовании гранулированного материала. Этот материал при надавливании на объект обтекает его, принимая его форму, а затем под воздействием вакуума быстро затвердевает, обеспечивая

надежное удержание без необходимости наличия обратной связи. Установлено, что изменение объема менее чем на 0,5% достаточно для надежного захвата объектов с усилием, значительно превышающим их вес (рисунок 1.14, а). Для сельского хозяйства, особенно для сбора фруктов, этот тип ЗУ представляет особый интерес. Он может эффективно адаптироваться к различным формам и размерам фруктов (рисунок 1.14, б), таких как яблоки, груши или даже более хрупкие культуры, например, персики или ягоды. Благодаря отсутствию необходимости точной настройки пальцев и минимальному риску повреждения объектов, такие устройства способны существенно повысить производительность и качество сбора урожая. Этот подход демонстрирует значительный потенциал для автоматизации процессов в сельском хозяйстве, особенно там, где требуется бережное обращение с продукцией.

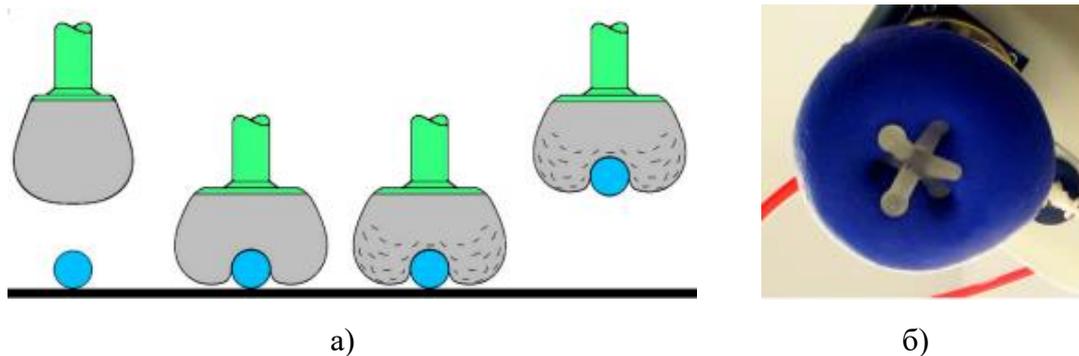


Рисунок 1.14 - Захватное устройство на основе гранул: а) схема применения;
б) прототип ЗУ на основе частиц.

Недостатком захватного устройства на основе гранул является значительные затраты время на перенос в корзину захваченного фрукта, а также захватное устройство такого типа не способно осуществлять захват фруктов снизу-вверх.

Несмотря на многообразие методов захвата фруктов, не менее важно учитывать, каким образом осуществляется отделение плода от растения. Эффективность этого процесса напрямую влияет на качество урожая и минимизацию повреждений. Существуют различные способы отделения, включая вибрационные воздействия, перерезание плодоножки, использование вакуумных систем или мягких манипуляторов. Каждый из них имеет свои преимущества и

ограничения, зависящие от типа фрукта, его прочности и условий сбора. Оптимальный выбор метода отделения должен основываться на биологических особенностях культуры и требуемой скорости сбора, для чего требуется сформулировать требования к РТС для сбора фруктов.

1.2 Анализ состояния исследований процесса сбора яблок

При формулировке требований для разработки робототехнической системы сбора фруктов необходимо учитывать тип фрукта, чтобы обеспечить эффективность, надёжность и безопасность работы. Определение типа собираемого фрукта является ключевым этапом, так как это влияет на конструкцию манипулятора, алгоритмы распознавания и метод отделения плода. В текущей работе в качестве собираемого фрукта выбраны яблоки.

Для сбора яблок существует четыре основных способа отделения плодов от дерева: отрывание, скручивание, отламывание и отрезание. Три из них детально исследованы в статье [26]. С точки зрения прикладываемой силы наиболее эффективным и быстрым методом отделения яблок является отрезание (рисунок 1.15).

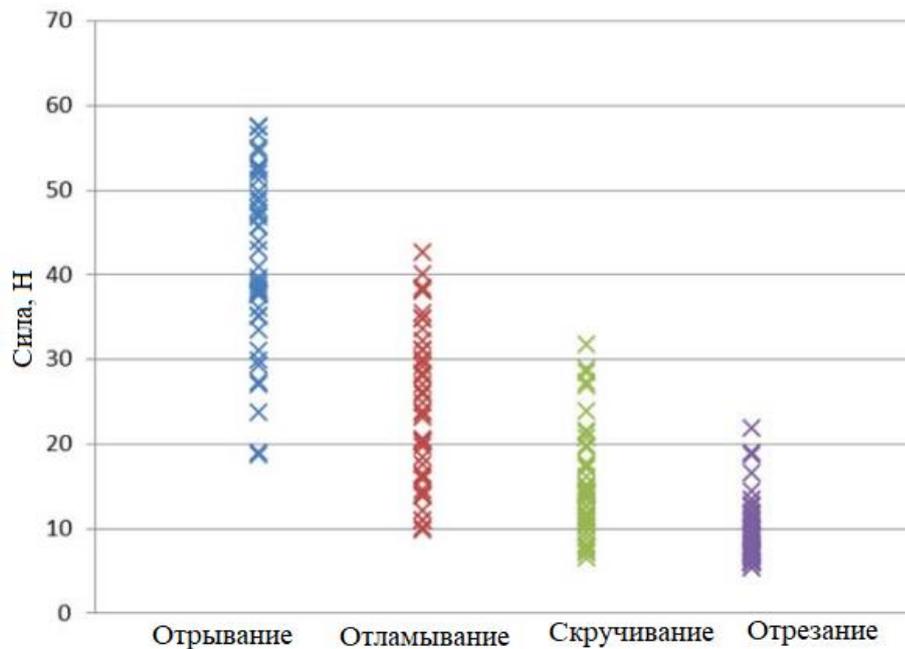


Рисунок 1.15 - График силы, приложенной при отделении фрукта различными способами

Однако этот способ сопряжён с рядом сложностей. Одной из ключевых проблем является необходимость точного определения положения плодоножки, что существенно усложняет процесс управления. Кроме того, острый конец отрезанной плодоножки может повреждать соседние фрукты при их транспортировке и складировании.

Для сбора яблок применяется как массовый сбор, после которого яблоки поступают на переработку так и штучный, после которого яблоки попадают на прилавки. Для формирования требований важно определить допустимый уровень воздействия на яблоко и допустимый уровень повреждений при сборе как собранных, так и ещё висящих яблок. Для этого проведём анализ литературы посвящённой методике сбора яблок.

В исследовании [36] рассмотрена проблема определения площади окисления и изменения цвета кожуры повреждённых яблок с использованием резистивного датчика силы (FRS) на базе Arduino и методов анализа изображений (рисунок 1.16, а). Для эксперимента разработана специальная FRS-перчатка, предназначенная для сбора урожая и измерения силы сжатия яблока во время их захвата. Эта перчатка позволила моделировать процесс сбора яблок и оценивать влияние действующей силы на их состояние (рисунок 1.16, б). После проведения испытаний площадь окисления каждого образца анализировалась с помощью веб-камеры, а данные обрабатывались на платформе Labview (рисунок 1.17, а, б). Программное обеспечение LabView обеспечило точные измерения площадей окисления, что позволило выявить корреляцию между силой сжатия и степенью повреждения кожуры. Результаты показали, что реакция яблок на действующую силу зависит не только от величины приложенной силы, но и от таких факторов, как плотность мякоти и особенности конкретных сортов. Это подчёркивает важность учёта механических свойств плодов при разработке технологий сбора урожая, чтобы минимизировать повреждения и сохранить качество продукции. Полученные данные могут быть полезны для совершенствования роботизированных систем сбора фруктов, где контроль силы захвата является ключевым аспектом.

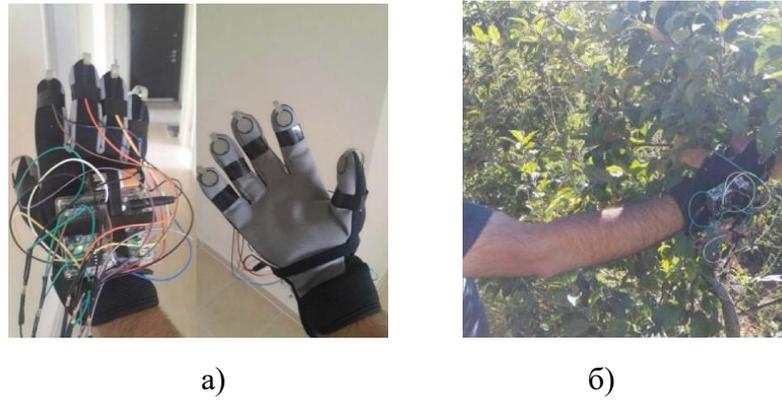


Рисунок 1.16 - Сенсорная перчатка для сбора фруктов: а) общий вид устройства перчатки; б) процесс сбора яблок сенсорной перчаткой.

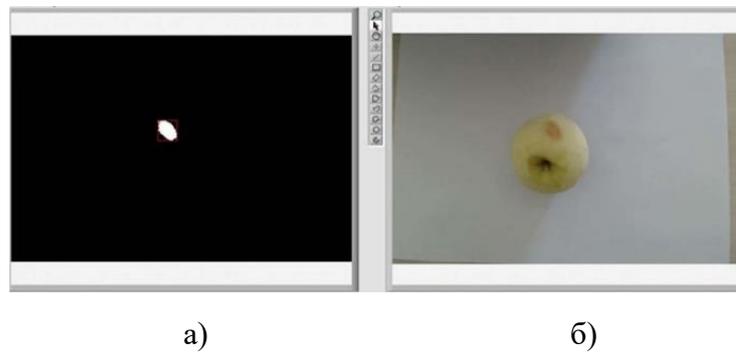


Рисунок 1.17 – Метод анализа повреждений на основе «LabVIEW»: а) выделение повреждений; б) реальное изображение.

В статье [37] исследуются методы ручного сбора яблок по различным траекториям для их применения в роботизированных системах. Традиционный подход требует знания ориентации плода и расположения плодоножки, что сложно реализовать автоматически из-за высоких вычислительных затрат (рисунок 1.18, а, б, в). В исследовании протестированы четыре метода сбора, в независимости от ориентации плода, на пяти сортах яблок в разных условиях выращивания. Для анализа использовались датчики силы и инерциальный измерительный блок. Эксперименты показали, что процесс отделения плодов можно чётко определить. Среднее расстояние для отделения плодов составило 3–7 см, а оптимальный метод сбора определён для каждого сорта с учётом особенностей крепления плодоножки. Результаты помогут разработать более эффективные алгоритмы управления роботизированными системами.

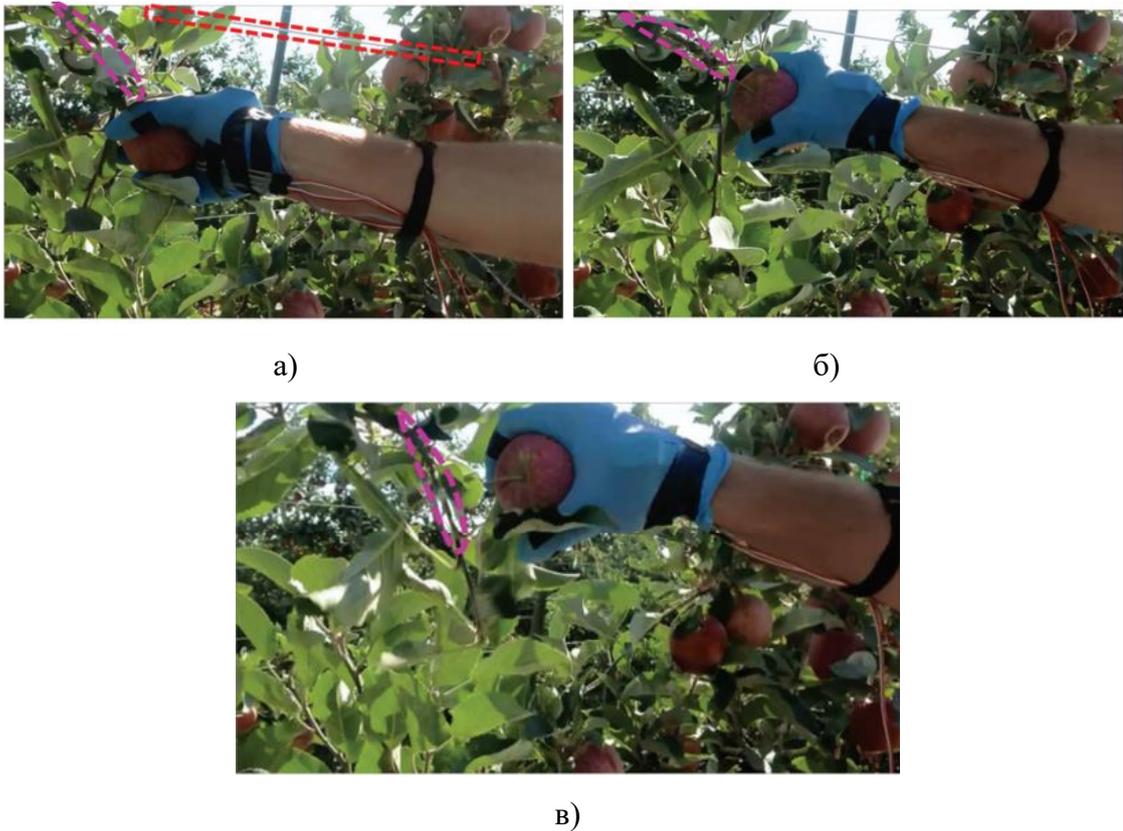
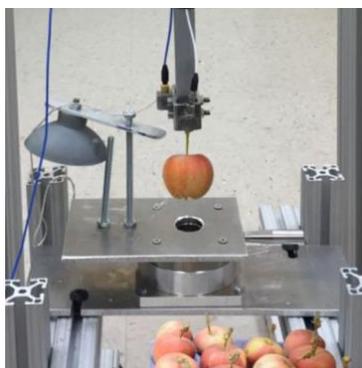


Рисунок 1.18 - Покадровые снимки сбора яблок методом излома: а) начало захвата; б) середина сбора; в) момент отделения фруктов.

Исследование [38] посвящено изучению процесса отделения яблок от дерева при различных способах сбора, включая ручной и три роботизированных метода. Цель работы заключалась в анализе ключевых параметров отделения плодов и определении наиболее эффективного способа сбора с использованием робота с трёхпальцевым захватом (рисунок 1.19, а, б). Особое внимание уделялось минимальному усилию, необходимому для отделения плодов, и уровню повреждений, возникающих при этом. Результаты показали, что ручной сбор создаёт изгибающий момент, снижающий необходимое давление захвата и минимизирующий повреждения плодов. В то же время использование трёхпальцевого захвата в роботизированных системах требует более высокого давления на отдельно взятого пальца, что приводит к большему уровню повреждений яблок. Среди трёх роботизированных алгоритмов один продемонстрировал более высокую производительность, чем остальные. Однако для его внедрения рекомендуется провести дополнительные исследования, чтобы

оценить потенциал и ограничения этого метода в более широком масштабе. Это позволит глубже понять его применимость для разработки эффективных роботизированных систем сбора урожая.



а)



б)

Рисунок 1.19 - Сбор яблок различными методами: а) лабораторный стенд для эксперимента по сбору яблок; б) захват при помощи сенсорной перчатки.

В исследовании [39] рассматриваются особенности процесса сбора яблок для разработки эффективных роботизированных систем. Проанализированы четыре основных метода отделения плодов: горизонтальное вытягивание, вертикальное натяжение, сгибание и скручивание. Установлено, что вертикальное вытягивание является доминирующим фактором в отделении плодов, так как может привести к отрыву плодоножки. Однако для яблок с длинными плодоножками требуется большее смещение и угол наклона, что увеличивает риск неудачного сбора. Для определения оптимальной модели захвата проведено динамическое моделирование, где в качестве параметров оптимизации использовались усилие и время отрыва (рисунок 1.20, а, б). С помощью методики response surface изучено влияние горизонтальной и вертикальной скорости на эти параметры. Результаты экспериментальных исследований показали, что комбинация горизонтального вытягивания с изгибом и скручивающим движением является наиболее перспективной для минимизации усилий и времени при сборе урожая (рисунок 1.21).

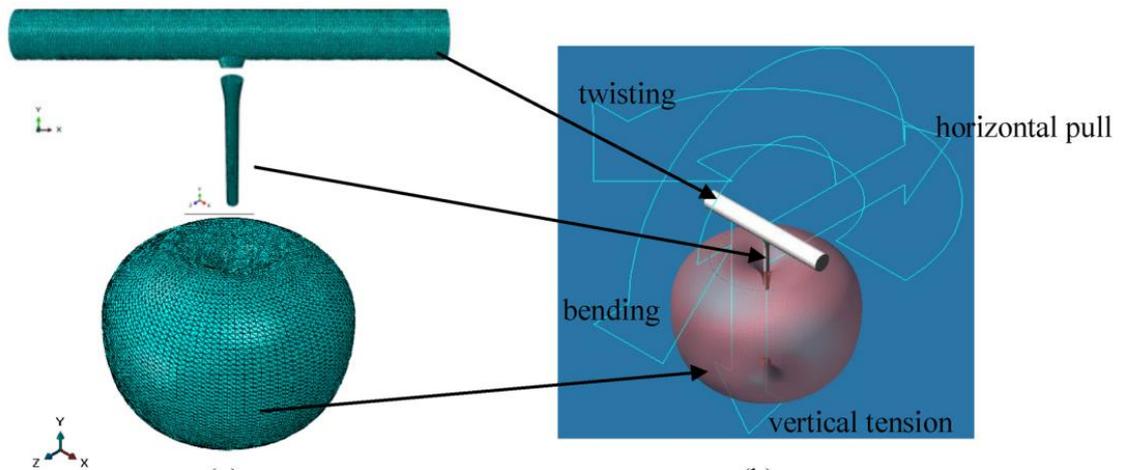


Рисунок 1.20 - Симуляция сбора фрукта: а) конечно-элементные модели ветвей, стебля и фрукта; б) модель в программном обеспечении ADAMS для динамического моделирования.

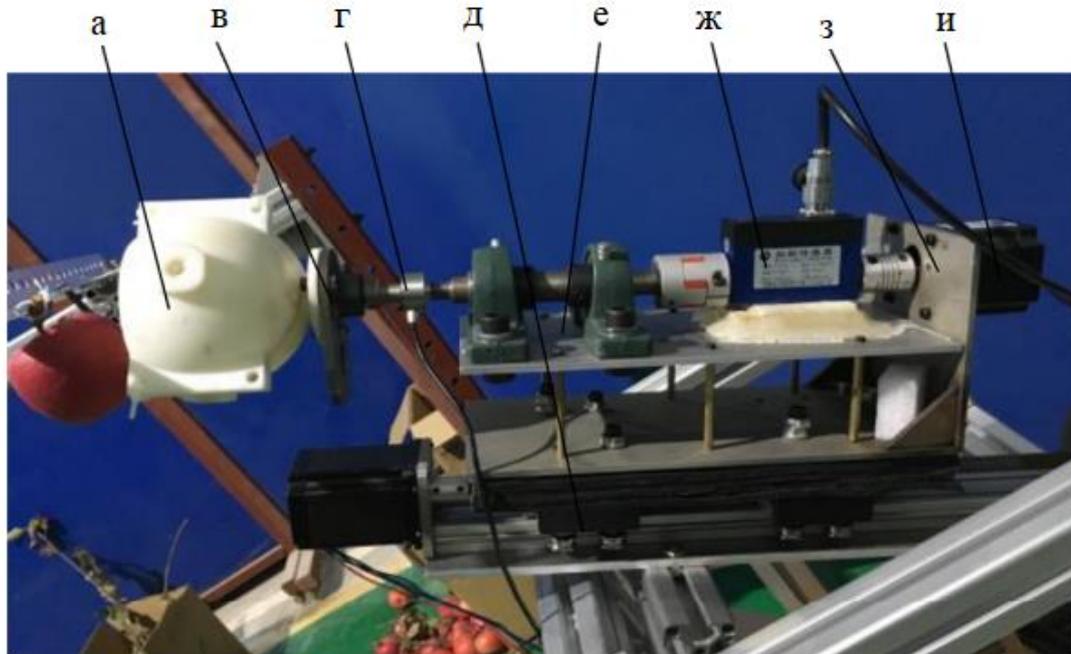


Рисунок 1.21 - Конструкция экспериментального стенда (а - чашеобразный захват; б - зажимная пластина; в - датчик усилия; г - модуль линейного перемещения; приводимый в движение шаговым двигателем; д - главный вал; е - датчик крутящего момента; ж - скользящая рама, установленная на модуле линейного перемещения; з - шаговый двигатель для вращения вала).

В результате проведённого обзора выявлено, что одновременное отрывание и скручивание является наиболее перспективным методом отделения яблок. Для отделения яблока требуется прикладывать силу до 24 ± 6 Н. Сила сжатия яблока при ручном сборе достигает 24 Н.

Существующие аналоги часто демонстрируют недостатки: низкую точность позиционирования манипуляторов, высокое энергопотребление и высокий процент поврежденных плодов. Для разрабатываемой РТС критически важно сформировать требования, которые учтут выявленные слабые стороны, такие как недостаточная адаптивность к изменяющимся условиям или избыточная сила сжатия плода и будут основаны на достоинствах перспективных систем.

1.3 Требования к разрабатываемой робототехнической системе

Для выбора параметров робота, предназначенного для сбора плодов, сформулированы следующие ключевые технические требования:

Автономность и адаптивность: робот должен быть оснащен системой автономного управления, способной анализировать внешнюю среду (например, расположение плодов, состояние дерева и погодные условия) и автоматически корректировать рабочие режимы в зависимости от текущих условий. Это обеспечит эффективное взаимодействие с различными типами деревьев и минимизирует риск повреждения плодов или растений.

Зона захвата и точность позиционирования: конструкция робота должна обеспечивать возможность захвата плодов как на поверхности кроны, так и в глубине до 200 мм. Точность позиционирования звеньев робота относительно целевых объектов должна находиться в пределах 10–20 мм. Такая точность необходима для аккуратного и надежного захвата плодов без их повреждения.

Захватное устройство должно обеспечивать аккуратное и надежное отделение плодов от растения за счет комбинированного воздействия, включающего одновременное скручивание и отрывание. Такой подход позволяет минимизировать механическое воздействие на само растение, что особенно важно для предотвращения повреждений деревьев и сохранения их долгосрочной продуктивности. Метод отделения должен быть универсальным для различных типов плодов.

Конструкция захватного устройства должна включать специальную полость для интеграции гибкой трубы, которая обеспечит быструю и непрерывную

транспортировку плодов после их отделения от растения. Гибкая труба позволит доставлять плоды непосредственно в контейнер для хранения фруктов. Это решение значительно повысит эффективность процесса сбора урожая, минимизируя задержки и потери времени на переукладывание плодов.

Одним из ключевых требований является обеспечение бережного обращения с плодами во время захвата и отделения. Устройство должно быть спроектировано таким образом, чтобы исключить механические повреждения, которые могут снизить товарный вид плодов или их пригодность для длительного хранения. Особенно это актуально для культур с тонкой кожурой, таких как яблоки.

Захватное устройство должно быть адаптировано для работы с плодами сферической формы, такими как яблоки, киви, апельсины и другие подобные культуры. Это требует учета различий в размерах, массе и поверхностных свойствах плодов. Математическая модель устройства должна быть адаптивной к диапазону размеров плодов, обеспечивая масштабируемость.

Рабочая область РТС по вертикали должна охватывать диапазон от 1000 до 3000 мм от уровня почвы, а по горизонтали — до 1500 мм. Такие параметры обеспечивают полный охват плодоносящей части дерева.

Сбор плодов диаметром от 40 мм до 120 мм. Сила, требуемая для отделения плода, составляет 24 Н — значение, учитывающее вариативность прочности. Сила давления каждого пальца захвата устройства 6 Н для предотвращения деформации и повреждения плодов. Материалы контактирующих поверхностей — пищевые полимеры.

Массогабаритные характеристики: для обеспечения мобильности и удобства эксплуатации масса робота не должна превышать 60 кг. При этом устройство должно поддерживать подъем объектов массой не менее 1 кг.

Таким образом, сформулированные требования направлены на создание высокоэффективного и универсального робота, способного работать в условиях промышленного сада с минимальным воздействием на деревья и плоды. Комплексный подход, сочетающий теоретические расчеты, моделирование и

практические испытания, позволит разработать устройство, максимально адаптированное к реальным условиям сельского хозяйства.

1.4 Анализ исследований в области численного моделирования и оптимизации параметров робототехнических систем

Роботы параллельной структуры обладают такими преимуществами, как жесткость, точность и высокое соотношение усилия и веса по сравнению с последовательными роботами [40, 41]. В процессе проектирования роботов параллельной структуры важной задачей является оптимизация конструктивных и геометрических параметров роботов. Различные методы оптимизации позволяют выбрать параметры, которые соответствуют требованиям механизма, и быстро определить рабочее пространство [42-45]. Для оптимизации параметров исследователи применяют методы цифрового проектирования. Так в статье [42] представлен кинематический анализ движения планарного параллельного робота с использованием приложения SolidWorks Motion на основе разработанной 3D-модели. Исследование учитывало ключевые параметры, такие как ограничения движения, свойства материалов, масса, скорости и взаимодействие компонентов. Реальные движения звеньев механизма заменены сопряжениями в SolidWorks Motion, что позволило определить максимальные перемещения звеньев, подтвердить корректность разработанных подвижных частей и отсутствие конфликтующих связей. Анализ показал, что траектории перемещения модулей подходят для выполнения поставленных задач. Для оптимизации применена метрика Чебышева, которая позволила минимизировать время позиционирования. Кроме того, рассмотрен алгоритм построения гиперповерхности ограничения, обеспечивающий безопасное расстояние траектории движения от границ рабочей области. Результаты математического моделирования подтвердили эффективность предложенных подходов.

В работе [43] авторы выполняют оптимизацию параллельного робота с тросовым приводом для перемещения груза в особых условиях. Конструкция включает жёсткую раму, тросы и подвижную платформу с устройством осевого

перемещения. Разработаны численные методы и алгоритмы для анализа влияния натяжения тросов на рабочую зону и определения минимальных размеров робота. Создан программный комплекс для экспорта 3D-моделей рабочих областей в формате STL, что возможно использовать на последующих стадиях проектирования. Исследования рабочей области механизмов открывают широкие возможности для оптимизации. В статье [44] исследуются методы преобразования рабочего пространства параллельного 3-РТР робота для снижения объёма хранимых данных и упрощения визуализации. Предложены два подхода, основанных на повторной декомпозиции рабочей области с сохранением качества аппроксимации. Первый подход использует алгоритм поиска в глубину для двудольного графа с целью нахождения минимального охвата прямоугольников для каждой внешней грани рабочей области. Второй подход применяет триангуляцию Делоне с ограничениями для тех же граней. Проведена сравнительная оценка эффективности обоих методов, подчёркивающая их преимущества и особенности применения.

Алгоритмы оптимизации также применяются для оптимизации конструктивных параметров. Формализация принимаемых решений при проектировании может внести существенный вклад для создания оптимизированной РТС для сбора фруктов. В статье [45] описывается методология оптимизации конструктивных параметров мобильной платформы 6-DOF для тренажёров водителей с использованием динамической имитационной модели. Оптимизация позволила снизить силовые реакции в шарнирах при выполнении различных траекторий перемещения с учётом полезных нагрузок.

Методы оптимизации делятся на две основные категории: эвристические и детерминированные.

Эвристические методы основываются на эмпирических знаниях и не всегда гарантируют нахождение глобального оптимума. Однако они могут быть очень эффективными в задачах с большим количеством переменных и сложными ограничениями. Например, метод случайного поиска предполагает генерацию

случайных точек в пространстве решений и выбор наилучшего. Хотя этот метод подходит для задач с небольшим числом переменных, он требует значительных вычислительных ресурсов в более сложных случаях.

Детерминированные методы оптимизации используют фиксированные алгоритмы без элементов случайности. Такие методы применяют точную информацию о функции, их результаты всегда воспроизводимы при одинаковых начальных условиях. Детерминированная оптимизация подходит для решения задач с чёткими функциональными возможностями, которые помогают вычислить глобально оптимальный результат. Однако у детерминированных алгоритмов могут возникнуть проблемы с решением задач «чёрного ящика» или чрезвычайно сложных и изменчивых функций оптимизации.

В области эвристических методов оптимизации широкое распространение получили генетические алгоритмы, вдохновлённые процессами биологической эволюции. Они позволяют получить оптимальные решения или приближённые решения в пространствах большой размерности. Результаты работ [46, 47] позволяют заключить, что генетические алгоритмы позволяют быстро находить приближённые решения высокого качества и могут быть использованы для решения прикладных задач. Недостатком генетического алгоритма является то, что его хромосомы кодируются дискретно. Соответственно, его нельзя использовать для решения непрерывных и бесконечных задач, а также генетические алгоритмы не гарантируют нахождение оптимального решения и вместо этого находят локальный экстремум вместо глобального.

В качестве альтернативы может использоваться жадный алгоритм оптимизации. Жадный алгоритм является однокритериальным, поэтому он неприменим для многомерных задач. Однако для непрерывных задач этот алгоритм находит наиболее оптимальное решение. В статье [48] исследуется поведение жадных алгоритмов для минимизационной задачи о ранце с произвольными распределениями коэффициентов. Результаты исследования имеют теоретическую значимость для развития методов дискретной оптимизации и могут найти

применение при решении практических задач распределения вычислительных ресурсов.

Меметические алгоритмы (МА) представляют собой гибридные методы поиска оптимальных решений, сочетающие глобальный поиск (например, эволюционные алгоритмы) с локальным поиском для решения задач оптимизации [49]. Они применяются в широком спектре задач, включая дискретную и непрерывную оптимизацию, задачи с ограничениями, многоцелевую оптимизацию и задачи в условиях неопределённостей. МА нашли успешное применение в инженерии, биоинформатике и других областях, где требуется решение сложных оптимизационных задач. К достоинствам МА можно отнести самоадаптивные и коэволюционные механизмы, благодаря которым МА могут автоматически настраивать свои параметры и критерии для конкретной задачи. В сравнении с традиционными генетическими алгоритмами и жадными алгоритмами, МА демонстрируют улучшенные результаты за счет комбинирования глобального и локального поиска. Недостатками МА в сравнении с жадными алгоритмами, которые просты в реализации, МА требуют более сложной структуры и настройки, а также производительность МА может зависеть от правильного выбора параметров, таких как баланс между глобальным и локальным поиском. МА, как и генетические алгоритмы эффективны в оптимизации конструктивных параметров механизмов. Таким образом, меметические алгоритмы предлагают новую парадигму в решении сложных задач оптимизации, сочетая гибкость, универсальность и высокую производительность.

Детерминированные методы, например, метод ветвей и границ разбивает пространство решений на подпространства и ищет оптимум в каждом из них. Этот метод эффективен при малом числе переменных, но требует больших вычислительных ресурсов для более сложных задач. Исследования в этой области сосредоточены на улучшении методов для дискретных задач. Так в работе [50] рассматривается графическая реализация метода динамического

программирования, который решает задачи большей размерности чем известных алгоритмов ветвей и границ, которая представлена в работе [51].

В работе [52] представлена унифицированная эвристика для решения пяти различных вариантов задачи маршрутизации транспортных средств. Все варианты задач преобразуются в единую модель поиска и доставки, которая решается с использованием адаптивной системы поиска по большой окрестности (ALNS). Алгоритм показал многообещающие результаты, улучшив 183 из 486 известных решений в тестовых задачах. Таким образом, представленный подход демонстрирует значительный потенциал для практического использования в логистике при сборе фруктов.

Другим известным методом оптимизации является метод поиска по симплексу, предложенный У. Спенли, Г.Р. Хекстом и Ф.Р. Химсвортом в 1962 году. Этот метод представляет собой численную процедуру, направленную на поиск оптимального решения в вершинах многогранника (симплекса), который моделирует пространство решений задачи. Основная идея заключается в последовательной трансформации симплекса за счёт отражения, растяжения и сжатия его вершин для постепенного приближения к оптимуму. Метод особенно эффективен для задач с небольшим числом переменных благодаря своей простоте и надежности. Однако для сложных задач с большим количеством параметров или нелинейными зависимостями метод требует значительных вычислительных ресурсов, что ограничивает его применимость [53].

Особое место в современной теории оптимизации занимают исследования в области многокритериальной оптимизации. Эта область активно развивается, так как позволяет находить решения, учитывающие сразу несколько целевых функций, что особенно важно для реальных задач, где необходимо балансировать различные противоречивые критерии. Значительный вклад в развитие этой области внесли такие учёные, как И.Х. Сигал, Д.И. Коган, Ю.Г. Евтушенко, М.А. Потапов и другие. Их работы охватывают широкий спектр вопросов: от теоретических исследований дискретных многокритериальных задач до разработки новых численных методов

их решения. Особое внимание уделяется применению методов Парето-оптимизации, которые позволяют находить множество компромиссных решений, каждое из которых нельзя улучшить по одному критерию, не ухудшив другой [54–58].

Применение методов многокритериальной оптимизации особенно актуально в таких областях, как проектирование технических систем, робототехника, управление производственными процессами и логистика. Например, в задачах проектирования роботизированных систем часто требуется одновременно минимизировать энергозатраты, время выполнения операций и массу конструкции, что приводит к необходимости использования многокритериальных подходов. В таких случаях методы Парето позволяют получить набор альтернативных решений, из которого инженеры могут выбрать наиболее подходящий вариант, исходя из дополнительных практических соображений [59, 60].

Таким образом, метод поиска по симплексу и многокритериальная оптимизация представляют собой важные направления в теории оптимизации, дополняя друг друга своими возможностями и областями применения. Первый метод демонстрирует высокую эффективность для маломерных задач, тогда как второе направление открывает возможности для решения сложных многоцелевых задач, характерных для современных технологий и научных исследований. Эти исследования позволяют не только повысить эффективность и экономичность механизмов, но и минимизировать энергозатраты, что особенно актуально для робототехники и промышленного оборудования [61–63]. Это особенно заметно в разработке тренажёрных комплексов и роботизированных платформ, где точность и производительность напрямую зависят от выбранных параметров [64, 43]. Такие исследования открывают новые горизонты для проектирования механических систем, делая их более надёжными, точными и адаптивными к различным условиям эксплуатации.

Многие научные работы сосредоточены на анализе вычислительной сложности алгоритмов оптимизации и поиске путей ее снижения. Современные

исследования подчеркивают необходимость разработки новых методов, которые обеспечивают высокую производительность без ущерба для точности решений. Это направление становится особенно значимым в условиях роста объемов данных и усложнения задач, требующих быстрого и качественного анализа. Успехи в этой области позволяют создавать более совершенные алгоритмы, применимые как в теоретических исследованиях, так и в практических приложениях [65–71]. Таким образом, развитие методов глобальной и многокритериальной оптимизации продолжает оставаться одним из ключевых факторов технологического прогресса.

1.5 Комбинированный метод проектирования робототехнической системы для сбора фруктов

Процесс проектирования РТС для сбора фруктов представляет собой сложную, многоэтапную задачу, требующую тщательного анализа и системного подхода. В условиях растущего спроса на автоматизацию сельскохозяйственных процессов разработка таких систем становится особенно актуальной. Однако успешное создание РТС возможно только при условии четкой формализации каждого этапа проектирования, что позволяет минимизировать ошибки, сократить временные и финансовые затраты, а также обеспечить высокое качество конечного продукта. Формализация процесса подразумевает, что каждое решение принимается на основе логически обоснованных критериев, таких как аналитический обзор, выбор оптимальной структуры и принципов работы, математическое моделирование, оптимизация параметров и имитационное моделирование процессов в цифровой среде. На рисунке 1.22 представлена блок-схема комбинированного метода проектирования робототехнической системы для сбора фруктов, включающий формализацию всех стадий проектирования.

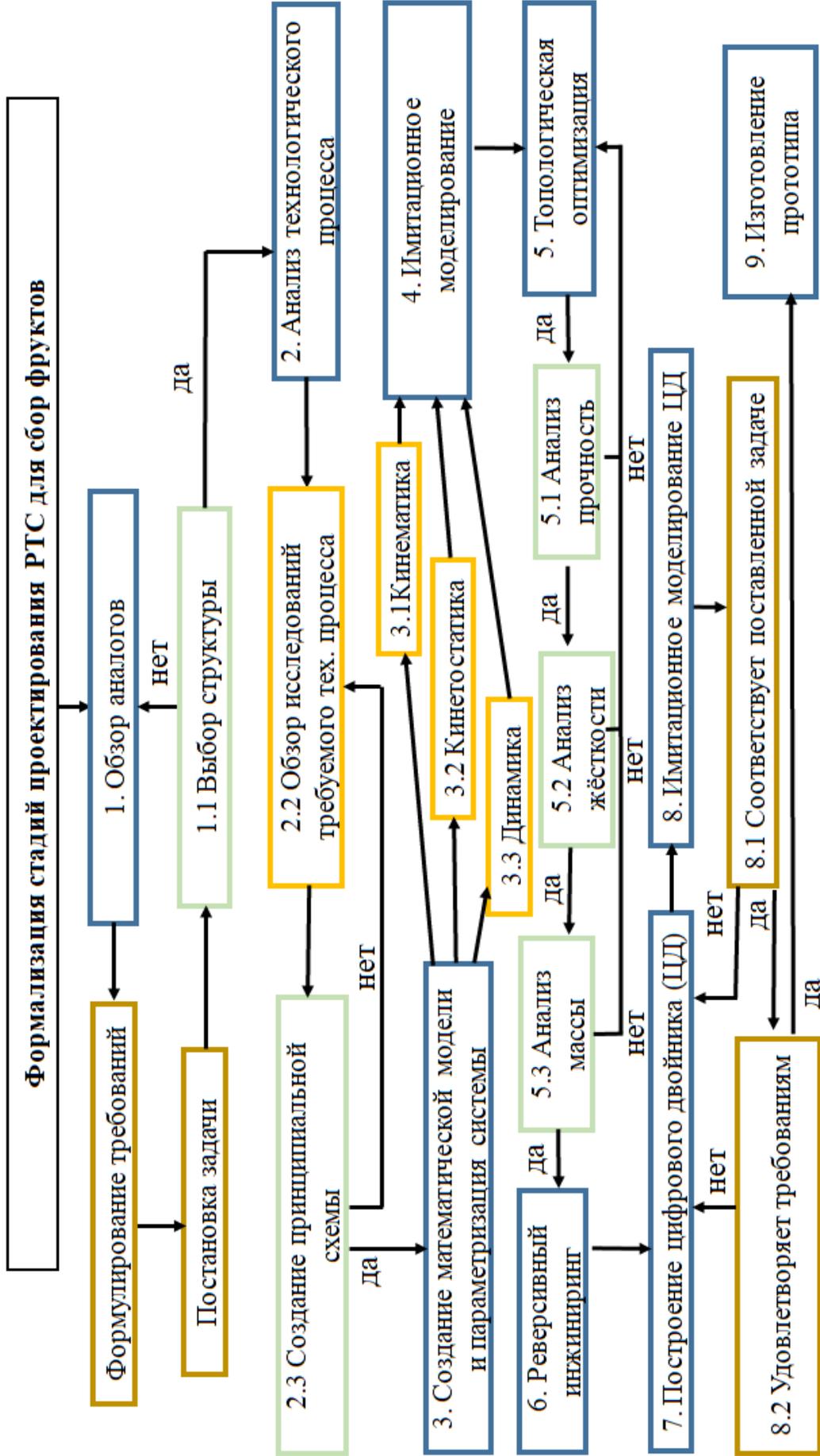


Рисунок 1.22 - Блок-схема комбинированного метода проектирования

Представленный на схеме подход не только повышает надежность и эффективность разработки, но и позволяет учитывать специфику задачи, например, особенности взаимодействия с фруктами, условия эксплуатации и требования к производительности. Последовательные этапы проектирования, которые тесно взаимосвязаны обеспечивают возможность корректировки на каждом шаге, что особенно важно в условиях неопределенности или изменяющихся требований. Например, на этапе выбора метода отделения плодов от дерева необходимо учитывать механические свойства фруктов, чтобы минимизировать повреждения. На этапе оптимизации конструкции деталей требуется баланс между массой, жесткостью и технологичностью изготовления. Таким образом, формализованный и поэтапный подход к проектированию РТС позволяет создавать системы, максимально адаптированные к решению конкретных задач, и является основой для успешного создания прототипа.

Этапы проектирования:

1. Обзор аналогов и выбор структуры РТС.

1.1. Логика принятия решения при выборе структуры РТС основывается на строгом анализе существующих решений. Проводится сравнительный анализ достоинств и недостатков аналогичных систем, что позволяет выявить их ключевые особенности. При этом учитываются такие параметры, как производительность, точность, энергоэффективность, сложность управления и стоимость.

1.2. В результате анализа выбирается базовая структура РТС, на основе сформулированных требований и поставленной задачи — например, робот с несколькими степенями свободы или мобильная платформа с захватным устройством, которая наилучшим образом соответствует цели и обеспечивает оптимальное соотношение характеристик для конкретного применения.

2. Выбор принципа работы захватного устройства.

На основе анализа механических свойств плодов, таких как плотность мякоти и прочность плодоножки, а также требований к качеству сбора (минимизация

повреждений), определяется наиболее эффективный метод отделения плодов. Рассматриваются различные варианты, такие как отрезание, скручивание или вытягивание, с учётом их применимости и влияния на целостность плодов. Формирование принципиальной схемы захватного устройства, включающей его основные элементы, такие как пальцы, ножницы или вакуумный механизм, которые обеспечивают реализацию выбранного метода отделения плодов наиболее качественно.

3. Создание математической модели и параметризация системы

На основе выбранной структуры создается математическая модель, описывающая кинематику, кинетостатику и взаимодействие с окружающей средой. Параметры модели (например, длины звеньев, углы поворота, силы захвата) выбираются на основе оптимизации системы. Если параметры можно определить аналитически, и он не имеет вариативности, используются стандартные методы проектирования. Если влияние параметров неочевидно, применяются методы многокритериальной оптимизации, например, метод Парето, чтобы найти компромисс между противоречивыми критериями и выбрать наиболее предпочтительный вариант.

4. Имитационное моделирование и корректировка кинематической схемы

Проводится имитационное моделирование, которое позволяет детально проанализировать предельные действующие силы в сочленениях роботизированной системы. Этот процесс включает в себя создание виртуальной модели системы, где воспроизводятся реальные условия эксплуатации, такие как взаимодействие с объектами, внешние нагрузки и динамические воздействия [72-74]. Моделирование выполняется с использованием специализированных программных инструментов, таких как Adams, что позволяет получить точные данные о распределении сил, моментах и деформациях в критических точках конструкции. На основе результатов моделирования проводится корректировка кинематической схемы системы. Это может включать изменение длин звеньев, углов поворота, расположения сочленений или других параметров, влияющих на

работоспособность и эффективность системы. Например, если выявлено, что определённые сочленения испытывают чрезмерные нагрузки, могут быть увеличены их размеры или изменена геометрия для снижения напряжений. В случае обнаружения проблем с маневренностью или точностью позиционирования выполняется адаптация кинематических параметров для улучшения этих характеристик. Уточненная кинематическая схема учитывает не только теоретические расчёты, но и реальные нагрузки, которые возникают в процессе работы системы. Это обеспечивает более высокую надежность и долговечность конструкции, а также минимизирует вероятность выхода из строя элементов механизма. Кроме того, учет ограничений, таких как физические возможности приводов, максимально допустимые углы поворота и требования к точности, делает систему более адаптированной к практическому применению. В результате получается оптимизированная кинематическая схема, которая служит основой для дальнейших этапов проектирования, включая топологическую оптимизацию и изготовление прототипа.

5. Топологическая оптимизация конструкции

На основе анализа предельных сил, действующих в сочленениях робототехнической системы, проводится топологическая оптимизация деталей. Этот процесс направлен на улучшение конструктивных характеристик, таких как минимизация массы при сохранении требуемой жесткости и прочности [75-79]. Топологическая оптимизация позволяет определить наиболее эффективное распределение материала внутри заданной области проектирования, что особенно важно для создания высокопроизводительных и энергоэффективных систем. Оптимизация основана на численных методах, таких как метод конечных элементов (МКЭ), которые позволяют моделировать напряженно-деформированное состояние деталей под воздействием внешних нагрузок. В процессе оптимизации учитываются критические параметры, такие как максимальные напряжения, деформации и запасы прочности, чтобы гарантировать надежность конструкции в реальных условиях эксплуатации.

Процесс может быть полностью автоматизирован, что значительно сокращает время разработки и повышает точность расчетов. Автоматизация позволяет пересчитывать параметры и перестраивать геометрию деталей при изменении базовой модели или условий эксплуатации. Например, если изменяются нагрузки, действующие на систему, или если вносятся корректировки в кинематическую схему, программа автоматически адаптирует форму и размеры деталей, сохраняя их функциональность и соответствие заданным требованиям. В результате топологической оптимизации получаются сложные геометрические формы, которые часто невозможно создать традиционными методами производства [80-81]. Эти формы характеризуются органичностью и минимальным использованием материала, что делает их оптимальными для современных технологий изготовления, таких как 3D-печать или аддитивное производство. Оптимизированные геометрические формы деталей не только снижают массу конструкции, но и улучшают ее динамические характеристики, такие как устойчивость, маневренность и энергоэффективность. Это особенно важно для робототехнических систем, где каждый лишний грамм веса может существенно повлиять на производительность и автономность устройства. Таким образом, топологическая оптимизация становится ключевым этапом в проектировании современных роботизированных систем, обеспечивая баланс между минимальной массой, высокой прочностью и технологичностью изготовления.

6. Реверсивный инжиниринг

Сложные формы, полученные в результате топологической оптимизации, часто характеризуются органичностью и минимальным использованием материала, что делает их оптимальными с точки зрения функциональности и эффективности. Однако такие формы могут быть крайне сложными для изготовления традиционными методами производства. Поэтому следующим важным этапом является адаптация этих форм для дальнейшего изготовления, что предполагает их упрощение с учетом технологических ограничений. На этом этапе учитываются особенности различных методов производства, таких как 3D-печать,

фрезерование или литье под давлением. Например, при использовании 3D-печати необходимо учитывать ограничения по углам наклона элементов и минимальной толщине стенок, чтобы избежать деформаций или обрушений во время печати. При фрезеровании важно обеспечить доступ инструмента ко всем участкам детали, что может потребовать изменения геометрии для исключения недоступных зон. В случае литья под давлением учитывается необходимость создания литниковых каналов и обеспечение равномерного заполнения формы расплавленным материалом. Этот этап играет критически важную роль в успешном переходе от цифровой модели к реальному изделию, гарантируя, что экспериментальный образец РТС будет соответствовать всем сформулированным требованиям.

7. Построение цифрового двойника

Все детали, полученные на предыдущих этапах проектирования, собираются в единый цифровой двойник РТС. Цифровой двойник представляет собой виртуальную модель, которая максимально точно воспроизводит геометрию, кинематику, динамику и другие ключевые характеристики реального объекта. Для этого используются параметры, которые определены и оптимизированы на этапах математического моделирования, топологической оптимизации и симуляции в виртуальной среде. Параметры цифрового двойника, такие как масса деталей, их жесткость, моменты инерции, коэффициенты трения и другие физические свойства, максимально приближены к характеристикам реального объекта. Это достигается за счет применения высокоточных методов расчета и учета всех возможных факторов, влияющих на работу системы. Такая детализация позволяет проводить точное моделирование поведения РТС в различных условиях эксплуатации, включая взаимодействие с окружающей средой, нагрузки на сочленения и динамические воздействия.

Полностью собранный цифровой двойник системы становится основой для дальнейшего анализа и оптимизации, а также служит инструментом для обучения операторов и тестирования новых алгоритмов управления до этапа физического изготовления прототипа.

8. Имитационное моделирование движений цифрового двойника

Проводится имитационное моделирование движений цифрового двойника РТС для всесторонней проверки её функциональных характеристик. В процессе моделирования анализируются пределы подвижности каждого сочленения, точность позиционирования и захвата объектов, а также распределение действующих сил в критических точках конструкции. Это позволяет оценить, насколько система способна справляться с поставленными задачами в условиях, максимально приближенных к реальным. Особое внимание уделяется выявлению потенциальных недостатков системы, таких как коллизии между элементами конструкции, перегрузки в сочленениях, ограничения по скорости выполнения операций или недостаточная точность работа. На основе результатов моделирования выполняется корректировка параметров системы. Например, могут быть изменены длины звеньев, углы поворота, расположение узлов или характеристики приводов для устранения выявленных проблем. Если обнаруживаются коллизии, проводится оптимизация траекторий движения или геометрии деталей. При возникновении перегрузок в сочленениях может потребоваться усиление конструкции или снижение нагрузок путем изменения алгоритмов управления. Ограничения по скорости устраняются за счет оптимизации динамических параметров системы, таких как моменты инерции или коэффициенты трения. Кроме того, моделирование позволяет протестировать различные сценарии работы системы, включая экстремальные условия, что помогает повысить её надежность и долговечность. После внесения корректировок проводятся повторные тесты для проверки эффективности внесенных изменений. Таким образом, данный этап является ключевым для выявления и устранения слабых мест системы до перехода к физическому прототипированию, что значительно снижает риски и затраты на последующих этапах разработки.

9. Изготовление прототипа

После устранения всех выявленных недостатков на этапе моделирования, который включает в себя как кинематическое, так и динамическое тестирование

цифрового двойника, приступают к изготовлению реального прототипа РТС. Этот этап является ключевым для проверки работоспособности системы в реальных условиях. При изготовлении прототипа учитываются результаты всех предыдущих этапов проектирования: от выбора базовой структуры и оптимизации параметров до топологической оптимизации деталей и имитационного моделирования. Это позволяет минимизировать риск ошибок, связанных с несоответствием расчетных и фактических характеристик, а также обеспечивает высокую степень готовности системы к выполнению поставленных задач.

Представленный подход позволяет формализовать процесс проектирования, обеспечивая логическую последовательность действий на каждом этапе. Благодаря использованию математического моделирования, многокритериальной оптимизации, топологической оптимизации и имитационного моделирования удастся создать высокоэффективную и надежную систему. Такой метод снижает затраты времени и ресурсов на разработку, а также минимизирует вероятность ошибок при переходе от проектирования к изготовлению.

Выводы по главе

1. Анализ состояния исследований показал, что создание и применение новых типов робототехнических систем для сбора фруктов с высокими показателями структурной жёсткости, автономности, компактности и скорости является важным направлением сбора урожая. Эти системы не являются универсальными, и многое зависит от типа и геометрии собираемого фрукта.

2. Важным элементом робототехнической системы для сбора фруктов является захватное устройство, которое не должно повреждать фрукты и обеспечивать высокую скорость сбора урожая. Наиболее перспективным методом сбора фруктов является одновременное отрывание и скручивание фрукта, что должно быть учтено при проектировании захватного устройства.

3. Обзор методов оптимизации показывает, что они делятся на эвристические и детерминированные. Эвристические методы эффективны для сложных задач с множеством переменных, но не гарантируют глобальный оптимум.

Детерминированные методы обеспечивают точные решения, но требуют значительных ресурсов для сложных задач. Для задач планирования траектории и навигации лучше применять методы многокритериальной оптимизации, такие как Парето-оптимизация, позволяющие находить компромиссные решения с учетом нескольких противоречивых критериев.

4. Предложен комбинированный метод проектирования робототехнической системы для сбора фруктов, включающий формализацию всех стадий проектирования: создание математических и электронно-цифровых моделей, цифровых двойников, топологическую оптимизацию распределения материала в конструктивных элементах с использованием CAD/CAE-систем, позволяющий на основе полной динамической и имитационной моделей получить детализированный цифровой двойник робототехнической системы и ее рациональную конструкцию.

ГЛАВА 2. МАТЕМАТИЧЕСКОЕ МОДЕЛИРОВАНИЕ РОБОТОТЕХНИЧЕСКОЙ СИСТЕМЫ

В главе рассмотрена структура и математическая модель робота-трипода с пассивной кинематической цепью и захватного устройства в соответствии с сформулированными требованиями. На основе математической модели определено рабочее пространство с учётом пересечений звеньев и особых положений. Для выбора оптимальных параметров выполнена многокритериальная оптимизация параметров робота-трипода с учётом важности критериев. Представлен разработанный алгоритм многокритериальной оптимизации параметров робота-трипода с учётом критериев компактности и допустимых ориентаций рабочей платформы с переменными коэффициентами важности.

2.1 Математическая модель кинематики робота-трипода с центральной пассивной кинематической цепью и телескопическим звеном

Разработка автономных робототехнических систем требует тщательного подхода к выбору конструктивных решений, учитывающих сложность окружающей среды и геометрию фруктов. Формализация стадий проектирования позволит снизить субъективность проведенного расчета. Формализация отдельных стадий проектирования систем для сбора фруктов уже применялась в работах [82-91]. Предлагаемая методология использует количественную оценку целевых функций, а также методы поиска по Парето для поиска оптимальных параметров конструкций. При этом в работе [83-86] описано, что роботы параллельной структуры имеют более высокую жёсткость по сравнению с роботами последовательной структуры, что позволят им добиться повышенной точности в условиях неструктурированной среды. Также на основе проведенного обзора определено, что наиболее предпочтительно для перемещения робота использовать колесное основание с относительно небольшой колесной базой в виду высокой

энергоэффективности. Таким образом, на основе наиболее перспективных решений предложена схема РТС для сбора фруктов, 3D-модель которой представлена на рисунке 2.1, а. В состав системы входит мобильная колёсная платформа 1 с корзиной для сбора фруктов 2. На колёсной платформе установлен робот-трипод 3 с телескопическим звеном 4, которое обеспечивает досягаемость и захват фруктов на большой высоте с помощью установленного на нем захватного устройства 5. Робот-трипод включает три приводные кинематические цепи 6 типа RRPS с центральной кинематической цепью 7, каждая из которых сопряжены с неподвижной 9 и подвижной 10 платформами через сферический шарнир 8 (рис. 2.1, б). Таким образом, за счёт изменения длин l_i штанг A_iB_i обеспечивается требуемый поворот подвижной платформы по всем осям относительно центра шарнира С. Неподвижная платформа $A_1A_2A_3$ и подвижная платформа $B_1B_2B_3$ механизма представляют собой правильные треугольники с радиусами R_1 и R_2 соответственно.

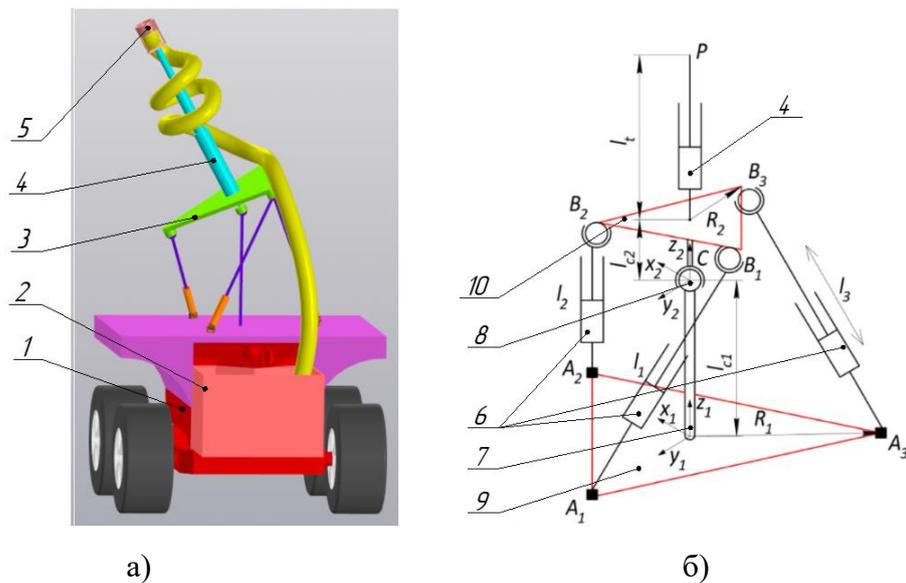


Рисунок 2.1 - Автономная РТС для сбора фруктов: а) 3D модель; б) расчётная схема робота-трипода.

Геометрические параметры разрабатываемой системы должны обеспечивать достижимость фруктов в заданной области кроны деревьев. Под достижимостью понимается, что для любой произвольной точки в пределах заданной области кроны существует хотя бы одно значение обобщенных координат робота-трипода,

при которой его рабочий орган (выходное звено) может физически достичь и выполнить захват фрукта. В зависимости от положения робота-трипода, требуемый фрукт может оказаться в точке, при достижении которой возникает особое положение или пересечение звеньев. Для исключения таких положений в процессе управления, важно определить рабочую область робота-трипода с учетом особых положений и пересечений звеньев.

Положение выходного звена робот-трипода зависит от выдвижения телескопического звена и углов ориентации подвижной платформы. Входными координатами являются длины приводных звеньев l_1, l_2, l_3 , выходными – координаты точки P выходного звена x_P, y_P, z_P . Точка P располагается на расстоянии l_t от центра подвижной платформы. Определим длины l_1, l_2, l_3 .

$$l_i = \sqrt{(x_{Bi} - x_{Ai})^2 + (y_{Bi} - y_{Ai})^2 + (z_{Bi} - z_{Ai})^2} \quad (2.1)$$

где x_{Ai}, y_{Ai}, z_{Ai} и x_{Bi}, y_{Bi}, z_{Bi} – координаты центров шарниров A_i и B_i соответственно в неподвижной системе координат $X_1Y_1Z_1$.

Определим координаты шарниров B_i в подвижной системе координат $X_2Y_2Z_2$

$$B_1^{(2)} = [R_2 \quad 0 \quad l_{c2} \quad 1]^T, \quad (2.2)$$

$$B_2^{(2)} = [0.5R_2 \quad 0.5\sqrt{3}R_2 \quad l_{c2} \quad 1]^T, \quad (2.3)$$

$$B_3^{(2)} = [-0.5R_2 \quad -0.5\sqrt{3}R_2 \quad l_{c2} \quad 1]^T. \quad (2.4)$$

Выразим координаты шарниров B_i в неподвижной системе координат $X_1Y_1Z_1$

$$P = M_{2_1} B_i^{(2)}, \quad (2.5)$$

где M_{2_1} – матрица перехода от подвижной системы координат $X_2Y_2Z_2$ к неподвижной системе $X_1Y_1Z_1$, которая включает смещения вдоль оси X_1, Y_1, Z_1 и поворот с использованием углов Эйлера α, β, γ , учитывающих ориентацию платформы

$$M_{2_1} = \begin{bmatrix} C_\alpha C_\gamma - C_\beta S_\alpha S_\gamma & -C_\alpha S_\gamma - C_\beta C_\gamma S_\alpha & S_\alpha S_\beta & 0 \\ C_\gamma S_\alpha + C_\alpha C_\beta S_\gamma & C_\alpha C_\beta C_\gamma - S_\alpha S_\gamma & -C_\alpha S_\beta & 0 \\ S_\beta S_\gamma & C_\gamma S_\beta & C_\beta & l_{c1} \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.6)$$

где $C_\alpha = \cos \alpha, S_\alpha = \sin \alpha, C_\beta = \cos \beta, S_\beta = \sin \beta, C_\gamma = \cos \gamma, S_\gamma = \sin \gamma$.

Подставив (2.6) и (2.2)-(2.4) в (2.5), получим

$$B_1 = M_{2_1} B_1^{(2)} = \begin{bmatrix} R_2(C_a C_\gamma - C_\beta S_a S_\gamma) + l_{c2} S_a S_\beta \\ R_2(C_\gamma S_a + C_a C_\beta S_\gamma) - l_{c2} C_a S_\beta \\ l_{c1} + R_2 S_\beta S_\gamma + l_{c2} C_\beta \\ 1 \end{bmatrix}, \quad (2.7)$$

$$B_2 = M_{2_1} B_2^{(2)} = \begin{bmatrix} 0.5R_2(C_a C_\gamma - C_\beta S_a S_\gamma - \sqrt{3}C_a S_\gamma - \sqrt{3}C_\beta C_\gamma S_a) + l_{c2} S_a S_\beta \\ 0.5R_2(C_\gamma S_a + C_a C_\beta S_\gamma + \sqrt{3}C_a C_\beta C_\gamma - \sqrt{3}S_a S_\gamma) - l_{c2} C_a S_\beta \\ l_{c1} + 0.5R_2(S_\beta S_\gamma + \sqrt{3}C_\gamma S_\beta) + l_{c2} C_\beta \\ 1 \end{bmatrix}, \quad (2.8)$$

$$B_3 = M_{2_1} B_3^{(2)} = \begin{bmatrix} 0.5R_2(C_\beta S_a S_\gamma + \sqrt{3}C_a S_\gamma + \sqrt{3}C_\beta C_\gamma S_a - C_a C_\gamma) + l_{c2} S_a S_\beta \\ 0.5R_2(\sqrt{3}S_a S_\gamma - C_\gamma S_a - C_a C_\beta S_\gamma - \sqrt{3}C_a C_\beta C_\gamma) - l_{c2} C_a S_\beta \\ l_{c1} - 0.5R_2(S_\beta S_\gamma + \sqrt{3}C_\gamma S_\beta) + l_{c2} C_\beta \\ 1 \end{bmatrix}, \quad (2.9)$$

Координаты шарниров A_i используемые в (2.1), определяются как

$$A_1 = [R_2 \quad 0 \quad 0 \quad 1]^T, \quad (2.10)$$

$$A_2 = [0.5R_2 \quad 0.5\sqrt{3}R_2 \quad 0 \quad 1]^T, \quad (2.11)$$

$$A_3 = [-0.5R_2 \quad -0.5\sqrt{3}R_2 \quad 0 \quad 1]^T \quad (2.12)$$

Определим координаты выходного звена робота-трипода, используя значения углов Эйлера в неподвижной системе координат X_1, Y_1, Z_1 .

Координаты P в подвижной системе координат $X_2 Y_2 Z_2$

$$P^{(2)} = [0 \quad 0 \quad (l_{c2} + l_t) \quad 1]^T. \quad (2.13)$$

$$P = M_{2_1} P^{(2)}, \quad (2.14)$$

После преобразования с учётом (2.13), (2.14) и (2.6) получим

$$P = \begin{bmatrix} S_a S_\beta (l_{c2} + l_t) \\ -C_a S_\beta (l_{c2} + l_t) \\ l_{c1} + C_\beta (l_{c2} + l_t) \\ 1 \end{bmatrix}. \quad (2.15)$$

Таким образом, задавая углы Эйлера, становится возможно определить положение выходного звена.

Определим рабочую область робота, в пределах которой должен находиться захват при сборе фруктов. Процесс определения рабочей области состоит из двух

этапов. Первый этап состоит в аппроксимации множества решений системы нелинейных неравенств, задающих ограничения рабочей области. На втором этапе, используя представление рабочей области в виде частично упорядоченного множества целых чисел в пространстве координат ориентации осуществим переход в пространство координат выходного звена. Для этого получим аналитические выражения, позволяющие выполнить отображение покрывающего множества из пространства координат углов ориентации подвижной платформы в пространство координат выходного звена.

Этап 1: Учтем следующие ограничения:

1. Геометрические параметры механизма, которые учитывают предельные значения длин штанг.
2. Особые положения робота-трипода определяемые через матрицу Якоби.
3. Ограничение на ориентацию платформы (телескопическое звено должно быть направлено вверх).
4. Пересечения звеньев, определяемые геометрическим методом.

Введём ограничения на геометрические параметры механизма

$$l_{\min} \leq l_i \leq l_{\max}, \quad (2.16)$$

где l_{\min} , l_{\max} определяются конструктивными параметрами механизма;

Для определения особых положений рассмотрим метод, основанный на анализе матрицы Якоби [92], определитель которой имеет вид

$$\det(J_A) = \begin{vmatrix} \frac{\partial l_1}{\partial \alpha} & \frac{\partial l_1}{\partial \beta} & \frac{\partial l_1}{\partial \gamma} \\ \frac{\partial l_2}{\partial \alpha} & \frac{\partial l_2}{\partial \beta} & \frac{\partial l_2}{\partial \gamma} \\ \frac{\partial l_3}{\partial \alpha} & \frac{\partial l_3}{\partial \beta} & \frac{\partial l_3}{\partial \gamma} \end{vmatrix}, \quad (2.17)$$

где l_i определяются с учётом формул (2.1), (2.7) -(2.12).

Ввиду громоздкости формул для каждого из элементов определителя приведём только первый из них

$$\frac{\partial l_1}{\partial \alpha} = \frac{-(2 s_1 (s_2 - R_1) - 2 s_1 s_2)}{2 \left((l_{c1} + l_{c2} C_\beta + R_2 S_\beta S_\gamma)^2 + (s_2 - R_1)^2 + s_1^2 \right)^{0.5}}$$

где

$$s_1 = R_2 (C_\gamma S_\alpha + C_\alpha C_\beta S_\gamma) - l_{c2} C_\alpha S_\beta, \quad s_2 = R_2 (C_\alpha C_\gamma - C_\beta S_\alpha S_\gamma) + l_{c2} S_\alpha S_\beta$$

Запишем условие наличия особых положений

$$\det(J_A) = 0 \quad (2.18)$$

Для исключения особых положений из рабочей области необходимо обеспечить знакопостоянство определителя матрицы Якоби. Чтобы обеспечить это условие необходимо добавить к условию (2.18) одно из условий: $\det(J_A) < 0$ или $\det(J_A) > 0$ в зависимости от знака определителя. Также следует учесть ограничение, связанное с обеспечением ориентации платформы, при которой телескопическое звено направлено вверх, так как ориентация платформы, при котором телескопическое звено направлено вниз, не исключается условием (2.18), ограничение имеет следующий вид

$$z_p > l_{c1}.$$

Пересечение звеньев определяется с использованием геометрического метода, основанного на определении расстояния между двумя произвольными прямыми в пространстве [93].

Таким образом, учитывая ограничения на геометрические параметры, особые положения, ограничения на ориентацию платформы и пересечения звеньев в соответствии с (2.16), (2.18) и (2.19), выполним построение рабочей области в пространстве координат ориентации подвижной платформы робота-трипода. Построение рабочей области выполнено с применением методов аппроксимации [94] и представлено в виде покрывающего множества, которое затем преобразовано в частично упорядоченное множество целых чисел для повышения точности и удобства обработки данных [95, 96]. Однако, полученная рабочая область не учитывает выдвижение телескопического звена, так как построена в пространстве ориентации α, β, γ .

Этап 2: рассмотрим перенос покрывающего множества в пространство координат x_P, y_P, z_P выходного звена робота-трипода. Множество положений выходного звена вычислим с использованием формулы (2.15). При этом телескопическое звено за счёт выдвижения представляет собой интервал, образующий в пространстве прямую линию

$$l_{t,\min} \leq l_t \leq l_{t,\max}.$$

Для того, чтобы учесть множество l_t (телескопическое звено) при построении рабочей области робота-трипода, требуется определить пересечение l_t с аппроксимированным пространством при всех ориентациях подвижной платформы. Используя представление рабочей области в виде упорядоченного множества целых чисел, определим множество B координат выходного звена P в пространстве целых чисел. Для этого используем алгоритм Брезенхема [97], который позволяет определить, какие ячейки двумерного дискретного пространства нужно закрасить, чтобы получить близкое приближение прямой линии между двумя заданными точками. Существует алгоритм Брезенхема для трёхмерного случая [98], однако он работает только в пространстве целочисленных значений, что ограничивает его применение. Это обусловлено тем, что прямая линия может соединять только центры ячеек дискретного пространства, за счет чего снижается точность аппроксимации. Для более точного определения покрывающего множества в пространстве координат выходного звена необходимо использовать дробные значения. В этом случае положение точек в пространстве десятичных чисел задается как доля между центрами соседних ячеек. Такой подход вызывает дополнительную проблему: возникают ситуации, когда исходная линия проходит вдоль границ ячеек дискретного пространства, и её пересечения с ячейками не фиксируются. Для устранения этого недостатка синтезируем алгоритм, применимый для трехмерного пространства с учётом использования исходных данных, принадлежащих пространству действительных чисел (рисунок 2.2) (координаты x_1, y_1, z_1 выходного звена при $l_{t,\min}$ и x_2, y_2, z_2 при $l_{t,\max}$). При этом координаты должны соответствовать покрывающему множеству рабочей

области, представленному в виде упорядоченного множества целых чисел, соответственно должны быть получены с учётом точности аппроксимации Δ , используя формулу (2.4)

$$x_1 = \frac{S_a S_\beta (l_{c2} + l_{t,\min})}{\Delta}, \quad y_1 = \frac{-C_a S_\beta (l_{c2} + l_{t,\min})}{\Delta}, \quad z_1 = \frac{l_{c1} + C_\beta (l_{c2} + l_{t,\min})}{\Delta}$$

$$x_2 = \frac{S_a S_\beta (l_{c2} + l_{t,\max})}{\Delta}, \quad y_2 = \frac{-C_a S_\beta (l_{c2} + l_{t,\max})}{\Delta}, \quad z_2 = \frac{l_{c1} + C_\beta (l_{c2} + l_{t,\max})}{\Delta}.$$

Входными данными алгоритма являются координаты начала и конца отрезка, заданные в виде множеств \mathcal{X}_b и \mathcal{X}_e соответственно.

```

Input:  $x_b = \{x_b^{(0)}, x_b^{(1)}, \dots, x_b^{(n)}\}, x_e = \{x_e^{(0)}, x_e^{(1)}, \dots, x_e^{(n)}\}$ 
1:  $x_I = \emptyset, B = \emptyset$ 
2: for  $i = 1, \dots, n$  do
3:    $\delta^{(i)} = x_e^{(i)} - x_b^{(i)}$ 
4:   if  $\delta^{(i)} = 0$  then  $x_b^{(i)} = \lfloor x_b^{(i)} \rfloor, x_e^{(i)} = \lfloor x_e^{(i)} \rfloor$  end if
5:    $x_I = x_I \cup x_b^{(i)}$ 
6:   if  $\delta^{(i)} < 0$  then  $x_m^{(i)} = \lfloor x_b^{(i)} + 0.5 \rfloor - 0.5$  else  $x_m^{(i)} = \lfloor x_b^{(i)} - 0.5 \rfloor + 0.5$  end if
7: end for
8:  $B = B \cup x_I$ 
9: for  $i = 1, \dots, n$  do
10:  while  $x_m^{(i)} \cdot \text{sign}(\delta^{(i)}) \leq x_2 \cdot \text{sign}(\delta^{(i)})$  do
11:   for  $k = 1, \dots, n$  do
12:    if  $i \neq k$  then
13:      $x_c^{(k)} = x_b^{(k)} + \delta^{(k)} (x_m^{(i)} - x_b^{(i)}) / \delta^{(i)}$ 
14:    end if
15:  end for
16:  flag = true
17:  for  $k = 1, \dots, i - 1$  do
18:   if  $\lfloor x_c^{(k)} \rfloor = x_c^{(k)} - 0.5$  then
19:    flag = false
20:    break
21:  end if
22: end for
23: if flag = true then
24:  for  $k = 1, \dots, n$  do
25:   if  $i = k$  then
26:     $x_I^{(k)} = x_m^{(i)} + 0.5 \text{sign}(\delta^{(i)})$ 
27:   else if  $k < i$  then
28:     $x_I^{(k)} = \lfloor x_c^{(k)} \rfloor$ 
29:   else
30:    if  $(\lfloor x_c^{(k)} \rfloor \neq x_c^{(k)} - 0.5) \vee (\text{sign}(x_c^{(k)}) = \text{sign}(\delta^{(k)}))$  then
31:      $x_I^{(k)} = \lfloor x_c^{(k)} \rfloor$ 
32:    else
33:      $x_I^{(k)} = \text{sign}(\delta^{(k)}) \lfloor |x_c^{(k)}| \rfloor$ 
34:    end if
35:  end if
36: end for
37:  $B = B \cup x_I$ 
38: end if
39:  $x_m^{(i)} = x_m^{(i)} + \text{sign}(\delta^{(i)})$ 
40: end while
41: end for
42: return B

```

Рисунок 2.2 - Модифицированный алгоритм Брезенхэма.

На шаге 1 алгоритма создаётся пустое множество B для сохранения пересекаемых отрезком ячеек. Для промежуточного сохранения координат каждой из ячеек создано вспомогательное пустое множество x_1 . Операции в цикле с шага 2 до 7 повторяются для каждого из измерений. На шаге 3 формируется множество $\delta^{(i)}$, соответствующее разнице между координатами начальной и конечной точки

отрезков. На шагах 4-5 в множество x_1 вносятся координаты ячейки, соответствующей начальной точке отрезка. На шаге 6 формируется вспомогательное множество x_m , используемое для последовательного перебора координат от начальной до конечной точки по каждому из измерений. На шаге 8 ячейка x_1 добавляется в B . На шагах 9-41 выполняется цикл, в ходе которого по каждому из измерений последовательно выполняется приращение координат и добавление в множество B ячеек, пересекаемых отрезком.

Построение рабочей области выглядит следующим образом: во всем массиве, представляющем дискретное пространство, выполняется перебор ячеек с учетом различных возможных ориентаций робота-трипода. Для каждой комбинации ориентации определяется отрезок, соответствующий траектории движения выходного звена механизма. Этот отрезок описывает изменение положения выходного звена при выдвигании или втягивании телескопического звена в заданном диапазоне длин l_t . Далее применяется модифицированный алгоритм Брезенхема для трехмерного случая, который позволяет вычислить, какие именно ячейки дискретного пространства пересекает данный отрезок. Алгоритм учитывает, как целочисленные координаты ячеек, так и дробные значения, что обеспечивает высокую точность аппроксимации. При этом учитываются ограничения, накладываемые геометрией механизма, такими как предельные углы поворота, максимальная и минимальная длины штанг, а также условия отсутствия пересечений звеньев и особых положений, определяемых через определитель матрицы Якоби. Ячейки, которые пересекаются отрезком, считаются принадлежащими рабочей области робота-трипода. Эти ячейки формируют покрывающее множество, которое затем преобразуется в частично упорядоченное множество целых чисел. Таким образом, каждая пересекаемая ячейка представляет собой допустимое положение выходного звена в пространстве, удовлетворяющее всем заданным ограничениям и обеспечивающее досягаемость фруктов. Это позволяет получить полное представление о рабочей области робота-трипода с учетом всех кинематических и геометрических особенностей его конструкции.

Выполним численное моделирование рабочей области. Для этого разработан программный комплекс на языке программирования C++. Визуализация пересечений звеньев выполнена с использованием скрипта на языке Python (библиотеки Matplotlib и JSON). Визуализация трёхмерных результатов выполнена посредством экспорта упорядоченного множества целых чисел, описывающих рабочую область, в формат STL.

2.2 Результаты моделирования построения рабочей области робота-трипода с учётом особых положений и пересечений звеньев

Численное моделирование рабочей области с учётом особых положений и пересечения звеньев выполнено для следующих параметров: $R_1 = 300$ мм, $R_2 = 200$ мм, $l_{min} = 360$ мм, $l_{max} = 600$ мм, $l_{c1} = 500$ мм, $l_{c2} = 0$ мм, $l_{t,min} = 500$ мм, диаметр звеньев $D_{link} = 20$ мм, минимальный угол между звеньями $\varphi_{min} = 10^\circ$. Рабочая область в пространстве координат α, β, γ ориентации подвижной платформы без учёта особых положений показана на рисунке 2.3, а. На рисунке 2.3, б показана рабочая область в пространстве координат выходного звена x_p, y_p, z_p .

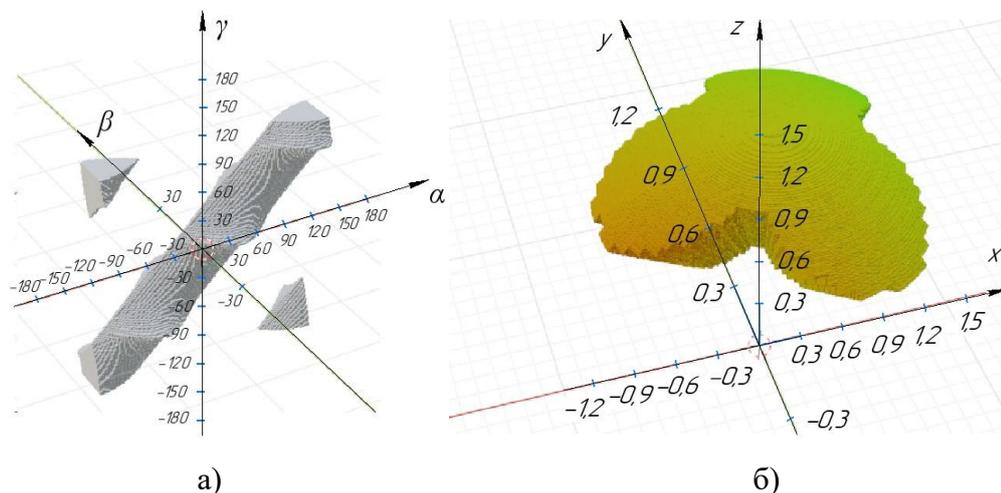


Рисунок 2.3 - Рабочее пространство концептуальной платформы:

а) в координатах (α, β, γ) ; б) в координатах (x_p, y_p, z_p) .

Для выбора знака в условии знакопостоянства определителя матрицы Якоби с целью исключения особых положений определена рабочая область как с

условием положительности (рисунок 2.4 а, в) определителя, так и с условием отрицательности (рисунок 2.4 б, г).

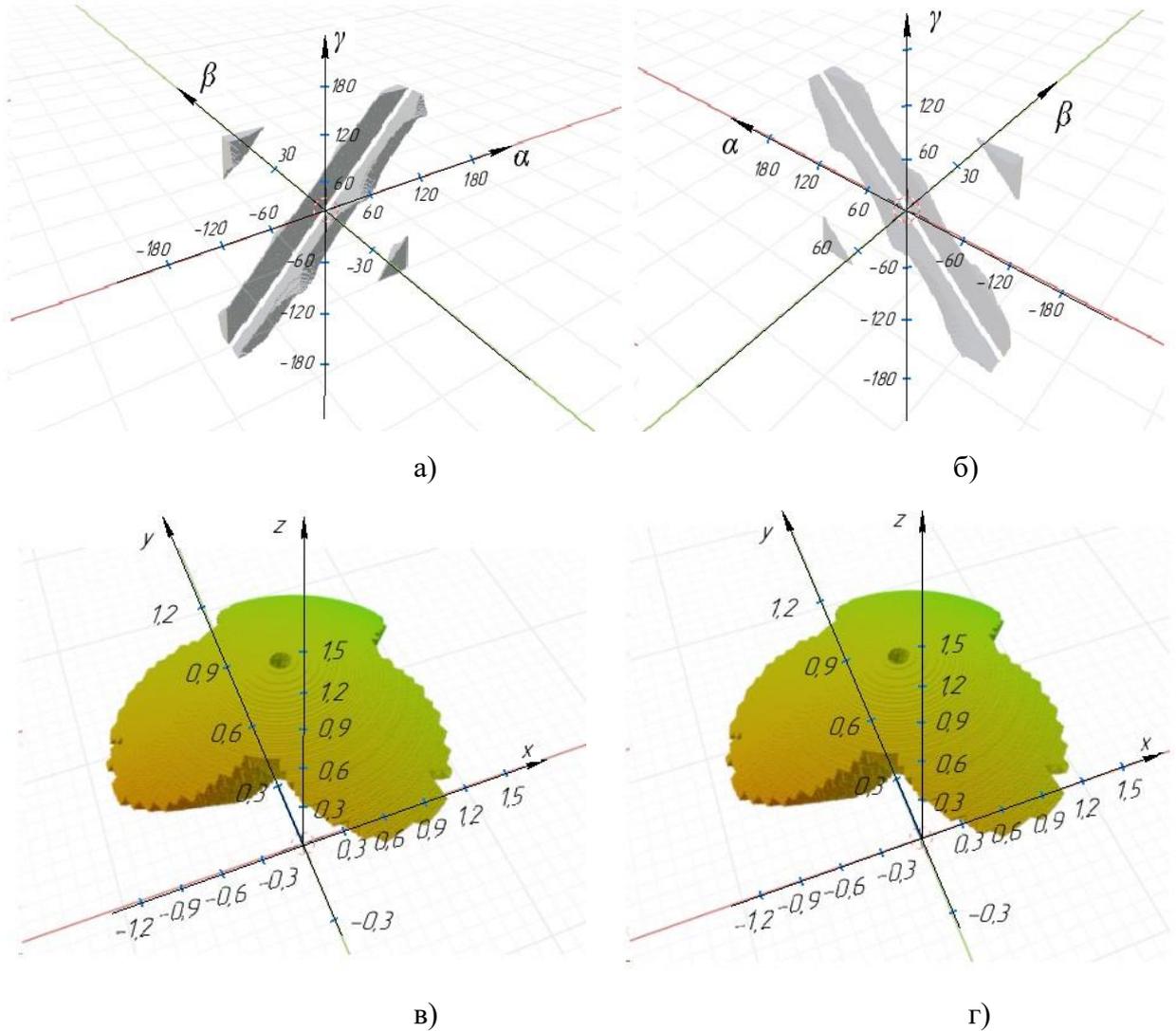


Рисунок. 2.4 - Рабочая область платформы: а) в координатах (α, β, γ) при отрицательном знаке определителя матрицы Якоби; б) в координатах (α, β, γ) при положительном знаке; в) в координатах (x_p, y_p, z_p) при отрицательном знаке определителя матрицы Якоби; г) в координатах (x_p, y_p, z_p) при положительном знаке матрицы Якоби.

Из рисунков 2.4 и видно, что рабочая область в пространстве координат (α, β, γ) разделяется на две части, однако рабочая область в пространстве координат (x_p, y_p, z_p) практически совпадает для различных знаков определителя матрицы Якоби. В обоих случаях из рабочей области исключена центральная зона, в которой определитель матрицы Якоби равен нулю. Таким образом, если конструкция захвата РТС вместе с закрепленной на нем гибкой трубой обеспечивает сбор

фруктов при любой ориентации выходного звена, это свидетельствует о том, что система функционирует корректно как при положительных, так и при отрицательных значениях определителя матрицы Якоби. Выявлено, что при заданных исходных данных для моделирования пересечение звеньев отсутствуют. Увеличим диапазоны изменения длин штанг с сохранением соотношения $l_{min}/l_{max}=0.6$ для обеспечения достаточного пространства внутри штанги для размещения шарико-винтовых пар. При увеличении размеров до $l_{min} = 420$ мм, $l_{max} = 700$ мм возникают пересечения центральной кинематической цепи с подвижной платформой. Для определения оптимальных соотношений длин звеньев и оптимальной рабочей области требуется выполнить оптимизацию геометрических параметров.

2.3 Алгоритм оптимизации геометрических параметров робота-трипода

Объём рабочей области зависит от диапазонов поступательного движения телескопического звена и ориентации подвижной платформы. Диапазоны поступательного движения телескопического звена прямо пропорциональны ходу его штока. В то время, как диапазоны ориентации подвижной платформы зависят от геометрических параметров трипода. Сформулируем задачу оптимизации этих параметров для уменьшения количества ориентаций, недостижимых из-за ограничений на длины приводных штанг, особые положения и пересечения звеньев. Количество недостижимых ориентаций в дискретизированном пространстве углов Эйлера будет использовано в качестве первого критерия. Вторым критерием будет компактность конструкции. Также введём параметрические ограничения, учитывающие максимально возможные значения геометрических параметров, исходя из обеспечения возможности установки на колёсную платформу, ширина которой должна позволять выполнять перемещение в пределах междурядного пространства промышленного сада.

Постановка задачи оптимизации:

1. В качестве параметров оптимизации использованы: радиусы платформ R_1 и R_2 , предельные значения длин приводных штанг L_{min} и L_{max} , а также длины звеньев центральной кинематической цепи l_{c1} и l_{c2} ;

2. В качестве критериев оптимизации выбраны K_1 и K_2 , при этом первый критерий оптимизации K_1 выбирается, исходя из обеспечения компактности конструкции по следующей зависимости

$$K_1 = R_1 + R_2 + l_{min} + l_{max} + p_1 \left(\left(\frac{k_R R_1}{R_2} \right)^{2\vartheta-1} - 1 \right) \rightarrow \min, \quad (2.19)$$

где R_1 – радиус неподвижной платформы, R_2 – радиус подвижной платформы, l_{min} и l_{max} – предельные значения изменения длин штанг, p_1 – заданный коэффициент штрафа, $k_R = 0.7$ – рекомендуемые согласно [99] соотношение радиусов платформ R_2/R_1 , ϑ – функция Хэвисайда

$$\vartheta = \begin{cases} 1, & \text{если } \frac{R_2}{R_1} < 0.7 \\ 0 & \text{– иначе} \end{cases}. \quad (2.20)$$

В качестве второго критерия K_2 выбрана функция, имеющая прямую зависимость с объёмом рабочего пространства и характеризующая диапазоны недостижимых ориентаций платформы N_w^- с учётом пересечений звеньев. Учитывая зависимость объёма рабочей области от диапазона изменения длины l_t телескопического звена и диапазонов ориентации подвижной платформы, получим

$$K_2 = N_w^- = |S| - |W|, W \subseteq S, \quad (2.21)$$

где S – множество возможных ориентаций платформы с дискретным шагом, W – множество достижимых ориентаций платформы, N_w^- – количество недостижимых ориентаций платформы.

3. Введем ограничение на соотношение минимальной и максимальной длин приводных штанг для обеспечения достаточного пространства внутри штанги для размещения ШВП

$$\frac{l_{min}}{l_{max}} \geq 0.6. \quad (2.22)$$

Синтезируем универсальный алгоритм многокритериальной оптимизации (АМО), применимый для широкого круга задач с различным числом параметров и

критериев. Суть разработанного алгоритма состоит в нормализации критериев и итеративной процедуры оптимизации с использованием однокритериальных эволюционных алгоритмов с использованием свёртки критериев с переменными коэффициентами важности. Коэффициенты важности α_i разделены на две составляющие. Для вычисления первой составляющей β_i , обеспечивающей равноценность критериев и являющейся постоянной, решим систему уравнений для получения коэффициентов важности, обеспечивающих взвешенную сумму равноценных критериев. Система уравнений имеет вид

$$\begin{cases} \beta_1 K_1 = \beta_2 K_2 = \dots = \beta_n K_n \\ \sum_{j=1}^n \beta_j = 1 \end{cases} \quad (2.23)$$

где K_i — среднее значение i -го критерия.

Решим систему уравнений, пусть $\beta_i K_i = y$, тогда

$$\beta_i = \frac{1}{K_i \sum_{j=1}^n \left(\frac{1}{K_j}\right)} \quad (2.24)$$

После выполнения нормализации критериев в алгоритме многокритериальной оптимизации выполняется итеративная однокритериальная оптимизация и обновление Парето множества с использованием эволюционных алгоритмов с различными коэффициентами важности α_i , которые определяются как

$$\alpha_i = \frac{\beta_i \gamma_i}{\sum_{i=1}^n \gamma_i} \quad (2.25)$$

где $\gamma_i \in [0,1]$ – составляющие коэффициентов важности, изменяемые после каждой итерации однокритериальной оптимизации в соответствии с n -мерной сеткой, имеющей диапазоны по каждому из измерений $[0; 1]$ и шагом δ_γ . Многократное выполнение процедуры позволяет последовательно сформировать участки Парето множества. Синтезированный алгоритм многокритериальной оптимизации для решения поставленной задачи представлен на рисунке 2.5.

Algorithm 1 Multi-objective Optimization

Input: $\delta, p = \{p_1, p_2, \dots, p_m\}$ ▷ δ is accuracy, p is parameters
Output: Pareto Set P

```

1: for j:=1 to l do ▷ l is number of iterations for average  $K_i$  calculation
2:   for k:=1 to m do ▷ m is number of parameters
3:      $p_k^{(j)} = \text{random}(\underline{p}_k, \overline{p}_k)$ 
4:   end for
5:   for i:=1 to n do ▷ n is number of criteria
6:      $K_i^{(j)} = f(p^{(j)})$  ▷ Criteria calculation
7:   end for
8: end for
9: for i:=1 to n do
10:   $K_i = \sum_{j=1}^l K_i^{(j)} / l$  ▷ Average criteria calculation
11: end for
12:  $\beta = \{\beta_1, \beta_2, \dots, \beta_n\} = \text{Normalization}(\delta, K)$  ▷  $\beta$  part of criteria weights
13: for each node  $i$  of grid do
14:    $s_\gamma = 0$ 
15:   for i:=1 to n do
16:      $\gamma_i = f(i)$  ▷  $\gamma$  part of criteria weights
17:      $s_\gamma = s_\gamma + \gamma_i$ 
18:   end for
19:   for i:=1 to n do
20:      $\alpha_i = \beta_i \gamma_i / s_\gamma$  ▷  $\alpha$  is criteria weights
21:   end for
22:   Single-objective optimization
23:   Update the Pareto Set  $P$ 
24: end for

```

Рисунок 2.5 - Алгоритм многокритериальной оптимизации параметров.

Описание алгоритма. На шагах 1–11 выполняется вычисление средних значений критериев K_i . Для вычисления среднего значения критериев необходимо произвести определенное количество расчётов с различными значениями параметров в заданном диапазоне. На первом шаге задаётся количество повторений процесса вычисления критериев от 1 до l . На 2–4 шаге выполняется цикл присвоения параметрам произвольных значений в заданном диапазоне для текущего расчёта критерия. На 5–7 шаге происходит вычисление i -того критерия j -тый раз расчёта с учетом произвольно заданных параметров. На шаге 8 происходит окончание цикла расчёта критериев. На 9–11 шагах выполняется цикл, определяющий средние арифметические значения каждого критерия. На шаге 12 выполняется нормализация критериев посредством вычисления коэффициентов важности β_i , при которых достигается равенство средних значений критериев K_i , по формуле (2.24). На шаге 13 выполняется перебор узлов дискретной сетки, наложенной на пространство значений переменных составляющих коэффициентов важности γ_i . Для каждого из узлов на шагах 14–18 вычисляются значения γ_i и

сумма s_γ этих значений. На шагах 19 –21 для каждого из узлов вычисляются значения коэффициентов важности α_i по формуле (2.25). На шаге 22 выполняется однокритериальная оптимизация для этого узла, а на шаге 23 обновление Парето множества. Алгоритм завершает свою работу после окончания перебора всех узлов сетки (всех комбинаций значений γ_i).

В качестве алгоритмов однокритериальной оптимизации используются модификации эволюционных алгоритмов с параллельным вычислением функций приспособленности, такие как Генетический алгоритм (GA) [81,86], Алгоритм серых волков (GWO) [101, 102, 103] и алгоритм Роя частиц (PSO)) [104, 105]. Проводится моделирование на основе многокритериального алгоритма АМО, где каждая итерация процедур однокритериальной оптимизации выполняется в два этапа. На первом этапе диапазон параметров изменяется в пределах, соответствующих габаритным размерам рабочего пространства по каждой из координат. На каждом последующем этапе диапазоны параметров уменьшаются в 10 раз, причем центр новых диапазонов соответствует лучшему результату, полученному на предыдущем этапе. Строятся аппроксимации Парето множеств с использованием каждого из алгоритмов. Затем выполняется объединение полученных множеств, из которого исключаются точки, не удовлетворяющие условию Парето-оптимальности. Таким образом, формируется результирующее множество решений для дальнейшего анализа.

Разработанный алгоритм может быть применён для решения задачи многокритериальной оптимизации с любым количеством критериев. Количество параметров, ограничений и характер целевых функций также не оказывают влияние на применимость алгоритма, однако должны быть учтены при выборе алгоритма однокритериальной оптимизации, выполняемой на шаге 22.

Зададим исходные данные для вычислительного эксперимента. Диапазоны параметров оптимизации: $R_1 \in [200; 1000]$, $R_2 \in [200; 1000]$, $L_{min} \in [200; 1200]$, $L_{max} \in [400; 1400]$, $l_{c1} \in [200; 1400]$, $l_{c2} \in [0; 800]$, коэффициент штрафа $p_1 =$

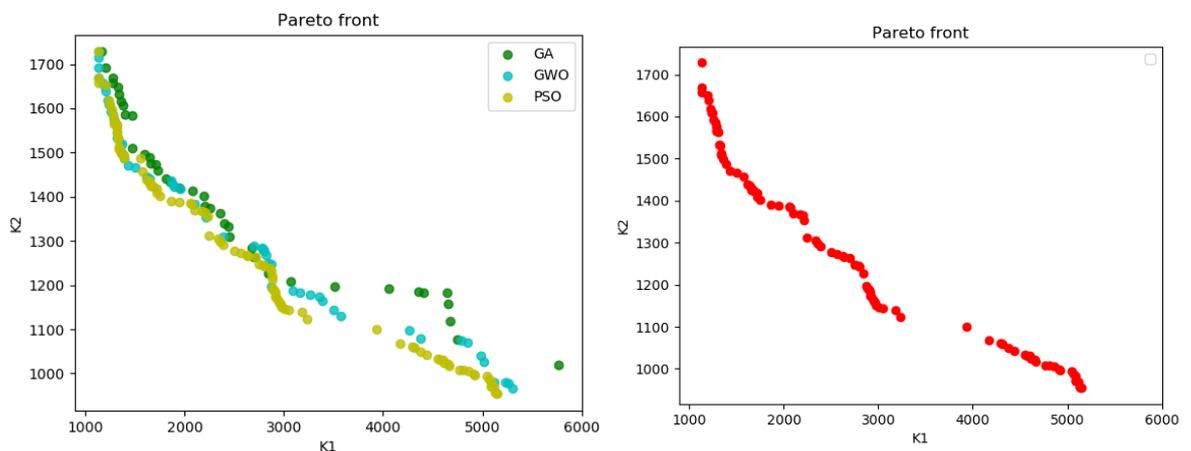
1000, количество узлов сетки для каждого измерения для перечисления компонента γ_i равно 20. Число особей, поколений выбрано экспериментально.

Параметры алгоритма GA: количество особей в начальной популяции $H = 1000$, число поколений $W = 20$, число скрещиваний на каждой итерации $S_{GA} = 500$, количество возможных значений каждого из параметров $g = 2^{25}$, вероятность мутации $p_m = 80\%$. Общее число поколений в результате равно 20 (количество шагов перебора значений коэффициента важности критерия K_1) * 20 (количество шагов перебора значений коэффициента важности критерия K_2) * 20 (количество поколений для одного веса) = 8000.

Параметры алгоритма GWO: $H = 1000$, $W = 20$, число новых особей на каждой итерации $S_{GWO} = 500$.

Параметры алгоритма PSO: $H = 1000$, $W = 20$, число групп $G = 2$, значения свободных параметров $\alpha = 0,7$, $\beta = 1,4$, $\gamma = 1,4$, выбранные на основе рекомендаций [96].

Каждая итерация однокритериальной оптимизации выполнена в два этапа. На первом этапе диапазон параметров изменялся в исходных диапазонах, приведённых выше. На каждом последующем этапе диапазоны уменьшались в 10 раз. При этом центр диапазонов соответствовал лучшему результату, полученному на предыдущем этапе. На рисунке 2.6, а показана аппроксимация Парето-множеств с использованием каждого из алгоритмов, после чего выполнено объединение множеств и исключение из результирующего множества точек, не удовлетворяющих условию по Парето-оптимальности (рисунок 2.6, б).



а)

б)

Рисунок 2.6 – Парето-множества: а) алгоритм GA, алгоритм GWO, алгоритм PSO; б) Парето множество, полученное на основе трех алгоритмов.

Минимальные, максимальные и среднеарифметические значения параметров и критериев Парето-множества (рисунок 2.6, б) представлены в таблице 2.1.

Таблица 2.1 - Результаты оптимизации

Параметр	Минимальное значение	Максимальное значение	Среднее значение
R_1	277,474	1000	481,077
R_2	200	281	225,354
L_{min}	240,011	847,629	603,981
L_{max}	400	1400	999,803
l_{c1}	200	1066,07	580,506
l_{c2}	80,0839	800	109,996
K_1	1125,74	5152,26	2769,685
K_2	954	1728	1282,74

В качестве метрики эффективности алгоритмов использована доля точек аппроксимации результирующего множества Парето, полученных каждым из алгоритмов. Множество Парето состоит из 96 точек. Из них лишь две получены с использованием алгоритма GA, 13 получено с использованием GWO алгоритма. Остальные 81 точки получены с использованием PSO алгоритма, что показывает его более высокую эффективность для решения поставленной задачи. В качестве метрики эффективности алгоритмов использована доля точек аппроксимации результирующего фронта Парето, полученных каждым из алгоритмов. Алгоритм PSO показал лучшие показатели эффективности и позволил получить более 86,4% точек Парето-множества, в то время как GA и GWO позволили получить лишь 2,1% и 13,5% соответственно.

2.4 Алгоритм технического зрения для локализации деревьев и распознавания фруктов

Современные системы технического зрения, основанные на применении нейронных сетей, значительно расширили возможности автоматизации различных процессов, включая распознавание объектов на изображениях. Одной из важных задач в области технического зрения является автоматическое распознавание и

классификация объектов, таких как фрукты, в реальных условиях. Одной из широко используемых архитектур для распознавания объектов является модель YOLO (You Only Look Once) [106], которая зарекомендовала себя как высокоскоростное и точное решение для обнаружения объектов в реальном времени. Одним из вариантов повышения качества распознавания для снижения количества ложно распознанных объектов является интеграция дополнительных алгоритмов, основанных на анализе геометрических свойств объектов. Учитывая близость формы плодов на изображении к форме окружности, в качестве дополнения может быть применён метод Хафа [107] для поиска объектов круглой формы на изображении. Преобразование Хафа является одним из наиболее широко используемых методов для обнаружения геометрических фигур на изображении. Метод разработан в 1962 году П. Хафом для обнаружения линий на изображении, но с тех пор расширены и применяются для обнаружения кругов и других геометрических фигур [108]. Преимуществом использования кругов Хафа для технического зрения является их способность обнаруживать круги различных размеров и ориентаций, а также устойчивость к шумам и искажениям на изображении. Кроме того, они могут быть применены для обнаружения не только кругов, но и других геометрических фигур, таких как эллипсы, прямоугольники и т.д [109].

Автоматизация процесса сбора фруктов предполагает точное и оперативное определение их местоположения, а также детальный анализ структуры крон деревьев и их пространственного размещения в междурядьях. Без такой информации РТС может сталкиваться с проблемами, связанными с планированием движения и безопасным доступом к плодам. Использование традиционных методов технического зрения с одной камерой ограничивается либо недостаточной детализацией, либо невозможностью охватить большие области для анализа. С учетом этого разработана система технического зрения, основанная на нейронной сети для определения координат фруктов с использованием данных, полученных с

двух бинокулярных камер. Камеры расположены на фронтальной части мобильной платформы и на захватном устройстве (рисунок 2.7).

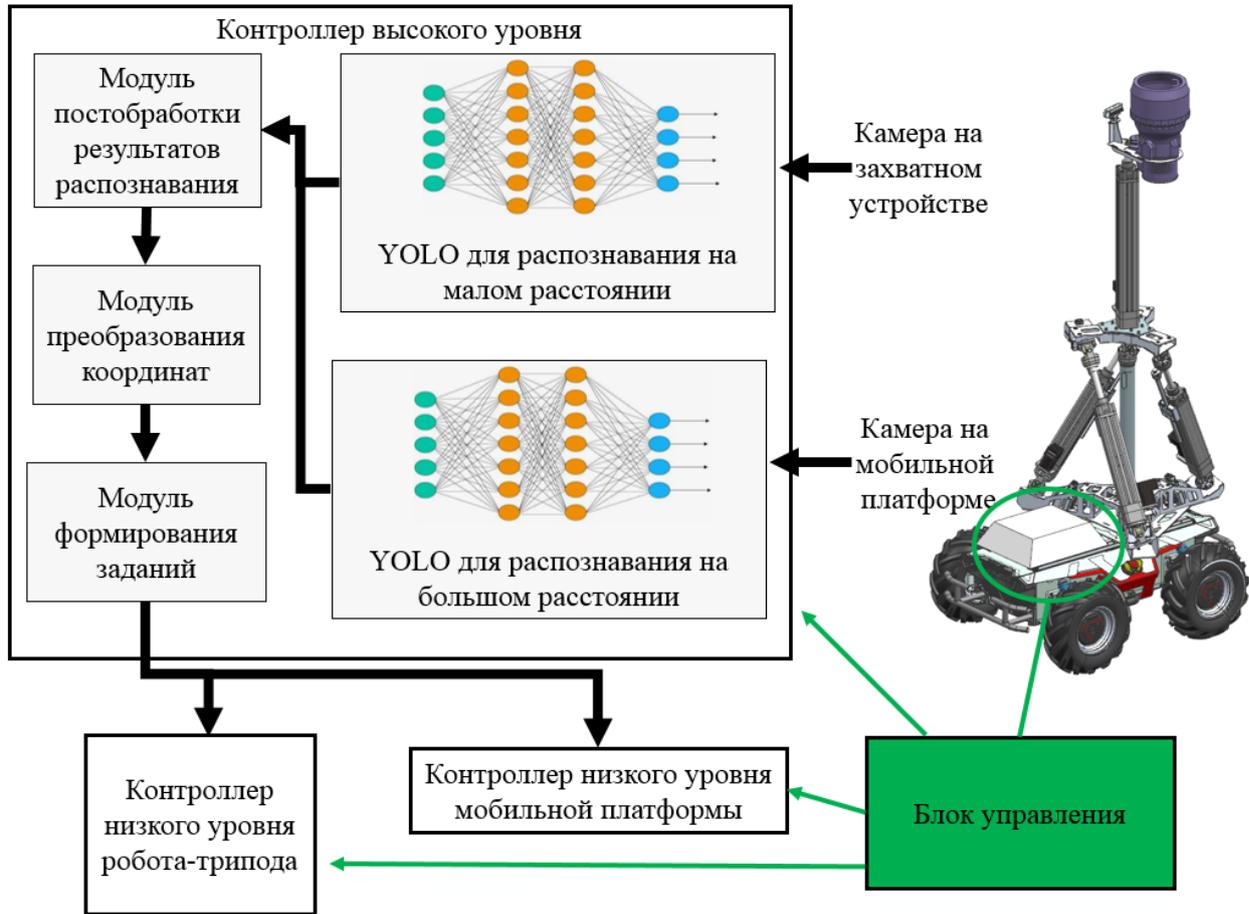


Рисунок 2.7 – Схема обработки данных с камер технического зрения.

Для успешного определения положения фруктов в условиях сада разработана система технического зрения, определяющая координаты фруктов. Схема системы представлена на рисунке 2.8. Система автоматического сбора фруктов выполняет распознавание в два этапа. На первом этапе фронтальная RGB-D камера, установленная на мобильной платформе, определяет положение деревьев и фруктов. Нейронная сеть сегментирует кроны деревьев и выявляет фрукты, используя данные глубины для фильтрации объектов за пределами дерева. Вычисленные 3D-координаты фруктов позволяют мобильной платформе ориентироваться относительно дерева. На втором этапе камера, установленная на захватном устройстве, уточняет координаты фруктов. С помощью нейронной сети и метода преобразования Хафа выделяются границы объектов, определяются размеры, а проверка в HSV-пространстве исключает неподходящие объекты.

Объединение данных двух камер с помощью фильтра Калмана исключает дубликаты, обеспечивая точное позиционирование для сбора.



Рисунок 2.8 – Алгоритм технического зрения.

Рассмотрим каждый этап более подробно.

Этап 1. Обработка данных с камеры на мобильной платформе.

На первом шаге алгоритма RGB-D камера, установленная на фронтальной части мобильной платформы, передает изображения и данные глубины в нейронную сеть.

Шаг 2: выполняется определение кроны деревьев с использованием нейронной сети YOLOv8. Нейронная сеть YOLOv8 обеспечивает определение объектов в реальном времени, имеет оптимизированную структуру, что позволяет ее интегрировать на устройства с низкими вычислительными способностями, имеет возможность объединять несколько снимков для определения частично

перекрытых объектов. На рисунке 2.9 представлена архитектура YOLOv8, включающая несколько ключевых компонентов. Backbone отвечает за извлечение признаков из входного изображения, формируя многомасштабные представления. Head выполняет окончательные предсказания, включая координаты объектов, их классы и маски сегментации.

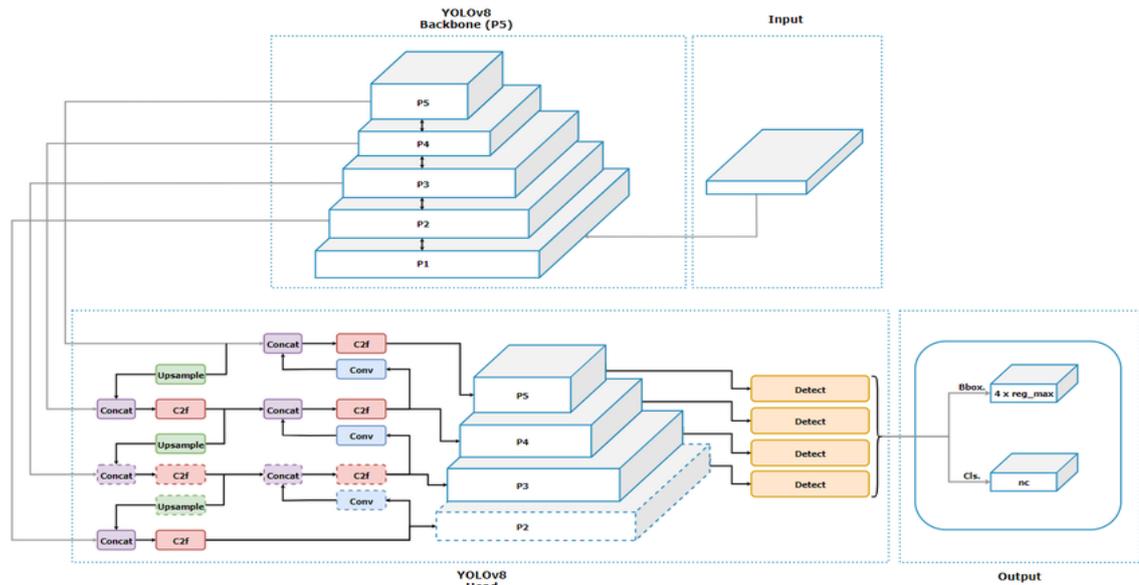


Рисунок 2.9 – Архитектура нейронной сети YOLOv8.

В конфигурации применяются следующие параметры: разрешение входных изображений 640×640 пикселей, порог уверенности (confidence threshold) установлен на уровне 0.25, а порог подавления перекрытия (NMS threshold) равен 0.45. Обработка RGB-D данных осуществляется на основе предварительно обученных моделей, дообученных на специализированном наборе данных, содержащем изображения крон деревьев с различной листвой.

Шаг 3: выполняется процесс создания отдельных изображений на основе распознавания крон деревьев и получения координат областей интереса, полученных на предыдущем шаге. Сначала система использует предоставленные координаты каждой выделенной области, которые представлены в виде ограничительных рамок (bounding boxes). Исходное изображение разделяется на части, соответствующие указанным рамкам. Для предотвращения обрезки краев крон к координатам bounding boxes добавляются небольшие отступы, чтобы расширить рамку. Это применимо, если крона дерева частично пересекает границы

выделенной области. Отступы рассчитываются так, чтобы оставаться в пределах размеров исходного изображения.

Шаг 4: алгоритма выполняется поиск и определение фруктов на отдельных изображениях крон, полученных на третьем шаге, учитывая особенности их формы и цвета. Для этого используется метод сегментации на базе нейронной сети YOLOv8. Данные глубины, полученные камерой бинокулярного зрения, позволяют отсеять объекты, находящиеся за пределами дерева. Для повышения точности распознавания модели обучались на дополнительном наборе данных, включающем изображения фруктов различных сортов и стадий зрелости.

Шаг 5: алгоритма рассчитываются координаты для перемещения РТС в область сбора фруктов. Трехмерные координаты каждого обнаруженного фрукта вычисляются в системе координат камеры на основе данных глубины. Глубинные карты фильтруются с использованием медианного фильтра для устранения шумов и искажений изображения. Преобразование координат из системы камеры в систему координат мобильной платформы выполняется с использованием матрицы преобразования, которая учитывает положение камеры на платформе. После расчета координат фрукта мобильная платформа ориентируется так, чтобы центр зоны обнаруженных объектов находился в пределах досягаемости робота.

Шаг 6: формируется список заданий для сбора, каждое из которых соответствует кроне дерева. В состав одного задания входит несколько подзадач, соответствующих распознанному плоду.

Результатом первого этапа является грубая фиксация положения деревьев и фруктов, позволяющая ориентировать мобильную платформу относительно дерева и минимизировать последующие корректировки на этапе уточнения. После перемещения РТС для выполнения задания из списка, сформированного на шестом шаге, камера, установленная на захватном устройстве, в достаточной мере приближается к кроне деревьев для точного определения расположения плодов.

Этап 2. Обработка данных с камеры на захватном устройстве.

На первом шаге алгоритма активируется камера, установленная на захватном устройстве. Камера на захвате работает на более близком расстоянии, обеспечивая повышенную детализацию. Разрешение выходных данных составляет 1920×1080 пикселей, частота кадров составляет 30 FPS.

Шаг 2: выполняется сегментация фруктов на изображениях с использованием нейронной сети YOLOv8 для уточнения трехмерных координат объектов. Архитектура нейронной сети аналогична сети, использованной на первом этапе, однако, модель, использованная на текущем этапе, обучена на изображениях фруктов, расположенных на более близком расстоянии.

Шаг 3: применяется метод преобразования Хафа для выделения границ объектов. Вычисляются центры масс и уточняются размеры фруктов.

Шаг 4: выполняется проверка цветовых характеристик распознанных фруктов в HSV-пространстве на соответствие эталонным значениям. Диапазон допустимых цветов сформирован на основе палитр RGB и HSV с учётом цветовых признаков спелых фруктов. Это позволяет на этапе фильтрации исключить посторонние объекты, ошибочно идентифицированные методом кругов Хафа из-за их круглой формы, а также незрелые фрукты, чьи цветовые характеристики отличаются от эталонных. Объекты с отклонениями по форме или размеру исключаются. Входными данными для алгоритма являются изображение (I_{RGB}) и массивы допустимых диапазонов цветов (R_{in} , R_{out}). Изображение представляет собой массив с информацией о значении атрибутов RGB для каждого из пикселей. Массив R_{in} включает допустимые диапазоны значений 6 цветовых атрибутов R, G, B, H, S и V пикселей фрукта. Массив R_{out} включает допустимые значения 4 цветовых атрибутов R, G, B и V пикселей фрукта. Атрибуты H и S не входят в массив R_{out} , так как для пикселей фрукта они могут принимать любое значение и не могут быть ограничены.

Входное изображение преобразуется в изображение (I_{GRAY}) в градациях серого цвета. С использованием преобразование Хафа на изображение (I_{GRAY}) определяются значения координат центров m окружностей и их радиусов, и

записываются в массив(c). Если массив(c) содержит хотя бы одну окружность, то выполняется фильтрация окружностей. Предварительно для алгоритма фильтрации из входного изображения (I_{RGB}) формируется массив (I_{HSV}) с информацией о значении атрибутов HSV для каждого из пикселей. Первый этап выполняет оценку цвета пикселя внутри и на контуре окружности. Если этот цвет не соответствует цвету фрукта, то алгоритм фильтрации удаляет эту окружность. Для каждого элемента массива (c) в функции ($Filter$) формируется первая часть массива (P_{bool}). В функции ($Filter$) с шагом t рассматриваются пиксели в окружности и их посчитанные координаты x_{pi} и y_{pi} записываются в n массивов (p_n):

$$x_{pi} = x_c + 0.7 * r * \cos(\alpha_i), \quad (2.27)$$

$$y_{pi} = y_c + 0.7 * r * \sin(\alpha_i), \quad (2.28)$$

где x_c , y_c , r – полученные значения в ходе работы алгоритма на основе преобразования окружностей Хафа. α_i – угол поворота с шагом t .

На пятом шаге формируется массив (P_{arr}) из n массивов (p_n). Каждый элемент массива (p_n) присваивается заданному параметру i_1 и j_1 , где $i_1=y_{pi}$ и $j_1=x_{pi}$. Для фильтрации формируется массив ($V_{1(arr)}$) из цветовых атрибутов R, G, B, H, S и V пикселей, координаты которых находятся в массивах (p_n). Полученный массив ($V_{1(arr)}$) сравнивается с массивом R_{in} , который имеет максимальные и минимальные значения цветовых атрибутов ($(R_{min}, R_{max}), (G_{min}, G_{max}), (B_{min}, B_{max}), (H_{min}, H_{max}), (S_{min}, S_{max}), (V_{min}, V_{max})$). Итог сравнения записывается в массив (V_{bool}). Подсчет количества соответствующих и ложных значений цветовых атрибутов производится в функции ($Boolarray$). Если количество ложных значений больше количества элементов массива (V_{bool}), то в массив (P_{bool}) записывается true, в ином случае false.

Для второй части массива (P_{bool}) также высчитываются координаты x_{pi} и y_{pi} , которые записываются в n массивов (p_n):

$$x_{pi} = x_c + r * \cos(\alpha_i),$$

$$y_{pi} = y_c + r * \sin(\alpha_i),$$

где x_c , y_c , r – полученные значения в ходе работы алгоритма на основе преобразования окружностей Хафа. α_i – угол поворота с шагом t .

Если элемент массива (P_{bool}) является true, то окружность, соответствующая этому элементу, удаляется из массива (c), который перезаписывается в массив (c'). С каждой удаленной окружностью корректируется параметр m . Псевдокод алгоритма определения плодов представлен на рисунке 2.10.

Algorithm 1 Hough circle detection

Input: I_{RGB}, R_{in}, R_{out}
Output: $c_{cor} = \{c_{1(cor)}, c_{2(cor)}, \dots, c_{n(cor)}\}$

```

1:  $I_{GRAY} = BGRTGRAY(I_{RGB})$ 
2:  $c = \{c_1, c_2, \dots, c_m\} = HoughCircles(I_{GRAY})$ 
3: if  $c \neq \emptyset$  then
4:    $I_{HSV} = RGBTOHSV(I_{RGB})$ 
5:   for  $i=0$  to  $m$  do
6:      $c_{coord} = \{c^{(0)}, c^{(1)}\}$   $\triangleright c_{coord}$  – array of two coordinates
7:      $r = \{c^{(2)}\}$   $\triangleright r$  – radius of the circle
8:      $P_{bool} = \{p_{1(bool)}, p_{2(bool)}, \dots, p_{k(bool)}\} = Filter(c_{coord}, r, I_{RGB}, I_{HSV})$ 
9:     for  $g = 0$  to  $k$  do  $\triangleright k$  – number of array elements
10:      if  $P_{bool}^{(g)} = true$  then
11:         $c' = c \setminus c^{(g)}$   $\triangleright c'$  – array of filtered circles
12:         $m = m - 1$ 
13:      end if
14:    end for
15:  end for
16:  for  $i=0$  to  $m$  do
17:     $c_{coord} = \{c^{(0)}, c^{(1)}\}$ 
18:     $r = \{c^{(2)}\}$ 
19:    for  $g=0$  to  $m$  do
20:       $d = \left( \left( c_{coord}^{(g+1)(0)} - c_{coord}^{(g)(0)} \right)^2 + \left( c_{coord}^{(g+1)(1)} - c_{coord}^{(g)(1)} \right)^2 \right)^{0.5}$ 
21:      if  $d < 0.8 * (r^{(g+1)} + r^{(g)})$  then
22:         $badC = \{0, 0\}$   $\triangleright badC$  – counter of unsuitable circles
23:        for  $f = 1$  to  $2$  do
24:           $P_{bool} = Filter(c_{coord}, r, I_{RGB}, I_{HSV})$ 
25:          for  $g = 0$  to  $k$  do
26:            if  $P_{bool}^{(g)} = true$  then
27:               $badC^{(f)} = badC^{(f)} + 1$ 
28:            end if
29:          end for
30:          if  $badC^{(2)} \geq badC^{(1)}$  then
31:             $c_{cor} = c' \setminus c^{(2)}$ 
32:          end if
33:          if  $badC^{(2)} < badC^{(1)}$  then
34:             $c_{cor} = c' \setminus c^{(1)}$ 
35:          end if
36:        end for
37:      end if
38:    end for
39:  end for
40:  if  $size(c_{cor}) > 0$  then
41:     $c_{cor} = \{c_{1(cor)}, c_{2(cor)}, \dots, c_{n(cor)}\}$ 
42:  end if
43: end if

```

Рисунок 2.10 – Псевдокод алгоритма определения фруктов с использованием преобразования Хафа.

На втором этапе фильтрации результатов выполняется сравнение соседних окружностей. Если одна из окружностей не соответствует эталонному цвету фрукта, то алгоритм фильтрации удаляет эту окружность. Алгоритм включает процедуру, которая позволяет исключить ошибки, связанные с некорректной идентификацией алгоритмом Хафа нескольких окружностей, соответствующих

одному и тому же плоду. Процедура предполагает попарную проверку расстояния между центрами идентифицированных окружностей по следующей формуле:

$$((x_2 - x_1)^2 + (y_2 - y_1)^2)^{0.5} < 0.8 * (r_2 + r_1),$$

где x_1, y_1, r_1 и x_2, y_2, r_2 – координаты центров и радиусы двух соседних окружностей соответственно.

Если условие выполняется, для каждой из двух окружностей выполняется проверка в функции Filter, и определяются пиксели, не соответствующие допустимому диапазону цветовых атрибутов. Если количество таких пикселей первой окружности больше либо равно количеству несоответствующих пикселей второй окружности, то удаляется первая окружность из массива (c), иначе - вторая. Итоговые результаты массива (c) перезаписываются в массив (c_{cor}). Если размер массива (c_{cor}) больше нуля, то алгоритм фильтрации выдает конечный результат и заканчивает программу. Псевдокод разработанного алгоритма фильтрации приведен на рисунке 2.11.

Algorithm 2 Filter

```

1: function FILTER( $c_{coord}, r, I_{RGB}, I_{HSV}$ )
2:    $t = 30$  ▷  $t$  - step angle
3:   for  $a_t = 0$  to 360 do
4:      $p_n = (c_m^{(0)} + 0.7 * \sin(a_t), c_m^{(1)} + 0.7 * \cos(a_t))$ 
5:      $V_{1(arr)} = \{I_{RGB}^{(p_n^{(0)})(p_n^{(1)})}, I_{HSV}^{(p_n^{(0)})(p_n^{(1)})}\}$ 
6:      $V_{bool} = \{V_{1(arr)} < R_{in}^{(min)} \text{ and } V_{1(arr)} > R_{in}^{(max)}\}$ 
7:      $B_{1(arr)} = \{B_1, B_2, \dots, B_n\} = Boolarray(P_{arr})$ 
8:     if  $B_{1(arr)} = true$  then
9:        $p_{bool} = true$ 
10:    end if
11:    if  $B_{1(arr)} = false$  then
12:       $p_{bool} = false$ 
13:    end if
14:  end for
15:  for  $a_t = 0$  to 360 do
16:     $p_n = (c_m^{(0)} + \sin(a_t), c_m^{(1)} + \cos(a_t))$ 
17:     $V_{2(arr)} = \{I_{RGB}^{(p_n^{(0)})(p_n^{(1)})}, I_{HSV}^{(p_n^{(0)})(p_n^{(1)})}\}$ 
18:     $V_{2(arr)} = V_{2(arr)} \setminus V_{2(arr)}^{(4)}$ 
19:     $V_{2(arr)} = V_{2(arr)} \setminus V_{2(arr)}^{(3)}$ 
20:     $V_{bool} = \{V_{2(arr)} < R_{out}^{(min)} \text{ and } V_{2(arr)} > R_{out}^{(max)}\}$ 
21:     $B_{2(arr)} = \{B_1, B_2, \dots, B_n\} = Boolarray(P_{arr})$ 
22:    if  $B_{2(arr)} = true$  then
23:       $p_{bool} = true$ 
24:    end if
25:    if  $B_{2(arr)} = false$  then
26:       $p_{bool} = false$ 
27:    end if
28:  end for
29: end function

```

Рисунок 2.11 – Псевдокод алгоритма фильтрации плодов при сравнении цветовых характеристик

Окружности, полученные после фильтрации, сопоставляются на пересечение с сегментированным контуром, полученным с использованием нейронной сети YOLO. В случае наличия пересечения формируется множество массив координат фруктов для последующего анализа на возможность включения в список подзадач РТС.

На шестом шаге трехмерные координаты фруктов, определенные фронтальной камерой, сопоставляются с данными камеры на захвате. Метрики схожести и фильтр Калмана используются для объединения данных и исключения дубликатов. Фильтр Калмана реализован с использованием следующей параметризации: процессный шум (Q) настроен на 0.01 для сглаживания данных, измерительный шум (R) составляет 0.1, что позволяет компенсировать небольшие ошибки сенсоров. Метрика схожести основывается на вычислении «Евклидова» расстояния между координатами объектов, где предельное значение для принятия совпадения установлено на уровне 10 мм. На основе этого корректируется список подзадач, сформированных на шестом шаге первого этапа.

Для обучения нейронной сети подготовлено три набора данных для деревьев, фруктов, расположенных на большом расстоянии, а также фруктов, расположенных на близком расстоянии. Каждый набор данных состоит из 1000 изображений, которые размечены в приложении LabelImg. В ходе обработки набора данных выполнена разметка каждого изображения: выделены области, содержащие деревья и фрукты. В процессе обучения модели применялись аугментации, такие как поворот, изменение яркости и насыщенности, для повышения устойчивости к изменяющимся условиям съёмки. Набор данных разделен на обучающую и тестовую выборки. При разметке для каждого изображения создается файл, в котором записаны координаты контура выделенной области. Объем обучающей выборки разделен на обучающую выборку и проверочные изображения в соотношении 70/30. Обучение нейронной сети проводилось на протяжении 100 эпох с использованием графического процессора

Nvidia GeForce RTX 3070. Графики обучения распознаванию фруктов приведены на рисунке 2.12.

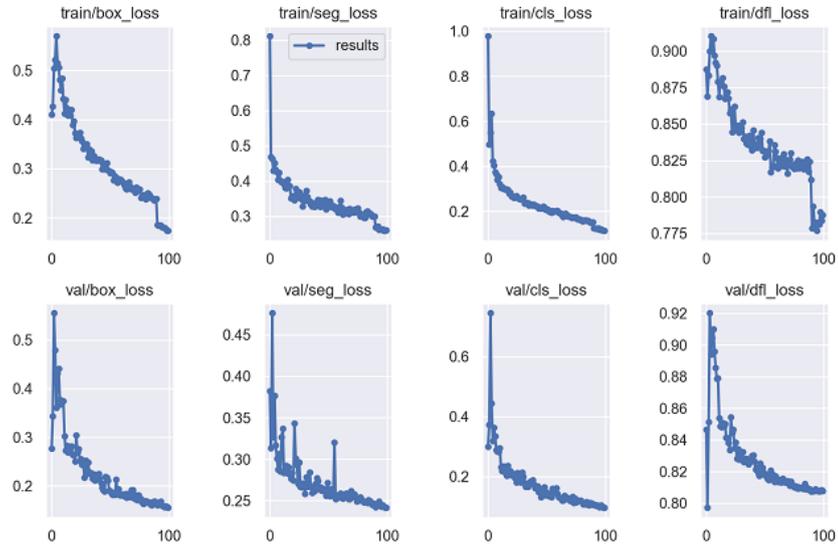


Рисунок 2.12 – Графики обучения нейронной сети.

По итогам обучения нейронной сети на протяжении 100 эпох получены графики, отражающие динамику потерь, точности и уверенности модели. Анализ графиков показал, что модель достигла состояния достаточной обученности, при котором дальнейшее увеличение количества эпох обучения не приводит к заметному улучшению точности или снижению потерь.

Проведено тестирование разработанной системы технического зрения на тестовом наборе данных. Результаты тестирования представлены на рисунке 2.13.



а)

б)



в)

г)

Рисунок 2.13 – Визуализация результатов определения фруктов на каждом этапе распознавания.

На рисунке 2.13 а показано исходное изображение фруктов на дереве. На рисунке 2.16 б показан результат определения фруктов с использованием нейронной сети YOLOv8. Как видно из рисунка, при определении фруктов при их близком расположении друг относительно друга возникает их объединение в один объект. Также при перекрытии распознаваемых объектов сегментацией изображения не удастся получить полноценную геометрию плода. На рисунке 2.13 в показан результат обработки изображения после применения преобразования методом кругов Хафа, что позволило уточнить размеры фруктов. На рисунке 2.13 г показано, что фрукт с несоответствующим цветом спелому исключен по итогу фильтрации.

2.5 Математическая модель захватного устройства

На основе сформулированных требований к захватному устройству, представленных в первом разделе диссертации, установлено, что конструкция с внутренней полостью и интегрированной гибкой трубой способна существенно повысить скорость сбора яблок за счет обеспечения непрерывного транспортирования плодов. Для отделения яблок от плодоножки требуется приложить силу в диапазоне от 24 до 65 Н. Учитывая их тонкую кожуру, важно обеспечить равномерное распределение этой силы, чтобы избежать повреждений. Одним из эффективных решений является увеличение количества пальцев захватного устройства, что позволяет разделить общую нагрузку на n равных частей, снижая риск локального давления на плод. Наиболее перспективным

методом отделения яблок является комбинированное воздействие, сочетающее одновременное отрывание и скручивание. Для реализации такого подхода предложено использовать принцип винтовой поверхности, аналогичный механизму шарико-винтовой передачи. Этот метод обеспечивает плавное и контролируемое отделение плодов, минимизируя механическое воздействие на дерево и плодоножку.

С учетом выявленных особенностей разработано схемно-техническое решение захватного устройства, предназначенного для сбора фруктов (рисунок 2.14). Предложенная конструкция сочетает в себе функциональность, надежность и возможность адаптации под различные типы плодов, что делает её универсальным инструментом для автоматизации процесса уборки урожая.

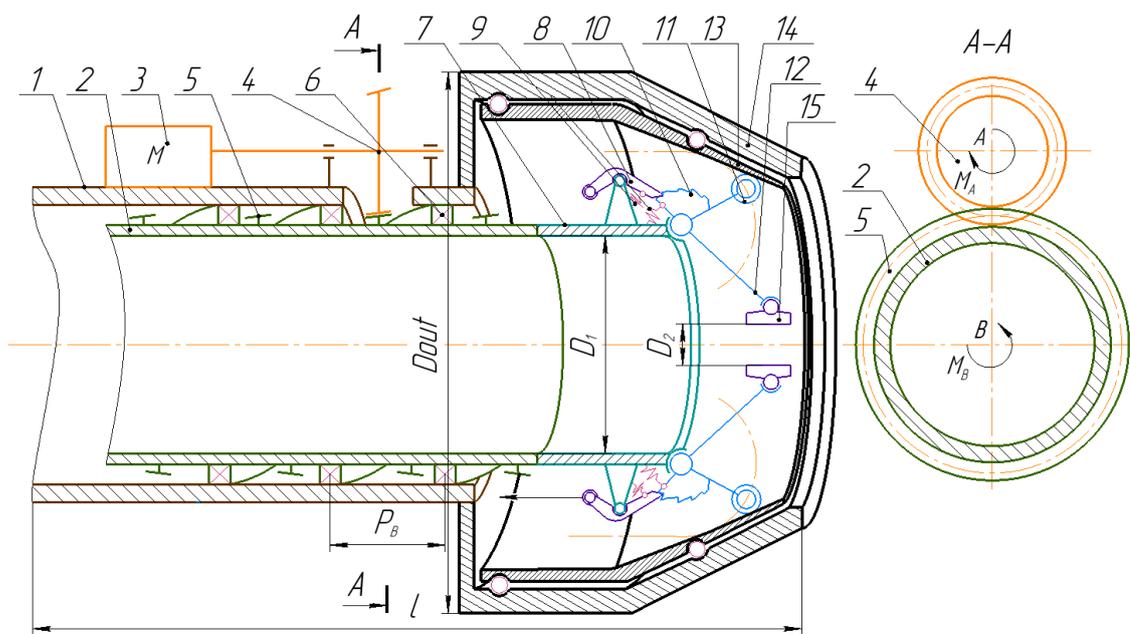


Рисунок 2.14 - Схема захватного устройства.

Захватное устройство состоит из внешней трубы 1 и внутренней трубы 2, которая вкручивается во внешнюю трубу. На внешней трубе 1 установлен шаговый двигатель 3, который вращает внутреннюю трубу 2 посредством зубчатой передачи 4. Внутренняя труба 2 на внешней стороне имеет зубчатый профиль 5, выполненный параллельно спирали шарико-винтовой передачи 6. На торце внутренней трубы 2 расположен корпус захвата 7, выполненный из 10 сегментов. Каждый сегмент представляет собой вращающийся палец 12 с закреплённым на

нём роликом 11 и храповым колесом 10. Поворот пальца против часовой стрелки блокируется фиксатором 8, которая прижимается к колесу 10 за счёт пружины 9. Сжатие пальцев происходит за счёт движения ролика 11 по внутренней поверхности обоймы конуса 13, которая вращается внутри конуса 14. На конце каждого пальца установлена мягкая губка 15, которая может поворачиваться в небольшом диапазоне.

Захватное устройство для сбора фруктов работает следующим образом: робот-трипод с телескопическим звеном позиционирует захватное устройство таким образом, чтобы фрукт оказался внутри конуса 14. Двигатель 3 вращает зубчатое колесо 4 и сопряжённую с ним внутреннюю трубу 2, которая совершает вращательно поступательное движение (скручивание). Пальцы 12, опираясь роликом 11 на внутреннюю поверхность обоймы конуса 13, сжимаются. В отличие от работы, посвящённой такому же захватному устройству [110] (рисунок 2.15, а), где ролик прижат к конусу 14 и движется по сложной винтовой траектории 16, добавлена обойма конуса 13, которая вращается вместе с трубой 2, корпусом захвата 7 и пальцами 12. Вследствие чего ролики 11 имеют однонаправленную траекторию, что значительно снижает силу трения (рисунок 2.15, б).

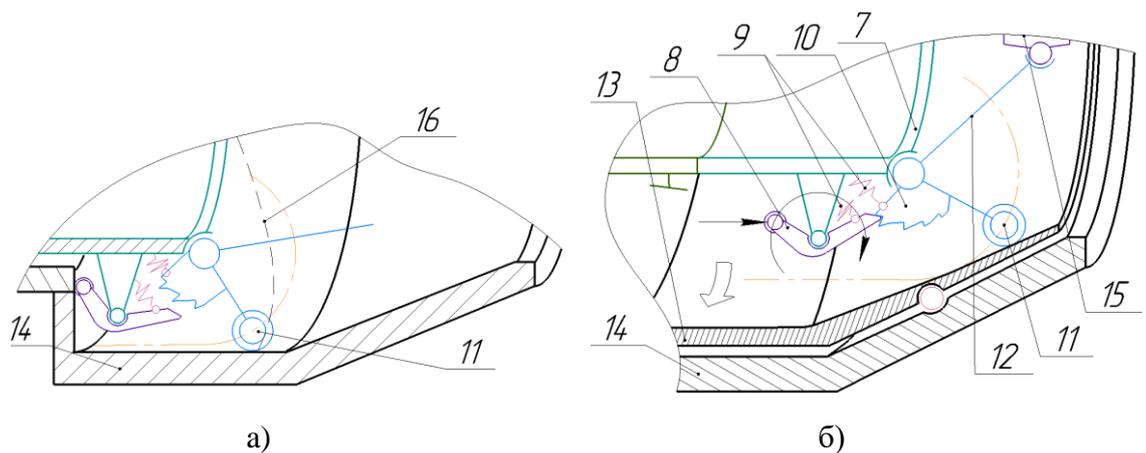


Рисунок 2.15 - Схема пальца захватного устройства: а) схема вращения пальца за счёт перемещения по поверхности конуса; б) схема вращения пальца за счёт перемещения по поверхности обоймы.

Движение ролика 11 по конической поверхности обеспечивает смыкание пальцев и фиксацию фрукта. Пальцы фиксируются храповым механизмом 8–10.

Отделение фрукта происходит за счёт закручивания внутренней трубы 2 в обратном направлении во внешнюю трубу 1. При достижении исходного положения фиксатором 8 упирается в корпус 14, освобождая храповое колесо 10, вследствие чего пальцы 12 разжимаются. Это позволяет, как собирать урожай классическим способом, складывая отдельно взятые фрукты в корзину, так и собирать фрукты непрерывно через мягкую трубу.

Важным этапом проектирования является определение силы сжатия фрукта, а также определение требуемого крутящего момента. Для определения этих параметров составим расчетную схему для пальца захватного устройства (рисунок 2.16).

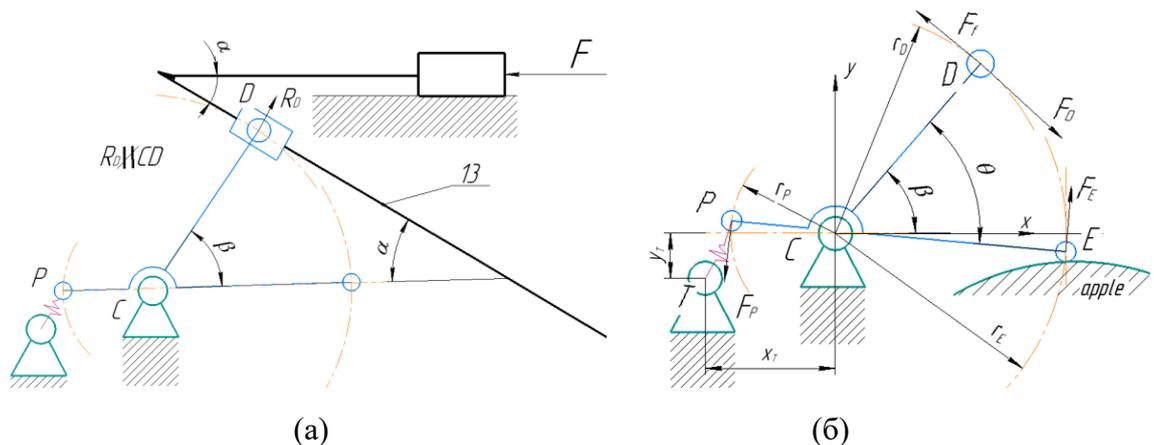


Рисунок 2.16 - Расчётная схема: а) схема взаимодействия пальца и конуса; б) схема действующих сил.

Представим механизм ЗУ в виде плоской схемы, где сила F является внешней силой. Сила F является результирующей поступательного движения внутренней трубы, которая прикладывается к одному пальцу. Поворот пальца относительно шарнира C осуществляется за счёт реакции конуса, которая возникает в результате действия силы ролика R_D . Рассмотрим систему в статическом равновесии, при этом считаем пружину не нагруженной, тогда определим реакция ползуна R_D на силу F по формуле

$$R_D = F \cos(90^\circ - \alpha - \beta), \quad (2.29)$$

где α — это угол конуса относительно оси, а β — угол поворота рычага. Тогда сила F_D , поворачивающая палец в результате действия силы F , определяется по следующей формуле

$$F_D = F \cos(90^\circ - \alpha - \beta) \cdot \cos(\alpha + \beta) \quad (2.30)$$

Захват имеет большое количество пальцев, из-за чего незначительная сила трения в одном пальце в совокупности оказывает значительную нагрузку на двигатель и может привести к заклиниванию. Определим силу трения на поверхности обоймы

$$F_f = f R_D, \quad (2.31)$$

где f - коэффициент трения.

Определим силу F_E поворота пальца (рисунок 2.16, б) по формуле

$$F_E = \frac{-F_D r_D + F_P r_P + F_f r_D}{r_E}, \quad (2.32)$$

где r_D, r_E, r_P — это радиусы поворота рычагов, а F_P сила упругости пружины рычаг. Для начала определим силу самой пружины, учитывая, что вектор силы действия пружины будет направлена перпендикулярно оси захватного устройства, тогда выражение F_P принимает следующий вид

$$F_P = -k \cdot \Delta x \cdot \cos(\theta - \beta), \quad (2.33)$$

где k коэффициент упругость, Δx разница между исходным и растянутым положением пружины при повороте рычага.

С учётом того, что Т — это точка крепления пружины в шарнире С, то F_P имеет вид

$$F_P = \left(-k \cdot \sqrt{(r_P \cos(\theta - \beta) + x_T)^2 + (r_P \sin(\theta - \beta) - y_T)^2} \right) \cdot \cos(\theta - \beta) \frac{r_P}{r_E}, \quad (2.34)$$

где θ - угол между пальцем и роликом захватного устройства, а x_T, y_T - координаты точки крепления пружины на корпусе.

С другой стороны, определим силу F с учетом момента двигателя M .

$$F = \frac{M 2\pi z_2 \eta}{P_B z_1 n} \quad (2.35)$$

Следовательно, принимая во внимание выражения (2.29–2.35), определим крутящий момент как

$$M = \frac{\frac{F_E r_E + F_P r_P + F_f r_D}{r_D \cos(90^\circ - \alpha - \beta) \cdot \cos(\alpha + \beta)} P_B z_1 n}{2\pi z_2 \eta}, \quad (2.36)$$

где η коэффициент полезного действия, принимаемый для шарико-винтовой передачи 0,88 [111]; n количество пальцев захватного устройства; P_B – шаг винтовой поверхности; z_1, z_2 число зубьев трубы 2 и приводного колеса 4 соответственно. С учётом наибольшего диаметра фрукта определим число зубьев z_1 на поверхности внутренней трубы 7 (рисунок 2.14).

$$z_1 = \frac{2,2m + D_1}{m}, \quad (2.37)$$

где m – это модуль зубчатой передачи, который необходимо минимизировать для получения меньшего габарита захватного устройства. Принимаем модуль $m = 4$, с учетом дальнейшего изготовления при помощи 3D печати. Исходя из выражения (2.35) полученное значение z_1 будет являться минимальным для сбора фруктов диаметром D_1 , таким образом, выбираем ближайшее большее значение z_1 с учетом модуля m исходя из ISO 1328-1-2017, получим оптимальное число зубьев $z_1 = 50$. Наибольшее передаточное отношение z_2/z_1 достигается, когда z_2 принимает минимальное значение, равное 10, согласно таблице стандартных зубчатых передач ISO 1328-1:2017.

Чем меньше шаг винтовой поверхности, тем меньше требуется крутящий момент, однако в описанной компоновке между витками винтовой поверхности расположено зубчатое зацепление. При проектировании макета ЗУ из пластика принято значение шага $P_B = 0,052$ м. Описанное захватное устройство может быть использовано для любых фруктов круглой формы, однако из-за существенных различий в размере, кожуре, способе отделения и требуемом усилии отделения могут быть использованы различные геометрические параметры. Для определения оптимальных геометрических параметров проведём многокритериальную оптимизацию.

2.6 Оптимизация параметров захватного устройства

Рассмотрим оптимизацию параметров захватного устройства для сбора фруктов. Для этого найдём множество решений с различными конструктивными параметрами механизма при заданных ограничениях. Найдём из всего множества те, которые будут иметь минимальные значения: момента M , наружного диаметра корпуса D_{out} и длины внутренней трубы l .

Постановка задачи оптимизации:

1. Критерии оптимизации:

а) момент двигателя стремится к минимуму $M \rightarrow \min$. Момент определяется по формуле (2.36);

б) наружный диаметр корпуса стремится к минимуму $D_{out} \rightarrow \min$, который определим по следующей формуле

$$D_{out} = D_1 + 2r_D \sin \theta; \quad (2.38)$$

в) длина трубы $l \rightarrow \min$, которую определим по следующей формуле

$$l = 3P_B. \quad (2.39)$$

2. Параметры оптимизации: радиус закрепления пружины на пальце r_p , длина пальца r_E , угол конуса относительно оси захватного устройства α° , угол между пальцем и рычагом кулачка θ° .

3. Ограничения оптимизации: радиус крайней точки пальца не должен превышать радиус наибольшего фрукта $D_1 = 120$ мм. При этом радиус крайней точки должен быть больше разности большего и меньшего фрукта $D_2 = 40$ мм, тогда $\frac{D_1 - D_2}{2} < r_E < \frac{D_2}{2}$. Диаметры яблок выбраны исходя из наиболее распространенных сортов, диаметр которых редко превышает 120 мм. Это неравенство позволит исключить варианты, когда пальцы сомкнутся между собой или в обратном случае, когда пальцы не дотянутся до фрукта малого диаметра. Угол между пальцем CE и рычагом CD ограничим как $0 < \theta < 90 - \alpha$, если условие не будет выполняться, вектор силы поворачивающий ролик будет равен нулю, при этом $0^\circ < \alpha < 90^\circ$. Ограничим длины рычагов следующим неравенством $0 < r_p < r_D$, так как только ролик D должен касаться поверхности

конуса в противном случае механизм заклинит, при этом длина рычага r_D будет равна

$$r_D = \frac{P_B}{\cot \alpha \sin \theta}.$$

Жёсткость пружины может принимать значения $k_{min} < k$, где k_{min} – это жесткость пружины, обеспечивающая поворот пальца в исходное состояние. В данной работе значение принято $k_{min} = 2$ Н/м, исходя из экспериментальных данных. Палец поворачивается на угол β , при этом $0 \leq \beta \leq \theta$.

Выбор исходного параметра силы сжатия захватного устройства F_E , производился исходя из сформулированных требований. Увеличение количества пальцев снижает давление на отдельно взятую область фрукта, так как требуемая сила удержания будет распределена между n пальцами. Исходное положение пальца, это когда рычаг PE расположен параллельно оси захватного устройства, в котором пружина полностью расслаблена. Тогда координаты точки крепления пружины $x_T = r_P$; $y_T = 0$.

Для решения поставленной задачи оптимизации используем один из пакетов MATLAB «gamultiobj», который позволяет использовать генетический алгоритм для решения задачи многокритериальной оптимизации [112-115]. Для формирования множества Парето и сокращения количества критериев предлагается объединить критерии - максимальный диаметр D_{out} и - максимальную длину устройства l в единый критерий — объём V_{out} , занимаемый захватным устройством. Такой подход позволяет упростить оценку конструкции, сохраняя при этом значимость габаритных характеристик используя следующее выражение для объема V_{out} цилиндра

$$V_{out} = \frac{l\pi D_{out}^2}{4}.$$

Результат расчёта представлен в виде множества Парето (рисунок 2.17, а). На графике видно, что распределение оптимальных значений имеет множество значений как при большом крутящем моменте и компактном захватном устройстве, так и при малом крутящем моменте и больших габаритах. Из всего набора

выбирается значение, при котором крутящий момент максимально низок, а объем остается в пределах, приемлемых для изготовления. Выбранная из множества Парето конфигурация имеет следующие значения параметров: $r_P = 0.005$ м, $r_E = 0.0587$ м, угол конуса $\alpha = 37.4^\circ$, жесткость пружины $k = 2.815 \cdot 10^4$.

Выполнен расчет требуемого крутящего момента двигателя ЗУ с выбранными параметрами по формуле (2.36). Получен график требуемого крутящего момента в зависимости от угла поворота пальцев (рисунок 2.17, б). Исходя из графика, следует, что в исходном положении, когда пальцы параллельны трубе, крутящий момент значительно выше, чем при сжатии пальцев. Эта зависимость не является линейной и позволяет оптимизировать конструкцию. Если учесть, что большинство фруктов имеют диаметр около 80–90 мм, то чаще всего захватное устройство будет работать в пределах сгибания пальцев от 10 до 30 градусов, с минимальным крутящим моментом. Максимальный требуемый крутящий момент двигателя составляет 0.73 Н·м.

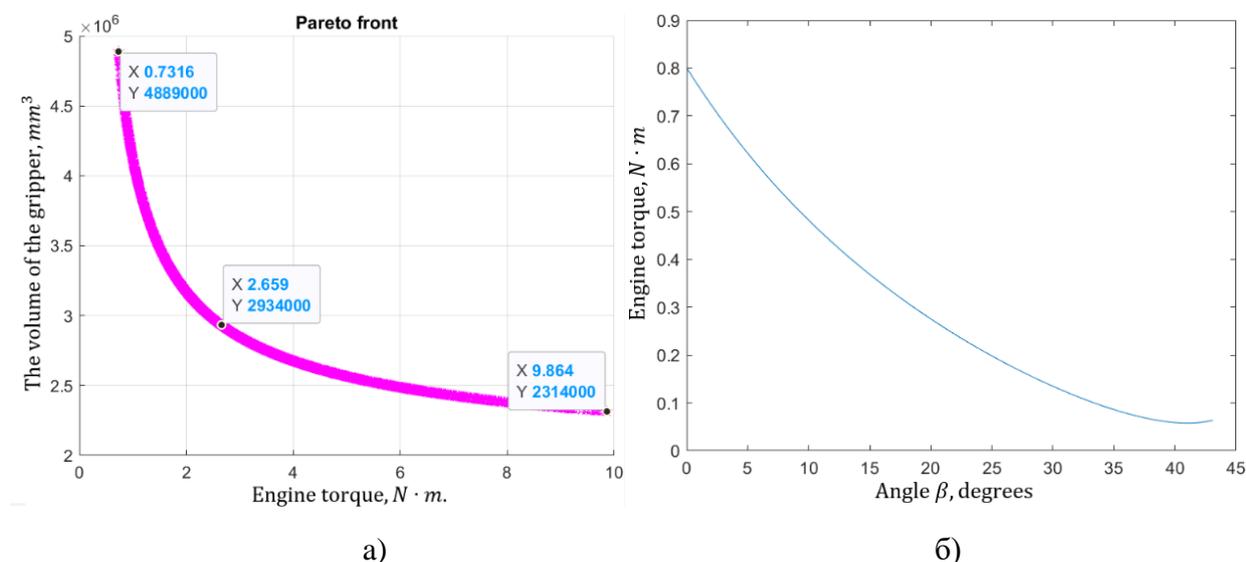


Рисунок 2.17 - Результаты оптимизации: а) множество Парето; б) график крутящего момента двигателя в зависимости от смыкания пальцев.

Выводы по главе

1. Предложена модульная структура робототехнической системы для сбора фруктов, выполненная на базе мобильной колёсной платформы и работа-

трипода, в котором для исключения избыточных степеней свободы и повышения управляемости использована пассивная центральная кинематическая цепь с выдвижным телескопическим звеном, а также захватное устройство с гибкой трубой для транспортировки и складирования фруктов.

2. Предложена структура и параметризованная модель специального захватного устройства, позволяющего комбинировать методы отрывания и скручивания фруктов за счёт наличия на корпусе винтовой поверхности, по которой перемещается внутренняя труба, имеющая на одном торце пальцы, обеспечивающие захват и отделение фруктов.

3. Разработана математическая модель кинематики робототехнической системы, описывающая зависимость положения телескопического звена с захватным устройством от углов ориентации подвижной платформы робота-трипода, учитывающая положения шарниров крепления кинематических цепей робота-трипода к платформам в виде систем нелинейных уравнений, определяющих ограничения на множество достижимых положений и ориентаций выходного звена и допустимых диапазонов движения линейных приводов.

4. Синтезирован эвристический алгоритм многокритериальной оптимизации параметров робота-трипода, на первом этапе которого выполняется нормализация критериев, а на втором этапе - итеративная процедура оптимизации на основе однокритериальных эволюционных алгоритмов с использованием свёртки критериев с переменными коэффициентами важности. Использование в качестве критериев компактности конструкции и критерия, зависящего от количества недостижимых ориентаций платформы при заданных ограничениях на диапазоны приводных штанг, и отсутствие пересечений звеньев позволяет расширить функциональные и эксплуатационные характеристики робототехнической системы.

5. Синтезирован двухэтапный алгоритм технического зрения для локализации деревьев и распознавания фруктов на основе интеграции нейросетевых алгоритмов и преобразования Хафа; на первом этапе алгоритма

выполняется формирование списка заданий на основе идентификации крон деревьев, а на втором - актуализация списка заданий на основе идентификации и сегментации плодов с учётом их спелости, что позволяет на основе визуальных данных комбинировать автономную навигацию и идентификацию координат объектов.

ГЛАВА 3. РАЗРАБОТКА ЭЛЕКТРОННО-ЦИФРОВОЙ МОДЕЛИ И ИМИТАЦИОННОЕ МОДЕЛИРОВАНИЕ

В данной главе рассматриваются вопросы проектирования основных конструктивных узлов РТС с использованием методов автоматизированного проектирования (САПР). Особое внимание уделяется корректировке параметров РТС, полученных в результате оптимизации геометрических характеристик робота-трипода и захватного устройства, описанных во второй главе. При проектировании учитываются такие критерии, как технологичность и эргономичность механизмов, силовые параметры, а также требования к собираемости и возможности изготовления компонентов. Подход направлен на обеспечение сбалансированности конструкции с точки зрения производственной реализуемости и эксплуатационной эффективности.

3.1 Имитационное моделирование робота-трипода с телескопическим звеном в динамической среде

Обратное динамическое моделирование служит ключевым инструментом для оценки динамических характеристик и реализации эффективного управления параллельными роботами. В рамках данного подхода определяются значения скоростей и ускорений шарниров в кинематических цепях, что позволяет установить соотношение между активными и пассивными движениями робота-трипода. На основе полученных данных рассчитываются скорости и ускорения центров масс звеньев, что обеспечивает комплексный анализ динамического поведения системы и формирует основу для её оптимизации [116-119]. Проведение имитационного моделирования в динамической среде позволяет оценить поведение системы в условиях, меняющихся со временем [120]. Это особенно важно для изучения движений робота-трипода в условиях сбора фруктов, где требуется адаптация к изменениям окружающей среды. Такое моделирование даёт возможность: тестировать реакции системы на различные сценарии и факторы

воздействия, улучшать прогнозирование эффективности работы в реальных динамических условиях, определять оптимальные стратегии управления и взаимодействия в условиях постоянных изменений, устранить возможные ошибки на ранних стадиях проектирования.

Для имитационного моделирования кинематических и динамических параметров под действием рабочих нагрузок в условиях эксплуатации механизма в программе MSC Adams разработана имитационная модель робота-трипода с телескопическим звеном, представленная на рисунке 3.1. Модель состоит из основных элементов, таких как неподвижная платформа A_1, A_2, A_3 , подвижная платформа B_1, B_2, B_3 привод AB , состоящий из штоков и поршней, штоки C_1 и C_2 и захватное устройство [121,122].

Для удобства модель создана в цилиндрической системе координат, поэтому точки крепления шарниров к основанию и рабочей платформе задаются с помощью радиусов R_1 и R_2 окружности и угловых положений точек $A_1 - A_3$ и $B_1 - B_3$. При вводе новых данных модель автоматически перестраивается в соответствии с наложенными ограничениями.

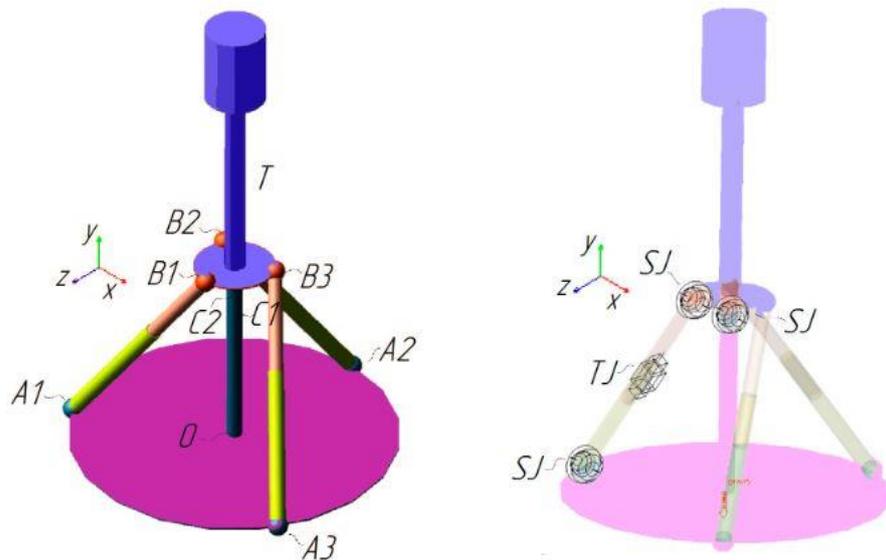


Рисунок 3.1 - Цифровая имитационная модель робота-трипода, выполненная в MSC Adams: A_1-A_3, B_1-B_3 –маркеры соответствующих шарнирных соединений; C_1 – стержень, жёстко соединённый с неподвижной платформой; C_2 - стержень, жёстко соединённый с подвижной платформой; T – захватное устройство; SJ – сферический шарнир; TJ – поступательный соединение; O - начало координат.

Для имитации движения привода (АВ) поршень (АІ) и шток (АО) соединены шарниром. Аналогично соединены звенья телескопического захвата (Т). В качестве внешней силы задана вертикальная, направленная вниз (осевое направление ОУ) сила тяжести G. Для имитации движения робота-трипода задаются поступательные движения актуаторов с помощью команды TranslationJointMotion, которая реализует законы поступательного движения в имитационной модели. Временная зависимость движения задаётся с помощью специальной функции STEP [123].

Для учёта массовых и инерционных характеристик исследуемого объекта в модели заданы массы всех элементов, соответствующие реальным компонентам проектируемого образца, а также сила тяжести и учтена полезная нагрузка (максимальная масса перемещаемого объекта) в выходном звене исполнительного органа. В таблице 3.1 представлены основные функциональные параметры робота-трипода для проведения оптимизации.

Таблица 3.1 - Основные технические характеристики исследуемого объекта

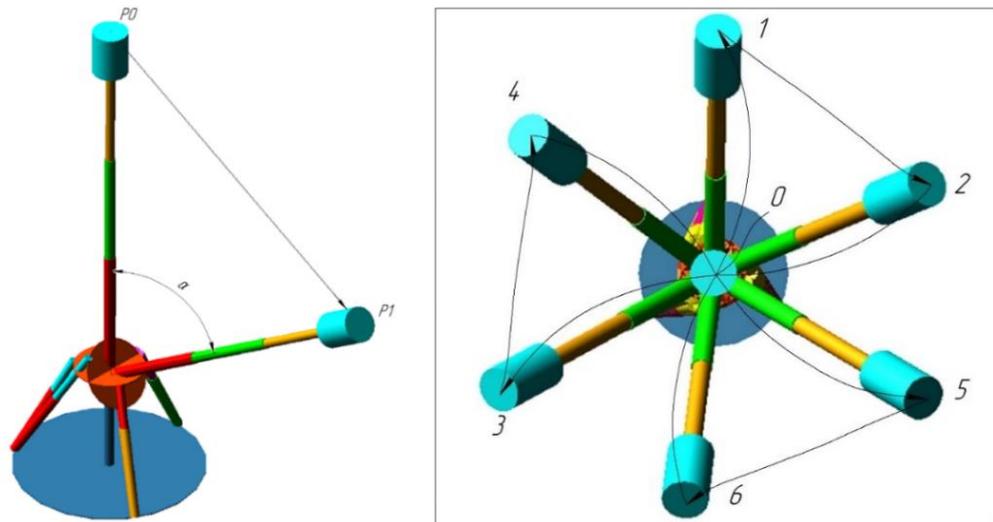
Максимальный ход штока ЭЦ	мм	1400
Максимальная скорость ЭЦ	мм/с	100
Максимальное ускорение ЭЦ	мм/с ²	5000

Точки сбора фруктов выбирались на основе средних геометрических параметров дерева. Координаты точек сбора фруктов приведены в таблице 3.2.

Таблица 3.2 - Координаты точек сбора фруктов и геометрические параметры конструкции (длины актуаторов) для необходимых перемещений исполнительного органа

№ точки	X	Y	Z	<i>l1</i>	<i>l2</i>	<i>l3</i>	<i>l_t</i>
1	-1400	3500	-1100	1260	1130	1300	2000
2	-1400	1100	1100	1150	1100	1420	700
3	-1400	3500	1100	1240	1240	1380	2000
4	-1400	3500	-1100	1260	1130	1300	2000
5	-1400	1100	-1100	1210	1040	1330	700
6	-1400	3500	1100	1240	1240	1380	2000

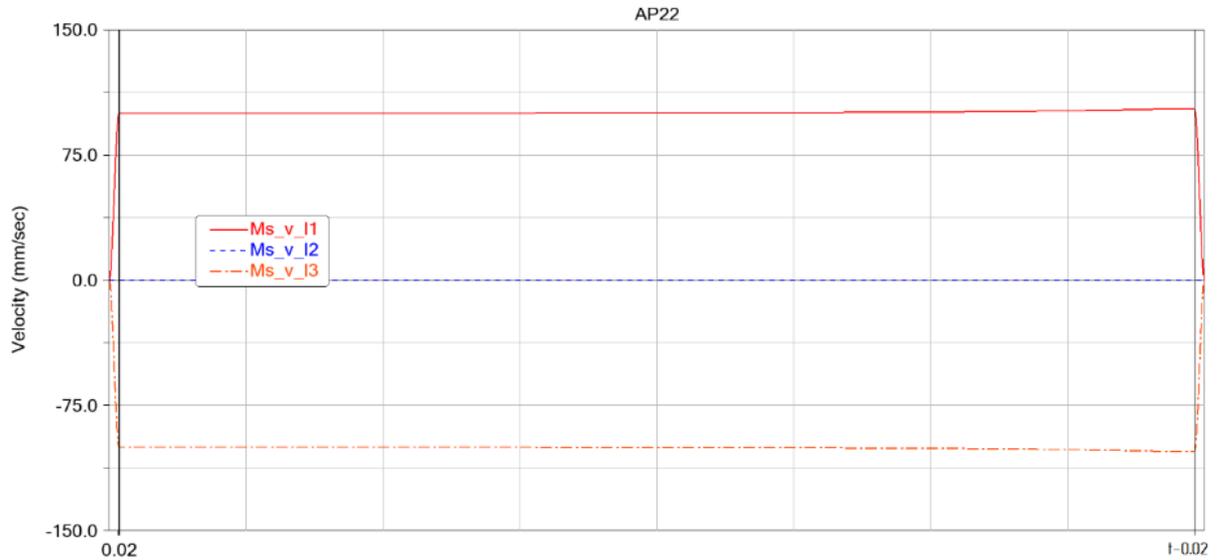
При проведении оптимизации по координатам точек сбора фруктов обрабатывается рабочая траектория (Рис. 3.2).



Рисунке 3.2 - Траектория движения робота-трипода.

Траектория выбрана на основании данных о параметрах рабочей зоны, а также кинематических и динамических характеристик применяемых приводных устройств (актуаторов). Данная траектория соответствует максимальному наклону подвижной платформы к уровню горизонта, которая учитывает наиболее нагруженное состояние в элементах конструкции робота-трипода. Для отработки, выбранной траекторий использованы максимально возможные скоростные характеристики выбранных актуаторов. В качестве исходного положения выбрана позиция робота-трипода, при которой подвижная платформа располагается горизонтально. При этом точки крепления актуаторов подвижной платформы имеют те же угловые коэффициенты, что и точки крепления актуаторов неподвижной платформы. Стержни C_1 и C_2 имеют значения длин 900 мм и 100 мм соответственно. Захватное устройство выдвинуто в максимально возможное рабочее состояние. При изменении длины актуаторов происходит поворот подвижной платформы на максимальный угол (α) относительно сферического шарнира (C). Актуаторы изменяют свою скорость в соответствии с графиком, представленным на рисунке 3.3. Из начального положения (0 мм/с) скорость актуаторов AB_1 и AB_3 возрастает до максимального значения (100 мм/с) за 0,02 секунды и остаётся постоянной на протяжении всего цикла работы актуатора, за

исключением торможения. За 0,02 секунды до остановки актуатор начинает совершать обратную работу и останавливается.



Рисунке 3.3 – График скорости движения штока в актуаторах.

В таблице 3.3 приведены пиковые значения результирующей суммы сил, приложенных к шарнирам A_1 - A_3 , B_1 - B_3 и C соответственно. MSC Adams предоставляет подробную информацию о состоянии механизма. В таблице 3.3 приведён пример, основанный на анализе 1000 значений результирующей суммы сил. Максимальное значение результирующей суммы сил на всех шарнирах происходит в момент на 30,6 секунде (выделено в таблице 3.3). В этот момент подвижная платформа останавливается при перемещении из точки P_4 в точку P_5 . Максимальную нагрузку испытывает центральный сферический шарнир C на 30,6 секунде симуляции (таблица 3.3).

Таблица 3.3 - Пиковые значения суммы сил, приложенных к шарнирам A_i , B_i

Время	$M_s_SF_A_1$	$M_s_SF_A_2$	$M_s_SF_A_3$	$M_s_SF_B_1$	$M_s_SF_B_2$	$M_s_SF_B_3$	$M_s_SF_C$
1.1	665.7	1168.7	695.2	908.9	934.7	919.4	1626.1
9.9	3152.9	3206.3	11236.1	2910.6	3420.5	11465.4	12747.6
11.9	838.5	268.3	2983.3	594.8	86.6	3213.7	3288.0
20.9	2203.4	1650.9	3136.4	1971.1	1856.0	3366.5	4344.3
30.6	19933.4	20783.8	17846.5	20190.8	20563.6	18074.5	31721.0
32.6	11082.4	11814.1	9774.9	11337.7	11595.5	10004.5	17772.7
34.6	786.6	298.8	958.9	553.8	488.9	1189.2	1597.0
42.5	1428.7	585.8	2237.9	1197.0	789.5	2466.1	2894.7

Таблица 3.4 - Пиковые значения суммы сил (Н) в проекции на оси OX, OY, OZ приложенных к наиболее нагруженному шарниру.

Время	OX	OY	OZ	M
0,92	-199,29	-952,08	743,95	1224,60
9,88	2677,86	-3467,27	595,58	4421,27
13,26	380,87	-794,12	27,69	872,68
20,94	1719,17	-1692,40	-0,38	2412,42
30,60	12,89	-982,18	677,97	2332,30
32,62	-286,27	-1118,57	1458,56	1860,26
42,50	700,68	-986,51	224,00	1230,58

Одной из задач, решаемых с использованием разработанной динамической модели, является определение оптимальных параметров взаимного расположения опорных точек. Эти параметры должны обеспечивать требуемые угловые перемещения исполнительного органа при минимизации значений реакции опор. В ходе исследования проведён полный 2-х факторный эксперимент 2-го порядка. Для этого построен ротатабельный центральный композиционный план (РЦКП) второго порядка. Выходным параметром функции отклика $F = f(s, d)$ является значение максимальной результирующей суммы сил (F), которая возникает в точке крепления шарнира к основанию при отработке самой нагруженной траектории. В качестве факторов эксперимента выбраны: радиус неподвижной платформы - R_1 и радиус подвижной платформы R_2 . В соответствии с изменяемыми факторами и схемой двухфакторного пространства (рисунок 3.4) построена матрица 2-х факторного эксперимента второго порядка

Таблица 3.5 представлены условия эксперимента.

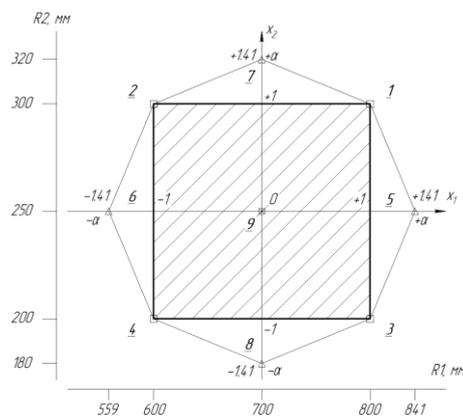


Рисунок 3.4 - Схема двухфакторного пространства для РЦКП второго порядка

Таблица 3.5 - Условия экспериментов

Обозначение факторов		Уровни факторов										Интервалы варьирования факторов
		“Звёздный”		Верхний		Основной		Нижний		“Звёздный”		
натуральное	кодирование	натуральное	кодированное	натуральное	кодированное	натуральное	кодированное	натуральное	кодированное	натуральное	кодированное	
R1, мм	x_1	841	+a	800	+1	700	0	600	-1	559	-a	100
R2, мм	x_2	320	+a	300	+1	250	0	200	-1	180	-a	50

В таблице 3.6 представлены изменяемые параметры имитационной модели совместно с геометрическими характеристиками R_1 и R_2 выраженными в мм (рисунок 3.5).

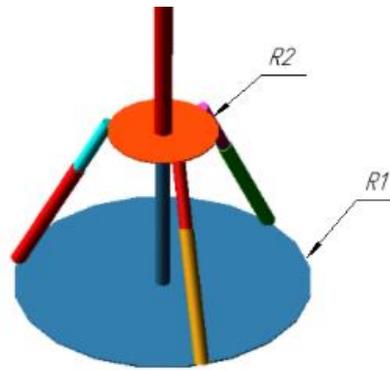


Рисунок 3.5 - Имитационная модель с изменяемыми параметрами.

Таблица 3.6. Изменяемые параметры имитационной модели.

№1 R1=800 R2=300	№2 R1=600 R2=300	№3 R1=800 R2=200
№4 R1=600 R2=200	№5 R1=841 R2=250	№6 R1=559 R2=250
№7 R1=700 R2=320	№8 R1=700 R2=180	№9 R1=700 R2=250

В таблице 3.7 представлена матрица ротатбельного униформ-планирования второго порядка применительно к задачам исследования. В таблице x_1 и x_2 это закодированные значения радиусов верхней и нижней платформ, необходимые для простоты расчета, а \tilde{x}_1 и \tilde{x}_2 натуральные значения радиусов верхней и нижней платформ.

Таблица 3.7. Матрица ротатбельного униформ-планирования второго порядка.

№ опыта	x_0	x_1	\widetilde{x}_1	x_2	\widetilde{x}_2	x_1x_2	x_1^2	x_2^2	y_i
1	+	+	800	+	300	+	+	+	y_1
2	+	-	600	+	300	-	+	+	y_2
3	+	+	800	-	200	-	+	+	y_3
4	+	-	600	-	200	+	+	+	y_4
5	+	+1,41	841	0	250	0	2	0	y_5
6	+	-1,41	559	0	250	0	2	0	y_6
7	+	0	700	+1,41	320	0	0	2	y_7
8	+	0	700	-1,41	180	0	0	2	y_8
9	+	0	700	0	250	0	0	0	y_9

В ходе динамического моделирования выполнялась фиксация силовых значений, возникающих в шарнирах в течение заданного времени и заносилась в таблицу 3.8.

Таблица 3.8. Пиковые значения результирующих сумм сил (реакций).

Номер опыта	1	2	3	4	5	6	7	8	9
Выходной параметр, y_i	y_1	y_2	y_3	y_4	y_5	y_6	y_7	y_8	y_9
F, Н	4800	6022	4868	7877	4660	6382	5320	7090	5325

После проведения вычислений коэффициентов регрессии математическая модель распределения пиковых значений суммы сил от радиусов подвижной и неподвижной платформы будет иметь вид:

$$F = 6457 - 722 \cdot x_1 - 479 \cdot x_2 + 387 \cdot x_1 \cdot x_2 - 831 \cdot x_1^2 - 535 \cdot x_2^2 \quad (3.1)$$

где F - пиковое значение суммы сил; x_1 - 1-й фактор – радиус неподвижной платформы, R_1 ; x_2 - 2-й фактор – радиус подвижной платформы, R_2 . Полученные в результате проведения эксперимента данные представлены на рисунке 3.6.

После раскодирования уравнение регрессии будет иметь вид:

$$F = -0,0831R_1^2 + 8,977R_1 + 0,0774R_1 \cdot R_2 + 43,24R_2 - 0,214R_2^2 - 26643. \quad (3.2)$$

Зависимость пиковых значений суммы сил в центральном сферическом шарнире (С) от радиусов подвижной и неподвижной платформ представлена в виде трёхмерного графика и в виде цветовой карты на рисунке 3.7.

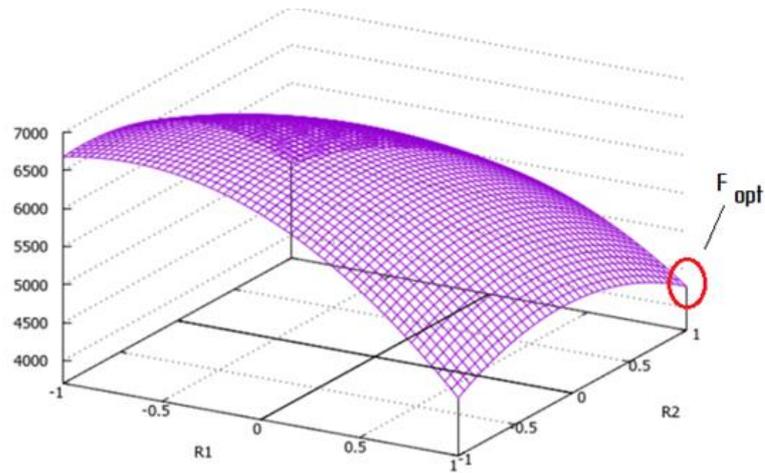


Рисунок 3.6 - График выходных параметров пикового значения результирующей суммы сил.

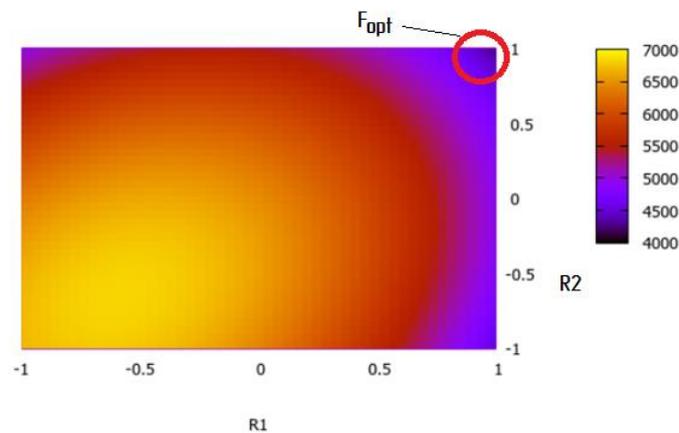


Рисунок 3.7 - Функция отклика и цветовая карта распределения пиковых значений суммы сил в шарнире (С) от радиусов подвижной и неподвижной платформ.

Из графиков следует, что минимальная сила возникает при наибольших диаметрах верхней и нижней платформ, используемых в эксперименте. Это соответствует радиусу нижней платформы 800 мм и верхней платформы 300 мм. Для детального изучения распределения сил в остальных шарнирах проведена серия моделирований выбранной конфигурации платформ. В результате моделирования получен график, отображающий действующие силы на все шарниры робота-трипода (рисунок 3.8).

На графике показаны изменения результирующих силовых значений в шарнирах основания A_1-A_3 , шарнирах подвижной платформы B_1-B_3 , а также в центральном сферическом шарнире С, где фиксируется наибольшая нагрузка. Дополнительно на графике (рисунок 3.9) представлена разбивка действующей

силы на шарнир C по проекциям на оси OX , OY и OZ . Из графика видно, что предельная сила возникает вдоль пассивной кинематической цепи при растяжении шарнира.

Полученные результаты демонстрируют распределение нагрузок в ключевых узлах конструкции и служат основой для проведения топологической оптимизации. Это позволит улучшить прочностные характеристики и минимизировать массу элементов без снижения их функциональности.

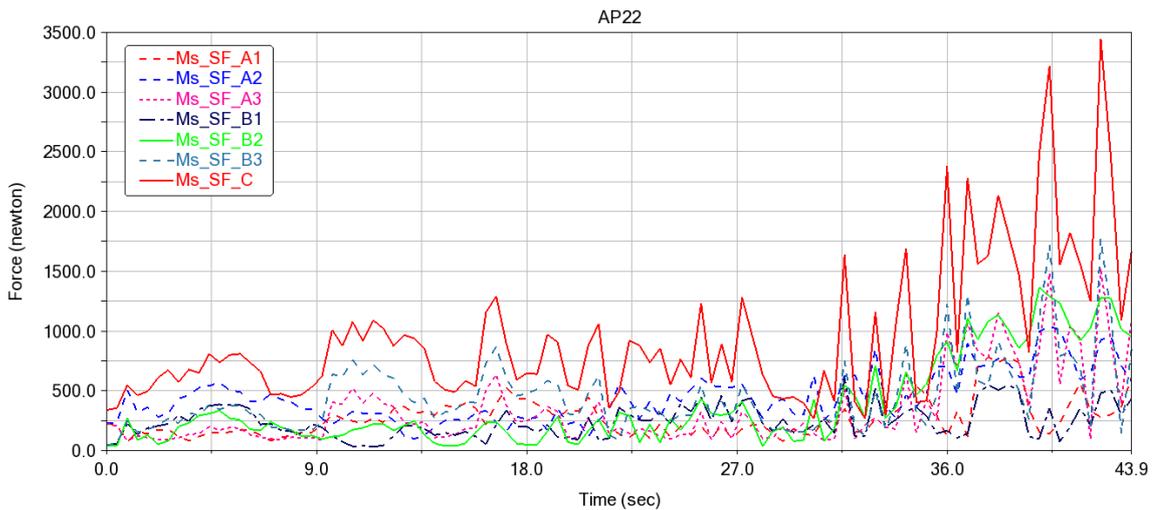


Рисунок 3.8 - Изменение результирующих силовых значений в шарнирах основания A_1 - A_3 , шарнирах подвижной платформы B_1 - B_3 и в центральном сферическом шарнире C , при проведении симуляции с оптимизированными параметрами.

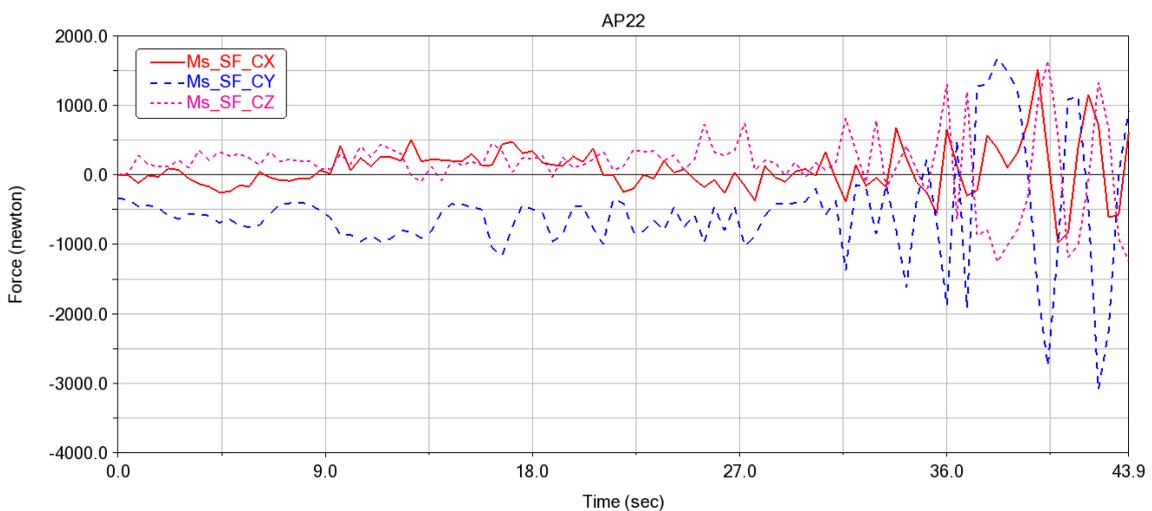


Рисунок 3.9 - Изменение суммы сил в проекции на оси OX , OY , OZ , приложенных к наиболее нагруженному центральному сферическому шарниру при проведении симуляции с оптимизированными параметрами.

3.2 Разработка цифрового двойника робота-трипода

Цифровые двойники играют ключевую роль в современном проектировании и разработке роботизированных систем. Они позволяют создавать виртуальные копии реальных устройств, точно воспроизводя их поведение, характеристики и работу в различных условиях. Это особенно важно для сложных механизмов, таких как РТС для сбора фруктов, где требуется высокая точность и надёжность. Использование цифровых двойников позволяет проводить анализ, оптимизацию и тестирование системы на этапе проектирования, что значительно снижает затраты на физическое прототипирование.

На основе результатов динамического моделирования построим цифровой двойник робота-трипода с использованием CAD/CAE системы NX. При моделировании важно сохранить соответствие геометрических параметров механизма и кинематической схемы. Ключевым элементом робота-трипода является электроцилиндр (ЭЦ). Существуют готовые решения ЭЦ различного типоразмера и грузоподъёмности. В разделе 2, в результате оптимизации для заданных параметров, таких как высота деревьев и желаемый объём рабочей области, определена максимальная длина ЭЦ равная 999,8 мм, и минимальная длина в сложенном состоянии 580,5 мм. Представленные размеры откладываются от центра вращения шарнира, следовательно, для сохранения полученных кинематических параметров необходимо выбрать ЭЦ и шарниры таким образом, чтобы расстояние между центрами вращения шарниров на торцах ЭЦ соответствовало размерам на кинематической схеме. Выбран ЭЦ с максимальной длиной 775 мм, и минимальной длиной 455 мм (рисунок 3.10, а). Для обеспечения необходимых углов поворота разработан карданный шарнир, расположенный на расстоянии 45 мм от оси вращения (рисунок 3.10, б).

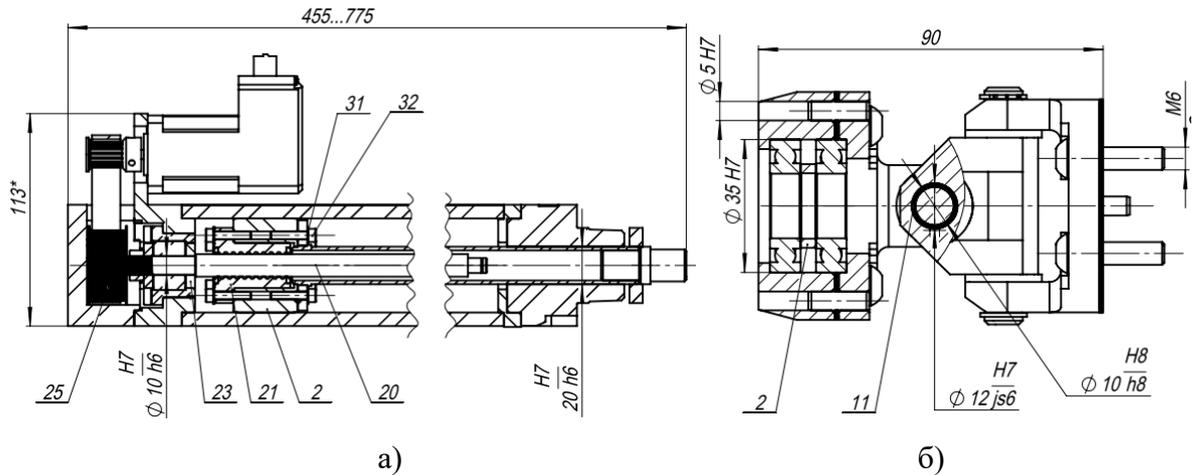


Рисунок 3.10 - Чертежи подобранных узлов: а) ЭЦ; б) карданный шарнир.

Общая минимальная длина телескопического звена с учётом выбранных элементов и фланцев крепления составила 545 мм, а максимальная длина составила 865 мм. Полученные геометрические размеры находятся в пределах допуска оптимального множества Парето. Основная сложность получения точного размера заключается в том, что существующие ЭЦ не обеспечивают необходимое удлинение при сохранении минимальной длины. При этом минимальная длина является важной, с точки зрения компактности и устойчивости всей системы. Верхняя и нижняя платформы робота-трипода должны обеспечивать полученные геометрические параметры и жёсткость механизма при минимальной металлоёмкости. Для этого проведём оптимизацию топологии распределения материала при помощи приложения САПР системы NX.

3.3 Топологическая оптимизация конструктивных элементов робота-трипода

Топологическая оптимизация представляет собой мощный инструмент современного проектирования, позволяющий эффективно распределять материал в конструкции для достижения наилучших характеристик при заданных ограничениях. Этот метод основан на математическом моделировании и численных методах, таких как метод конечных элементов (МКЭ), что позволяет минимизировать массу деталей, сохраняя их прочность и жёсткость. В условиях разработки сложных механических систем, таких как роботы-триподы,

топологическая оптимизация становится особенно важной, так как она помогает создавать легкие и одновременно надежные конструкции, способные выдерживать высокие нагрузки.

Топологическая оптимизация конструктивных элементов робота-трипода выполнена в CAD\CAE системе NX. Создана упрощённая электронно-цифровая модель нижнего основания с ключевыми элементами: фланцы ЭЦ, отверстия под крепления к подвижной платформе (рисунок 3.11).

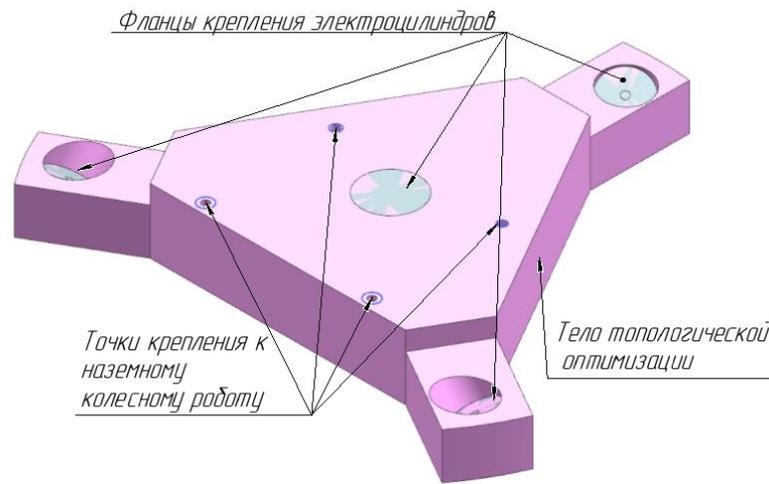


Рисунок 3.11 - Упрощённая ЭЦМ нижней платформы.

Упрощенная электронно-цифровая модель (ЭЦМ) нижней платформы загружается в приложение NX «Topological Optimization», где выполняется поэтапное назначение материала, приложение действующих сил к соответствующим поверхностям и настройка граничных условий.

Проведенный в предыдущем разделе динамический анализ движения по различным траекториям выявил, что максимальная нагрузка на нижний шарнирный узел может достигать 4500 Н. Учитывая, что рабочая область робота имеет приближенно сферическую форму, пиковое значение нагрузки распределялось поочередно на каждый фланец, к которому монтируются элементы конструкции ЭЦ. На основе предварительных расчетов определены векторы действия пиковых нагрузок, что позволило точно смоделировать условия их приложения. С использованием МКЭ выполнено детальное исследование распределения нагрузки в металлической конструкции. Это позволило выявить

ключевые области, которые принимают на себя основное воздействие прикладываемой силы, а также зоны, практически не воспринимающие нагрузку. Таким образом, происходит поэтапное удаление ненагруженного объема материала внутри тела топологической оптимизации (рисунок 3.12 а, б). Расчёт представленной конструкции требует значительной вычислительной мощности и для начала необходимо выполнить расчёт с конечными элементами ребра, которые равны 60 мм. С уменьшением размера конечных элементов происходит более детальное формирование геометрии и уменьшение массы конструкции. Тело топологической оптимизации, представленное на рисунке 3.11 массой 80 кг. В результате первой итерации масса снижена до 20 кг.

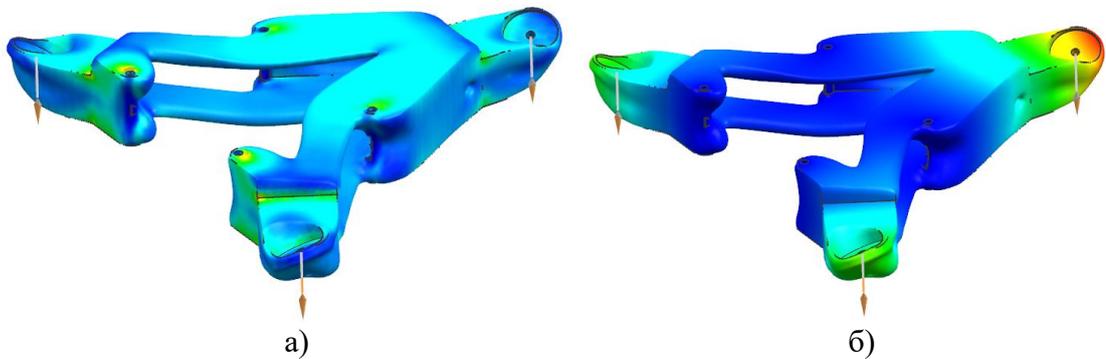


Рисунок 3.12. Результаты топологической оптимизации: а) напряжения по Мизесу; б) предельный изгиб платформы.

На начальной итерации не учтена центральная стойка, испытывающая предельную нагрузку в 3500 Н. Для уточнения результатов в модель добавлена дополнительная область приложения силы, после чего проведено несколько итераций с последующим уменьшением размеров конечных элементов с целью снижения массы нижнего основания (рисунок 3.13). Анализ уточненной расчетной модели (рисунок 3.13, б) демонстрирует практически полное отсутствие участков без нагрузки, что указывает на эффективное распределение по всей конструкции. Однако выявлен существенный недостаток используемого алгоритма: зона крепления к подвижной платформе интерпретируется как абсолютно жесткая область. Это приводит к тому, что алгоритмы топологической оптимизации не учитывают возможные деформации места закрепления (рисунок 3.13, г), что может

повлиять на точность полученных результатов и реальную работоспособность конструкции.

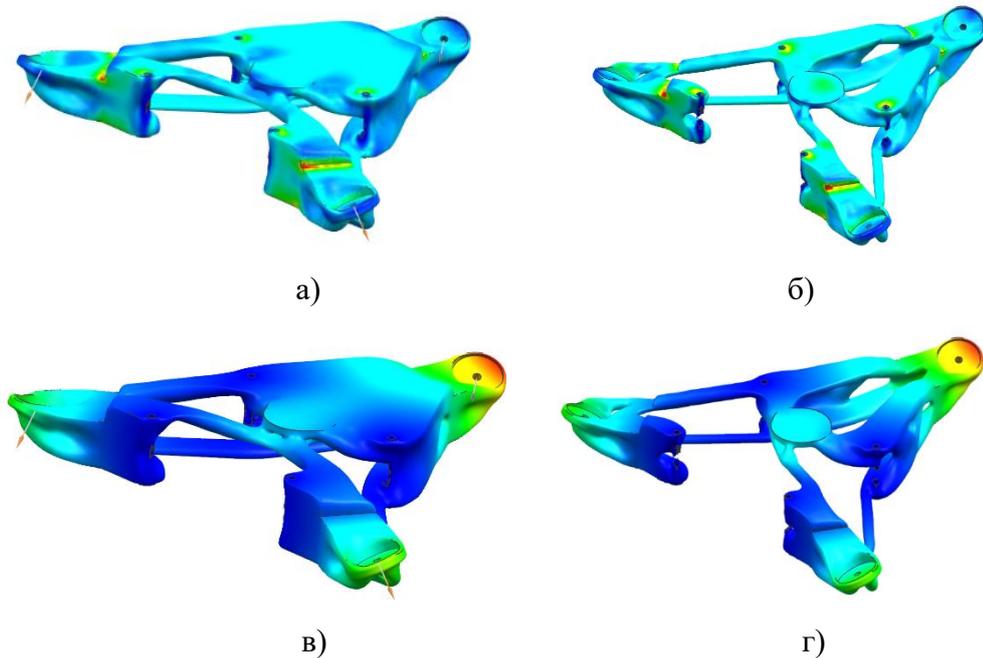


Рисунок 3.13 - Результаты двух итераций топологической оптимизации: а, б) напряжения по Мизесу; в, г) предельный изгиб платформы.

Так как окончания платформы идентичны, проведём оптимизацию топологии только одного отдельно взятого наиболее нагруженного окончания, что увеличит скорость расчёта и позволит снизить размер конечных элементов (рисунок 3.14, а, б). В результате четвертой итерации топологической оптимизации, после того как массивом построена неподвижная платформа при размере ребра конечных элементов 20 мм масса неподвижной платформы составила 6,7 кг. На рисунке 3.14, б видно, что участков, которые не воспринимают нагрузку, осталось минимальное количество.

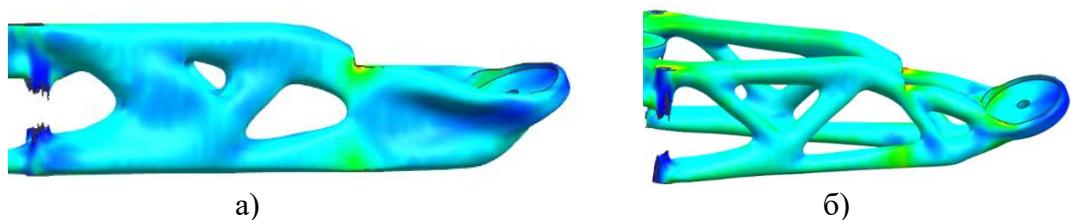


Рисунок 3.14 - Результаты оптимизации топологии четвертой итерации: а) исходное окончание платформы; б) окончание платформы после четвертой итерации оптимизации топологии.

Полученная конструкция нижней платформы имеет сложную поверхность, что затрудняет её изготовление. Для решения данной задачи применен метод реверс-инжиниринга, позволяющего создать конструкцию на основе исходного тела. На её основе выполнено проектирование рамы основания с использованием геометрически правильных форм, обеспечивающих технологичность и удобство изготовления. Процесс построения модели включал следующие этапы: сначала каждая сторона оптимизированной конструкции обводилась с помощью команды «Эскиз» (рисунок 3.15, а), после чего выполнялось «вытягивание» элементов для формирования трехмерной структуры. После завершения моделирования одной из секций нижней платформы применялась команда «массив», позволяющая создать симметричные окончания относительно центра платформы. Эти элементы затем объединялись общей нижней пластиной, завершая формирование конструкции (рисунок 3.15, б). Такой подход обеспечивает высокую степень унификации деталей и упрощает процесс сборки, что особенно важно для последующего масштабирования производства.

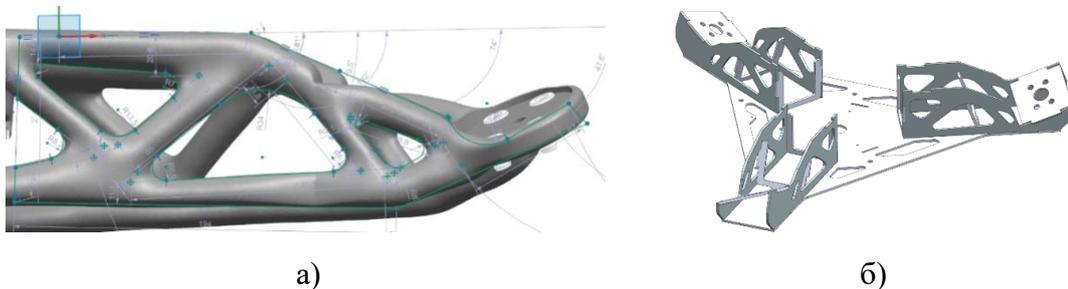


Рисунок 3.15 - Создание нижней платформы: а) эскиз профиля; б) сборка основания.

По аналогии с описанными действиями также выполнено построение верхней подвижной платформы. На основе чертежей шарнирных узлов и ЭЦ построены детализированные ЭЦМ при помощи САПР системы NX. Каждый из собранных конструктивных узлов робота-трипода тщательно проверен с использованием анализа зазоров. Этот этап является критически важным, так как он позволяет выявить потенциальные проблемы, связанные с неточностями сборки или недостаточной согласованностью между элементами. Особое внимание уделялось взаимодействию подвижных частей, чтобы исключить коллизии и

обеспечить плавную работу системы. После завершения разработки всех элементов робота-трипода выполнено их сопряжение, что позволило создать единую сборочную модель устройства (рисунок 3.16).

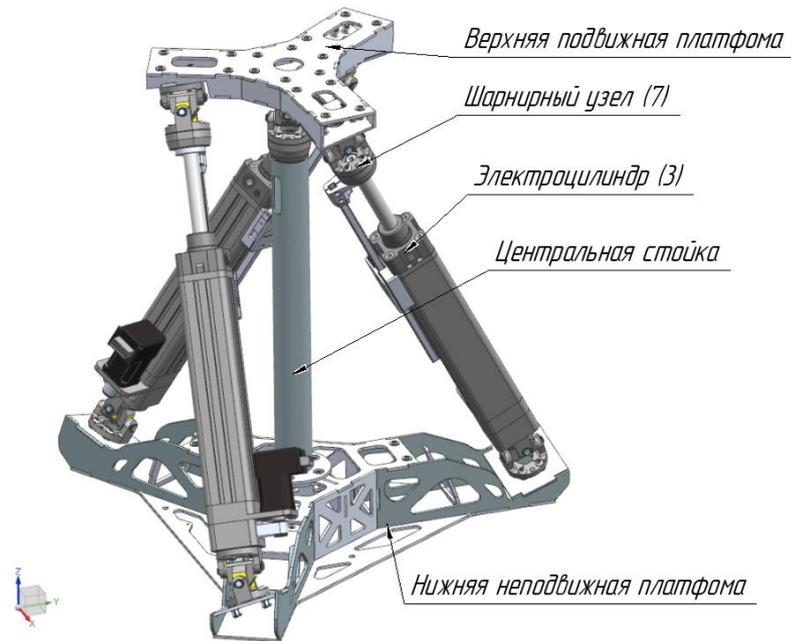


Рисунок 3.16 - ЭЦМ робота-трипода.

По аналогии с проектированием ЭЦ робота-трипода выполнена разработка телескопического звена, на которое установлено крепление для захватного устройства. В результате сборки, выявлено ограничение поворота платформы из-за ограниченного угла поворота шарниров. На рисунке 3.17 представлено положение робота-трипода, при котором происходит пересечение или заклинивание шарнира.

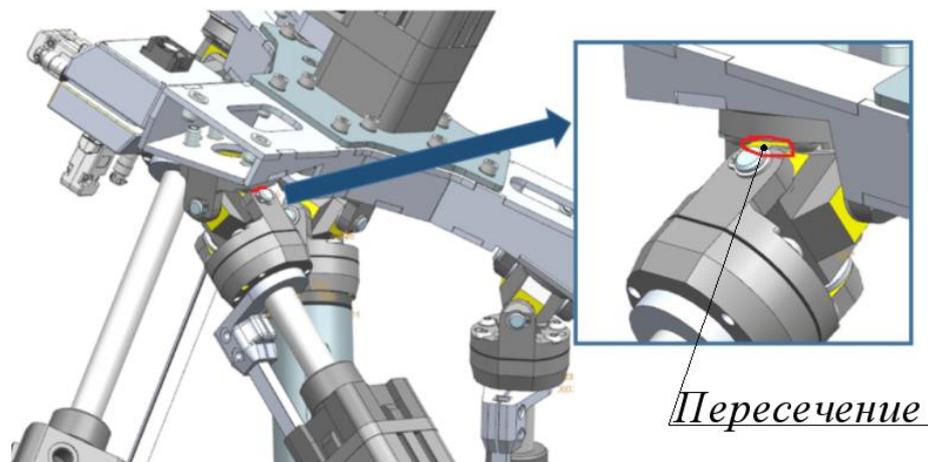


Рисунок 3.17 - ЭЦМ робота-трипода с телескопическим звеном в крайнем положении в момент возникновения пересечения.

При выбранной конфигурации робота-трипода шарнирные узлы функционируют лишь в части своего потенциального диапазона вращения. Это обусловлено тем, что ЭЦ изначально установлены под определённым наклоном, что влияет на кинематику системы. Для оптимизации работы шарниров фланцы на верхней и нижней платформах выполнены с наклоном в сторону соответствующих ЭЦ. Такая конструктивная особенность обеспечивает, чтобы в исходном положении верхней платформы карданные шарниры находились в своём центральном положении. (рисунок 3.18). Для реализации центрального положения шарниров нижний фланец повернут в сторону верхней платформы в двух взаимно перпендикулярных плоскостях на $13^{\circ}5'$ и $21^{\circ}23'$, что суммарно составило наклон нижних фланцев относительно горизонтальной плоскости $29^{\circ}2'$.

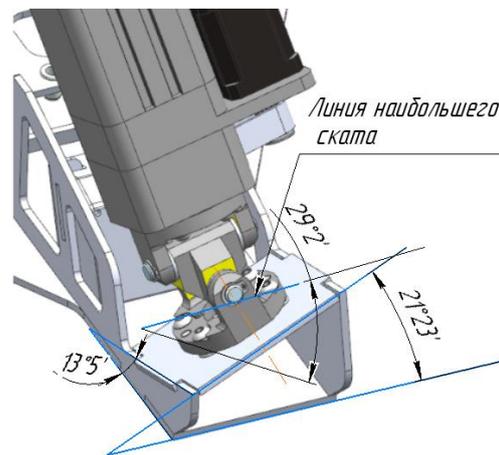


Рисунок 3.18 - Исполнение фланца нижней платформы с обозначением углов наклона.

Важным техническим решением стало использование карданного шарнира с поворотным основанием вместо традиционного сферического шарнира. Этот выбор обусловлен необходимостью увеличения предельного угла поворота и повышения общей маневренности системы. Карданный шарнир, благодаря своей конструкции, позволяет достичь более широкого диапазона угловых перемещений при сохранении высокой точности позиционирования. Кроме того, такая конструкция снижает механические нагрузки на элементы механизма за счет равномерного распределения усилий, что способствует увеличению срока службы устройства.

3.4 Имитационная модель устройства для складирования фруктов

При выдвинутом телескопическом звене гибкая труба образует множество изгибов, по которым фрукт плавно опускается без значительного ускорения. Однако при максимальном выдвижении телескопического звена гибкая трубка выпрямляется, и фрукт значительно ускоряется. Силу удара разогнанного фрукта можно определить по известной формуле $F_a = m * (V_2 - V_1)/\Delta t$, где V_1 - скорость в начале удара, V_2 - скорость после удара, Δt - время контакта в секундах. Учтём, что высота телескопического звена позволяет собирать фрукты на высоте двух метров, а контейнер для фруктов установлена на автономном шасси высотой около 0,5 метра. Тогда скорость свободно падающего фрукта определяет ускорение свободного падения и масса фрукта. Предполагается, что скорость падения фрукта после удара равна нулю. При скорости 15 м/с фрукт весом 240 граммов ударяется о другой фрукт, лежащий в корзине, с силой около 70 Н, что значительно превышает допустимое поверхностное давление на фрукт даже при значительной площади контакта. Это значение может варьироваться в разных пределах, поскольку условия самого удара могут меняться, однако в большинстве случаев фрукт будет поврежден. Силу удара можно значительно уменьшить, если снизить скорость или избежать прямого удара. Для этого на выходе из трубы построим поверхность сложной формы, которая будет демпфировать удар по касательной и не допустит полной остановки фрукта после контакта, поддерживая скорость V_2 . В том месте, где фрукт останавливается, выполним отверстие, через которое фрукт будет падать в корзину. Общий вид системы показан на (рисунок 3.19, а). Для проверки теории смоделированы две системы транспортировки яблок, в которых яблоко 3 падает с максимальной высоты без начального сопротивления в трубу 2, содержащую прямолинейный участок для ускорения. В первой системе фрукт непосредственно попадает из трубы в предполагаемый контейнер 5 для сбора урожая (рисунок 3.19, б). Во второй системе после трубы установлен инерциальный тормоз 4, который снижает скорость фрукта перед его попаданием в контейнер 5 (рисунок 3.19, б, в). При проведении моделирования не учитывалось

аэродинамическое сопротивление среды, однако учитывались кулоновское трение между контактирующими поверхностями и ускорение свободного падения. Эти факторы позволили получить более точные данные о динамике движения фрукта в обеих системах. Результаты расчетов представлены на рисунках 3.20, а и 3.20, б, где отображены графики изменения скорости падения яблока в зависимости от наличия или отсутствия инерциального тормоза.

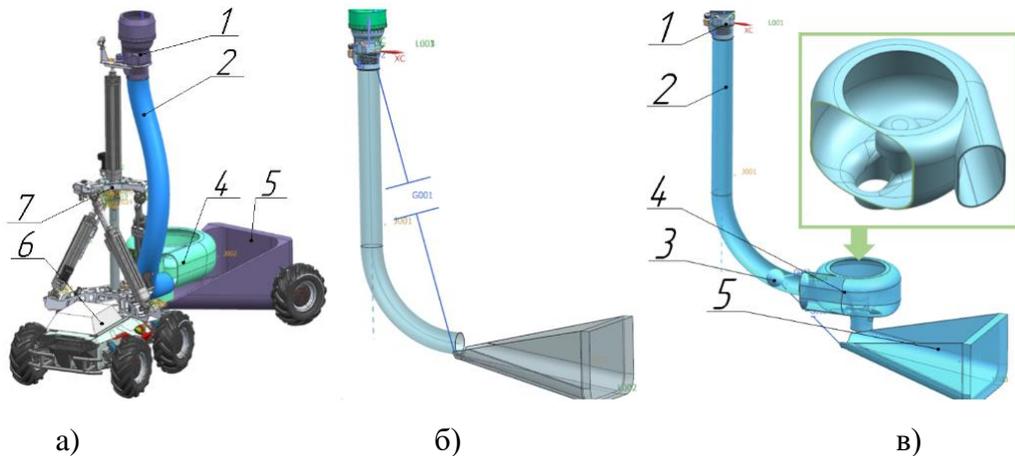


Рисунок 3.19 - Моделирование: а) общий вид системы; б) фрукт, падающий по трубе в контейнер для сбора фруктов; в) фрукт, падающий по трубе в контейнер для сбора фруктов через инерциальный тормоз (1 – захват, 2 – гибкая труба, 3 – фрукт, 4 – инерциальный тормоз, 5 – контейнер для фруктов, 6 – мобильная колёсная платформа, 7 – робот-трипод).

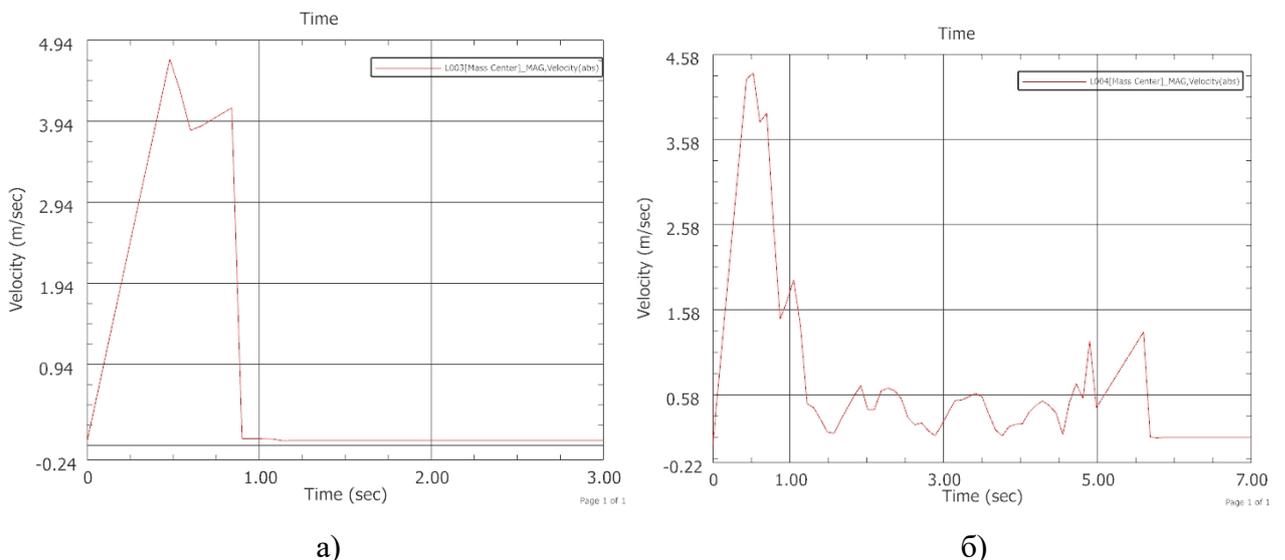


Рисунок 3.20 - Результаты моделирования: а) график скорости падения фрукта в трубу; б) график скорости падения фрукта в трубу и инерциальный тормоз.

Из графиков следует, что в первой системе фрукт попадал в контейнер со скоростью 3,94 м/с, что могло привести к значительным повреждениям из-за высокой силы удара. Во второй системе, оборудованной инерциальным тормозом, скорость фрукта существенно снижалась: при выходе из тормоза она составляла всего 0,2 м/с, а при попадании в контейнер — 1,01 м/с. Благодаря такому замедлению сила удара уменьшалась до диапазона от 0,45 Н до 4,5 Н, что значительно ниже допустимого поверхностного давления на кожуру фрукта. Таким образом, использование инерциального тормоза обеспечивает безопасное транспортирование фруктов, минимизируя риск их повреждения.

Для усиления эффекта трения предлагается использовать мягкий материал для изготовления инерциального тормоза. Это увеличит площадь контакта между фруктом и поверхностью тормоза, что, в свою очередь, значительно сократит время остановки и минимизирует ударные нагрузки. Однако для корректной работы системы необходимо правильно выполнить соединение между гибкой трубкой и инерциальным тормозом. Гибкая трубка должна быть установлена таким образом, чтобы фрукт не ударялся о стенки тормоза, а плавно перекачивался по его поверхности. Такой подход обеспечит равномерное распределение силы трения и исключит резкие воздействия на плод, сохраняя его целостность.

3.5 Имитационное моделирование робототехнической системы с захватным устройством

На основе построенной ЭЦМ выполнено имитационное моделирование движения робота-трипода с телескопическим звеном по наиболее неблагоприятной траектории, выявленной в предыдущем разделе. Траектория движения представлена на рисунке 3.2. Моделирование проведено в среде системы цифрового проектирования NX, которая обеспечивает высокую точность и детализацию анализа динамического поведения механизма. Это позволило оценить работоспособность конструкции в критических условиях и выявить потенциальные области для оптимизации. Для создания имитационной модели создаётся кинематическая структура, каждое звено которой представляет собой набор ЭЦМ

узлов исследуемого механизма. Первым кинематическим звеном задано захватное устройство, его крепление, камера технического зрения и штوك ЭЦ (рисунок 3.21, а). В качестве второго кинематического звена выбраны неподвижная часть ЭЦ, двигатель и верхняя платформа (рисунок 3.21, в). Такой подход позволяет точно определить массово-инерционные параметры отдельных звеньев и приблизить имитационную модель к реальному экспериментальному образцу.

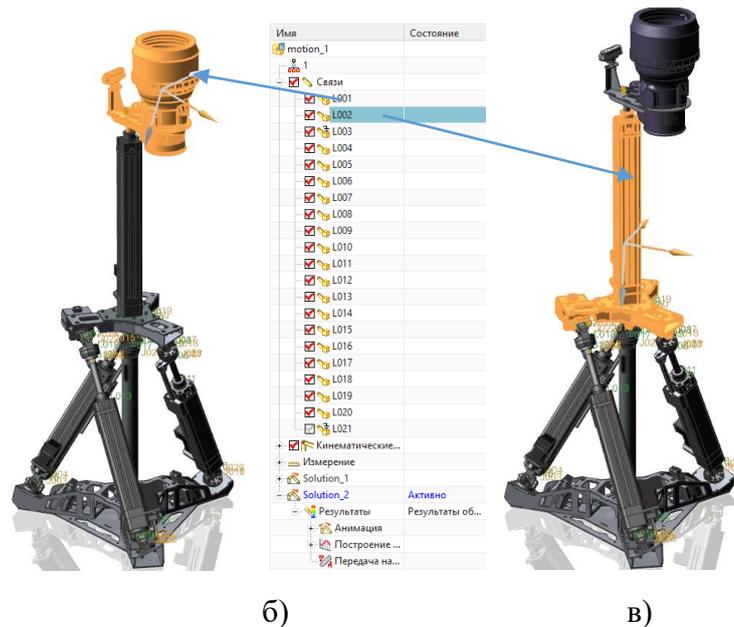


Рисунок 3.21 - Разработка имитационной модели: а) телескопическое звено; б) список всех звеньев модели; в) подвижная платформа.

После компоновки деталей цифрового двойника в кинематические звенья между ними создаются кинематические связи, которые определяют характер их взаимодействия. На данном этапе также задаются действующие силы, такие как внешние нагрузки, моменты и другие динамические воздействия, а также устанавливаются граничные условия для проведения расчетов. Это позволяет смоделировать реальное поведение механизма в различных рабочих сценариях и оценить его функциональность, прочность. Заданы следующие параметры: коэффициент трения в шарнирах 0.04; модуль упругости равен $2,1 \times 10^5$ Н/мм²; материал всех элементов назначена сталь; демпфирование и жёсткость поверхности контакта составляет 10 Н×с/мм и 10000 Н/мм соответственно.

В результате имитационного моделирования максимальный наклон телескопического звена составил 70° , при этом на 0.46 секунде, механизм принял положение, при котором происходит коллизия (рисунок 3.22, а). Таким образом, при наиболее неблагоприятной траектории предельный угол наклона робота-трипода составил $43,67$ градусов (рисунок 3.22б). На рисунках механизм представлен в крайнем положении.

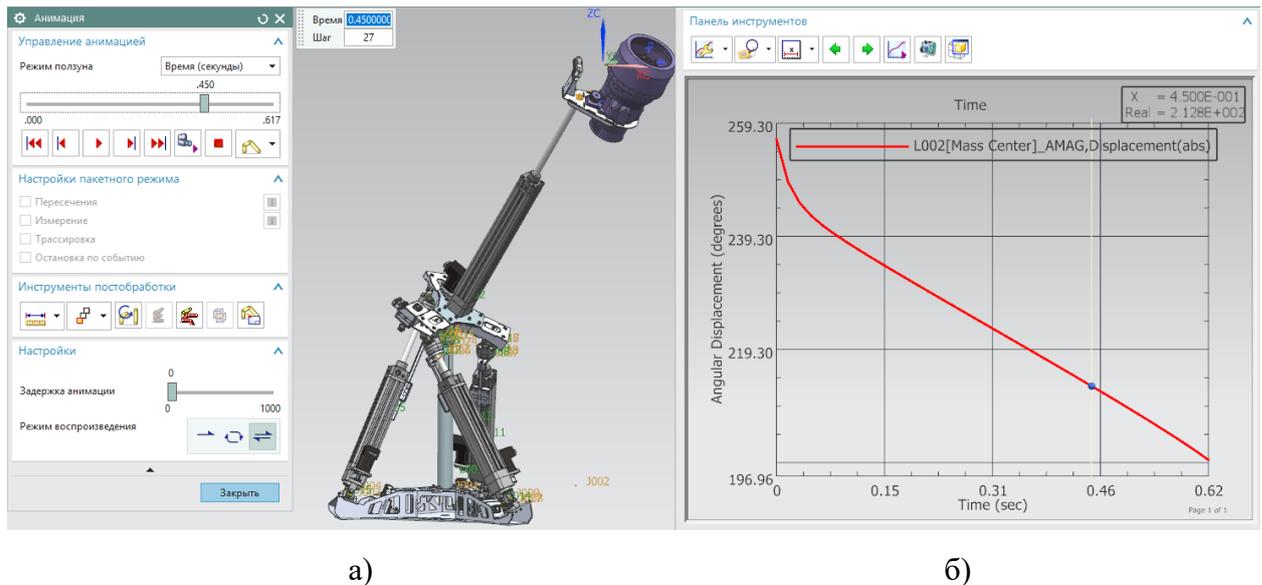


Рисунок 3.22 - Результаты имитационного моделирования: а) визуализация положения платформы в крайнем положении; б) график изменения угла ориентации платформы.

Из графика следует, что выбранная конфигурация робота-трипода способна обеспечить значительно больший угол наклона, однако происходит коллизия на 0.46 секунде. Для устранения этого необходимо модернизировать шарниры верхней платформы или изменить её конфигурацию.

На основе полученных параметров оптимизации захватного устройства с использованием САПР NX выполнено построение его ЭЦМ. Основным конструктивным элементом является внутренняя труба, объединяющая зубчатую и шарико-винтовую передачи. Процесс моделирования включает следующие этапы: создание цилиндра диаметром 110 мм, на поверхности которого формируется винтовая линия с заданным шагом P_B (рис. 3.23, а). Затем, в соответствии с углом наклона винтовой линии, на внутренней поверхности трубы строится зубчатый профиль и формируется внутренняя полость. Завершающим этапом является

создание сферической проточки вдоль траектории винтовой поверхности. На конце внутренней трубы выполняем резьбу для крепления корпуса пальцев. Далее в контексте сборки используя геометрию внутренней трубы, выполняется построение ведущей шестерни, которая расположена под тем же углом, что и зубчатый профиль трубы (рисунок 3.23, б). Строится внешняя труба, на внутренней стороне которой создается замкнутая винтовая поверхность с шагом P_B с помощью сплайна (рисунок 3.23, в). Замкнутая винтовая поверхность — это спираль, концы которой соединены между собой дугой, проходящей соосно с осью спирали. Для размещения тел качения внутри винтовой поверхности, часть желоба корпуса выполнена отдельно и расположена на крышке корпуса.

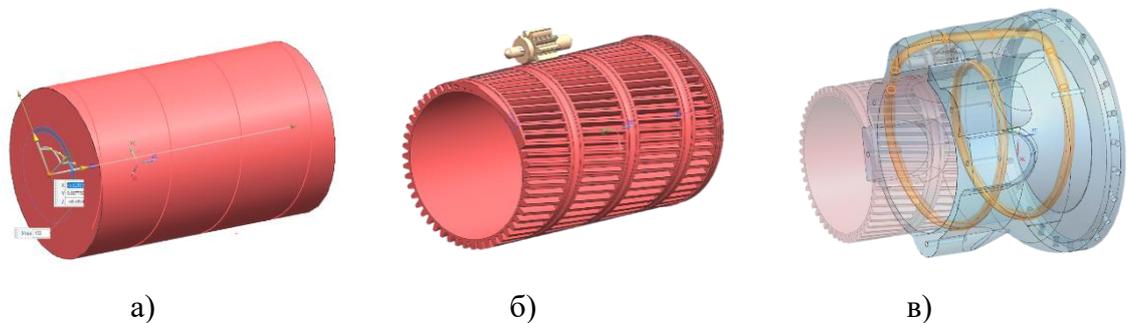


Рисунок 3.23 - Создание электронно-цифровой модели шарико-винтовой и зубчатой передач
ЗУ: а, б, в) этапы создания.

Чтобы увеличить площадь контакта с поверхностью фрукта и тем самым снизить вероятность повреждения, в ЗУ используется 10 пальцев. Каждый из пальцев должен иметь храповой механизм, чтобы блокировать разжатие при обратном закручивании трубы. При этом освобождение храпового колеса происходит в исходном положении трубы, когда фрукт находится внутри. Построение ЭЦМ механизма выполнено в двух положения, когда захват находится в исходном и конечном положениях. Собираемость обеспечена путём разделения механизма каждого пальца на сегменты, которые собираются в общий корпус (рисунок 3.24, а). Сегмент состоит из двух частей, каждая из которых соединена с предыдущей и последующей (рисунок 3.24, б). Когда все элементы собраны в общий корпус, они соединяются с внутренней трубой с помощью переходного кольца, которое обеспечивает надёжную фиксацию (рисунок 3.24, в).

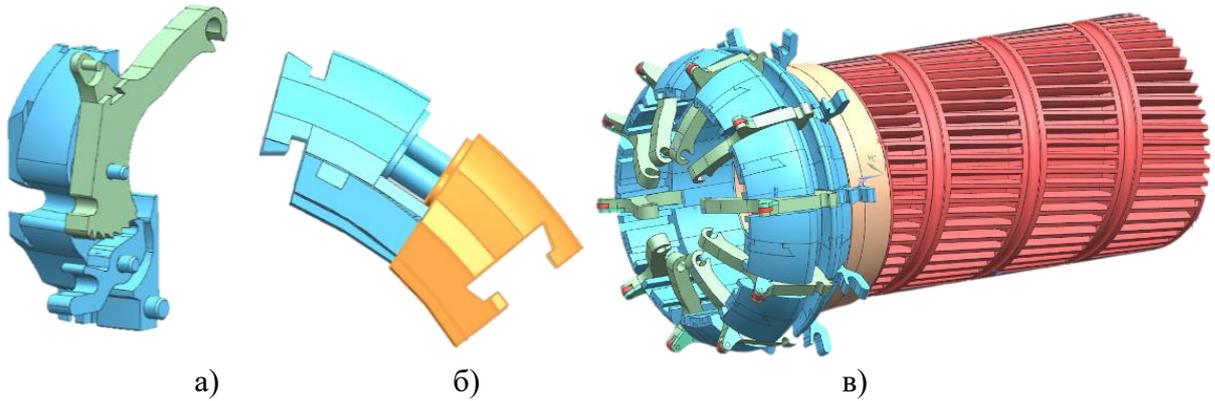


Рисунок 3.24 - Создание электронно-цифровой модели пальцев ЗУ: а) механизм одного пальца; б) две части сегмента; в) корпус в сборе.

На рисунке 3.25 показана окончательная ЭЦМ захватного устройства, которая построена в соответствии с геометрическими параметрами, полученными в ходе оптимизации.

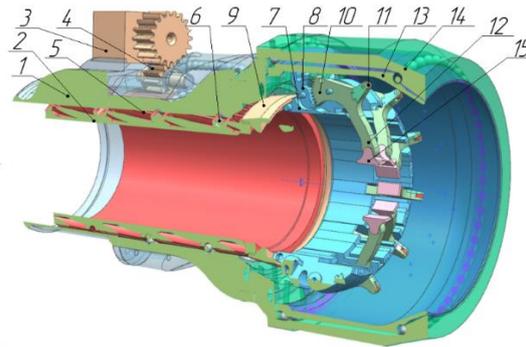


Рисунок 3.25 - Электронно-цифровая модель ЗУ (1 – внешняя труба; 2 – внутренняя труба; 3 – шаговый двигатель; 4 – зубчатая передача; 5 – зубчатый профиль; 6 – шарико-винтовая передача; 7 – корпус захвата; 8 – фиксатор; 9 – переходное кольцо; 10 – храповое колесо; 11 – ролик; 12 – палец; 13 – обойма; 14 – конус; 15 – мягкая губка).

Электронно-цифровая модель автономной РТС для сбора фруктов является головной сборкой и содержит все разработанные ЭЦМ. РТС построен на базе мобильной колесной платформы SCOUT 2.0, построение ЭЦМ, которого не потребовалось, так как производитель предоставляет модель в открытом доступе. На основе полученной модели робота SCOUT 2.0 выполнена сборка разработанных комплектующих (рисунок 3.26). ЭЦМ камеры технического зрения 5 также предоставляется производителем. Установка захватного устройства и камеры на телескопическое звено осуществляется с помощью крепления ЗУ 3.

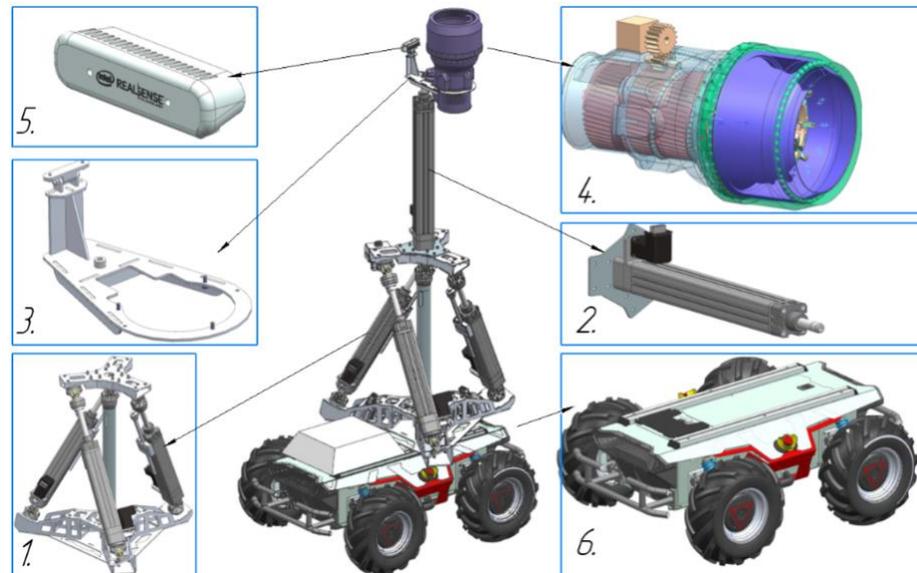


Рисунок 3.26 - Электронно-цифровая модель автономной РТС для сбора фруктов (1 – робот-трипод; 2 – телескопическое звено; 3 – крепление ЗУ; 4 – захватное устройство; 5 – камера технического зрения; 6 – наземный автономный колёсный робот).

Сбор фруктов осуществляется путём установки захватного устройства, которое отделяет фрукты и перемещает их в контейнер посредством гибкой трубы. Контейнер для фруктов является прицепным устройством и содержит инерциальный тормоз, которая снижает скорость фрукта после перемещения по гибкой трубе и сохраняет целостность кожуры. Электронно-цифровая модель РТС с корзиной для фруктов показана на (рисунок 3.27).

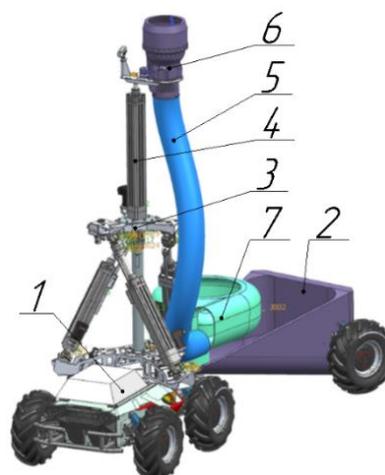


Рисунок 3.27 - Электронно-цифровая модель автономной РТС для сбора фруктов: 1 – мобильная колесная платформа; 2 – контейнер для сбора фруктов; 3 – робота-трипода; 4 – телескопическое звено; 5 – гибкая трубка; 6 – захватное устройство; 7 - инерциальный тормоз.

Выводы по главе

1. Реализован комбинированный метод проектирования робототехнической системы для сбора фруктов, включающий формализацию всех стадий проектирования: создание математических и электронно-цифровых моделей, цифровых двойников, топологическую оптимизацию распределения материала в конструктивных элементах, что позволило повысить точность и достоверность расчетов, сократить время проектирования.

2. С использованием разработанного метода проектирования на базе CAD/CAE - систем получена точная параметризованная электронно-цифровая модель автономной робототехнической системы для сбора фруктов, которая обладает высоким уровнем идентичности с реальным механизмом, что позволяет автоматизировать процесс проектирования и значительно ускорить его реализацию.

3. На основе разработанной ЭЦМ выполнено имитационное моделирование в динамической среде робота-трипода при заданных траекториях движения, что позволило получить полное представление о взаимосвязях кинематических и динамических параметров, а также учесть инерционные характеристики и силы, действующие в приводных парах при различных сценариях сбора фруктов.

ГЛАВА 4. ЭКСПЕРИМЕНТАЛЬНЫЕ ИССЛЕДОВАНИЯ

В главе рассмотрен выбор технических средств, разработка системы управления и полномасштабного экспериментального образца РТС. Выполнены экспериментальные исследования разработанного захватного устройства и экспериментального образца, сопоставление и анализ результатов теоретических и экспериментальных исследований.

4.1 Выбор технических средств и разработка механической части экспериментального образца робототехнической системы

4.1.1 Конструкция захватного устройства

Разработанное захватное устройство характеризуется сложной структурой и топологией, а также требованием к минимизации веса, в связи с чем наиболее предпочтительным методом изготовления прототипа является 3D-печать. Этот подход позволяет точно воспроизводить сложные геометрические формы, обеспечивая при этом оптимальное соотношение прочности и массы конструкции. Кроме того, использование аддитивных технологий предоставляет возможность быстро вносить изменения в конструкцию устройства на этапе прототипирования, что значительно сокращает время и затраты на разработку.

Методология проектирования и изготовления деталей с использованием 3D-печати включает несколько ключевых этапов, каждый из которых направлен на обеспечение высокого качества и функциональности конечного продукта. На первом этапе определяются цель создания детали и её технические требования, что позволяет выбрать наиболее подходящие параметры печати и материал, отвечающие условиям эксплуатации [124, 125]. Затем выполняется этап проектирования, где создаётся 3D-модель детали с учётом особенностей технологии 3D-печати, таких как геометрические ограничения, требования к поддержкам и ориентация модели при печати. После завершения проектирования файл преобразуется в формат STL и подготавливается для печати с помощью

специализированного программного обеспечения, которое разделяет модель на слои. Далее деталь изготавливается на 3D-принтере. Заключительным этапом является контроль качества, тестирование полученной детали и, при необходимости, её доработка для обеспечения соответствия заданным требованиям и надёжности. Такая методология позволяет эффективно использовать 3D-печать для создания сложных и высокоточных деталей в различных областях, таких как машиностроение, авиация, медицина и другие [126, 127].

Для подтверждения теоретических результатов разработан экспериментальный образец РТС для сбора фруктов. Конструкция захватного устройства спроектирована с учётом особенностей изготовления с использованием аддитивных технологий. Для каждой детали создана STL-модель, которая затем разделялась на слои с помощью программного обеспечения PolygonX (рисунок 4.1). Это позволило обеспечить точность и технологичность процесса печати, а также минимизировать временные и материальные затраты на производство.

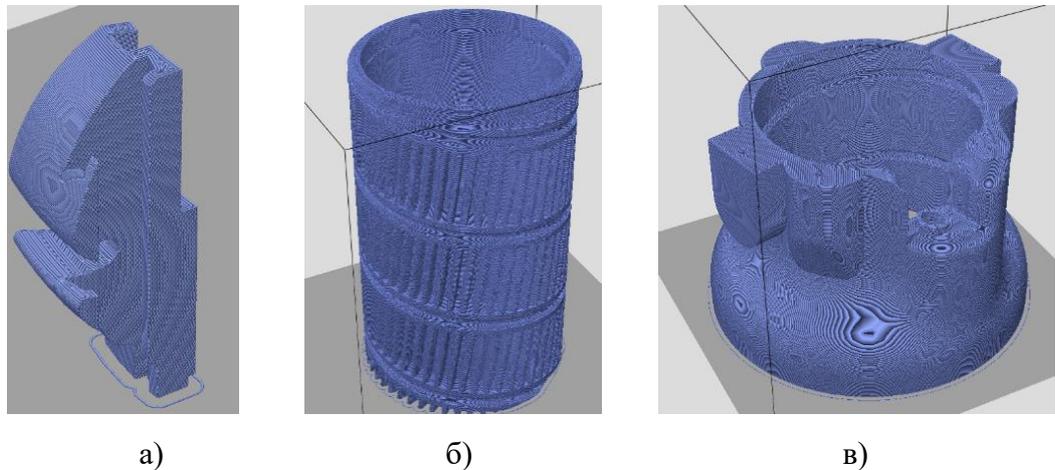


Рисунок 4.1 – Деление на слои ЭЦМ: а) корпус пальца; б) внутренняя труба с зубчатым и шарико-винтовым профилем; в) корпус.

При изготовлении макета выявлены следующие недостатки: для плавного перемещения внутренней трубы необходим кожух, поскольку в результате попадания мелких предметов между корпусом и внутренней трубой происходит заклинивание; конус на корпусе слишком узкий и не позволяет пальцам захватывать предметы, которые не попали внутрь (рисунок 4.2, а). С учётом выявленных недостатков изготовлен первый опытный образец, который на момент

тестирования не имел внутренней обоймы и обладал повышенным трением (рисунок 4.2, б). С учётом проведённых испытаний изготовлен третий образец с внутренней обоймой и роликами на пальцах (рисунок 4.2, в).

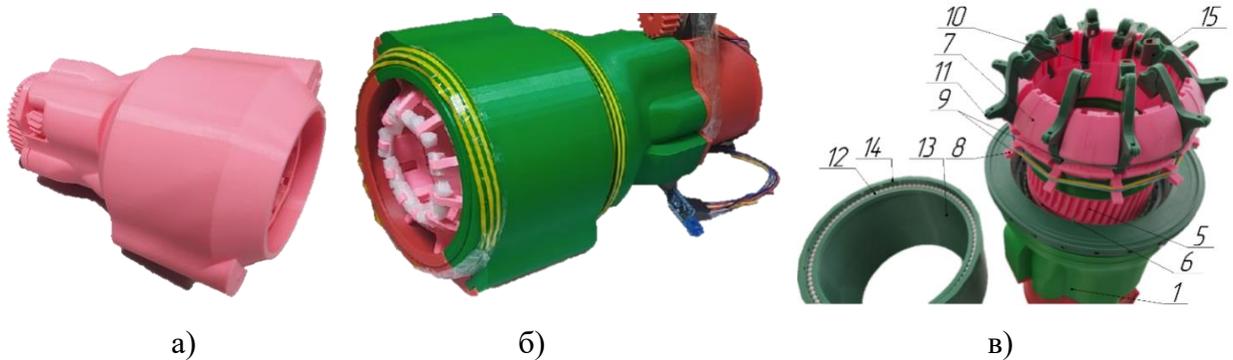


Рисунок 4.2. Экспериментальный образец захватного устройства: а) макет 1: 2; б) экспериментальный образец без обоймы; в) экспериментальный образец с обоймой (1 – наружная труба; 2 – внутренняя труба; 5 – зубчатый профиль; 6 – шариковая винтовая передача; 7 - корпус пальцев; 8 – фиксатор; 9 – пружины; 10 – палец с храповым колесом; 11 – ролик; 12 – подшипники; 13 – обойма; 14 – конус; 15 – мягкая губка.)

4.1.2 Экспериментальный образец робота-трипода с телескопическим звеном

На первом этапе изготовления экспериментального образца робота-трипода с телескопическим звеном необходимо разработать конструкторскую документацию для всех сборочных узлов и деталей РТС. Документация должна включать сборочные чертежи нижней и верхней платформ, ЭЦ, центральной стойки, шарниров, телескопического звена, крепления ЗУ, а также структурную схему системы управления. Изготовление верхней платформы, сборочный чертёж которой представлен на рисунке 4.3, выполнен из листов металла с использованием технологии шип-паз, что обеспечивает высокую точность сборки и надёжность конструкции. Для этого предварительно выполнены детальные построения контуров всех составляющих элементов в натуральном масштабе, адаптированные для обработки на станках с числовым программным управлением (рисунок 4.4). На этапе подготовки данных для ЧПУ-обработки учитывались не только геометрические параметры, допуски реза, но и физические свойства материала, такие как толщина листа и его прочностные характеристики.

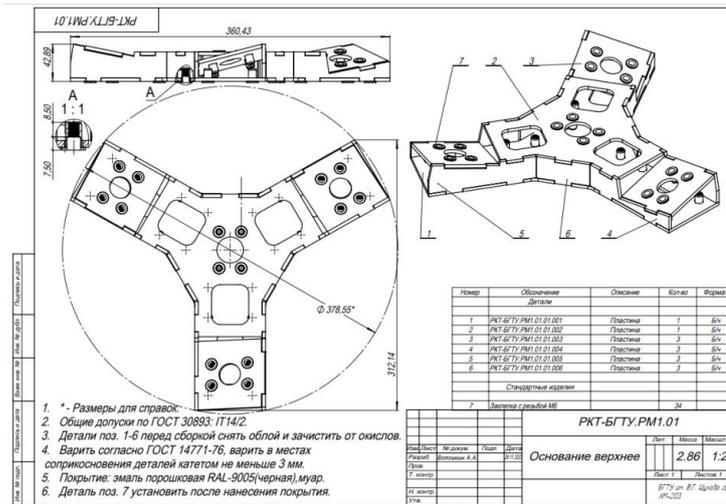


Рисунок 4.3 - Сборочный чертёж верхней платформы.

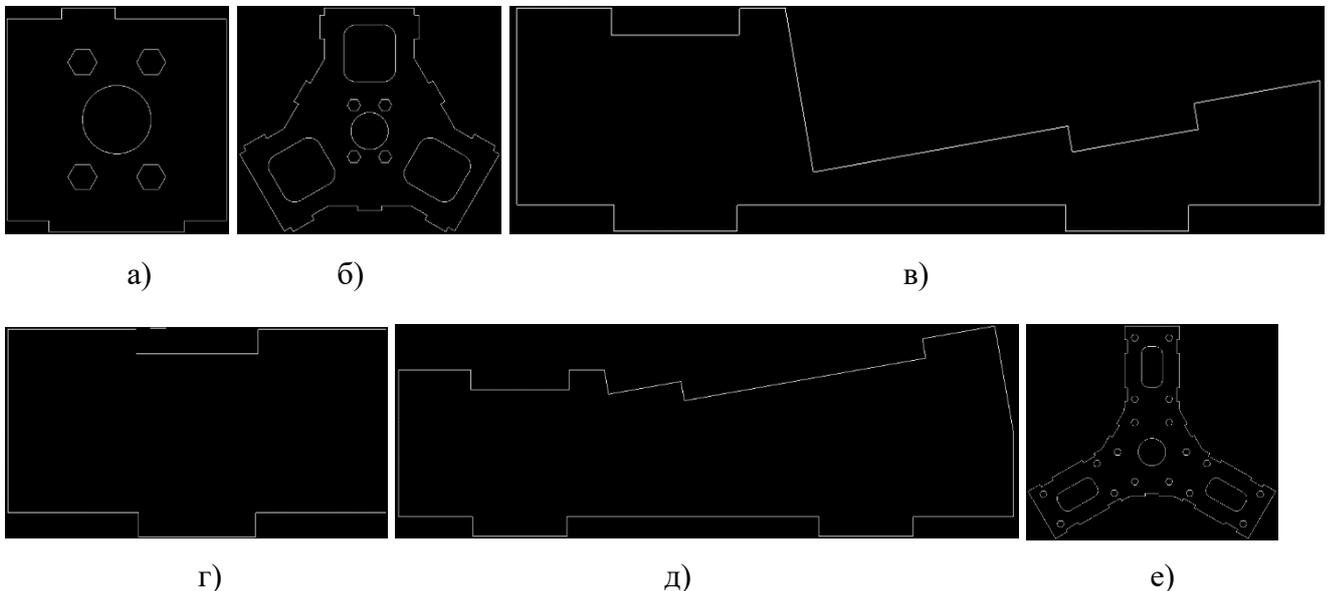


Рисунок 4.4 - Контуры свариваемых деталей верхней платформы: а) фланец; б) пластина верхняя; в) пластина правая; г) пластина переходная; д) левая пластина; е) пластина нижняя.

4.2 Система управления и программное обеспечение

4.2.1 Система управления мобильной колесной платформой

Стабильное и точное управление мобильной колесной платформой требует быстрой оценки положения и скорости движения. В средах с большим количеством окружающих объектов для планирования траектории движения необходимо знание трёхмерной среды, окружающей робот. Данные об окружающей среде можно получать с использованием различных датчиков: стереокамеры, камеры бинокулярного зрения, датчики Lidar. Камеры бинокулярного зрения (RGB-D)

захватывают цветные изображения, дополненные информацией о глубине каждого пикселя. Для получения оценки глубины применимы такие методы, как структурированное световое стерео, плотное пассивное стерео, лазерное сканирование диапазона. Совместное использование RGB-D камер и датчика LiDAR для картографирования и навигации местности позволяет получить более точную и полную информацию о окружающем пространстве. RGB-D камера: захватывает цветные (RGB) и глубинные данные, создавая трехмерные изображения сцены, что позволяет детально распознавать объекты и поверхности. LiDAR: дополняет данные глубины с высокой точностью, измеряя расстояния до объектов с использованием лазерного сигнала, независимо от освещенности. Комбинируя эти датчики, система эффективно создает точные 3D-карты и улучшает навигацию, особенно в условиях с переменной освещенностью и препятствиями. Для реализации совместного использования RGB-D камер и LiDAR для картографирования и навигации необходимо интегрировать данные с обоих датчиков в единую систему:

1. Синхронизация данных - камера RGB-D и LiDAR должны работать синхронно, чтобы получать одновременные данные о глубине и цвете (RGB), а также точные лазерные измерения расстояний.
2. Фьюжн данных (Data Fusion) - используется алгоритм слияния данных, который объединяет облака точек (Point Clouds) от LiDAR с глубинной информацией от камеры для создания единой карты местности.
3. Алгоритмы SLAM (Simultaneous Localization and Mapping) - алгоритмы SLAM обрабатывают совмещённые данные для одновременного построения карты и отслеживания положения робота в пространстве.
4. Калибровка - для точного совмещения данных необходимо откалибровать оба датчика, определив их относительное положение и ориентацию.
5. Обработка в реальном времени - используются мощные вычислительные ресурсы для обработки больших объёмов данных от обоих

сенсоров в режиме реального времени, что позволяет эффективно строить карты и ориентироваться в окружающей среде.

Для определения положения и ориентации мобильной колёсной платформы в пространстве на основе данных с RGB-D камеры использован визуальная одометрия. Визуальная одометрия позволяет оценивать трёхмерное движение, анализируя последовательность визуальных изображений.

Алгоритм локализации и картографирования реализован на мобильной колёсной платформе Scout 2.0, представленной на рисунке 4.5. Платформа оснащена RGB-D камерой Intel RealSense D415, подключённой к бортовому компьютеру, который обеспечивает выполнение алгоритмов оценки положения и управления в реальном времени. Камера установлена в передней части платформы на высоте 40 см, что позволяет эффективно анализировать окружающее пространство.



Рисунок 4.5 - Мобильная колесная платформа «Scout 2.0».

Перед выполнением локализации и картографирования выполнена калибровка RGB-D камеры (рисунок 4.6).

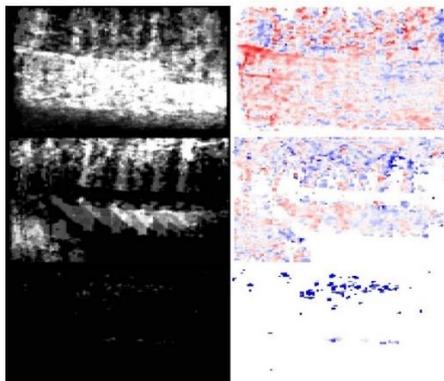
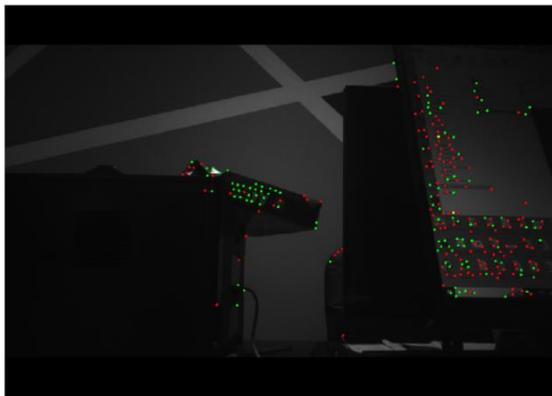
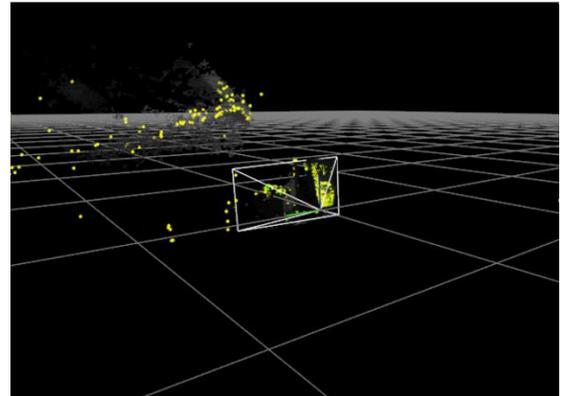


Рисунок 4.6 - Калибровка RGB-D камеры.

Визуальная одометрия состоит из шести основных этапов. На первом этапе выполняется предварительная обработка изображения: преобразование в оттенки серого и сглаживание компонента RGB, полученного с помощью камеры RGB-D. На втором этапе выполняется извлечение признаков на изображении с помощью детектора признаков FAST. Порог детектора выполняется с использованием простого пропорционального регулятора для обеспечения достаточного количества признаков в каждом кадре (рисунок 4.7, а). На третьем этапе выполняется оценка ориентации (рисунок 4.7, б). Для небольших движений большая часть видимого движения объекта в плоскости изображения вызвана трёхмерным вращением. Оценка этого поворота позволяет ограничить окно поиска при сопоставлении объектов между кадрами.



а)



б)

Рисунок 4.7 - Визуальная одометрия: а) извлечение признаков на изображении; б) оценка ориентации.

На четвёртом этапе выполняется сопоставление. Каждому объекту присваивается 80-байтовый дескриптор, состоящий из значений яркости участка 9×9 пикселей вокруг объекта, нормализованного до нулевого среднего значения и исключаяющего нижний правый пиксель. Отсутствие одного из пикселей делает длину дескриптора оптимальной для векторизованных инструкций. Затем функции сопоставляются между кадрами путём сравнения значений их дескрипторов функций с использованием проверки взаимной согласованности. Как только исходное совпадение найдено, местоположение объекта в новейшем кадре уточняется для получения субпиксельного совпадения. Уточнение вычисляется

путём минимизации суммы квадратических ошибок дескрипторов для решения итеративной нелинейной задачи наименьших квадратов. На пятом этапе выполняется обнаружение неверных совпадений признаков между кадрами. Хотя ограничения, налагаемые начальной оценкой ориентации, существенно снижают частоту неправильных совпадений признаков между кадрами, необходим дополнительный шаг для их дальнейшего исключения. Для этого применён подход Ховарда, заключающийся в вычислении графа непротиворечивых совпадений признаков. Таким образом, каждая пара совпадающих объектов в кадрах является вершиной графа, и между двумя такими парами совпадающих объектов формирует ребро, если трехмерное расстояние между объектами существенно не меняется от предыдущего кадра к последующему. На заключительном этапе выполняется оценка движения. Первоначальная оценка выполняется путём минимизации евклидовых расстояний между совпадениями внутренних объектов. Затем оценка движения уточняется путём минимизации ошибки перепроецирования объекта с использованием нелинейного решателя наименьших квадратов. Функции, превышающие фиксированный порог ошибки перепроецирования, отбрасываются и оценка движения уточняется.

Для обеспечения высокой точности планирования траектории использован алгоритм SLAM Gmapping. Этот алгоритм проецирует последовательные лазерные сканы на 2D-плоскость для построения карты, а угол поворота робота между сканированиями оценивается на основе данных гироскопа. Одометрия включает позу и вектор скорости, которые вычисляются с помощью данных датчиков Холла, которые фиксируют повороты ротора с шагом в 1 градус, сигнализируя о его положении. Использование датчика LiDAR для измерения расстояния до объектов значительно повышает эффективность алгоритма. Принцип работы заключается в том, что лазерный луч посылается на объект, отражается от него и возвращается к датчику, что позволяет точно измерять расстояние по следующей формуле

$$d = \frac{1}{2} \cdot C \cdot \Delta t, \quad (4.1)$$

где C – скорость света, Δt – время между испусканием и отражением лазерного луча. Так как угол излучения лазерного луча γ известен, то координата точки препятствия может быть записана в виде:

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} d \cdot \cos\gamma \\ d \cdot \sin\gamma \end{bmatrix}. \quad (4.2)$$

Картографирование проводилось при различных скоростях и траекториях движения, в результате чего выполнено 6 экспериментов. В первых трёх экспериментах движение осуществлялось в одном направлении на скоростях 0.5 м/с, 1 м/с и 1.5 м/с соответственно. В следующих трёх экспериментах использовались те же скорости, но движение происходило в двух направлениях. По результатам построено 6 карт, на основе которых оценено качество картографирования в условиях сада с деревьями. Лучшая карта получена при скорости 0.5 м/с, с поочередным движением в обе стороны выбрана эталоном (рисунок 4.8, а, б).



а)



б)

Рисунок 4.8 - Картографирование в условиях сада с деревьями: а) сада с деревьями; б) полученная в результате картографирования карта.

Для автоматического сравнения карт написан программный модуль на языке Python с использованием библиотеки OpenCV. Погрешность построения карты вычисляется как

$$\Delta = \frac{N_d}{N_m} \cdot 100\%, \quad (4.3)$$

где N_m - количество пикселей, не относящихся к фону хотя бы на одном из двух сравниваемых изображений, N_d - количество пикселей, имеющих на двух изображениях различный цвет по цветовой палитре RGB. Результаты сравнения различных экспериментов приведены в таблице 4.1.

Таблица 4.1 - Результаты сравнения экспериментов

Количество проходов	Погрешность картографирования при различной скорости		
	0,5 м/с	1 м/с	1,5 м/с
1 проход	8,98%	19,09%	32,85%
2 прохода	-	16,78%	18,78%

Проведена локализация и построение карты местности с использованием визуальной одометрии, полученной с RGB-D камеры с использованием датчиков LiDAR и Visual SLAM алгоритма. Предложенный метод позволяет безопасно обходить препятствия мобильной колесной платформой в сложных условиях неизвестной среды. Данные с RGB-D камеры обрабатывались для получения траектории движения и построения карты местности.

4.2.2 Система управления движением робота-трипода

Управление ЭЦ робота-трипода реализуется с использованием частотных преобразователей, которые интегрированы в систему управления низкого уровня. Данная система обеспечивает точное управление за счет обработки данных, поступающих от энкодеров и концевых датчиков, что позволяет контролировать положение и состояние каждого цилиндра. Разработанная архитектура системы управления подробно представлена на рисунке 4.9.

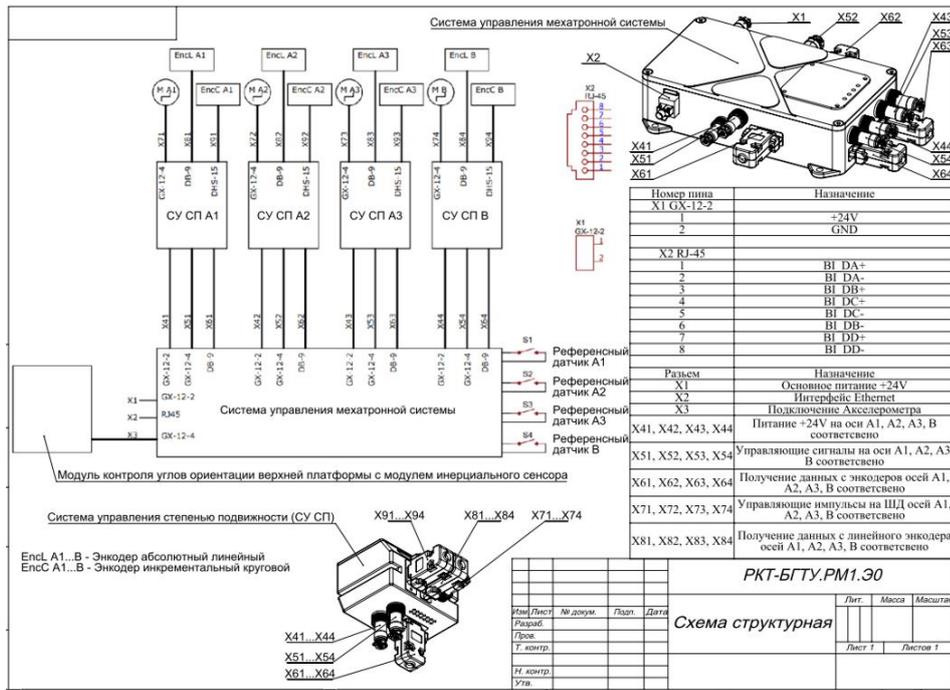


Рисунок 4.9 - Структурная схема управления.

Для движения экспериментального образца выбраны четыре шаговых двигателя бесщёточного типа с энкодером PL42H6403-D5-1K, каждый из энкодеров управляется контроллером TMC5160 со встроенным частотным преобразователем. Схема подключения представлена на рисунке 4.10. Все контроллеры объединены в общую схему управления. Коммуникация с устройствами управления верхнего уровня осуществляется по интерфейсу Ethernet по протоколу UDP с помощью клиент-серверной технологии.

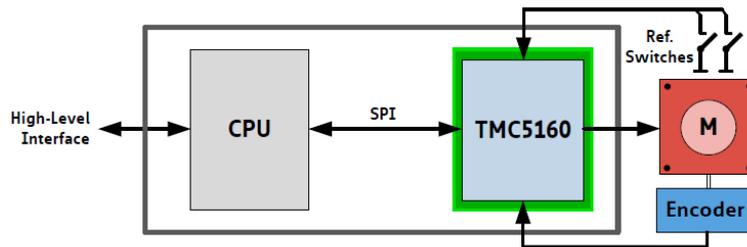


Рисунок 4.10 - Схема подключения шаговых двигателей PL42H6403-D5-1K.

Общая схема управления представлена на рисунке 4.11. Робот-трипод работает следующим образом: оператор задаёт положение захватного устройства в специальном интерфейсе. Информация обрабатывается в блоке управления, где происходит одновременно определение углов ориентации платформы qi_{DP} и

длины телескопического звена. Определение углов платформы происходит при помощи обратной задачи кинематики. Для отслеживания ошибки позиционирования робота-трипода, параллельно решается прямая задача кинематики по данным энкодеров, что позволяет получить фактический угол ориентации платформы q_{i_m} . На основе решения прямой и обратной задачи кинематики происходит коррекция ошибки позиционирования, при этом коррекция длины штанги происходит только по информации инкрементального энкодера. Каждый из ЭЦ имеет датчик крайнего положения. Для дополнительного контроля кинематической точности приводов робот-трипод содержит модуль с гироскопом и акселерометром или (inertial measurement unit) IMU сенсор. Измеренные значения обрабатываются блоком управления и преобразуются в корректирующие значения Δq_i , встроенным в контроллер алгоритмом.

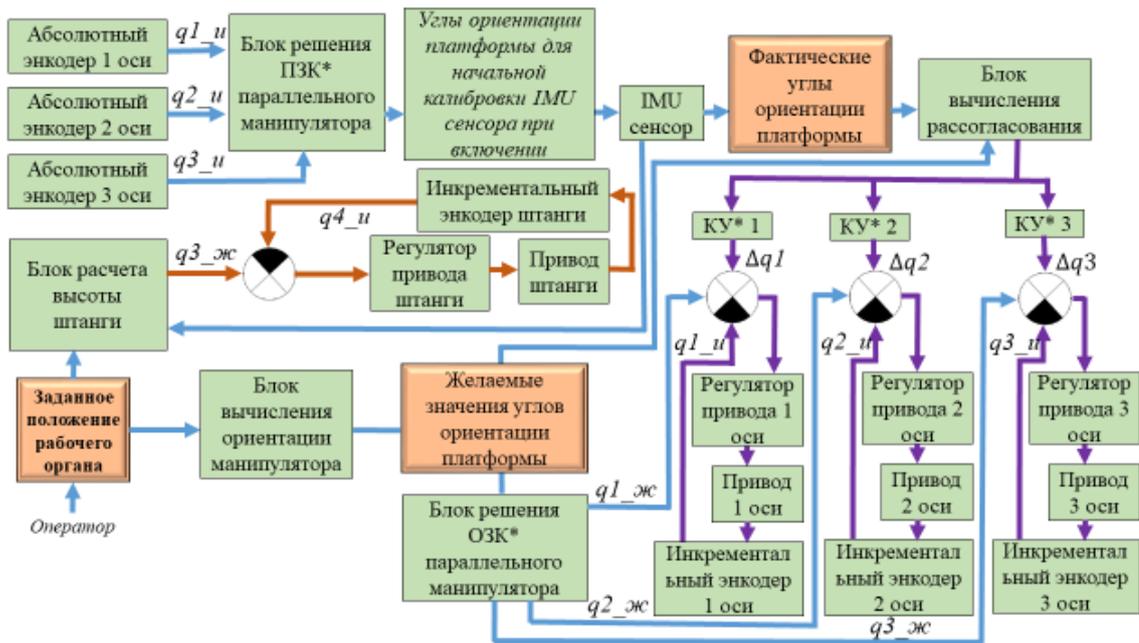


Рисунок 4.11 - Схема управления роботом-триподом. (q_{i_DP} – желаемое положение i оси, q_{i_m} – измеряемое положение i оси, Δq_i – корректирующее значение длины i оси).

Экспериментальный образец автономной РТС для сбора фруктов представлен на рисунке 4.12.



Рисунок 4.12 - Экспериментальный образец автономной РТС для сбора фруктов.

Экспериментальный образец состоит из мобильной колесной платформы «Scout 2.0» 1 и робота-трипода, на котором установлено телескопическое звено 2 выполненное в виде ЭЦ. На штоке ЭЦ 3 выполнено крепление для захватного устройства 4. Крепление имеет кронштейн 5 для установки камеры технического зрения. Захватное устройство 6 и его крепление 4 имеют соосные отверстия и соединены посредством винтов. Робот-трипод выполнен из верхней 7 и нижней 8 платформ, которые соединены тремя ЭЦ 9 и центральной стойкой 10. На каждом из ЭЦ установлен линейный датчик 11, определяющий длину выдвижения ЭЦ. Каждый из двигателей 12 ЭЦ имеет энкодер 13, в результате чего становится возможным выполнять сравнение реального удлинения ЭЦ при заданном количестве оборотов двигателя 12 и определять ошибку. Винт каждого ЭЦ приводится в движение посредством ремённой передачи 14. «SCOUT 2.0» 1 имеет съёмную батарею 15 на 24V и 30 ампер часов, которая имеет разъёмы 16 для подключения дополнительного оборудования. В передней части «Scout 2.0» 1 установлен металлический корпус 17, на котором выполнен монтаж лазерного датчика 18 LiDAR, камеры технического зрения 19 «Intel Realsense 3d»,

вычислительной системой 20 и монитора 21. Подключение камеры технического зрения 19 и периферийных устройств выполняется через USB-хаб 22.

Для корректного функционирования робота-трипода необходимо провести испытания и определить базовое расхождение движений экспериментального образца и имитационной модели, после чего внести соответствующие корректировки в регулятор. Роботу-триподу задана траектория движения телескопического звена в соответствии с координатами, используемыми в моделировании. Опытным путём достигнуто крайнее положение, при котором возникает заклинивание в шарнире (рисунок 4.13, а). Для фиксации результата наклона РТС использовалось приложение «Accelerometer Sensor», которое позволяет измерять крен, тангаж и угловую скорость. Определено вращение относительно двух осей и возникающие в результате этого ускорения (рисунок 4.13, б). Для калибровки измерений проведено несколько предварительных тестов. Показания приложения Accelerometer Sensor в исходном положении составляли $90^{\circ}1'$ градус крена и $180^{\circ}47'$ градусов тангажа. При максимальном наклоне по заданной траектории значение крена составляло $46^{\circ}02'$, а значение тангажа - $180^{\circ}26'$. При выполнении наклона платформы показания угла изменялись в пределах 1° . Погрешность измерения в один процент обусловлена колебаниями платформы во время движения.

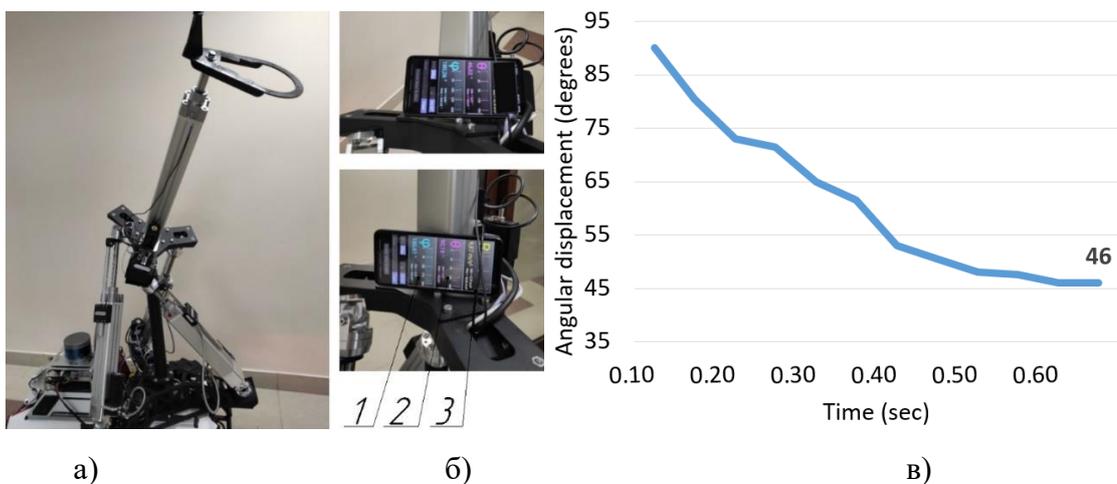


Рисунок 4.13 - Испытания экспериментального образца: а) положение экспериментального образца в крайнем положении; б) показания датчиков при наклоне платформы; (1 – крен; 2 – тангаж, 3 – угловая скорость); в) график измерения угла наклона платформы.

В результате эксперимента максимальный наклон верхней платформы при неблагоприятной траектории составил $44^{\circ}08'$ градусов (рисунок 4.13, в). Проведены измерения длины каждого ЭЦ, в результате которых не обнаружено расхождений со значениями низкоуровневого контроллера. Значение наклона для имитационной модели составило $43^{\circ}67'$, а для экспериментального образца - $44^{\circ}08'$, что является расхождением с имитационной моделью менее чем на 1%.

4.3 Проведение экспериментальных исследований

Испытания проводились в одном из промышленных садов компании «БелСад». Основной целью эксперимента является определение следующих параметров: достаточность силы захвата для отделения яблок от дерева, отсутствие повреждений яблок при захвате и отделении в зависимости от положения захватного устройства (под или над фруктом).

Для проведения эксперимента отобраны деревья разной высоты, включая низкие, высокие и деревья с необрезанной кроной. На рисунках 4.14, а и 4.14, б показан процесс сбора фруктов. Позиционирование захватного устройства выполнялось роботом-триподом на высотах от 1,4 до 2 метров.

Одним из ключевых аспектов стала защита яблока от повреждений. Пластиковые пальцы захватного устройства, обладающие острыми краями, могли повреждать поверхность плодов. Для предотвращения этого использованы мягкие губки, изготовленные из материала, соответствующего требованиям пищевой промышленности. Данный материал не оставляет следов на поверхности яблок и обеспечивает бережный захват.

Захват яблока осуществлялся в области его центра. Если во время позиционирования устройство не достигало середины яблока, захват становился невозможным, и фрукт выскальзывал. Это подчеркивает важность точности позиционирования при выполнении операций захвата и отделения. Полученные данные позволили выявить основные факторы, влияющие на успешность захвата,

и определить условия, в которых захватное устройство может быть менее эффективным.

Этот недостаток может быть связан с проблемой избыточной мягкости губок и неточности управления. В срывании фруктов методом одновременного отрывания и скручивания иногда происходит отделение плодоножки. Сила, при которой происходит срывание фрукта, оказалась значительно ниже рассчитанных 24 Н. В случае, когда пальцы захватного устройства смыкаются далеко за центром фрукта, отделение происходит и без полного цикла скручивания.



а)



б)

Рисунок 4.14 - Сбор фруктов: а) наибольший диаметр, б) минимальный диаметр.

Для проверки эффективности и безопасности захвата проведён сравнительный тест сбора фруктов. Аналогично эксперименту, описанному в научной работе [128], отобрано в общей сложности 25 образцов фруктов, собранных захватным устройством без видимых повреждений, а точки соприкосновения пальцев и фрукта отмечены маркерами. Собранные фрукты хранились в помещении при одинаковой температуре в течение 7 дней. Проанализировано, сколько фруктов осталось неповреждёнными, сколько из них выскальзывали при захвате и ли они повреждены при этом. Данные приведены в таблице 4.2.

Таблица 4.2 - Данные, полученные в результате сбора фруктов

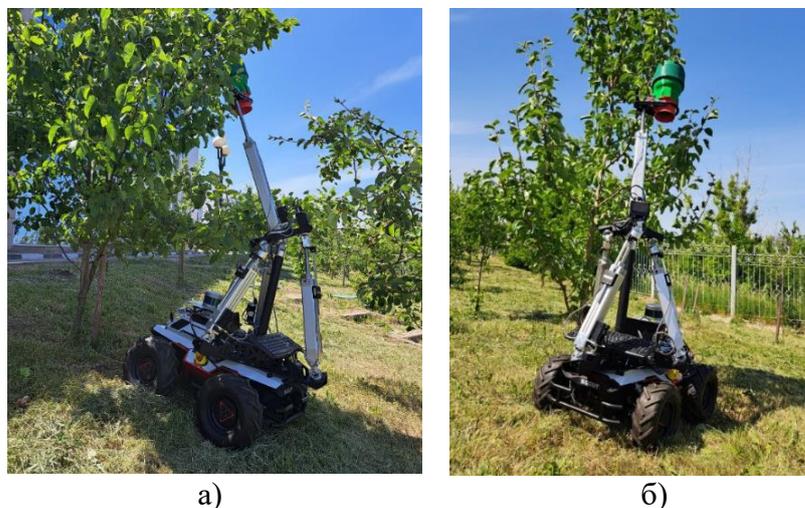
Средний диаметр (мм)	Средняя масса (г)	Кол-во соскальзывания пальцев	Кол-во собранных фруктов	Кол-во фруктов с видимыми повреждениями	Общее кол-во фруктов с повреждениями
81.26	234.98	4	24	1	5
Характеристики собранных фруктов					

Диаметр (мм)	масса (г)	Зафиксировано соскальзывание пальцев	Успешно собраны	Собраны с повреждением	Причины повреждения
83.26	241.6	нет	да	да	При сборе
83.1	241	да	да	нет	-
76.6	215.2	да	да	нет	-
91	290.4	да	нет	да	При захвате
73.5	201.5	да	да	нет	-
80.1	220.2	нет	да	нет	-

После семидневного ожидания выяснилось, что повреждено 5 фруктов. В то же время одно из них, диаметром 91 мм, имело явные повреждения. В захвате с 10 пальцами проскальзывание происходит только вдоль оси хватного устройства, так как между пальцами расстояние настолько мало, что не позволяет выскользнуть фрукту между ними. Проскальзывание внутрь хватного устройства считается положительным результатом. При осуществлении захвата наблюдалось положение мягкой губки, в котором мягкая часть не обращена к поверхности фрукта, в результате чего острая кромка пальца повредила фрукт. По сравнению с захватом с мягкими пальцами в работе [128], разработанный захват показывает на 12% большее количество повреждённых фруктов, но при этом и скорость сбора значительно выше. Мягкие губки имеют пластиковую основу с твердой кромкой, которая при соскальзывании повреждает плоды. Для предотвращения повреждений при соскальзывании, на окончание пальцев добавлен шарнир, который позволяет выполнять ориентацию губки по нормали к поверхности собираемого плода. Это позволяет уберечь фрукт от повреждений острой кромкой.

Для определения досягаемости плодов хватным устройством с учётом различных положений остановки мобильной колесной платформы, проведены испытания в промышленном саду (рисунок 4.15, а). На первом этапе проводились экспериментальные исследования робота-трипода с телескопическим звеном. Проведено несколько испытаний, в результате которых РТС осуществлялась остановка возле плодовых деревьев и выполнялось позиционирование хватного устройства в местах, где обнаружены плоды (рисунок 4.15, б). РТС демонстрировала способность корректно осуществлять навигацию, выполнять

локализацию деревьев, и остановку с учетом досягаемости плодов. Робот-трипод с захватным устройством успешно выполнял захват плодов, расположенных на различной высоте и под разными углами.



а)

б)

Рисунок 4.15 - Сбор фруктов на местности с уклоном: а) сбор при позиционировании против уклона; б) сбор при позиционировании вниз уклона.

Испытания РТС проводились в садовом хозяйстве, характеризующемся сложным рельефом и наличием множества неровностей в междурядьях плодовых деревьев. Указанные условия позволили оценить, как манёвренность мобильной платформы, так и её устойчивость при движении по пересечённой местности. Особое внимание уделялось способности системы сохранять стабильность при выполнении технологических операций в автономном режиме. Для проведения испытаний построена траектория движения мобильной платформы в режиме ручного управления. Для этого оператор задавал контрольные точки на карте местности, учитывая особенности рельефа и расположение деревьев. Траектория строилась таким образом, чтобы обеспечить минимальное отклонение платформы от оптимального маршрута и избежать столкновений с препятствиями. После завершения ручного программирования система переводилась в автономный режим, где выполнялось автоматическое следование по заданной траектории. В процессе движения платформы фиксировались ключевые параметры, такие как углы крена и тангажа, а также вертикальные и горизонтальные ускорения, возникающие при преодолении неровностей. Для анализа устойчивости

использовались данные с инерциальных измерительных модулей (IMU), установленных на платформе. Эти данные позволяли оценить степень отклонения системы от равновесного положения и выявить потенциально опасные участки пути, где риск потери устойчивости был наиболее высоким.

В результате на участках с уклоном или значительными неровностями наблюдались колебания корпуса робота-трипода, в пределах до 4° . Когда угол наклона поверхности, по которой выполнялось передвижение превышал 15° , требовалась корректировка положения платформы для предотвращения опрокидывания или смещения центра масс. В результате движения на максимальной скорости видеоизображение, передаваемое с камеры, являлось непригодным для дальнейшей обработки.

На основе полученных результатов сделан вывод, что в зависимости от кривизны участка требуется корректировка скорости движения для получения видеоизображения требуемого качества, а также предотвращения опрокидывания в результате раскачивания.

Результаты экспериментов подтвердили перспективность использования роботизированных систем для автоматизации сельскохозяйственных процессов. Применение РТС особенно актуально для крупных плодовых хозяйств, где требуется быстрый и качественный сбор урожая с возможностью прогнозирования и анализа собираемого урожая. Полученные данные легли в основу дальнейших исследований по совершенствованию адаптивной системы управления верхнего уровня и конструкции РТС.

Выводы по главе

1. Выбраны технические средства для оснащения экспериментального образца робототехнической системы, выполнена интеграция системных компонентов, датчиков, контроллера. Разработан и изготовлен полномасштабный экспериментальный образец робототехнической системы для сбора фруктов.

2. Разработана модель и конструкция захватного устройства, которое по сравнению с аналогом, обеспечивает большую производительность при сборе

фруктов. На основе экспериментальных исследований выполнена доработка экспериментального образца захватного устройства посредством установки мягких губок для исключения повреждений фруктов острой кромкой пальца.

3. Проведены экспериментальные исследования прототипа захватного устройства в садовом хозяйстве, в рамках которых выполнено сравнение с известным аналогом. Определено, что, предложенное захватное устройство обеспечивает большую производительность при сборе фруктов, но при этом использование жестких пальцев приводит к увеличению повреждаемых фруктов.

4. Проведены экспериментальные исследования образца робототехнической системы на основе технического зрения для отработки задач локализации и картографирования местности в автономном режиме с использованием Visual SLAM алгоритма и информации, полученной с датчика LiDAR. В результате исследований показано, что РТС выполняет безопасный обход препятствий в условиях неструктурированной среды.

5. Разработана и апробирована в реальных условиях конструкция полномасштабного экспериментального образца автономной робототехнической системы для сбора фруктов, позволяющая комбинировать методы отрывания и скручивания, а также складирование и транспортировку плодов.

ЗАКЛЮЧЕНИЕ

1. Анализ состояния исследований показал, что создание и применение новых типов робототехнических систем для сбора фруктов с высокими показателями структурной жёсткости, автономности, компактности и скорости является важным направлением сбора урожая. Эти системы не являются универсальными, и многое зависит от типа и геометрии собираемого фрукта.

2. Определен наиболее перспективный метод сбора: одновременное отрывание и скручивание фрукта, что учтено при проектировании захватного устройства.

3. Предложена модульная структура робототехнической системы для сбора фруктов, выполненная на базе мобильной колёсной платформы и робота-трипода, в котором для исключения избыточных степеней свободы и повышения управляемости использована пассивная центральная кинематическая цепь с выдвижным телескопическим звеном, а также захватное устройство с гибкой трубой для транспортировки и складирования фруктов.

4. Предложена структура и параметризованная модель специального захватного устройства, позволяющего комбинировать методы отрывания и скручивания фруктов за счёт наличия на корпусе винтовой поверхности, по которой перемещается внутренняя труба, имеющая на одном торце пальцы, обеспечивающие захват и отделение фруктов.

2. Разработана математическая модель кинематики робототехнической системы, описывающая зависимость положения телескопического звена с захватным устройством от углов ориентации подвижной платформы робота-трипода, учитывающая положения шарниров крепления кинематических цепей робота-трипода к платформам в виде систем нелинейных уравнений, определяющих ограничения на множество достижимых положений и ориентаций выходного звена и допустимых диапазонов движения линейных приводов.

3. Синтезирован эвристический алгоритм многокритериальной оптимизации параметров робота-трипода, на первом этапе которого выполняется

нормализация критериев, а на втором этапе - итеративная процедура оптимизации на основе однокритериальных эволюционных алгоритмов с использованием свёртки критериев с переменными коэффициентами важности. Использование в качестве критериев компактность конструкции и критерия, зависящего от количества недостижимых ориентаций платформы при заданных ограничениях на диапазоны приводных штанг, и отсутствие пересечений звеньев позволяет расширить функциональные и эксплуатационные характеристики робототехнической системы.

4. Синтезирован двухэтапный алгоритм технического зрения для локализации деревьев и распознавания фруктов на основе интеграции нейросетевых алгоритмов и преобразования Хафа; на первом этапе алгоритма выполняется формирование списка заданий на основе идентификации крон деревьев, а на втором - актуализация списка заданий на основе идентификации и сегментации плодов с учётом их спелости, что позволяет на основе визуальных данных комбинировать автономную навигацию и идентификацию координат объектов.

5. Разработан комбинированный метод проектирования робототехнической системы для сбора фруктов, включающий формализацию всех стадий проектирования: создание математических и электронно-цифровых моделей, цифровых двойников, топологическую оптимизацию распределения материала в конструктивных элементах, что позволило повысить точность и достоверность расчетов, сократить время проектирования.

6. С использованием разработанного метода проектирования на базе CAD/ CAE - систем получена точная параметризованная электронно-цифровая модель автономной робототехнической системы для сбора фруктов, которая обладает высоким уровнем идентичности с реальным механизмом, что позволяет автоматизировать процесс проектирования и значительно ускорить его реализацию.

7. На основе разработанной электронно-цифровой модели выполнено имитационное моделирование в динамической среде робота-трипода при заданных траекториях движения, что позволило получить полное представление о взаимосвязях кинематических и динамических параметров, а также учесть инерционные характеристики и силы, действующие в приводных парах при различных сценариях сбора фруктов.

8. Выбраны технические средства для оснащения экспериментального образца робототехнической системы, выполнена интеграция системных компонентов, датчиков, контроллера. Разработан и изготовлен полномасштабный экспериментальный образец робототехнической системы для сбора фруктов.

9. Проведены экспериментальные исследования прототипа захватного устройства в садовом хозяйстве, в рамках которых выполнено сравнение с известным аналогом. Определено что, предложенное захватное устройство обеспечивает большую производительность при сборе фруктов, но при этом использование жестких пальцев приводит к увеличению повреждаемых фруктов.

10. Проведены экспериментальные исследования образца робототехнической системы на основе технического зрения для отработки задач локализации и картографирования местности в автономном режиме с использованием Visual SLAM алгоритма и информации, полученной с датчика LiDAR. В результате исследований показано, что РТС выполняет безопасный обход препятствий в условиях неструктурированной среды.

11. Разработана и апробирована в реальных условиях конструкция полномасштабного экспериментального образца автономной робототехнической системы для сбора фруктов, позволяющая комбинировать методы отрывания и скручивания, а также складирование и транспортировку плодов.

СПИСОК ЛИТЕРАТУРЫ

1. Eigenbrod C., Gruda N. Urban vegetable for food security in cities. A review // *Agron. Sustain. Dev.* – 2015. – Т. 35. – P. 483–498.
2. Horrigan L., Lawrence R. S., Walker P. How sustainable agriculture can address the environmental and human health harms of industrial agriculture // *Environ. Health Perspect.* – 2002. – V. 110. – No. 5. – P. 445–456.
3. Gongal A., Amatya S., Karkee M., Zhang Q., Lewis K. Sensors and systems for fruit detection and localization: a review // *Comput. Electron. Agric.* – 2015. – Т. 116. – P. 8–19.
4. Jia W., Zhang Y., Lian J., Zheng Y., Zhao D., Li C. Apple harvesting robot under information technology: a review // *Int. J. Adv. Rob. Syst.* – 2020. – V. 17, – № 3. – P. 1–16.
5. Khort D.O., Kutyrev A.I., Smirnov I.G. Research into the Parameters of a Robotic Platform for Harvesting Apples. *Advances in Computer Science for Engineering and Manufacturing. ISEM 2021 // Lecture Notes in Networks and Systems.* Springer, Cham. – 2022. – V. 463.
6. Khort D., Kutyrev A., Filippov R., Kiktev N., Komarchuk D. Robotized platform for picking of strawberry berries // *2019 IEEE International Scientific-Practical Conference: Problems of Infocommunications Science and Technology, PIC S and T 2019 - Proceedings.* – 2019. – P. 869–872.
7. Khort D.O., Filippov R.A., Kutyrev A.I. Development of a universal robotic tool for agricultural purposes // *Journal of Advanced Research in Technical Science.* – 2019. – No. 13. – P. 69–73.
8. Wei J., Dean Z., Fengyi C., Bo X., Ying Z., Jinjing W. Automatic recognition vision system guided for apple harvesting robot // *Computers & Electrical Engineering.* – 2012. – V. 38. – № 5. – P. 1186–1195.
9. Zhao D., Lv J., Ji W., Zhang Y., Chen Y. Design and control of an apple harvesting robot // *Biosystems Engineering.* – 2011. – V. 110. – P. 112–122.

10. Zhang Z., Igathinathane C., Li J., Cen H., Lu Y., Flores P. Technology progress in mechanical harvest of fresh market apples // *Comput. Electron. Agric.* – 2020. – V. 175. – P105606.
11. Joseph R. D., Abhisesh S., Cameron H. J., Manoj K., Changki M., Qin Z. Proof-of-concept of a robotic apple harvester // *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2016.
12. Mehta S. S., Burks T. F. Vision-based control of robotic manipulator for citrus harvesting // *Computers and Electronics in Agriculture*. 2014. T. 102. C. 146–158.
13. Torregrosa A., Molina J. M., Pérez M., Ortí E., Xamani P., Ortiz C. Mechanical Harvesting of Ornamental Citrus Trees in Valencia, Spain // *Agronomy*. – 2019. – V. 9. – № 12. 827.
14. Wu D., Ding D., Cui B., Jiang S., Zhao E., Cao C. Design and experiment of vibration plate type camellia fruit picking machine // *International Journal of Agricultural and Biological Engineering*. 2022. V. 15. P. 130–138.
15. Jutras P. J., Coppock G. E., Patterson J. M. Harvesting citrus fruit with an oscillating air blast // *Trans. ASAE*. – 1963. – V. 6. – № 2. – P. 192–194.
16. Singh N., Sharma R., Dubey A., Awasthi O., Porat R., Supradip F. Rsc, Bharadwaj C., Sevanthi A. M., Kumar A., Sharma N., Carmi N. Harvesting maturity assessment of newly developed citrus hybrids for optimum juice quality // *Plants*. – 2023. – T. 12. – № 23. – C. 3978.
17. Karkee M., Zhang Q., Silwal A. Agricultural robots for precision agricultural tasks in tree fruit orchards // *Innovation in agricultural robotics for precision agriculture: a roadmap for integrating robots in precision agriculture*. – 2021. – P. 63–89.
18. Tillett N. D. Robotic manipulators in horticulture: a review // *J. Agric. Eng. Res.* – 1993. – V. 55, № 2. – P. 89–105.
19. Hohimer C. J., Wang H., Bhusal S., Miller J., Mo C., Karkee M. Design and field evaluation of a robotic apple harvesting system with a 3D-printed soft-robotic end-effector // *Trans. ASABE*. – 2019. – V. 62. – No. 2. – P. 405–414.

20. Zhang Z., Pothula A. K., Lu R. A review of bin filling technologies for apple harvest and postharvest handling // *Appl. Eng. Agric.* – 2018. – V. 34, – № 4. – P. 687–703.
21. Zhang Q., Karkee M., Tabb A. The use of agricultural robots in orchard management // *arXiv Preprint.* – 2019.
22. De-An Z., Jidong L., Wei J., Ying Z., Yu C. Design and control of an apple harvesting robot // *Biosyst. Eng.* – 2011. – Т. 110, – V 2. – P. 112–122.
23. Электронный ресурс, дата обращения 10.04.2024. Tevel: <https://www.tevel-tech.com>.
24. Bu L., Chen C., Hu G., Sugirbay A., Sun H., Chen J. Design and evaluation of a robotic apple harvester using optimized picking patterns // *Comput. Electron. Agric.* 2022. – V. 198. – No. 107092.
25. Zhang J., Kang N., Qu Q., Zhou L., Zhang H. Automatic fruit picking technology: a comprehensive review of research advances // *Artif. Intell. Rev.* – 2024. – V. 57.
26. Tong J., Zhang Q., Karkee M., Jiang H., Zhou J. Understanding the Dynamics of Hand Picking Patterns of Fresh Market Apples // *ASABE and CSBE/SCGAB Annual International Meeting, Montreal, Canada.* – 2014.
27. Li Z., Yuan X., Yang Z. Design, simulation, and experiment for the end effector of a spherical fruit picking robot. *International Journal of Advanced Robotic Systems.* – 2023. – V. 20.
28. Dimeas F., Sako D. V., Moulianitis V. C., Aspragathos N. A. Design and fuzzy control of a robotic gripper for efficient strawberry harvesting // *Robotica.* – 2015. – V. 33. – P. 1085–1098.
29. Rodríguez F., Moreno, J. C., Sánchez J. A., Berenguel M. Grasping in Agriculture: State-of-the-Art and Main Characteristics // *Mechanisms and Machine Science.* – 2012. – P. 385–409.

30. Pin skier J., Wang X., Liow L., Xie, Y., Kumar P., Langelaar M., Howard, G. Diversity-Based Topology Optimization of Soft Robotic Grippers // *Advanced Intelligent Systems*. – 2024.
31. Wang, X., Kang, H., Zhou, H., Au, W., Wang, M., Chen, C. Development and evaluation of a robust soft robotic gripper for apple harvesting // *Computers and Electronics in Agriculture*. – 2023. – V. 107552.
32. Davidson, J., He, L., Kutz, L. New Design of a Soft Gripper for Delicate Fruit Harvesting // *Journal of Agricultural Robotics*. 2023. – V. 15, – No. 2. – P. 125-136.
33. Yang J., Ren, C., Yang C. et al. Design of a Flexible Capture Mechanism Inspired by Sea Anemone for Non-cooperative Targets // *Chin. J. Mech. Eng.* – 2021. – V. 34. – P. 77.
34. Brown E., Rodenberg N., Amend J., Mozeika A., Steltz E., Zakin M. R., Lipson H., Jaeger H. M. Universal robotic gripper based on the jamming of granular material // *Proceedings of the National Academy of Sciences*. – 2010. – V. 107. – No. 44. – P. 18809–18814.
35. Li S., Yin X., Xia C., Ye L., Wang X., Liang B. TaTa: A Universal Jamming Gripper with High-Quality Tactile Perception and Its Application to Underwater Manipulation // *2022 IEEE International Conference on Robotics and Automation (ICRA)*, May 23–27, 2022. Philadelphia, PA, USA. – P. 6151–6157.
36. Beyaz A. Harvest glove and LabView based mechanical damage determination on apples // *Scientia Horticulturae*. – 2018. – V. 228. – P. 49–55.
37. Davidson J., Silwal A., Karkee M., Mo C., Zhang Q. Hand-picking dynamic analysis for undersensed robotic apple harvesting // *Trans. Asabe*. – 2016. – V. 59. – P. 745-758.
38. Li J., Karkee M., Zhang Q., Xiao K., Feng T. Characterizing apple picking patterns for robotic harvesting // *Computers and Electronics in Agriculture*. – 2016. – V. 127. – P. 633–640.

39. Bu L., Hu G., Chen C., Sugirbay A., Chen J. Experimental and simulation analysis of optimum picking patterns for robotic apple harvesting // *Scientia Horticulturae*. – 2019. – V. 108937.
40. Merlet J.-P. *Parallel Robots. Second Edition* / J.-P. Merlet. - Dordrecht: Springer, 2006. - 402 p.
41. Kong X., Gosselin C.M. *Type Synthesis of Parallel Mechanisms* / X. Kong, C.M. Gosselin. – Berlin: Springer, 2007. - 276 p.
42. Вирабян Л.Г., Халапян С.Ю., Кузьмина В.С. Оптимизация траектории позиционирования выходного звена планарного параллельного робота / Л.Г. Вирабян, С.Ю. Халапян, В.С. Кузьмина // *Вестник Белгородского государственного технологического университета им. В.Г. Шухова*. - 2018. - № 9. - С. 106–113.
43. Gaponenko E.V., Malyshev D.I., Kuzmina V.S., Rybak L.A. Geometric Parameters Optimization of Cable-Driven Parallel Robot with a Movable Gripper / E.V. Gaponenko, D.I. Malyshev, V.S. Kuzmina, L.A. Rybak // *Studies in Systems, Decision and Control*. – 2021. - V. 342. – P. 61–71
44. Pisarenko A., Malyshev D., Rybak L., Pervuznik V. Application of Recursive Algorithms for Optimization and Approximation of Workspace of Parallel Robots / A. Pisarenko, D. Malyshev, L. Rybak, V. Pervuznik // *Communications in Computer and Information Science*. - 2023. – V. 1913. – P. 256–267.
45. Rybak L.A., Khurtasenko A.V., Pervuznik V.S., Chuev K.V., Malyshev D.I. Optimization of the Design Parameters of a 6-DOF Mobility Platform / L.A. Rybak, A.V. Khurtasenko, V.S. Pervuznik, K.V. Chuev, D.I. Malyshev // *Mechanisms and Machine Science*. – 2023. – V. 134 MMS. –P. 115–124
46. Гладков Л.А. *Генетические алгоритмы. Учебное пособие* / Л.А. Гладков, В.В. Курейчик, В.М. Курейчик. - М.: Физматлит, - 2004. – 400 с.
47. Кочетов Ю.А., Плясунов А.В. Генетический локальный поиск для задачи о разбиении графа на доли ограниченной мощности / Ю.А. Кочетов, А.В.

Плясунов // Журнал вычислительной математики и математической физики. - 2012. - Т. 52. - № 1. - С. 164–176.

48. Дюбин Г.Н., Корбут А.А. Поведение в среднем жадных алгоритмов для минимизационной задачи о ранце - общие распределения коэффициентов / Г.Н. Дюбин, А.А. Корбут // Журнал вычислительной математики и математической физики. - 2008. - Т. 48. - № 9. - С. 1556–1570.

49. Neri F., Cotta C., Moscato P. Handbook of memetic algorithms / F. Neri, C. Cotta, P. Moscato. – Berlin: Springer, - 2011. - 370 p.

50. Лазарев А.А. Графический подход к решению задач комбинаторной оптимизации / А.А. Лазарев // Автоматика и телемеханика. - 2007. - № 4. - С. 13–23.

51. Меламед И.И., Сигал И.Х., Исследование параметров алгоритмов ветвей и границ решения симметричной задачи коммивояжера / И.И. Меламед, И.Х. Сигал // Автоматика и телемеханика. - 1997. - № 10. - С. 186–192.

52. Pisinger D., Ropke S. A general heuristic for vehicle routing problems / D. Pisinger, S. Ropke // Computers & operations research. - 2007. - V. 34. - No. 8. - P. 2403–2435.

53. Statnikov R., Matusov J., Statnikov A. Multicriteria Engineering Optimization Problems: Statement, Solution and Applications / R. Statnikov, J. Matusov, A. Statnikov // Journal of Optimization Theory and Applications. – 2012. – V. 155. – No. 2. – P. 355–375

54. Меламед И.И., Сигал И.Х. Вычислительное исследование линейной параметризации критериев в многокритериальном дискретном программировании / И.И. Меламед, И.Х. Сигал // Журнал вычислительной математики и математической физики. - 1996. - Т. 36. - № 10. - С. 23–25.

55. Ghosh B., Chakraborty D. A new Pareto set generating method for multi-criteria optimization problems / B. Ghosh, D. Chakraborty // Operations Research Letters. - 2014. - V. 42. - No. 8. - P. 514–521.

56. Ivlev V.I., Bozrov V.M., Misyurin S.Y. et al. Parameterization of an air motor based on multiobjective optimization and decision support / V.I. Ivlev, V.M. Bozrov, S.Y. Misyurin et al. // J. Mach. Manuf. Reliab. – 2013. – V. 42. – P. 353–358.

57. Misyurin S, Kreynin G, Nelyubin A, Nosova N. Multicriteria Optimization of a Dynamic System by Methods of the Theories of Similarity and Criteria Importance / S. Misyurin, G. Kreynin, A. Nelyubin, N. Nosova // Mathematics. 2021. – V. 9. – No. 22:2854

58. Misyurin S.Y., Nelyubin A.P., Kreinin G.V., Nosova N.Y. Multicriteria Optimization of a Hydraulic Lifting Manipulator by the Methods of Criteria Importance Theory / S.Y. Misyurin, A.P. Nelyubin, G.V. Kreinin, N.Y. Nosova // Biologically Inspired Cognitive Architectures 2021. – 2022. – V. 288–296

59. Misyurin S.Y., Nelyubin A.P., Potapov M.A. Multicriteria Approach to Control a Population of Robots to Find the Best Solutions / S.Y. Misyurin, A.P. Nelyubin, M.A. Potapov // Biologically Inspired Cognitive Architectures 2019. – 2019. – P. 358–363

60. Misyurin S.Y., Nelyubin A.P., Potapov M.A. Designing Robotic Groups Under Incomplete Information About the Dominance of Many Goals / S.Y. Misyurin, A.P. Nelyubin, M.A. Potapov // Advanced Technologies in Robotics and Intelligent Systems. – 2020. – P. 267–273.

61. Глазунов В.А., Дугин Е.Б., Кистанов В.А., Ву Нгок Бик Оптимизация параметров механизмов параллельной структуры на основе моделирования рабочего пространства / В.А. Глазунов, Е.Б. Дугин, В.А. Кистанов, Ву Нгок Бик // Проблемы машиностроения и надежности машин. Машиноведение. - 2005. - № 6. - С. 12–16.

62. Глазунов В.А., Новикова Н.Н., Рашоян Г.В., Нгуен Минь Тхань Оптимизация параметров механизма параллельной структуры для агрессивных сред при учете особых положений / В.А. Глазунов, Н.Н. Новикова, Г.В. Рашоян, Нгуен Минь Тхань // Проблемы машиностроения и надежности машин. - 2006. - № 2. - С.102-109.

63. Меламед И.И., Сигал И.Х. Вычислительное исследование трехкритериальных задач о деревьях и назначениях / И.И. Меламед, И.Х. Сигал // Журнал вычислительной математики и математической физики. - 1998. - Т. 38. - № 10. - С. 1780–1787.
64. Duyun T., Duyun I., Rybak L., Perevuznik V. Simulation of the structural and force parameters of a robotic platform using co-simulation // Procedia Computer Science. – 2022. – V. 213. – P. 720–727
65. Кочетов Ю.А., Пащенко М.Г., Плясунов А.В. О сложности локального поиска в задаче о p -медиане / Ю.А. Кочетов, М.Г. Пащенко, А.В. Плясунов // Дискретный анализ и исследование операций. - 2005. - Т. 12. - № 2. - С. 44–71.
66. Плясунов А.В. Параллельная сложность. Теория и приложения / А.В. Плясунов // Методы оптимизации и их приложения. - 2008. - С. 147–158.
67. Тихоплав О.А., Иванова В.В., Гурьянова Е.А., Иванов И.Н. Эффективность роботизированной механотерапии комплекса «LokomatPro» у пациентов, перенесших инсульт / О.А. Тихоплав, В.В. Иванова, Е.А. Гурьянова, И.Н. Иванов // Вестник восстановительной медицины. - 2019. - № 5. - С. 57–64.
68. Хачай М.Ю. Вопросы вычислительной сложности процедур обучения распознаванию в классе комитетных кусочно-линейных решающих правил / М.Ю. Хачай // Автоматика и телемеханика. - 2010. - № 3. - С. 178–189.
69. 67 Хачай М.Ю., Поберий М.И. Вычислительная сложность и аппроксимируемость серии геометрических задач о покрытии / М.Ю. Хачай, М.И. Поберий // Тр. ИММ УрО РАН. - 2012. - Т. 18. - № 3. - С. 247–260.
70. Кельманов А.В., Пяткин А.В. О сложности некоторых задач кластерного анализа векторных последовательностей / А.В. Кельманов, А.В. Пяткин // Дискретный анализ и исследование операций. - 2013. - Т. 20. - № 2. - С. 47–57.
71. Statnikov R., Bordetsky A., Matusov J., Sobol' I., Statnikov A. Definition of the feasible solution set in multicriteria optimization problems with continuous,

discrete, and mixed design variables / R. Statnikov, A. Bordetsky, J. Matusov, I. Sobol', A. Statnikov // *Nonlinear Analysis*. - 2009.- V. 71. – No. 12. - P. e109-e117.

72. Гончаров П. С., Ельцов М. Ю., Коршиков С. Б., Лаптев, И. В., Осиюк, В. А. NX для конструктора-машиностроителя: учебное пособие. — Москва: ДМК Пресс, 2010. — 503 с.

73. Ведмидь П. А., Сулинов А. В. Программирование обработки в NX CAM: учебное пособие. — Москва: ДМК Пресс, – 2014. — 305 с.

74. Ельцов М. Ю., Козлов А. А., Седойкин А. В., Широкова, Л. Ю. Проектирование в NX под управлением Teamcenter. — Москва: ДМК Пресс, 2013. — 752 с.

75. Конюхов А.В. Основы анализа конструкций в ANSYS. Казань: Казанский государственный университет, 2001.

76. Wang, S., Cao, Y. Analysis of Planetary Gear Transmission Characteristics Based on ANSYS // *Journal of Engineering Research and Reports*. – 2022.

77. Han E., Zhibin L., Jianyu Z., Zhiying L., Chen Y., Han W. ABAQUS and ANSYS implementations of peridynamics-based finite element method (PeriFEM) for brittle fracture. – 2022. – V. 1.

78. Tanjung I., Umurani K. Numerical Analysis of Tensile Strength on Welding Joints Using Solidworks Software // *Rekayasa Energi Manufaktur*. – 2022. – V. 7. – No.1.

79. Абдазимов А., Примкулов Б. Использование программы компас-3d при разработке усовершенствованной конструкции шпиндельного барабана с механизмом завода // XXII Международная научно-практическая конференция «Инновация-2017». – 2018. – P. 85–89.

80. Williamson M. P. Finite-element analysis // *Computer-Aided Engineering Journal*. – 1985. – V. 2. – No. 2. – P. 66–69.

81. Srinivasan S., Centea D. Problem Based Learning in Finite Element Analysis // *Visions and Concepts for Education 4.0*. – 2021. – P. 240–246.

82. Gupta S., Ehsani R., Kim N. Optimization of a Citrus Canopy Shaker Harvesting System: Mechanistic tree damage and Fruit // Transactions of the ASAE. – V. 59. – P. 761-776.
83. Shah F., Hussain F., Jamwal P., Hussain S. Design and Performance Analysis of a Parallel Mechanism Based Virtual Biomechanical Shoulder Robot Model // Conference: 2024 30th International Conference on Mechatronics and Machine Vision in Practice (M2VIP). — 2024. — V. 30.
84. Shen Z., Xie M., Song Z., Bao D. Design and Performance Analysis of a Parallel Pipeline Robot // Electronics. — 2024. — V. 13.
85. Khalil W., Dombre E. Modeling, Identification and Control of Robots. Butterworth-Heinemann, 2004.
86. Merlet J. P. Parallel Robots. Springer. — 2006.
87. Feng Q., Zou X. Design and Experiment of an Apple Picking Robot with Flexible Gripper // Biosystems Engineering. — 2018.
88. Hamner B., Narayanan S. Robotics in Precision Agriculture // IEEE Spectrum. — 2021.
89. Han J. et al. Minimizing Bruises in Robotic Apple Harvesting Using Controlled Grip Strength // Journal of Field Robotics. — 2017.
90. Nguyen V. H. et al. Energy-Efficient Path Planning for Autonomous Mobile Robots in Agriculture // Applied Sciences. — 2020.
91. Shridhar D. et al. Vision-Based Control Strategies for Autonomous Fruit Harvesting // Computers and Electronics in Agriculture. — 2020.
92. Gosselin C., Angeles J. Singularity analysis of closed-loop kinematic chains // IEEE Trans. Robotics & Autom. – 1990. – V. 6. – No. 3. – P. 281–290.
93. Behera L., Rybak L., Malyshev D., Gaponenko E. Determination of Workspaces and Intersections of Robot Links in a Multi-Robotic System for Trajectory Planning // Appl. Sci. – 2021. – V. 11. – P. 4961.

94. Evtushenko Y., Posypkin M., Rybak L., Turkin A. Approximating a solution set of nonlinear inequalities // *Journal of Global Optimization*. – 2018. – V. 71. – P. 129–145.
95. Rybak L., Malyshev D., Gaponenko E. Optimization Algorithm for Approximating the Solutions Set of Nonlinear Inequalities Systems in the Problem of Determining the Robot Workspace // *Communications in Computer and Information Science*. – 2020. – P. 27–37.
96. Дивеев А.И. Константинов С.В. М.: Эволюционные алгоритмы для решения задачи оптимального управления / *Вестник РУДН. Инженерные исследования*, – 2017– С. 254-265.
97. Электронный ресурс, дата обращения 01.06.2024. ine3d - 3D Bresenham's (a 3D line drawing algorithm): <https://ftp.isc.org/pub/usenet/comp.sources.unix/volume26/line3d>
98. Malyshev D., Cherkasov V., Rybak L., Diveev A. Synthesis of trajectory planning algorithms using evolutionary optimization algorithms // *Communications in Computer and Information Science*. – 2022. – V. 1739.
99. Maminov A. D., Posypkin M. A. Research and developing methods of solving engineering optimization problems for parallel structure robots // *International Journal of Open Information Technologies*. – 2019. – V. 7. – No. 11. – P. 1–7.
100. Rybak, L., Malyshev, D., Voloshkin, A. Algorithm for Multi-criteria optimization of robot parameters for fruit harvesting based on evolutionary methods // *Advances in Optimization and Applications. OPTIMA 2023. - 2024. - V. 1913*.
101. Qu Zhanghao Z. Peng Hu, Yaohua Y., Huanbo G., Taifeng Z., Kaili Z., Junchang. Optimal design of agricultural mobile robot suspension system based on NSGAIII and TOPSIS // *Agriculture*. – 2023. – V. 13. – No. 207.
102. Stocco L., Salcudean S., Sassani F. Fast constrained global minimax optimization of robot parameters // *Robotica*. 1998. – Vol. 16. – No. 6. – P. 595–605.
103. Holland J. H. *Adaptation in Natural and Artificial Systems*. Cambridge: MIT Press. – 1975.

104. Diveev A. I., Konstantinov S. V. Evolutionary algorithms for the problem of optimal control // RUDN Journal of Engineering Research. 2017. – V. 18. – No. 2. – P. 254–265.

105. Mirjalili S., Mirjalili S. M., Lewis A. Grey wolf optimizer // Advances in Engineering Software. – 2014. – V. 69. – P. 46–61.

106. Terven J., Córdova-Esparza D.-M., Romero-González J.-A. A Comprehensive Review of YOLO Architectures in Computer Vision: From YOLOv1 to YOLOv8 and YOLO-NAS / J. Terven, D.-M. Córdova-Esparza, J.-A. Romero-González // Machine Learning and Knowledge Extraction. – 2023. – V. 5. – No. 4. – P. 1680–1716. – DOI: 10.3390/make5040083.

107. Tropin D., Ilyuhin S., Nikolaev D., Arlazarov V. Approach for Document Detection by Contours and Contrasts / D. Tropin, S. Ilyuhin, D. Nikolaev, V. Arlazarov // 2020 25th International Conference on Pattern Recognition (ICPR). — 2021. — P. 9689—9695.

108. Method and means for recognizing complex patterns: US 3069654 A / P. V. C. Hough. — 1962.

109. Jacobs L., Weiss J., Dolan D. Object tracking in noisy radar data: Comparison of Hough transform and RANSAC / L. Jacobs, J. Weiss, D. Dolan // IEEE International Conference on Electro-Information Technology, EIT 2013. — 2013. — P. 1—6.

110. Voloshkin A., Rybak L., Malyshev D., Carbone G., Mohan, S. The design of a gripper device with screw and gear gears in a robotic fruit picking system // AIR '23: Proceedings of the 2023 6th International Conference on Advances in Robotics. – 2023. – V. 24. – P. 1–6.

111. Wei C.-C., Lai R.S. Kinematical analyses and transmission efficiency of a preloaded ball screw operating at high rotational speeds // Mechanism and Machine Theory. – 2011. – V. 46. – P. 880–898.

112. Steuer R. Multiple Criteria Optimization: Theory, Computation, and Application. Malabar: Robert E. Krieger Publishing, 1989.

113. Simon, D. Evolutionary Optimization Algorithms: Biologically-Inspired and Population-Based Approaches to Computer Intelligence. Canada, – 2013. – P. 776.
114. Jing L., XiaoBo S., Yu Z., Gang, H. Multi-objective shape optimization of developable Bézier-like surfaces using non-dominated sorting genetic algorithm // *Mechanics & Industry*. – 2023.
115. Duong V., Cuong, N. Expert opinion-based multi objective optimization: an application in plasma coating technology // *EUREKA: Physics and Engineering*. – 2022. – P. 175–184.
116. Lingmin X, Wei Y, Qinchuan L. Geometric Algebra-based Method for Inverse Dynamic Modeling of Parallel Robots // *Journal of Mechanical Engineering*. — 2022. — № 58. — P. 1–11.
117. Lu Hao, Yang Zhiqiang, Zhu Deliang, Deng Fei, Guo Shijie. Dynamics Modeling and Parameter Identification for a Coupled-Drive Dual-Arm Nursing Robot // *Chinese Journal of Mechanical Engineering*. — 2024.
118. Pham H., Do H., Quang M., Asadi F., Moore P. Proposed Multi-ST Model for Collaborating Multiple Robots in Dynamic Environments // *Machines*. — 2024. — Vol. 12. — No. 797.
119. Hu H., Shen Z., Chungang Z. A PINN-Based Friction-Inclusive Dynamics Modeling Method for Industrial Robots // *IEEE Transactions on Industrial Electronics*. — 2024. — PP. 1–9.
120. Hroncová D., Delyová I. COMPUTER SIMULATION USING MSC ADAMS // *Acta Mechatronica*. — 2020. — № 5. — P. 41–46.
121. Çakan A., Botsali F. Inverse kinematics analysis of a puma robot by using msc adams // *acta technica corviniensis – Bulletin of Engineering*. — 2017. — № 10. — P. 25–28.
122. Çakan A., Botsali F. Inverse kinematics analysis of a puma robot by using msc adams. — 2016.

123. Khurtasenko A. V., Chuev K. V., Rybak L. A. Dynamic model of a robotic platform with 6 degrees of freedom // International Conference on High-tech and Innovations in Research and Manufacturing (HIRM). – 2021. – V. 2176.
124. Budzik G., Woźniak J., Paszkiewicz A., Przesłowski Ł., Dziubek T., Dębski M. Methodology for the Quality Control Process of Additive Manufacturing Products Made of Polymer Materials // Materials. — 2021. — № 14. — P. 2202.
125. Huang J., Chen Q., Jiang H., Zou B., Li L., Liu J., Yu H. A survey of design methods for material extrusion polymer 3D printing // Virtual and Physical Prototyping. — 2020. — Vol. 15, № 2. — P. 148–162.
126. Wang X., Jiang M., Zhou Z., Gou J., Hui D. 3D printing of polymer matrix composites: A review and prospective // Composites Part B: Engineering. — 2017. — Vol. 110. — P. 442–458.
127. Kemavnar T., Shannon A., O'Sullivan L. W. The application of additive manufacturing / 3D printing in ergonomic aspects of product design: A systematic review // Applied Ergonomics. — 2021.
128. Chen K., Li T., Tongjie Y., Xie F., Qingchun F., Zhu, Q., Zhao C. A Soft Gripper Design for Apple Harvesting with Force Feedback and Fruit Slip Detection // Agriculture. – 2022. – V. 12. – No. 11. – P. 1802.

ПРИЛОЖЕНИЕ 1

Программа для определения рабочей и оптимизации работа-трипода с телескопическим звеном для сбора фруктов

```

#include "WS_processing.h"
#include "genetic_algorithm.h"
#include <math.h>
using namespace std;

int interfer_show_counter;
struct agroGeomParams{
    double Rmov;
    double Rfix;
    double lmin;
    double lmax;
    double central_chain_11;
    double central_chain_12;
    double lEndEffectorMin;
    double lEndEffectorMax;
    double dLinks;
    double angle_limit;
};
struct agroSimParams{
double s1,s2,accuracy_for_euler_platform_coords,accuracy_for_end_effector_coords,jacobSign;
double eulers_range,orientation_check,tubeOrientationToWhichJointB,max_angle_of_tube_error,singularity_check;
};

template<typename GeomParamsStruct, typename simParamsStruct>
class agroGeometryClass:public mechanismGeometryWithInitPointsAndParameters<GeomParamsStruct, simParamsStruct>{
public:
    agroGeometryClass(){
    };
    agroGeometryClass(simAndGeomParametersClass<GeomParamsStruct,simParamsStruct> &simAndGeomParametersInput){
        this->simAndGeomParameters = simAndGeomParametersInput;
        vector<double> linksR (11,simAndGeomParametersInput.geomParams.dLinks/2);
        this->radiiOfLinks = linksR;
        addPoints();
        addLinks();
        addPrisms();
    }
private:
    void addPoints(){
        this->vec_points_init.resize(2);
        this->vec_points_init[0].resize(6);
        this->vec_points_init[1].resize(6);

        for(int i=0;i<3;i++)
        {
            double angle=120.0*i;
            this->vec_points_init[0][i](0)=cos(angle*M_PI/180)*this->simAndGeomParameters.geomParams.Rfix;
            this->vec_points_init[0][i](1)=sin(angle*M_PI/180)*this->simAndGeomParameters.geomParams.Rfix;
            this->vec_points_init[0][i](2)=0;
            this->vec_points_init[1][i](0)=cos(angle*M_PI/180)*this->simAndGeomParameters.geomParams.Rmov;
            this->vec_points_init[1][i](1)=sin(angle*M_PI/180)*this->simAndGeomParameters.geomParams.Rmov;
            this->vec_points_init[1][i](2)=this->simAndGeomParameters.geomParams.central_chain_12;
        }
        this->vec_points_init[0][3](0)=0;
        this->vec_points_init[0][3](1)=0;
        this->vec_points_init[0][3](2)=0;

        this->vec_points_init[0][4](0)=0;
        this->vec_points_init[0][4](1)=0;
        this->vec_points_init[0][4](2)=this->simAndGeomParameters.geomParams.central_chain_11;
        this->vec_points_init[0][5](0)=0;
        this->vec_points_init[0][5](1)=0;
    }
};

```

```

    this->vec_points_init[0][5](2)=this->simAndGeomParameters.geomParams.central_chain_11-this-
>simAndGeomParameters.geomParams.dLinks*2;
    this->vec_points_init[1][3](0)=0;
    this->vec_points_init[1][3](1)=0;
    this->vec_points_init[1][3](2)=this->simAndGeomParameters.geomParams.central_chain_12;
    this->vec_points_init[1][4](0)=0;
    this->vec_points_init[1][4](1)=0;
    this->vec_points_init[1][4](2)=this->simAndGeomParameters.geomParams.central_chain_12+this-
>simAndGeomParameters.geomParams.lEndEffectorMin;
    this->vec_points_init[1][5](0)=0;
    this->vec_points_init[1][5](1)=0;
    this->vec_points_init[1][5](2)=this->simAndGeomParameters.geomParams.central_chain_12+this-
>simAndGeomParameters.geomParams.lEndEffectorMax;
}
void addLinks(){
    for (int i = 0; i<3; i++)
    {
        this->links.push_back(Eigen::Matrix2Xi {{0,i},{1,i}});
    }
    this->links.push_back(Eigen::Matrix2Xi {{0,3},{0,5}});
    this->links.push_back(Eigen::Matrix2Xi {{0,5},{0,4}});
    this->links.push_back(Eigen::Matrix2Xi {{0,4},{1,5}});
}
void addPrisms(){
    mechanismGeometry::trianglePrismInMechanism FixedPlatform;
    FixedPlatform.basePoints[0] = {0,0};
    FixedPlatform.basePoints[1] = {0,1};
    FixedPlatform.basePoints[2] = {0,2};
    FixedPlatform.otherPoints.push_back({0,3});
    FixedPlatform.otherPoints.push_back({0,5});
    FixedPlatform.height = this->simAndGeomParameters.geomParams.dLinks;
    FixedPlatform.pointsForAngleCheck.push_back({{0,3},{0,0},{1,0}});
    FixedPlatform.pointsForAngleCheck.push_back({{0,3},{0,1},{1,1}});
    FixedPlatform.pointsForAngleCheck.push_back({{0,3},{0,2},{1,2}});
    this->prisms.push_back(FixedPlatform);
    mechanismGeometry::trianglePrismInMechanism MovingPlatform;
    MovingPlatform.basePoints[0] = {1,0};
    MovingPlatform.basePoints[1] = {1,1};
    MovingPlatform.basePoints[2] = {1,2};
    MovingPlatform.pointsForAngleCheck.push_back({{1,3},{1,0},{0,0}});
    MovingPlatform.pointsForAngleCheck.push_back({{1,3},{1,1},{0,1}});
    MovingPlatform.pointsForAngleCheck.push_back({{1,3},{1,2},{0,2}});
    MovingPlatform.otherPoints.push_back({1,3});
    if (this->simAndGeomParameters.geomParams.central_chain_12<this->simAndGeomParameters.geomParams.dLinks*2){
        MovingPlatform.otherPoints.push_back({0,4});
    }
    MovingPlatform.height = this->simAndGeomParameters.geomParams.dLinks;
    this->prisms.push_back(MovingPlatform);
};
};
template<typename GeomParamsStruct, typename simParamsStruct>
class agroPosition:public mechanismPosition<agroGeometryClass<GeomParamsStruct,simParamsStruct>>{
public:
    Eigen::Vector3d EulerAnglesOfPlatform;
    agroPosition(agroGeometryClass<GeomParamsStruct,simParamsStruct> InputGeometry, Eigen::Vector3d
EulerAnglesOfPlatformInput){
        this->Geometry = InputGeometry;
        this->vec_points_current.resize(this->Geometry.vec_points_init.size());
        this->EulerAnglesOfPlatform = EulerAnglesOfPlatformInput;
    }
    bool isSingularityZone()
    {
        double alpha = EulerAnglesOfPlatform(0);
        double beta = EulerAnglesOfPlatform(1);
        double gamma = EulerAnglesOfPlatform(2);
        double Rmov = this->Geometry.simAndGeomParameters.geomParams.Rmov;
        double Rfix = this->Geometry.simAndGeomParameters.geomParams.Rfix;
        double central_chain_11=this->Geometry.simAndGeomParameters.geomParams.central_chain_11;

```

```

double central_chain_l2=this->Geometry.simAndGeomParameters.geomParams.central_chain_l2;
double sqrt3=sqrt(3);
double jacob_sign=this->Geometry.simAndGeomParameters.simParams.jacobSign;
double var_add_01 = (cos(gamma)*sin(alpha) + cos(alpha)*cos(beta)*sin(gamma));
double var_add_15 = central_chain_l2*cos(alpha)*sin(beta);
double var_add_02 = (Rmov*var_add_01 - var_add_15);
double var_add_33 = sin(alpha)*sin(gamma);
double var_add_03 = (cos(alpha)*cos(gamma) - cos(beta)*var_add_33);
double var_add_04 = (Rmov*var_add_03 - Rfix + central_chain_l2*sin(alpha)*sin(beta));
double var_add_05 = (central_chain_l1 + central_chain_l2*cos(beta) + Rmov*sin(beta)*sin(gamma));
double var_add_09 = Rmov*var_add_03 + central_chain_l2*sin(alpha)*sin(beta);
double var_add_06 = (- Rfix + var_add_09);
double var_add_07 = (var_add_05*var_add_05 + var_add_06*var_add_06 + var_add_02*var_add_02);
double var_add_08 = (2*sqrt(var_add_07));
double var_add_10 = (Rmov*var_add_03)/2;
double var_add_28 = (cos(alpha)*sin(gamma) + cos(beta)*cos(gamma)*sin(alpha));
double var_add_11 = (Rmov*sqrt3*var_add_28)/2;
double var_add_12 = var_add_10 - var_add_11 + central_chain_l2*sin(alpha)*sin(beta);
double var_add_31 = cos(alpha)*cos(beta)*cos(gamma);
double var_add_14 = (Rmov*sqrt3*(var_add_33 - var_add_31));
double var_add_17 = (Rmov*var_add_01)/2;
double var_add_13 = ((Rfix*sqrt3)/2 - var_add_17 + var_add_14/2 + var_add_15);
double var_add_16 = (Rfix/2 - var_add_10 + var_add_11 - central_chain_l2*sin(alpha)*sin(beta));
double var_add_32 = cos(gamma)*sin(beta);
double var_add_30 = Rmov*sqrt3*var_add_32;
double var_add_18 = (central_chain_l1 + central_chain_l2*cos(beta) + (Rmov*sin(beta)*sin(gamma))/2 + (var_add_30)/2);
double var_add_19 = (2*sqrt(var_add_16*var_add_16 + var_add_13*var_add_13 + var_add_18*var_add_18));
double var_add_20 = ((Rfix*sqrt3)/2 + var_add_17 - var_add_14/2 - var_add_15);
double var_add_21 = (Rfix/2 + var_add_12);
double var_add_22 = (2*sqrt(var_add_21*var_add_21 + var_add_20*var_add_20 + var_add_18*var_add_18));
double var_add_23 = central_chain_l2*cos(beta)*sin(alpha);
double var_add_24 = Rmov*cos(beta)*sin(gamma);
double var_add_25 = central_chain_l2*cos(alpha)*cos(beta);
double var_add_26 = Rmov*cos(alpha)*sin(beta)*sin(gamma);
double var_add_27 = Rmov*sin(alpha)*sin(beta)*sin(gamma);
double var_add_29 = Rmov*var_add_32;
double det1x = -(2*var_add_02*var_add_04 - 2*var_add_02*var_add_09)/var_add_08;
double det2x = -(2*var_add_12*var_add_13 + 2*(var_add_14/2 - var_add_17 + var_add_15)*var_add_16)/var_add_19;
double det3x = (2*var_add_12*var_add_20 + 2*(var_add_14/2 - var_add_17 + var_add_15)*var_add_21)/var_add_22;
double det1y = -(2*(central_chain_l2*sin(beta) - var_add_24)*var_add_05 - 2*(var_add_23 + var_add_27)*var_add_04 +
2*var_add_02*(var_add_25 + var_add_26))/var_add_08;
double det2y = (2*(var_add_25 + var_add_26/2 + (Rmov*sqrt3*cos(alpha)*var_add_32)/2)*var_add_13 - 2*(var_add_23 +
(var_add_27)/2 + (var_add_30*sin(alpha))/2)*var_add_16 + 2*(var_add_24/2 - central_chain_l2*sin(beta) +
(Rmov*sqrt3*cos(beta)*cos(gamma))/2)*var_add_18)/var_add_19;
double det3y = (2*(var_add_23 + (var_add_27)/2 + (var_add_30*sin(alpha))/2)*var_add_21 - 2*(var_add_25 + var_add_26/2 +
(Rmov*sqrt3*cos(alpha)*var_add_32)/2)*var_add_20 + 2*(var_add_24/2 - central_chain_l2*sin(beta) +
(Rmov*sqrt3*cos(beta)*cos(gamma))/2)*var_add_18)/var_add_22;
double det1z = -(2*Rmov*var_add_28*var_add_04 + 2*Rmov*(var_add_33 - var_add_31)*var_add_02 -
2*var_add_29*var_add_05)/var_add_08;
double det2z = (2*(var_add_29/2 - (Rmov*sqrt3*sin(beta)*sin(gamma))/2)*var_add_18 + 2*((Rmov*var_add_28)/2 +
(Rmov*sqrt3*var_add_03)/2)*var_add_16 + 2*((Rmov*(var_add_33 - var_add_31))/2 +
(Rmov*sqrt3*var_add_01)/2)*var_add_13)/var_add_19;
double det3z = -(2*((Rmov*var_add_28)/2 + (Rmov*sqrt3*var_add_03)/2)*var_add_21 - 2*(var_add_29/2 -
(Rmov*sqrt3*sin(beta)*sin(gamma))/2)*var_add_18 + 2*((Rmov*(var_add_33 - var_add_31))/2 +
(Rmov*sqrt3*var_add_01)/2)*var_add_20)/var_add_22;
double det=det1x*(det2y*det3z-det2z*det3y)+det1y*(det2z*det3x-det2x*det3z)+det1z*(det2x*det3y-det2y*det3x);

if (det*jacob_sign<=0){return true;}
return false;
}

void calcPoints()
{
Eigen::Matrix3d rotation_matrix;
rotation_matrix=
Eigen::AngleAxisd(EulerAnglesOfPlatform(0), Eigen::Vector3d::UnitZ())*
Eigen::AngleAxisd(EulerAnglesOfPlatform(1), Eigen::Vector3d::UnitX())*
Eigen::AngleAxisd(EulerAnglesOfPlatform(2), Eigen::Vector3d::UnitZ());
}

```

```

unsigned numberOfMovablePoints = this->Geometry.vec_points_init[1].size();
this->vec_points_current[1].resize(this->Geometry.vec_points_init[1].size());
for (int i=0;i<numberOfMovablePoints;i++)
{
    Eigen::Vector3d point;
    this->vec_points_current[1][i] = rotation_matrix*this->Geometry.vec_points_init[1][i];
    this->vec_points_current[1][i] += this->Geometry.vec_points_init[0][4];
}
this->vec_points_current[0] = this->Geometry.vec_points_init[0];
};
double lenght_leg(Eigen::Vector3d point_1,Eigen::Vector3d point_2){
    return sqrt(pow(point_1(0)-point_2(0),2)+pow(point_1(1)-point_2(1),2)+pow(point_1(2)-point_2(2),2));
};
WSProcLocalStructures::boxStatus check_WS(agroGeometryClass<agroGeomParams,agroSimParams>
&InputData,vector<IntervalRC::Interval<double>> &vector_s_variable_real){
    bool check_ok=false;
    bool check__not_ok=false;
    double lmin = InputData.simAndGeomParameters.geomParams.lmin;
    double lmax = InputData.simAndGeomParameters.geomParams.lmax;
    bool orientation_check = InputData.simAndGeomParameters.simParams.orientation_check;
    bool singularity_check = InputData.simAndGeomParameters.simParams.singularity_check;
    int s=InputData.simAndGeomParameters.simParams.s1;
    if (vector_s_variable_real[2].ub-vector_s_variable_real[2].lb<20){s=ceil(s/8);}
    else if (vector_s_variable_real[2].ub-vector_s_variable_real[2].lb<40){s=ceil(s/4);}
    int n1=s,p1=s,t1=s;
    for (int n = -1; (n < s && (!check_ok || !check__not_ok)); n++,n1=n)
    {
        for (int p = -1; (p < s&& (!check_ok || !check__not_ok)); p++,p1=p)
        {
            for (int t = -1; (t < s&& (!check_ok || !check__not_ok)); t++,t1=t)
            {
                bool check_condition=true;
                double alpha=(vector_s_variable_real[0].lb+(vector_s_variable_real[0].ub-vector_s_variable_real[0].lb)*n1/s)*M_PI/180;
                double beta=(vector_s_variable_real[1].lb+(vector_s_variable_real[1].ub-vector_s_variable_real[1].lb)*p1/s)*M_PI/180;
                double gamma=(vector_s_variable_real[2].lb+(vector_s_variable_real[2].ub-vector_s_variable_real[2].lb)*t1/s)*M_PI/180;
                agroPosition<agroGeomParams,agroSimParams> POS1(InputData,{alpha,beta,gamma});
                POS1.calcPoints();
                if (check_condition && orientation_check){
                    double max_error_of_angle_cos =
cos(InputData.simAndGeomParameters.simParams.max_angle_of_tube_error*M_PI/180);
                    int numberOfJoint = InputData.simAndGeomParameters.simParams.tubeOrientationToWhichJointB;
                    double part1=(POS1.vec_points_current[1][5](0)-POS1.vec_points_current[1][3](0))*(-
POS1.vec_points_current[1][numberOfJoint](0)-POS1.vec_points_current[1][3](0));
                    double part2=(POS1.vec_points_current[1][5](1)-POS1.vec_points_current[1][3](1))*(-
POS1.vec_points_current[1][numberOfJoint](1)-POS1.vec_points_current[1][3](1));
                    double l1=sqrt(pow((POS1.vec_points_current[1][5](1)-
POS1.vec_points_current[1][3](1)),2)+pow((POS1.vec_points_current[1][5](0)-POS1.vec_points_current[1][3](0)),2));
                    double l2=sqrt(pow((POS1.vec_points_current[1][numberOfJoint](1)-
POS1.vec_points_current[1][3](1)),2)+pow((POS1.vec_points_current[1][numberOfJoint](0)-POS1.vec_points_current[1][3](0)),2));
                    double orientation_cos=(part1+part2)/(l1*l2);
                    if (orientation_cos<max_error_of_angle_cos)
                    {
                        check_condition=false;
                    }
                }

                if (check_condition)
                {
                    double leg[3];
                    for (int i=0;i<3;i++){
                        leg[i]=lenght_leg(POS1.vec_points_current[0][i],POS1.vec_points_current[1][i]);

                        if(leg[i]<lmin||leg[i]>lmax)
                        {
                            check_condition=false;break;
                        }
                    }
                }
            }
        }
    }
}

```

```

    }
    if (check_condition && singularity_check){check_condition=!POS1.isSingularityZone();}
    if (check_condition)
    {
        //agro_position.show();
        check_ok=true;
    }
    else{check__not_ok=true;}
}
}
}
}
WSProcLocalStructures::boxStatus statusOfThisBox;
if (!check_ok)
{
    statusOfThisBox = WSProcLocalStructures::boxDelete;
}
else if (!check__not_ok)
{
    statusOfThisBox = WSProcLocalStructures::boxInsert;
}
else{
    statusOfThisBox = WSProcLocalStructures::boxDivide;
}
return statusOfThisBox;
}
bool check_WS_POS_interference(agroGeometryClass<agroGeomParams,agroSimParams> &InputData,vector<double
>vector_s_variable_real_1){
    double alpha=vector_s_variable_real_1[0]*M_PI/180;
    double beta=vector_s_variable_real_1[1]*M_PI/180;
    double gamma=vector_s_variable_real_1[2]*M_PI/180;
    agroPosition<agroGeomParams,agroSimParams> POS1(InputData,{alpha,beta,gamma});
    POS1.calcPoints();
    bool interference_flag = POS1.isInterferenceForPosition();
    if (interference_flag && interfer_show_counter<5)
    {
        interfer_show_counter++;
    }
    return (!interference_flag);
    //return (!agro_position.formulateAndCheckIntersections());
}
vector<vector<int>> POS_WS_3_to_POS_WS_3_calculation(vector<double
>vector_s_variable_real_in,agroGeometryClass<agroGeomParams,agroSimParams> &InputData,double accuracy)
{
    double alpha=vector_s_variable_real_in[0]*M_PI/180;
    double beta=vector_s_variable_real_in[1]*M_PI/180;
    double gamma=vector_s_variable_real_in[2]*M_PI/180;
    agroPosition<agroGeomParams,agroSimParams> POS1(InputData,{alpha,beta,gamma});
    POS1.calcPoints();
    vector<S_variable_real> line_segment(3);

    line_segment[0].lb=POS1.vec_points_current[1][4](0)/accuracy;
    line_segment[1].lb=POS1.vec_points_current[1][4](1)/accuracy;
    line_segment[2].lb=POS1.vec_points_current[1][4](2)/accuracy;
    line_segment[0].rb=POS1.vec_points_current[1][5](0)/accuracy;
    line_segment[1].rb=POS1.vec_points_current[1][5](1)/accuracy;
    line_segment[2].rb=POS1.vec_points_current[1][5](2)/accuracy;
    vector<vector<int>> interferenced_boxes = RCgeometryFuncs::detect_interferenced_boxes_in_3D(line_segment);
    return interferenced_boxes;
}
template<typename GeomParamsStruct, typename simParamsStruct>
class Agro_mechanism:public
mechanism<agroGeometryClass<GeomParamsStruct,simParamsStruct>,GeomParamsStruct,simParamsStruct>
{
public:
    POS_3 WS_euler_angles;
    POS_3 WS_end_effector;

    bisection_class bisection;

```

```

static simAndGeomParametersClass<agroGeomParams,agroSimParams> getAutoParams(){
    simAndGeomParametersClass<agroGeomParams,agroSimParams> params;
    params.geomParams.Rmov=200;
    params.geomParams.Rfix=300;
    params.geomParams.lmin=360;
    params.geomParams.lmax=600;
    params.geomParams.central_chain_11=500;
    params.geomParams.central_chain_12=0;
    params.geomParams.lEndEffectorMin=500;
    params.geomParams.lEndEffectorMax=1000;
    params.geomParams.dLinks=20;
    params.geomParams.angle_limit=10;
    params.simParams.s1=64;
    params.simParams.s2=10;
    params.simParams.accuracy_for_euler_platform_coords=5;
    params.simParams.accuracy_for_end_effector_coords=5;
    params.simParams.jacobSign=1;
    params.simParams.eulers_range=360;
    params.simParams.orientation_check=0;
    params.simParams.tubeOrientationToWhichJointB=1;
    params.simParams.max_angle_of_tube_error=10;
    params.simParams.singularity_check=0;
    return params;
}
void setParameters(simAndGeomParametersClass<agroGeomParams,agroSimParams> params = getAutoParams())
{
    bisection.vec_input_intervals.push_back(WS_proc_func::makeIntervalForGrid("alpha",-
params.simParams.eulers_range/2,params.simParams.eulers_range/2,params.simParams.accuracy_for_euler_platform_coords));
    bisection.vec_input_intervals.push_back(WS_proc_func::makeIntervalForGrid("beta",-
params.simParams.eulers_range/2,params.simParams.eulers_range/2,params.simParams.accuracy_for_euler_platform_coords));
    bisection.vec_input_intervals.push_back(WS_proc_func::makeIntervalForGrid("gamma",-
params.simParams.eulers_range/2,params.simParams.eulers_range/2,params.simParams.accuracy_for_euler_platform_coords));

    this->initMechanismGeometry(params);
};

void loadWSOrBisectAndTransferToPOS(){
    WS_euler_angles.WSname="Agro_WS_euler_angles";WS_euler_angles.dir_path=this->mechanism_dir_path();
    string Path_to_WS=WS_euler_angles.dir_path+"/"+WS_euler_angles.WSname+".txt";
    ifstream f_WS_POS_3_singular_minus_interference(Path_to_WS);bool
check_ifstream=f_WS_POS_3_singular_minus_interference.good();
    if(!check_ifstream)
    {
        WSProcLocalStructures::S_vector_WS_output<vector<IntervalRC::Interval<int>>> WSAfterBisection =
bisection.run(check_WS,false,this->mechanismGeometry);
        WS_euler_angles.transfer_to_Partially_ordered_set<vector<IntervalRC::Interval<int>>>(WSAfterBisection,
check_WS_POS_interference,this->mechanismGeometry);//check_WS_POS_interference
        WS_euler_angles.save_POS_WS_3_to_file();
        WS_euler_angles.export_to_STL();
    }
    else
    {
        WS_euler_angles.read_POS_WS_3_from_file();
    }
}
void loadOrTransferPOS_WS_3_euler_angles_to_POS_WS_3()
{
    WS_end_effector.WSname="Agro_WS_end_effector";
    WS_end_effector.dir_path=this->mechanism_dir_path();
    string Path_to_WS=WS_end_effector.dir_path+"/"+WS_end_effector.WSname+".txt";
    ifstream f_WS_POS_3_singular_minus_interference(Path_to_WS);
    bool check_ifstream=f_WS_POS_3_singular_minus_interference.good();
    if(!check_ifstream)
    {
        WS_end_effector.POS_WS_3_to_POS_WS_3_parameters.coord3_interval[0]=-this-
>mechanismGeometry.simAndGeomParameters.geomParams.lEndEffectorMax;
        WS_end_effector.POS_WS_3_to_POS_WS_3_parameters.coord3_interval[1]=this-
>mechanismGeometry.simAndGeomParameters.geomParams.lEndEffectorMax;
    }
}

```

```

    WS_end_effector.POS_WS_3_to_POS_WS_3_parameters.accuracy=this-
>mechanismGeometry.simAndGeomParameters.simParams.accuracy_for_end_effector_coords;
    WS_end_effector.POS_WS_3_to_POS_WS_3_parameters.grid_size=this-
>mechanismGeometry.simAndGeomParameters.simParams.s2;
    WS_end_effector.POS_WS_3_to_POS_WS_3(WS_euler_angles.WS, this->mechanismGeometry,
POS_WS_3_to_POS_WS_3_calculation);

    WS_end_effector.save_POS_WS_3_to_file();
    //WS_end_effector.Delone_triangulation_and_export();
    WS_end_effector.export_to_STL();
}
else
{
    WS_end_effector.read_POS_WS_3_from_file();
}
}
private:
};

bool isParametersGoodv3(simAndGeomParametersClass<agroGeomParams,agroSimParams> params, int modeOfCondition){
    Agro_mechanism<agroGeomParams,agroSimParams> agro1;
    agro1.setParameters(params);
    double lmin = params.geomParams.lmin;
    double lmax = params.geomParams.lmax;
    bool okFlag = false;
    bool okFlagArray[8] = { false,false,false,false,false,false,false,false };
    int minSectors = 4;
    if (modeOfCondition==0){
        minSectors = 8;
    }
#pragma omp parallel for schedule(dynamic) default(none) shared(cout,lmax,lmin,okFlagArray,agro1,modeOfCondition)
    for (int i=0;i<1728;i++)
    {
        int betaI = i%12;
        int alphaI = ((i - betaI)/12)%12;
        int gammaI = i/144;
        double beta = (-180 + 30 * betaI)*M_PI/180;
        double gamma = (-180 + 30 * alphaI)*M_PI/180;
        double alpha = (-180 + 30 * gammaI)*M_PI/180;
        agroPosition<agroGeomParams,agroSimParams> Pos1(agro1.mechanismGeometry,{alpha,beta,gamma});
        Pos1.calcPoints();
        bool conditionForPoint1 = Pos1.vec_points_current[1][4](2)<1500;
        bool conditionForPoint2 = Pos1.vec_points_current[1][5](2)>3500;
        bool conditionForPoint;
        int addNumberToIndex=0;
        if (modeOfCondition==0){
            conditionForPoint= conditionForPoint1 || conditionForPoint2;
        }
        if (modeOfCondition==1){
            conditionForPoint= conditionForPoint1;
        }
        if (modeOfCondition==2){
            conditionForPoint= conditionForPoint2;
        }
        if (conditionForPoint){
            bool check_condition1 = true;
            double leg[3];
            for (int i=0;i<3;i++){
                leg[i]=lenght_leg(Pos1.vec_points_current[0][i],Pos1.vec_points_current[1][i]);
                if(leg[i]<lmin||leg[i]>lmax)
                {
                    check_condition1=false;break;
                }
            }
        }
        if (check_condition1)
        {
            check_condition1 = !Pos1.isInterferenceForPosition();
        }
    }
}

```

```

    if (check_condition1){
#pragma omp critical
{
    if (conditionForPoint1){
        addNumberToIndex = 4;
    }
    if (Pos1.vec_points_current[1][4](0)>0 && Pos1.vec_points_current[1][4](1)>0)
    {
        okFlagArray[0+addNumberToIndex]=true;
    }
    else if (Pos1.vec_points_current[1][4](0)<0 && Pos1.vec_points_current[1][4](1)>0)
    {
        okFlagArray[1+addNumberToIndex]=true;
    }
    else if (Pos1.vec_points_current[1][4](0)>0 && Pos1.vec_points_current[1][4](1)<0)
    {
        okFlagArray[2+addNumberToIndex]=true;
    }
    else if (Pos1.vec_points_current[1][4](0)<0 && Pos1.vec_points_current[1][4](1)<0)
    {
        okFlagArray[3+addNumberToIndex]=true;
    }
}
}
}
int numberOfGoodGlag=0;
for (int i=0; i<8;i++){
    if (okFlagArray[i]){
        numberOfGoodGlag++;
    }
}
if (numberOfGoodGlag>=minSectors){
    okFlag = true;
}
return okFlag;
}
int isParametersGoodEu(simAndGeomParametersClass<agroGeomParams,agroSimParams> params){
    Agro_mechanism<agroGeomParams,agroSimParams> agro1;
    agro1.setParameters(params);
    double lmin = params.geomParams.lmin;
    double lmax = params.geomParams.lmax;
    int sumOfGoodAngles=0;
#pragma omp parallel for schedule(dynamic) default(none) shared(cout,lmax,lmin,agro1,sumOfGoodAngles)
    for (int i=0;i<1728;i++)
    {
        int betaI = i% 12;
        int alphaI = ((i - betaI)/12)% 12;
        int gammaI = i/144;
        double beta = (-180 + 30 * betaI)*M_PI/180;
        double gamma = (-180 + 30 * alphaI)*M_PI/180;
        double alpha = (-180 + 30 * gammaI)*M_PI/180;
        agroPosition<agroGeomParams,agroSimParams> Pos1(agro1.mechanismGeometry,{alpha,beta,gamma});
        Pos1.calcPoints();
        bool check_condition1 = true;
        double leg[3];
        for (int i=0;i<3;i++){
            leg[i]=length_leg(Pos1.vec_points_current[0][i],Pos1.vec_points_current[1][i]);
            if(leg[i]<lmin||leg[i]>lmax)
            {
                check_condition1=false;break;
            }
        }
    }
    if (check_condition1)
    {
        check_condition1 = !Pos1.isInterferenceForPosition();
    }
}
if (check_condition1){

```

```

#pragma omp critical
{
    sumOfGoodAngles++;
}
}
return sumOfGoodAngles;
}
void caclEulerAngles(vector<Eigen::Vector3d> unitVectors){
    Eigen::Matrix3d A{
        {unitVectors[0](0),unitVectors[1](0),unitVectors[2](0)},
        {unitVectors[0](1),unitVectors[1](1),unitVectors[2](1)},
        {unitVectors[0](2),unitVectors[1](2),unitVectors[2](2)}
    };
    cout<< A.eulerAngles(2,0,2)<<endl;
}
void runParametersSelectionV3(){
    simAndGeomParametersClass<agroGeomParams,agroSimParams> agroParams;
    agroParams.geomParams.lEndEffectorMin=2000;
    agroParams.geomParams.lEndEffectorMax=3000;
    agroParams.geomParams.dLinks=80;
    vector<simAndGeomParametersClass<agroGeomParams,agroSimParams>> parametersVector;
    double step = 100;
    for (agroParams.geomParams.Rmov=200;agroParams.geomParams.Rmov<=1000;agroParams.geomParams.Rmov+=step){
        cout<< "Rmov " << agroParams.geomParams.Rmov << " / 350" << endl;
        for
(agroParams.geomParams.Rfix=agroParams.geomParams.Rmov;agroParams.geomParams.Rfix<=1000;agroParams.geomParams.Rfix+=
step){
            cout<< "Rfix " << agroParams.geomParams.Rfix << " / 650 " << endl;
            for (agroParams.geomParams.lmin=200;agroParams.geomParams.lmin<=1200;agroParams.geomParams.lmin+=step){
                for (agroParams.geomParams.lmax=agroParams.geomParams.lmin+200;(agroParams.geomParams.lmax<=1400 &&
agroParams.geomParams.lmax<agroParams.geomParams.lmin/0.6);agroParams.geomParams.lmax+=step){
                    for
(agroParams.geomParams.central_chain_11=200;agroParams.geomParams.central_chain_11<=1400;agroParams.geomParams.central_ch
ain_11+=step){
                        for
(agroParams.geomParams.central_chain_12=0;agroParams.geomParams.central_chain_12<=800;agroParams.geomParams.central_chain_
12+=step){
                            if (agroParams.geomParams.central_chain_11+agroParams.geomParams.central_chain_12<=3100)
                            {
                                {
                                    if (isParametersGoodv3(agroParams,0))
                                    {
                                        cout<< agroParams.geomParams.Rmov << " " << agroParams.geomParams.Rfix << " " <<
agroParams.geomParams.lmin << " " << agroParams.geomParams.lmax << " " << agroParams.geomParams.central_chain_11 << " "
<< agroParams.geomParams.central_chain_12<<endl;
                                        parametersVector.push_back(agroParams);
                                    }
                                }
                            }
                        }
                    }
                }
            }
        }
    }
}
cout<<"_____ " <<endl;
for(int vecParamIndex = 0; vecParamIndex<parametersVector.size(); vecParamIndex++){
    agroParams = parametersVector[vecParamIndex];
    int numOfEuGood = isParametersGoodEu(agroParams);
    if (numOfEuGood>150){
        cout<< agroParams.geomParams.Rmov << " " << agroParams.geomParams.Rfix << " " << agroParams.geomParams.lmin << "
" << agroParams.geomParams.lmax << " " << agroParams.geomParams.central_chain_11 << " " <<
agroParams.geomParams.central_chain_12<< " " << numOfEuGood <<endl;
    }
}
}
template < typename T>
vector<double> testFuncMultiobj(vector <double> local_parameters,T Data_for_target_function){
    return (vector<double>{2*local_parameters[0]+local_parameters[1],-local_parameters[0]-3*local_parameters[1]});
}

```

```

}
void testMultiobj(){
    multiObjectiveOptimization<bool> testMObj({{0,10},{0,10}}, testFuncMultiobj);
    testMObj.visualization = true;
    testMObj.runDistributed(5);
}
int calcPossibleAngles(simAndGeomParametersClass<agroGeomParams,agroSimParams> params){
    Agro_mechanism<agroGeomParams,agroSimParams> agro1;
    agro1.setParameters(params);
    double lmin = params.geomParams.lmin;
    double lmax = params.geomParams.lmax;
    int sumOfGoodAngles=0;
    for (int i=0;i<1728;i++){
        {
            int betaI = i% 12;
            int alphaI = ((i - betaI)/12)% 12;
            int gammaI = i/144;

            double beta = (-180 + 30 * betaI)*M_PI/180;
            double gamma = (-180 + 30 * alphaI)*M_PI/180;
            double alpha = (-180 + 30 * gammaI)*M_PI/180;

            agroPosition<agroGeomParams,agroSimParams> Pos1(agro1.mechanismGeometry,{alpha,beta,gamma});
            Pos1.calcPoints();
            bool check_condition1 = true;
            double leg[3];
            for (int i=0;i<3;i++){
                leg[i]=length_leg(Pos1.vec_points_current[0][i],Pos1.vec_points_current[1][i]);
                if(leg[i]<lmin||leg[i]>lmax)
                {
                    check_condition1=false;break;
                }
            }

            if (check_condition1)
            {
                check_condition1 = !Pos1.isInterferenceForPosition();
            }
            if (check_condition1){

                sumOfGoodAngles++;
            }
        }
    }
    return sumOfGoodAngles;
}
void updateParetoSetDetectAlgs(vector<double> &FTargets,vector<double> &Params,vector<ParetoSetS> &ParetoSetForUpdate, int
algsCounter, vector<int> &vecOfAlgs){
    bool isAdd=true;
    for (int i=0;i<ParetoSetForUpdate.size();i++){
        bool isAllCoeffiatsBad = true;
        bool isAllCoeffiatsBetter = true;
        for (int i1=0; i1<FTargets.size();i1++){
            if (FTargets[i1]<ParetoSetForUpdate[i].Front[i1]){
                isAllCoeffiatsBad = false;
            }
            if (FTargets[i1]>ParetoSetForUpdate[i].Front[i1]){
                isAllCoeffiatsBetter = false;
            }
        }
    }
    if (isAllCoeffiatsBetter){
        vecOfAlgs.erase(vecOfAlgs.begin()+i);
        vector<int>(vecOfAlgs).swap(vecOfAlgs);
        ParetoSetForUpdate.erase(ParetoSetForUpdate.begin()+i);
        vector<ParetoSetS>(ParetoSetForUpdate).swap(ParetoSetForUpdate);
        i--;
    }
    if (isAllCoeffiatsBad){
        isAdd = false;
    }
}

```

```

    break;
    }
    }
if (isAdd){
    ParetoSetS S1;
    S1.Front = FTargets;
    S1.Params = Params;
    ParetoSetForUpdate.push_back(S1);
    vecOfAlgs.push_back(algsCounter);
}
}
vector<double> FuncAgroMultiobj(vector<double> local_parameters,double penaltyForRadii){
    simAndGeomParametersClass<agroGeomParams,agroSimParams> agroParams;
    agroParams.geomParams.lEndEffectorMin=2000;
    agroParams.geomParams.lEndEffectorMax=3000;
    agroParams.geomParams.dLinks=80;
    agroParams.geomParams.Rmov=local_parameters[0];
    agroParams.geomParams.Rfix=local_parameters[1];
    agroParams.geomParams.lmin=local_parameters[2];
    agroParams.geomParams.lmax=local_parameters[3];
    if (agroParams.geomParams.lmax<agroParams.geomParams.lmin/0.6){
        agroParams.geomParams.central_chain_l1=local_parameters[4];
        agroParams.geomParams.central_chain_l2=local_parameters[5];
        double ecual_r_default=0.7;
        double ecual_r_real=agroParams.geomParams.Rmov/agroParams.geomParams.Rfix;
        double penalty_1 = 0;
        if (ecual_r_real<ecual_r_default)
        {
            penalty_1+=penaltyForRadii*(ecual_r_default/ecual_r_real-1);
        }
        else
        {
            penalty_1+=penaltyForRadii*(ecual_r_real/ecual_r_default-1);
        }
        double K1 = penalty_1 + agroParams.geomParams.Rmov + agroParams.geomParams.Rfix + agroParams.geomParams.lmin +
        agroParams.geomParams.lmax;
        double K2 = 1728 - calcPossibleAngles(agroParams);
        return vector<double>{K1,K2};
    }
    else{
        return vector<double>{ 10000,10000 };
    }
    return (vector<double>{2*local_parameters[0]+local_parameters[1],-local_parameters[0]-3*local_parameters[1]});
}
void exportResultsAndAnalyzeAlgs(vector<vector<ParetoSetS>> AllParetoSets){
    vector<ParetoSetS> bestPareto;
    vector<int> vecOfAlgs;
    for (int algsCounter=0; algsCounter<AllParetoSets.size();algsCounter++){
        for (int paretoPointsCounter=0; paretoPointsCounter<AllParetoSets[algsCounter].size();paretoPointsCounter++){
            updateParetoSetDetectAlgs(AllParetoSets[algsCounter][paretoPointsCounter].Front,AllParetoSets[algsCounter][paretoPointsCounter].Params,bestPareto,algsCounter,vecOfAlgs);
        }
    }
    AllParetoSets.push_back(bestPareto);
    vector<vector<vector<double>>> ParetoSetAllAlgsVisualization;
    for (int stepsCounter = 0; stepsCounter< AllParetoSets.size(); stepsCounter++){
        ParetoSetAllAlgsVisualization.push_back(vector<vector<double>>{ });
        for (int pointsCounter = 0; pointsCounter< AllParetoSets[stepsCounter].size(); pointsCounter++){
            ParetoSetAllAlgsVisualization[stepsCounter].push_back(AllParetoSets[stepsCounter][pointsCounter].Front);
        }
    }
}
{
    string fullPath = "log.txt";
    FILE *log= fopen(fullPath.c_str(),"a");
    ofstream log_fout (fullPath, std::ios::binary|std::ios::app);
    for (int stepsCounter = 0; stepsCounter< AllParetoSets.size(); stepsCounter++){
        log_fout<<"Алгоритм "<< stepsCounter + 1 <<endl;
    }
}

```

```

        for (int pointsCounter = 0; pointsCounter < AllParetoSets[stepsCounter].size(); pointsCounter++){
            log_fout<<"Точка " << pointsCounter + 1 <<endl;
            for (int criterionsCounter = 0; criterionsCounter < AllParetoSets[stepsCounter][pointsCounter].Front.size();
criterionsCounter ++){
                log_fout<< AllParetoSets[stepsCounter][pointsCounter].Front[criterionsCounter ]<< " ";
            }
            log_fout<< endl;
            for (int parametersCounter = 0; parametersCounter < AllParetoSets[stepsCounter][pointsCounter].Params.size();
parametersCounter ++){
                log_fout<< AllParetoSets[stepsCounter][pointsCounter].Params[parametersCounter ]<< " ";
            }
            log_fout<< endl;
        }
        log_fout<<"Соотнесение точек и алгоритмов " <<endl;
        for (int pointsCounter = 0; pointsCounter < vecOfAlgs.size(); pointsCounter ++){
log_fout<< vecOfAlgs[pointsCounter ]<< " ";
        }
log_fout<< endl;
        fclose(log);
    }
    RCexportFuncs::VVVvectorToJson(ParetoSetAllAlgsVisualization,"pointsData");
    RCexportFuncs::save_labels({"JointGraph","K1","K2","GA","GWO","PSO","Filtered"});
    RCexportFuncs::runPy("showPareto3Algs");
}
void agroMultiobj(){
    double penaltyForRadii = 1000;
    vector<vector<double>> Params = {{ 200,1000},{ 200,1000},{ 200,1200},{ 400,1400},{ 200,1400},{ 0,800}};
    int coefGrid = 2;
    multiObjectiveOptimization<double> testMObjGA(Params, FuncAgroMultiobj,penaltyForRadii);
    testMObjGA.paramsForTargetFuncs.algorithmType = GA;
    testMObjGA.paramsForTargetFuncs.GA.genetation_amount=1;
    testMObjGA.paramsForTargetFuncs.GA.initial_population_amount=50;
    testMObjGA.visualization = true;
    testMObjGA.labelsForVisualization[0] = "GA_"+to_string(coefGrid);
    testMObjGA.runDistributed(coefGrid);
    multiObjectiveOptimization<double> testMObjGWO(Params, FuncAgroMultiobj,penaltyForRadii);
    testMObjGWO.paramsForTargetFuncs.algorithmType = GWO;
    testMObjGWO.paramsForTargetFuncs.GWO.genetation_amount=1;
    testMObjGWO.paramsForTargetFuncs.GWO.initial_population_amount=50;
    testMObjGWO.labelsForVisualization[0] = "GWO_"+to_string(coefGrid);
    testMObjGWO.visualization = true;
    testMObjGWO.runDistributed(coefGrid);
    multiObjectiveOptimization<double> testMObjPSO_G(Params, FuncAgroMultiobj,penaltyForRadii);
    testMObjPSO_G.paramsForTargetFuncs.algorithmType = PSO_G;
    testMObjPSO_G.paramsForTargetFuncs.PSO_groups.genetation_amount=1;
    testMObjPSO_G.paramsForTargetFuncs.PSO_groups.initial_population_in_group_amount=20;
    testMObjPSO_G.labelsForVisualization[0] = "PSO_G_"+to_string(coefGrid);
    testMObjPSO_G.visualization = true;
    testMObjPSO_G.runDistributed(coefGrid);
    exportResultsAndAnalyzeAlgs({testMObjGA.ParetoSet,testMObjGWO.ParetoSet,testMObjPSO_G.ParetoSet});
        cout<< "Done" <<endl;
    }
}

int main()
{
    RCserviceFuncs::func_program_begin();
    agroMultiobj();
    RCserviceFuncs::func_log_end();
    return 0;
}

```

ПРИЛОЖЕНИЕ 2

**Программа для оптимизации геометрических и конструктивных параметров
захватного устройства для Matlab**

```

D1 = 0.12;
D2 = 0.04;

reInterval = [(D1-D2)/2+0.00001;D1/2];
rpInterval = [0.00001;1];
alphaInterval = [0.00001;pi/2-0.00001];
kInterval = [0.00001;100000];
ObjectiveFunction = @calculateObjectiveFunctions4; % Function handle to the fitness function
numberOfVariables = 4; % Number of decision variables
lb = [reInterval(1) rpInterval(1) kInterval(1) alphaInterval(1)];
ub = [reInterval(2) rpInterval(2) kInterval(2) alphaInterval(2)];
A = [];
b = [];
Aeq = [];
beq = [];
options = gaoptimset('PlotFcns',@gaplotpareto, "PopulationSize",5000, "Generations", 50, "StallGenLimit", 30);
[x,Fval,exitFlag,Output] = gamultiobj(ObjectiveFunction,numberOfVariables,A,b,Aeq,beq,lb,ub,options);
fprintf('The number of points on the Pareto front was: %d\n', size(x,1));
fprintf('The number of generations was : %d\n', Output.generations);
for i=1:4
    figure
        boxplot(x(:,i));
end
figure
pareto = Fval
results = sortrows([pareto x]);
re=results(1,3)
rp=results(1,4)
k=results(1,5)
alpha=results(1,6)
Mbest = results(1,1)
SecondCriteria = results(1,2)
Fe = 2.4;
D1 = 0.12;
D2 = 0.04;
Fpmin = 0.1;
nu = 0.88;
z2 = 10;
z1 = 50;
theta=asin(((D1/2)-(D2/2))/re);
Pb = 0.052;
fii=0.02;
y = zeros(2,1);
rd = Pb/(cot(alpha)*sin(theta));
xt=rp;
yt=0;
n=10;
betaRange = [0;theta];
Mlimit=9999999;
M = [Mlimit,Mlimit];
steps = 100
for i=1:steps
    beta(i)=betaRange(1)+(betaRange(2)-betaRange(1))*(i-1)/(steps-1)
end
for i=1:steps
betaDegree(i)=beta(i)*180/pi
Fp=(-k*sqrt((rp*cos(theta-beta(i))+xt)^2+(rp*sin(theta-beta(i))-yt)^2))*cos(theta-beta(i))*(rp/re);

```

```

Rd=Fe+Fp;
Ff=fii*Rd;
Mup=((Fe*re+Fp*rp+Ff*rd)*(Pb*z1*n)/(rd*cos(pi/2-alpha-beta(i))*cos(alpha+beta(i))));
Mdown=2*pi*z2*nu;
M(i)=Mup/Mdown
end
figure
plot(betaDegree, M)
function y = calculateObjectiveFunctions4(x)
re=x(1);
rp=x(2);
k=x(3);
alpha=x(4);
Fe = 2.4;
D1 = 0.12;
D2 = 0.04;
Fpmin = 0.1;
nu = 0.88;
z2 = 10;
z1 = 50;
theta=asin(((D1/2)-(D2/2))/re);
Pb = 0.052;
fii=0.02;
y = zeros(2,1);
rd = Pb/(cot(alpha)*sin(theta));
xt=rp;
yt=0;
n=10;
beta = [0;theta];
Mlimit=9999999;
M = [Mlimit,Mlimit];
if (theta<pi/2-alpha && rp>0 && rp<rd && yt<rd && rd<re)
for i=1:2
Fp=(-k*sqrt((rp*cos(theta-beta(i))+xt)^2+(rp*sin(theta-beta(i))-yt)^2))*cos(theta-beta(i))*(rp/re);
Rd=Fe+Fp;
Ff=fii*Rd;
Mup=((Fe*re+Fp*rp+Ff*rd)*(Pb*z1*n)/(rd*cos(pi/2-alpha-beta(i))*cos(alpha+beta(i))));
Mdown=2*pi*z2*nu;
if (Mup / Mdown>0 && Mup / Mdown<10)
M(i) = Mup / Mdown;
end
end
Dout=D1+2*rd*sin(theta);
D_outOptions=[D1+2*rd*sin(theta)];
D_out=max(D_outOptions);
H_out=Pb*3;
V_out=pi*H_out*((D_out^2)/4);
end
y = zeros(2,1);
y(1) = max(M);
if (y(1)==Mlimit)
y(2)=Mlimit;
else
y(2)=abs(V_out * 1000 * 1000 * 1000);
end
end
end

```

РОССИЙСКАЯ ФЕДЕРАЦИЯ

**ПАТЕНТ**НА ИЗОБРЕТЕНИЕ
№ 2829514**АВТОМАТИЗИРОВАННОЕ ЗАХВАТНОЕ
УСТРОЙСТВО ДЛЯ СБОРА ПЛОДОВ**

Патентообладатель: *Федеральное государственное бюджетное образовательное учреждение высшего образования "Белгородский государственный технологический университет им. В.Г. Шухова" (RU)*

Авторы: *Рыбак Лариса Александровна (RU), Волошкин Артем Александрович (RU), Гапоненко Елена Владимировна (RU), Малышев Дмитрий Иванович (RU), Перевузник Виктория Сергеевна (RU), Черкасов Владислав Витальевич (RU), Скитова Валерия Михайловна (RU)*

Заявка № 2024116272

Приоритет изобретения **13 июня 2024 г.**Дата государственной регистрации
в Государственном реестре изобретений
Российской Федерации **31 октября 2024 г.**Срок действия исключительного права
на изобретение истекает **13 июня 2044 г.**

*Руководитель Федеральной службы
по интеллектуальной собственности*

Ю.С. Зубов



Общество с ограниченной ответственностью
"РОБОКОМПОНЕНТ",
ИНН 9703113163, КПП 772201001,
111024, Город Москва, вн.тер.г. муниципальный
округ Лефортово, ул.Авиамоторная, д.73А, стр. 6

Исх. от «17» января 2025г. № 6

СПРАВКА

об использовании результатов диссертационной работы

Волошкина Артема Александровича по теме: «Методы проектирования и оптимизации автономной робототехнической системы для сбора фруктов» в проектно-конструкторской и производственной деятельности ООО «РобоКомпонент»

Полученные в диссертационной работе результаты использованы в ООО «РобоКомпонент». При непосредственном участии Волошкина А.А. синтезирована структура нового захватного устройства, на основе которой разработана параметризованная математическая модель. Выполнена оптимизация геометрических и конструктивных параметров, а также разработана электронно-цифровая модель. Разработанное захватное устройство предназначено для сбора фруктов, а также может быть применено для использования в промышленной робототехнике, которое представляет собой перспективную альтернативу существующим решениям

Генеральный директор
ООО «РобоКомпонент»

Шмаков Д.А.



«УТВЕРЖДАЮ»

Проректор по цифровой
трансформации и
образовательной деятельности
БГТУ им. В.Г. Шухова



В. М. Поляков

4 января 2025 г.

АКТ

об использовании результатов диссертационной работы

Волошкина Артема Александровича в учебном процессе БГТУ им. В.Г. Шухова

Комиссия в составе:

Дуюн Татьяна Александровна – заведующий кафедрой технологии машиностроения, д.т.н., профессор – председатель комиссии,

Воронкова Марина Николаевна – доцент кафедры технологии машиностроения, к.т.н., доцент – член комиссии,

Перевузник Виктория Сергеевна - доцент кафедры технологии машиностроения, к.т.н. – член комиссии.

составили настоящий акт о том, что результаты диссертационной работы Волошкина А.А. «Методы проектирования и оптимизации автономной робототехнической системы для сбора фруктов» используются в учебном процессе в БГТУ им. В.Г. Шухова при подготовке студентов по направлению подготовки 15.03.05 «Конструкторско-технологическое обеспечение машиностроительных производств» - профиль программы «Технология машиностроения» и направлению подготовки 15.04.06 «Мехатроника и робототехника» - профиль программы «Робототехника и искусственный интеллект»

Председатель комиссии:

Т.А. Дуюн

Члены комиссии:

М.Н. Воронкова

В.С. Перевузник