





**МИНОБРНАУКИ РОССИИ**  
Федеральное государственное бюджетное образовательное  
учреждение высшего образования  
«Национальный исследовательский университет «МЭИ»

Институт \_\_\_\_\_ ИВТ  
Кафедра \_\_\_\_\_ УИТ

**ЗАДАНИЕ**  
**НА ВЫПУСКНУЮ КВАЛИФИКАЦИОННУЮ РАБОТУ**  
**(бакалаврскую работу)**  
**УПРАВЛЕНИЕ В ТЕХНИЧЕСКИХ СИСТЕМАХ**

Направление \_\_\_\_\_ 27.03.04 «Управление в технических системах»  
(код и наименование)

Направленность (профиль) \_\_\_\_\_

Системы и технические средства автоматизации и управления

Форма обучения \_\_\_\_\_ очная  
(очная/очно-заочная/заочная)

Тема: \_\_\_\_\_ Разработка математической модели манипулятора типа «трипод»

Студент \_\_\_\_\_ А-02-16 \_\_\_\_\_ Чугунова А.В.  
группа \_\_\_\_\_ подпись \_\_\_\_\_ фамилия и инициалы

Научный \_\_\_\_\_ К.Т.Н. \_\_\_\_\_ ст. преподаватель \_\_\_\_\_ Шилин Д.В.  
руководитель \_\_\_\_\_ уч. степень \_\_\_\_\_ должность \_\_\_\_\_ подпись \_\_\_\_\_ фамилия и инициалы

Консультант \_\_\_\_\_ \_\_\_\_\_ \_\_\_\_\_ \_\_\_\_\_  
уч. степень \_\_\_\_\_ должность \_\_\_\_\_ подпись \_\_\_\_\_ фамилия и инициалы

Консультант \_\_\_\_\_ \_\_\_\_\_ \_\_\_\_\_ \_\_\_\_\_  
уч. степень \_\_\_\_\_ должность \_\_\_\_\_ подпись \_\_\_\_\_ фамилия и инициалы

Зав. кафедрой \_\_\_\_\_ Д.Т.Н. \_\_\_\_\_ доцент \_\_\_\_\_ Бобряков А.В.  
уч. степень \_\_\_\_\_ звание \_\_\_\_\_ подпись \_\_\_\_\_ фамилия и инициалы

Место выполнения работы \_\_\_\_\_ кафедра управления и интеллектуальных  
технологий

## СОДЕРЖАНИЕ РАЗДЕЛОВ ЗАДАНИЯ И ИСХОДНЫЕ ДАННЫЕ

|  |
|--|
| Введение   |
| 1. Описание выбранной конструкции промышленного манипулятора.<br>Обзор и описание компонентов в составе промышленного манипулятора.              |
| 2. Решение прямой задачи кинематики. Решение обратной задачи кинематики.<br>Проверка работоспособности кинематической модели на крайних условиях |
| 3. Разработка динамической модели манипулятора. Построение и исследование переходных характеристик.  |
| 4. Разработка пользовательского приложения для расчетов.   |
| Заключение   |
| Библиографический список   |
| Приложение 1. Листинг кода, разработанного приложения  |
| Исходные данные:<br>ПО MATLAB  |

### ПЕРЕЧЕНЬ ГРАФИЧЕСКОГО МАТЕРИАЛА

|                                  |       |
|----------------------------------|-------|
| Количество листов                | _____ |
| Количество слайдов в презентации | _____ |

### РЕКОМЕНДУЕМАЯ ЛИТЕРАТУРА

|   |
|---|
| 1) ГОСТ 7.1-84, Библиографическое описание документа. Общие требования и правила составления.   |
| 2) ГОСТ 7.9-95. Реферат и аннотация. Общие требования.  |
| 3) ГОСТ 7.5-98. Журналы, сборники, информационные издания. Издательское оформление публикуемых материалов.  |
| 4) Электрические схемы подключения станции хранения манипулятора типа «трипод» [Электронный ресурс] // URL: <a href="https://www.festo-didactic.com/ru-ru">https://www.festo-didactic.com/ru-ru</a> (дата обращения: 20.12.2019). |
| 5) Макаров И.М. Системные принципы создания гибких автоматизированных производств. М.: Высш. шк., 1986. 175 с   |
| 6) Управление робототехническими системами и гибкими автоматизированными производствами / И.М.Макаров, В.З.Рахманкулов, В.М.Назаретов и др. М.: Высш. шк., 1986. 159 с  |
| 7) Дж.Дэбни, Т.Харман Simulink 4. Секреты мастерства. Издательство: Бином. Лаборатория знаний, 2003 г. Твердый переплет, 404 стр. ISBN 5-94774-048-6, 0-13-1-017085-2. Тираж: 3000 экз. Формат: 70x100/16                         |
| 8) В.П. Дьяконов. MATLAB 6.5 SP1/7 + Simulink 5/6. Основы применения. М.: СОЛОН--Пресс, 2005. - 800с  |



## **АННОТАЦИЯ**

Выпускная работа студентки Чугуновой А.В. группы А-02-16. Работа посвящена разработке математической модели для робота - манипулятора типа «трипод». В частности, были выполнены следующие шаги:

- Обзор и изучение конструкций механизмов с параллельной структурой;
- Построение геометрической модели манипулятора Festo;
- Решение прямой и обратной задач кинематики движения;
- Построение на основе кинематических уравнений связи зоны достижимости и анализ полученного результата;
- Решение прямой и обратной задач динамики движения и анализ полученных результатов;
- Разработка пользовательского приложения в среде разработки MATLAB 6 для простоты получения решения подобных задач.

Работа представлена на 91 страницах, 51 рисунке, и 6 приложениях.

## **SUMMARY**

Graduation work of student Chugunova A.V. Group A-02-16. The work is devoted to the development of a mathematical model for a robot manipulator of the "tripod" type. In particular, the following steps have been taken:

- Review and study of structures of mechanisms with parallel structure;
- Construction of geometric model of Festo manipulator;
- Solving forward and reverse motion kinematics problems;
- Construction of the reachability zone connection on the basis of kinematic equations and analysis of the obtained result;
- Solving forward and reverse motion dynamics problems and analyzing the results obtained;
- Develop a user application in the MATLAB 6 development environment for ease of solving similar problems.

The work is presented on 91 pages, 51 illustrations and 6 annexes.

# СОДЕРЖАНИЕ

|   |    |
|---|----|
| <b>АННОТАЦИЯ</b> .....  | 5  |
| <b>СОДЕРЖАНИЕ</b> .....   | 6  |
| <b>ВВЕДЕНИЕ</b> .....   | 7  |
| <b>1 ОБЗОР И ОПИСАНИЕ КОНСТРУКЦИИ МАНИПУЛЯТОРА С ПАРАЛЛЕЛЬНОЙ КИНЕМАТИКОЙ</b> .....                                   | 9  |
| 1.1. Обзор конструкций манипуляторов типа «трипод» .....  | 9  |
| 1.2. Описание конструкции «трипода» EXPT компании Festo .....   | 15 |
| <b>2 КИНЕМАТИКА РОБОТА С ПАРАЛЛЕЛЬНОЙ КИНЕМАТИКОЙ</b> .....   | 18 |
| 2.1 Анализ манипулятора и вывод описания кинематической модели .....  | 18 |
| 2.2 Поиск решения прямой и обратной задачи кинематики. ....   | 30 |
| 2.3 Краткие выводы по главе.....  | 39 |
| <b>3 ДИНАМИКА РОБОТА С ПАРАЛЛЕЛЬНОЙ КИНЕМАТИКОЙ</b> .....   | 41 |
| 3.1. Анализ манипулятора и вывод описания динамической модели .....   | 41 |
| 3.2. Поиск решения прямой и обратной задачи динамики. ....  | 48 |
| 3.3. Краткие выводы по главе.....   | 52 |
| <b>4 РАЗРАБОТКА ПРИЛОЖЕНИЯ ДЛЯ РЕШЕНИЯ ЗАДАЧ</b> .....  | 54 |
| 4.1 Описание интерфейса разработанной программы по решению прямой, обратной задач кинематики и зоны достижимости..... | 54 |
| 4.2 Описание интерфейса разработанной программы по решению прямой, обратной задач динамики. ....                      | 62 |
| 4.2.1 Алгоритм работы окна «Прямая задача динамики» приложения. ....  | 62 |
| 4.2.2 Алгоритм работы окна «Обратная задача динамики» приложения.....   | 66 |
| <b>5 ЗАКЛЮЧЕНИЕ</b> .....   | 71 |
| <b>6 СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ</b> .....   | 72 |
| <b>7 ПРИЛОЖЕНИЕ</b> .....   | 74 |
| <b>Приложение 1. Организация главной программы приложения</b> .....   | 74 |
| <b>Приложение 2. Организация решения задачи прямой кинематики</b> .....   | 75 |
| <b>Приложение 3. Организация решения задачи обратной кинематики</b> .....   | 75 |
| <b>Приложение 4. Организация решения получения зоны достижимости</b> .....  | 85 |
| <b>Приложение 5. Организация решения задачи прямой динамики</b> .....   | 90 |
| <b>Приложение 6. Организация решения задачи обратной динамики</b> .....   | 93 |

## **ВВЕДЕНИЕ**

В настоящий момент времени рост производительности промышленных компаний, а также повышение количества и качества выпускаемой продукции представляет собой важную задачу. Организация автоматизированных процессов при производстве может быть достигнуто введением вычислительной техники, а также роботов-манипуляторов.

Роботы-манипуляторы, чаще всего, обладают высокой стоимостью по сравнению с классическими механизмами, однако из-за гибкости конфигураций способны совершать широкий круг всевозможных задач в условиях производства. В частности те, какие ранее был способен выполнять лишь человек. Необходимо понимать то, что способности каждого манипулятора ограничены. Он действует в конкретной области рабочего пространства. Ограничения по положению, ориентации, скорости, а также ускорениям обладают своими пределами. С целью того, чтобы выяснить допустимые границы работоспособности манипулятора рационально осуществить математическое моделирование робота и его работы.

Одним из наиболее интересных видов применяемых манипуляторов на производстве являются механизмы с параллельной структурой, а именно с параллельной кинематикой.

Манипуляторы параллельной кинематики – это механизм, исполнительное звено которого крепится к основанию несколькими независимыми кинематическими цепями. Этим устройствам характерны надежность и высокая точность срабатывания, а также возможность выдерживать немаленькие нагрузки. Аналогичные системы используются во всевозможных сферах: начиная с астрономии и заканчивая сортировкой продукции на предприятии. Однако существует одна проблема в таких системах, которая связана с высокой математической сложностью, возникающей из-за нелинейности системы, а именно

возможностью в режиме реального времени принимать во внимание большое количество ограничительных критериев.

Частным случаем такого механизма является манипулятор типа «трипод». Он представляет собой параллельное кинематическое устройство, состоящее из основания, установленного на трех опорах, которые меняя свою длину, меняют положение и угол наклона. Кинематика и динамика таких роботов значительно отличается от кинематики и динамики обычных манипуляторов. Данное различие возникает из-за того, что любая степень подвижности параллельного механизма в отдельности является элементарной разомкнутой системой, при этом они соединены между собой и двигаются как единый механизм. Тут же и возникает сложность управления тягами такого робота, чтобы движение всех степеней подвижности было согласованным.

Поэтому целью текущей работы стала разработка математической модели манипулятора типа «трипод», а именно расчет кинематики и проверка работоспособности при краевых условиях, расчет динамики и построение соответствующих характеристик для анализа работы «трипода», а так же для упрощения данного процесса разработка приложения с графическим интерфейсом.



# 1 ОБЗОР И ОПИСАНИЕ КОНСТРУКЦИИ МАНИПУЛЯТОРА С ПАРАЛЛЕЛЬНОЙ КИНЕМАТИКОЙ

## 1.1. Обзор конструкций манипуляторов типа «трипод»

Роботы-манипуляторы, используемые на сегодняшний день, имеют классическую схему, представляющую незамкнутую цепочку звеньев, поочередно соединённых с помощью всевозможных кинематических пар. Подобные конструкции обладают часто встречающимися недостатками - сравнительно небольшая степень грузоподъемности, значительные инерционные нагрузки, относительно повышенная металлоемкость.

Отыскание решений, которые позволяют сократить отмеченные выше недочеты привели к возникновению механизмов с параллельной структурой. Как, верно, замечено в [6], достоинства подобных манипуляторов состоят в наилучшей грузоподъемности, высокой жесткости конструкции, достижении высоких скоростей, а также ускорений грузозахватного устройства, существенного уровня унификации узлов. Помимо этого, конструкция данных манипуляторов совмещает функции металлоконструкции механизма, что значительно сокращает итоговую массу манипулятора.

Выходное звено подобных манипуляторов соединено с основанием несколькими кинематическими цепями, из которых любая оборудована приводом и налагает определенное число связей на перемещение выходного звена. В связи с этим, недостатками механизмов параллельной структуры являются ограничение зоны достижимости, сравнительно малая манипулятивность, а вдобавок затруднительность их управлением.

Базовым манипулятором параллельной структуры считается механизм Гью-Стюарта (рис. 1.1) [7], который состоит из двух платформ, шарнирно соединенных

шестью поступательными кинематическими парами («гексапод»). Последующее исследование данного типа конструкции выявило возможность получения существенного количества комбинаций стержней, связывающих две платформы, при этом характерные черты каждой структуры подразумевают ее применение с определенной технологической целью.



Рисунок 1.1 – Платформа Гью-Стюарта

Одним из первых промышленных роботов обладающим параллельной кинематикой принято считать робот Полларда [8], который представляет пятистепенной манипулятор с тремя параллельными кинематическими цепями, определяющие положение схвата, а также двумя избыточными степенями подвижности для управления схватом. В нашей стране первым подобным манипулятором является манипулятор Данилевского [9], обладающий шестью степенями подвижности.

Сегодняшние механизмы с параллельной кинематикой имеют всевозможные виды их исполнения, кроме того их функционал обуславливает их использование в различных отраслях промышленности, например, машиностроение, приборостроение, и одна из интересных отраслей космические исследования.

Нынешние проблемы машиностроения представляют собой создание эффективных многофункциональных механизмов, которые смогут обеспечить высокую точность, надежность и производительность.

Сейчас на мировом рынке робототехнического станочного оборудования представлены механизмы современной концепции, так называемые «гексаподы», основанные на стержневой конструкции, обладающие шестью степенями свободы или подобные им «триподы», единственное отличие — это наличие всего трех степеней свободы. Аналогичные механизмы обладают существенным достоинством по сравнению с классическими многокоординатными манипуляторами и трехкоординатными измерительными машинами. Роботы-манипуляторы с параллельной структурой обладают высокой жесткостью и точностью по отношению к манипуляторам классического исполнения.

Типичным представителем робототехнических станков типа «гексапод» является разработка компании OKUMA (рис. 1.2) [10]. Основан на шести механизмах поступательного перемещения, которые обеспечивают полноценный доступ схвата к детали для дальнейшей обработки. Есть возможность использовать в качестве схвата свыше 20 всевозможных инструментов. Изменение длины штанг производится с помощью регулируемых электроприводов. Величину перемещения контролирует датчик положения. Концы штанг шарнирно соединены соответственно с основанием и подвижной платформой, на которой установлен рабочий инструмент. Одной из интересных особенностей является то, что в процессе обработки деталь остается совершенно неподвижной, а схват перемещается вокруг нее. С помощью этого станка возможно подвергать обработке различные отверстия и поверхности под любыми углами.



Рисунок 1.2 – Станок OKUMA серия PM

Популярной разработкой манипулятора типа «трипод» является станок Tricept шведской фирмы Neos Robotics (рис. 1.3) [11]. Механизм представляет собой три симметрично установленных телескопических штанги, которые по средством шарниров связаны с неподвижным основанием и рабочим инструментом. У данного станка имеется три подвижные штанги, которые растягиваются – сжимаются с помощью двигателей, и одна центральная неподвижная штанга. Она обладает заметно большим размером относительно приводных штанг.



Рисунок 1.3 – Станок Tricept Neos Robotics

Сортировка, упаковка разнообразных изделий считается трудоемким процессом промышленности. Одним из основных требований к данному действию – это быстрота упаковки предметов в различные контейнеры, лотки. Помимо этого, важной составляющей являются требования безопасности при упаковке, например, пищевых изделий, откуда возникает необходимость в поиске способов с наименьшим контактом человека и изделия. По этим причинам значительно вырос интерес к автоматизации процесса и использования промышленных роботов.

Уникальный дельта-робот ABB «FlexPicker» (рис. 1.4) [12] является ярчайшим примером автоматического механизма, который специализируется на упаковке маленьких и легких изделий, например, конфеты. Манипулятор выполнен из прочных, а главное легких материалов, таких как углеродное волокно, пластик. Самая тяжелая часть – двигатели, которые находятся на неподвижной части конструкции. Способен обеспечить максимальную скорость до 10 м/с и максимальное ускорение до 150 м/с<sup>2</sup>, а точность позиционирования составляет  $\pm 0,1$  мм при максимальной нагрузке 3 кг.

Еще одним интересным представителем промышленной робототехники является дельта-робот компании FANUC серия M (рис. 1.5) [13]. Их называю так же «робот-паучок» за счет своих размеров и особенностей конструкции. Двигательные возможности данного манипулятора сравнимы с человеческой кистью и подходят для обширного спектра задач. При своих конструктивных особенностях обладает очень малым весом 26 кг при максимальной нагрузке 1 кг, точность позиционирования составляет  $\pm 0,03$  мм, а область достижимости 420 мм и тем самым занимает маленькую площадь.



Рисунок 1.4 Робот-манипулятор  
FlexPicker ABB



Рисунок 1.5 Дельта-робот FANUC M

Одним из наиболее оснащенных манипуляторов с параллельной кинематикой является разработка немецкой компании Festo (рис. 1.6) [14]. Конструкция робота произведена по схеме «трипод», в которой применяются линейные приводы для перемещения захватного устройства. Представленный механизм имеет сравнительно маленький вес до 66 кг, при этом может обеспечить максимальную скорость до 7 м/с, а ускорение до  $110 \text{ м/с}^2$  при максимальной нагрузке 5 кг с точностью позиционирования  $\pm 0,5 \text{ мм}$



Рисунок 1.6 Робот-манипулятор Festo

Выполненный анализ существующих манипуляторов параллельной структуры показал прекрасные функциональные качества и возможность выполнять довольно внушительный объем разнообразных производственных операций. Заинтересованность в создании робототехнических систем обусловлена, в первую очередь, высокими динамическими характеристиками, модульным построением. А из-за особенностей конструкции манипуляторов предполагается их работа в составе стационарного производственного оборудования.

## **1.2. Описание конструкции «трипода» EXPT компании Festo**

Построение математической модели производится на основе конструкции манипулятора типа «трипод» EXPT от компании Festo (рис. 1.7) [14], поэтому рассмотрим ее подробнее и произведем описание основных элементов.

Промышленный манипулятор является одной из разновидностей дельта – роботов, в данном случае кинематический механизм имеет три пары параллельных тяг (рис. 1.7-②), оси которых расположены не параллельно друг другу, тем самым не образуют декартово пространство координат. Рассматриваемый трипод обладает тремя степенями свободы пространственного перемещения и одной степенью свободы для ориентации по углу. Высокую динамику около 150

циклов/мин обеспечивает тяги, выполненные из полимера из углеродного волокна, что делает их легкими и приводит к минимальной подвижной массе.

Жесткость конструкции создается за счет закрепленных между собой линейных приводов (рис. 1.7-④), которые составляют пирамидальную форму. Правильно выбранные, мощные двигатели (рис. 1.7-⑦) помогают осуществить высокоскоростное и точное перемещение в трехмерном пространстве. Неподвижное треугольное основание (рис. 1.7-⑥) обеспечивает устойчивость всей конструкции, наилучшее распределение усилий и уменьшение вибраций. Подвижная площадка (рис. 1.7-①) является элементарным интерфейсом с возможностью присоединения различных видов вакуумных и механических схватов.

Данный манипулятор обладает целиком конфигурируемым программным управлением, приспособленное для различных применений. Управление всей системой осуществляет стандартизированный шкаф СМСА, который позволяет максимально просто ввести «трипод» в эксплуатацию.

Шкаф управления состоит из многокоординатного контроллера для осуществления перемещений и контроллеров, управляющих работой двигателей, также предусмотрены автоматические аварийные остановки для наибольшей безопасности работы.



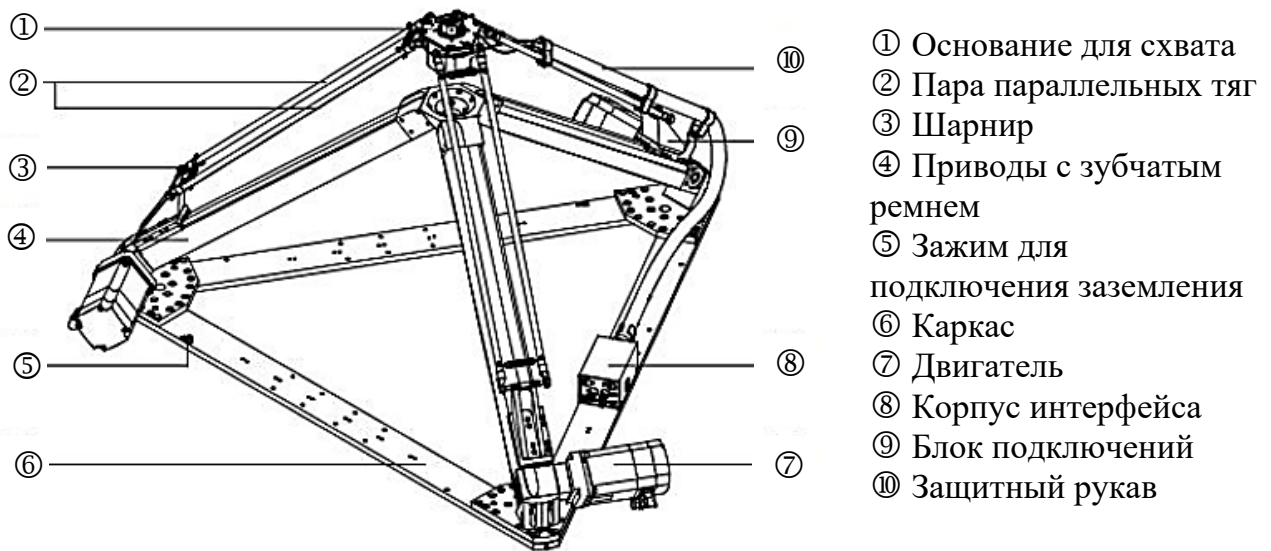


Рисунок 1.7 Конструкция манипулятора типа «трипод» Festo

Мозгом всей системы управления является контроллер CMXR-C2, с помощью которого легко можно запрограммировать траектории движения схвата.

Так же в систему входит компактная система технического зрения SBOC-Q-R2B. С ее помощью можно наблюдать за перемещением объектов, анализировать их свойства и особенности, что позволяет с высокой точностью находить ненадлежащего качества объекты и определять их пространственные координаты.

Далеко небесполезным является пульт оператора CDSA для ручного управления с возможностью онлайн программирования.

Сферы использования как отмечено в [5], рассмотренного дельта-робота могут быть весьма обширными. Самыми стандартными примерами считается задача вертикальной паковки различных продуктов в коробку, трей или коррекс. Более интересными примерами использования в промышленных масштабах являются нанесение клея по контуру детали, итоговая стадия упаковки напитков, установка различных крышек.

## 2 КИНЕМАТИКА РОБОТА С ПАРАЛЛЕЛЬНОЙ СТРУКТУРОЙ

### 2.1 Анализ манипулятора и вывод описания кинематической модели

Математическое описание рассматриваемого манипулятора невозможно без его геометрической схемы, которая представлена на рисунке 2.1.

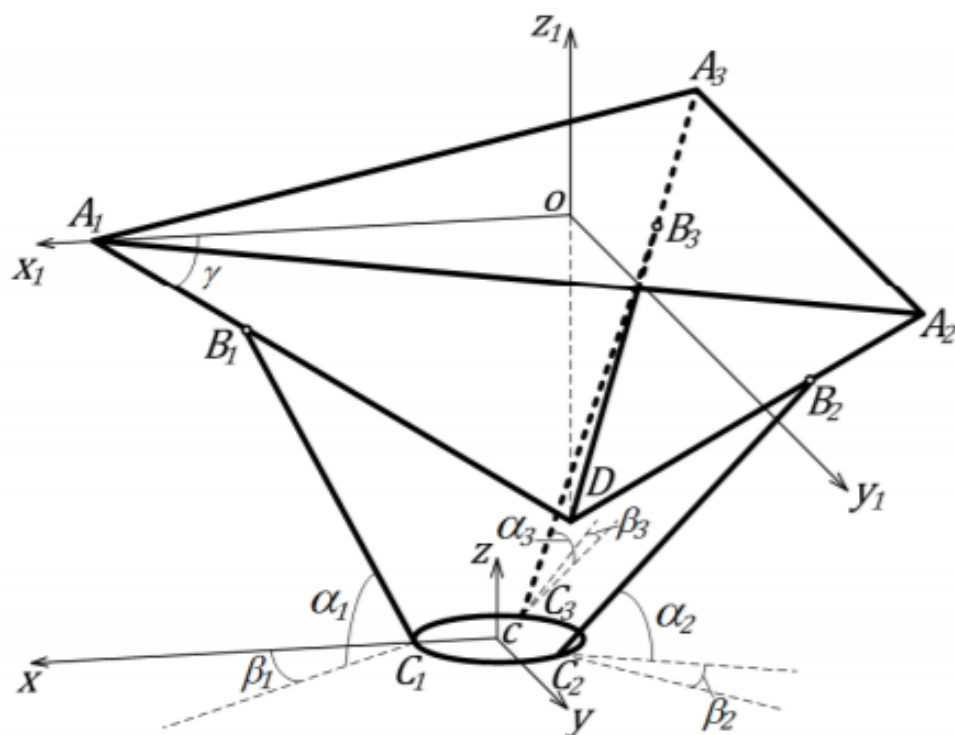


Рисунок 2.1 – Расчетная схема манипулятора типа «трипод»

За счет конструктивных особенностей манипулятора, а точнее его тяги, соединяющие каретки с подвижной платформой расположены парами и параллельны относительно друг друга. Поэтому плоскость подвижной платформы всегда параллельна плоскости неподвижного основания.

Из-за того, что пары параллельных тяг соединяют между собой подвижное и неподвижное основания, плоскости, которые их описывают, всегда остаются параллельны друг другу.

Тогда для простоты проведения последующих расчетов рассмотрим части построенной геометрической модели:

$A_1A_3 = A_3A_2 = A_1A_2 = L$  – длины сторон неподвижного основания, которое представляет собой равносторонний треугольник;

$B_1C_1 = B_2C_2 = B_3C_3 = l$  – длина тяги;

$A_1B_1 = S_1, A_2B_2 = S_2, A_3B_3 = S_3$  – расстояния, определяющие перемещение кареток по направляющим;

$\gamma$  – угол между неподвижным основанием и приводом;

$\alpha_i, \beta_i$  – вспомогательные расчетные углы, где  $i = \overline{1,3}$ ;

$CC_1 = CC_2 = CC_3 = a$  – радиус подвижного основания;

$Ox_1y_1z_1, Cx_uz$  – координатные оси неподвижного и подвижного основания соответственно;

$C_1C_2C_3 \parallel A_1A_2A_3$  – всегда, где  $C_1C_2C_3$ -описывает плоскость подвижной платформы, а  $A_1A_2A_3$ -плоскость неподвижного основания.

Установив связь между координатами кареток и подвижной платформой, сможем отыскать решение задачи кинематики [1]. Введем две системы координат  $Ox_1y_1z_1$ -для неподвижного основания,  $Cx_uz$ -для подвижного основания, что следует из рисунка 2.1.

Вектора, описывающие неподвижное основание  $A_1A_2A_3$ , являются постоянными величинами  $\overline{A_1A_2} = const, \overline{A_2A_3} = const, \overline{A_1A_3} = const$ , при условии рассмотрения их относительно системы координат  $S_{xyz}$ . Тогда можем записать соотношения связывающие координаты данных векторов, для дальнейшего решения достаточно двух:

$$x_{A_2} - x_{A_1} = const, x_{A_3} - x_{A_1} = const, \quad (2.1)$$

$$y_{A_2} - y_{A_1} = const, y_{A_3} - y_{A_1} = const. \quad (2.2)$$

$Ox_1y_1z_1$  и  $S_{xyz}$  параллельные системы координат, т.к. основания всегда остаются параллельны друг другу. А значит, координаты  $z_{A_i}$  для неподвижной платформы относительно  $S_{xyz}$  одинаковые. Тогда получаем верные равенства:

$$z_{A_1} = z_{A_2}, z_{A_2} = z_{A_3}. \quad (2.3)$$

Известно, что координаты точки  $O$  относительно системы координат  $S_{xyz}$  можно вычислить с помощью следующих соотношений:

$$\begin{aligned} x_O &= \frac{x_{A_1} + x_{A_2} + x_{A_3}}{3}, \\ y_O &= \frac{y_{A_1} + y_{A_2} + y_{A_3}}{3}, \\ z_O &= \frac{z_{A_1} + z_{A_2} + z_{A_3}}{3}. \end{aligned} \quad (2.4)$$

Тогда координаты точки  $C$  относительно системы координат  $Ox_1y_1z_1$  могут быть описаны следующим образом:

$$\begin{aligned} x_C &= -\frac{x_{A_1} + x_{A_2} + x_{A_3}}{3}, \\ y_C &= -\frac{y_{A_1} + y_{A_2} + y_{A_3}}{3}, \\ z_C &= -\frac{z_{A_1} + z_{A_2} + z_{A_3}}{3}. \end{aligned} \quad (2.5)$$

Соединим полученные ранее соотношения (2.1 – 2.3, 2.5) и сведем все в систему из 9-ти уравнений связи (2.6).

$$\begin{cases} z_{A_1} = z_{A_2}, \\ z_{A_2} = z_{A_3}, \\ x_{A_2} - x_{A_1} = const, \\ x_{A_3} - x_{A_1} = const, \\ y_{A_2} - y_{A_1} = const, \\ y_{A_3} - y_{A_1} = const, \\ x_C = -\frac{x_{A_1} + x_{A_2} + x_{A_3}}{3}, \\ y_C = -\frac{y_{A_1} + y_{A_2} + y_{A_3}}{3}, \\ z_C = -\frac{z_{A_1} + z_{A_2} + z_{A_3}}{3}. \end{cases} \quad (2.6)$$

В полученной системе неизвестными являются координаты вершин  $A_1, A_2, A_3$ , выразив их сможем найти решение системы (2.6).

Поиск выражений для каждой из координат будем делать геометрическим способом. Для этого изобразим геометрические модели всех вершин неподвижного основания последовательно.

Геометрическая интерпретация вершины  $A_1$  представлена на рисунке 2.3.

Координаты будем вычислять посредством сложения длин проекций, которые получают, идя последовательно от точки  $C$  до точки  $A_1$  по звеньям.

Из рисунка 2.3 видно, что:

$$x_{A_1} = |A_1''B_1''| + |B_1''C_1| + a, \quad (2.7)$$

$$y_{A_1} = |A_1'''C|, \quad (2.8)$$

$$z_{A_1} = |KL| + |LC|. \quad (2.9)$$

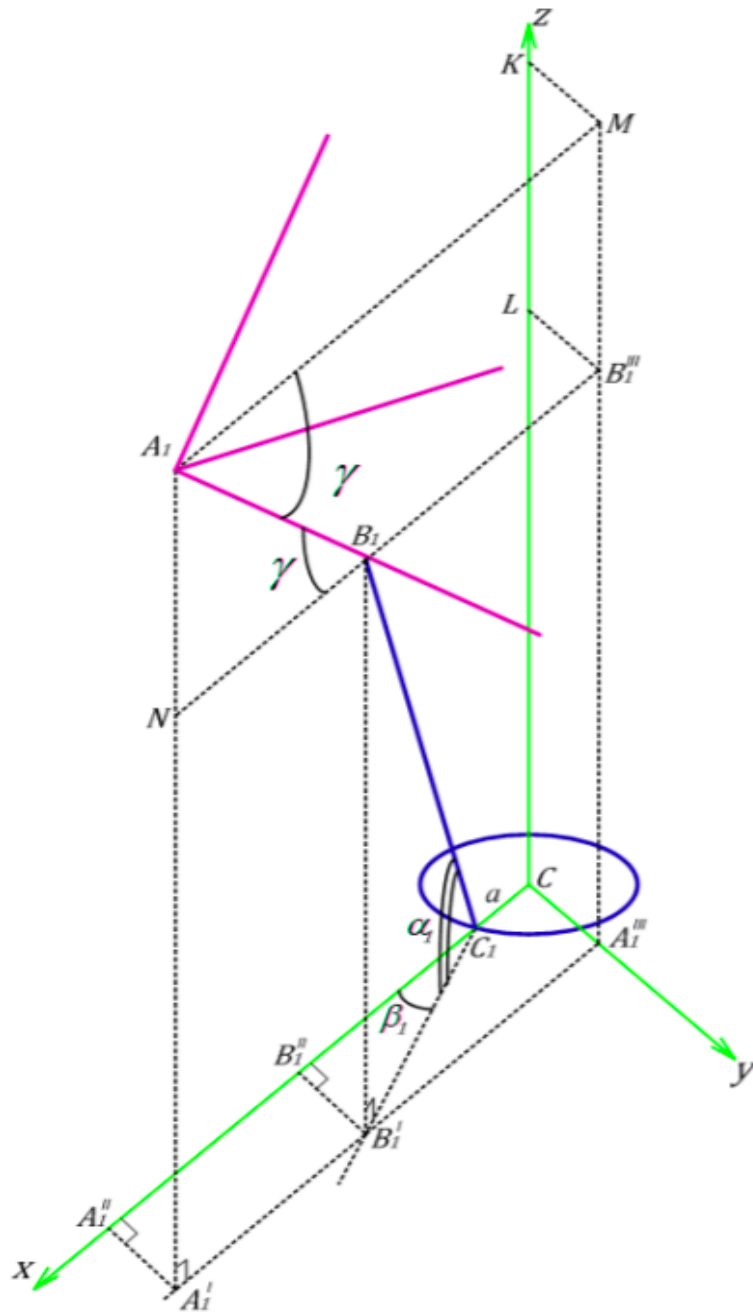


Рисунок 2.3 – Геометрическое отображение вершины  $A_1$

Получим выражение для каждой проекции через известные величины:

$$|A_1''B_1''| = |A_1'B_1'| = |A_1B_1| \cos \gamma = S_1 \cos \gamma, \quad (2.10)$$

$$|B_1''C_1| = |B_1'C_1| \cos \beta_1 = |B_1C_1| \cos \beta_1 \cos \alpha_1 = l \cos \beta_1 \cos \alpha_1, \quad (2.11)$$

$$|A_1'''C_1| = |B_1''B_1'| = |B_1'C_1| \sin \beta_1 = l \cos \alpha_1 \sin \beta_1, \quad (2.12)$$

$$|KL| = |MB_1''| = |M'B_1| = |A_1B_1| \sin \gamma = S_1 \sin \gamma, \quad (2.13)$$

$$|LC| = |A_1'''B_1'''| = |B_1'B_1'| = |B_1C_1| \sin \alpha_1 = l \sin \alpha_1, \quad (2.14)$$

где  $\alpha_1$  – угол между тягой и проекцией тяги на плоскость  $Sxу$ ,  $\beta_1$  – угол между проекциями тяги и направляющей  $A_1D$  на плоскость  $Sxу$ .

Для записи итоговых выражений для геометрического представления вершины  $A_1$ , подставим полученные соотношения (2.10 – 2.14) в (2.7 – 2.9):

$$x_{A_1} = S_1 \cos \gamma + l \cos \beta_1 \cos \alpha_1 + a, \quad (2.15)$$

$$y_{A_1} = l \cos \alpha_1 \sin \beta_1, \quad (2.16)$$

$$z_{A_1} = S_1 \sin \gamma + l \sin \alpha_1. \quad (2.17)$$

Геометрическая интерпретация вершины  $A_2$  представлена на рисунке 2.4.

Вычислять координаты будем за счет сложения длин проекций, которые получаются, идя последовательно от точки  $C$  до точки  $A_2$  по звеньям.

Из рисунка 2.4 видно, что:

$$x_{A_2} = -|A_2''B_2''| - |B_2''C_2''| - |CC_2''|, \quad (2.18)$$

$$y_{A_2} = |A_2'''B_2'''| + |B_2'''C_2'''| + |CC_2'''|, \quad (2.19)$$

$$z_{A_2} = |TE| + |EC|. \quad (2.20)$$

Получим выражение для каждой проекции через известные величины:

$$|A_2''B_2''| = |B_2'W| = |A_2'B_2'| \sin \frac{\pi}{6} = |B_2R| \sin \frac{\pi}{6} = S_2 \cos \gamma \sin \frac{\pi}{6}, \quad (2.21)$$

$$|B_2''C_2''| = |C_2C_2'| = |C_2B_2'| \sin(\beta_2 + \frac{\pi}{6}) = l \sin(\beta_2 + \frac{\pi}{6}) \cos \alpha_2, \quad (2.22)$$

$$|C_2C_2''| = |CC_2| \sin \frac{\pi}{6} = a \sin \frac{\pi}{6}, \quad (2.23)$$

$$|A_2'''B_2'''| = |B_2'P| = |A_2'W| = |A_2'B_2'| \cos \frac{\pi}{6} = S_2 \cos \frac{\pi}{6} \cos \gamma, \quad (2.24)$$

$$|C_2'''B_2'''| = |C_2'B_2'| = |C_2B_2'| \cos(\beta_2 + \frac{\pi}{6}) = l \cos \alpha_2 \cos(\beta_2 + \frac{\pi}{6}), \quad (2.25)$$

$$|C_2C_2'''| = |CC_2| \cos \frac{\pi}{6} = a \cos \frac{\pi}{6}, \quad (2.26)$$

$$|TE| = |T'E'| = |A_2B_2| \sin \gamma = S_2 \sin \gamma, \quad (2.27)$$

$$|EC| = |E'C'| = |B_2'B_2| = |B_2C_2| \sin \alpha_2 = l \sin \alpha_2, \quad (2.28)$$

где  $\alpha_2$  – угол между тягой и проекцией тяги на плоскость  $S_{xy}$ ,  $\beta_2$  – угол между проекциями тяги и направляющей  $A_2D$  на плоскость  $S_{xy}$ .

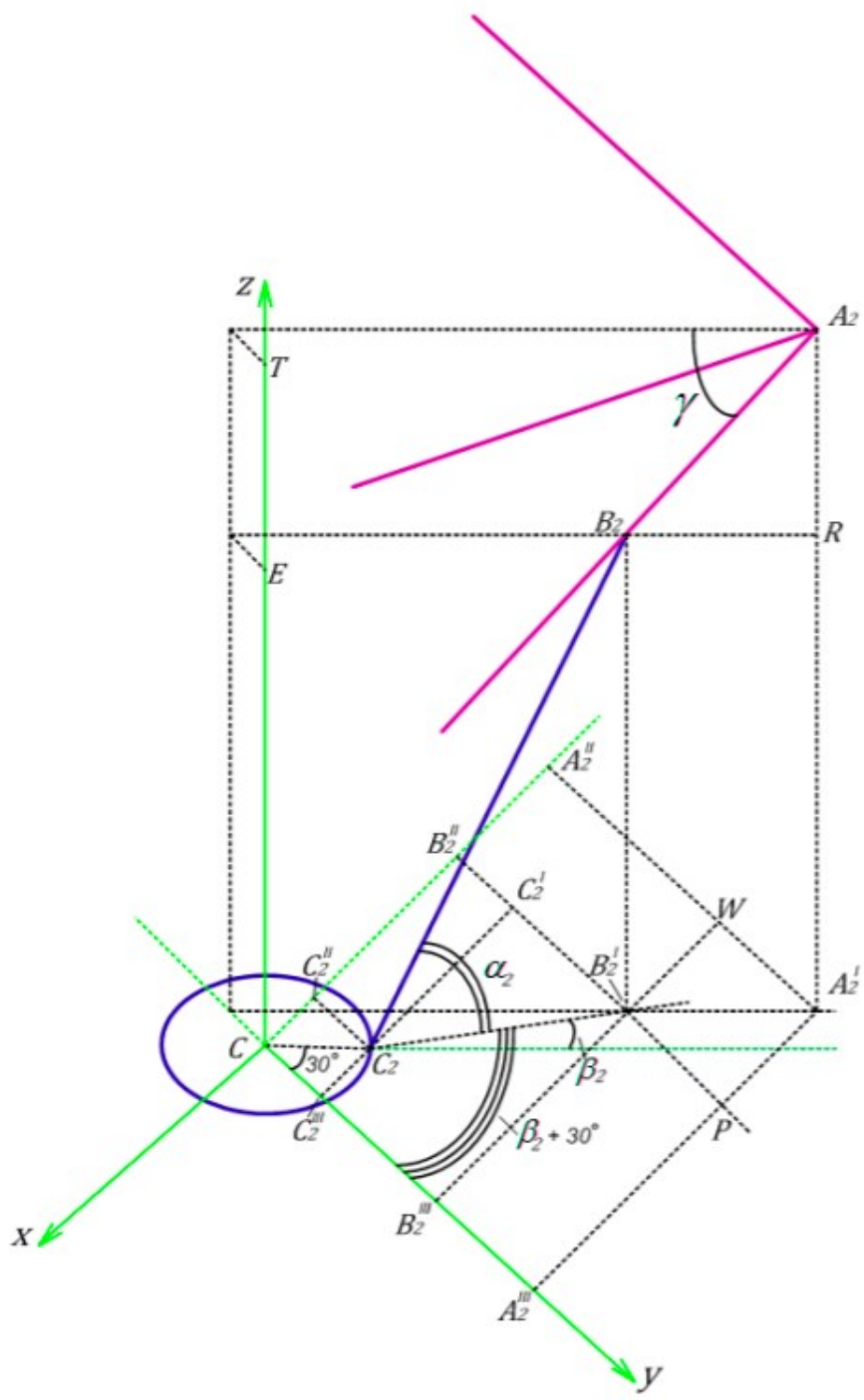


Рисунок 2.4 – Геометрическое отображение вершины  $A_2$

Для записи итоговых выражений для геометрического представления вершины  $A_2$ , подставим полученные соотношения (2.21 – 2.28) в (2.18 – 2.20):



$$x_{A_2} = -S_2 \cos \gamma \sin \frac{\pi}{6} - l \sin \left( \beta_2 + \frac{\pi}{6} \right) \cos \alpha_2 - a \sin \frac{\pi}{6}, \quad (2.29)$$

$$y_{A_2} = S_2 \cos \frac{\pi}{6} \cos \gamma + l \cos \alpha_2 \cos \left( \beta_2 + \frac{\pi}{6} \right) + a \cos \frac{\pi}{6}, \quad (2.30)$$

$$z_{A_2} = S_2 \sin \gamma + l \sin \alpha_2. \quad (2.31)$$

Геометрическая интерпретация вершины  $A_3$  представлена на рисунке 2.5.

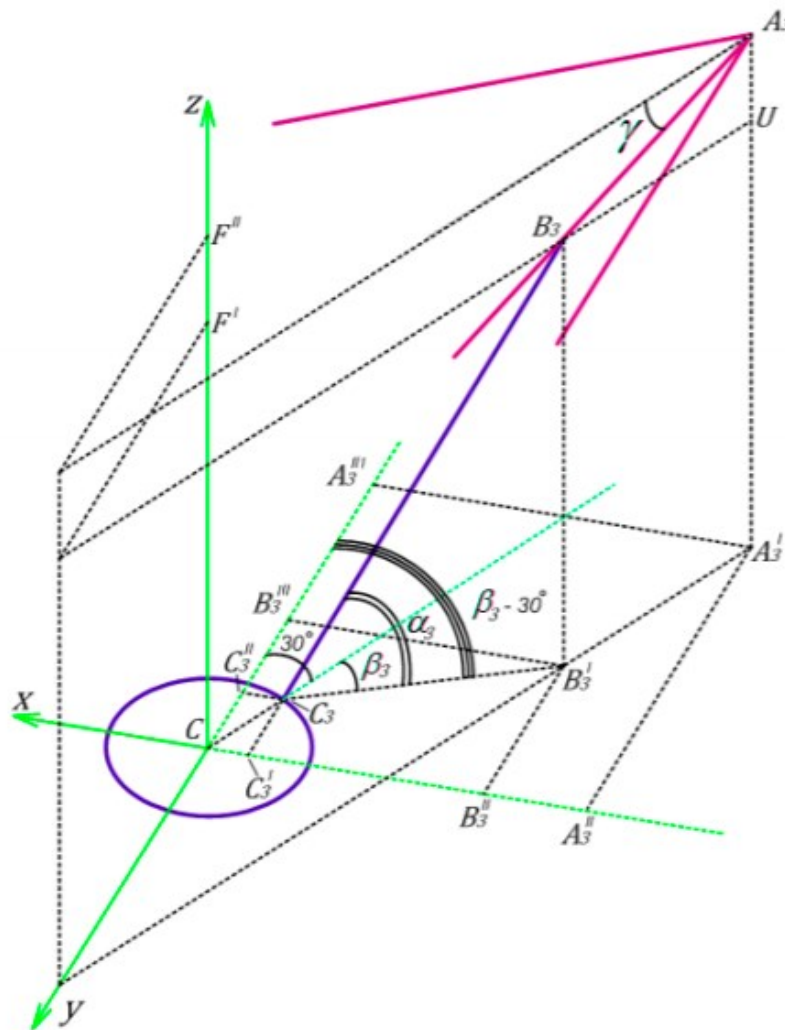


Рисунок 2.5 – Геометрическое отображение вершины  $A_3$

Вычислять координаты будем за счет сложения длин проекций, которые получаются, идя последовательно от точки  $C$  до точки  $A_3$  по звеньям.

Из рисунка 2.5 видно, что:

$$x_{A_3} = -|A_3''B_3''| - |B_3''C_3'| - |CC_3'|, \quad (2.32)$$

$$y_{A_3} = -|A_3'''C_3''| - |B_3'''C_3''| - |CB_3'''|, \quad (2.33)$$

$$z_{A_3} = |F'F''| + |F'C|. \quad (2.34)$$

Получим выражение для каждой проекции через известные величины:

$$|A_3''B_3''| = |B_3'P| = |A_3'B_3'| \sin \frac{\pi}{6} = |B_3U| \sin \frac{\pi}{6} = S_3 \cos \gamma \sin \frac{\pi}{6}, \quad (2.35)$$

$$|B_3''C_3'| = |C_3B_3'| \sin(\beta_3 - \frac{\pi}{6}) = l \sin(\beta_3 - \frac{\pi}{6}) \cos \alpha_3, \quad (2.36)$$

$$|CC_3'| = |CC_3| \sin \frac{\pi}{6} = a \sin \frac{\pi}{6}, \quad (2.37)$$

$$|A_3'''B_3'''| = |A_3'B_3'| \cos \frac{\pi}{6} = |B_3U| \cos \frac{\pi}{6} = S_3 \cos \frac{\pi}{6} \cos \gamma, \quad (2.38)$$

$$|C_3''B_3'''| = |C_3B_3'| \cos(\beta_3 - \frac{\pi}{6}) = l \cos \alpha_3 \cos(\beta_3 - \frac{\pi}{6}), \quad (2.39)$$

$$|C_3C_3''| = |CC_3| \cos \frac{\pi}{6} = a \cos \frac{\pi}{6}, \quad (2.40)$$

$$|F'F''| = |E''E'| = |A_3B_3| \sin \gamma = S_3 \sin \gamma, \quad (2.41)$$

$$|F'C| = |E'E| = |B_3'B_3| = |B_3C_3| \sin \alpha_3 = l \sin \alpha_3, \quad (2.42)$$

где  $\alpha_3$  – угол между тягой и проекцией тяги на плоскость  $Sx_3y_3$ ,  $\beta_3$  – угол между проекциями тяги и пневмоцилиндра  $A_3D$  на плоскость  $Sx_3y_3$ .

Для записи итоговых выражений для геометрического представления вершины  $A_3$ , подставим полученные соотношения (2.35 – 2.42) в (2.32 – 2.34):

$$x_{A_3} = -S_3 \cos \gamma \sin \frac{\pi}{6} - l \sin(\beta_3 - \frac{\pi}{6}) \cos \alpha_3 - a \sin \frac{\pi}{6}, \quad (2.43)$$

$$y_{A_3} = -S_3 \cos \frac{\pi}{6} \cos \gamma - l \cos \alpha_3 \cos(\beta_3 - \frac{\pi}{6}) - a \cos \frac{\pi}{6}, \quad (2.44)$$

$$z_{A_3} = S_3 \sin \gamma + l \sin \alpha_3. \quad (2.45)$$

Теперь, когда найдены все неизвестные ранее координаты вершин неподвижного основания  $A_1A_2A_3$ , можем записать итоговые выражения для координат центра схвата, т.е. координаты точки  $C$ . Для этого подставим

соотношения (2.15 – 2.17), (2.29 – 2.31), (2.43 – 2.45) в полученное выше уравнение (2.5):

$$x_C = -\frac{1}{3} \left[ S_1 \cos \gamma + l \cos \beta_1 \cos \alpha_1 + a - S_2 \cos \gamma \sin \frac{\pi}{6} - l \sin \left( \beta_2 + \frac{\pi}{6} \right) \cos \alpha_2 - a \sin \frac{\pi}{6} - S_3 \cos \gamma \sin \frac{\pi}{6} - l \sin \left( \beta_3 - \frac{\pi}{6} \right) \cos \alpha_3 - a \sin \frac{\pi}{6} \right], \quad (2.46)$$

$$y_C = -\frac{1}{3} \left[ l \cos \alpha_1 \sin \beta_1 + S_2 \cos \frac{\pi}{6} \cos \gamma + l \cos \alpha_2 \cos \left( \beta_2 + \frac{\pi}{6} \right) + a \cos \frac{\pi}{6} - S_3 \cos \frac{\pi}{6} \cos \gamma - l \cos \alpha_3 \cos \left( \beta_3 - \frac{\pi}{6} \right) - a \cos \frac{\pi}{6} \right], \quad (2.47)$$

$$z_C = -\frac{1}{3} [S_1 \sin \gamma + l \sin \alpha_1 + S_2 \sin \gamma + l \sin \alpha_2 + S_3 \sin \gamma + l \sin \alpha_3]. \quad (2.48)$$

Вернемся к исходной системе (2.6), которая описывает полностью конструкцию нашего «трипода» и подставим в нее все найденные соотношения, после чего она будет выглядеть следующим образом:

$$\left\{ \begin{array}{l} S_1 \sin \gamma + l \sin \alpha_1 = S_2 \sin \gamma + l \sin \alpha_2, \\ S_2 \sin \gamma + l \sin \alpha_2 = S_3 \sin \gamma + l \sin \alpha_3, \\ -S_2 \cos \gamma \sin \frac{\pi}{6} - l \cos \alpha_2 \sin \left( \beta_2 + \frac{\pi}{6} \right) - a \sin \frac{\pi}{6} \\ -(S_1 \cos \gamma + l \cos \beta_1 \cos \alpha_1 + a) = const, \\ -S_3 \cos \gamma \sin \frac{\pi}{6} - l \sin \left( \beta_3 - \frac{\pi}{6} \right) \cos \alpha_3 - a \sin \frac{\pi}{6} - \\ -(S_1 \cos \gamma + l \cos \beta_1 \cos \alpha_1 + a) = const, \\ S_2 \cos \frac{\pi}{6} \cos \gamma + l \cos \alpha_2 \cos \left( \beta_2 + \frac{\pi}{6} \right) + a \cos \frac{\pi}{6} - l \cos \alpha_1 \sin \beta_1 = const, \\ -S_3 \cos \frac{\pi}{6} \cos \gamma - l \cos \alpha_3 \cos \left( \beta_3 - \frac{\pi}{6} \right) - a \cos \frac{\pi}{6} - l \cos \alpha_1 \sin \beta_1 = const, \\ x_C = -\frac{1}{3} [S_1 \cos \gamma + l \cos \beta_1 \cos \alpha_1 + a - S_2 \cos \gamma \sin \frac{\pi}{6} - l \cos \alpha_2 \sin \left( \beta_2 + \frac{\pi}{6} \right) - \\ - a \sin \frac{\pi}{6} - S_3 \cos \gamma \sin \frac{\pi}{6} - l \sin \left( \beta_3 - \frac{\pi}{6} \right) \cos \alpha_3 - a \sin \frac{\pi}{6}], \\ y_C = -\frac{1}{3} \left[ l \cos \alpha_1 \sin \beta_1 + S_2 \cos \frac{\pi}{6} \cos \gamma + l \cos \alpha_2 \cos \left( \beta_2 + \frac{\pi}{6} \right) + a \cos \frac{\pi}{6} - \right. \\ \left. - S_3 \cos \frac{\pi}{6} \cos \gamma - l \cos \alpha_3 \cos \left( \beta_3 - \frac{\pi}{6} \right) - a \cos \frac{\pi}{6} \right], \\ z_C = -\frac{1}{3} [S_1 \sin \gamma + l \sin \alpha_1 + S_2 \sin \gamma + l \sin \alpha_2 + S_3 \sin \gamma + l \sin \alpha_3]. \end{array} \right. \quad (2.49)$$

В полученной системе (2.49) остались неизвестные константы, которые легко можно найти, изобразив геометрически неподвижную платформу  $A_1A_2A_3$  «трипода» (рис. 2.6).

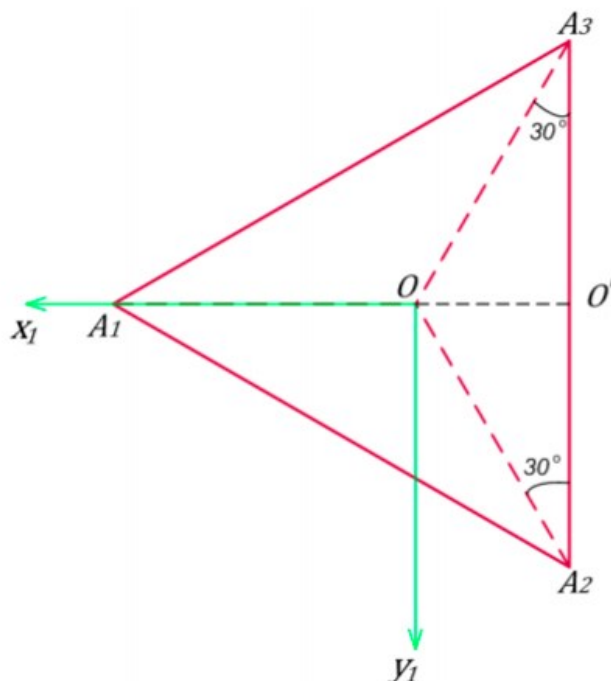


Рисунок 2.6 – Геометрическое представление основания  $A_1A_2A_3$

Вокруг основания можно описать окружность радиуса  $R$ , тем самым сможем найти неизвестные константы. Спроецируем каждый из радиусов, которые соединяют центр описанной окружности и вершины равностороннего треугольника  $A_1A_2A_3$  на координатную ось  $Ox_1y_1$ . Судя по рисунку 2.6, можем записать следующие соотношения:

$$\begin{aligned} x_{A_1} &= |A_1O|, \\ x_{A_2} &= x_{A_3} = -|OO'|, \\ y_{A_1} &= 0, \\ y_{A_2} &= |A_2O'|, \\ y_{A_3} &= -|A_3O'|. \end{aligned}$$

Получим:

$$x_{A_2} - x_{A_1} = x_{A_3} - x_{A_1} = -|OO'| - |A_1O| = -|A_1O'| = -L \cos \frac{\pi}{6}, \quad (2.50)$$

$$y_{A_2} - y_{A_1} = |A_2O'| = L \sin \frac{\pi}{6}, \quad (2.51)$$

$$y_{A_3} - y_{A_1} = -|A_3O'| = -L \sin \frac{\pi}{6}. \quad (2.52)$$

Найденные выражения для констант (2.50 – 2.52) подставим в систему уравнений (2.49), тем самым получим полное геометрическое описание конструкции манипулятора.

Система 2.53:

$$\left\{ \begin{array}{l} S_1 \sin \gamma + l \sin \alpha_1 = S_2 \sin \gamma + l \sin \alpha_2, \\ S_2 \sin \gamma + l \sin \alpha_2 = S_3 \sin \gamma + l \sin \alpha_3, \\ -S_2 \cos \gamma \sin \frac{\pi}{6} - l \cos \alpha_2 \sin (\beta_2 + \frac{\pi}{6}) - a \sin \frac{\pi}{6} - \\ \quad -(S_1 \cos \gamma + l \cos \beta_1 \cos \alpha_1 + a) = -L \cos \frac{\pi}{6}, \\ -S_3 \cos \gamma \sin \frac{\pi}{6} - l \sin (\beta_3 - \frac{\pi}{6}) \cos \alpha_3 - a \sin \frac{\pi}{6} - \\ \quad -(S_1 \cos \gamma + l \cos \beta_1 \cos \alpha_1 + a) = -L \cos \frac{\pi}{6}, \\ S_2 \cos \frac{\pi}{6} \cos \gamma + l \cos \alpha_2 \cos (\beta_2 + \frac{\pi}{6}) + a \cos \frac{\pi}{6} - l \cos \alpha_1 \sin \beta_1 = L \sin \frac{\pi}{6}, \\ -S_3 \cos \frac{\pi}{6} \cos \gamma - l \cos \alpha_3 \cos (\beta_3 - \frac{\pi}{6}) - a \cos \frac{\pi}{6} - l \cos \alpha_1 \sin \beta_1 = -L \sin \frac{\pi}{6}, \\ x_C = -\frac{1}{3} [S_1 \cos \gamma + l \cos \beta_1 \cos \alpha_1 + a - S_2 \cos \gamma \sin \frac{\pi}{6} - l \cos \alpha_2 \sin (\beta_2 + \frac{\pi}{6}) - \\ \quad - a \sin \frac{\pi}{6} - S_3 \cos \gamma \sin \frac{\pi}{6} - l \sin (\beta_3 - \frac{\pi}{6}) \cos \alpha_3 - a \sin \frac{\pi}{6}], \\ y_C = -\frac{1}{3} [l \cos \alpha_1 \sin \beta_1 + S_2 \cos \frac{\pi}{6} \cos \gamma + l \cos \alpha_2 \cos (\beta_2 + \frac{\pi}{6}) + a \cos \frac{\pi}{6} - \\ \quad - S_3 \cos \frac{\pi}{6} \cos \gamma - l \cos \alpha_3 \cos (\beta_3 - \frac{\pi}{6}) - a \cos \frac{\pi}{6}], \\ z_C = -\frac{1}{3} [S_1 \sin \gamma + l \sin \alpha_1 + S_2 \sin \gamma + l \sin \alpha_2 + S_3 \sin \gamma + l \sin \alpha_3]. \end{array} \right.$$

В итоге получили систему, которая состоит из 9-ти уравнений и 12-ти неизвестных, а точнее  $\alpha_1, \alpha_2, \alpha_3, \beta_1, \beta_2, \beta_3, S_1, S_2, S_3, x_C, y_C, z_C$  и является нелинейной. Если задаться численными значениями  $S_1, S_2, S_3$ , то сможем найти решение прямой задачи кинематики. Если же задаться численными значениями  $x_C, y_C, z_C$ , найдем решение обратной задачи кинематики «трипода».

## 2.2 Поиск решения прямой и обратной задачи кинематики

Как правило решение прямой задачи кинематики единственное и его поиск не представляет больших вычислительных трудностей. Интереснее дела обстоят с поиском решения обратной задачи кинематики, где по известным координатам положения захвата необходимо вычислить положение кареток привода. Для единственного положения захвата может существовать бесконечно много начальных положений звеньев. Для поиска решения наших задач воспользуемся итерационным методом Ньютона [15].

Метод Ньютона позволяет найти решения нелинейной системы уравнений (ситуация, когда количество неизвестных больше количества уравнений) с заданной точностью. Он (метод Ньютона) прост в применение и для дальнейшего программирования. Такой подход позволяет добиться весьма плавных движений характеристической рабочей точки, не допуская разрывов траектории движения.

В общем виде система нелинейных уравнений выглядит так:

$$\begin{cases} f_1(x_1, x_2, \dots, x_n) = 0, \\ f_2(x_1, x_2, \dots, x_n) = 0, \\ \dots \\ f_n(x_1, x_2, \dots, x_n) = 0. \end{cases} \quad (2.54)$$

Перепишем ее в векторной форме:

$$f(x) = 0, \quad (2.55)$$

$$\text{где } f = \begin{pmatrix} f_1 \\ f_2 \\ \dots \\ f_n \end{pmatrix}, x = \begin{pmatrix} x_1 \\ x_2 \\ \dots \\ x_n \end{pmatrix}. \quad (2.56)$$

Воспользуемся одним из существующих итерационным методом, чтобы найти решение системы (2.54).

Пусть, известна  $k$ -ая точка приближения  $x^{(k)} = (x_1^{(k)}, x_2^{(k)}, \dots, x_n^{(k)})$  к одному из изолированных корней  $x = (x_1, x_2, \dots, x_n)$  векторного уравнения (2.55). Тогда точный корень уравнения (2.54) можно записать следующим образом:

$$x = x^{(k)} + \Delta x^{(k)}, \quad (2.57)$$

где  $\Delta x^{(k)} = (\Delta x_1^{(k)}, \Delta x_2^{(k)}, \dots, \Delta x_n^{(k)})$  – погрешность приближенного корня.

При подстановке соотношения (2.57) в векторное уравнение (2.55), получим:

$$f(x^{(k)} + \Delta x^{(k)}) = 0. \quad (2.58)$$

Пусть функция  $f(x)$  непрерывно дифференцируема в некоторой выпуклой области, которая содержит  $x$  и  $x^{(k)}$ , разложим левую часть уравнения (2.58) по степеням малого вектора  $\Delta x^{(k)}$ , ограничившись только линейными элементами:

$$f(x^{(k)} + \Delta x^{(k)}) = f(x^{(k)}) + f'(x^{(k)})\Delta x^{(k)} = 0. \quad (2.59)$$

Перейдем от векторной записи к подробной, тогда выражение (2.59) будет выглядеть так:

$$\left\{ \begin{array}{l} f_1(x_1^{(k)} + \Delta x_1^{(k)}, \dots, x_n^{(k)} + \Delta x_n^{(k)}) = f_1(x_1^{(k)}, \dots, x_n^{(k)}) + \\ \quad + \Delta x_1^{(k)} \frac{\partial f_1}{\partial x_1} + \dots + \Delta x_n^{(k)} \frac{\partial f_1}{\partial x_n} = 0, \\ \dots \\ f_n(x_1^{(k)} + \Delta x_1^{(k)}, \dots, x_n^{(k)} + \Delta x_n^{(k)}) = f_n(x_1^{(k)}, \dots, x_n^{(k)}) + \\ \quad + \Delta x_1^{(k)} \frac{\partial f_n}{\partial x_1} + \dots + \Delta x_n^{(k)} \frac{\partial f_n}{\partial x_n} = 0. \end{array} \right. \quad (2.60)$$

Так как в данной работе мы рассматриваем систему уравнений, то под производной  $f'(x)$  следует понимать матрицу Якоби системы функций  $f_1, f_2, \dots, f_n$  относительно переменных  $x_1, x_2, \dots, x_n$ , т.е.:

$$f'(x) = J(x) = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \dots & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \dots & \frac{\partial f_2}{\partial x_n} \\ \dots & \dots & \dots & \dots \\ \frac{\partial f_n}{\partial x_1} & \frac{\partial f_n}{\partial x_2} & \dots & \frac{\partial f_n}{\partial x_n} \end{bmatrix}. \quad (2.61)$$

Или в упрощенной форме записи:

$$f'(x) = J(x) = \left[ \frac{\partial f_i}{\partial x_j} \right], \quad (2.62)$$

где  $i, j = 1, 2, \dots, n$ .

Перепишем итоговое итерационное выражение (2.58) с учетом всех полученных соотношений:

$$f(x^{(k)}) + J(x^{(k)})\Delta x^{(k)} = 0. \quad (2.63)$$

Если  $\det J(x) = \det \left[ \frac{\partial f_i}{\partial x_j} \right] \neq 0$ , то  $\Delta x^{(k)} = -J^{-1}(x^{(k)})f(x^{(k)}) = 0$ .

Можем утверждать, что поиск решения системы (2.54) по средством метода Ньютона [15] основан на построение итерационной последовательности:



$$x^{(k+1)} = x^{(k)} - J^{-1}(x^{(k)})f(x^{(k)}) = 0, \quad (2.64)$$

где  $k = 0, 1, 2, \dots, n$

Вычисление останавливается, когда все погрешности приближенных корней становятся достаточно малыми, т.е.  $\Delta x^{(k)} < \varepsilon$ , где  $\varepsilon$  – необходимая точность решения. Если точность не достигнута, то новые значения  $x_i$  используются как приближенные значения корней, и итерационный процесс повторяется до тех пор, пока не найдено решение или не станет ясно, что получить его нельзя.

Упростим систему уравнений связи для простоты составления матрицы Якоби:

$$\left\{ \begin{array}{l}
S_1 \sin \gamma + l \sin \alpha_1 = +S_2 \sin \gamma + l \sin \alpha_2, \\
S_2 \sin \gamma + l \sin \alpha_2 = S_3 \sin \gamma + l \sin \alpha_3, \\
-S_2 \frac{\cos \gamma}{2} - \frac{l \cos \alpha_2 \cos \beta_2}{2} - \frac{\sqrt{3} l \cos \alpha_2 \sin \beta_2}{2} - \frac{a}{2} - \\
-S_1 \cos \gamma - l \cos \beta_1 \cos \alpha_1 - a = -\frac{\sqrt{3}L}{2}, \\
-S_3 \frac{\cos \gamma}{2} - \frac{\sqrt{3} l \cos \alpha_3 \sin \beta_3}{2} + \frac{l \cos \alpha_3 \cos \beta_3}{2} - \frac{a}{2} - \\
-S_1 \cos \gamma - l \cos \beta_1 \cos \alpha_1 - a = -\frac{\sqrt{3}L}{2}, \\
S_2 \frac{\sqrt{3} \cos \gamma}{2} + \frac{\sqrt{3} l \cos \alpha_2 \cos \beta_2}{2} - \frac{l \cos \alpha_2 \sin \beta_2}{2} + \frac{\sqrt{3}a}{2} - \\
-l \cos \alpha_1 \sin \beta_1 = \frac{L}{2}, \\
-S_3 \frac{\sqrt{3} \cos \gamma}{2} - \frac{\sqrt{3} l \cos \alpha_3 \cos \beta_3}{2} - \frac{l \cos \alpha_3 \sin \beta_3}{2} - \frac{\sqrt{3}a}{2} - \\
-l \cos \alpha_1 \sin \beta_1 = -\frac{L}{2}, \\
x_C + \frac{1}{3} \left[ S_1 \cos \gamma + l \cos \beta_1 \cos \alpha_1 + a - S_2 \frac{\cos \gamma}{2} - \frac{l \cos \alpha_2 \cos \beta_2}{2} - \right. \\
\left. \frac{\sqrt{3} l \cos \alpha_2 \sin \beta_2}{2} - \frac{a}{2} - S_3 \frac{\cos \gamma}{2} - \frac{\sqrt{3} l \cos \alpha_3 \sin \beta_3}{2} + \right. \\
\left. + \frac{l \cos \alpha_3 \cos \beta_3}{2} - \frac{a}{2} \right] = 0, \\
y_C + \frac{1}{3} \left[ l \cos \alpha_1 \sin \beta_1 + S_2 \frac{\sqrt{3} \cos \gamma}{2} + \frac{\sqrt{3} l \cos \alpha_2 \cos \beta_2}{2} - \right. \\
\left. - \frac{l \cos \alpha_2 \cos \beta_2}{2} + \frac{\sqrt{3}a}{2} - \right. \\
\left. - S_3 \frac{\sqrt{3} \cos \gamma}{2} - \frac{\sqrt{3} l \cos \alpha_3 \cos \beta_3}{2} - \frac{l \cos \alpha_3 \sin \beta_3}{2} - \frac{\sqrt{3}a}{2} \right] = 0, \\
z_C + \frac{1}{3} [S_1 \sin \gamma + l \sin \alpha_1 + S_2 \sin \gamma + l \sin \alpha_2 + S_3 \sin \gamma + l \sin \alpha_3] = 0.
\end{array} \right. \quad (2.65)$$

Для простоты и наглядности произведем замену неизвестных переменных:

$$x_1 = \sin \alpha_1, x_2 = \sin \alpha_2, x_3 = \sin \alpha_3, x_4 = \sin \beta_1, x_5 = \sin \beta_2, x_6 = \sin \beta_3,$$

$$y_1 = \cos \alpha_1, y_2 = \cos \alpha_2, y_3 = \cos \alpha_3, y_4 = \cos \beta_1, y_5 = \cos \beta_2, y_6 = \cos \beta_3,$$

$$x_7 = S_1, x_8 = S_2, x_9 = S_3, y_7 = x_C, y_8 = y_C, y_9 = z_C.$$

Перепишем исходную систему уравнений (2.65) с учетом введенных неизвестных и добавим в нее еще 6 уравнений, которые являются по своей сути тригонометрическим тождеством:

$$\left\{ \begin{array}{l}
\frac{x_7 \sin \gamma}{l} + x_1 - \frac{x_8 \sin \gamma}{l} - x_2 = 0, \\
\frac{x_8 \sin \gamma}{l} + x_2 - \frac{x_3 \sin \gamma}{l} - x_2 = 0, \\
-x_8 \frac{\cos \gamma}{2} - \frac{ly_2 y_5}{2} - \frac{\sqrt{3ly_2 x_5}}{2} - \frac{3a}{2} - x_7 \cos \gamma - ly_1 y_4 + \frac{\sqrt{3}L}{2} = 0, \\
-x_9 \frac{\cos \gamma}{2} + \frac{ly_3 y_6}{2} - \frac{\sqrt{3ly_3 x_6}}{2} - \frac{3a}{2} - x_7 \cos \gamma - ly_1 y_4 + \frac{\sqrt{3}L}{2} = 0, \\
x_8 \frac{\sqrt{3} \cos \gamma}{2} + \frac{\sqrt{3ly_2 y_5}}{2} - \frac{ly_2 x_5}{2} + \frac{\sqrt{3a}}{2} - ly_1 x_4 - \frac{L}{2} = 0, \\
-S_3 \frac{\sqrt{3} \cos \gamma}{2} - \frac{\sqrt{3ly_3 x_6}}{2} - \frac{ly_3 x_6}{2} - \frac{\sqrt{3a}}{2} - ly_1 x_4 + \frac{L}{2} = 0, \\
y_7 + \frac{1}{3} [x_7 \cos \gamma + ly_4 y_1 + a - x_8 \frac{\cos \gamma}{2} - \frac{ly_2 y_5}{2} - \\
\frac{\sqrt{3ly_2 x_5}}{2} - \frac{a}{2} - x_9 \frac{\cos \gamma}{2} - \frac{\sqrt{3ly_3 x_6}}{2} + \frac{ly_3 y_6}{2} - \frac{a}{2}] = 0, \\
y_8 + \frac{1}{3} [ly_1 x_4 + x_8 \frac{\sqrt{3} \cos \gamma}{2} + \frac{\sqrt{3ly_2 y_5}}{2} - \frac{ly_2 y_5}{2} + \frac{\sqrt{3a}}{2} - \\
-x_9 \frac{\sqrt{3} \cos \gamma}{2} - \frac{\sqrt{3ly_3 y_6}}{2} - \frac{ly_3 x_6}{2} - \frac{\sqrt{3a}}{2}] = 0, \\
y_9 + \frac{1}{3} [x_7 \sin \gamma + lx_1 + x_8 \sin \gamma + lx_2 + x_9 \sin \gamma + lx_3] = 0, \\
x_1^2 + y_1^2 - 1 = 0, \\
x_2^2 + y_2^2 - 1 = 0, \\
x_3^2 + y_3^2 - 1 = 0, \\
x_4^2 + y_4^2 - 1 = 0, \\
x_5^2 + y_5^2 - 1 = 0, \\
x_6^2 + y_6^2 - 1 = 0.
\end{array} \right. \quad (2.66)$$

Составим матрицу Якоби, которая необходима для поиска решения исходной нелинейной системы уравнений. Найдем частные производные каждой функции по каждой неизвестной.

$$\begin{aligned}
\frac{\partial f_1}{\partial x_1} &= 1, \frac{\partial f_1}{\partial x_2} = -1, \frac{\partial f_1}{\partial x_7} = \frac{\sin \gamma}{l}, \frac{\partial f_1}{\partial x_8} = -\frac{\sin \gamma}{l}; \\
\frac{\partial f_2}{\partial x_2} &= 1, \frac{\partial f_2}{\partial x_3} = -1, \frac{\partial f_2}{\partial x_8} = \frac{\sin \gamma}{l}, \frac{\partial f_2}{\partial x_9} = -\frac{\sin \gamma}{l}; \\
\frac{\partial f_3}{\partial x_5} &= -\frac{\sqrt{3}ly_2}{2}, \frac{\partial f_3}{\partial x_7} = -\cos \gamma, \frac{\partial f_3}{\partial x_8} = -\frac{\cos \gamma}{2}, \frac{\partial f_3}{\partial y_1} = -ly_4, \frac{\partial f_3}{\partial y_2} = -\frac{ly_5}{2} - \frac{\sqrt{3}lx_5}{2}, \\
\frac{\partial f_3}{\partial y_4} &= -ly_1, \frac{\partial f_3}{\partial y_5} = -\frac{ly_2}{2};
\end{aligned}$$

$$\frac{\partial f_4}{\partial x_6} = -\frac{\sqrt{3}lx_6}{2}, \frac{\partial f_4}{\partial x_7} = -\cos\gamma, \frac{\partial f_4}{\partial x_9} = -\frac{\cos\gamma}{2}, \frac{\partial f_4}{\partial y_1} = -ly_4, \frac{\partial f_4}{\partial y_3} = \frac{ly_6}{2} - \frac{\sqrt{3}lx_6}{2},$$

$$\frac{\partial f_4}{\partial y_4} = -ly_1, \frac{\partial f_4}{\partial y_6} = \frac{ly_3}{2};$$

$$\frac{\partial f_5}{\partial x_4} = -ly_1, \frac{\partial f_5}{\partial x_5} = -\frac{ly_2}{2}, \frac{\partial f_5}{\partial x_8} = \frac{\sqrt{3}\cos\gamma}{2}, \frac{\partial f_5}{\partial y_1} = -lx_4, \frac{\partial f_5}{\partial y_2} = \frac{\sqrt{3}ly_5}{2} - \frac{lx_5}{2},$$

$$\frac{\partial f_5}{\partial y_5} = \frac{\sqrt{3}ly_2}{2};$$

$$\frac{\partial f_6}{\partial x_4} = -ly_1, \frac{\partial f_6}{\partial x_6} = -\frac{ly_3}{2}, \frac{\partial f_6}{\partial x_9} = -\frac{\sqrt{3}\cos\gamma}{2}, \frac{\partial f_6}{\partial y_1} = -lx_4, \frac{\partial f_6}{\partial y_3} = -\frac{\sqrt{3}ly_6}{2} - \frac{lx_6}{2},$$

$$\frac{\partial f_6}{\partial y_6} = -\frac{\sqrt{3}ly_3}{2};$$

$$\frac{\partial f_7}{\partial x_5} = -\frac{\sqrt{3}ly_2}{6}, \frac{\partial f_7}{\partial x_6} = -\frac{\sqrt{3}ly_3}{2}, \frac{\partial f_7}{\partial x_7} = \frac{\cos\gamma}{3}, \frac{\partial f_7}{\partial x_8} = -\frac{\cos\gamma}{6}, \frac{\partial f_7}{\partial x_9} = -\frac{\cos\gamma}{6},$$

$$\frac{\partial f_7}{\partial y_1} = \frac{ly_4}{3}, \frac{\partial f_7}{\partial y_2} = -\frac{ly_5}{6} - \frac{\sqrt{3}lx_5}{6}, \frac{\partial f_7}{\partial y_3} = \frac{ly_6}{6} - \frac{\sqrt{3}lx_6}{6}, \frac{\partial f_7}{\partial y_4} = \frac{ly_1}{3}, \frac{\partial f_7}{\partial y_5} = -\frac{ly_2}{6},$$

$$\frac{\partial f_7}{\partial y_6} = \frac{ly_3}{6};$$

$$\frac{\partial f_8}{\partial x_4} = \frac{ly_1}{3}, \frac{\partial f_8}{\partial x_5} = \frac{ly_2}{6}, \frac{\partial f_8}{\partial x_6} = -\frac{ly_3}{6}, \frac{\partial f_8}{\partial x_8} = \frac{\sqrt{3}\cos\gamma}{6}, \frac{\partial f_8}{\partial x_9} = -\frac{\sqrt{3}\cos\gamma}{6},$$

$$\frac{\partial f_8}{\partial y_1} = \frac{lx_4}{3}, \frac{\partial f_8}{\partial y_2} = \frac{\sqrt{3}ly_5}{6} - \frac{lx_5}{6}, \frac{\partial f_8}{\partial y_3} = -\frac{\sqrt{3}ly_6}{6} - \frac{lx_6}{6}, \frac{\partial f_8}{\partial y_5} = \frac{\sqrt{3}ly_2}{6},$$

$$\frac{\partial f_8}{\partial y_6} = -\frac{\sqrt{3}ly_3}{6};$$

$$\frac{\partial f_9}{\partial x_1} = \frac{l}{3}, \frac{\partial f_9}{\partial x_2} = \frac{l}{3}, \frac{\partial f_9}{\partial x_3} = \frac{l}{3}, \frac{\partial f_9}{\partial x_7} = \frac{\sin\gamma}{3}, \frac{\partial f_9}{\partial x_8} = \frac{\sin\gamma}{3}, \frac{\partial f_9}{\partial x_9} = \frac{\sin\gamma}{3};$$

$$\frac{\partial f_{10}}{\partial x_1} = 2x_1, \frac{\partial f_{10}}{\partial y_1} = 2y_1;$$

$$\frac{\partial f_{11}}{\partial x_2} = 2x_2, \frac{\partial f_{11}}{\partial y_2} = 2y_2;$$

$$\frac{\partial f_{12}}{\partial x_3} = 2x_3, \frac{\partial f_{12}}{\partial y_3} = 2y_3;$$

$$\frac{\partial f_{13}}{\partial x_4} = 2x_4, \frac{\partial f_{13}}{\partial y_4} = 2y_4;$$
$$\frac{\partial f_{14}}{\partial x_5} = 2x_5, \frac{\partial f_{14}}{\partial y_5} = 2y_5;$$
$$\frac{\partial f_{15}}{\partial x_6} = 2x_6, \frac{\partial f_{15}}{\partial y_6} = 2y_6.$$

Полученные частные производные составляют матрицу Якоби. Решения прямой и обратной задач кинематики были получены с помощью разработанного приложения в среде MATLAB 6 [4]. Подробное описание о работе разработанного приложения с графическим интерфейсом представлено в 4 главе.

В представленной работе прямая задача кинематики состоит в нахождении координат захвата при заданных координатах кареток приводов, обратная в точности, наоборот.

Алгоритмы программ для поиска решений обратной и прямой задач кинематики практически не имеют структурных отличий. Единственной разницей является задание начальных условий, оно (начальное условие) различно для двух решаемых задач.

Код программы для решения обратной задачи кинематики указан в приложении 2. Код программы для решения прямой задачи кинематики указан в приложении 3.

Неотъемлемой задачей при анализе кинематики манипулятора является формирование области обслуживания, которая представляет собой множество точек, полученных при пространственных перемещениях захвата «трипода».

Основываясь на систему уравнений связи, полученную ранее и разработав функционал в приложении, который позволяет вычислить координаты всех точек, с учетом конструктивных ограничений.

Итогом выполнения данной программы является точечное графическое отображение области достижимости в пространстве, как представлено на рисунке 2.7.

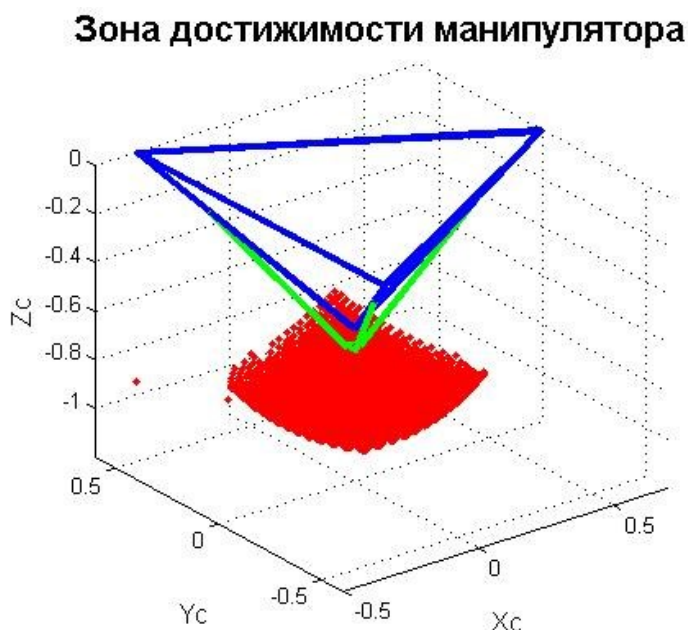


Рисунок 2.7 – График зоны достижимости «трипода»

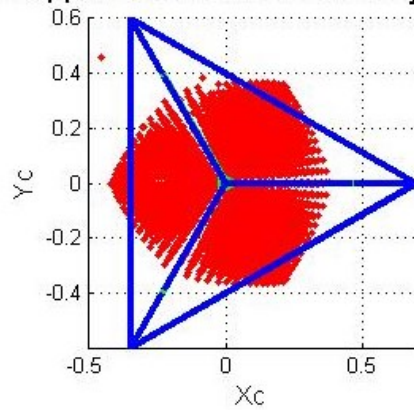
На рисунке 2.8 изображена область достижимости в различных плоскостях.

Отметим следующие особенности, которые явно просматриваются на рисунке 2.8:

- Наличие выпуклой относительно оси  $Z$  куполообразной области.
- Видна симметрия относительно приводов манипулятора.
- Явно выражены вершины области достижимости при максимальном удалении одной из кареток привода.

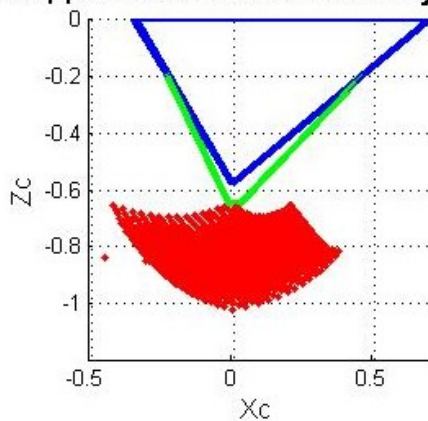
а)

**Зона достижимости манипулятора**



б)

**Зона достижимости манипулятора**



в)

**Зона достижимости манипулятора**

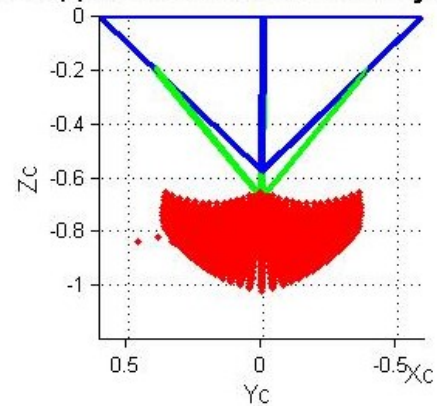


Рисунок 2.8 – Зоны достижимости в координатах а)  $0X_cY_c$ , б)  $0X_cZ_c$ , в)  $0Y_cZ_c$ .

Листинг программы для определения зоны достижимости «трипода» представлен в приложении 4. Описание разработанного для программы интерфейса подробнее представлено в 4 главе.

## 2.3 Краткие выводы по главе

1. Сформирована система уравнений, которая связывает координаты двух оснований данного манипулятора и дает полное описание его кинематики движения.

2. Получены решения прямой и обратной задачи кинематики посредством использования итерационного метода Ньютона. Совпадение полученных решений говорит о правильности выбранного метода, а также учета всех конструктивных ограничений.

3. Получена зона обслуживания манипулятора при различных начальных положениях кареток на направляющей. Траектории точек перемещения захвата формируют параболическую симметричную относительно приводов поверхность. Части, которые отображают максимумы траектории описывают граничные положения кареток по каждой из направляющих.

4. Разработаны программы в среде MATLAB 6 [4], которые позволяют получить расчет кинематики движения и графическое отображение зоны обслуживания для манипуляторов подобной конструкции, с возможностью задания различных технических характеристик.



# 3 ДИНАМИКА РОБОТА С ПАРАЛЛЕЛЬНОЙ КИНЕМАТИКОЙ

## 3.1. Анализ манипулятора и вывод описания динамической модели

Для описание динамической модели манипулятора необходимо рассмотреть его общую схему, представленную на рисунке 3.1. В данном случае динамическая модель учитывает только силы, действующие на подвижную платформу.

Для решения задачи динамики [16] воспользуемся самым распространенным способом - составлением уравнения Лагранжа второго рода.

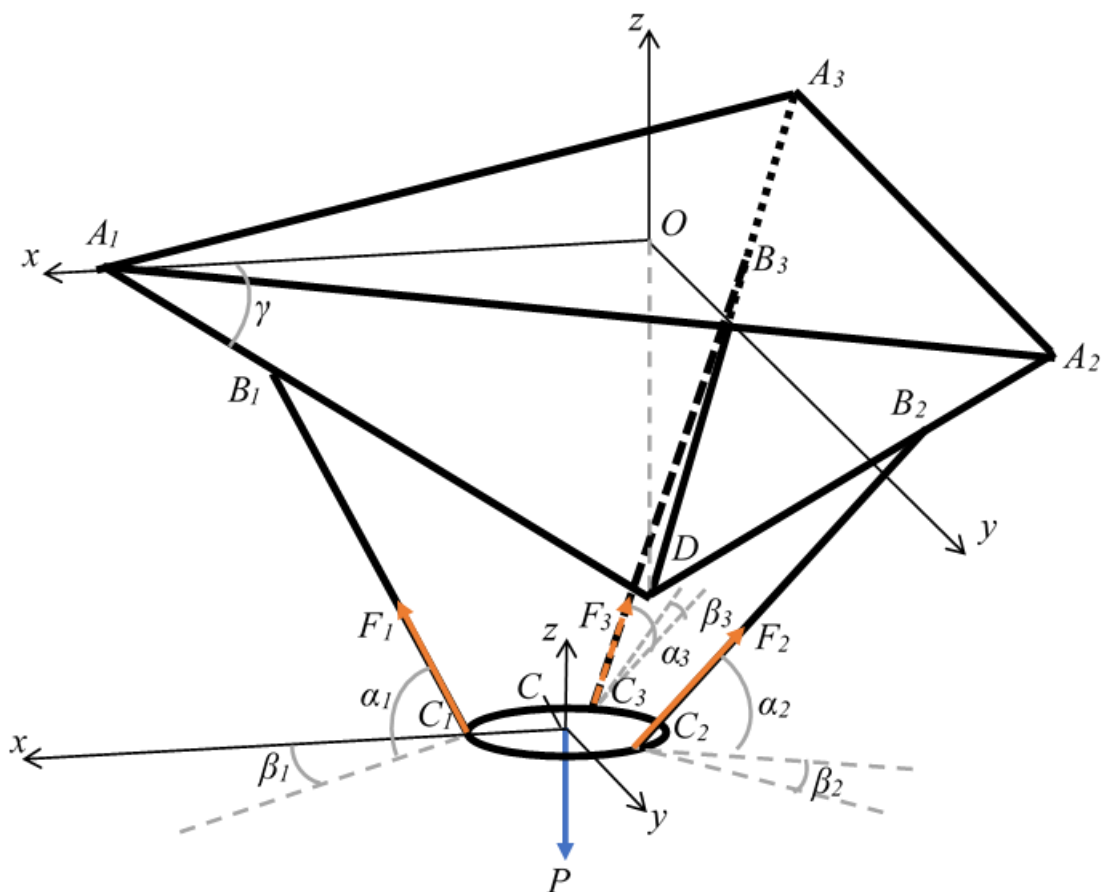


Рисунок 3.1 Схема механизма с системой сил

Будем рассматривать нашу систему с невесомыми штоками, пусть массой обладает только нижнее основание  $C_1 C_2 C_3$ , его центр масс находится в точке  $C$ . На рис. 3.1 видны силы, с которыми штоки действуют на нижнее основание «трипода», а именно силы  $F_1, F_2, F_3$ .

Исследуем динамику этой механической системы со штоками переменной длины.

Пусть в качестве обобщенных координат будут выступать координаты точки  $C$ :  $q_1 = x_C, q_2 = y_C, q_3 = z_C$ .

Уравнение Лагранжа второго рода выглядит следующим образом:

$$\frac{d}{dt} \left( \frac{\partial T}{\partial \dot{q}_i} \right) - \frac{\partial T}{\partial q_i} = Q_i, \quad i=1, 2, 3, \quad (3.1)$$

где  $T$  – кинетическая энергия системы,  $Q_i$  – обобщенные силы, которые соответствуют обобщенным координатам  $q_i$ .

Рассмотрим подвижное основание как материальную точку  $C$ , которая описывает центр масс платформы и имеет массу  $m$ , а штоки предполагаем невесомыми.

Запишем выражение для кинетической энергии:

$$T = \frac{m}{2} \sum_{i=1}^3 \dot{q}_i^2, \quad i=1, 2, 3. \quad (3.2)$$

Из выражения (3.2) понимаем, что частные производные по обобщенным координатам будут нулевыми.

Тогда итоговая система уравнений связи (3.1) переписется в виде:

$$m\ddot{q}_i = Q_i, \quad i=1, 2, 3. \quad (3.3)$$

Найдем обобщенные силы  $Q_i$  в проекциях на оси системы  $Cxyz$  для составления выражения для них.

На точку  $C$  действует сила тяжести  $P=(0, 0, -mg)$  и три силы  $F_1, F_2, F_3$ , направленные вдоль штоков.

Найдем проекции сил поочередно, геометрическим методом. Для этого для каждой из сил, действующих со стороны штоков, изобразим геометрическое представление.

На рисунке 3.2 изображено геометрическое представление силы  $F_1$ .

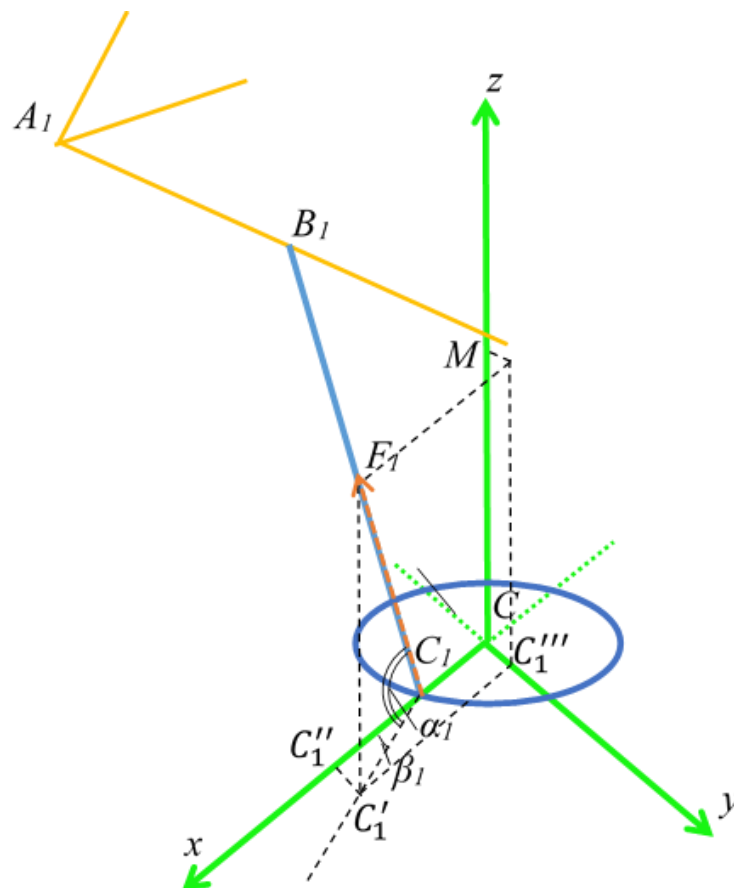


Рисунок 3.2 Проекция силы  $F_1$

Из рис. 3.2 видно, что:

$$F_1^x = |C_1 C_1''|,$$

$$|C_1 C_1''| = |C_1 C_1'| \cos \beta_1 = |F_1 C_1| \cos \beta_1 \cos \alpha_1 = F_1 \cos \beta_1 \cos \alpha_1,$$

$$F_1^y = |C C_1'''|,$$

$$|C C_1'''| = |C_1'' C_1'| = |C_1' C_1| \sin \beta_1 = F_1 \cos \alpha_1 \sin \beta_1,$$

$$F_1^z = |CM|,$$

$$|CM| = F_1 \sin \alpha_1.$$

Тогда для силы  $F_1$  проекции будут равны:

$$\begin{cases} F_1^x = F_1 \cos \beta_1 \cos \alpha_1, \\ F_1^y = F_1 \cos \beta_1 \cos \alpha_1, \\ F_1^z = F_1 \sin \alpha_1. \end{cases} \quad (3.4)$$

На рисунке 3.3 изображено геометрическое представление силы  $F_2$ .

Из рис. 3.3 видно, что:

$$F_2^x = |H C_2''|,$$

$$|H C_2''| = |C_2 C_2'''| = |C_2 C_2'| \sin(\beta_2 + \frac{\pi}{6}) = F_2 \sin(\beta_2 + \frac{\pi}{6}) \cos \alpha_2,$$

$$F_2^y = -|V V_1|,$$

$$|V V_1| = |C_2''' C_2'| = |C_2 C_2'| \cos(\beta_2 + \frac{\pi}{6}) = F_2 \cos \alpha_2 \cos(\beta_2 + \frac{\pi}{6}),$$

$$F_2^z = |H_1 C|,$$

$$|H_1 C| = F_2 \sin \alpha_2.$$

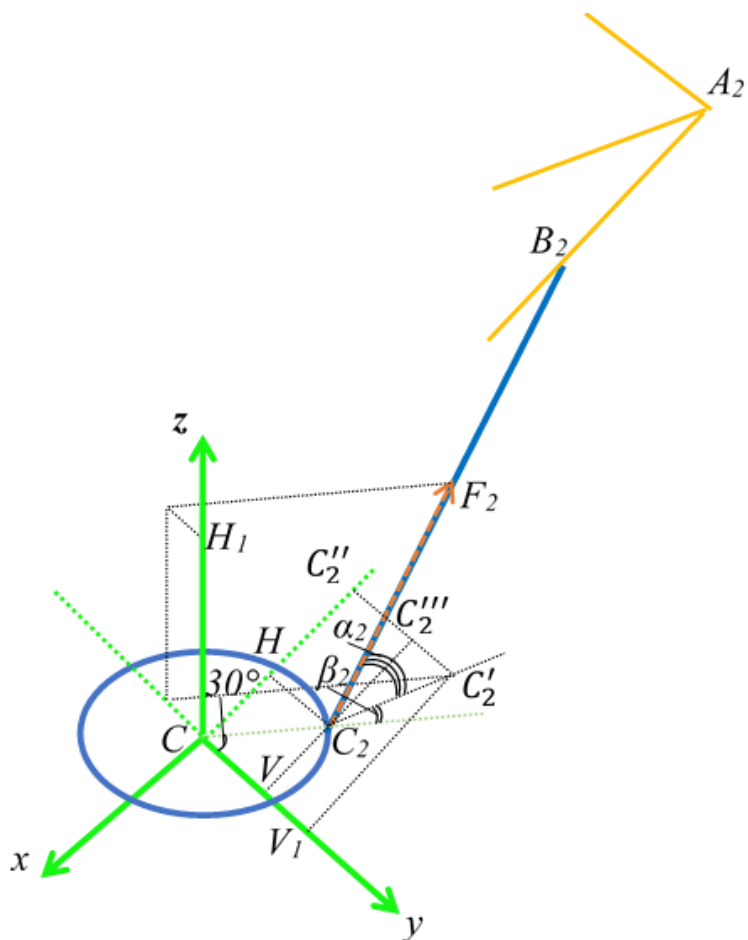


Рисунок 3.3 Проекция силы  $F_2$

Тогда для силы  $F_1$  проекции будут равны:

$$\begin{cases} F_2^x = F_2 \sin(\beta_2 + \frac{\pi}{6}) \cos \alpha_2, \\ F_2^y = -F_2 \cos \alpha_2 \cos(\beta_2 + \frac{\pi}{6}), \\ F_2^z = F_2 \sin \alpha_2. \end{cases} \quad (3.5)$$

На рисунке 3.4 изображено геометрическое представление силы  $F_3$ .

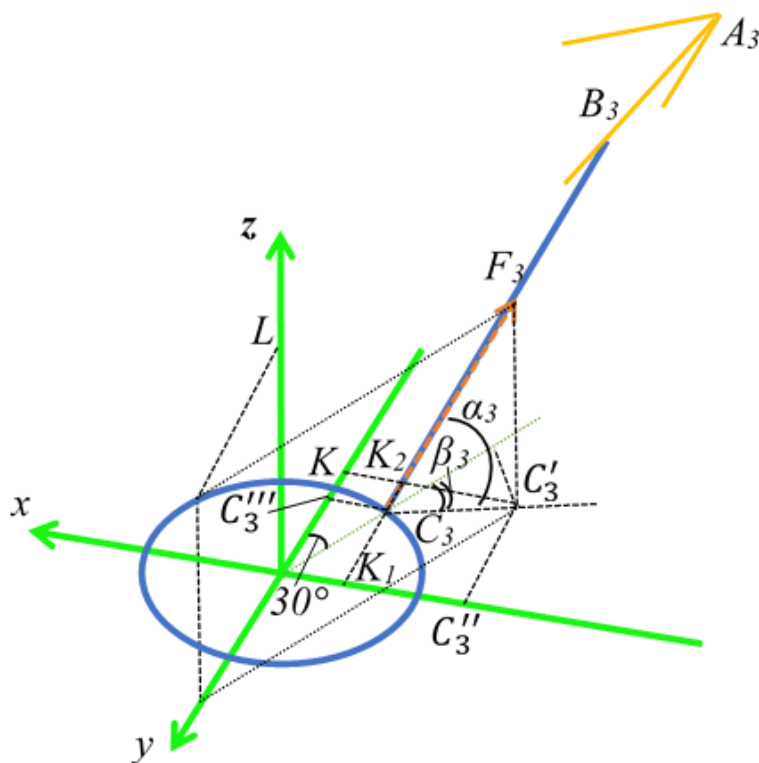


Рисунок 3.4 Проекция силы  $F_3$

Из рис. 3.4 видно, что:

$$F_3^x = -|K_1 C_3''|,$$

$$|K_1 C_3''| = |C_3' K_2| = |C_3 C_3'| \sin(\beta_3 - \frac{\pi}{6}) = F_3 \sin(\beta_3 - \frac{\pi}{6}) \cos \alpha_3,$$

$$F_3^y = -|K C_3'''|,$$

$$|K C_3'''| = |C_3 K_2| = |C_3 C_3'| \cos(\beta_3 - \frac{\pi}{6}) = F_3 \cos \alpha_3 \cos(\beta_3 - \frac{\pi}{6}),$$

$$F_3^z = |L C|,$$

$$|L C| = F_3 \sin \alpha_3.$$

Тогда для силы  $F_1$  проекции будут равны:

$$\begin{cases} F_3^x = -F_3 \sin(\beta_3 - \frac{\pi}{6}) \cos \alpha_3, \\ F_3^y = -F_3 \cos \alpha_3 \cos(\beta_3 - \frac{\pi}{6}), \\ F_3^z = F_3 \sin \alpha_3. \end{cases} \quad (3.6)$$

Выражение элементарной работы на возможном перемещении системы будет выглядеть:

$$\delta A = -mg\delta q_3 + \sum_{i=1}^3 (F_i^x \delta q_1 + F_i^y \delta q_2 + F_i^z \delta q_3), \quad i=1, 2, 3, \quad (3.7)$$

где  $F_i^x, F_i^y, F_i^z$  – проекции сил  $F_i$  на оси системы  $Sxyz$ .

Из полученного отношения (3.7) можем записать выражения для обобщенных сил, которые равны получившимся коэффициентам при независимых вариациях обобщенных координат  $\delta q_i$ :

$$Q_1 = \sum_{i=1}^3 F_i^x, \quad Q_2 = \sum_{i=1}^3 F_i^y, \quad Q_3 = -mg + \sum_{i=1}^3 F_i^z. \quad (3.8)$$

Подставим в (3.8) геометрические выражения сил (3.4 – 3.6) и получим итоговую запись обобщенных сил:

$$\begin{cases} Q_1 = F_1 \cos \beta_1 \cos \alpha_1 + F_2 \sin(\beta_2 + \frac{\pi}{6}) \cos \alpha_2 - F_3 \sin(\beta_3 - \frac{\pi}{6}) \cos \alpha_3, \\ Q_2 = F_1 \cos \beta_1 \cos \alpha_1 - F_2 \cos \alpha_2 \cos(\beta_2 + \frac{\pi}{6}) - F_3 \cos \alpha_3 \cos(\beta_3 - \frac{\pi}{6}), \\ Q_3 = -mg + F_1 \sin \alpha_1 + F_2 \sin \alpha_2 + F_3 \sin \alpha_3. \end{cases} \quad (3.9)$$

Подставив (3.9) в (3.3) окончательно составим уравнение Лагранжа 2-го рода.

$$\begin{cases} m\ddot{q}_1 = F_1 \cos \beta_1 \cos \alpha_1 + F_2 \sin(\beta_2 + \frac{\pi}{6}) \cos \alpha_2 - F_3 \sin(\beta_3 - \frac{\pi}{6}) \cos \alpha_3, \\ m\ddot{q}_2 = F_1 \cos \beta_1 \cos \alpha_1 - F_2 \cos \alpha_2 \cos(\beta_2 + \frac{\pi}{6}) - F_3 \cos \alpha_3 \cos(\beta_3 - \frac{\pi}{6}), \\ m\ddot{q}_3 = -mg + F_1 \sin \alpha_1 + F_2 \sin \alpha_2 + F_3 \sin \alpha_3. \end{cases} \quad (3.10)$$

Перейдем к решению прямой и обратной задач динамики. Обобщенные координаты могут быть заданы как функции от времени  $q_i = q_i(t)$ . В этом случае, решая систему (3.10), мы можем найти аналитически зависимость управляющих сил от времени:  $F_i = F_i(t)$ . Наоборот, при заданном законе изменения управляющих сил как функций времени можно, решив систему дифференциальных уравнений (3.10), найти закон движения материальной точки  $S$ .

### 3.2. Поиск решения прямой и обратной задачи динамики

Рассматривая динамику роботов-манипуляторов, выделяют две основные задачи. Прямая задача динамики подразумевает определение равнодействующих сил, действующих на тело по изначально заданному закону движения координаты (скорости или ускорения). Обратная задача позволяет установить характер перемещения тела, т.е. изменение координаты (скорости или ускорения) за счет задания сил.

Поиск решения прямой задачи динамики нетрудоемкий и подразумевает решение обыкновенного уравнения или системы уравнений, а также процесс дифференцирования обобщенных координат. Задача будет иметь единственное решение.

Интереснее дела обстоят с поиском решения обратной задачи. Здесь необходимо решить дифференциальное уравнение 2-го порядка. Процедура интегрирования подразумевает появление в решении постоянных интегрирования, в данном случае их будет по две на каждое уравнение. Наличие постоянных интегрирования указывает на то, что у системы есть бесконечное множество решений, иначе говоря силы не определяют однозначно движение точки. Одним и тем же силам могут соответствовать разные траектории движения. Чтобы найти постоянные интегрирования и свести количество решение к одному необходимо задать начальные условия, тем самым решить задачу Коши.



Для поиска решения воспользуемся стандартными пакетными функциями среды разработки MATLAB 6 [4]: «solve» – для поиска решения обыкновенного уравнения в символьном виде; «dsolve» - для поиска решения дифференциального уравнения в символьном виде.

Для дальнейшей простоты необходимо упростить систему (3.10). Воспользуемся тригонометрическими формулами приведения, тогда после преобразований система примет вид.

Система 3.11:

$$\begin{cases} m\ddot{q}_1 = F_1 \cos \beta_1 \cos \alpha_1 + F_2 \left( \frac{\sqrt{3}}{2} \sin \beta_2 - \frac{1}{2} \cos \beta_2 \right) \cos \alpha_2 - F_3 \left( \frac{\sqrt{3}}{2} \sin \beta_3 + \frac{1}{2} \cos \beta_3 \right) \cos \alpha_3, \\ m\ddot{q}_2 = F_1 \cos \beta_1 \cos \alpha_1 - F_2 \cos \alpha_2 \left( \frac{\sqrt{3}}{2} \cos \beta_2 + \frac{1}{2} \sin \beta_2 \right) - F_3 \cos \alpha_3 \left( \frac{\sqrt{3}}{2} \cos \beta_3 + \frac{1}{2} \sin \beta_3 \right), \\ m\ddot{q}_3 = -mg + F_1 \sin \alpha_1 + F_2 \sin \alpha_2 + F_3 \sin \alpha_3. \end{cases}$$

Введем замену, пусть  $x_1 = \alpha_1$ ,  $x_2 = \alpha_2$ ,  $x_3 = \alpha_3$ ,  $x_4 = \beta_1$ ,  $x_5 = \beta_2$ ,  $x_6 = \beta_3$ .

Тогда итоговая система примет вид:

$$\begin{cases} m\ddot{q}_1 = F_1 y_4 y_1 + F_2 \left( \frac{\sqrt{3}}{2} x_5 - \frac{1}{2} y_5 \right) y_2 - F_3 \left( \frac{\sqrt{3}}{2} x_6 + \frac{1}{2} y_6 \right) y_3, \\ m\ddot{q}_2 = F_1 y_4 y_1 - F_2 y_2 \left( \frac{\sqrt{3}}{2} + \frac{1}{2} x_5 \right) - F_3 y_3 \left( \frac{\sqrt{3}}{2} y_6 + \frac{1}{2} x_6 \right), \\ m\ddot{q}_3 = -mg + F_1 x_1 + F_2 x_2 + F_3 x_3. \end{cases} \quad (3.12)$$

Код программы и алгоритм работы приложения для решения задач динамики приведены в приложениях 6 и 7.

Проверим как себя поведет система при крайних случаях, т.е. зададимся законом движения во времени только для одной из трех обобщенных координат.

Для начала пусть  $q_1 = 0^*t$ ,  $q_2 = 0^*t$ ,  $q_3 = t$ . Это означает, что движение манипулятора происходит только по оси Z. Графически это можно увидеть на рисунке 3.5.

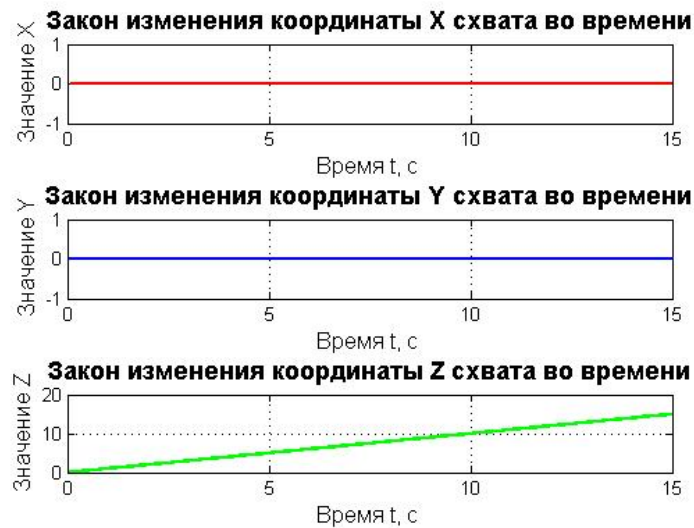


Рисунок 3.5 – Изменение координат во времени.

Так как закон движения задан прямолинейный, то ожидаемым результатом должно быть, что сила, действующая со стороны штоков, не изменяется во времени. Математически это связано с проведением дифференцирования обобщенных координат. Физически прямолинейный закон движения подразумевает отсутствие ускорения, а как известно ускорение - вторая производная от перемещения. Результирующие силы видны на рисунке 3.6.

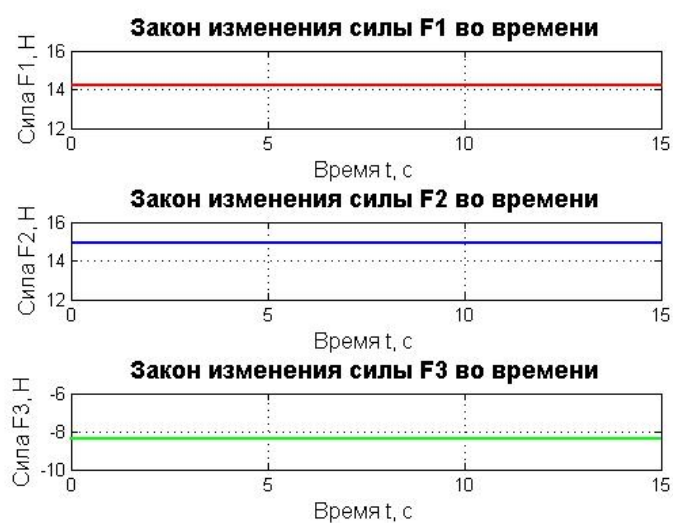


Рисунок 3.6 – Изменение сил во времени.

Для полученных сил решим обратную задачу динамики. Зададимся нулевыми начальными условиями. Решение обратной задачи подразумевает нахождение закона движения обобщенных координат. Результат представлен на рисунке 3.7.

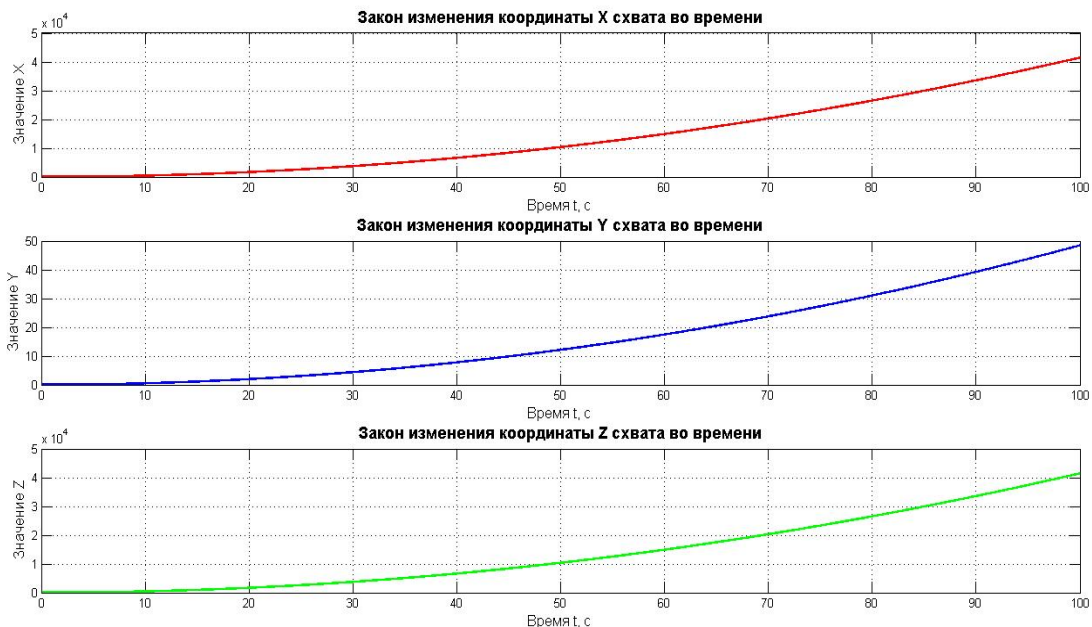


Рисунок 3.7 – Изменение координат во времени.

На графиках изменения видно, что законы движения совершенно не совпадают с изначально заданными. Т.к. решение дифференциального уравнения второго порядка подразумевает существование множества решений, а одно из возможных решений можно найти, задавшись определенными начальными условиями.

Те же результаты мы получим при рассмотрении случаев, когда прямолинейное движение происходит по оси X, т.е.  $q_1 = t, q_2 = 0*t, q_3 = 0*t$  (рисунки 3.8 и 3.9). И тоже для случая движения по оси Y:  $q_1 = 0*t, q_2 = t, q_3 = 0*t$  (рисунки 3.10 и 3.11).

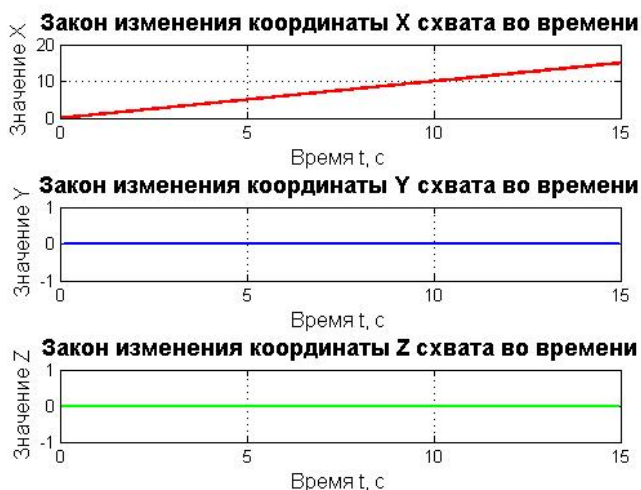


Рисунок 3.8 – Изменение координат во времени.

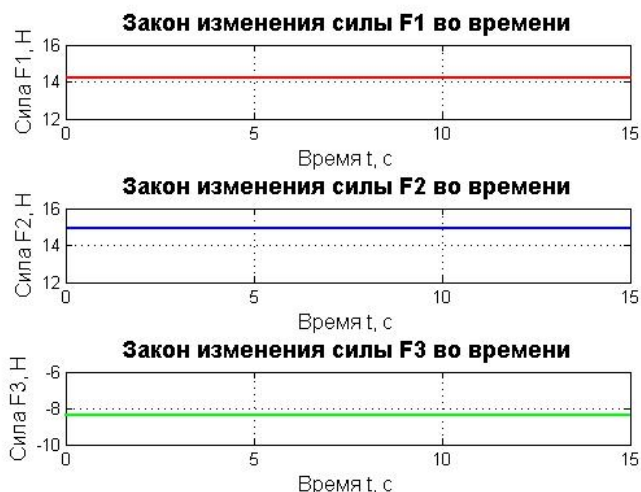


Рисунок 3.9 – Изменение сил во времени.

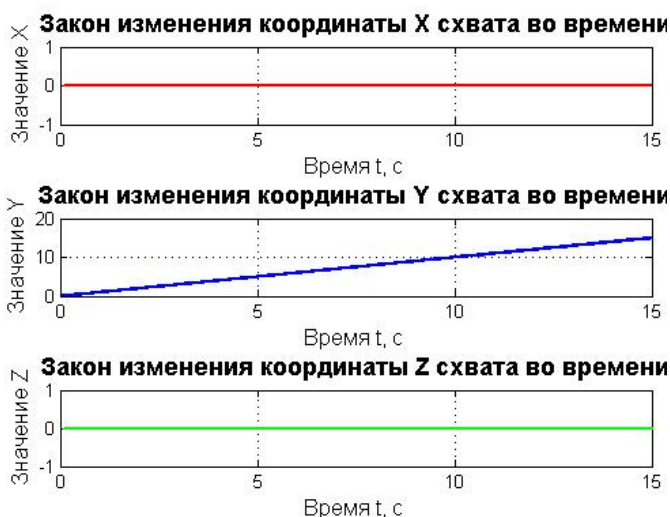


Рисунок 3.10 – Изменение координат во времени.

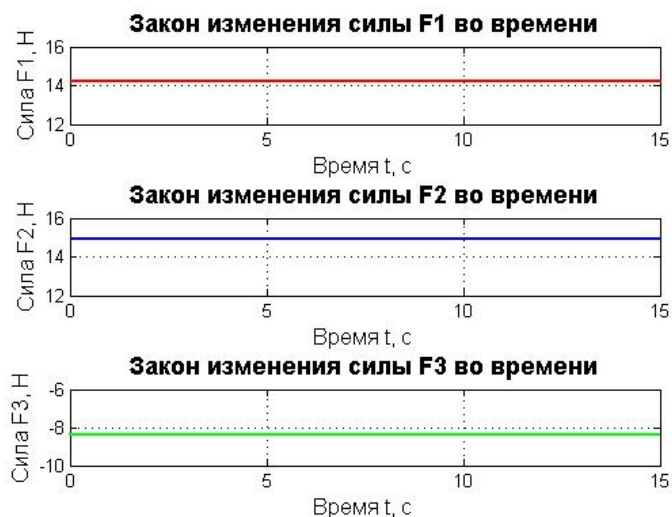


Рисунок 3.11 – Изменение сил во времени.

### 3.3. Краткие выводы по главе

1. Сформирована система уравнений Лагранжа второго рода, которая показывает влияние сил на подвижное основание в зависимости от закона

движения координат и дает полное описание динамики движения основания.

2. Получены решения прямой и обратной задачи динамики посредством использования встроенных программных методов в MATLAB 6 [4]. Полученные графические зависимости теоретически интерпретируются математическими законами дифференцирования и особенностями решения дифференциальных уравнений.
3. Разработаны программы в среде MATLAB 6 [4], которые позволяют получить расчет динамики движения и графическое отображение решений в виде временных законов движения координат схвата, а также законы изменения сил во времени, действующих на подвижную платформу.

## 4 РАЗРАБОТКА ПРИЛОЖЕНИЯ ДЛЯ РЕШЕНИЯ ЗАДАЧ

Получение аналитического решения для задач кинематики и динамики движения манипуляторов с параллельной структурой является сложным. Поэтому разработка подобного приложения является актуальной и значимой задачей. Ниже приведено подробное описание алгоритма работы всего приложения, разработка которого осуществлялась в среде MATLAB 6 [4].

### 4.1 Описание интерфейса разработанной программы по решению прямой, обратной задач кинематики и зоны достижимости

#### 4.1.1 Алгоритм работы «Главного окна» приложения

На рисунке 4.1 представлено главное окно разработанной пользовательского приложения.

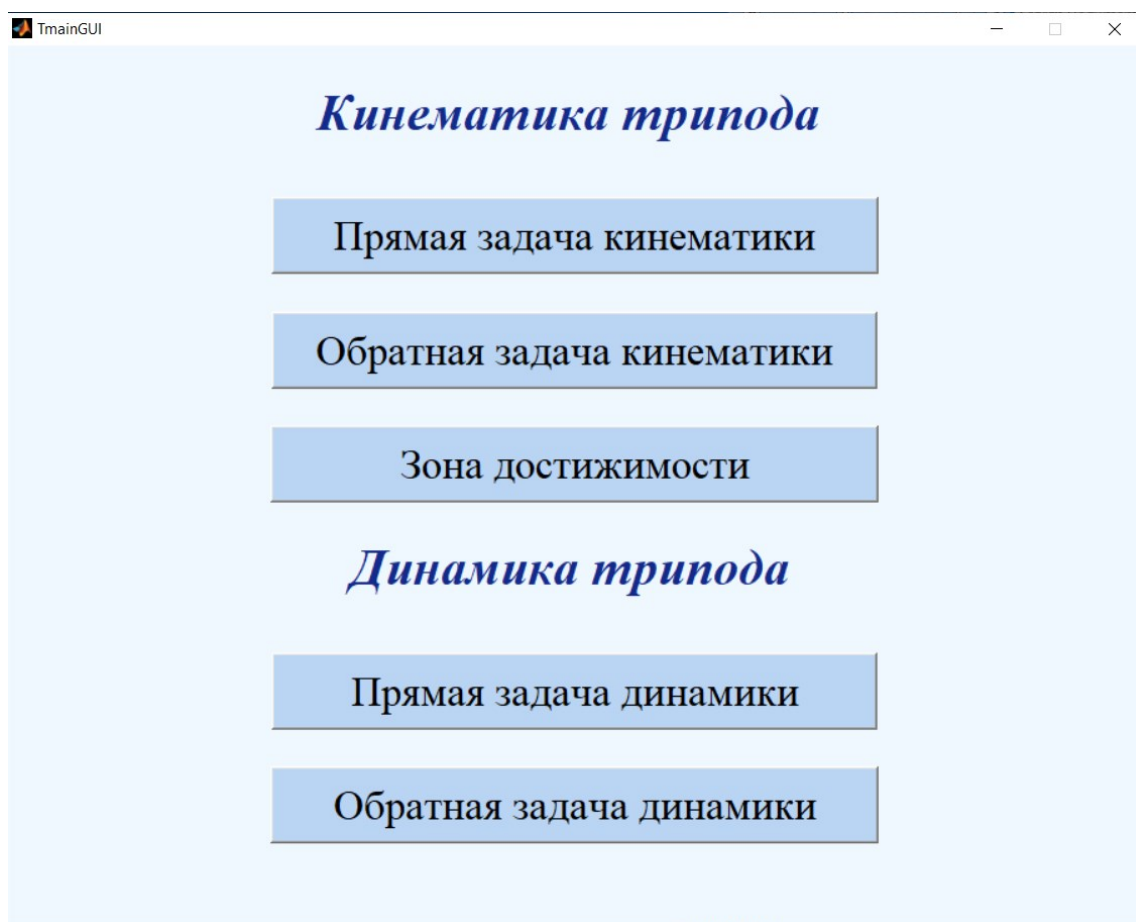


Рисунок 4.1. - Главное окно программы.

На главном окне (рис. 4.1) располагается 5 кнопок, разделенных на два раздела решаемых задач. Раздел кинематики содержит три кнопки: «Прямая задача кинематики», «Обратная задача кинематики» и «Зона достижимости». Раздел динамики имеет две кнопки: «Прямая задача динамики», «Обратная задача динамики». По нажатию любой из кнопок открывается соответствующее окно приложения.

Листинг кода для главного окна приложения с комментариями приведен в *приложении 1*.

#### 4.1.2 Алгоритм работы окна «Прямая задача кинематики» приложения

При решении прямой задачи кинематики открывается окно «Прямая задача кинематики» (рисунок 4.2).

| Прямая задача кинематики                  |                      |                                |                      |
|---|----------------------|--------------------------------|----------------------|
| Задайте начальные параметры манипулятора: |                      |                                |                      |
| Длина основания манипулятора, м           | <input type="text"/> | Конец хода кареток, м          | <input type="text"/> |
| Длина подвижного плеча трипода, м         | <input type="text"/> | Радиус подвижного основания, м | <input type="text"/> |
| Длина хода привода, м                     | <input type="text"/> | Шаг, м                         | <input type="text"/> |
| Начало хода кареток, м                    | <input type="text"/> | Точность решения               | <input type="text"/> |
| <b>Рассчитать</b>                         |                      |                                |                      |
| Дано:                                     |                      | Решение:                       |                      |
| Положение 1-ой каретки $S_1$ , м          | <input type="text"/> | Координата захвата $X_c$ , м   | <input type="text"/> |
| Положение 2-ой каретки $S_2$ , м          | <input type="text"/> | Координата захвата $Y_c$ , м   | <input type="text"/> |
| Положение 3-ой каретки $S_3$ , м          | <input type="text"/> | Координата захвата $Z_c$ , м   | <input type="text"/> |

Рисунок 4.2 Прямая задача кинематики

Для выполнения расчета необходимо задать параметры конструкции «трипода», точность решения и шаг, т.к. вычисление основано на методе Ньютона, а также положения кареток  $S_1$ ,  $S_2$ ,  $S_3$ . На рисунке 4.3 приведен пример корректно

введенных данных в выделенные динамические поля. Дробные числа должны вводиться в поле с разделителем «точка».

**Прямая задача кинематики**

Задайте начальные параметры манипулятора:

|                                   |      |                                |           |
|-----------------------------------|------|--------------------------------|-----------|
| Длина основания манипулятора, м   | 1.2  | Конец хода кареток, м          | 0.25      |
| Длина подвижного плеча трипода, м | 0.63 | Радиус подвижного основания, м | 0.027     |
| Длина хода привода, м             | 0.9  | Шаг, м                         | 0.025     |
| Начало хода кареток, м            | 0.3  | Точность решения               | 0.0000001 |

**Рассчитать**

|                              |     |                          |  |
|------------------------------|-----|--------------------------|--|
| <b>Дано:</b>                 |     | <b>Решение:</b>          |  |
| Положение 1-ой каретки S1, м | 0.1 | Координата захвата Xc, м |  |
| Положение 2-ой каретки S2, м | 0.2 | Координата захвата Yc, м |  |
| Положение 3-ой каретки S3, м | 0.3 | Координата захвата Zc, м |  |

Рисунок 4.3 Пример ввода корректных данных

После нажатия на кнопку «Рассчитать» выполняется программа, которая находит поиск решения прямой задачи кинематики. В соответствующей области в правом нижнем углу появятся координаты положения центра подвижной платформы, как это видно на рисунке 4.4.





Рисунок 4.4 Решение прямой задачи кинематики

В приложении предусмотрена проверка некорректного ввода данных, а именно параметров конструкции. Любая геометрическая величина, будь то длина или радиус не могут быть отрицательными, а величины, отражающие точность решения и шаг в методе Ньютона, должны лежать в пределах от 0 до 1 для получения верных результатов. В случае ввода некорректных данных будет появляться соответствующее окно с ошибкой. Пример, представлен на рисунке 4.5. После нажатия «ОК» динамическая область текста очищается и необходимо повторить попытку ввода параметров.

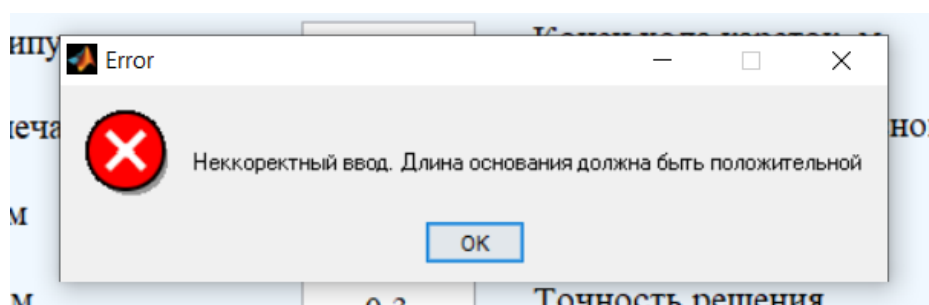


Рисунок 4.5 Окно при некорректном вводе параметров

Листинг кода для окна «Прямая задача кинематики» приложения с комментариями приведен в *приложении 2*.

### 4.1.3 Алгоритм работы окна «Обратная задача кинематики» приложения

При решении обратной задачи кинематики открывается окно «Обратная задача кинематики» (рисунок 4.6).

**Обратная задача кинематики**

Задайте начальные параметры манипулятора:

|                                   |                      |                                |                      |
|-----------------------------------|----------------------|--------------------------------|----------------------|
| Длина основания манипулятора, м   | <input type="text"/> | Конец хода кареток, м          | <input type="text"/> |
| Длина подвижного плеча трипода, м | <input type="text"/> | Радиус подвижного основания, м | <input type="text"/> |
| Длина хода привода, м             | <input type="text"/> | Шаг, м                         | <input type="text"/> |
| Начало хода кареток, м            | <input type="text"/> | Точность решения               | <input type="text"/> |

**Рассчитать**

|                              |                      |                                  |  |
|------------------------------|----------------------|----------------------------------|--|
| <b>Дано:</b>                 |                      | <b>Решение:</b>                  |  |
| Координата захвата $X_c$ , м | <input type="text"/> | Положение 1-ой каретки $S_1$ , м |  |
| Координата захвата $Y_c$ , м | <input type="text"/> | Положение 2-ой каретки $S_2$ , м |  |
| Координата захвата $Z_c$ , м | <input type="text"/> | Положение 3-ой каретки $S_3$ , м |  |

Рисунок 4.6 Обратная задача кинематики

Для выполнения вычислений необходимо задать параметры конструкции «трипода», точность решения и шаг, т.к. вычисление основано на методе Ньютона, а также координаты центра подвижной платформы  $x_c, y_c, z_c$ . На рисунке 4.7 приведен пример корректно введенных данных в выделенные динамические поля. Дробные числа должны вводиться в поле с разделителем «точка».

**Обратная задача кинематики**

Задайте начальные параметры манипулятора:

|                                   |                                   |                                |  |
|-----------------------------------|-----------------------------------|--------------------------------|--|
| Длина основания манипулятора, м   | <input type="text" value="1.2"/>  | Конец хода кареток, м          | <input type="text" value="0.25"/>      |
| Длина подвижного плеча трипода, м | <input type="text" value="0.63"/> | Радиус подвижного основания, м | <input type="text" value="0.027"/>     |
| Длина хода привода, м             | <input type="text" value="0.9"/>  | Шаг, м                         | <input type="text" value="0.025"/>     |
| Начало хода кареток, м            | <input type="text" value="0.3"/>  | Точность решения               | <input type="text" value="0.0000001"/> |

**Рассчитать**

Дано:

|                              |                                     |
|------------------------------|-------------------------------------|
| Координата захвата $X_c$ , м | <input type="text" value="0.188"/>  |
| Координата захвата $Y_c$ , м | <input type="text" value="0.137"/>  |
| Координата захвата $Z_c$ , м | <input type="text" value="-0.846"/> |

Решение:

Рисунок 4.7 Пример ввода корректных данных

Нажатие на кнопку «Рассчитать» запускает процесс выполнения программы, которая находит решение обратной задачи кинематики. В соответствующей области в правом нижнем углу появятся положения кареток манипулятора, как это видно на рисунке 4.8.

**Обратная задача кинематики**

Задайте начальные параметры манипулятора:

|                                   |                                   |                                |  |
|-----------------------------------|-----------------------------------|--------------------------------|--|
| Длина основания манипулятора, м   | <input type="text" value="1.2"/>  | Конец хода кареток, м          | <input type="text" value="0.25"/>      |
| Длина подвижного плеча трипода, м | <input type="text" value="0.63"/> | Радиус подвижного основания, м | <input type="text" value="0.027"/>     |
| Длина хода привода, м             | <input type="text" value="0.9"/>  | Шаг, м                         | <input type="text" value="0.025"/>     |
| Начало хода кареток, м            | <input type="text" value="0.3"/>  | Точность решения               | <input type="text" value="0.0000001"/> |

**Рассчитать**

Дано:

|                              |                                     |
|------------------------------|-------------------------------------|
| Координата захвата $X_c$ , м | <input type="text" value="0.188"/>  |
| Координата захвата $Y_c$ , м | <input type="text" value="0.137"/>  |
| Координата захвата $Z_c$ , м | <input type="text" value="-0.846"/> |

Решение:

|                              |                                    |
|------------------------------|------------------------------------|
| Положение 1-ой каретки S1, м | <input type="text" value="0.100"/> |
| Положение 2-ой каретки S2, м | <input type="text" value="0.200"/> |
| Положение 3-ой каретки S3, м | <input type="text" value="0.300"/> |

Рисунок 4.8 Решение обратной задачи кинематики

При вводе параметров конструкции предусмотрена проверка некорректного ввода данных. Любая геометрическая величина, будь то длина или радиус не могут быть отрицательными, а величины, отражающие точность решения и шаг в методе Ньютона, должны лежать в пределах от 0 до 1 для получения верных результатов. В случае ввода некорректных данных будет появляться соответствующее окно с ошибкой. Все виды ошибок представлены на рисунке 4.9. После нажатия «ОК» динамическая область текста очищается и необходимо повторить попытку ввода параметров.

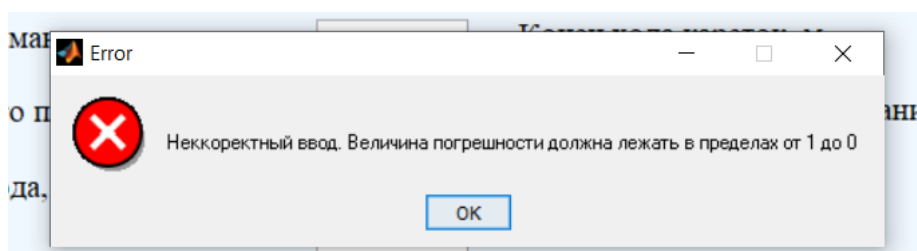


Рисунок 4.9 Окно при некорректном вводе параметров

Листинг кода для окна «Обратная задача кинематики» приложения с комментариями приведен в *приложении 3*.

#### 4.1.4 Алгоритм работы окна «Зона достижимости» приложения

При переходе в окно «Зона достижимости манипулятора» открывается окно, представленное на рисунке 4.10

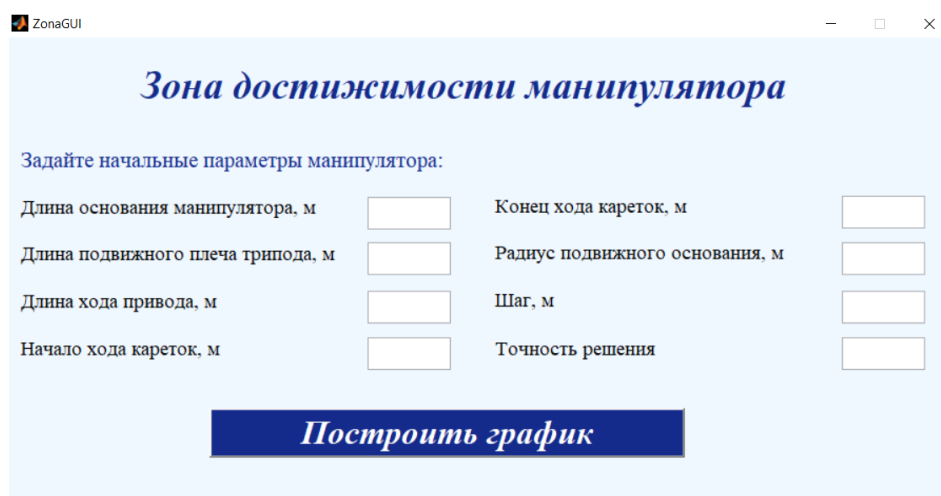


Рисунок 4.10 Окно приложения «Зона достижимости»

На рисунке 4.11 представлен пример корректного ввода параметров конструкции «трипода». Все дробные величины в динамические поля вводятся с разделителем «точка».



The screenshot shows a window titled "ZonaGUI" with the main heading "Зона достижимости манипулятора". Below the heading, it asks the user to "Задать начальные параметры манипулятора:". There are eight input fields arranged in two columns. The first column contains: "Длина основания манипулятора, м" (1.2), "Длина подвижного плеча трипода, м" (0.63), "Длина хода привода, м" (0.9), and "Начало хода кареток, м" (0.3). The second column contains: "Конец хода кареток, м" (0.25), "Радиус подвижного основания, м" (0.027), "Шаг, м" (0.025), and "Точность решения" (0.0000001). At the bottom center, there is a blue button with the text "Построить график".

| Задать начальные параметры манипулятора: |      |                                |           |
|--|------|--------------------------------|-----------|
| Длина основания манипулятора, м          | 1.2  | Конец хода кареток, м          | 0.25      |
| Длина подвижного плеча трипода, м        | 0.63 | Радиус подвижного основания, м | 0.027     |
| Длина хода привода, м                    | 0.9  | Шаг, м                         | 0.025     |
| Начало хода кареток, м                   | 0.3  | Точность решения               | 0.0000001 |

**Построить график**

Рисунок 4.11 Пример корректного ввода данных

После нажатия кнопки «Построить график» запускается процесс вычисления и на экран выводится 3-х мерное изображение области достижимости манипулятора. Пример отображения графика представлен на рисунке 4.12.

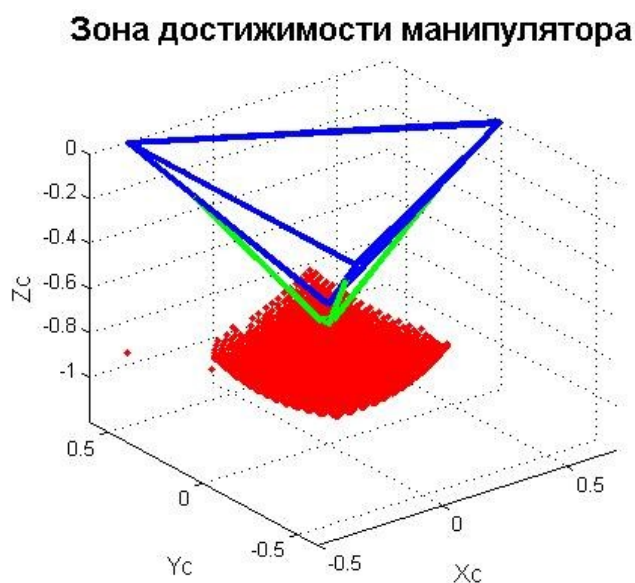


Рисунок 4.12 Пример вывод графика области достижимости

В данной части интерфейса на вводимые данные в динамические поля также наложены ограничения, которые отображаются в виде окон с ошибками, которые представлены ниже на рисунке 4.13.

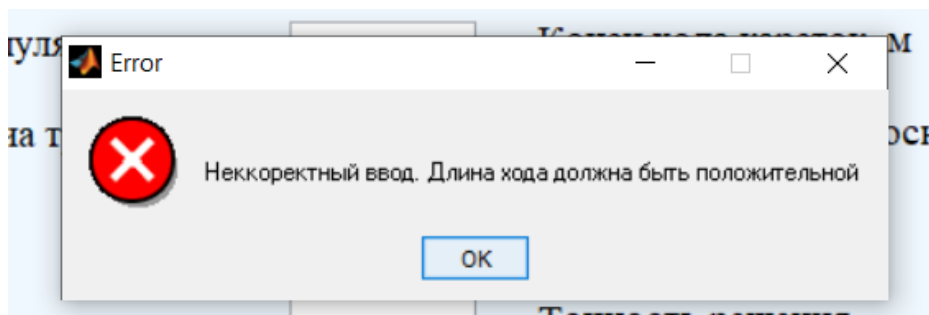


Рисунок 4.13 Окно при некорректном вводе параметров

Листинг кода для окна «Зона достижимости» приложения с комментариями приведен в *приложении 4*.

## **4.2 Описание интерфейса разработанной программы по решению прямой, обратной задач динамики**

### **4.2.1 Алгоритм работы окна «Прямая задача динамики» приложения**

При решении прямой задачи динамики открывается окно «Прямая задача динамики» (рисунок 4.14).

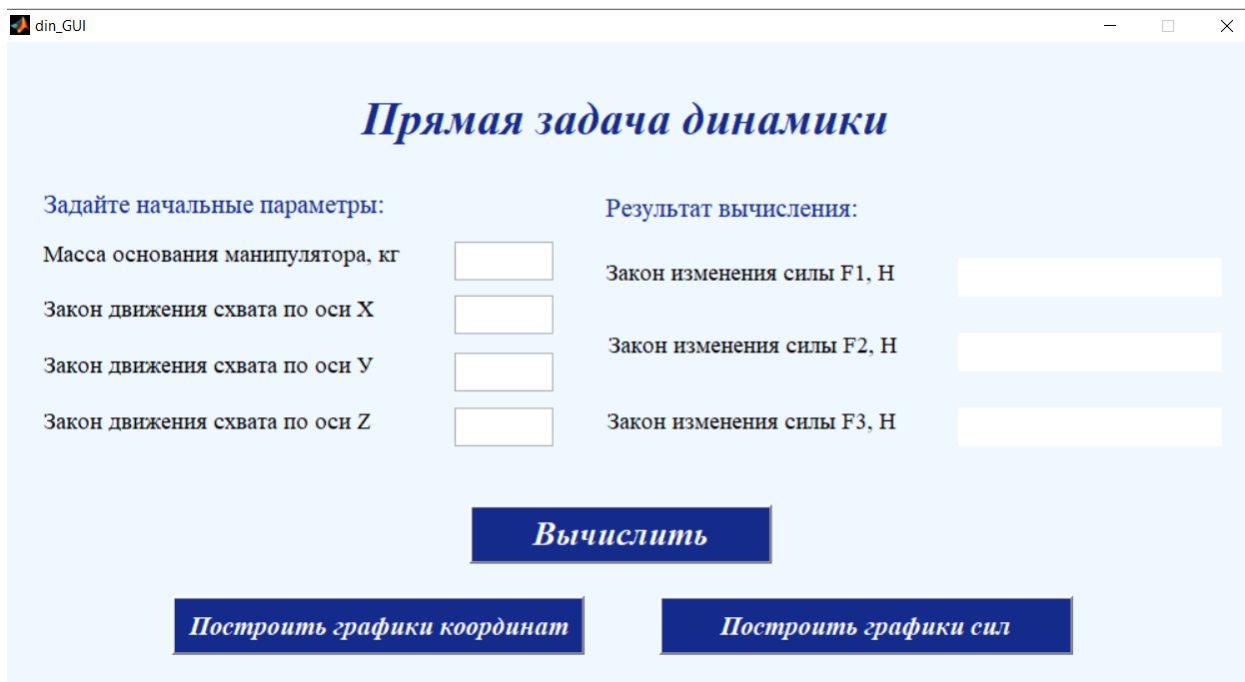


Рисунок 4.14 - Окно «Прямая задача»

Для выполнения вычисления необходимо ввести начальные параметры, а точнее: массу подвижного основания -  $m$ , законы изменения движения координат во времени –  $q_1(t)$ ,  $q_2(t)$ ,  $q_3(t)$ . На рисунке 4.15 приведен пример корректного ввода данных в поля динамического текста.

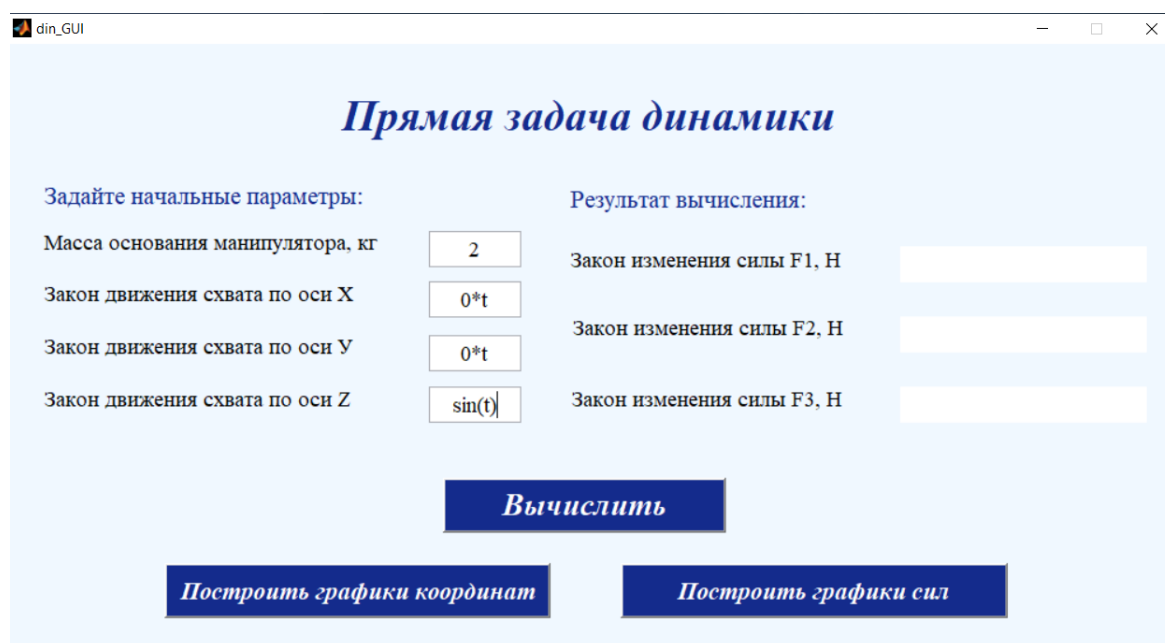
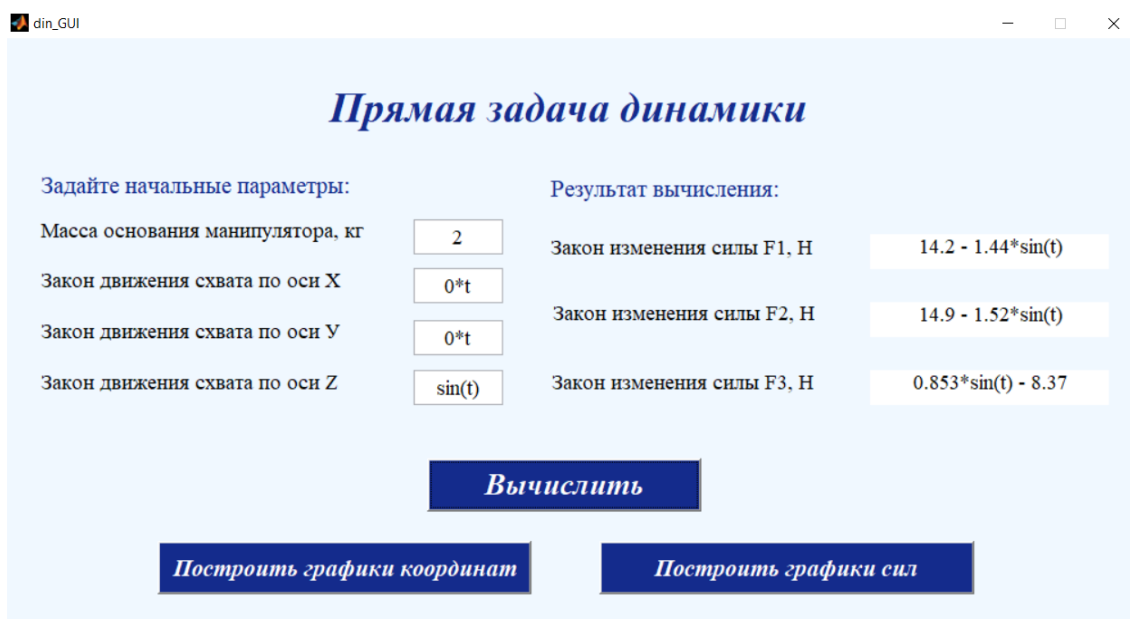


Рисунок 4.15 – Пример корректного ввода данных

После нажатия на кнопку «Вычислить» происходит выполнение программы, решающей прямую задачу динамики, а именно вычисление законов изменения сил, действующих со стороны штоков, во времени. Математическое решение прямой задачи подразумевает решение системы уравнений.

На рисунке 4.16 приведен пример расчета. В интерфейсе предусмотрено построение графиков для визуального отображения законов изменения сил и координат во времени.



**Прямая задача динамики**

Задайте начальные параметры:

|                                  |                                     |
|----------------------------------|-------------------------------------|
| Масса основания манипулятора, кг | <input type="text" value="2"/>      |
| Закон движения схвата по оси X   | <input type="text" value="0*t"/>    |
| Закон движения схвата по оси Y   | <input type="text" value="0*t"/>    |
| Закон движения схвата по оси Z   | <input type="text" value="sin(t)"/> |

Результат вычисления:

|                            |  |
|----------------------------|--|
| Закон изменения силы F1, Н | <input type="text" value="14.2 - 1.44*sin(t)"/>  |
| Закон изменения силы F2, Н | <input type="text" value="14.9 - 1.52*sin(t)"/>  |
| Закон изменения силы F3, Н | <input type="text" value="0.853*sin(t) - 8.37"/> |

**Вычислить**

**Построить графики координат**      **Построить графики сил**

Рисунок 4.16 – Окно «Прямая задача» с вычислениями

По нажатию на кнопку «Построить графики сил» появляется окно, представленное на рисунке 4.17. В данном окне, строится сразу три графика, которые визуально отражают законы изменения сил, действующих со стороны штоков.



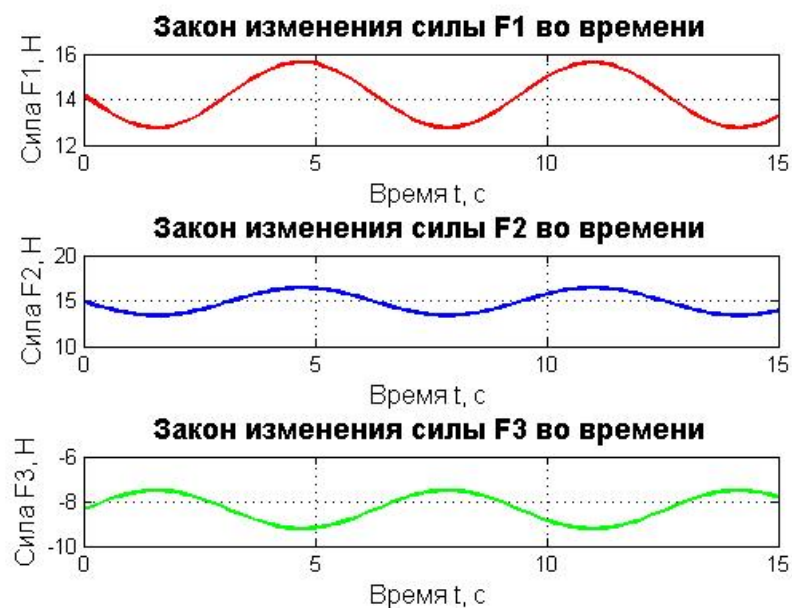


Рисунок 4.17 - Окно графиков изменения сил

По нажатию на кнопку «Построить графики координат» появляется окно, представленное на рисунке 4.18. В данном окне строится сразу три графика, которые визуальнo отражают законы изменения координат  $X_c$ ,  $Y_c$ ,  $Z_c$ .

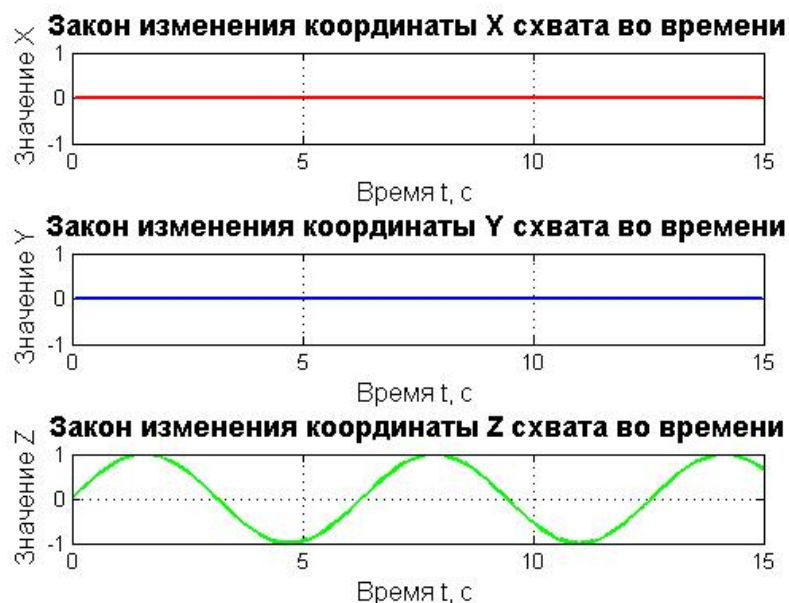


Рисунок 4.18 – Окно графиков изменения координат

В приложении предусмотрена проверка некорректного ввода данных, а именно массы. Масса – величина, которая не может быть отрицательной. В случае ввода массы меньше или равной 0 будет появляться окно с ошибкой, представленное на рисунке 4.19. После нажатия «ОК» необходимо повторить попытку ввода параметров.

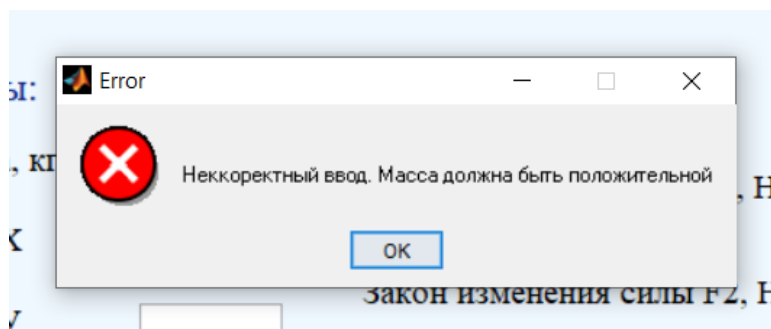


Рисунок 4.19 – Окно ошибки

Листинг кода для приложения решения «прямой задачи динамики» с комментариями приведен в *приложении 5*.

#### 4.2.2 Алгоритм работы окна «Обратная задача динамики» приложения

При решении обратной задачи динамики открывается окно «Обратная задача динамики» (рисунок 4.20).



Рисунок 4.20 - Окно «Обратная задача»

Для выполнения вычисления необходимо ввести начальные параметры, а точнее: массу подвижного основания -  $m$ , законы изменения сил, действующих на основание, во времени –  $F_1(t)$ ,  $F_2(t)$ ,  $F_3(t)$ . На рисунке 4.21 приведен пример корректного ввода данных в поля динамического текста.

Рисунок 4.21 – Пример корректного ввода данных

После нажатия на кнопку «Вычислить» происходит выполнение программы, решающей обратную задачу динамики, а именно вычисление законов изменения координат во времени. Математически решение обратной задачи подразумевает решение дифференциального уравнения 2-го порядка. Здесь возникает проблема не единственности решения, т.е. уравнение имеет бесконечное множество решений в зависимости от начальных условий (далее по тексту НУ). При чем НУ по перемещению и по скорости подразумевают задание значение координат в момент времени 0 и первой производной в момент времени 0. Из-за множественности решений можем видеть не совпадение решений прямой и обратной задачи динамики в примере расчета, представленного на рисунке 4.22.

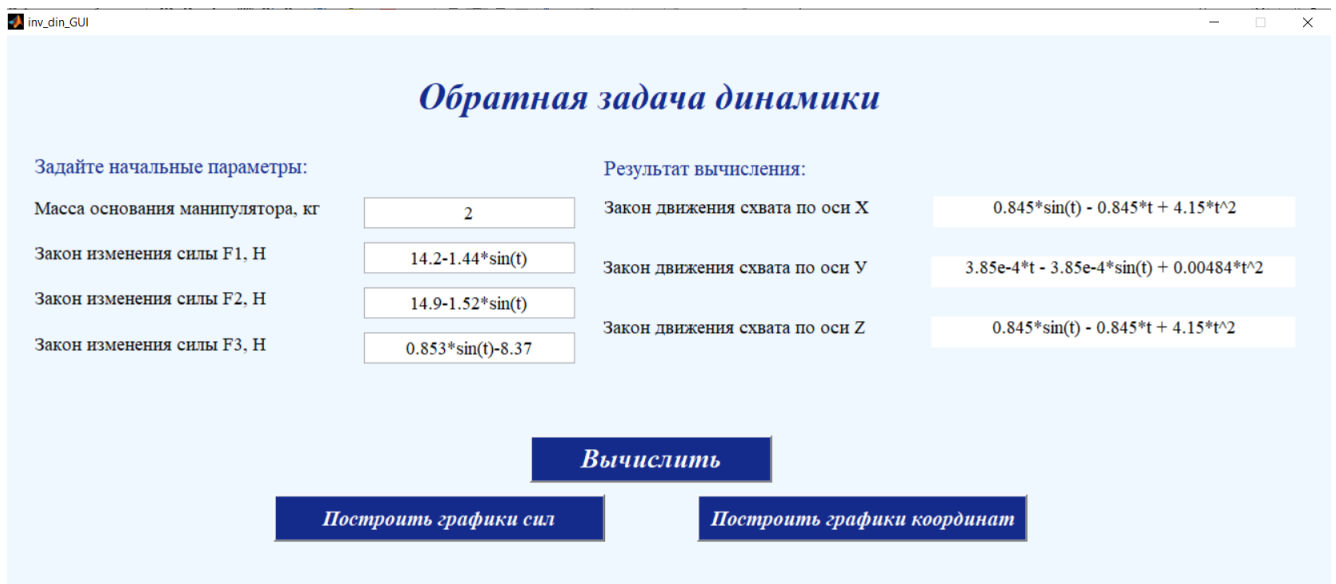


Рисунок 4.22 - Окно «Обратная задача» с вычислениями.

В интерфейсе предусмотрено построение графиков для визуального отображения законов изменения сил и координат во времени.

По нажатию на кнопку «Построить графики координат» появляется окно, представленное на рисунке 4.23. В данном окне строится сразу три графика, которые визуально отражают законы изменения координат  $X_c$ ,  $Y_c$ ,  $Z_c$ .

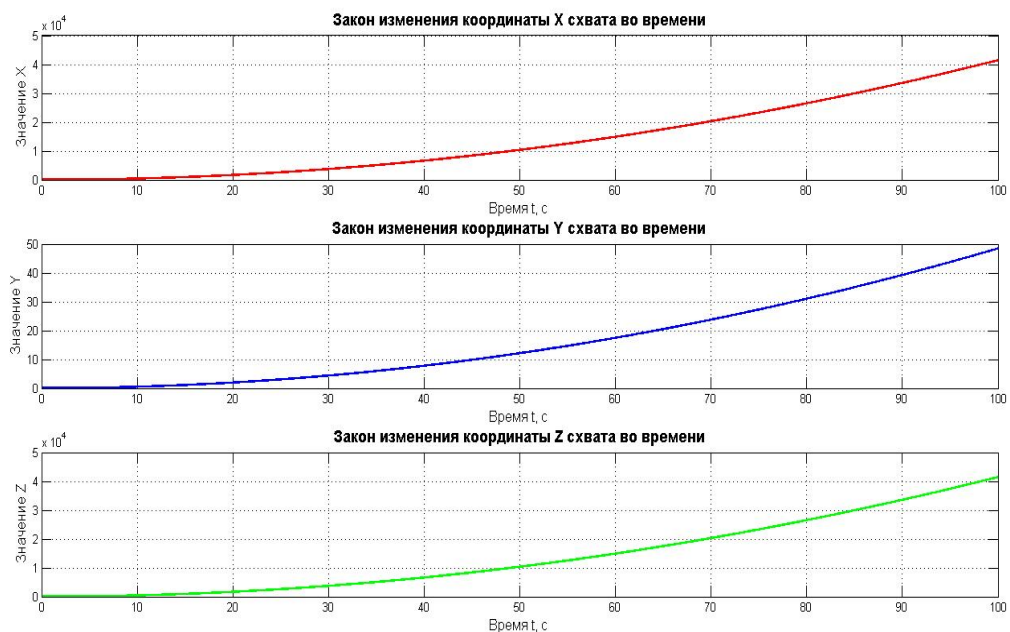


Рисунок 4.23 – Окно графиков изменения координат.

По нажатию на кнопку «Построить графики сил» появляется окно, представленное на рисунке 4.24. В данном окне строится сразу три графика, которые визуально отражают законы изменения сил, действующих со стороны штоков.

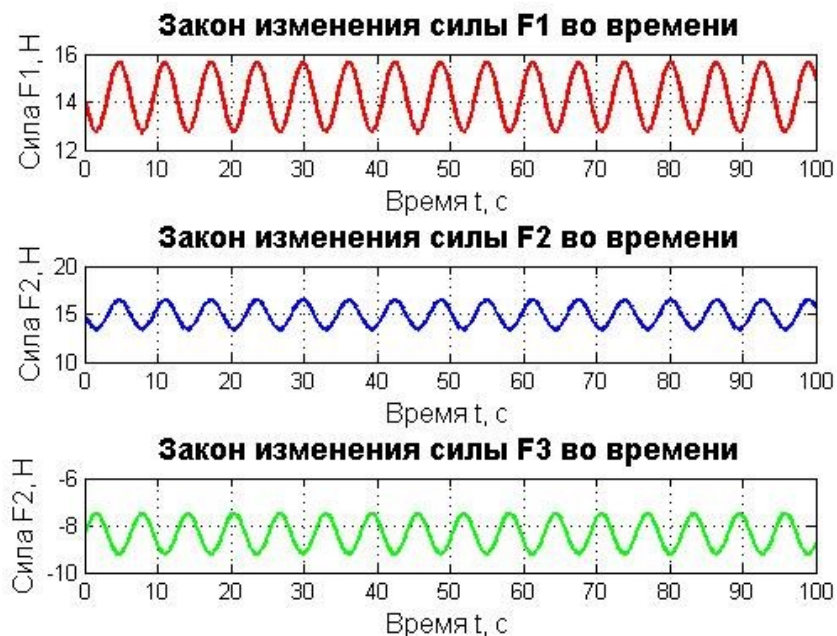


Рисунок 4.24 – Окно графиков изменения сил.

В приложении для решения обратной задачи также предусмотрена проверка некорректного ввода данных, а именно массы. В случае ввода массы меньше или равной 0 будет появляться окно с ошибкой, представленное на рисунке 4.25. После нажатия «ОК» необходимо повторить попытку ввода параметров.

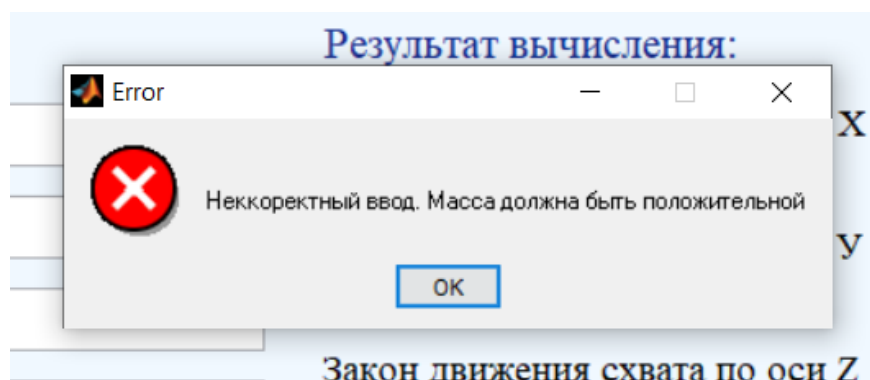


Рисунок 4.25 – Окно ошибки.

Листинг кода для приложения решения «обратной задачи динамики» с комментариями приведен в *приложении 6*.

## 5 ЗАКЛЮЧЕНИЕ

Целью текущей работы является разработка математической модели манипулятора с параллельной кинематикой.

В ходе выполнения работы был проведен обзор механизмов с подобной структурой, построена модель манипулятора типа «трипод», получены уравнения связи для кинематики и динамики движения, разработано пользовательское приложение.

Для кинематики движения получены решения прямой и обратной задач кинематики, найдена зона достижимости манипулятора и краткий анализ полученных результатов.

Для динамики движения также были получены решения прямой и обратной задач динамики, проведен краткий анализ полученных характеристик при частных случаях работы «трипода».

Для простоты поиска решений всех этих задач и их анализа разработано приложение с пользовательским интерфейсом с помощью среды разработки MATLAB 6 [4]. Описан алгоритм работы каждого пункта и проиллюстрирован рисунками и графиками.

## 6 СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ

1. Борисов О.И., Громов В.С., Пыркин А.А., Методы управления робототехническими приложениями. Учебное пособие. — СПб.: Университет ИТМО, 2016. — 108 с.
2. А.С. Климчик Р.И. Гомолицкий Ф.В. Фурман К.И. Сёмкин, «Разработка управляющих программ промышленных роботов». Курс лекций. Минск 2008 г.
3. А.М. Грибков, Д.В. Шилин Математическое моделирование манипулятора типа трипод на базе бесштоковых пневматических приводов, ж. «Известия высших учебных заведений. Машиностроение», №9, 2011, С.3-10
4. Дьяконов В. MATLAB 6: учебный курс СПб.: Питер, 2001. - 592с.
5. Дмитрий Васильев Трипод EXPT - манипулятор высшей категории, ж. «CONTROL ENGINEERING РОССИЯ» #1 (55), 2015 С.48-50
6. Каганов Ю.Т., Карпенко А.П. Математическое моделирование кинематики и динамики робота-манипулятора типа «хобот». 1. Математические модели секции манипулятора, как механизма параллельной кинематики типа «трипод» / Наука и образование: электронное научно-техническое издание. 2009. № 10. С. 4-14.
7. Stewart, D. A platform with six degrees of freedom / D. Stewart // Proceedings of the Institution of mechanical engineers, 1965. - V. 180, pt. 1, № 15. - P. 371-385.
8. Pollard W.L.G., "Spray painting machine," US Patent No. 2,213,108, August 26, 1940.
9. А.с. 558788 СССР, МКИ В25J 1/02. Манипулятор / Данилевский В.Н. //Открытия. Изобретения. 1977. - № 19. - С.35-36.
10. OKUMA PM-600 Vertical Machining Centers [Электронный ресурс]: Вертикальные обрабатывающие центры. - Режим доступа: <http://www.okuma.com/pm-600>.
11. Olazagoitia J.L., Wyatt S. New PKM Tricept T9000 and its application to flexible manufacturing at Aerospace Industry / SAE International, Paper, 2007.



12. Робот-манипулятор FlexPicker IRB 360 компании AAB  
www.abb.com/robotics [Электронный ресурс] - Режим доступа:  
[http://www.roboticturnkeysolutions.com/robots/abb/datasheet/IRB\\_360.pdf](http://www.roboticturnkeysolutions.com/robots/abb/datasheet/IRB_360.pdf)
13. Робот серии M-1 компании FANUC www.fanuc.eu/ru/ru/роботы  
[Электронный ресурс] - Режим доступа:  
<https://www.fanuc.eu/ru/ru/роботы/страница-фильтра-роботов/серия-m1>
14. Роботы с параллельной кинематикой EXPT [Электронный ресурс] -  
Режим доступа:  
[https://www.festo.com/cat/ru\\_ru/data/doc\\_RU/PDF/RU/EXPT\\_RU.PDF](https://www.festo.com/cat/ru_ru/data/doc_RU/PDF/RU/EXPT_RU.PDF)
15. Мэтьюз Д, Финк К. Численные методы. Использование MATLAB. М.:  
Издательский дом «Вильямс», 2001, 272 с
16. Колюбин С.А. Динамика робототехнических систем. Учебное пособие.  
— СПб.: Университет ИТМО, 2017. — 117 с.

## 7 ПРИЛОЖЕНИЕ

### Приложение 1. Организация главной программы приложения

Листинг главной программы “TmainGUI.m”, которая объединила в себе два раздела решаемых задач и пять пунктов расчетов.

```
function varargout = TmainGUI(varargin)
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',  gui_Singleton, ...
                  'gui_OpeningFcn', @TmainGUI_OpeningFcn, ...
                  'gui_OutputFcn',  @TmainGUI_OutputFcn, ...
                  'gui_LayoutFcn',  [] , ...
                  'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargin
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
function TmainGUI_OpeningFcn(hObject, eventdata, handles, varargin)
handles.output = hObject;
guidata(hObject, handles);
function varargout = TmainGUI_OutputFcn(hObject, eventdata, handles)
varargout{1} = handles.output;
function pb1_Callback(hObject, eventdata, handles)
TdirGUI;
function pb2_Callback(hObject, eventdata, handles)
INVGUI;
function pb3_Callback(hObject, eventdata, handles)
ZonaGUI;
function pb4_Callback(hObject, eventdata, handles)
din_GUI;
function pb5_Callback(hObject, eventdata, handles)
inv_din_GUI;
```

## Приложение 2. Организация решения задачи прямой кинематики

Листинг программы “TdirGUI.m”, решающей прямую задачу кинематики:

```
function varargout = TdirGUI(varargin)
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',   gui_Singleton, ...
                  'gui_OpeningFcn',   @TdirGUI_OpeningFcn, ...
                  'gui_OutputFcn',    @TdirGUI_OutputFcn, ...
                  'gui_LayoutFcn',    [], ...
                  'gui_Callback',     []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end
if nargin
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
function TdirGUI_OpeningFcn(hObject, eventdata, handles, varargin)
handles.output = hObject;
guidata(hObject, handles);
function varargout = TdirGUI_OutputFcn(hObject, eventdata, handles)
varargout{1} = handles.output;
function e1_Callback(hObject, eventdata, handles)
global L;
L=str2double(get(handles.e1, 'string'));
if L<=0
    msgbox('Некорректный ввод. Длина основания должна быть положительной',
'Error', 'error');
    set(handles.e1, 'String', '');
end;
function e1_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject, 'BackgroundColor'),
get(0, 'defaultUicontrolBackgroundColor'))
    set(hObject, 'BackgroundColor', 'white');
end
function e2_Callback(hObject, eventdata, handles)
global l;
l=str2double(get(handles.e2, 'string'));
if l<=0
    msgbox('Некорректный ввод. Длина тяги должна быть положительной', 'Error',
'error');
    set(handles.e2, 'String', '');
end;
function e2_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject, 'BackgroundColor'),
get(0, 'defaultUicontrolBackgroundColor'))
    set(hObject, 'BackgroundColor', 'white');
end
function e3_Callback(hObject, eventdata, handles)
global S;
S=str2double(get(handles.e3, 'string'));
if S<=0
    msgbox('Некорректный ввод. Длина хода должна быть положительной', 'Error',
'error');
    set(handles.e3, 'String', '');
end;
```

```

function e3_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
function e4_Callback(hObject, eventdata, handles)
global Sstart;
Sstart=str2double(get(handles.e4,'string'));
if Sstart<=0
    msgbox('Некорректный ввод. Начало хода должно быть положительным', 'Error',
'error');
    set(handles.e4,'String','');
end;
function e4_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
function e5_Callback(hObject, eventdata, handles)
global Send;
Send=str2double(get(handles.e5,'string'));
if Send<=0
    msgbox('Некорректный ввод. Конец хода должно быть положительным', 'Error',
'error');
    set(handles.e5,'String','');
end;
function e5_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
function e6_Callback(hObject, eventdata, handles)
global a;
a=str2double(get(handles.e6,'string'));
if a<=0
    msgbox('Некорректный ввод. Радиус основания должен быть положительным',
'Error', 'error');
    set(handles.e6,'String','');
end;
function e6_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
function e7_Callback(hObject, eventdata, handles)
global h;
h=str2double(get(handles.e7,'string'));
if (h<=0) || (h>=1)
    msgbox('Некорректный ввод. Величина шага должна лежать в пределах от 1 до 0',
'Error', 'error');
    set(handles.e7,'String','');
end;
function e7_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
function e8_Callback(hObject, eventdata, handles)
global e;
e=str2double(get(handles.e8,'string'));
if ((e<=0) || (e>=1))
    msgbox('Некорректный ввод. Величина погрешности должна лежать в пределах от 1
до 0', 'Error', 'error');

```

```

        set(handles.e8, 'String', '');
    end;
    function e8_CreateFcn(hObject, eventdata, handles)
    if ispc && isequal(get(hObject, 'BackgroundColor'),
    get(0, 'defaultUicontrolBackgroundColor'))
        set(hObject, 'BackgroundColor', 'white');
    end
    function e9_Callback(hObject, eventdata, handles)
    global S1;
    S1=str2double(get(handles.e9, 'string'));
    if S1<=0
        msgbox('Неккоректный ввод. Положение 1-ой каретки должно быть положительным',
        'Error', 'error');
        set(handles.e9, 'String', '');
    end;
    function e9_CreateFcn(hObject, eventdata, handles)
    if ispc && isequal(get(hObject, 'BackgroundColor'),
    get(0, 'defaultUicontrolBackgroundColor'))
        set(hObject, 'BackgroundColor', 'white');
    end
    function e10_Callback(hObject, eventdata, handles)
    global S2;
    S2=str2double(get(handles.e10, 'string'));
    if S2<=0
        msgbox('Неккоректный ввод. Положение 2-ой каретки должно быть положительным',
        'Error', 'error');
        set(handles.e10, 'String', '');
    end;
    function e10_CreateFcn(hObject, eventdata, handles)
    if ispc && isequal(get(hObject, 'BackgroundColor'),
    get(0, 'defaultUicontrolBackgroundColor'))
        set(hObject, 'BackgroundColor', 'white');
    end
    function e11_Callback(hObject, eventdata, handles)
    global S3;
    S3=str2double(get(handles.e11, 'string'));
    if S3<=0
        msgbox('Неккоректный ввод. Положение 3-ой каретки должно быть положительным',
        'Error', 'error');
        set(handles.e11, 'String', '');
    end;
    function e11_CreateFcn(hObject, eventdata, handles)
    if ispc && isequal(get(hObject, 'BackgroundColor'),
    get(0, 'defaultUicontrolBackgroundColor'))
        set(hObject, 'BackgroundColor', 'white');
    end
    function pushbutton1_Callback(hObject, eventdata, handles)
    %Начальные параметры трипода
    global L l S Sstart Send a h e S1 S2 S3;
    L = str2double(get(handles.e1, 'string'));
    l = str2double(get(handles.e2, 'string'));
    S = str2double(get(handles.e3, 'string'));
    Sstart = str2double(get(handles.e4, 'string'));
    Send = str2double(get(handles.e5, 'string'));
    a = str2double(get(handles.e6, 'string'));
    h = str2double(get(handles.e7, 'string'));
    e = str2double(get(handles.e8, 'string'));
    S1 = str2double(get(handles.e9, 'string'));
    S2 = str2double(get(handles.e10, 'string'));
    S3 = str2double(get(handles.e11, 'string'));
    %Косинус и синус угла между цилиндром и неподвижным основанием
    cos_g = (sqrt(3)*L)/(3*S);
    sin_g = sqrt(1-cos_g^2);

```

```

%Расчет координат кареток
S1 = S1+Sstart;
S2 = S2+Sstart;
S3 = S3+Sstart;
%Вводим 1-ое приближение
X0 = zeros(12,1);
%Вводим 2-ое приближение
X1 = ones(12,1);
% Замена констант
c1 = (S1*sin_g)/l;B=(S2*sin_g)/l;C=(S3*sin_g)/l;
c2 = -(3*a/2)-(S2*cos_g/2)-S1*cos_g+sqrt(3)*L/2;
c3 = -(3*a/2)-(S3*cos_g/2)-S1*cos_g+sqrt(3)*L/2;
c4 = sqrt(3)*S2*cos_g/2+sqrt(3)*a/2-L/2;
c5 = -sqrt(3)*S3*cos_g/2-sqrt(3)*a/2+L/2;
%Метод Ньютона с точностью e
while (abs(X1-X0) > e)
%Присвоение первому приближению второе приближение
X0 = X1;
%Система уравнений
F = zeros(12,1);
%Обозначение неизвестных
x1 = X0(1,1);x2=X0(2,1);x3=X0(3,1);x4=X0(4,1);x5=X0(5,1);x6=X0(6,1);y1=X0(7,1);
y2 = X0(8,1);y3=X0(9,1);y4=X0(10,1);y5=X0(11,1);y6=X0(12,1);
%Уравнения системы
f1 = x1+c1-x2-B;
f2 = x2+B-x3-C;
f3 = (-1*y2*y5)/2-(sqrt(3)*1*y2*x5)/2-1*y1*y4+c2;
f4 = -(sqrt(3))*1*y3*x6)/2+(1*y3*y6)/2-1*y1*y4+c3;
f5 = (sqrt(3)*1*y2*y5)/2-(1*y2*x5)/2-1*y1*x4+c4;
f6 = -(sqrt(3))*1*y3*y6)/2-(1*y3*x6)/2-1*y1*x4+c5;
f7 = x1^2+y1^2-1;
f8 = x2^2+y2^2-1;
f9 = x3^2+y3^2-1;
f10 = x4^2+y4^2-1;
f11 = x5^2+y5^2-1;
f12 = x6^2+y6^2-1;
%Матрица значений уравнений системы с подстановкой первого приближения
F(1,1) = f1; F(2,1) = f2; F(3,1) = f3; F(4,1) = f4; F(5,1) = f5; F(6,1) = f6;
F(7,1) = f7; F(8,1) = f8; F(9,1) = f9; F(10,1) = f10; F(11,1) = f11; F(12,1) =
f12;
%матрица Якоби
J = zeros(12,12);
%Нахождение производных
f1x1 = 1; f1x2 = -1;
f2x2 = 1; f2x3 = -1;
f3x5 = (-sqrt(3)*1*y2)/2; f3y1 = -1*y4; f3y2 = (-1*y5)/2-(sqrt(3)*1*x5)/2; f3y4 =
-1*y1; f3y5 = (-1*y2)/2;
f4x6 = (-sqrt(3)*1*y3)/2; f4y1 = -1*y4; f4y3 = (-sqrt(3)*1*x6)/2+(1*y6)/2; f4y4 =
-1*y1; f4y6 = (1*y3)/2;
f5x4 = -1*y1; f5x5 = (-1*y2)/2; f5y1 = -1*x4; f5y2 = (sqrt(3)*1*y5)/2-(1*x5)/2;
f5y5 = (sqrt(3)*1*y2)/2;
f6x4 = -1*y1; f6x6 = (-1*y3)/2; f6y1 = -1*x4; f6y3 = (-sqrt(3))*1*y6)/2-(1*x6)/2;
f6y6 = (-sqrt(3))*1*y3)/2;
f7x1 = 2*x1; f7y1 = 2*y1;
f8x2 = 2*x2; f8y2 = 2*y2;
f9x3 = 2*x3;f9y3 = 2*y3;
f10x4 = 2*x4; f10y4 = 2*y4;
f11x5 = 2*x5; f11y5 = 2*y5;
f12x6 = 2*x6; f12y6 = 2*y6;
%Формирование матрицы Якоби
J(1,1) = f1x1; J(1,2) = f1x2;
J(2,2) = f2x2; J(2,3) = f2x3;
J(3,5) = f3x5; J(3,7) = f3y1; J(3,8) = f3y2; J(3,10) = f3y4; J(3,11) = f3y5;

```

```

J(4,6) = f4x6; J(4,7) = f4y1; J(4,9) = f4y3; J(4,10) = f4y4; J(4,12) = f4y6;
J(5,4) = f5x4; J(5,5) = f5x5; J(5,7) = f5y1; J(5,8) = f5y2; J(5,11) = f5y5;
J(6,4) = f6x4; J(6,6) = f6x6; J(6,7) = f6y1; J(6,9) = f6y3; J(6,12) = f6y6;
J(7,1) = f7x1; J(7,7) = f7y1;
J(8,2) = f8x2; J(8,8) = f8y2;
J(9,3) = f9x3; J(9,9) = f9y3;
J(10,4) = f10x4; J(10,10) = f10y4;
J(11,5) = f11x5; J(11,11) = f11y5;
J(12,6) = f12x6; J(12,12) = f12y6;
%Нахождение обратной матрицы
Obr=J^(-1);
%Нахождение второго приближения
X1=X0-Obr*F;
end
%Нахождение координат точек неподвижного основания манипулятора
xA1= S1*cos_g+l*X1(7,1)*X1(10,1)+a;
yA1=l*X1(7,1)*X1(4,1);
zA1=S1*sin_g+l*X1(1,1);
xA2=-S2*cos_g*sin(pi/6)-l*X1(8,1)*(sin(pi/6)*X1(11,1)+cos(pi/6)*X1(5,1))-
a*sin(pi/6);
yA2=S2*cos_g*cos(pi/6)+l*X1(8,1)*(cos(pi/6)*X1(11,1)-
sin(pi/6)*X1(5,1))+a*cos(pi/6);
zA2=S2*sin_g+l*X1(2,1);
xA3=-S3*cos_g*sin(pi/6)-l*X1(9,1)*(X1(6,1)*cos(pi/6)-X1(12,1)*sin(pi/6))-
a*sin(pi/6);
yA3=-S3*cos_g*cos(pi/6)-l*X1(9,1)*(X1(12,1)*cos(pi/6)+sin(pi/6)*X1(6,1))-
a*cos(pi/6);
zA3=S3*sin_g+l*X1(3,1);
%Расчет координат центра подвижной платформы
xC=-(xA1+xA2+xA3)/3;
yC=-(yA1+yA2+yA3)/3;
zC=-(zA1+zA2+zA3)/3;
%Вывод значений координат захвата
set(handles.t1,'string',num2str(xC, '% 10.3f'));
set(handles.t2,'string',num2str(yC, '% 10.3f'));
set(handles.t3,'string',num2str(zC, '% 10.3f'));

```

## Приложение 3. Организация решения задачи обратной кинематики

Листинг программы “INVGUI.m”, решающей обратную задачу кинематики:

```
function varargout = INVGUI(varargin)
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',   gui_Singleton, ...
                  'gui_OpeningFcn',   @INVGUI_OpeningFcn, ...
                  'gui_OutputFcn',   @INVGUI_OutputFcn, ...
                  'gui_LayoutFcn',    [], ...
                  'gui_Callback',     []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end
if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
function INVGUI_OpeningFcn(hObject, eventdata, handles, varargin)
handles.output = hObject;
guidata(hObject, handles);
function varargout = INVGUI_OutputFcn(hObject, eventdata, handles)
varargout{1} = handles.output;
function ed1_Callback(hObject, eventdata, handles)
global L;
L=str2double(get(handles.ed1, 'string'));
if L<=0
    msgbox('Некорректный ввод. Длина основания должна быть положительной',
'Error', 'error');
    set(handles.ed1, 'String', '');
end;
function ed1_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject, 'BackgroundColor'),
get(0, 'defaultUicontrolBackgroundColor'))
    set(hObject, 'BackgroundColor', 'white');
end
function ed2_Callback(hObject, eventdata, handles)
global l;
l=str2double(get(handles.ed2, 'string'));
if l<=0
    msgbox('Некорректный ввод. Длина тяги должна быть положительной', 'Error',
'error');
    set(handles.ed2, 'String', '');
end;
function ed2_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject, 'BackgroundColor'),
get(0, 'defaultUicontrolBackgroundColor'))
    set(hObject, 'BackgroundColor', 'white');
end
function ed3_Callback(hObject, eventdata, handles)
global S;
S=str2double(get(handles.ed3, 'string'));
if S<=0
    msgbox('Некорректный ввод. Длина хода должна быть положительной', 'Error',
'error');
    set(handles.ed3, 'String', '');
end;
```



```

function ed3_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
function ed4_Callback(hObject, eventdata, handles)
global Sstart;
Sstart=str2double(get(handles.ed4,'string'));
if Sstart<=0
    msgbox('Некорректный ввод. Начало хода должно быть положительным', 'Error',
'error');
    set(handles.ed4,'String','');
end;
function ed4_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
function ed5_Callback(hObject, eventdata, handles)
global Send;
Send=str2double(get(handles.ed5,'string'));
if Send<=0
    msgbox('Некорректный ввод. Конец хода должно быть положительным', 'Error',
'error');
    set(handles.ed5,'String','');
end;
function ed5_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end;
function ed6_Callback(hObject, eventdata, handles)
global a;
a=str2double(get(handles.ed6,'string'));
if a<=0
    msgbox('Некорректный ввод. Радиус основания должен быть положительным',
'Error', 'error');
    set(handles.ed6,'String','');
end;
function ed6_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
function ed7_Callback(hObject, eventdata, handles)
global h;
h=str2double(get(handles.ed7,'string'));
if (h<=0) || (h>=1)
    msgbox('Некорректный ввод. Величина шага должна лежать в пределах от 1 до 0',
'Error', 'error');
    set(handles.ed7,'String','');
end;
function ed7_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
function ed8_Callback(hObject, eventdata, handles)
global e;
e=str2double(get(handles.ed8,'string'));
if ((e<=0) || (e>=1))
    msgbox('Некорректный ввод. Величина погрешности должна лежать в пределах от 1
до 0', 'Error', 'error');

```

```

        set(handles.ed8, 'String', '');
    end;
    function ed9_Callback(hObject, eventdata, handles)
    function ed9_CreateFcn(hObject, eventdata, handles)
    if ispc && isequal(get(hObject, 'BackgroundColor'),
    get(0, 'defaultUicontrolBackgroundColor'))
        set(hObject, 'BackgroundColor', 'white');
    end
    function ed10_Callback(hObject, eventdata, handles)
    function ed10_CreateFcn(hObject, eventdata, handles)
    if ispc && isequal(get(hObject, 'BackgroundColor'),
    get(0, 'defaultUicontrolBackgroundColor'))
        set(hObject, 'BackgroundColor', 'white');
    end
    function ed11_Callback(hObject, eventdata, handles)
    function ed11_CreateFcn(hObject, eventdata, handles)
    if ispc && isequal(get(hObject, 'BackgroundColor'),
    get(0, 'defaultUicontrolBackgroundColor'))
        set(hObject, 'BackgroundColor', 'white');
    end
    function pushbutton2_Callback(hObject, eventdata, handles)
    %Начальные параметры трипода
    global L l S Sstart Send a h e;
    L = str2double(get(handles.ed1, 'string'));
    l = str2double(get(handles.ed2, 'string'));
    S = str2double(get(handles.ed3, 'string'));
    Sstart = str2double(get(handles.ed4, 'string'));
    Send = str2double(get(handles.ed5, 'string'));
    a = str2double(get(handles.ed6, 'string'));
    h = str2double(get(handles.ed7, 'string'));
    e = str2double(get(handles.ed8, 'string'));
    Xc =str2double(get(handles.ed9, 'string'));
    Yc= str2double(get(handles.ed10, 'string'));
    Zc = str2double(get(handles.ed11, 'string'));
    %Косинус и синус угла между цилиндром и неподвижным основанием
    cos_g = (sqrt(3)*L)/(3*S);
    sin_g = sqrt(1-cos_g^2);

    %Задаем 1-ое приближение
    X0 = zeros(15,1);
    %Задаем 2-ое приближение
    X1 = ones(15,1);

    %Метод Ньютона с точностью e
    while (abs(X1-X0) > e)

    %Присвоение первому приближению второе приближение
    X0 = X1;
    %Система уравнений
    F = zeros(15,1);
    %Вводим переобозначения неизвестных
    x1 = X0(1,1); x2 = X0(2,1); x3 = X0(3,1); x4 = X0(4,1); x5 = X0(5,1); x6 =
    X0(6,1);
    y1 = X0(7,1); y2 = X0(8,1); y3 = X0(9,1); y4 = X0(10,1); y5 = X0(11,1); y6 =
    X0(12,1);
    %Переобозначение неизвестных, которые необходимо найти(положение кареток)
    x7 = X0(13,1); x8 = X0(14,1); x9 = X0(15,1);

    %Уравнения системы
    f1=x1+(x7*sin_g)/l-x2-(x8*sin_g)/l;
    f2=x2+(x8*sin_g)/l-x3-(x9*sin_g)/l;
    f3=(-1*y2*y5)/2-(sqrt(3)*l*y2*x5)/2-1*y1*y4-(3*a/2)-(x8*cos_g/2)-
    x7*cos_g+sqrt(3)*L/2;

```

```

f4=(-(sqrt(3))*1*y3*x6)/2+(1*y3*y6)/2-1*y1*y4-(3*a/2)-(x9*cos_g/2)-
x7*cos_g+sqrt(3)*L/2;
f5=(sqrt(3)*1*y2*y5)/2-(1*y2*x5)/2-1*y1*x4+sqrt(3)*x8*cos_g/2+sqrt(3)*a/2-L/2;
f6=(-(sqrt(3))*1*y3*y6)/2-(1*y3*x6)/2-1*y1*x4-sqrt(3)*x9*cos_g/2-sqrt(3)*a/2+L/2;
f7=Xc+(x7*cos_g+1*y1*y4+a-x8*cos_g/2-1*y2*(y5/2+sqrt(3)*x5/2)-a/2-x9*cos_g/2-
1*y3*(sqrt(3)*x6/2-y6/2)-a/2)/3;
f8=Yc+(1*y1*x4+x8*cos_g*sqrt(3)/2+1*y2*(sqrt(3)*y5/2-x5/2)+a*sqrt(3)/2-
x9*cos_g*sqrt(3)/2-1*y3*(sqrt(3)*y6/2+x6/2)-a*sqrt(3)/2)/3;
f9=Zc+(x7*sin_g+1*x1+x8*sin_g+1*x2+x9*sin_g+1*x3)/3;
f10=x1^2+y1^2-1;
f11=x2^2+y2^2-1;
f12=x3^2+y3^2-1;
f13=x4^2+y4^2-1;
f14=x5^2+y5^2-1;
f15=x6^2+y6^2-1;

```

**%Матрица значений уравнений системы с подстановкой первого приближения**

```

F(1,1) = f1; F(2,1) = f2; F(3,1) = f3; F(4,1) = f4; F(5,1) = f5; F(6,1) = f6;
F(7,1) = f7; F(8,1) = f8;
F(9,1) = f9; F(10,1) = f10; F(11,1) = f11; F(12,1) = f12; F(13,1)=f13; F(14,1) =
f14; F(15,1) = f15;

```

**%матрица Якоби**

```
J = zeros(15,15);
```

**%Нахождение проиизводных**

```

f1x1 = 1; f1x2 = -1; f1x7 = sin_g/1; f1x8 = -sin_g/1;
f2x2 = 1; f2x3 = -1; f2x8 = sin_g/1; f2x9 = -sin_g/1;
f3x5 = (-sqrt(3)*1*y2)/2; f3y1 = -1*y4; f3y2 = (-1*y5)/2-(sqrt(3)*1*x5)/2;
f3y4 = -1*y1; f3y5 = (-1*y2)/2; f3x7 = -cos_g; f3x8 = -cos_g/2;
f4x6 = (-sqrt(3)*1*y3)/2; f4x7 = -cos_g; f4x9 = -cos_g/2; f4y1 = -1*y4;
f4y3 = (-sqrt(3)*1*x6)/2+(1*y6)/2; f4y4 = -1*y1; f4y6 = (1*y3)/2;
f5x4 = -1*y1; f5x5 = (-1*y2)/2; f5x8 = sqrt(3)*cos_g/2; f5y1 = -1*x4;
f5y2 = (sqrt(3)*1*y5)/2-(1*x5)/2; f5y5 = (sqrt(3)*1*y2)/2;
f6x4 = -1*y1; f6x6 = (-1*y3)/2; f6x9 = -sqrt(3)*cos_g/2; f6y1 = -1*x4;
f6y3 = (-sqrt(3))*1*y6)/2-(1*x6)/2; f6y6 = (-sqrt(3))*1*y3)/2;
f10x1 = 2*x1; f10y1 = 2*y1;
f11x2 = 2*x2; f11y2 = 2*y2;
f12x3 = 2*x3; f12y3 =2*y3;
f13x4 = 2*x4; f13y4 = 2*y4;
f14x5 = 2*x5; f14y5 = 2*y5;
f15x6 = 2*x6; f15y6 = 2*y6;
f7x5 = -sqrt(3)*1*y2/6; f7x6 = -sqrt(3)*1*y3/6; f7x7 = cos_g/3; f7x8 = -cos_g/6;
f7x9 = -cos_g/6; f7y1 = 1*y4/3; f7y2 = -1*(y5/2+sqrt(3)*x5/2)/3;
f7y3 = -1*(-y6/2+sqrt(3)*x6/2)/3; f7y4 = 1*y1/3; f7y5 = -1*y2/6; f7y6 = 1*y3/6;
f8x4 = 1*y1/3; f8x5 = 1*y2/6; f8x6 = -1*y3/6; f8x8 = sqrt(3)*cos_g/2;
f8x9 = -sqrt(3)*cos_g/2; f8y1 = 1*x4/3; f8y2 = 1*(-x5/2+sqrt(3)*y5/2)/3;
f8y3 = -1*(x6/2+sqrt(3)*y6/2)/3; f8y5 = sqrt(3)*1*y2/6; f8y6 = -sqrt(3)*1*y3/6;
f9x1 = 1/3; f9x2 =1/3; f9x3 = 1/3; f9x7 = sin_g/3; f9x8 = sin_g/3; f9x9 = sin_g/3;

```

**%Составление матрицы Якоби**

```

J(1,1)=f1x1; J(1,2)=f1x2; J(1,13)=f1x7; J(1,14)=f1x8;
J(2,2)=f2x2; J(2,3)=f2x3; J(2,14)=f2x8; J(2,15)=f2x9;
J(3,5)=f3x5; J(3,8)=f3y2; J(3,7)=f3y1; J(3,10)=f3y4; J(3,11)=f3y5; J(3,13)=f3x7; J(3,14)
=f3x8;
J(4,6)=f4x6; J(4,7)=f4y1; J(4,9)=f4y3; J(4,10)=f4y4; J(4,12)=f4y6; J(4,13)=f4x7; J(4,15)
=f4x9;
J(5,4)=f5x4; J(5,5)=f5x5; J(5,7)=f5y1; J(5,8)=f5y2; J(5,11)=f5y5; J(5,14)=f5x8;
J(6,4)=f6x4; J(6,6)=f6x6; J(6,7)=f6y1; J(6,9)=f6y3; J(6,12)=f6y6; J(6,15)=f6x9;
J(7,5)=f7x5; J(7,6)=f7x6; J(7,7)=f7y1; J(7,8)=f7y2; J(7,9)=f7y3; J(7,10)=f7y4;
J(7,11)=f7y5; J(7,12)=f7y6; J(7,13)=f7x7; J(7,14)=f7x8; J(7,15)=f7x9;
J(8,4)=f8x4; J(8,5)=f8x5; J(8,6)=f8x6; J(8,7)=f8y1; J(8,8)=f8y2;
J(8,9)=f8y3; J(8,11)=f8y5; J(8,12)=f8y6; J(8,14)=f8x8; J(8,15)=f8x9;
J(9,1)=f9x1; J(9,2)=f9x2; J(9,3)=f9x3; J(9,13)=f9x7; J(9,14)=f9x8; J(9,15)=f9x9;
J(10,1)=f10x1; J(10,7)=f10y1;

```

```

J(11,2)=f11x2;J(11,8)=f11y2;
J(12,3)=f12x3;J(12,9)=f12y3;
J(13,4)=f13x4;J(13,10)=f13y4;
J(14,5)=f14x5;J(14,11)=f14y5;
J(15,6)=f15x6;J(15,12)=f15y6;

%Обратная матрица
Obr = J^(-1);
%Нахождение второго приближения
X1 = X0-Obr*F;
end
%Полученные значения координат кареток
x7=x7-Sstart;x8=x8-Sstart;x9=x9-Sstart;
%Итоговое положение кареток привода
set(handles.tex1,'string',num2str(x7, '% 10.3f'));
set(handles.tex2,'string',num2str(x8, '% 10.3f'));
set(handles.tex3,'string',num2str(x9, '% 10.3f'));

```

## Приложение 4. Организация решения получения зоны достижимости

Листинг программы “ZonaGUI.m”, позволяющая построить область достижимости манипулятора:

```
function varargout = ZonaGUI(varargin)
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',  gui_Singleton, ...
                  'gui_OpeningFcn', @ZonaGUI_OpeningFcn, ...
                  'gui_OutputFcn',  @ZonaGUI_OutputFcn, ...
                  'gui_LayoutFcn',  [], ...
                  'gui_Callback',    []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end
if nargin
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
function ZonaGUI_OpeningFcn(hObject, eventdata, handles, varargin)
handles.output = hObject;
guidata(hObject, handles);
function varargout = ZonaGUI_OutputFcn(hObject, eventdata, handles)
varargout{1} = handles.output;
function d1_Callback(hObject, eventdata, handles)
global L;
L=str2double(get(handles.d1, 'string'));
if L<=0
    msgbox('Некорректный ввод. Длина основания должна быть положительной',
'Error', 'error');
    set(handles.d1, 'String', '');
end;
function d1_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject, 'BackgroundColor'),
get(0, 'defaultUicontrolBackgroundColor'))
    set(hObject, 'BackgroundColor', 'white');
end
function d2_Callback(hObject, eventdata, handles)
global l;
l=str2double(get(handles.d2, 'string'));
if l<=0
    msgbox('Некорректный ввод. Длина тяги должна быть положительной', 'Error',
'error');
    set(handles.d2, 'String', '');
end;
function d2_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject, 'BackgroundColor'),
get(0, 'defaultUicontrolBackgroundColor'))
    set(hObject, 'BackgroundColor', 'white');
end
function d3_Callback(hObject, eventdata, handles)
global S;
S=str2double(get(handles.d3, 'string'));
if S<=0
    msgbox('Некорректный ввод. Длина хода должна быть положительной', 'Error',
'error');
    set(handles.d3, 'String', '');
```

```

end;
function d3_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
function d4_Callback(hObject, eventdata, handles)
global Sstart;
Sstart=str2double(get(handles.d4,'string'));
if Sstart<=0
    msgbox('Некорректный ввод. Начало хода должно быть положительным', 'Error',
'error');
    set(handles.d4,'String','');
end;
function d4_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
function d5_Callback(hObject, eventdata, handles)
global Send;
Send=str2double(get(handles.d5,'string'));
if Send<=0
    msgbox('Некорректный ввод. Конец хода должно быть положительным', 'Error',
'error');
    set(handles.d5,'String','');
end;
function d5_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
function d6_Callback(hObject, eventdata, handles)
global a;
a=str2double(get(handles.d6,'string'));
if a<=0
    msgbox('Некорректный ввод. Радиус основания должен быть положительным',
'Error', 'error');
    set(handles.d6,'String','');
end;
function d6_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
function d7_Callback(hObject, eventdata, handles)
global h;
h=str2double(get(handles.d7,'string'));
if (h<=0) || (h>=1)
    msgbox('Некорректный ввод. Величина шага должна лежать в пределах от 1 до 0',
'Error', 'error');
    set(handles.d7,'String','');
end;
function d7_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
function d8_Callback(hObject, eventdata, handles)
global e;
e=str2double(get(handles.d8,'string'));
if ((e<=0) || (e>=1))

```

```

    msgbox('Некорректный ввод. Величина погрешности должна лежать в пределах от 1
до 0', 'Error', 'error');
    set(handles.d8,'String','');
end;
function d8_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
function pushbutton2_Callback(hObject, eventdata, handles)
%Исходные данные
global L l S Sstart Send a h e;
L = str2double(get(handles.d1,'string'));
l = str2double(get(handles.d2,'string'));
S = str2double(get(handles.d3,'string'));
Sstart = str2double(get(handles.d4,'string'));
Send = str2double(get(handles.d5,'string'));
a = str2double(get(handles.d6,'string'));
h = str2double(get(handles.d7,'string'));
e = str2double(get(handles.d8,'string'));
%Косинус и синус угла между цилиндром и неподвижным основанием
cos_g = (sqrt(3)*L)/(3*S);
sin_g = sqrt(1-cos_g^2);
%Текущее количество позиций матрицы LARG
i = 0;
%Создание матрицы LARG, nnn-максимальное кол-во позиций в матрице LARG
N = round(((S-Sstart-Send)/h)+1)^3);
LARG = zeros(N,3);
for s1 = Sstart:h:S-Send
    for s2 = Sstart:h:S-Send
        for s3 = Sstart:h:S-Send
            i = i+1;
%Вводим 1-ое приближение
X0 = zeros(12,1);
%Вводим 2-ое приближение
X1 = ones(12,1);
%Заменяем константы
c1 = (s1*sin_g)/l;
c2 = (s2*sin_g)/l;
c3 = (s3*sin_g)/l;
c4 = -(3*a/2)-(s2*cos_g/2)-s1*cos_g+sqrt(3)*L/2;
c5 = -(3*a/2)-(s3*cos_g/2)-s1*cos_g+sqrt(3)*L/2;
c6 = sqrt(3)*s2*cos_g/2+sqrt(3)*a/2-L/2;
c7 = -sqrt(3)*s3*cos_g/2-sqrt(3)*a/2+L/2;

%Метод Ньютона с заданной точностью e
while (abs(X1-X0) > e)
%Присвоение первому приближению второе приближение
X0 = X1;
%Система уравнений
F = zeros(12,1);
%Обозначение неизвестных
x1 = X0(1,1); x2 = X0(2,1); x3 = X0(3,1); x4 = X0(4,1); x5 = X0(5,1); x6 =
X0(6,1);
y1 = X0(7,1); y2 = X0(8,1); y3 = X0(9,1); y4 = X0(10,1); y5 = X0(11,1); y6 =
X0(12,1);
%Уравнения системы
f1 = x1+c1-x2-c2;
f2 = x2+c2-x3-c3;
f3 = (-1*y2*y5)/2-(sqrt(3)*1*y2*x5)/2-1*y1*y4+c4;
f4 = -(sqrt(3))*1*y3*x6)/2+(1*y3*y6)/2-1*y1*y4+c5;
f5 = (sqrt(3)*1*y2*y5)/2-(1*y2*x5)/2-1*y1*x4+c6;
f6 = -(sqrt(3))*1*y3*y6)/2-(1*y3*x6)/2-1*y1*x4+c7;

```

```

f7 = x1^2+y1^2-1;
f8 = x2^2+y2^2-1;
f9 = x3^2+y3^2-1;
f10 = x4^2+y4^2-1;
f11 = x5^2+y5^2-1;
f12 = x6^2+y6^2-1;
%Матрица значений уравнений системы с подстановкой первого приближения
F(1,1) = f1; F(2,1) = f2; F(3,1) = f3; F(4,1) = f4; F(5,1) = f5; F(6,1) = f6;
F(7,1) = f7; F(8,1) = f8; F(9,1) = f9; F(10,1) = f10; F(11,1) = f11; F(12,1) =
f12;
%матрица Якоби
J = zeros(12,12);
%Нахождение производных
f1x1 = 1; f1x2 = -1;
f2x2 = 1 ; f2x3 = -1;
f3x5 = (-sqrt(3)*1*y2)/2; f3y1=-1*y4; f3y2 = (-1*y5)/2-(sqrt(3)*1*x5)/2;
f3y4 = -1*y1; f3y5 = (-1*y2)/2;
f4x6 = (-sqrt(3)*1*y3)/2; f4y1 = -1*y4; f4y3 = (-sqrt(3)*1*x6)/2+(1*y6)/2;
f4y4 = -1*y1; f4y6 = (1*y3)/2;
f5x4 = -1*y1; f5x5 = (-1*y2)/2; f5y1 = -1*x4; f5y2 = (sqrt(3)*1*y5)/2-(1*x5)/2;
f5y5 = (sqrt(3)*1*y2)/2;
f6x4 = -1*y1; f6x6 = (-1*y3)/2; f6y1 = -1*x4; f6y3 = (-sqrt(3))*1*y6)/2-(1*x6)/2;
f6y6 = (-sqrt(3))*1*y3)/2;
f7x1 = 2*x1; f7y1 = 2*y1;
f8x2 = 2*x2; f8y2 = 2*y2;
f9x3 = 2*x3; f9y3 = 2*y3;
f10x4 = 2*x4; f10y4 = 2*y4;
f11x5 = 2*x5; f11y5 = 2*y5;
f12x6 = 2*x6; f12y6 = 2*y6;
%Формирование матрицы Якоби
J(1,1) = f1x1; J(1,2) = f1x2;
J(4,12) = f4y6; J(5,4) = f5x4; J(5,5) = f5x5; J(5,7) = f5y1;
J(5,8) = f5y2; J(5,11) = f5y5; J(6,4) = f6x4; J(6,6) = f6x6; J(6,7) = f6y1; J(6,9)
= f6y3;
J(6,12) = f6y6; J(2,2) = f2x2; J(2,3) = f2x3; J(7,1) = f7x1; J(7,7) = f7y1; J(8,2)
= f8x2;
J(8,8) = f8y2; J(9,3) = f9x3; J(9,9) = f9y3; J(10,4) = f10x4; J(10,10) = f10y4;
J(11,5) = f11x5; J(3,5) = f3x5; J(3,8) = f3y2; J(3,7) = f3y1; J(3,10) = f3y4;
J(3,11) = f3y5;
J(4,6) = f4x6; J(4,7) = f4y1; J(4,9) = f4y3; J(4,10) = f4y4; J(11,11) = f11y5;
J(12,6) = f12x6; J(12,12) = f12y6;
%Нахождение обратной матрицы
Obr = J^(-1);
%Нахождение второго приближения
X1 = X0-Obr*F;
end
%Нахождение координат точек неподвижного основания манипулятора
xA1= s1*cos_g+l*X1(7,1)*X1(10,1)+a;
yA1=l*X1(7,1)*X1(4,1);
zA1=s1*sin_g+l*X1(1,1);
xA2=-s2*cos_g*sin(pi/6)-l*X1(8,1)*(sin(pi/6)*X1(11,1)+cos(pi/6)*X1(5,1))-
a*sin(pi/6);
yA2=s2*cos_g*cos(pi/6)+l*X1(8,1)*(cos(pi/6)*X1(11,1)-
sin(pi/6)*X1(5,1))+a*cos(pi/6);
zA2=s2*sin_g+l*X1(2,1);
xA3=-s3*cos_g*sin(pi/6)-l*X1(9,1)*(X1(6,1)*cos(pi/6)-X1(12,1)*sin(pi/6))-
a*sin(pi/6);
yA3=-s3*cos_g*cos(pi/6)-l*X1(9,1)*(X1(12,1)*cos(pi/6)+sin(pi/6)*X1(6,1))-
a*cos(pi/6);
zA3=s3*sin_g+l*X1(3,1);
%Расчет координат захвата
xC = -(xA1+xA2+xA3)/3;
yC = -(yA1+yA2+yA3)/3;

```



```

zC = -(zA1+zA2+zA3)/3;
%Запись координат захвата в матрицу LARG
LARG(i,1)=xC;
LARG(i,2)=yC;
LARG(i,3)=zC;
end
end
end
%Радиус окружности описанной вокруг основания трипода
R = (sqrt(3)*L)/3;
%Радиус окружности описанной вокруг основания подвижной платформы
RR = (sqrt(3)*a)/3;
%Координаты вершин основания
A1 = [R 0 0];
A2 = [-A1(1)/2 L/2 0];
A3 = [-A1(1)/2 -L/2 0];
%Высота основания трипода
H = S*sin_g;
%Координаты нижней платформы
C1 = [a 0 -Sstart*sin_g-sqrt(l^2-(S*cos_g-a-Sstart*cos_g)^2)];
C2 = [-a/2 a*sqrt(3)/2 -Sstart*sin_g-sqrt(l^2-(S*cos_g-a-Sstart*cos_g)^2)];
C3 = [-a/2 -a*sqrt(3)/2 -Sstart*sin_g-sqrt(l^2-(S*cos_g-a-Sstart*cos_g)^2)];
%Положение кареток
S1 = [R-Sstart*cos_g 0 -Sstart*sin_g];
S2 = [-R/2+Sstart*cos_g/2 L/2-Sstart*cos_g*cos(pi/6) -Sstart*sin_g];
S3 = [-R/2+Sstart*cos_g/2 -L/2+Sstart*cos_g*cos(pi/6) -Sstart*sin_g];
%Создание графика
scrsz = get(0,'ScreenSize');
figure('Position',[40 50 450 400]);
axis([-0.5 0.7 -L/2 L/2 -1.2 0.0])
hold on;
%Основание треугольника
plot3([A1(1) A2(1)], [A1(2) A2(2)], [A1(3) A2(3)], 'b', 'LineWidth', 3)
plot3([A1(1) A3(1)], [A1(2) A3(2)], [A1(3) A3(3)], 'b', 'LineWidth', 3);
plot3([A2(1) A3(1)], [A2(2) A3(2)], [A2(3) A3(3)], 'b', 'LineWidth', 3);
%Пересечение цилиндров
plot3([A1(1) 0], [A1(2) 0], [A1(3) -H], 'b', 'LineWidth', 3);
plot3([A2(1) 0], [A2(2) 0], [A2(3) -H], 'b', 'LineWidth', 3);
plot3([A3(1) 0], [A3(2) 0], [A3(3) -H], 'b', 'LineWidth', 3);
%Положение подвижных плеч
plot3([S1(1) C1(1)], [S1(2) C1(2)], [S1(3) C1(3)], 'g', 'LineWidth', 3)
plot3([S2(1) C2(1)], [S2(2) C2(2)], [S2(3) C2(3)], 'g', 'LineWidth', 3);
plot3([S3(1) C3(1)], [S3(2) C3(2)], [S3(3) C3(3)], 'g', 'LineWidth', 3);
%Обрисовка подвижной платформы
plot3([C1(1) C2(1)], [C1(2) C2(2)], [C1(3) C2(3)], 'g', 'LineWidth', 3)
plot3([C2(1) C3(1)], [C2(2) C3(2)], [C2(3) C3(3)], 'g', 'LineWidth', 3);
plot3([C3(1) C1(1)], [C3(2) C1(2)], [C3(3) C1(3)], 'g', 'LineWidth', 3);
grid on;
title('Зона достижимости манипулятора');
%Отрисовка всех точек области достижимости
for i=1:1:N
plot3(LARG(i,1),LARG(i,2),LARG(i,3), 'r. ');
end
hold off

```

## Приложение 5. Организация решения задачи прямой динамики

Листинг программы “din\_GUI.m”, решающей прямую задачу кинематики:

```
function varargout = din_GUI(varargin)
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',   gui_Singleton, ...
                  'gui_OpeningFcn',   @din_GUI_OpeningFcn, ...
                  'gui_OutputFcn',   @din_GUI_OutputFcn, ...
                  'gui_LayoutFcn',    [], ...
                  'gui_Callback',     []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end
if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
function din_GUI_OpeningFcn(hObject, eventdata, handles, varargin)
handles.output = hObject;
guidata(hObject, handles);
function varargout = din_GUI_OutputFcn(hObject, eventdata, handles)
varargout{1} = handles.output;
function e1_Callback(hObject, eventdata, handles)
global m;
m=str2double(get(handles.e1, 'string'));
if m<=0
    msgbox('Некорректный ввод. Масса должна быть положительной', 'Error',
'error');
    set(handles.e1, 'String', '');
end;
function e1_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject, 'BackgroundColor'),
get(0, 'defaultUicontrolBackgroundColor'))
    set(hObject, 'BackgroundColor', 'white');
end
function e2_Callback(hObject, eventdata, handles)
function e2_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject, 'BackgroundColor'),
get(0, 'defaultUicontrolBackgroundColor'))
    set(hObject, 'BackgroundColor', 'white');
end
function e3_Callback(hObject, eventdata, handles)
function e3_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject, 'BackgroundColor'),
get(0, 'defaultUicontrolBackgroundColor'))
    set(hObject, 'BackgroundColor', 'white');
end
function e4_Callback(hObject, eventdata, handles)
function e4_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject, 'BackgroundColor'),
get(0, 'defaultUicontrolBackgroundColor'))
    set(hObject, 'BackgroundColor', 'white');
end
function e5_Callback(hObject, eventdata, handles)
function e5_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject, 'BackgroundColor'),
get(0, 'defaultUicontrolBackgroundColor'))
    set(hObject, 'BackgroundColor', 'white');
```

```

end
function e6_Callback(hObject, eventdata, handles)
function e6_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
function e7_Callback(hObject, eventdata, handles)
function e7_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
function pb3_Callback(hObject, eventdata, handles)
%Построение графиков изменения сил, действующих на схват, во времени
global F1_an F2_an F3_an;
figure;
t=0:0.1:15;
subplot(3,1,1);
plot(t,eval(F1_an),'r');
grid on;
title('Закон изменения силы F1 во времени');
xlabel('Время t, с');
ylabel('Сила F1, Н');
subplot(3,1,2);
plot(t,eval(F2_an),'b');
grid on;
title('Закон изменения силы F2 во времени');
xlabel('Время t, с');
ylabel('Сила F2, Н');
subplot(3,1,3);
plot(t,eval(F3_an),'g');
grid on;
title('Закон изменения силы F3 во времени');
xlabel('Время t, с');
ylabel('Сила F3, Н');
function pb2_Callback(hObject, eventdata, handles)
%Построение графиков изменения координат схвата во времени
global q1 q2 q3;
figure;
t=0:0.1:15;
subplot(3,1,1);
plot(t,eval(q1),'r');
grid on;
title('Закон изменения координаты X схвата во времени');
xlabel('Время t, с');
ylabel('Значение X');
subplot(3,1,2);
plot(t,eval(q2),'b');
grid on;
title('Закон изменения координаты Y схвата во времени');
xlabel('Время t, с');
ylabel('Значение Y');
subplot(3,1,3);
plot(t,eval(q3),'g');
grid on;
title('Закон изменения координаты Z схвата во времени');
xlabel('Время t, с');
ylabel('Значение Z');
function pb1_Callback(hObject, eventdata, handles)
%Подгрузка значений углов из решения задачи кинематики
load ugol;
%Переопределение переменных, которые содержат значения углов

```

```

x1=double(sa1); x2=double(sa2); x3=double(sa3);
x4=double(sb1); x5=double(sb2); x6=double(sb3);
y1=double(ca1); y2=double(ca2); y3=double(ca3);
y4=double(cb1); y5=double(cb2); y6=double(cb3);
%Ввод массы подвижного основания
global m;
m=str2double(get(handles.e1,'string'));
%Задание значения величины ускорения свободного падения
g=9.806;
%Задание символьных переменных, от которых зависит функция
syms t F1 F2 F3;
%Ввод желаемых законов движения схвата по координатам
global q1 q2 q3;
q1=sym(get(handles.e2,'string'));
q2=sym(get(handles.e3,'string'));
q3=sym(get(handles.e4,'string'));
%Поиск второй производной по каждой обобщенной координате (ускорений)
q1_pr=diff(q1,2);
q2_pr=diff(q2,2);
q3_pr=diff(q3,2);
%Задание функций, которые описывают систему уравнений Лагранжа 2-го рода
f1=vpa(m*q1_pr+F1*y4*y1-F2*((sqrt(3)*x5)/2)+(y5/2))*y2-
F3*y3*((sqrt(3)*x6)/2)+(y6/2)),3);
f2=vpa(m*q2_pr+F1*y4*y1+F2*((sqrt(3)*y5)/2)-(x5/2))*y2-
F3*y3*((sqrt(3)*y6)/2)+(x6/2)),3);
f3=vpa(m*q3_pr+m*g+F1*x1+F2*x2+F3*x3,3);
%Поиск решения системы уравнений относительно переменных F1,F2,F3-силы
[F1_ans, F2_ans, F3_ans]=solve(f1,f2,f3,F1,F2,F3);
%Округление результатов до 3-х знаков после запятой
global F1_an F2_an F3_an;
F1_an=vpa(F1_ans,3);
F2_an=vpa(F2_ans,3);
F3_an=vpa(F3_ans,3);
F1_a=string(F1_an);
F2_a=string(F2_an);
F3_a=string(F3_an);
%Вывод законов изменения сил
set(handles.e5,'string', F1_a);
set(handles.e6,'string', F2_a);
set(handles.e7,'string', F3_a);

```

## Приложение 6. Организация решения задачи обратной динамики

Листинг программы “inv\_din\_GUI.m”, решающей обратную задачу кинематики:

```
function varargout = inv_din_GUI(varargin)
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',   gui_Singleton, ...
                  'gui_OpeningFcn',   @inv_din_GUI_OpeningFcn, ...
                  'gui_OutputFcn',   @inv_din_GUI_OutputFcn, ...
                  'gui_LayoutFcn',    [], ...
                  'gui_Callback',     []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end
if nargin
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
function inv_din_GUI_OpeningFcn(hObject, eventdata, handles, varargin)
handles.output = hObject;
guidata(hObject, handles);
function varargout = inv_din_GUI_OutputFcn(hObject, eventdata, handles)
varargout{1} = handles.output;
function e1_Callback(hObject, eventdata, handles)
global m;
m=str2double(get(handles.e1, 'string'));
if m<=0
    msgbox('Некорректный ввод. Масса должна быть положительной', 'Error',
'error');
    set(handles.e1, 'String', '');
end;
function e1_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject, 'BackgroundColor'),
get(0, 'defaultUicontrolBackgroundColor'))
    set(hObject, 'BackgroundColor', 'white');
end
function e2_Callback(hObject, eventdata, handles)
function e2_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject, 'BackgroundColor'),
get(0, 'defaultUicontrolBackgroundColor'))
    set(hObject, 'BackgroundColor', 'white');
end
function e3_Callback(hObject, eventdata, handles)
function e3_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject, 'BackgroundColor'),
get(0, 'defaultUicontrolBackgroundColor'))
    set(hObject, 'BackgroundColor', 'white');
end
function e4_Callback(hObject, eventdata, handles)
function e4_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject, 'BackgroundColor'),
get(0, 'defaultUicontrolBackgroundColor'))
    set(hObject, 'BackgroundColor', 'white');
end
function pb2_Callback(hObject, eventdata, handles)
global F1 F2 F3;
```

```

figure;
t=0:0.1:100;
subplot(3,1,1);
plot(t,eval(F1),'r');
grid on;
title('Закон изменения силы F1 во времени');
xlabel('Время t, с');
ylabel('Сила F1, Н');
subplot(3,1,2);
plot(t,eval(F2),'b');
grid on;
title('Закон изменения силы F2 во времени');
xlabel('Время t, с');
ylabel('Сила F2, Н');
subplot(3,1,3);
plot(t,eval(F3),'g');
grid on;
title('Закон изменения силы F3 во времени');
xlabel('Время t, с');
ylabel('Сила F2, Н');
function pb3_Callback(hObject, eventdata, handles)
global q1_ans q2_ans q3_ans;
figure;
t=0:0.1:100;
subplot(3,1,1);
plot(t,eval(q1_ans),'r');
grid on;
title('Закон изменения координаты X схвата во времени');
xlabel('Время t, с');
ylabel('Значение X');
subplot(3,1,2);
plot(t,eval(q2_ans),'b');
grid on;
title('Закон изменения координаты Y схвата во времени');
xlabel('Время t, с');
ylabel('Значение Y');
subplot(3,1,3);
plot(t,eval(q3_ans),'g');
grid on;
title('Закон изменения координаты Z схвата во времени');
xlabel('Время t, с');
ylabel('Значение Z');
% --- Executes on button press in pb1.
function pb1_Callback(hObject, eventdata, handles)
% hObject      handle to pb1 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
%Подгрузка значений углов из решения задачи кинематики
load ugol;
%Переопределение переменных, которые содержат значения углов
x1=double(sa1); x2=double(sa2); x3=double(sa3);
x4=double(sb1); x5=double(sb2); x6=double(sb3);
y1=double(ca1); y2=double(ca2); y3=double(ca3);
y4=double(cb1); y5=double(cb2); y6=double(cb3);
%Задание символьных переменных, от которых зависит функция
syms t;
%Ввод массы подвижного основания
global m;
m=str2double(get(handles.e1,'string'));
%Задание значения величины ускорения свободного падения
g=9.806;
%Ввод желаемых законов движения схвата по координатам
global F1 F2 F3;

```

```

F1=sym(get(handles.e2,'string'));
F2=sym(get(handles.e3,'string'));
F3=sym(get(handles.e4,'string'));
%Задание функций, которые описывают систему уравнений Лагранжа 2-го рода
f1=vpa(-(F1*y4*y1-F2*((sqrt(3)*x5)/2)+(y5/2))*y2-
F3*y3*((sqrt(3)*x6)/2)+(y6/2))/m,3);
f2=vpa(-(F1*y4*y1+F2*((sqrt(3)*y5)/2)-(x5/2))*y2-
F3*y3*((sqrt(3)*y6)/2)+(x6/2))/m,3);
f3=vpa(-(m*g+F1*x1+F2*x2+F3*x3)/m,3);
%Зададим как символьные функции от времени обобщенные координаты
syms q1(t) q2(t) q3(t);
%Функции производных для обобщенных координат
Dq1=diff(q1,t);
Dq2=diff(q2,t);
Dq3=diff(q3,t);
%Задание НУ по перемещению и по скорости
q1_0=0;
Dq1_0=0;
q2_0=0;
Dq2_0=0;
q3_0=0;
Dq3_0=0;
%Поиск решения диф уравнений относительно переменных q1,q2,q3-координаты
global q1_ans q2_ans q3_ans;
q1_ans=vpa(dsolve(diff(q1,2)==f1, q1(0)==q1_0, Dq1(0)==Dq1_0),3);
q2_ans=vpa(dsolve(diff(q2,2)==f2, q2(0)==q2_0, Dq2(0)==Dq2_0),3);
q3_ans=vpa(dsolve(diff(q3,2)==f3, q3(0)==q3_0, Dq3(0)==Dq3_0),3);
q1_an=string(q1_ans);
q2_an=string(q2_ans);
q3_an=string(q3_ans);
%Вывод законов изменения координат
set(handles.t1,'string', q1_an);
set(handles.t2,'string', q2_an);
set(handles.t3,'string', q3_an);

function e5_Callback(hObject, eventdata, handles)
function e5_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
function e6_Callback(hObject, eventdata, handles)
function e6_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
function e7_Callback(hObject, eventdata, handles)
function e7_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
function e8_Callback(hObject, eventdata, handles)
function e8_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
function e9_Callback(hObject, eventdata, handles)
function e9_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))

```

```
        set(hObject, 'BackgroundColor', 'white');
end
function e10_Callback(hObject, eventdata, handles)
function e10_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject, 'BackgroundColor'),
get(0, 'defaultUicontrolBackgroundColor'))
    set(hObject, 'BackgroundColor', 'white');
end
```