

Санкт–Петербургский государственный университет

ХАНЕЕВА Динара Владимировна

Выпускная квалификационная работа

Управление движением робота-манипулятора

Уровень образования: бакалавриат

Направление 27.03.03 «Системный анализ и управление»

Основная образовательная программа СВ.5164.2020

«Прикладные компьютерные технологии»

Научный руководитель:

профессор, кафедра моделирования

электромеханических и компьютерных систем,

д.ф.-м.н. Малькова Юлия Вениаминовна

Рецензент:

доцент Санкт-Петербургского

политехнического университета Петра Великого,

к.ф.-м.н. Семакова Анна Анатольевна

Санкт-Петербург

2024 г.

Содержание

1	Введение	3
2	Постановка задачи	4
2.1	Динамическая модель трехзвенного робота-манипулятора	4
2.2	Динамическая модель точек на изображении	5
3	Используемые методы	6
3.1	Линеаризация обратной связью	6
3.2	Расчет якобиана	7
3.3	IBVS (Image-Based Visual Servoing)	8
4	Результаты экспериментов	10
4.1	Валидация линеаризации	11
4.2	Валидация движения в декартову точку в скоростном управлении	12
4.3	Визуальное позиционирование	13
4.4	Визуальное позиционирование с постоянной L	15
5	Заключение	17
6	Список литературы	18

1 Введение

В современном мире робототехники и автоматизации производства визуальное позиционирование с манипулятором становится все более актуальным и востребованным направлением. Этот метод позволяет значительно повысить точность, гибкость и эффективность работы роботизированных систем в различных отраслях промышленности, таких как сборка, сортировка, упаковка и контроль качества.

Визуальное позиционирование с манипулятором основывается на интеграции систем компьютерного зрения и робототехнических манипуляторов. Системы компьютерного зрения, оснащенные камерами и алгоритмами обработки изображений, позволяют получать детальную информацию о положении, ориентации и свойствах объектов в рабочей зоне. Эта информация затем используется для точного управления движениями манипулятора, обеспечивая высокую точность позиционирования и манипулирования объектами.

Например, в [6] рассмотрено позиционирование и навигация мобильного робота относительно искусственного ориентира. В качестве искусственных ориентиров выступают линии, ограничивающие трассу робота, и метки в виде двумерных штриховых кодов. В [7] рассматривается система визуального позиционирования беспилотников для точной посадки. Система основана на использовании фрактальных маркеров и светодиодов разного цвета. В [8] позиционирование колесного робота производится с помощью blob-объекта (каска рабочего белого цвета) для обеспечения безопасности людей.

Одним из ключевых преимуществ визуального позиционирования является возможность адаптации к изменениям в рабочей среде. Традиционные методы программирования роботов, основанные на заранее заданных траекториях и координатах, часто не способны эффективно справляться с вариациями в положении и ориентации объектов. Визуальное позиционирование, напротив, позволяет роботу динамически корректировать свои движения на основе визуальной обратной связи, что делает систему более гибкой и устойчивой к изменениям.

Кроме того, визуальное позиционирование открывает новые возможности для автоматизации сложных задач, требующих высокой точности и повторяемости. Например, в электронной промышленности визуальная вставка компонентов на печатные платы является критически важной операцией, требующей точного позиционирования в масштабе микрометров. Использование систем технического зрения и прецизионных манипуляторов позволяет автоматизировать этот процесс, обеспечивая высокое качество и производительность.

Целью данной работы является реализация визуального позиционирования трехзвенного робота-манипулятора с видеокамерой, с учетом управления по моментам.

- Задачи:**
- 1) Построить динамическую модель робота и точек на изображении
 - 2) Провести линеаризацию обратной связью динамической модели робота
 - 3) Рассчитать якобиан для перехода от декартовых скоростей к линейным
 - 4) Реализовать визуальное позиционирование робота по маркеру
 - 5) Проверить результаты

2 Постановка задачи

2.1 Динамическая модель трехзвенного робота-манипулятора

Рассмотрим трехзвенного робота-манипулятора (рис. 1), для которого $m_1 = m_2 = m$ — точечные массы на концах звеньев, $L_1 = L_2 = L$ — длины звеньев, $\theta = (\theta_1, \theta_2, \theta_3)^T$ — углы отклонения звеньев.

Примем $m = 1$ кг, $L = 0.5$ м.

Получим его динамическую модель с помощью лагранжиана, который представляет собой разность кинетической и потенциальной энергий, с последующей его подстановкой в уравнения Эйлера-Лагранжа.

$$\tau_i = \frac{d}{dt} \frac{\partial L}{\partial \dot{\theta}_i} - \frac{\partial L}{\partial \theta_i}, \quad i = 1, 2, 3.$$

Таким образом, получим уравнение, описывающее динамику робота

$$\tau = M(\theta)\ddot{\theta} + c(\theta, \dot{\theta}) + g(\theta) \quad (2.1.1)$$

$$M(\theta) = \begin{bmatrix} 3mL^2 + 2mL^2\cos(\theta_2) & mL^2 + mL^2\cos(\theta_2) & 0 \\ mL^2 + mL^2\cos(\theta_2) & mL^2 & 0 \\ 0 & 0 & J \end{bmatrix},$$

$$c(\theta, \dot{\theta}) = \begin{bmatrix} -2mL^2\dot{\theta}_1\dot{\theta}_2\sin(\theta_2) - mL^2\dot{\theta}_2^2\sin(\theta_2) \\ 2mL^2\dot{\theta}_1^2\sin(\theta_2) \\ 0 \end{bmatrix},$$

$$g(\theta) = \begin{bmatrix} 2mgL\sin(\theta_1) + mgL\sin(\theta_1 + \theta_2) \\ mgL\sin(\theta_1 + \theta_2) \\ 0 \end{bmatrix},$$

где $M(\theta)$ — симметричная положительно-определенная матрица инерции, $c(\theta, \dot{\theta})$ — вектор моментов Кориолиса и центробежных моментов, $g(\theta)$ — вектор гравитационных моментов, τ — вектор управляющих моментов, g — ускорение свободного падения, J — момент инерции.

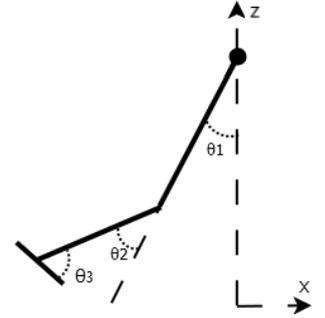


Рис. 1: Трехзвенный робот-манипулятор

2.2 Динамическая модель точек на изображении

Предположим, что на фланце робота жестко закреплена видеокамера, оптическая ось которой направлена по оси Ox , а оптический центр находится в точке O . Пусть $Ox_c y_c z_c$ – система координат, связанная с камерой. При этом координатная ось Oz_c системы координат камеры сонаправлена оси Ox связанной системы координат, ось Ox_c сонаправлена оси Oy , а ось Oy_c – оси Oz .

Рассмотрим также нормированную плоскость изображения. Введем в этой плоскости систему координат $O_s x_s y_s$, центр которой находится на оптической оси, ось $O_s y_s$ сонаправлена с осью Oy_c , а ось $O_s x_s$ противоположно направлена с осью Ox_c . Пусть (X_c, Y_c, Z_c) – координаты некоторой точки пространства, заданные в системе координат камеры. Тогда координаты проекции этой точки на плоскость изображения определяются следующими формулами

$$x_s = -\frac{X_c}{Z_c}, y_s = \frac{Y_c}{Z_c}.$$

Будем считать, что в поле зрения камеры постоянно находится некоторый визуальный маркер, в плоскости изображения представленный набором точек $s = (x_i, y_i), i = \overline{1, N}$, которые в свою очередь являются проекциями точек маркера (X_i, Y_i, Z_i) , заданными в системе координат камеры. Поставим задачу позиционирования объекта относительно маркера таким образом, чтобы проекции точек маркера s на изображении оказались в желаемом положении s^* , т.е. необходимо обеспечить выполнение условия:

$$\lim_{t \rightarrow \infty} s(t) = s^*.$$

3 Используемые методы

3.1 Линеаризация обратной связью

Линеаризация обратной связью — это подход в теории управления нелинейными системами, который позволяет преобразовать исходную нелинейную систему в эквивалентную линейную систему путем введения нелинейной обратной связи.

В математической модели трехзвеного робота-манипулятора

$$M(\theta)\ddot{\theta} + C(\theta, \dot{\theta}) + g(\theta) = \tau \quad (3.1.1)$$

матрицы $M(\theta)$, $C(\theta, \dot{\theta})$ и $g(\theta)$ являются нелинейными.

Разрешим (3.1.1) относительно старших производных

$$\ddot{\theta} = M^{-1}(\theta)(\tau - C(\theta, \dot{\theta}) - g(\theta)). \quad (3.1.2)$$

Далее приведем (3.1.2) к системе ОДУ первого порядка. Рассмотрим вектор $\omega(t) = \dot{\theta}(t)$, тогда (3.1.2) можно представить в виде

$$\begin{cases} \dot{\theta} = \omega, \\ \dot{\omega} = M^{-1}(\theta)(\tau - C(\theta, \omega) - g(\theta)) \end{cases} \quad (3.1.3)$$

Управляющий сигнал представим как

$$\tau = C(\theta, \omega) + g(\theta) + M(\theta)u, \quad (3.1.4)$$

где $u = u(t)$ — некоторая векторная функция.

Подставим (3.1.4) в (3.1.3) и получим линейную систему

$$\begin{cases} \dot{\theta} = \omega, \\ \dot{\omega} = u, \end{cases} \quad (3.1.5)$$

для которой u - управляющий сигнал.

Для обеспечения заданной скорости будет использован пропорциональный регулятор

$$u = k(\omega_{\text{жел}} - \omega_{\text{тек}}), \quad (3.1.6)$$

где k - коэффициент пропорциональности, $\omega_{\text{жел}}$ - желаемая скорость робота, $\omega_{\text{тек}}$ - текущая скорость робота.

3.2 Расчет якобиана

Рассмотрим задачу приведения системы в заданную точку декартова пространства $X_{\text{жел}} = (x, z, \psi)$. Для реализации управления необходимо преобразовать желаемые скорости системы из декартова пространства в пространство обобщенных координат (углы сочленений, joint space). Этот процесс преобразования осуществляется с помощью матрицы Якоби. Выразим декартовы координаты через обобщенные

$$\begin{bmatrix} x \\ z \\ \psi \end{bmatrix} = \begin{bmatrix} -L\sin(q_1) - L\sin(q_1 + q_2) \\ -L\cos(q_1) - L\cos(q_1 + q_2) \\ \theta_1 + \theta_2 + \theta_3 \end{bmatrix}$$

$$\begin{bmatrix} \dot{x} \\ \dot{z} \\ \dot{\psi} \end{bmatrix} = J \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \end{bmatrix},$$

где J - матрица Якоби.

Вычислим производные по времени, тогда матрица Якоби примет вид

$$J = \begin{bmatrix} -L\cos(q_1) - L\cos(q_1 + q_2) & -L\cos(q_1 + q_2) & 0 \\ L\sin(q_1) + L\sin(q_1 + q_2) & L\sin(q_1 + q_2) & 0 \\ 1 & 1 & 1 \end{bmatrix} \quad (3.2.1)$$

Теперь можем получить связь между скоростями в пространстве обобщенных координат $\dot{\theta}$ и в декартовом пространстве \dot{x} :

$$\omega = \dot{\theta} = J^{-1}\dot{x}$$

Управление будем производить с помощью пропорционального скоростного регулятора в декартовом пространстве, который формирует управляющий сигнал u , пропорциональный ошибке положения $e = X_{\text{тек}} - X_{\text{жел}}$:

$$u = de,$$

где d — коэффициент пропорциональности. В итоге получаем желаемую скорость в обобщенных координатах:

$$\omega = d * J^{-1}e$$

3.3 IBVS (Image-Based Visual Servoing)

Визуальное сервоуправление на основе изображения (IBVS) представляет собой метод управления роботами, основанный на визуальной обратной связи. Этот подход использует информацию, полученную с камеры, для определения положения и ориентации робота относительно целевого объекта, а также для генерации управляющих сигналов, необходимых для достижения желаемого положения робота.

В данном исследовании камера закреплена на фланце робота (третьем звене), в качестве целевого объекта применяется ArUco-маркер (рис. 2). ArUco-маркер представляет собой квадрат, окаймлённый широкой чёрной границей, содержащий внутри бинарную квадратную матрицу, которая однозначно его идентифицирует (рис. 2).

Дополним систему (2.1.1) уравнением динамики изменения проекций точек маркера на плоскость изображения в соответствии с математическими моделями, описанными в [5]:

$$\dot{s} = L_s(s, Y_c)v_c \quad (3.3.1)$$

Вектор Y_c содержит значения $Y_i, i = \overline{1, N}$ для каждой спроецированной точки маркера, $L_s(s, Y_c)$ — матрица взаимосвязей, компоненты которой для каждой пары (x_i, z_i) задаются следующим образом:

$$L_s(x_i, y_i, Z_i) = \begin{bmatrix} \frac{x_i}{Z_i} & -\frac{1}{Z_i} & 0 & y_i & x_i y_i & -(1 + x_i^2) \\ \frac{y_i}{Z_i} & 0 & -\frac{1}{Z_i} & -x_i & 1 + y_i^2 & -x_i y_i \end{bmatrix}$$

Систему (3.3.1) необходимо дополнить уравнением динамики вектора Y_c

$$\dot{Z}_c = L_z(s, Z_c)v_c,$$

где $L_z(s, Z_c)$ — матрица, строки которой равны

$$L_z(x_i, y_i, Z_i) = (-1 \quad 0 \quad 0 \quad 0 \quad -y_i Z_i \quad x_i Z_i)$$

Таким образом, динамика робота и проекций маркера в плоскости изображения полностью описываются уравнениями (2.1.1), (3.3.1) Матрица взаимодействия в данном случае упрощается:

$$L(x_i, y_i, Z_i) = \begin{bmatrix} -\frac{1}{Z_i} & 0 & y_i \\ 0 & -\frac{1}{Z_i} & -x_i \end{bmatrix}$$

Цель управления — минимизировать ошибку $e(t)$, которая определяется как

$$e(t) = s - s^* \quad (3.3.2)$$

Из (3.3.1) и (3.3.2) получаем

$$\dot{e} = L_e v_c, \quad (3.3.3)$$

где $L_e = L_s$

Рассматривая v_c в качестве входного сигнала для контроллера робота, а также желая обеспечить экспоненциальную зависимость уменьшения ошибки ($\dot{e} = -\lambda e$), получаем, используя

$$v_c = -\lambda L_e^+ e,$$

где $L_e^+ = (L_e^T L_e)^{-1} L_e^T$ - псевдообратная матрица к L_e .

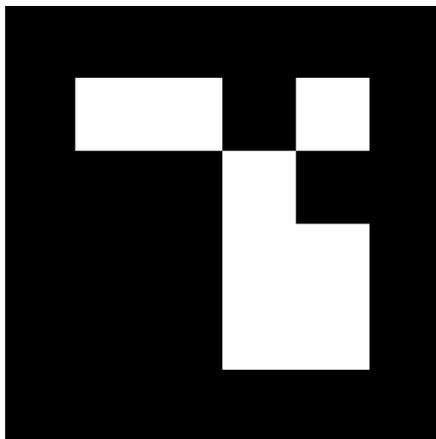


Рис. 2: ArUco-маркер

4 Результаты экспериментов

В настоящее время доступен широкий спектр программных средств и инструментов, предназначенных для виртуального моделирования и симуляции различных физических систем, в том числе роботизированных устройств. Использование таких инструментов позволяет существенно сократить затраты материальных ресурсов на проведение экспериментов с реальными роботами, что особенно актуально на этапах проектирования, отладки и тестирования алгоритмов управления.

Одним из популярных и эффективных инструментов виртуального моделирования является модуль `pybullet`, разработанный на языке программирования Python. Данный модуль представляет собой физический движок с открытым исходным кодом, который позволяет создавать и симулировать динамику твердых тел, включая роботизированные системы различной сложности, а также позволяет снимать изображения с виртуальной камеры.

Для моделирования трехзвенного робота-манипулятора использовался формат URDF (Unified Robot Description Format), который является широко распространенным стандартом для описания кинематической и динамической структуры роботов. URDF - это XML-спецификация, которая позволяет детально определить геометрию, кинематику и физические свойства робототехнической системы, состоящей из множества взаимосвязанных твердых тел.

В формате URDF робот представляется как иерархическая древовидная структура, состоящая из звеньев (`links`), соединенных сочленениями (`joints`). Каждое звено описывает жесткое тело с определенными геометрическими и инерционными свойствами, такими как форма, размеры, масса и тензор инерции. Сочленения, в свою очередь, определяют кинематические связи между звеньями, задавая степени свободы и ограничения на относительное движение.

Модель робота в среде `pybullet` представлена на рис. 3.

Для проведения всех экспериментов была написана программа на языке Python:

<https://github.com/pawidlow/vkr>

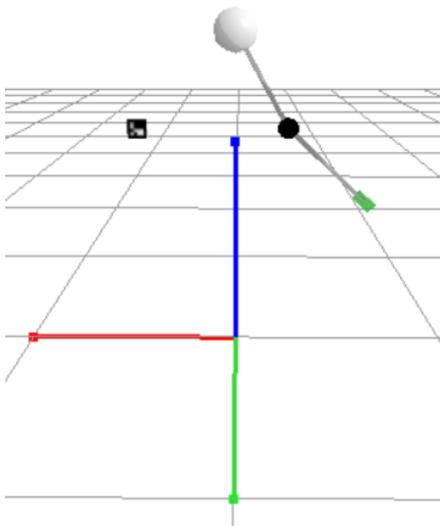


Рис. 3: Модель робота в среде `pybullet`

4.1 Валидация линеаризации

Для проверки справедливости линеаризации (3.1.4) проведем эксперимент, заключающийся в задании постоянного значения управляющему сигналу u (3.1.6)

$$u = \begin{pmatrix} 0.1 \\ 0.1 \\ 0.1 \end{pmatrix}$$

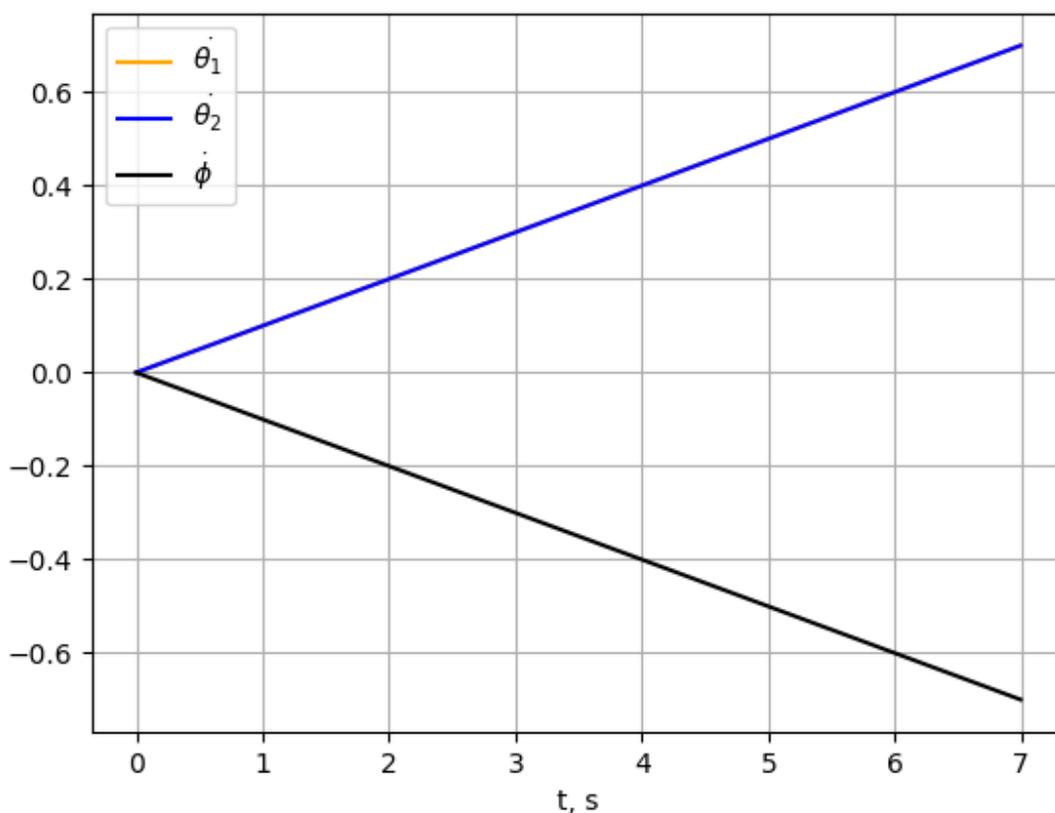


Рис. 4: Динамика изменения скорости звеньев

Отметим, что изменение скорости звеньев подчиняется линейной зависимости, что наглядно продемонстрировано на графике (рис. 4). Таким образом, линеаризация обратной связи проведена корректно. Для подтверждения визуальных результатов также были вычислены значения угловых ускорений (ось третьего джоинта направлена противоположно осям первого и второго джоинтов)

$$\begin{bmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \\ \ddot{\theta}_3 \end{bmatrix} = \begin{bmatrix} 0.1 \\ 0.1 \\ -0.1 \end{bmatrix}.$$

4.2 Валидация движения в декартову точку в скоростном управлении

Рассмотрим задачу движения робота в желаемую декартову точку $X = (0.6, 0.8, 0.5)$. С использованием якобиана, определенного в уравнении (3.2.1), произведем преобразование декартовой скорости в угловую скорость, затем применяем скоростное управление для регулировки системы, коэффициент пропорциональности $k = 10$.

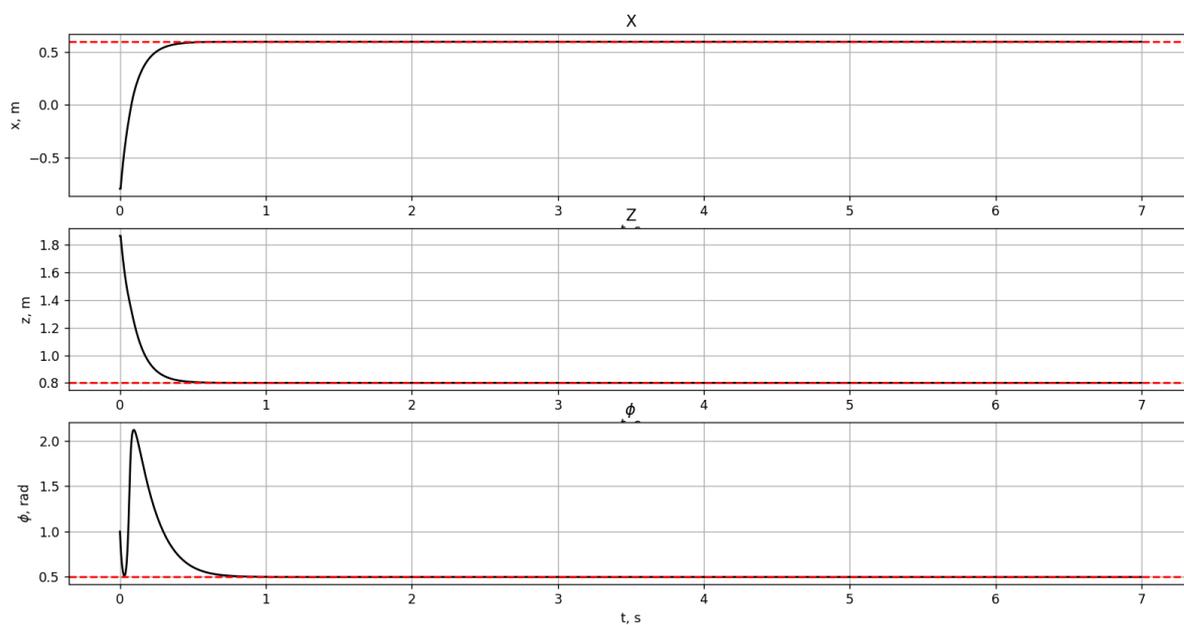


Рис. 5

Как видно на графике (рис. 5), робот успешно достиг заданной позиции, что является подтверждением корректности вычислений якобиана и связи между декартовой и угловой скоростями.

4.3 Визуальное позиционирование

Проведем эксперимент по визуальному позиционированию в разные желаемые положения системы $X(x, z, \phi)$, применяя линеаризацию по обратной связи и переход от декартовых скоростей к угловым.

1) $X_{\text{жел}} = (0.5, 1, 0)$, $X_{\text{нач}} = (0.55, 1.05, 0.1)$

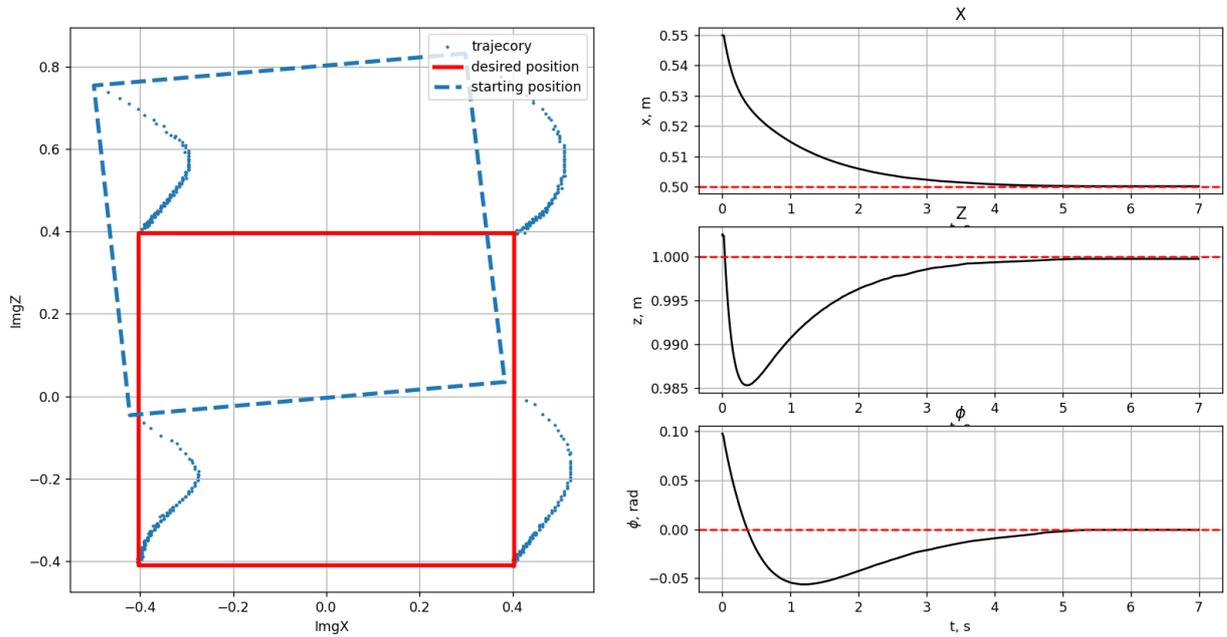


Рис. 6: Динамика движения робота в первую точку

2) $X_{\text{жел}} = (0.55, 1, 0.2)$, $X_{\text{нач}} = (0.5, 1, 0.2)$

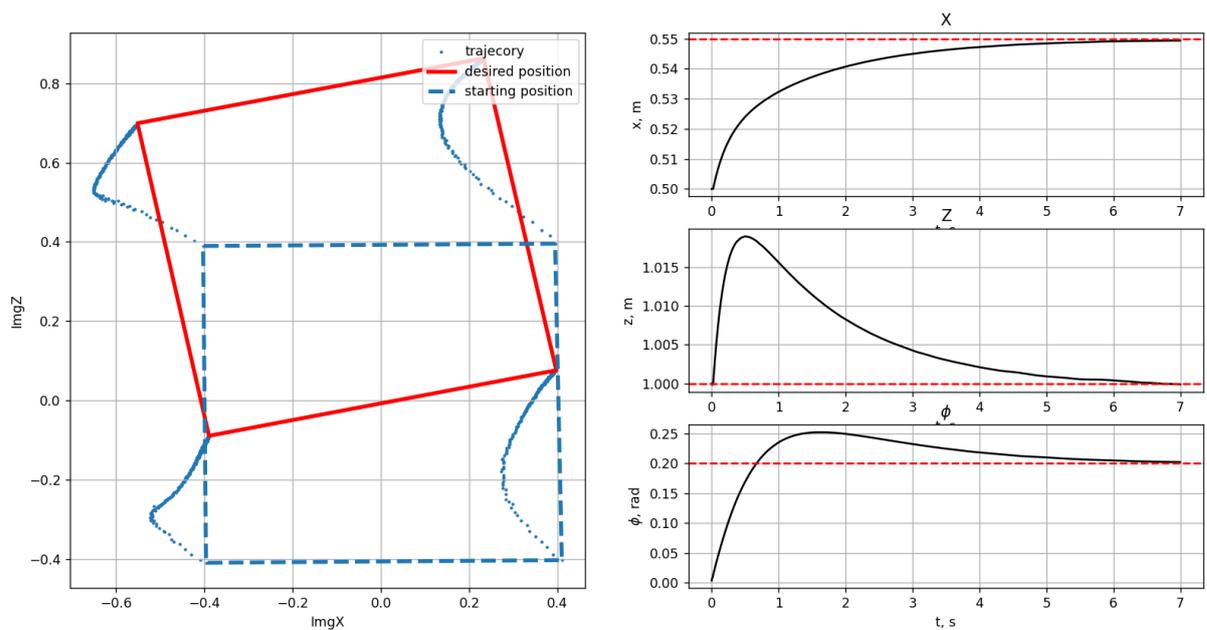


Рис. 7: Динамика движения робота во вторую точку

3) $X_{\text{жел}} = (0.45, 1, -0.1)$, $X_{\text{нач}} = (0.5, 1, 0.1)$

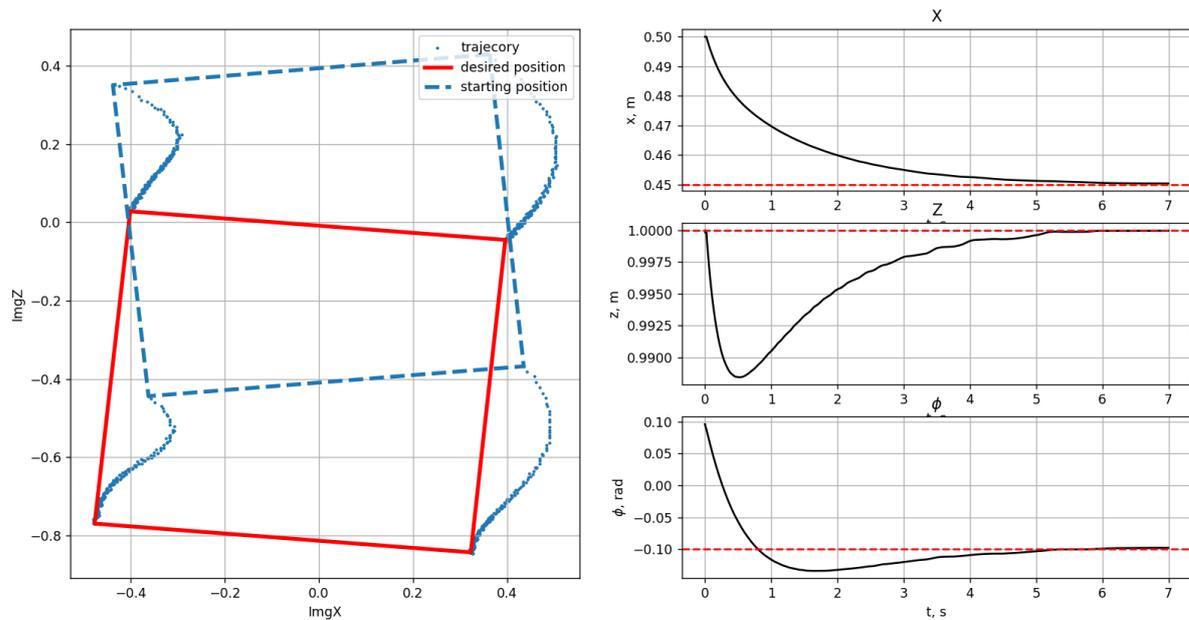


Рис. 8: Динамика движения робота в третью точку

Как видно на рисунках 6 - 8 робот успешно достигает заданного положения.

4.4 Визуальное позиционирование с постоянной L

Проведем эксперимент по визуальному позиционированию робота как в предыдущем пункте, только матрицу взаимодействия L вычислим только один раз для желаемого положения, таким образом, матрица L будет постоянной.

1) $X_{\text{жел}} = (0.5, 1, 0)$, $X_{\text{нач}} = (0.55, 1.05, 0.1)$

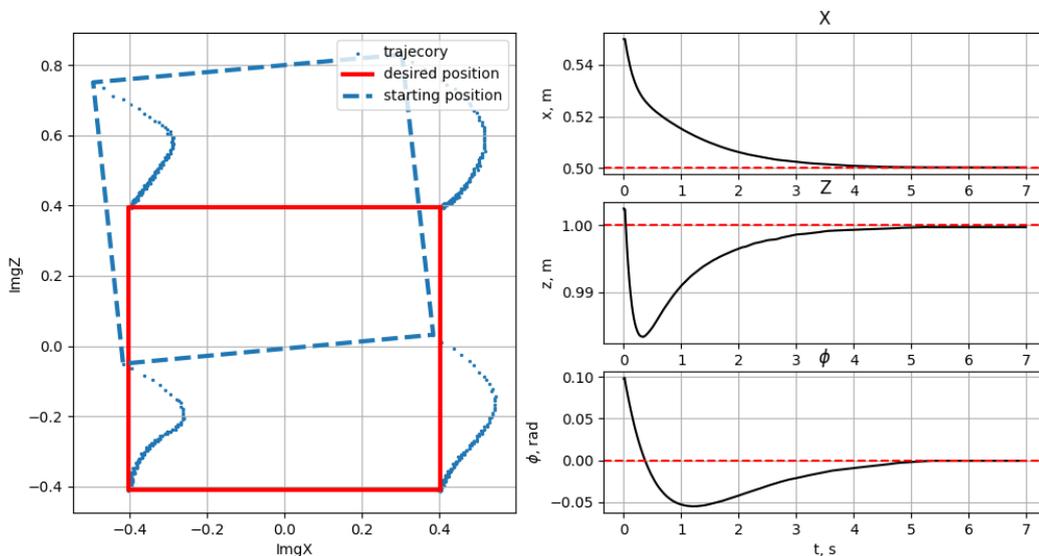


Рис. 9: Динамика движения робота в первую точку

Визуальное позиционирование с константной матрицей L (рис. 9) не показало больших отличий в скорости достижения заданного положения в сравнении с предыдущим пунктом.

2) $X_{\text{жел}} = (0.55, 1, 0.2)$, $X_{\text{нач}} = (0.5, 1, 0.2)$

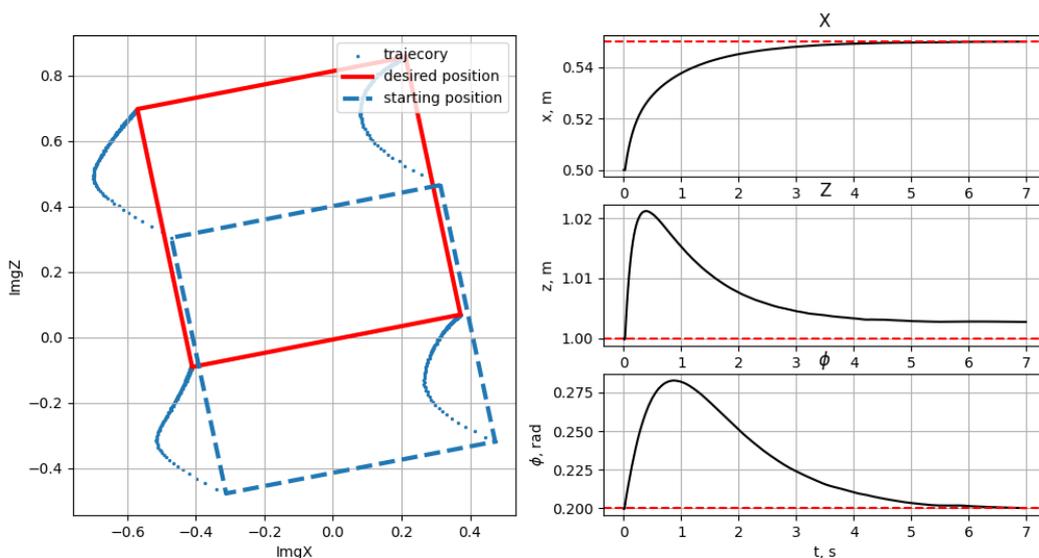


Рис. 10: Динамика движения робота во вторую точку

Как видно из рис. 10 визуальное позиционирование с константной матрицей L не привело робота в заданное положение.

3) $X_{\text{жел}} = (0.45, 1, -0.1)$, $X_{\text{нач}} = (0.5, 1, 0.1)$

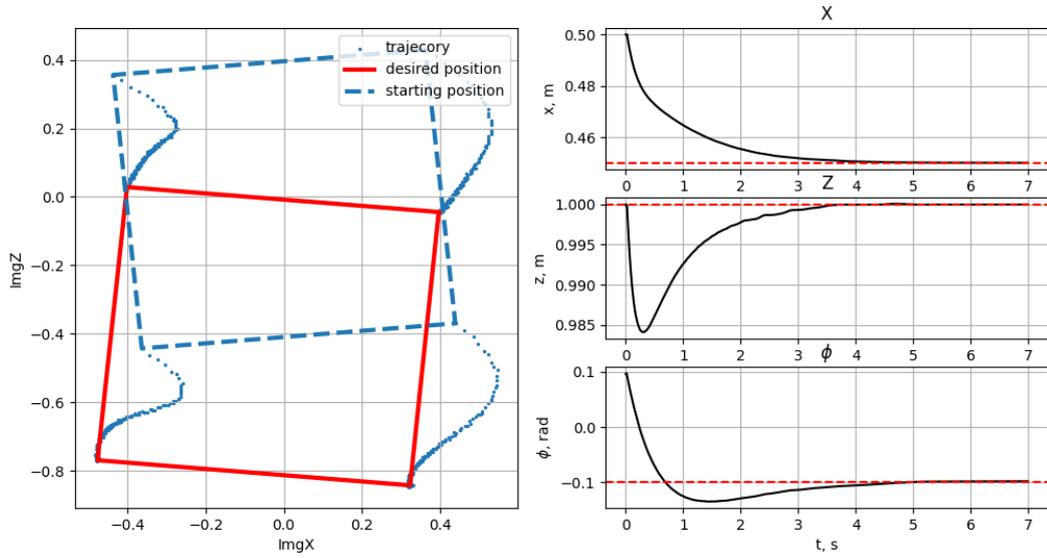


Рис. 11: Динамика движения робота в третью точку

Можно заметить, что визуальное позиционирование с константной матрицей L (рис. 11) привело робота в заданную точку быстрее, чем в предыдущем пункте.

Таким образом, визуальное позиционирование с константной матрицей L привело к неоднозначным результатам.

5 Заключение

Таким образом, поставленные задачи были успешно выполнены, результаты моделирования подтверждают эффективность предложенного подхода к визуальному позиционированию трехзвеного робота-манипулятора. Робот способен точно позиционироваться относительно целевого объекта, используя визуальную обратную связь и алгоритмы управления.

Разработанный метод визуального позиционирования может быть применен в различных областях. Визуальное позиционирование роботов-манипуляторов в производственных линиях позволяет повысить точность и эффективность сборочных операций, упаковки, сортировки и контроля качества продукции. Внедрение таких систем уменьшает вероятность ошибок и увеличивает производительность за счет автоматизации рутинных задач.

В складских системах роботы с визуальным позиционированием могут эффективно выполнять задачи по перемещению и укладке товаров, обеспечивая оптимальное использование пространства и повышение скорости обработки заказов. Они способны самостоятельно ориентироваться в пространстве, распознавать объекты и точно их размещать.

6 Список литературы

1. Kevin M. Lynch and Frank C. Park. Modern Robotics: Mechanics, Planning, and Control//Cambridge University Press, 2017
2. Поляк Б.Т., Хлебников М.В., Рапопорт Л.Б. Математическая теория автоматического управления: учебное пособие//М.: ЛЕНАНД, 2019. — 500 с.: ил.
3. Документация pybullet.
<https://docs.google.com/document/d/10sXEhzFRSnvFcl3XxNGhnD4N2SedqwdAvK3dsihxVUA/edit>
4. Документация opencv.
<https://docs.opencv.org/4.x/>
5. Malis E., Chaumette F., Boudet S. 2 1/2 D Visual Servoing // IEEE Transactions on Robotics and Automation. 1999. Vol. 15, № 2. P. 238–250.
6. Юдин Д.А., Проценко В.В., Постольский Г.Г., Кижук А.С., Магергут В.З. Система технического зрения для автоматического ориентирования и позиционирования мобильного робота//Робототехника и техническая кибернетика. 2014. №1 стр. 70-75.
7. Севостьянов И.Е., Девиэт Д.В. Система визуального позиционирования многороторный беспилотников для совершения высокоточной атомной посадки//Science Time. 2021
8. Павлов В. В., Орлов С. А. Применение системы технического зрения в задаче автоматизации сборочных процессов при совместной работа робота-манипулятора и человека//StudNet. 2022
9. Ефанов, А.В. Теория автоматического управления : учебник для вузов // А. В. Ефанов, В.А. Ярош. — 3-е изд., стер. — Санкт-Петербург : Лань, 2024. — 160 с.
10. Охорзин, В. А. Теория управления : учебник // В. А. Охорзин, К. В. Сафонов. — Санкт-Петербург : Лань, 2022. — 224 с.
11. Обратная линеаризация.
<https://akhatib.com/feedback-linearization-its-performance-and-limitations/>
12. Кэлер, А. Изучаем OpenCV 3. Разработка программ компьютерного зрения на C++ с применением библиотеки OpenCV // А. Кэлер, Г. Брэдски; перевод с английского А. А. Слинкина. — Москва : ДМК Пресс, 2017. — 826 с.
13. Зубов В.И. Лекции по теории управления: Учебное пособие — 2-е изд., испр. — Санкт-Петербург: Лань, 2009. - 496 с.