

Санкт–Петербургский государственный университет

*Уткин Алексей Васильевич*

**Выпускная квалификационная работа**

*Синтез законов управления подвижными объектами с использованием компьютерного зрения*

Уровень образования: магистратура

Направление 01.04.02 «Прикладная математика и информатика»

Основная образовательная программа ВМ.5718.2018 «Прикладные информационные технологии. Информационные экспертные системы»

Профиль «Вычислительные методы и информационные технологии в современном естествознании»

Научный руководитель:

профессор, кафедра компьютерных технологий и систем, д.ф. - м.н. Сотникова Маргарита Викторовна

Рецензент:

инженер, СПбГБУЗ «Медицинский информационно-аналитический центр», Арбузова Юлия Михайловна

Санкт-Петербург

2020 г.

# Содержание

<b>Введение</b> . . . . .	3
<b>Постановка задачи</b> . . . . .	4
<b>Обзор литературы</b> . . . . .	5
<b>Глава 1. Разработка закона управления колесным роботом с использованием алгоритмов компьютерного зрения</b> . . . . .	6
<b>Глава 2. Проверка закона управления в среде MatLab</b> . . . . .	8
<b>Глава 3. Проектирование программного комплекса с элементами виртуальной реальности</b> . . . . .	11
3.1. Gazebo . . . . .	11
3.2. Robot Operating System . . . . .	11
3.3. Turtlebot3 . . . . .	12
3.4. Выбор метода извлечения ключевых точек . . . . .	12
3.5. Объединение выбранных частей в комплекс . . . . .	13
<b>Глава 4. Тестирование программного комплекса</b> . . . . .	14
<b>Глава 5. Введение учета динамики</b> . . . . .	17
5.1. Модель робота с учетом динамики . . . . .	17
5.2. Моделирование . . . . .	19
<b>Выводы</b> . . . . .	29
<b>Заключение</b> . . . . .	31
<b>Список литературы</b> . . . . .	32

## Введение

Современная промышленность движется по направлению полной автоматизации производственного процесса. Одно из характерных решений автоматизации — замена человека на робота. Неотъемлемой частью робота является набор датчиков, наибольший потенциал среди которых имеет видеокамера. В статье [1] описаны методы, позволяющие реализовывать управление с визуальной обратной связью или Visual servo control (VS). Частный случай, называемый Image-Based VS (IBVS), позволяет осуществлять управление без триангуляции положения камеры. Такой метод может быть применен к задачам следования в колонне и позиционирования в конкретной точке, которые являются типичными в логистике промышленного склада.

Немаловажным фактором, характеризующим современные задачи автоматического управления, является повышение требований к динамике управляемого движения подвижного объекта в различных режимах работы. В частности, нередко требуется обеспечение заданного поведения системы в случае наличия внешних возмущений различного характера. Таким образом, при проектировании системы управления необходимо принимать во внимание динамику управляемого объекта и выдвигаемые требования к качеству управления.

Современный уровень развития компьютерных технологий позволяет осуществлять проверку и анализ алгоритмов управления при помощи моделирования на электронной вычислительной машине (ЭВМ), тем самым исключая из результатов ошибку, связанную с неидеальностью реальных управляемых объектов. В случае алгоритмов VS использование компьютерного зрения накладывает на моделирующий комплекс требование о возможности моделирования трехмерного пространства и получении его изображения с виртуальной камеры.

Данная работа рассматривает вопросы управления с визуальной обратной связью мобильным роботом с двумя степенями свободы, а также вопросы проверки полученных соотношений в среде виртуальной реальности.

## Постановка задачи

Рассмотрим двухколесного робота с возможностью поворота на месте. Камеру на роботе данного типа логично расположить таким образом, чтобы ее оптическая ось совпадала с осью поступательной степени свободы робота.

Введем следующие системы координат. Пусть  $Ox^*y^*z^*$  — неподвижная система координат, связанная с Землей,  $O_rxyz$  — система координат, соответствующая роботу, а  $O_cx'y'z'$  — система координат камеры. Все системы координат являются правыми.

Математическая модель кинематики колесного робота, движущегося на плоскости, имеет вид

$$\begin{cases} \dot{x}^* = v \cos(\phi), \\ \dot{y}^* = v \sin(\phi), \\ \dot{\phi} = \omega, \end{cases}$$

где  $v$  — продольная скорость робота,  $\omega$  — угловая скорость робота,  $\phi$  — угол между скоростью робота и осью  $Ox^*$ .

Задача заключается в позиционировании робота относительно некоторого объекта, находящегося в области видимости камеры, на основе использования визуальной информации в контуре обратной связи. Также необходимо обеспечить ряд требований к динамике управляемого движения, в частности — устойчивость к влиянию внешнего постоянного возмущения. Для проверки синтезированных регуляторов необходимо спроектировать и реализовать комплекс виртуальной реальности.

## Обзор литературы

Одна из первых работ об управлении с визуальной обратной связью [2] была написана в 1979 году, однако применение описанных методов к реальным задачам управления долгое время было затруднительным из-за вычислительной сложности задач компьютерного зрения. В статье [1] обобщаются существующие на тот момент подходы и на их основе синтезируются системы управления по скорости, а также оцениваются их качества.

Этот подход нашел развитие в работах [3] и [4], где были получены результаты для мобильных роботов, учитывающие динамику движения объекта управления. Однако решаемая авторами задача не позволяет воспользоваться современными методами компьютерного зрения и соответственно оценить влияние увеличения объемов обрабатываемых данных как на быстродействие системы, так и на ее точность.

В открытой литературе практически не встречаются работы, учитывающие динамику движения объектов управления при решении задач IBVS в различных трехмерных симуляционных комплексах.

# Глава 1. Разработка закона управления колесным роботом с использованием алгоритмов компьютерного зрения

Схемы управления на основе визуальной информации опираются на уравнение

$$\dot{s} = L_s v', \quad (1)$$

в котором  $s$  представляет собой набор геометрических признаков, чья производная по времени  $\dot{s}$  связана с пространственной скоростью  $v'$  камеры через матрицу взаимодействия  $L_s$ . Схемы управления строятся с целью минимизации ошибки между текущим значением вектора визуальных особенностей  $s$  и его желаемым значением  $s^*$ :  $e = s - s^*$ . В случае экспоненциального закона уменьшения ошибки ( $\dot{e} = -\lambda e$ ) классическая пропорциональная схема управления определяется как

$$v' = -\lambda \widehat{L}_e^+ e,$$

где  $\widehat{L}_e^+$  является аппроксимацией псевдообратной матрицы к  $L_e$ , и  $L_e$  определяется из уравнения

$$\dot{e} = L_e v'. \quad (2)$$

Так как система координат камеры и система координат робота жестко связаны, между скоростью камеры  $v' = (v'_x, v'_y, v'_z, \omega'_x, \omega'_y, \omega'_z)$  и скоростью робота  $v = (v_x, v_y, v_z, \omega_x, \omega_y, \omega_z)$  в их собственных системах координат будет иметь место соотношение

$$v_x = v'_z, \quad \omega_z = -\omega'_y. \quad (3)$$

В статье [1] показано, что для трехмерной точки с координатами  $X' = (X', Y', Z')$ , проецируемой на изображении в двумерную точку с координата-

ми  $\mathbf{x}' = (x', y')$ , и камеры со скоростью  $\mathbf{v}'$  верно следующее:

$$\begin{cases} \dot{x}' = -v'_x/Z' + x'v'_z/Z' + x'y'\omega'_x - (1 + x'^2)\omega'_y + y'\omega'_z, \\ \dot{y}' = -v'_y/Z' + y'v'_z/Z' + (1 + y'^2)\omega'_x - x'y'\omega'_y - x'\omega'_z. \end{cases} \quad (4)$$

Из выражений (3), (4) и ограничений для степеней свободы следует, что координаты точки на изображении связаны со скоростью робота соотношением

$$\begin{cases} \dot{x} = xv_x/Z + (1 + x^2)\omega_z, \\ \dot{y} = yv_x/Z + xy\omega_z. \end{cases}$$

Приняв в (1)  $\dot{\mathbf{s}} = \dot{\mathbf{x}}$ , для рассматриваемого случая имеем матрицу взаимодействия следующего вида:

$$\mathbf{L}_x = \begin{pmatrix} \frac{x}{Z} & (1 + x^2) \\ \frac{y}{Z} & xy \end{pmatrix}.$$

В следующей главе приводятся результаты моделирования полученного алгоритма управления в среде MatLab.

## Глава 2. Проверка закона управления в среде MatLab

Для проверки работоспособности алгоритма управления в рамках данной работы создан иммитационно-моделирующий комплекс в среде MatLab. Блок-схема комплекса изображена на рисунке 1, а описание блоков приведено в таблице 1. В качестве тестовых задач могут быть рассмотрены простейшие случаи с четырьмя особыми точками. В первом случае камера смещена относительно желаемого положения на 1 по ее оптической оси, а во втором — повернута на угол  $\pi/6$  вокруг оси, перпендикулярной оптической. Изображение с камеры в момент начала движения с отмеченным желаемым положением ключевых точек приведено на рисунке 2. В результате моделирования камера достигает желаемого положения. Полученные графики скоростей изображены на рисунке 3.

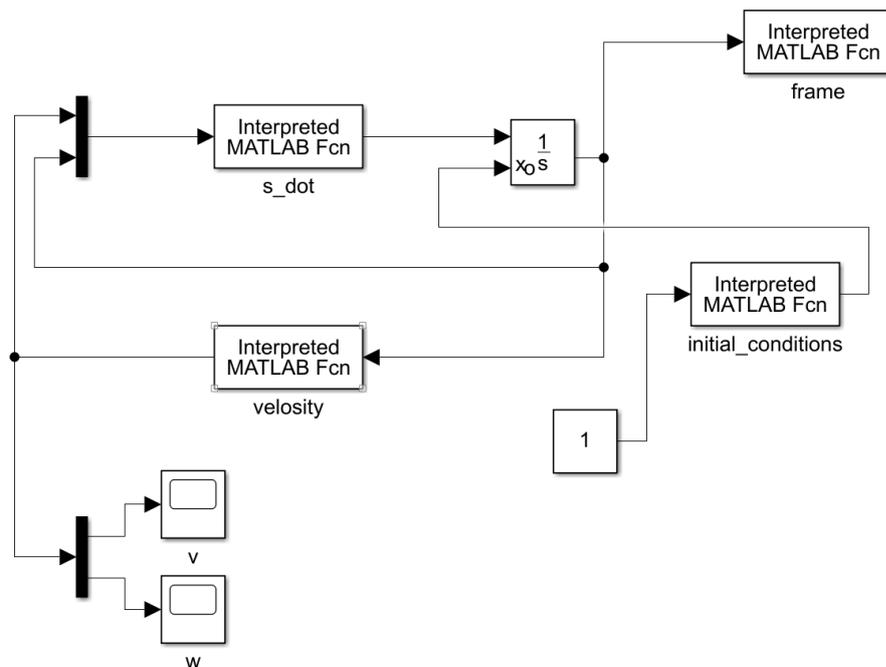
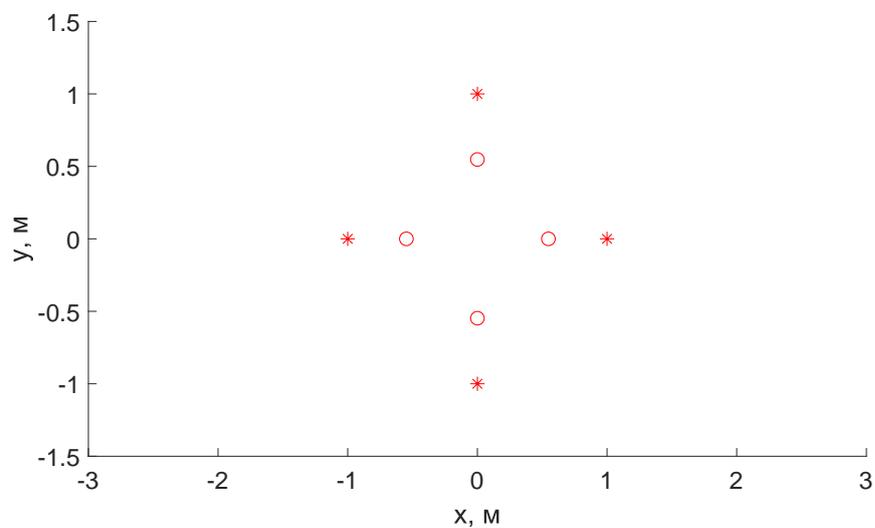


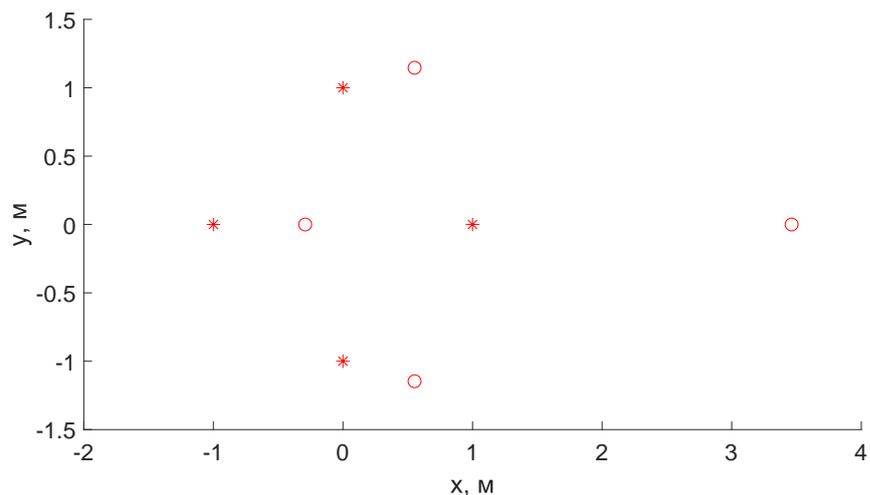
Рис. 1: Схема иммитационно-моделирующего комплекса.

Название блока	Описание функций
initial_conditions	Блок инициализации начальных значений состояния системы
velocity	Блок вычисления скорости управляемого объекта
s_dot	Блок вычисления скорости изменения $s$ и $X$ на основе полученного сигнала управления
v	Блок построения графиков поступательных скоростей объекта
w	Блок построения графиков угловых скоростей объекта

**Таблица 1:** Описание блоков схемы Simulink.

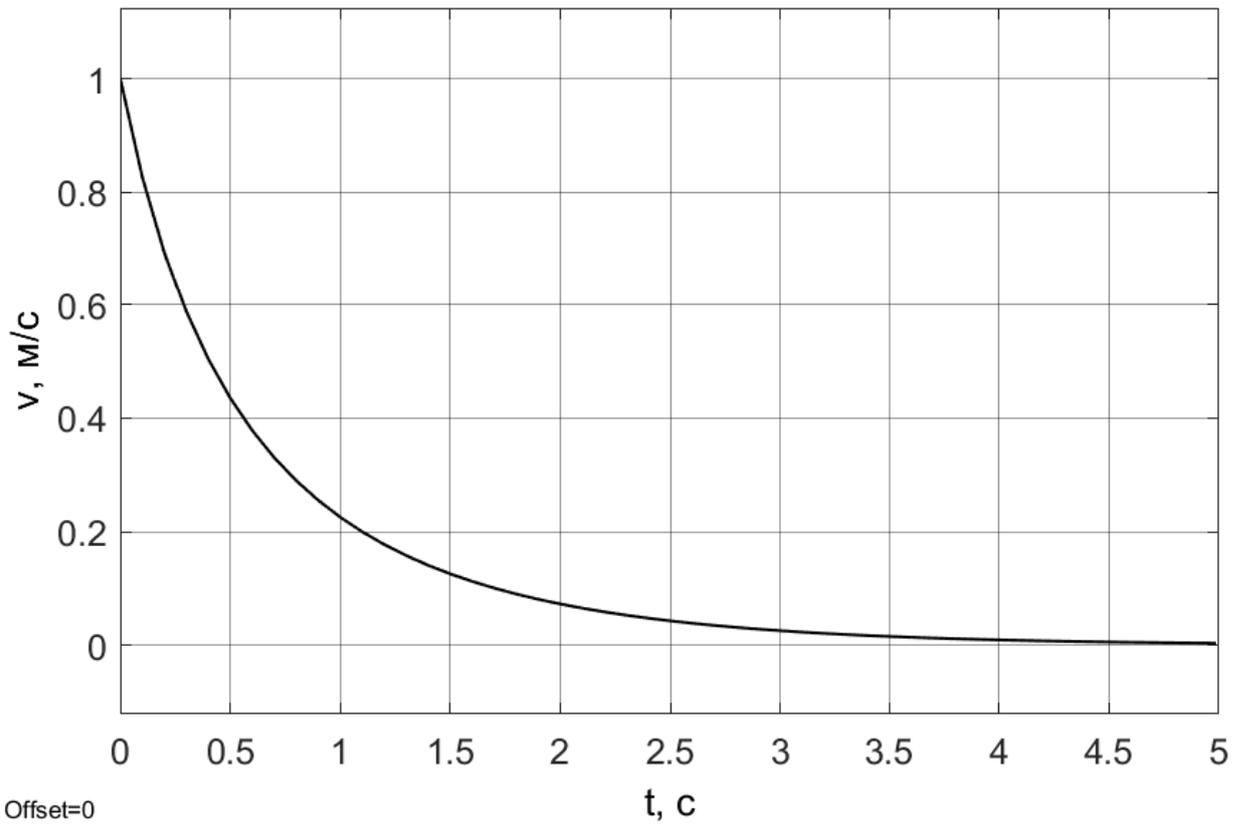


**(a)** Задача со смещением вдоль оптической оси.

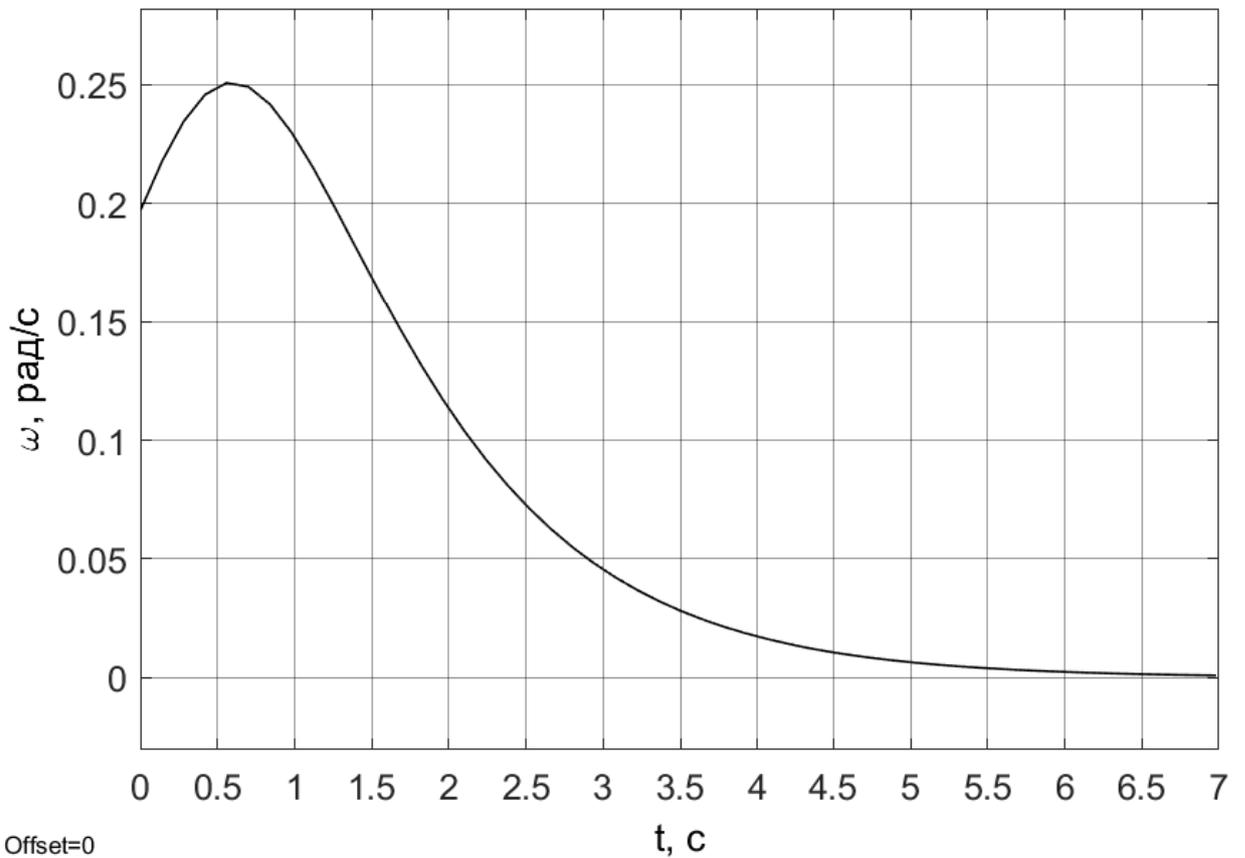


**(b)** Задача с поворотом на угол  $\pi/6$ .

**Рис. 2:** Стартовое (\*) и желаемое (o) положение ключевых точек на изображении с камеры в тестовой задаче в среде MatLab.



(а) График поступательной скорости в случае смещения.



(б) График угловой скорости в случае поворота.

**Рис. 3:** Графики скоростей при выполнении тестовых задач в среде MatLab.

## **Глава 3. Проектирование программного комплекса с элементами виртуальной реальности**

Ключевым моментом моделирования является необходимость смоделировать процесс получения изображения с камеры. Поэтому проектирование моделирующего комплекса было начато с выбора среды 3D моделирования.

### **3.1 Gazebo**

В силу результатов сравнения различных сред симуляционного моделирования, представленных в статье [5], было решено использовать Gazebo.

Gazebo — разрабатываемая организацией Open Source Robotics Foundation среда симуляционного моделирования с открытым исходным кодом, ориентированная на робототехнику. Gazebo может использовать несколько высокопроизводительных физических библиотек, таких как ODE, Bullet. Они обеспечивают реалистичную визуализацию окружающей среды, включая высококачественное освещение, тени и текстуры. Также Gazebo может моделировать датчики, которые «видят» моделируемую среду, например лазерные дальнометры, камеры (включая широкоугольные), датчики в стиле Kinect.

Для осуществления управления объектами в данной среде используется Robot Operating System.

### **3.2 Robot Operating System**

Robot Operating System (ROS) — это фреймворк для программирования роботов, предоставляющий функциональность для распределенной работы. ROS обеспечивает стандартные службы операционной системы, такие как аппаратная абстракция, низкоуровневый контроль устройств, реализация часто используемых функций, передача сообщений между процессами и управление пакетами. ROS основан на архитектуре графов, где обработка данных происходит в узлах, которые могут получать и передавать сообщения между собой.

Язык программирования Python имеет специализированную библиотеку, позволяющую взаимодействовать написанным на нем программам с сетью

ROS2. Существует множество роботов с поддержкой управления при помощи ROS. Учитывая поставленные ранее ограничения на число степеней свободы, в качестве объекта управления был выбран Turtlebot3.

### 3.3 Turtlebot3

TurtleBot3 — компактный модульный программируемый мобильный робот (Рис. 4) с открытым программным и аппаратным обеспечением, построенный на основе сервоприводов Dynamixel серии XM, которые поддерживают несколько режимов управления: по положению, скорости, току и комбинации вышеперечисленных режимов. Характеристики робота приведены в таблице 2. Модель для среды Gazebo, доступная на официальном сайте производителя, имеет физические характеристики, соответствующие реальному роботу.

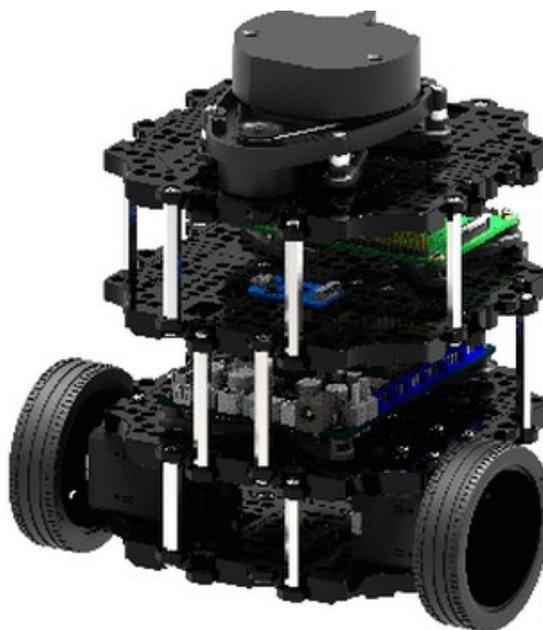


Рис. 4: Turtlebot.

### 3.4 Выбор метода извлечения ключевых точек

Так как поиск и сравнение ключевых точек является отдельной достаточно сложной задачей, решено воспользоваться готовыми алгоритмами из библиотеки алгоритмов OpenCV. Вследствие того, что методы SIFT и SURF

Максимальная скорость поступательного движения	0.22 м/с
Максимальная угловая скорость	2.84 рад/с
Размер (Д × Ш × В)	138 × 178 × 192 мм
Вес в сборе	1 кг
Время работы	2 ч 30 мин
Датчики	<ul style="list-style-type: none"> <li>• 360° лазерный дальномер</li> <li>• Трёхосевой гироскоп</li> <li>• Трёхосевой акселерометр</li> <li>• Трёхосевой магнитометр</li> </ul>

**Таблица 2:** Характеристики робота Turtlebot3 Burger

запатентованы, было решено использовать бесплатные альтернативные методы Oriented FAST and Rotated BRIEF (ORB) и Accelerated KAZE (AKAZE). В [6] проведено сравнение методов, показавшее сопоставимость AKAZE и ORB с SIFT, SURF и между собой по точности и быстродействию. При этом в ряде задач ORB быстрее, но менее точен, чем AKAZE. В связи с тем, что на данном этапе оценка важности быстродействия алгоритма выделения ключевых точек не представляется возможной, было решено произвести выбор по итогам тестового моделирования. OpenCV содержит алгоритм, позволяющий отсеять ложные совпадения при помощи метода RANSAC.

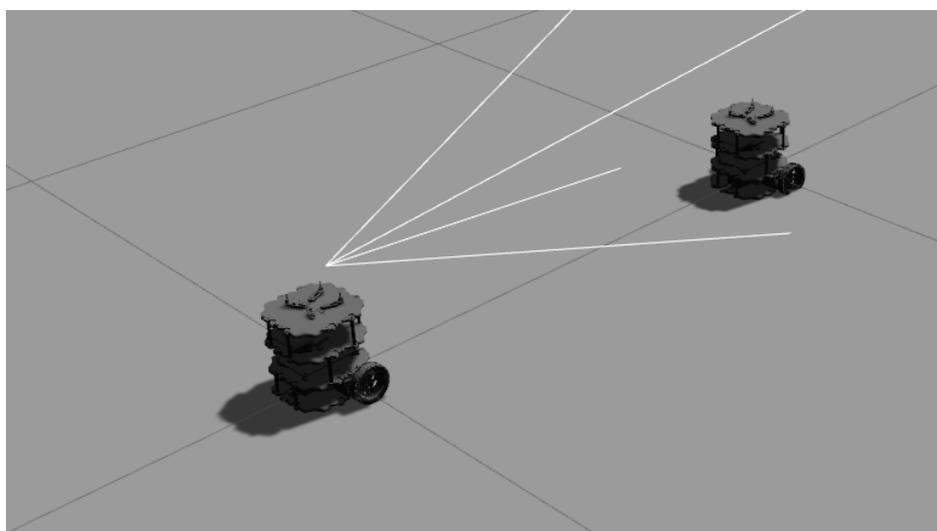
### 3.5 Объединение выбранных частей в комплекс

Ввиду отсутствия готовых решений для связи выбранных элементов комплекса в рамках настоящей работы был написан ряд программ на языке Python3.

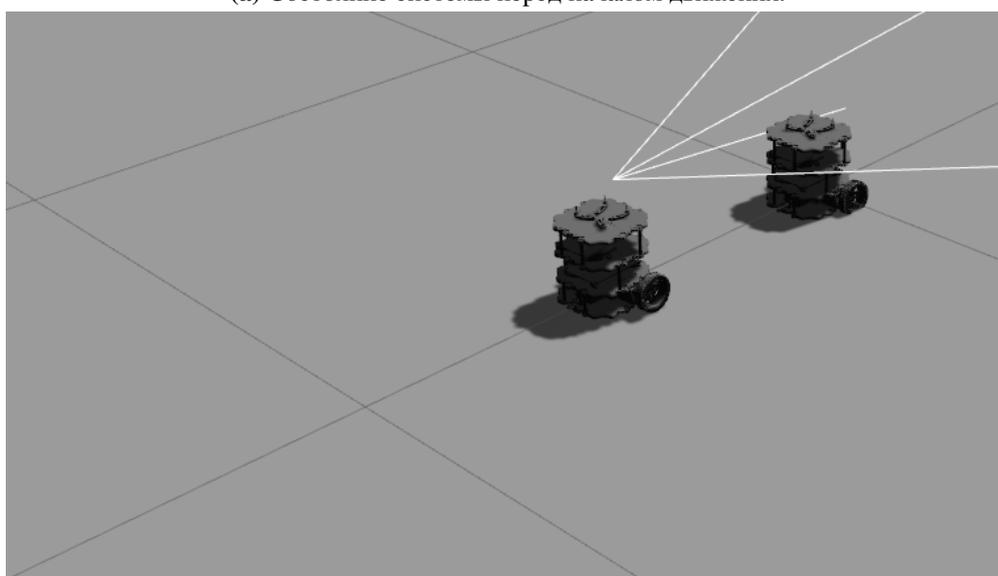
В ходе тестирования производительности алгоритмов AKAZE и ORB при запущенном моделировании в Gazebo был обнаружен недостаток производительности процессора для выполнения обработки в реальном времени. Проблема была частично решена путем использования технологии CUDA, позволяющей производить параллельные вычисления на графическом процессоре.

## Глава 4. Тестирование программного комплекса

Для проверки работоспособности программного комплекса и выбора между предложенными методами выделения ключевых точек рассматривается случай, когда робот сдвинут относительно начального положения вдоль оптической оси камеры на фиксированное расстояние (Рис. 5). Так как управление осуществляется при помощи ключевых точек, в зону обзора камеры был помещен еще один статичный робот, относительно которого и будет происходить позиционирование. Начальное расстояние до цели — 1 метр, желаемое — 0.5 метра.



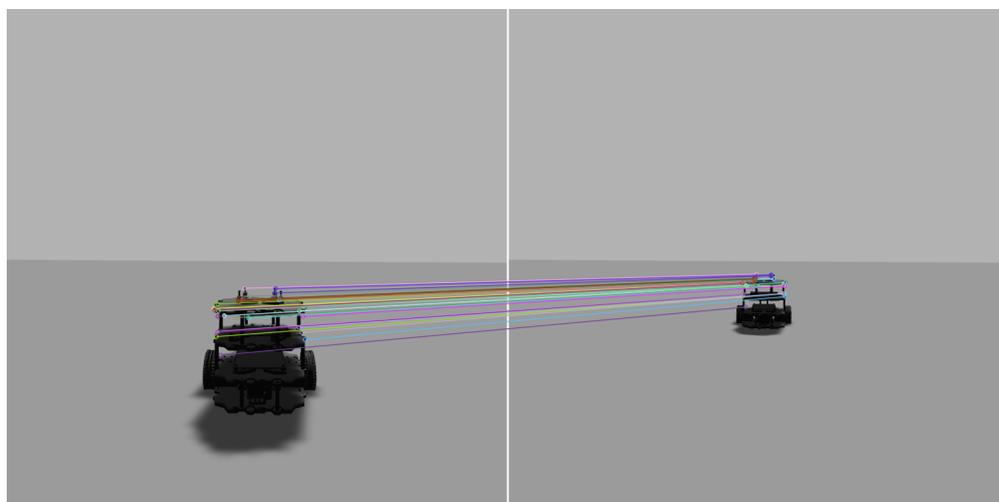
(a) Состояние системы перед началом движения.



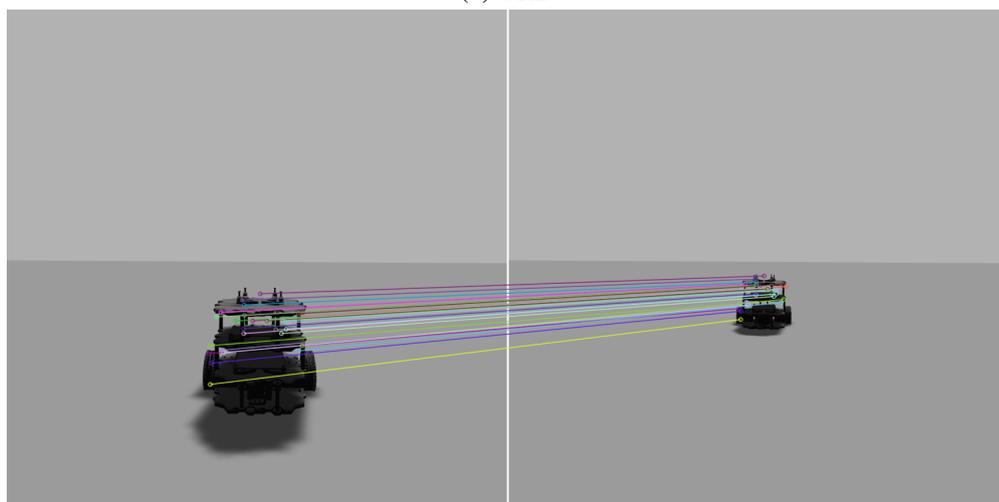
(b) Желаемое состояние системы.

**Рис. 5:** Тестовая система из двух роботов.

На рисунке 6 показаны изображения в начальном и желаемом положениях. Также на рисунке визуализирован результат выделения ключевых точек, их сравнения и применения RANSAC.



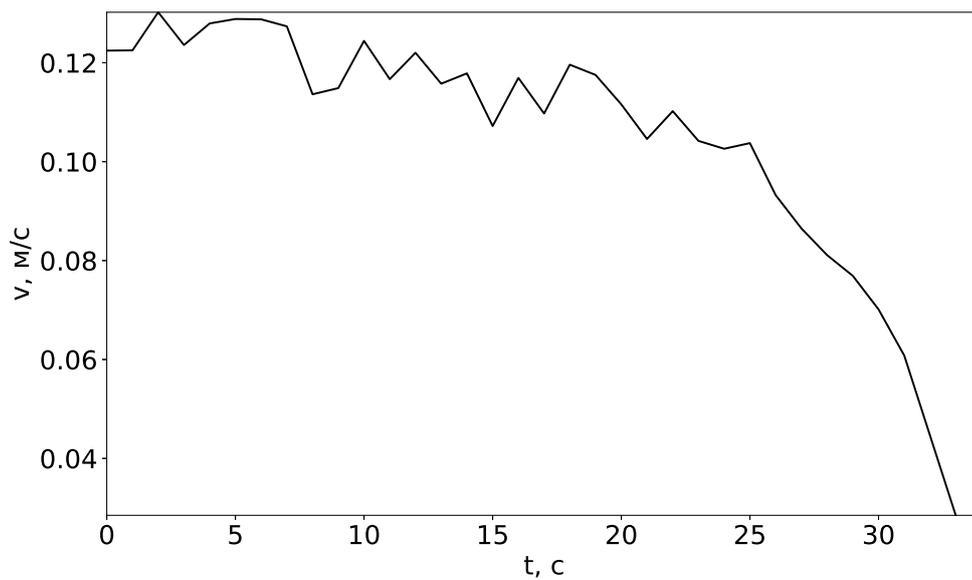
(a) ORB.



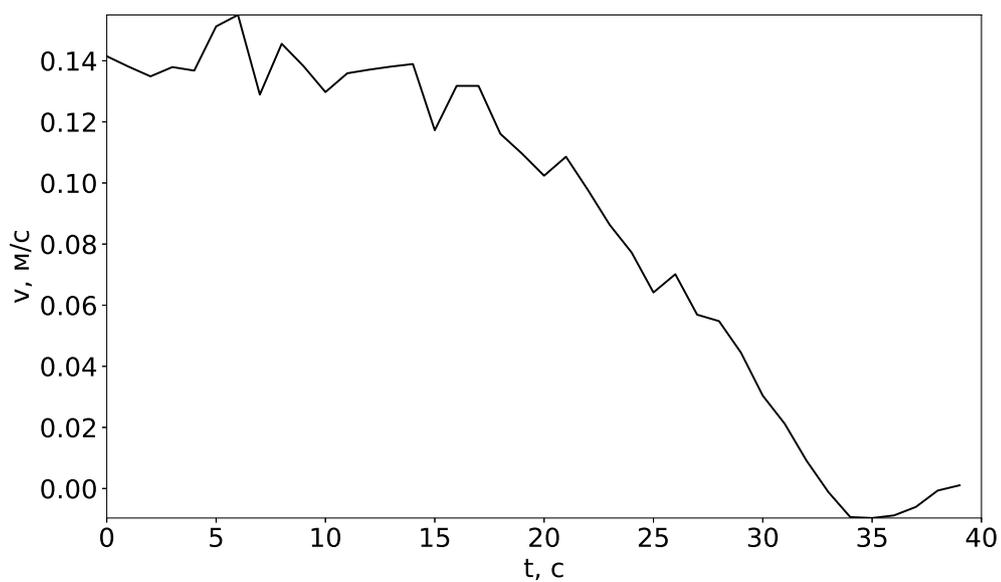
(b) AKAZE.

**Рис. 6:** Изображения с камеры в стартовом и желаемом положениях.

В результате выполнения тестовой задачи были получены графики скорости, представленные на рисунке 7. Ошибка позиционирования в обоих случаях составила меньше 0.1%. На основе полученных результатов можно сделать вывод о том, что AKAZE проигрывает ORB по скорости. Также AKAZE допускает флуктуации в окрестности ненулевой точки до достижения требуемой точности.



(a) ORB.



(b) AKAZE.

**Рис. 7:** Графики скоростей в ходе выполнения тестовой задачи.

## Глава 5. Введение учета динамики

### 5.1 Модель робота с учетом динамики

Динамику робота в общем случае можно описать следующей системой уравнений [7]:

$$\begin{aligned}\dot{\mathbf{v}} &= \mathbf{A}\mathbf{v} + \mathbf{B}\boldsymbol{\tau} + \boldsymbol{\tau}_e, \\ \dot{\boldsymbol{\eta}} &= \mathbf{R}(\boldsymbol{\eta})\mathbf{v},\end{aligned}$$

где  $\boldsymbol{\eta} = (x, y, \phi)$  — вектор кинематических параметров робота;  $\mathbf{A}$  — диагональная матрица коэффициентов трения;  $\boldsymbol{\tau}$  — вектор управляющих воздействий;  $\mathbf{B}$  — матрица коэффициентов при управляющих воздействиях;  $\boldsymbol{\tau}_e$  — вектор внешних возмущений.

Матрица поворота для рассматриваемого случая имеет вид

$$\mathbf{R}(\boldsymbol{\eta}) = \begin{pmatrix} \cos(\phi) & 0 \\ \sin(\phi) & 0 \\ 0 & 1 \end{pmatrix}.$$

Учитывая специфику поставленной задачи, на основе подхода, описанного в [3], предлагается использовать многоцелевую структуру управления в форме

$$\begin{aligned}\dot{\mathbf{z}}_v &= \mathbf{A}\mathbf{z}_v + \mathbf{B}\boldsymbol{\tau} + \mathbf{R}^T(\boldsymbol{\eta})\mathbf{K}_1(\boldsymbol{\eta} - \mathbf{z}_\eta), \\ \dot{\mathbf{z}}_\eta &= \mathbf{R}(\boldsymbol{\eta})\mathbf{z}_v + \mathbf{K}_2(\boldsymbol{\eta} - \mathbf{z}_\eta), \\ \dot{\mathbf{z}}_e &= \mathbf{L}\mathbf{z}_v + \mathbf{H}_e(\mathbf{e} - \mathbf{z}_e), \\ \dot{\mathbf{p}} &= \boldsymbol{\alpha}\mathbf{p} + \boldsymbol{\beta}_\eta(\boldsymbol{\eta} - \mathbf{z}_\eta) + \boldsymbol{\beta}_e(\boldsymbol{\eta} - \mathbf{z}_\eta), \\ \dot{\boldsymbol{\zeta}} &= \boldsymbol{\gamma}\mathbf{p} + \boldsymbol{\mu}_\eta(\boldsymbol{\eta} - \mathbf{z}_\eta) + \boldsymbol{\mu}_e(\boldsymbol{\eta} - \mathbf{z}_\eta), \\ \boldsymbol{\tau} &= -\mathbf{K}_e\mathbf{z}_e - \mathbf{K}_v\mathbf{z}_v + \boldsymbol{\zeta}.\end{aligned}$$

Первые три уравнения характеризуют асимптотический наблюдатель, используемый для оценки компонентов, недоступных для прямого измерения, а также позволяющий ввести в систему динамический корректор, который описывается четвертым и пятым уравнением. Динамический корректор определяет реакцию системы управления на внешние возмущения. В зависимости от поставленной задачи он позволяет либо минимизировать влияние

возмущения на динамику системы (в данной работе используется этот вариант), либо минимизировать управляющие воздействия, например, в целях экономии топлива или ресурса актуаторов. Также в состав системы входят следующие регулируемые параметры:

- а) матрицы  $\mathbf{K}_v$  и  $\mathbf{K}_e$  базового закона управления;
- б) матрицы  $\mathbf{K}_1$ ,  $\mathbf{K}_2$  и  $\mathbf{H}_e$  асимптотического наблюдателя;
- в) матрицы  $\alpha$ ,  $\beta_\eta$ ,  $\beta_e$ ,  $\gamma$ ,  $\mu_\eta$ ,  $\mu_e$ .

Поиск настраиваемых параметров на основе требований к динамике движения и является задачей синтеза многоцелевой структуры. Важной особенностью многоцелевой структуры является то, что ее настраиваемые элементы можно искать последовательно [8]. Часто удобно формализовать поиск настраиваемых элементов в качестве задач оптимизации на соответствующих допустимых множествах.

Рассмотрим задачу поиска матриц  $\mathbf{K}_v$  и  $\mathbf{K}_e$  для базового закона управления

$$\tau = -\mathbf{K}_e e - \mathbf{K}_v v. \quad (5)$$

В соответствии с методом функций Ляпунова введем квадратичную форму

$$V = \frac{1}{2} e^T e + \frac{1}{2} v^T v.$$

С учетом формул (2), (5) и динамической модели, производная  $V$  принимает вид

$$\dot{V} = -v^T \mathbf{A} v + v^T (\mathbf{L}^T e + \mathbf{B} \tau).$$

Из этого следует, что если задать  $\mathbf{K}_e = \mathbf{B}^{-1} \mathbf{L}^T$ , то стабильность замкнутой системы обеспечивается при любом таком  $\mathbf{K}_v$ , при котором для матрицы  $\mathbf{A} - \mathbf{B} \mathbf{K}_v$  выполняется условие устойчивости Гурвица.

Как показано в [9], глобальная асимптотическая устойчивость нулевого положения равновесия  $(\eta - z_\eta) = 0$ ,  $(e - z_e) = 0$  гарантируется, если матрицы  $\mathbf{K}_1$  и  $\mathbf{K}_2$  — диагональные и положительно определенные. Матрица  $\mathbf{H}_e$  также должна быть положительно определенной.

На последнем шаге синтеза нам необходимо найти  $\alpha, \beta_\eta, \beta_e, \gamma, \mu_\eta, \mu_e$ .  
Перепишем динамический корректор в форме

$$\zeta = \mathbf{F}_1(s)(\eta - \mathbf{z}_\eta) + \mathbf{F}_2(s)(\mathbf{e} - \mathbf{z}_e),$$

где

$$\begin{aligned}\mathbf{F}_1(s) &= \gamma(\mathbf{E}s - \alpha)^{-1}\beta_\eta + \mu_\eta, \\ \mathbf{F}_2(s) &= \gamma(\mathbf{E}s - \alpha)^{-1}\beta_e + \mu_e.\end{aligned}$$

Форма матриц  $\mathbf{F}_1(s)$  и  $\mathbf{F}_2(s)$  зависит от требований к динамике в соответствующих режимах. Учитывая постановку задачи, необходимо обеспечить астатическое свойство системы управления. Предполагая, что в равновесном положении  $\eta = \eta_0$  и  $\mathbf{L} = \mathbf{L}_0$  и проводя дифференцирования, аналогичные [10], получим, что матрицы переноса должны удовлетворять следующим условиям:

$$\begin{aligned}\mathbf{F}_1(0) &= -\mathbf{R}^\top(\eta_0)\mathbf{K}_1, \\ \mathbf{F}_2(0) &= -\mathbf{K}_e + \mathbf{A}\mathbf{T} + \mathbf{K}_v\mathbf{T}, \\ \mathbf{T} &= -(\mathbf{L}_0^\top\mathbf{L}_0)^{-1}\mathbf{L}_0^\top\mathbf{H}_e.\end{aligned}$$

Исходя из возможностей моделирующего комплекса, в качестве управляющего воздействия выбран вектор моментов  $\boldsymbol{\tau} = (\tau_v, \tau_\omega)$ .

## 5.2 Моделирование

Динамика движения робота описывается следующими матрицами:

$$\mathbf{A} = \begin{pmatrix} -0.2451 & 0 \\ 0 & -0.2368 \end{pmatrix}, \quad \mathbf{B} = \begin{pmatrix} 0.2350 & 0 \\ 0 & 0.2384 \end{pmatrix}.$$

Для моделирования внешнего постоянного возмущения к роботу был приложен момент, соответствующий воздействию  $\boldsymbol{\tau}_e = (-0.03, -0.12)$ . Матрицы асимптотического наблюдателя зададим так:

$$\mathbf{K}_1 = 10\mathbf{E}_{3 \times 3}, \quad \mathbf{K}_2 = 5\mathbf{E}_{3 \times 3}, \quad \mathbf{H}_e = 0.5\mathbf{E}_{k \times k}.$$

Особенности методов извлечения точек не могут гарантировать фикс-

сированную длину  $k$  вектора  $e$ , поэтому матрица  $H_e$  вычисляется на каждой итерации для соответствия размерностей.

В силу необходимости получения достаточного количества данных для оценки качества процесса управления выбрана матрица

$$K_v = \begin{pmatrix} 2 & 0 \\ 0 & 8 \end{pmatrix}.$$

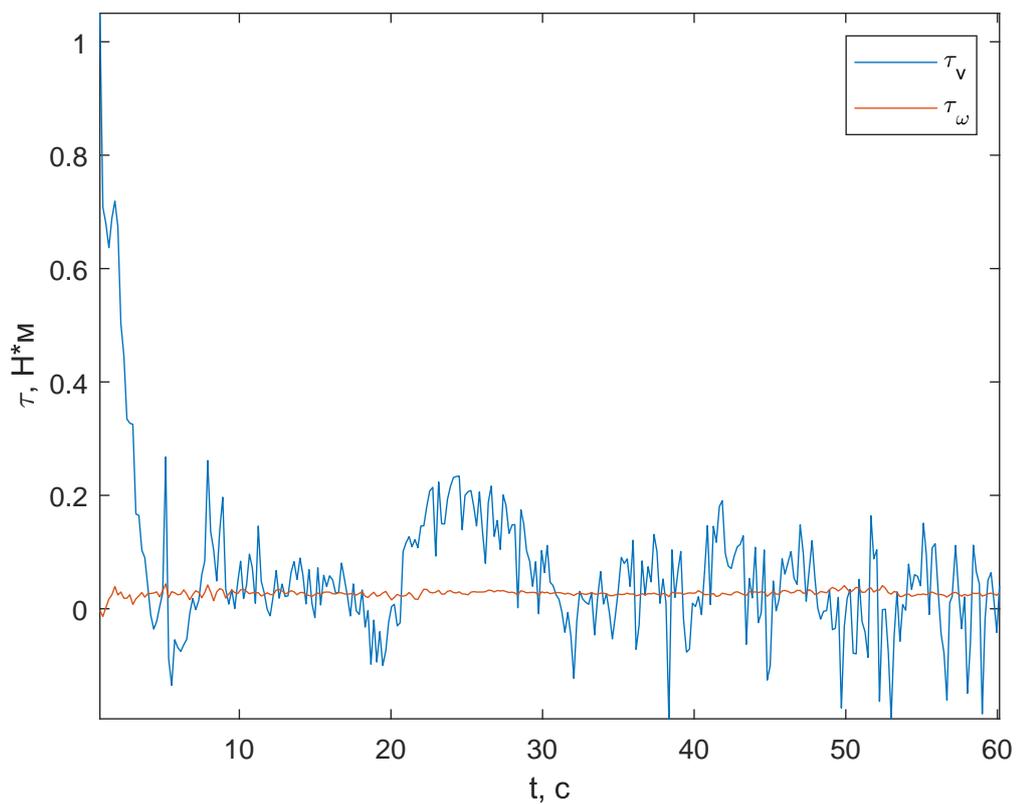
Коэффициенты динамического корректора определим следующим образом:

$$\alpha = -E_{2 \times 2}, \quad \gamma = E_{2 \times 2}, \quad \mu_\eta = \mu_e = 0.$$

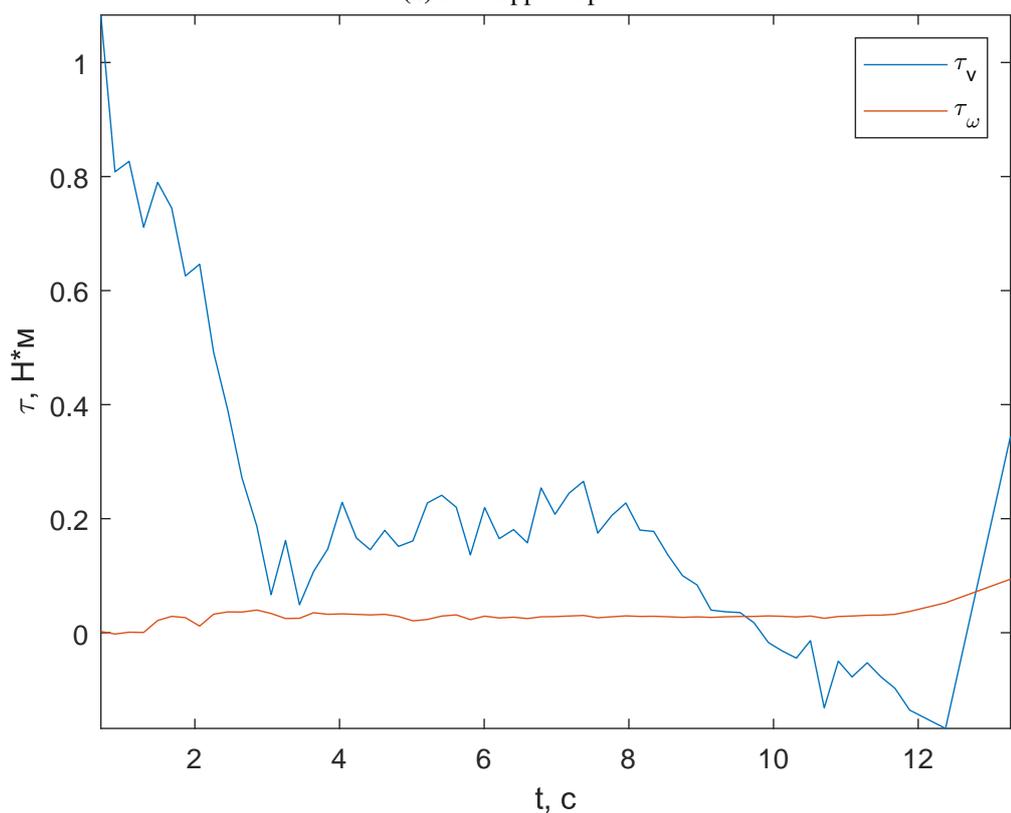
Для поставленной ранее задачи с увеличенным расстоянием до цели на 1 метр был проведен ряд экспериментов:

1. с использованием алгоритма ORB, но без использования корректора,
2. с использованием корректора и алгоритма ORB,
3. с использованием алгоритма AKAZE, но без использования корректора,
4. с использованием корректора и алгоритма AKAZE.

Результаты экспериментов представлены на рисунках 8-15. График траектории показан в системе координат цели, для нее управляемый робот в начале движения находится сзади на расстоянии 2 метра, а желаемое расстояние составляет 0.5 метра.

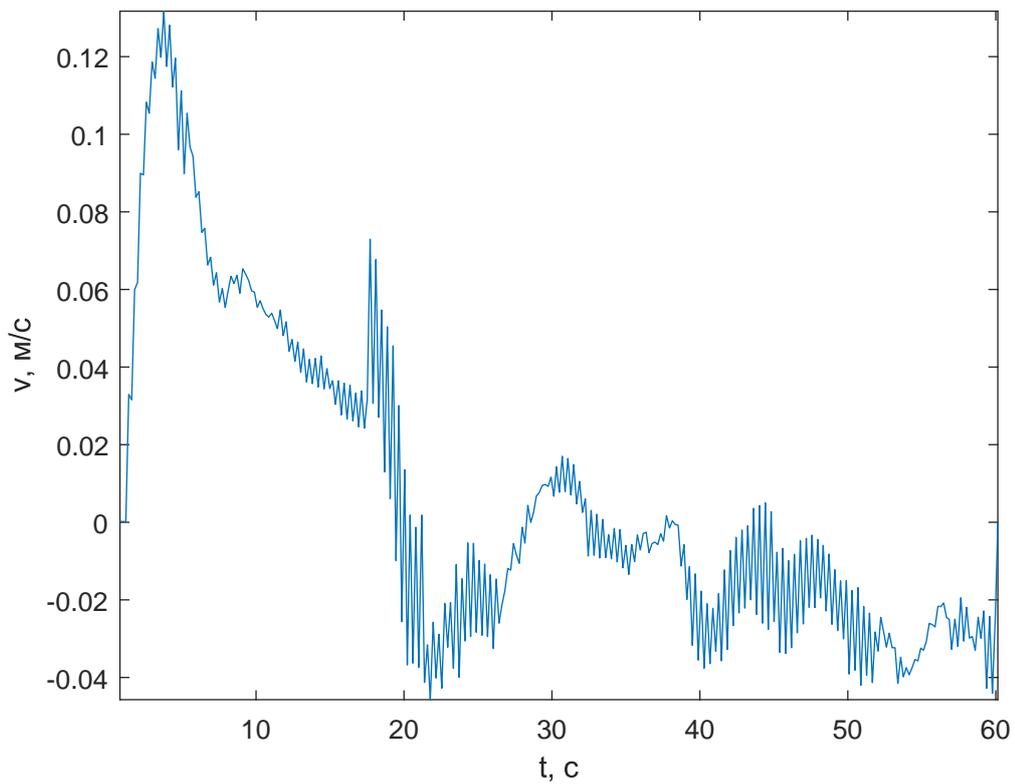


(a) Без корректора.

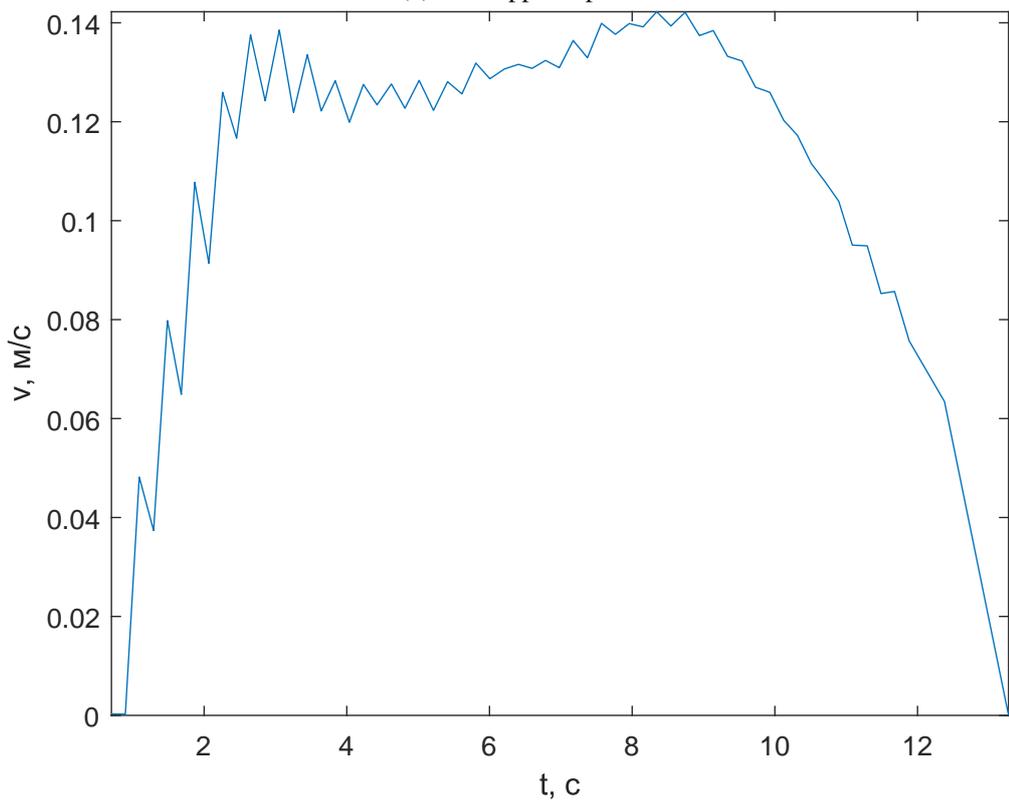


(b) С использованием корректора.

**Рис. 8:** Графики моментов при использовании ORB. Внешнее возмущение заставляет систему без корректора изменять моменты приводов с большой частотой, что может привести к быстрому износу двигателя. Робот под управлением системы с корректором достигает цели за 13 секунд, в то время как система без него продолжает движение до потери цели из виду.

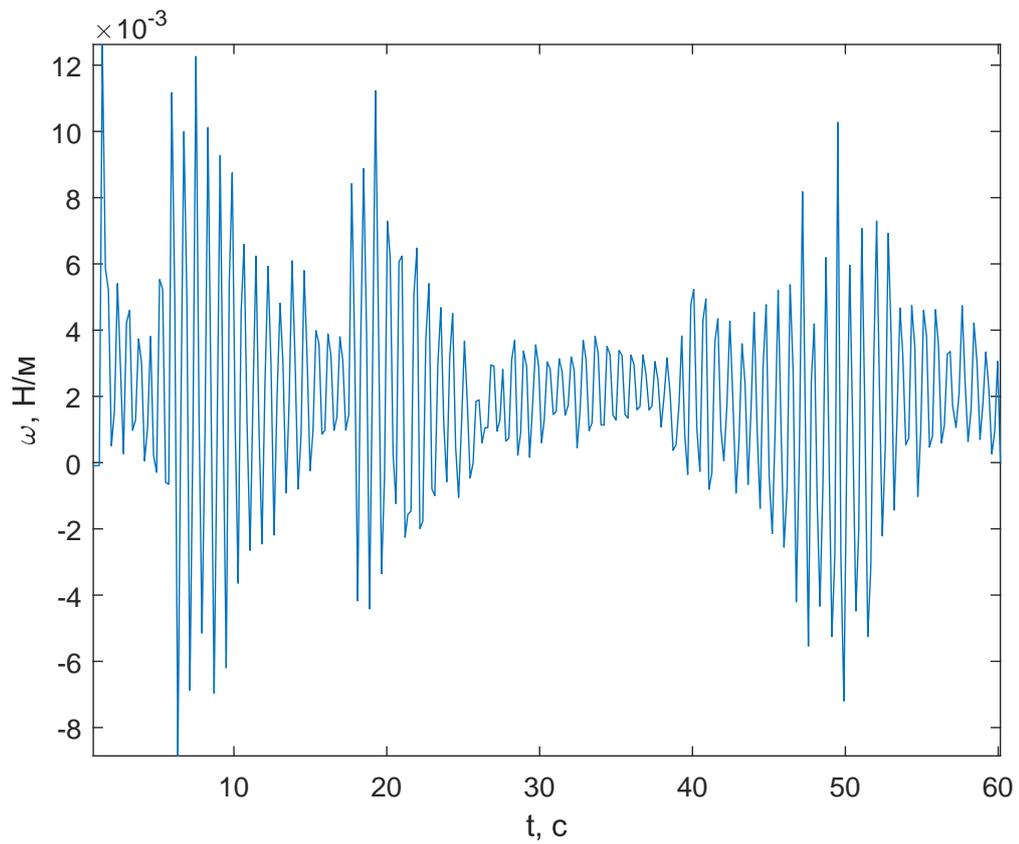


(а) Без корректора.

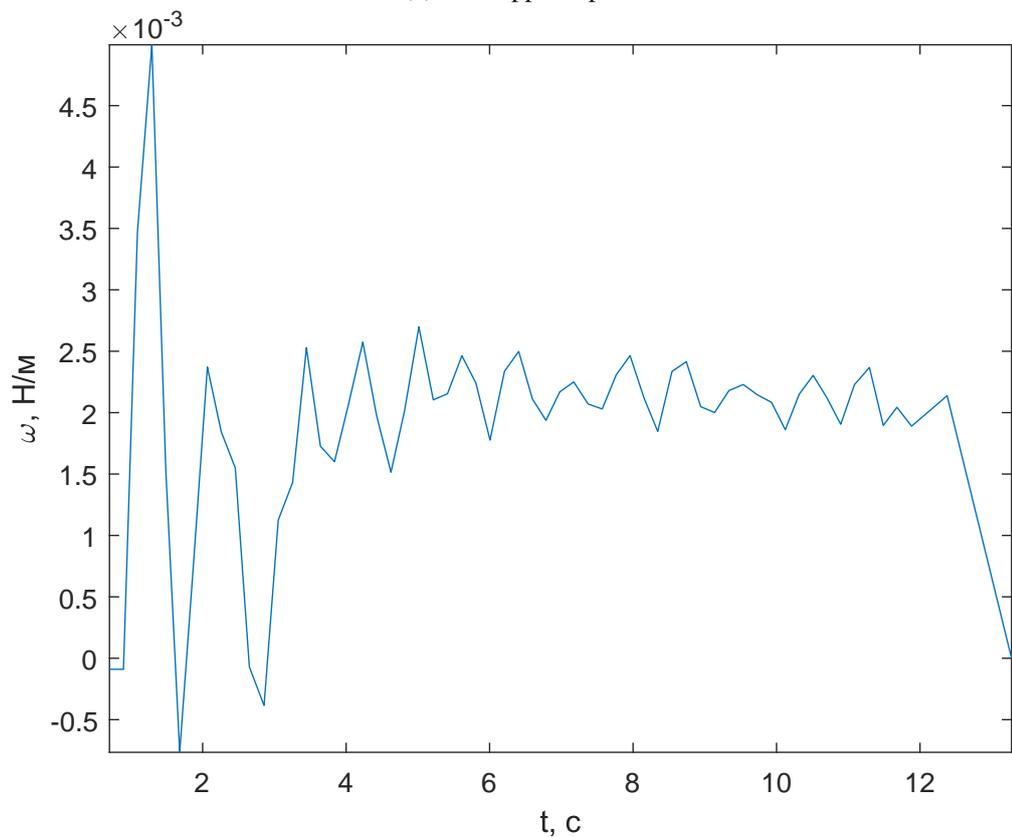


(б) С применением корректора.

**Рис. 9:** Графики линейных скоростей при использовании ORB. Идеальность приводов позволяет почти мгновенно обрабатывать входной момент, что приводит к наследованию проблем управляющего воздействия. Наличие рывков в первой половине графиков обусловлена недостаточной частотой обновления при достаточно быстром изменении скорости.

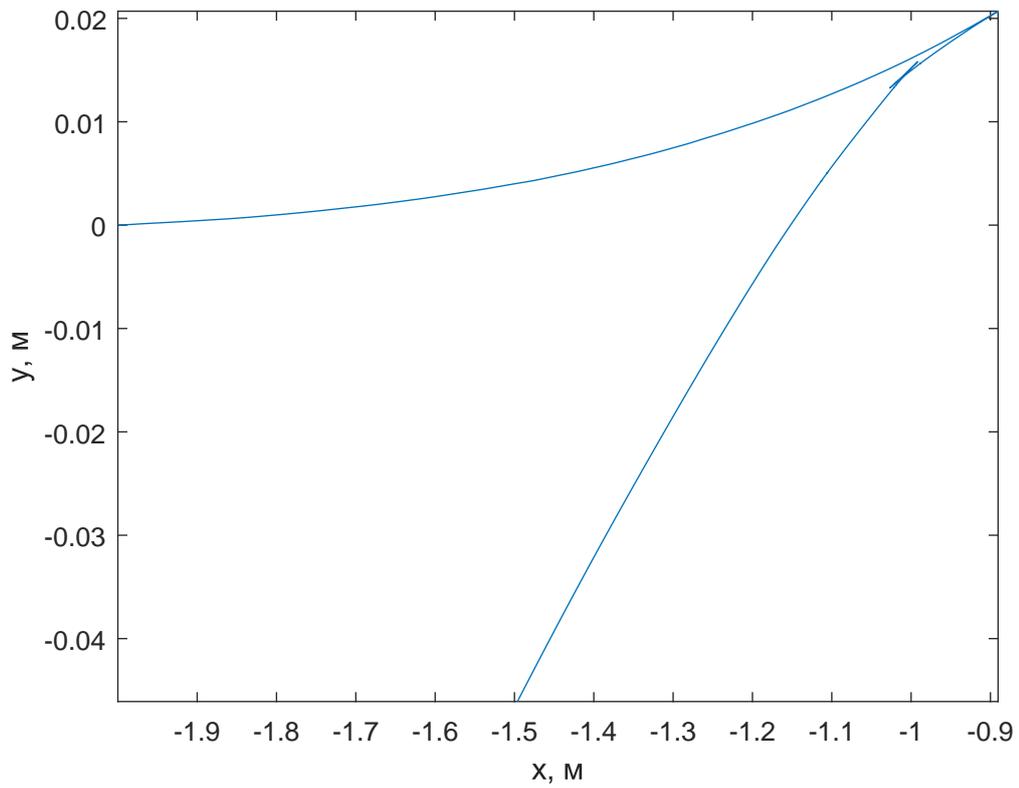


(a) Без корректора.

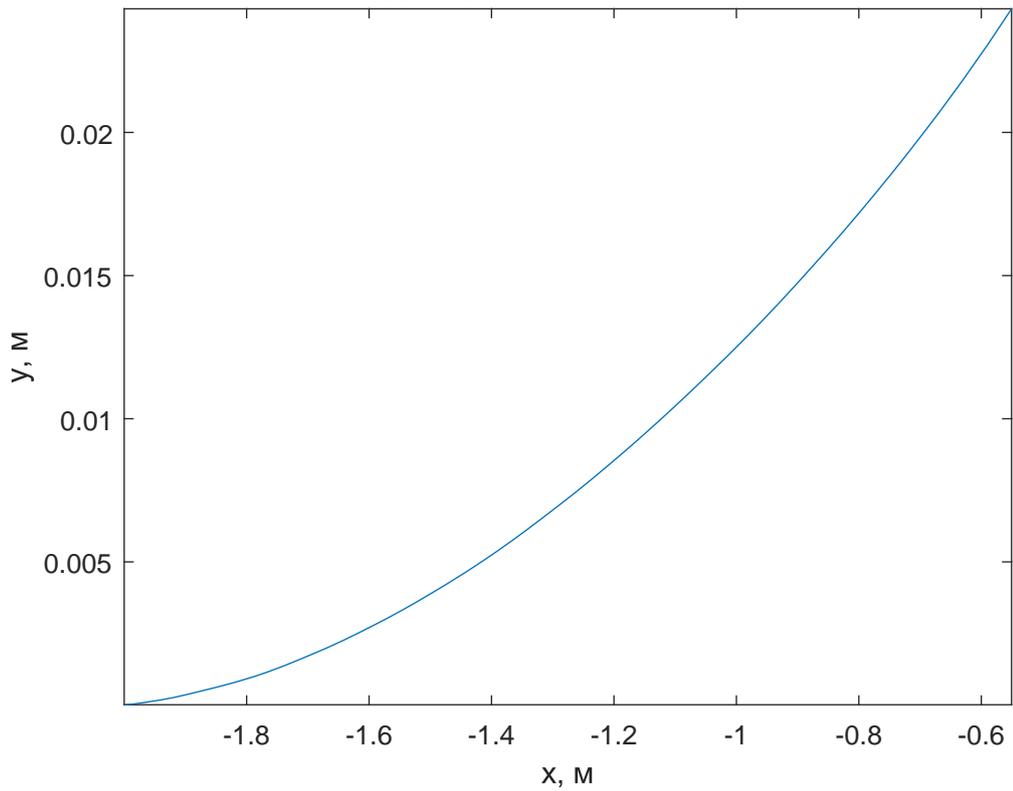


(b) С применением корректора.

**Рис. 10:** Графики угловых скоростей при использовании ORB. Амплитуда колебаний угловой скорости без корректора больше в 2 раза. Высокая частота изменения угловой скорости повышает требования к характеристикам используемой камеры.

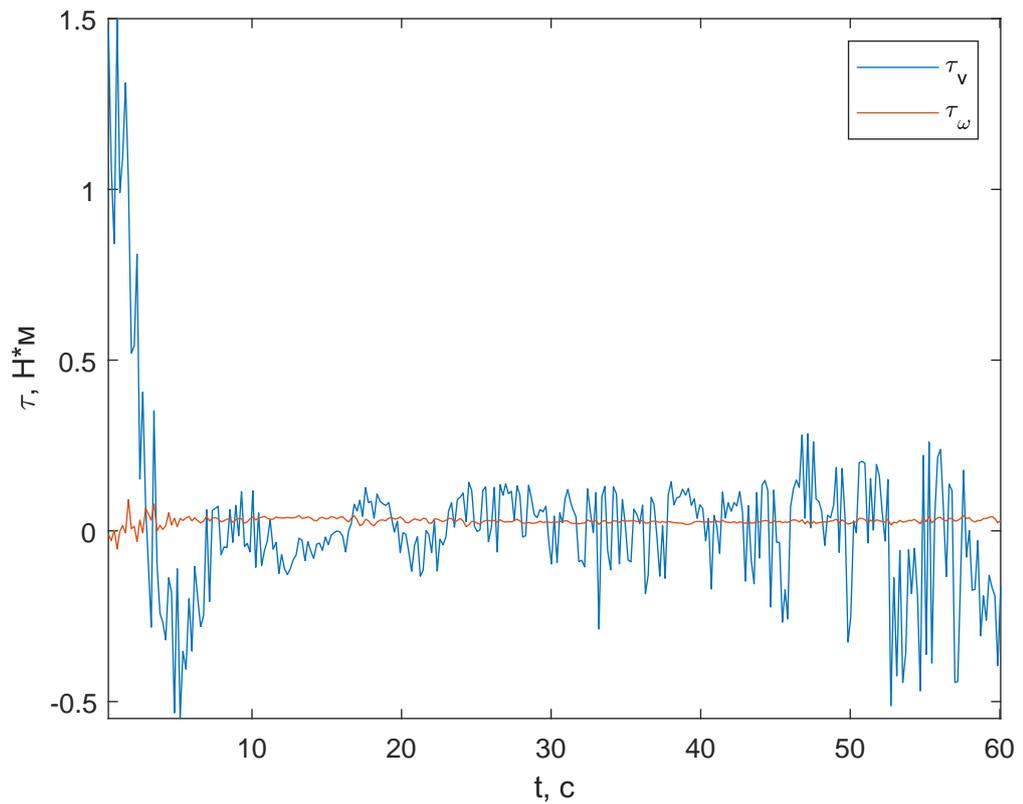


(a) Без корректора. Задача не выполнена.

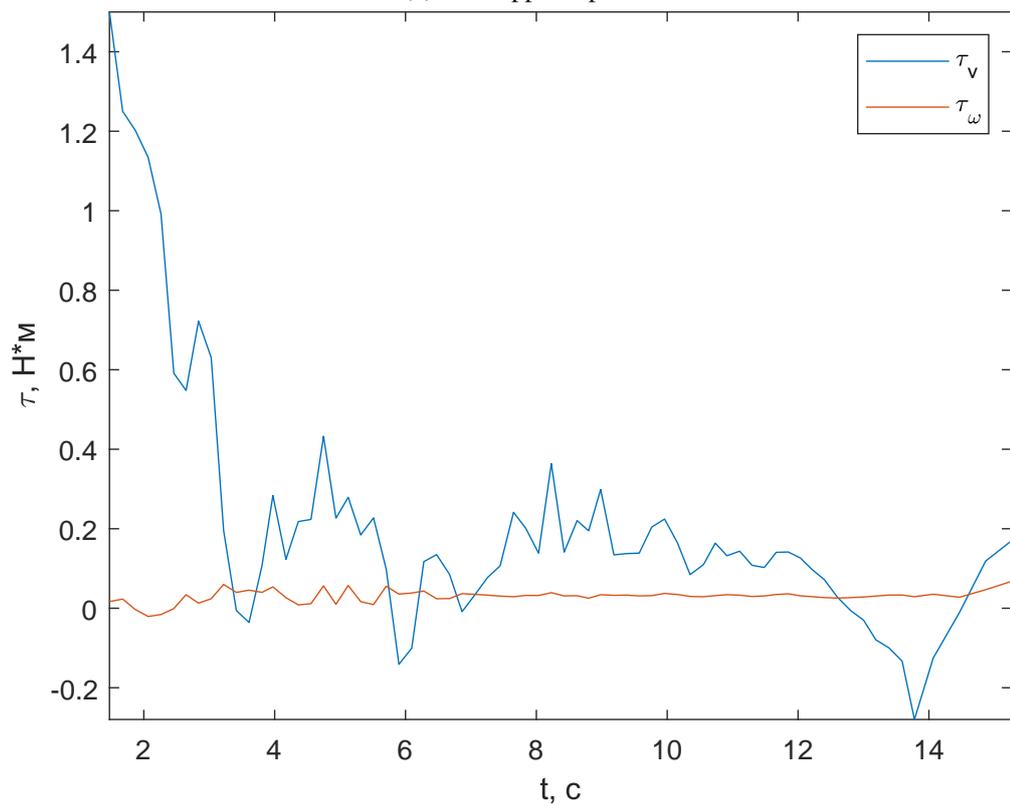


(b) С применением корректора. Точка остановки:  $(-0.55, 0.024)$ .

**Рис. 11:** Траектории при использовании ORB. Робот без корректора не смог достигнуть желаемого положения. Траектория, полученная под управлением системы с корректором, не прямая, но находится в пределах допустимого отклонения. Отклонение по оси  $y$  связано с неточностью определения ключевых точек.

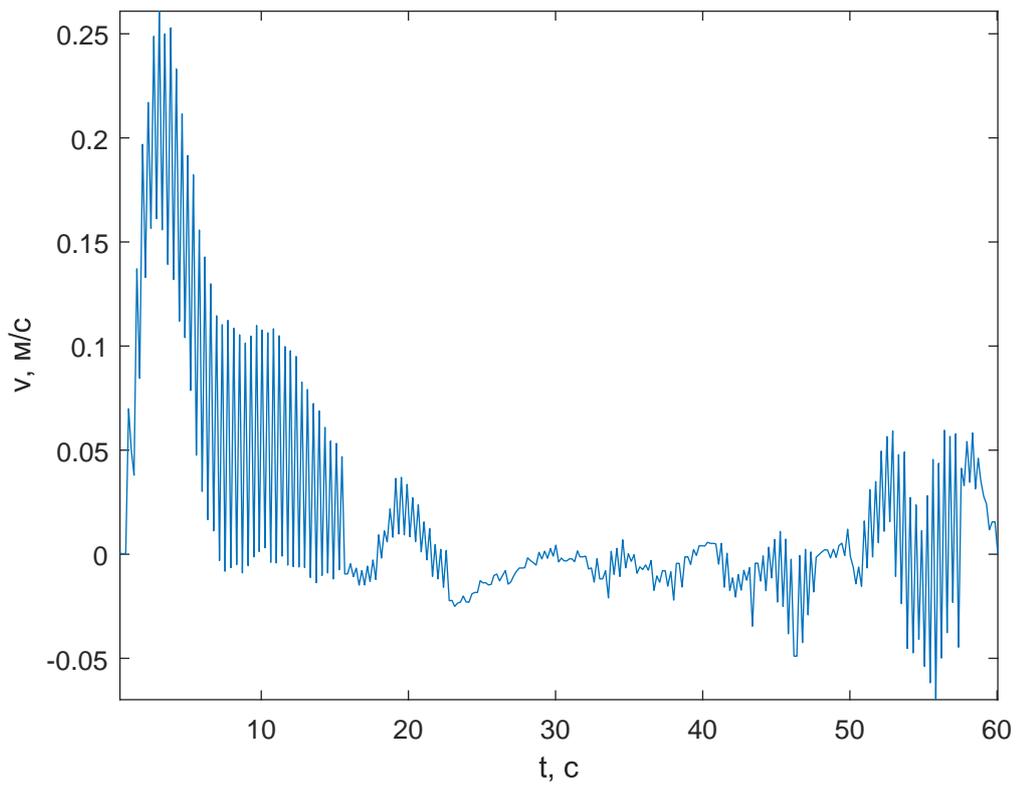


(a) Без корректора.

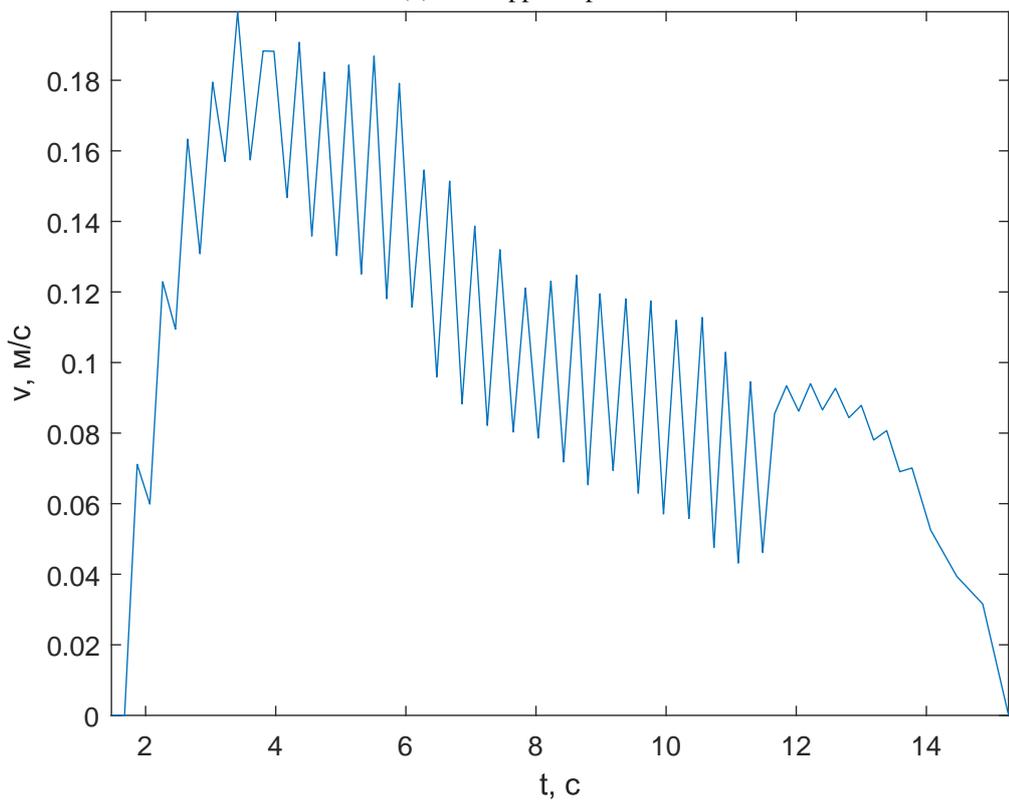


(b) С применением корректора.

**Рис. 12:** Графики моментов при использовании AKAZE. Изменение алгоритма выделения ключевых точек не дало качественного изменения картины для системы без корректора. При этом значительно увеличились значения всплесков моментов, что в реальной системе может привести к сгоранию двигателя.

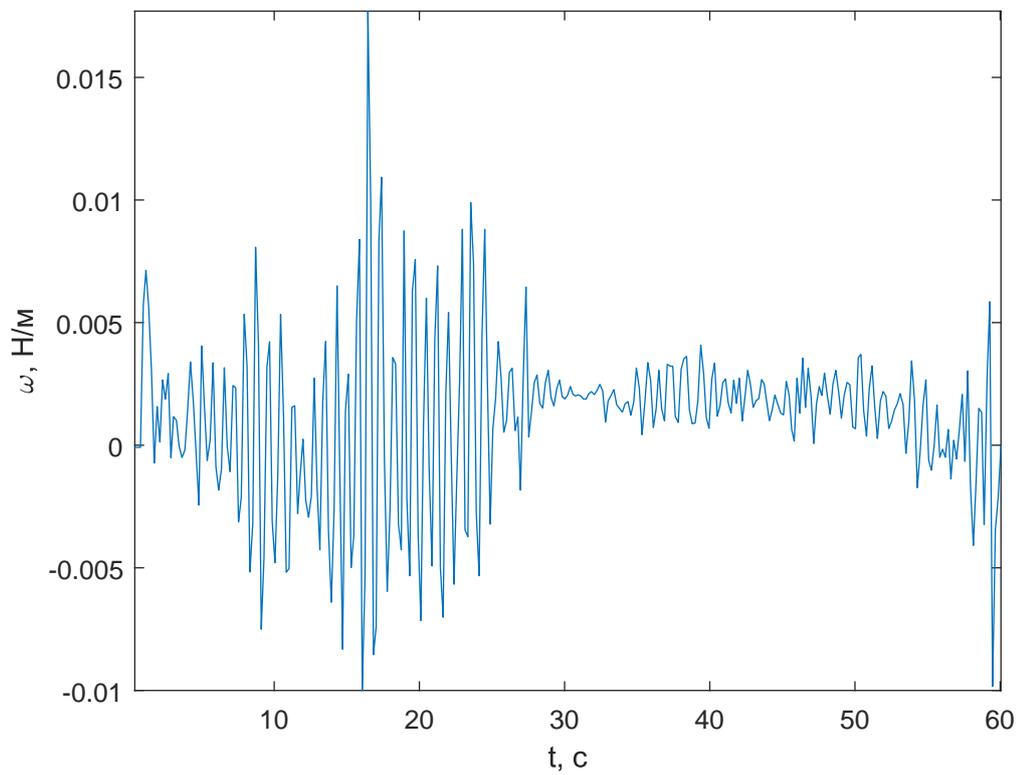


(a) Без корректора.

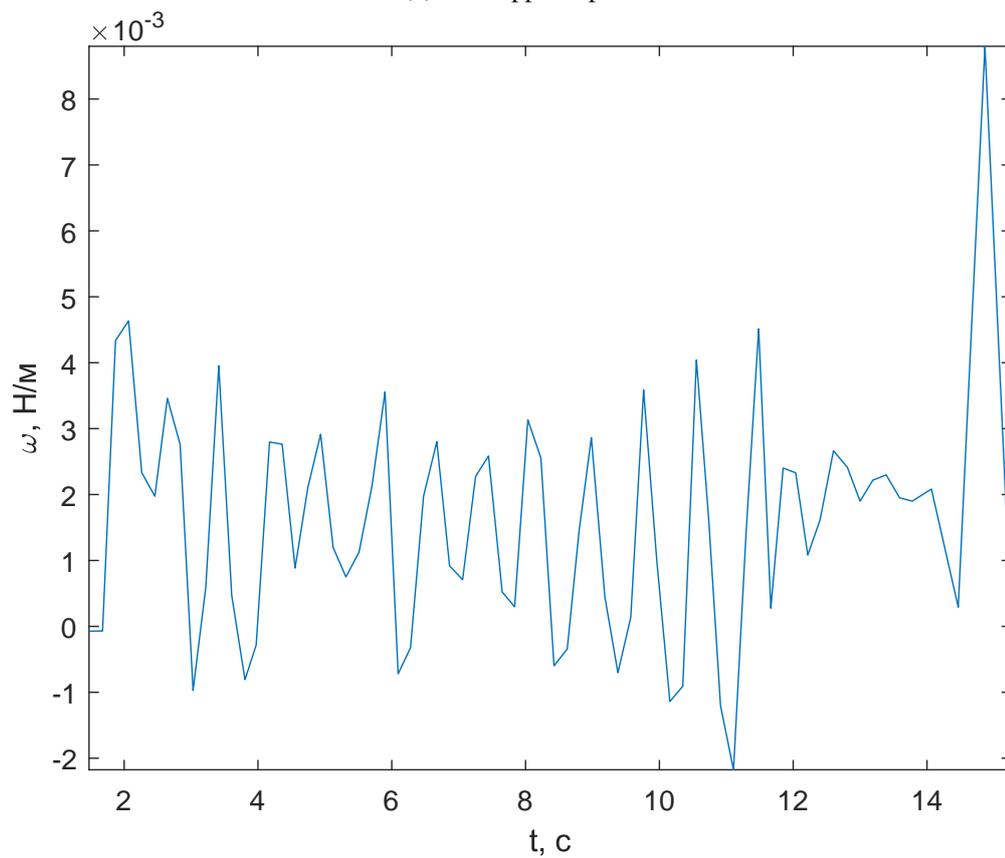


(b) С применением корректора.

**Рис. 13:** Графики линейных скоростей при использовании AKAZE. Качество процесса значительно хуже, чем при использовании алгоритма ORB.

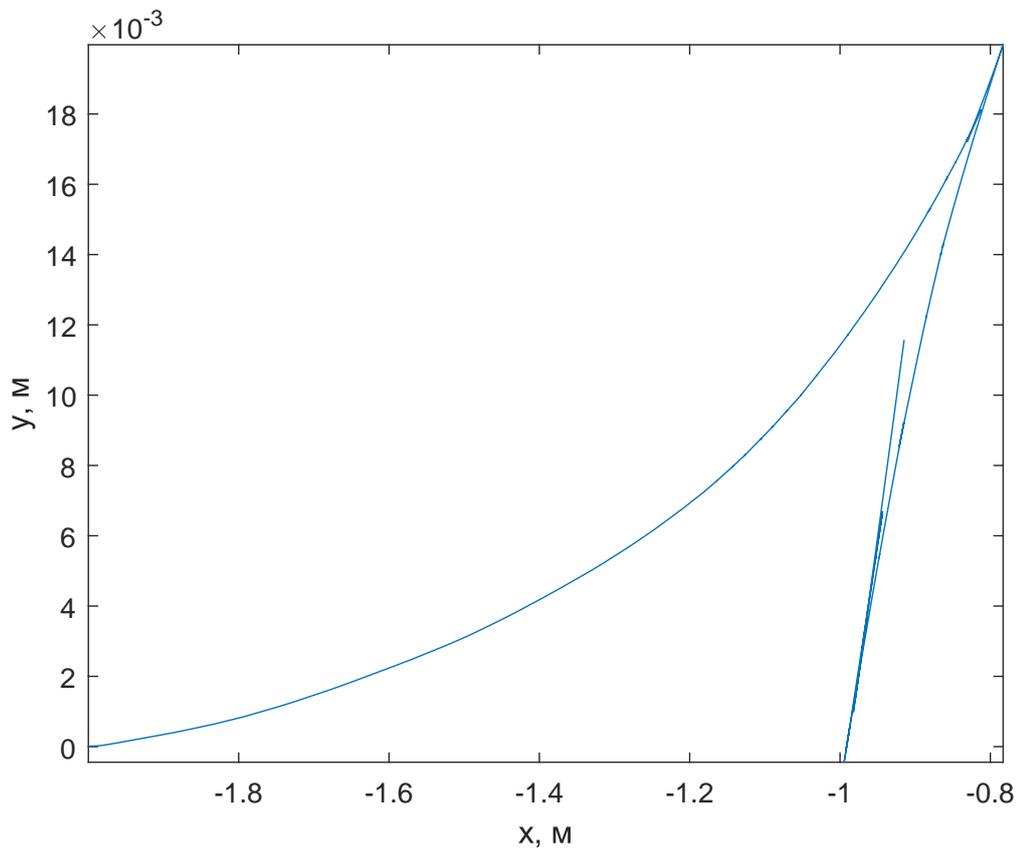


(a) Без корректора.

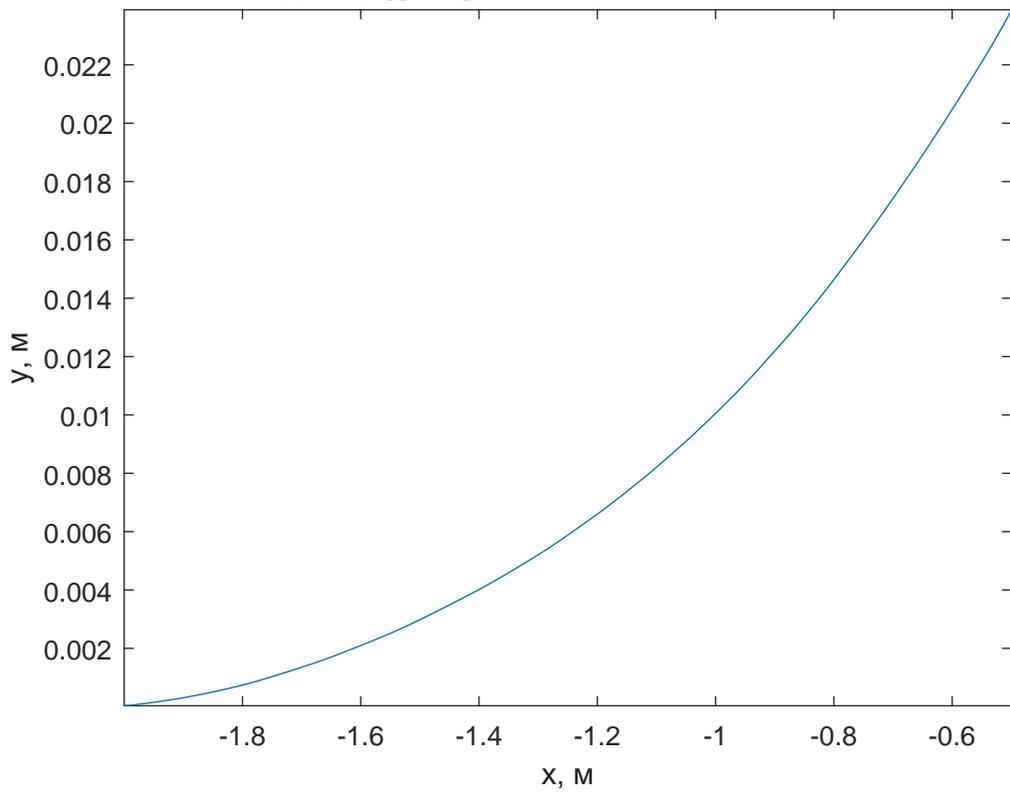


(b) С применением корректора.

**Рис. 14:** Графики угловых скоростей при использовании AKAZE.



(a) Без корректора. Задача не выполнена.



(b) С применением корректора. Точка остановки:  $(-0.499, 0.0238)$ .

**Рис. 15:** Траектории при использовании AKAZE. Несмотря на плохое качество процесса, позиционирование произведено точнее.

## Выводы

Предложенный в настоящей работе метод синтеза обратной связи позволяет получить закон управления, обеспечивающий позиционирование мобильного робота в положении с желаемым изображением с камеры. При этом система управления устойчива по отношению к внешнему возмущению. Разработанный моделирующий комплекс удовлетворяет поставленным требованиям и, при наличии достаточной вычислительной мощности, может использоваться для моделирования широкого круга реальных условий эксплуатации различных мобильных роботов с визуальной обратной связью. Важно отметить тот факт, что использование системы ROS позволяет практически без изменений использовать полученный для симуляционной модели код на реальном роботе.

Сравнение методов извлечения ключевых точек показало, что при использовании алгоритма ORB робот под управлением разработанного регулятора решает задачу быстрее, но менее точно, нежели с использованием AKAZE. Однако выигрыш в точности не значителен, поэтому можно сделать вывод о том, что алгоритм ORB является более подходящим для подобных систем управления.

Логичным является развитие настоящей работы в нескольких направлениях. Во-первых, возможно более детальное исследование алгоритмов подбора параметров регулятора с учетом математических основ методов извлечения ключевых точек. Также может быть рассмотрен оптимизационный подход к подбору параметров.

Во-вторых, актуальной задачей является построение алгоритма извлечения ключевых точек с учетом динамики робота. Потенциально это может позволить системе значительно снизить требования к вычислительной мощности управляющего процессора робота, так как на текущий момент большая часть вычислительной сложности приходится именно на алгоритмы компьютерного зрения.

В-третьих, можно повысить качество динамики и обеспечить явный учет ограничений за счет использования методов управления с прогнозирующими моделями.

Наконец, необходимо принять во внимание тот факт, что комплекс виртуальной реальности не моделирует внутренние процессы электродвигателей. Это приводит к появлению двух задач: построение модели электродвигателя с последующей ее интеграцией и построение алгоритма корректировки модели на основе экспериментальных данных, полученных от реального робота.

## Заключение

В ходе выполнения настоящей работы получены следующие результаты:

1. Разработан метод синтеза управления с многоцелевой структурой, обеспечивающий позиционирование мобильного робота в положении с желаемым изображением с камеры.
2. Разработан моделирующий комплекс виртуальной реальности, позволяющий производить моделирование движения мобильного робота под управлением с визуальной обратной связью с учетом динамики и внешних возмущений.
3. Протестирован моделирующий комплекс и проведена численная проверка синтезированного закона управления. В ходе тестирования проведено сравнение методов извлечения ключевых точек AKAZE и ORB.

## Список литературы

- [1] Chaumette F., Hutchinson S., Visual servo control. I. Basic approaches // IEEE Trans. Robot. Automat. 2006. Vol. 13, No 4. P. 82–90.
- [2] Agin G. J., Real time control of a robot with a mobile camera // Technical Note 179. SRI International. 1979.
- [3] Севостьянов Р. А. Программная поддержка процессов управления мобильным роботом с визуальной обратной связью. Системы управления и информационные технологии. 2019. Сер. 78, вып. 4. С. 83–86.
- [4] Сотникова М. В. Алгоритм автоматического удержания колесного робота на визуально заданной линии // Вестник СПбГУ. 2016. Сер. 10, вып. 1. С. 99–108.
- [5] Pitonakova L., Giuliani M., Pipe A., Winfield A., Feature and performance comparison of the V-REP, Gazebo and ARGoS robot simulators // Proc. of the 19th Towards Autonomous Robotic Systems Conf. (TAROS 2018) / Lecture Notes in Computer Science. Springer, 2018. Vol. 10965, P. 357–368.
- [6] Tareen S. A. K., Saleem Z., A comparative analysis of sift surf kaze akaze orb and brisk // 2018 Int. Conf. on Computing Mathematics and Engineering Technologies (iCoMET). 2018. P. 1–10.
- [7] Carona R., Aguiar A. P., Gaspar J., Control of unicycle type robots: tracking, path following and point stabilization // Proc. of IV Jornadas de Engenharia de Electronica e Telecomunicacoes e de Computadores. 2008. P. 180–185.
- [8] Veremei E. I., Korchanov V. M. Multiobjective stabilization of a certain class of dynamic systems // Automation and Remote Control. 1989. Vol. 49, No 9. P. 1210–1219.
- [9] Veremey E. I., Separate filtering correction of observer-based marine positioning control laws // International Journal of Control, 2017. Vol. 90, No. 8. P. 1561–1575.

- [10] Veremey E. I., Dynamical correction of positioning control laws // Proc. of the 9th IFAC Conf. on Control Applications in Marine Systems. 2013. P. 31–36.