

Санкт–Петербургский государственный университет

*Абушик Вероника Павловна*

**Выпускная квалификационная работа**

*Многоцелевое управление омниколесным роботом*

Уровень образования: бакалавриат

Направление 01.03.02 «Прикладная математика и информатика»

Основная образовательная программа СВ.5005.2017 «Прикладная математика, фундаментальная информатика и программирование»

Профиль «Исследование и проектирование систем управления и обработки сигналов»

Научный руководитель:

кандидат ф.-м. наук,

доцент кафедры КТС

Коровкин М.В.

Рецензент: Семакова А.А.

Санкт-Петербург

2021 г.

# Содержание

<b>Введение</b> . . . . .	3
<b>Обзор литературы</b> . . . . .	5
<b>Постановка задачи</b> . . . . .	6
<b>Глава 1. Управление движением с учётом внешних возмущений</b> . . . . .	8
1.1. Синтез регулятора . . . . .	8
1.2. Результаты экспериментов . . . . .	9
1.3. Вывод . . . . .	12
<b>Глава 2. Исследование движения с помощью компьютерного зрения</b> . . . . .	13
2.1. Синтез регулятора . . . . .	13
2.2. Результаты экспериментов . . . . .	13
2.3. Вывод . . . . .	20
<b>Глава 3. Случай кругового движения</b> . . . . .	21
3.1. Синтез регулятора . . . . .	21
3.2. Результаты экспериментов . . . . .	22
3.2.1 Случай без помощи технического зрения . . . . .	23
3.2.2 Случай с использованием технического зрения . . . . .	24
3.3. Вывод . . . . .	28
<b>Выводы</b> . . . . .	29
<b>Заключение</b> . . . . .	30
<b>Список литературы</b> . . . . .	31
<b>Приложения</b> . . . . .	31

## Введение

Мобильные роботы играют все большую роль в современной промышленности. В частности, особое место занимают роботы на омниколесах, способные двигаться в любом направлении. Например, такие роботы часто используются для решения логистических проблем на складах. Как правило, основной задачей мобильного робота является движение по заданной траектории. Естественно, для промышленных целей удобно автоматизировать функционирование роботов, для достижения этой цели нередко применяются технологии компьютерного зрения.

В настоящее время графические маркеры различных видов широко используются в самых разных областях. Штрих-коды в магазинах, qr-коды для упрощения процесса передачи важных данных на смартфон. Отдельно стоит выделить еще один тип маркеров — ArUco [1], которые особенно полезны в задачах дополненной реальности. Другая важная особенность - возможность упростить задачи робототехники. Aruco-маркеры легко распознать на изображении, также они имеют хорошую устойчивость к шуму. Если известен реальный размер маркера, а также внутренние параметры камеры, то относительно легко можно получить оценку положения маркера в системе координат камеры.

Стоит отметить, что общая задача оценки положения робота очень сложная. Инерционные датчики дают довольно зашумленный результат и имеют ошибки интеграции. Бортовые лидары и камеры очень сложны с вычислительной точки зрения, к тому же требуется либо предварительное знание карты, либо необходимость сначала построить ее. GPS-подобные датчики имеют очень низкую точность для небольших лабораторных роботов. Для задач такого рода внешняя камера — самое простое решение, поэтому логичным представляется использование aruco-маркеров.

По указанным темам уже проведено достаточно много исследований и разработано множество алгоритмов. В данной работе представляется собственный подход, который может быть полезен в случаях, когда есть какие-то требования к динамике движения робота. Например, есть задача обеспечить специфическую реакцию на разного рода внешние возмущения. Для таких

задач целесообразно использовать так называемый многоцелевой регулятор [2], применение которого описывается в данной работе. Настраиваемые параметры этого регулятора могут быть вычислены относительно независимо, что позволяет разбить изначально тяжелую задачу на несколько индивидуальных более простых задач. Также учитывается динамика робота, повышающая эффективность управления. Результаты проделанной работы продемонстрированы с помощью компьютерной модели движения робота. Программное обеспечение разработано с использованием языка программирования Python.

## Обзор литературы

Проблема устойчивого передвижения робота активно обсуждается на конференциях, посвящённых робототехнике и теории управления, но в большинстве своем статьи содержат лишь постановку задачи, общие слова о способе её решения и приведение полученных результатов [3].

Многие из доступных материалов – статьи с международных конференций или из специализированных журналов, поэтому они написаны на английском языке. Существуют и русскоязычные материалы [2]. Однако часто они описывают лишь теоретическую часть без примеров испытаний на реальных или близким к ним моделям [6].

Если же примеры и есть [5], в основном результаты показаны только в виде демонстрации, код программы нельзя интегрировать прямым образом в программу по управлению. Существует множество популярных, часто используемых из-за своего удобства в разработке, языков программирования [7], а также инструментов [8, 9, 10] для решения подобных задач, но про использование их в данной области сказано очень мало.

Управление омниколесным роботом — весьма специфическая задача. Эти роботы используются для самых различных целей, поэтому построение управления для них является весьма актуальной темой.

## Постановка задачи

В данной работе объект управления — мобильный робот. Он представляет собой платформу, установленную на трех омниколесах, что обеспечивает полноприводность системы. Вращение колёс обеспечивается электромоторами. В соответствии с [3], схема робота изображена на рис.1.

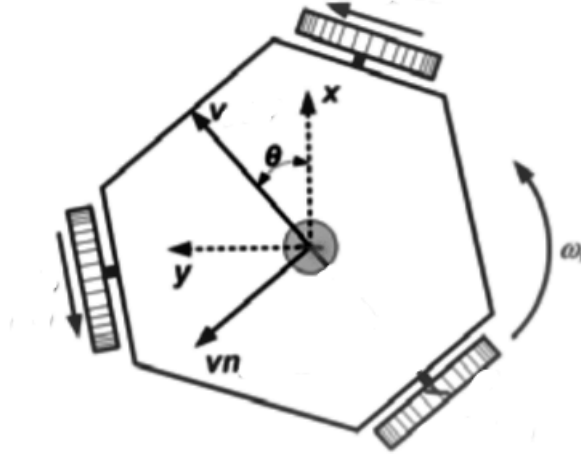


Рис. 1: Схема омниколесного робота.

Математическая модель робота может быть описана системой:

$$\begin{aligned} \dot{v} &= Av + B\tau + h, \\ \dot{\eta} &= R(\eta)v. \end{aligned} \quad (1)$$

Здесь  $v = (v, v_n, \omega)^T$  — вектор скоростей, где  $v$  — линейная скорость,  $v_n$  — нормальная линейная скорость,  $\omega$  — угловая скорость;  $\tau = (\tau_1, \tau_2, \tau_3)^T$  — вектор напряжений на моторах;  $\eta = (x, y, \varphi)^T$  — положение центра масс и курсовой угол робота;  $A$  — диагональная матрица коэффициентов трения;  $B$  — матрица коэффициентов при управляющих воздействиях. Единственная нелинейность системы определяется матрицей

$$R(\eta) = R(\varphi) = \begin{pmatrix} \cos(\varphi) & -\sin(\varphi) & 0 \\ \sin(\varphi) & \cos(\varphi) & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

В первой главе будет рассматриваться задача построения управления движением омниколесного робота при движении по заданной прямой, в том числе случай с учётом внешних возмущений. Во второй главе представлено решение этой же задачи с использованием технологий компьютерного зрения для определения положения робота. Для этого на верхней части корпуса робота лицевой стороной вверх установлен агисо-маркер. Над сценой установлена стационарная видеокамера, направленная вертикально вниз. Информация о положении и ориентации робота поступает из изображения после обработки с помощью алгоритма компьютерного зрения. Цель - попасть на желаемый путь и двигаться вдоль него с желаемой постоянной скоростью  $v_d$  даже при наличии постоянных внешних возмущений (например, боковой ветер или наклон плоскости). В третьей главе описан пример движения по криволинейной траектории, в качестве которой принята окружность.

# Глава 1. Управление движением с учётом внешних возмущений

## 1.1 Синтез регулятора

Суть подхода заключается в минимизации невязки между фактическим положением робота и ближайшей точкой заданной линии, при этом курсовой угол робота должен стремиться к направлению касательной к линии. Для простоты в качестве заданной линии рассматривается прямая.

Пусть  $(\hat{x}, \hat{y}, \theta)^T$  — вектор текущих координат и курсового угла робота;  $v_y$  — желаемая скорость движения по линии. Тогда уравнения расхождений имеют вид:

$$e(t) = \begin{pmatrix} \hat{x} \\ (\hat{y} - v_y t) \\ \theta \end{pmatrix} = \eta - \begin{pmatrix} 0 \\ v_y t \\ 0 \end{pmatrix}.$$

Динамика расхождений описывается при помощи уравнения:

$$\dot{e}(t) = \dot{\eta} - \begin{pmatrix} 0 \\ v_y \\ 0 \end{pmatrix} = R(\eta)v - \begin{pmatrix} 0 \\ v_y \\ 0 \end{pmatrix}.$$

Экспоненциальной сходимости невязок можно добиться, выбрав вектор скоростей:

$$v = v_d = -\mu R^T(\eta)e(t).$$

Для обеспечения вектора  $v$  вводится в рассмотрение многоцелевая структура управления, которая в пространстве состояний описывается уравнениями [5]

$$\begin{aligned} \dot{z}_v &= Az_v + B\tau + R^T(\eta)K_1(\eta - z_\eta), \\ \dot{z}_\eta &= R(\eta)z_v + K_2(\eta - z_\eta), \\ \dot{p} &= A_p p + B_p(\eta - z_\eta), \\ \xi &= C_p p + D_p(\eta - z_\eta), \\ \tau &= -K_d(z_v - K_v v_d) + \xi, \end{aligned}$$

где  $K_1$  и  $K_2$  — положительно определенные диагональные матрицы, которые



задают динамику сходимости оценок вектора состояния, а также могут влиять на динамику реакции системы на внешние возмущения. Уравнение  $\tau$  задаёт закон управления, где  $K_d$  — настраиваемая матрица, обеспечивающая асимптотическую устойчивость,  $K_v$  — вспомогательная матрица, обеспечивающая равенство фактической и желаемой скоростей в положении равновесия [6]. Элемент  $\xi$  в уравнении  $\tau$  является динамическим корректором, обеспечивающим астатизм системы. Передаточная матрица корректора [3] представляется в виде:

$$F(s) = C_p(Es - A_p)^{-1}B_p + D_p,$$

причем матрица  $A_p$  — гурвицева. Для обеспечения астатизма в рамках данной задачи передаточная матрица корректора должна удовлетворять условию:

$$F(0) = B^{-1}(A - BK_d)R^T(\varphi_0)K_2 - R^T(\varphi_0)K_1.$$

## 1.2 Результаты экспериментов

В соответствии с [5], принимаются следующие значения для матриц системы (1):

$$A = \begin{pmatrix} 0.48 & 0 & 0 \\ 0 & 0.53 & 0 \\ 0 & 0 & 0.023 \end{pmatrix}, \quad B = \begin{pmatrix} -0.1745 & 0 & 0,1745 \\ 0.1 & 0.2 & 0.1 \\ 1.7938 & 1.7938 & 1.7938 \end{pmatrix}.$$

Настраиваемые параметры выбраны следующим образом:

$$\mu = 8, \quad K_1 = K_2 = E_{3 \times 3},$$

$$K_d = \begin{pmatrix} -3.81 & -0.0012 & 0.5733 \\ 0 & 0,0083 & 0.5727 \\ 3.8106 & -0.0012 & 0.5733 \end{pmatrix},$$

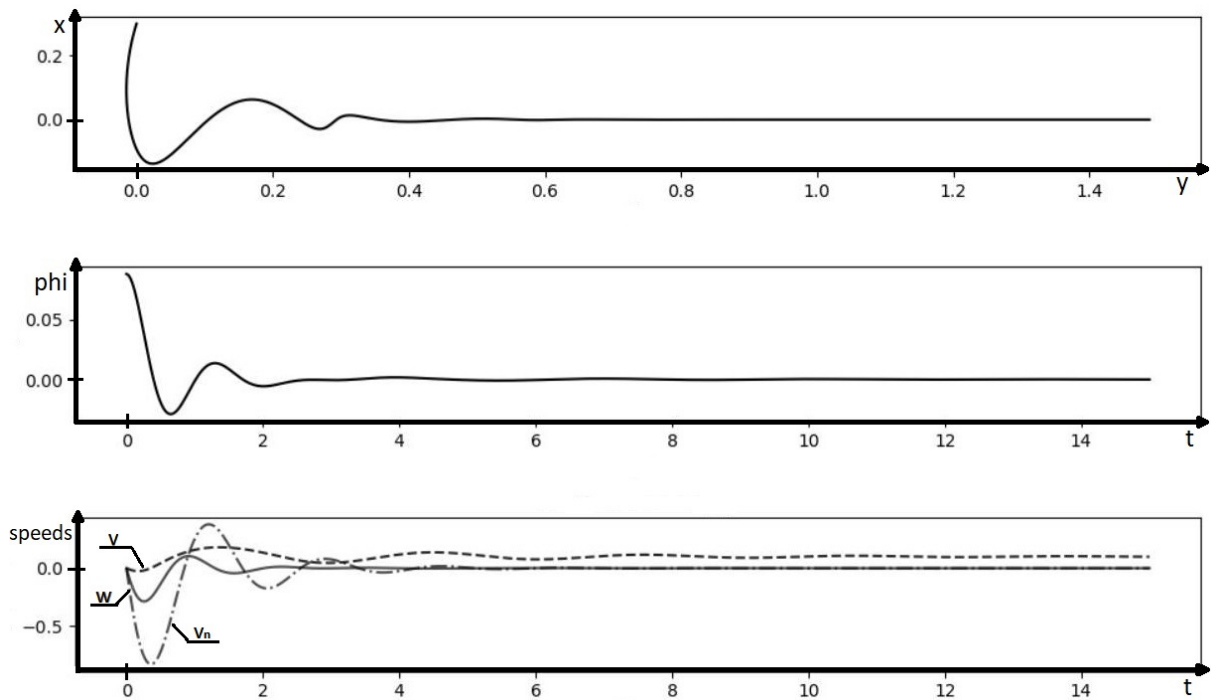
$$K_v = \begin{pmatrix} 1.3608 & 0 & 0 \\ 0 & 831.2 & -2.6982 \\ 0 & -2.7928 & 1.0165 \end{pmatrix}.$$

Рассматривается ситуация, при которой желаемая скорость принята  $v_y = 0.1$  м/с, в качестве целевой прямой выбрана ось  $Oy$ , в начальный момент робот имеет координаты

$$\eta_0 = \left( 0.3 \quad 0 \quad \frac{\pi}{36} \right)^T.$$

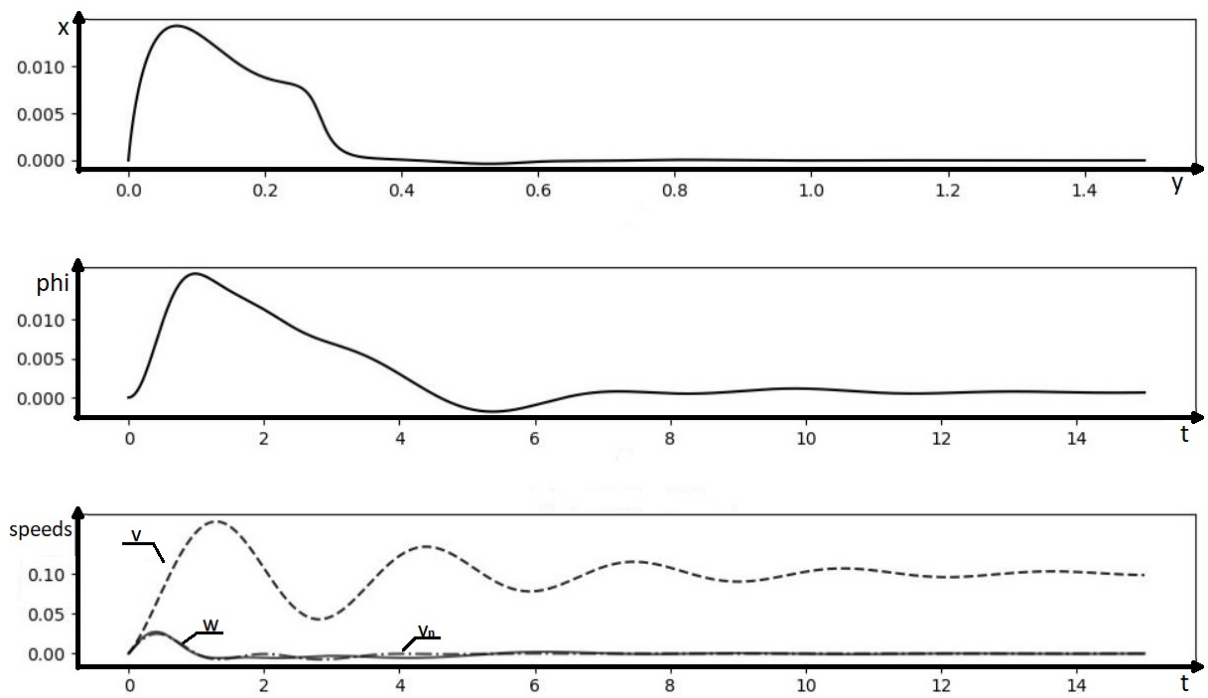
Для проведения численного моделирования была реализована программа на языке Python [7] с использованием таких библиотек, как `numpy`, `scipy` [9], в частности, функция `odeint` [10] для численного решения систем дифференциальных уравнений. Результаты экспериментов на рис. 2, 3 показывают траекторию движения робота  $y(x)$ , изменение курсового угла  $\varphi(t)$  и изменение скоростей  $v$ ,  $v_n$  и  $\omega$  с изменением момента времени  $t$ .

На рис. 2 представлена динамика для случая движения без постоянного возмущения, заметим, что робот действительно выходит на заданную прямую и движется с заданной скоростью.



**Рис. 2:** Динамика движения робота без постоянного возмущения.

На рис. 3 представлена динамика при наличии постоянного возмущения  $h = \left( 0.1 \quad 0.1 \quad 0.1 \right)^T$ . В этом случае робот изначально находится на заданной прямой с правильной ориентацией.



**Рис. 3:** Динамика движения робота с учетом постоянного возмущения.

Можно видеть, что при воздействии возмущения, робот отклоняется от заданной линии, но затем возвращается обратно, а это значит, что система действительно обладает астатизмом.

### **1.3 Вывод**

В данной главе представлено описание алгоритма решения задачи управления движением омниколесного робота с постоянной скоростью вдоль заданной линии с учетом внешних возмущений на основе обратной связи в виде многоцелевого регулятора. Результаты экспериментов демонстрируют работоспособность и эффективность принятого подхода.

## Глава 2. Исследование движения с помощью компьютерного зрения

### 2.1 Синтез регулятора

Снова стоит задача о минимизации невязки между фактическим положением робота и ближайшей точкой целевой линии движения, курсовой угол робота должен стремиться к направлению касательной к этой линии. Для простоты в качестве заданной линии рассматривается прямая  $y = 0$ , а  $v_d$  – желаемая постоянная скорость. Тогда вектор расхождений:

$$e(t) = \begin{pmatrix} x(t) \\ (y(t) - v_d t) \\ \phi(t) \end{pmatrix}.$$

Динамика расхождений описывается при помощи уравнения:

$$\dot{e}(t) = \dot{\eta} - \begin{pmatrix} 0 \\ v_y \\ 0 \end{pmatrix} = R(\eta)v - v_d.$$

Экспоненциальной сходимости невязок можно добиться, выбрав вектор скоростей в виде:

$$v = v^* = -\mu R^T(\eta)e(t), \mu > 0.$$

Для обеспечения этого вектора вводится в рассмотрение многоцелевой регулятор.

### 2.2 Результаты экспериментов

Чтобы проверить данный метод решения подобных задач, было разработано программное обеспечение для моделирования на языке Python [7]. В частности, использовались библиотеки `numpy` и `scipy` [9] для вычислений, `matplotlib` [8] для рисования графиков, а также `OpenCV` [4] для генерации, рисования и обнаружения агисо-маркеров. Ядром системы являются два блока

систем ОДУ — один для динамической модели робота, а другой для интегрирования в систему регулятор, у которого есть собственная динамика. Сначала интегрируется дельта-время шаг, получая текущее положение робота. Затем используются полученные данные, чтобы нарисовать агисо-маркер на виртуальной схеме движения в пространстве, как если бы была настоящая камера. Обнаружив маркер, используются его углы, чтобы оценить положение маркера относительно объектива камеры, используя библиотека OpenCV. При настройке матрицы камеры, предполагается, что все коэффициенты искажения равны нулю, тогда матрица имеет вид:

$$\begin{pmatrix} 1 & 0 & (W - 1)/2 \\ 0 & 1 & (H - 1)/2 \\ 0 & 0 & 1 \end{pmatrix},$$

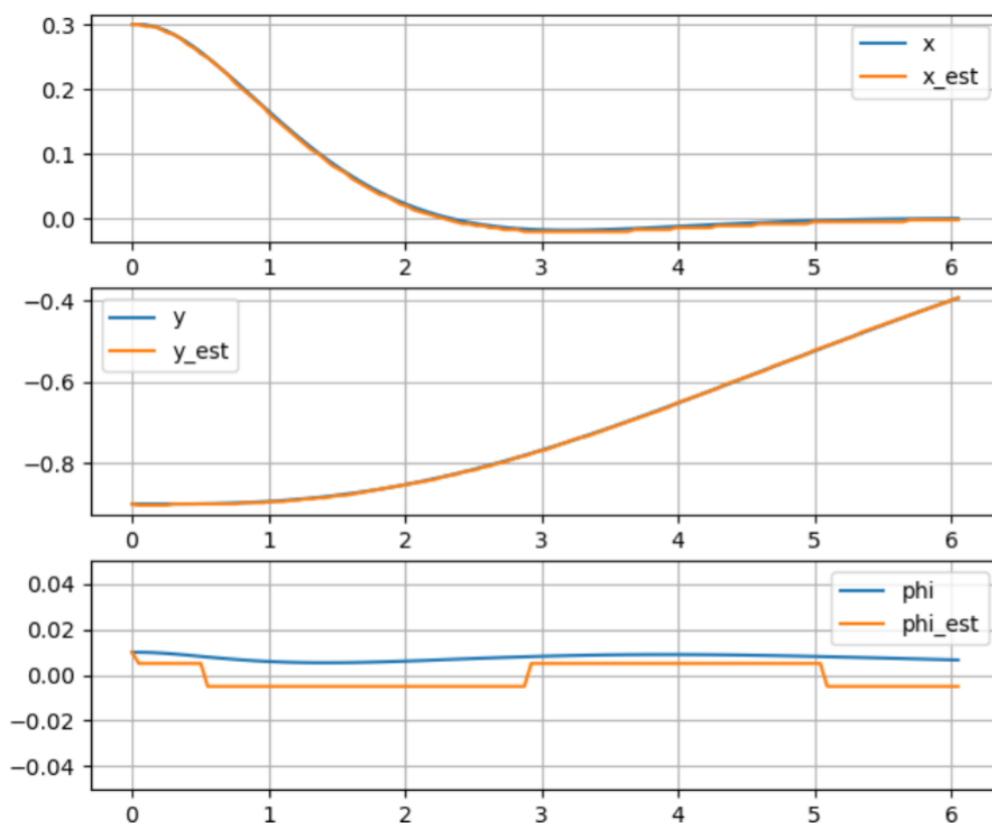
где W и H — ширина и длина получаемого изображения в пикселях (в данном случае используется разрешение 1000x1000 пикселей). Расчетное положение робота переходит в часть контроллера, интегрируется динамика регулятора, затем вычисляется вектор ошибки и текущая скорость, необходимая для достижения желаемого пути, после чего на выходе получается управляющий сигнал для следующего шага интегрирования.

В соответствии с [5], матрицы модели робота имеют следующие значения:

$$A = \begin{pmatrix} 0.48 & 0 & 0 \\ 0 & 0.53 & 0 \\ 0 & 0 & 0.023 \end{pmatrix}, \quad B = \begin{pmatrix} -0.1745 & 0 & 0,1745 \\ 0.1 & 0.2 & 0.1 \\ 1.7938 & 1.7938 & 1.7938 \end{pmatrix}.$$

Прежде всего, стоит проверить точность оценки состояния агисо - детектором. Для этого проводится простой эксперимент, получая данные исходя из «идеальной» модели и из оценочных позиций робота при помощи аруко-маркеров. На рис.4 показано сравнение данных для этого случая. Видны резкие скачки при оценке курсового угла — они вызваны дискретностью компьютерных изображений. Однако также заметно, что точность довольно велика, ошибка составляет всего около 0,01 радиана, что дает весьма хороший результат. Конечно, в реальном мире ошибки будут несколько больше,

но необходимо, по крайней мере, протестировать данный подход в случае виртуальной модели.



**Рис. 4:** Точность агисо-детектора.

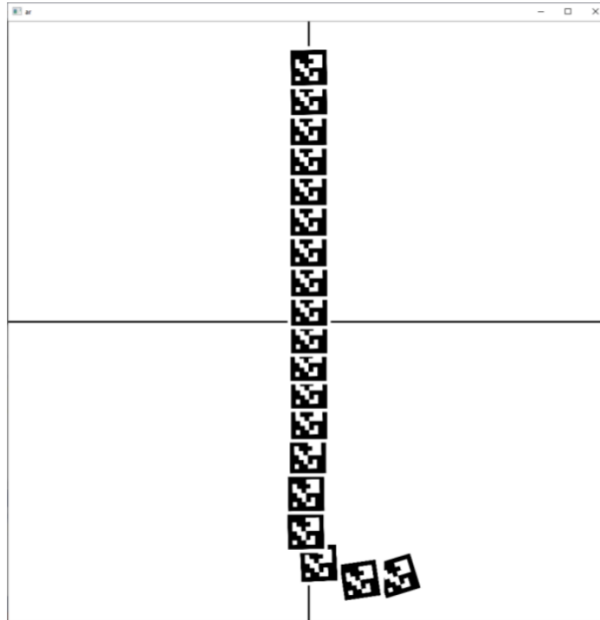
Теперь проверяется эффективность в случае отсутствия внешних возмущений, а динамический корректор отключен. Настраиваемые параметры контроллера имеют следующие значения:  $K_d$  была найдена в результате решения задачи LQR с матрицей  $Q = \begin{pmatrix} 50 & 0.1 & 1 \end{pmatrix}$ ,  $R = 1$ .

$$\mu = 1.0, K_1 = K_2 = E_{3 \times 3},$$

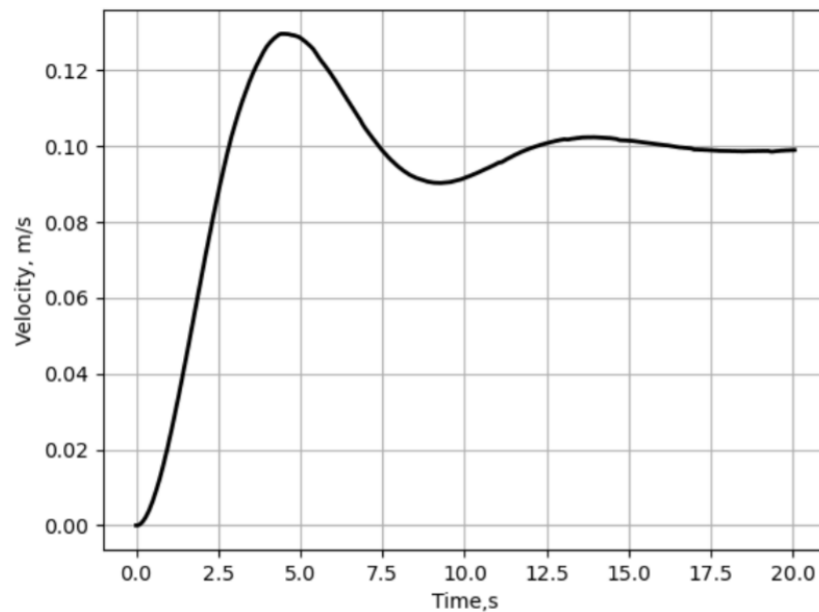
$$K_d = \begin{pmatrix} -3.81 & -0.0012 & 0.5733 \\ 0 & 0,0083 & 0.5727 \\ 3.8106 & -0.0012 & 0.5733 \end{pmatrix},$$

$$K_v = \begin{pmatrix} 1.3608 & 0 & 0 \\ 0 & 831.2 & -2.6982 \\ 0 & -2.7928 & 1.0165 \end{pmatrix}.$$

Исходное положение робота  $x_0 = 0.3$  м,  $y_0 = -0.9$  м,  $\phi_0 = 0.3$  рад., желаемая скорость  $v_0 = 0.1$  м/с. На рис.5 и 6 показаны траектория и скорость робота — видно, что робот успешно доходит до линии, следуя по ней с желаемой скоростью.



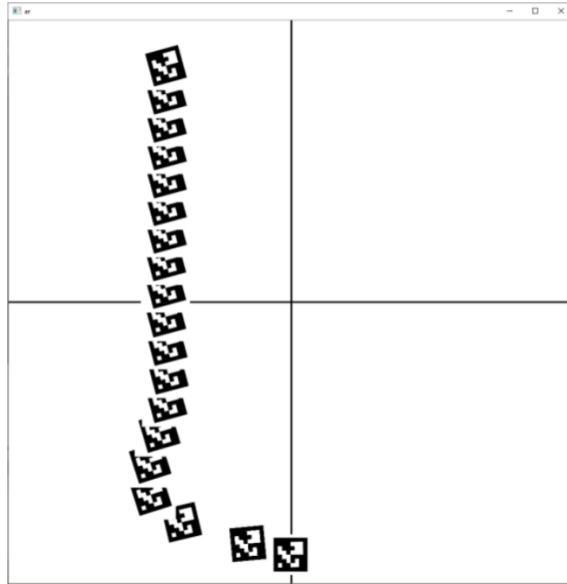
**Рис. 5:** Траектория робота без учета внешних возмущений.



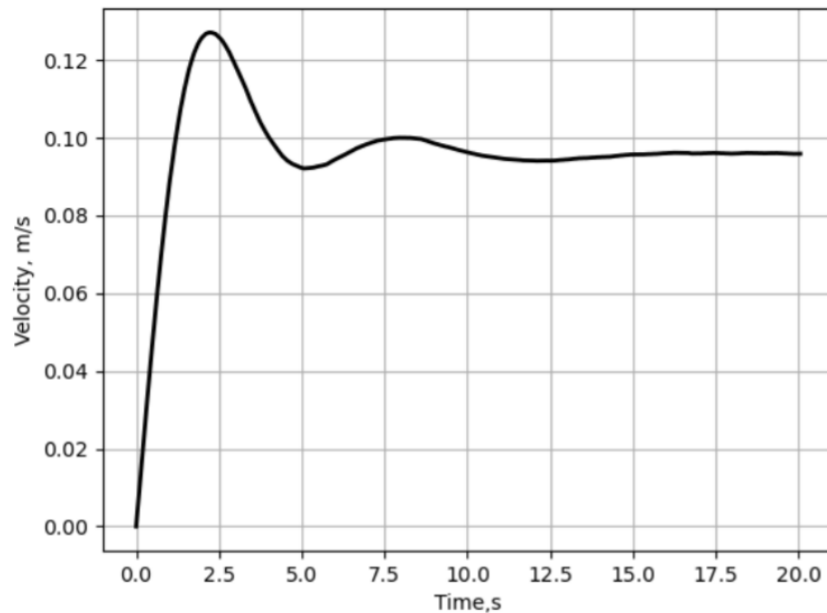
**Рис. 6:** Скорость робота без учета внешних возмущений.



Теперь вводится вектор возмущения  $h = \begin{pmatrix} -0.3 & 0.1 & 0.3 \end{pmatrix}$ , однако оставим динамический корректор выключенным. Здесь рассматриваются другие начальные данные — робот уже стоит на прямой:  $x_0 = 0.0$  м,  $y_0 = -0.9$  м,  $\phi_0 = 0.0$  рад. Рисунки 7 и 8 демонстрируют, что робот отходит от прямой, курсовой угол не нулевой, а также не двигается с постоянной желаемой скоростью.

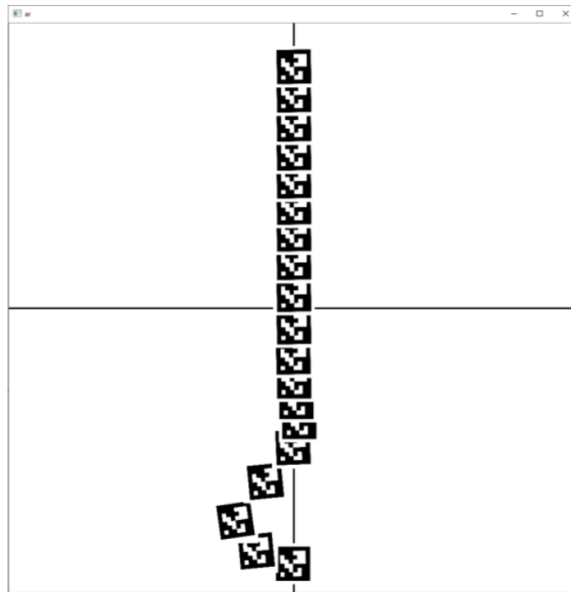


**Рис. 7:** Траектория робота с учетом возмущений, без использования корректора.

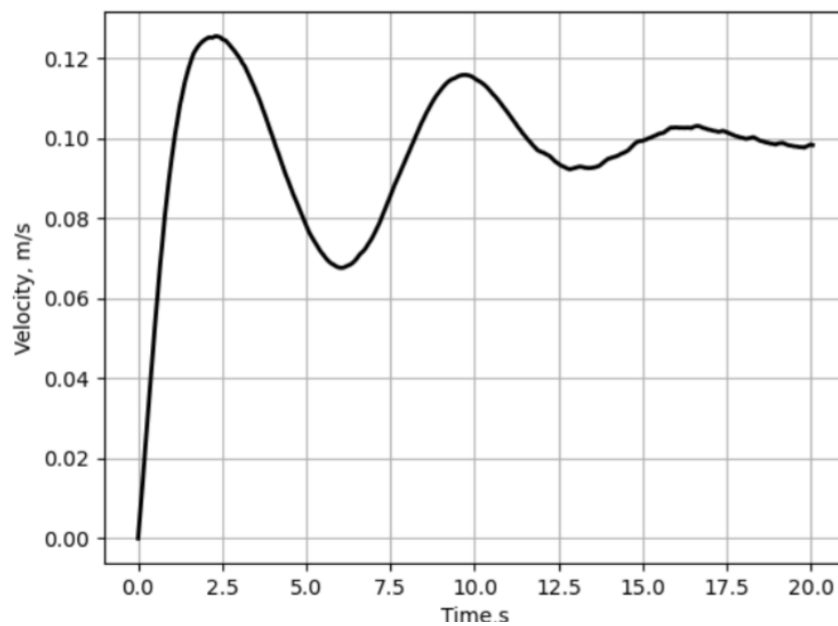


**Рис. 8:** Изменение скорости робота с учетом внешних возмущений, без использования корректора.

Далее включается корректор. Как видно из рисунков 9 и 10, в этом случае, после небольшого перерегулирования, робот возвращается на линию, достигает правильного курсового угла и желаемой скорости.

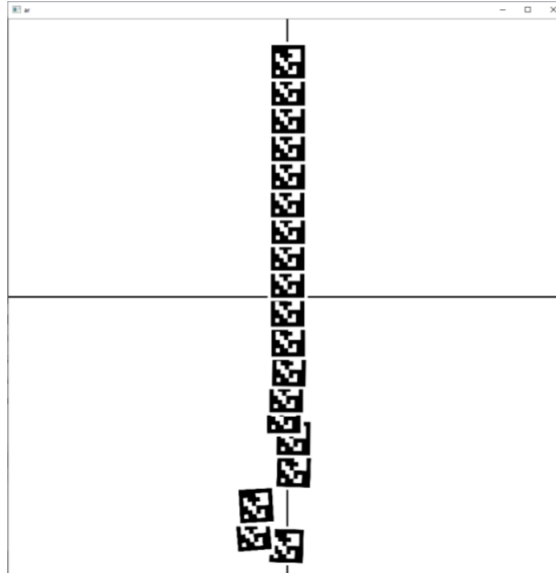


**Рис. 9:** Траектория робота с учетом возмущений, с использованием корректора.

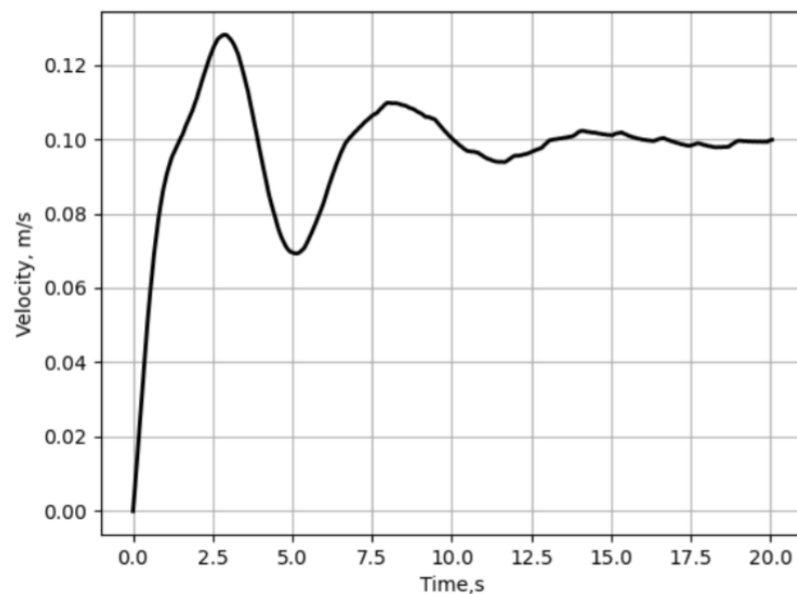


**Рис. 10:** Изменение скорости робота с учетом внешних возмущений, с использованием корректора.

Стоит отметить, что контролировать выбросы возможно за счет настройки матрицы наблюдения. Например, увеличим значения в матрице  $K_1 = 4E_{3 \times 3}$ . Рисунки 11 и 12 демонстрируют значительное уменьшение выхода за пределы траектории.



**Рис. 11:** Траектория робота с учетом внешних возмущений и исправленными матрицами наблюдения.



**Рис. 12:** Изменение скорости робота с учетом внешних возмущений и исправленными матрицами наблюдения.

## **2.3 Вывод**

В данной главе представлено описание алгоритма решения задачи управления движением омниколесного робота с постоянной скоростью вдоль заданной линии с учетом внешних возмущений на основе обратной связи в виде многоцелевого регулятора и технологий компьютерного зрения. Результаты экспериментов демонстрируют работоспособность и эффективность принятого подхода.

## Глава 3. Случай кругового движения

В случае, если необходимо изменить траекторию робота с прямой на круг, регулятор применим в той же мере. Для достижения цели достаточно изменить уравнения целевой функции. Рассматриваются случаи без использования технологий компьютерного зрения.

### 3.1 Синтез регулятора

Чтобы минимизировать невязки между фактическим положением робота и ближайшей точкой целевой линии движения, курсовой угол робота должен стремиться к направлению касательной к этой линии. Для простоты рассматривается в качестве заданной функции окружность радиуса  $r = 0.1\text{м}$  и центром в начале координат, а желаемая скорость задается с помощью выбора отсчетов времени  $t$ . Тогда вектор расхождений:

$$e(t) = \begin{pmatrix} x(t) - r \cos(t) \\ y(t) - r \sin(t) \\ \phi(t) \end{pmatrix}.$$

Динамика расхождений:

$$\dot{e}(t) = \dot{\eta} - \begin{pmatrix} x(t) - r \sin(-t) \\ y(t) - r \cos(t) \\ \phi(t) \end{pmatrix}.$$

Закон управления в случае кругового движения будет выглядеть следующим образом:  $\tau = -K_v v - R^T K_p (\eta - z_\eta)$ ,

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

## 3.2 Результаты экспериментов

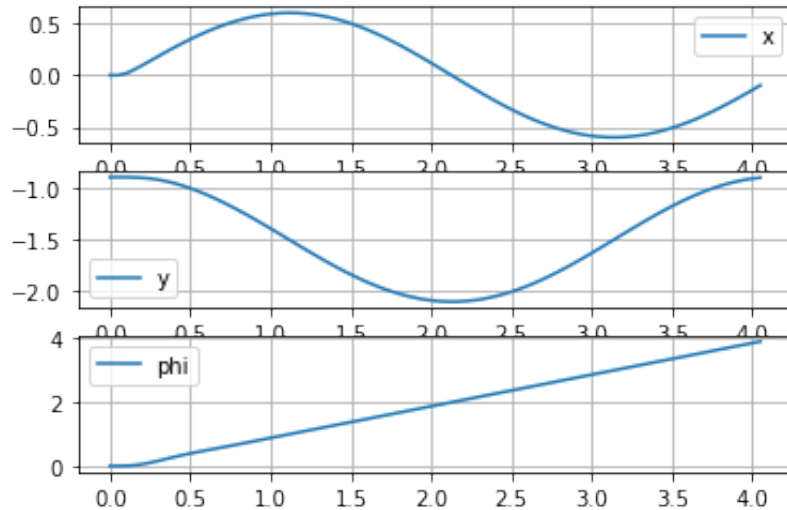
Для проверки решений данных задач используется то же программное обеспечение для моделирования, что и в предыдущей главе. В данном случае используется разрешение 10000x10000 пикселей. В соответствии с [5], матрицы модели робота имеют следующие значения:

$$A = \begin{pmatrix} 0.48 & 0 & 0 \\ 0 & 0.53 & 0 \\ 0 & 0 & 0.023 \end{pmatrix}, B = \begin{pmatrix} -0.1745 & 0 & 0,1745 \\ 0.1 & 0.2 & 0.1 \\ 1.7938 & 1.7938 & 1.7938 \end{pmatrix}.$$

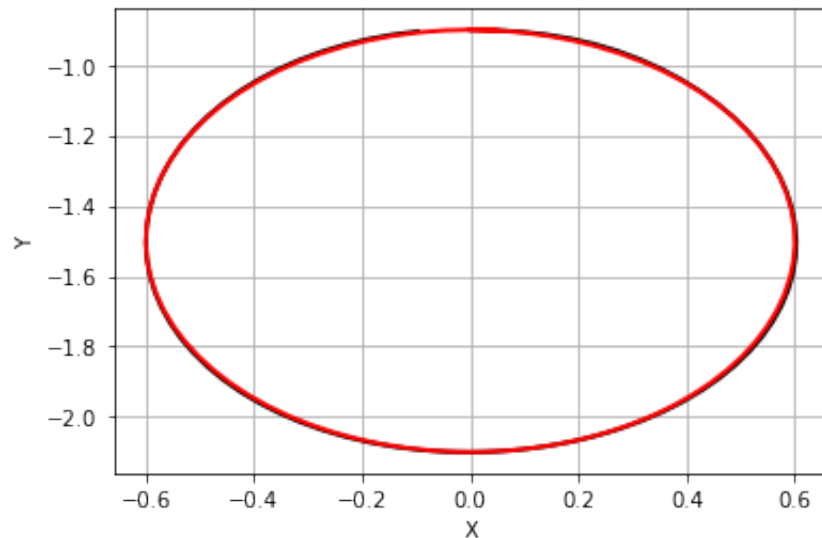
Настраиваемые параметры контроллера имеют следующие значения:  $K_d$  была найдена в результате решения задачи LQR с матрицей  $Q = \begin{pmatrix} 40 & 1 & 1 \end{pmatrix}$ ,  $R = 1$ . Другие параметры принимаются за следующие значения:  $\mu = 10$ ,  $K_1 = K_2 = E_{3 \times 3}$ , а постоянное возмущение  $h = \begin{pmatrix} 0.1 & 0.1 & 0.1 \end{pmatrix}$ .

### 3.2.1 Случай без помощи технического зрения

На рис.13, рис.14 представлены динамика и траектория для случая движения без постоянного возмущения. Робот действительно двигается по кругу.

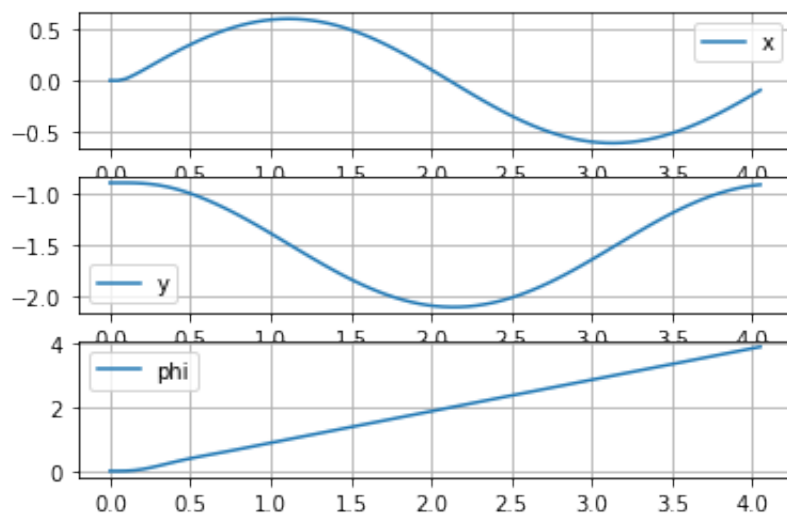


**Рис. 13:** Динамика движения робота без постоянного возмущения.

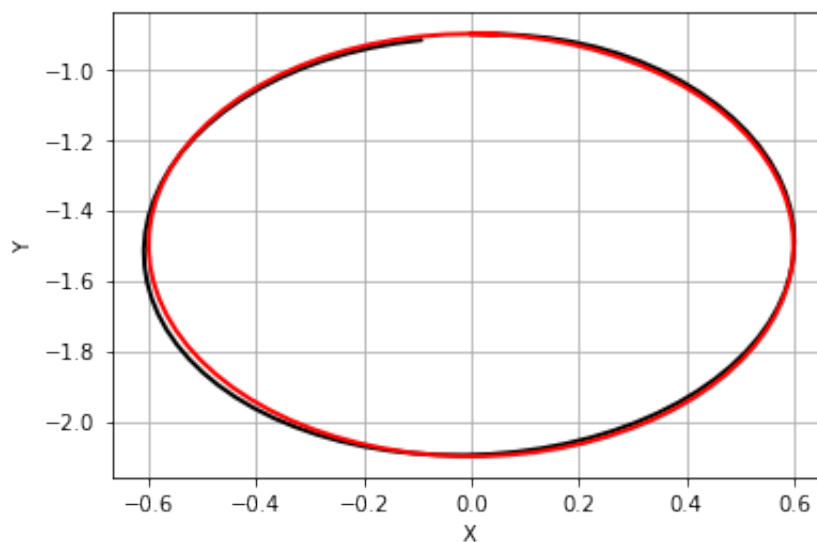


**Рис. 14:** Траектория движения робота без постоянного возмущения.

Теперь рассматривается случай с учётом постоянных возмущений. Красная кривая — желаемая траектория движения, в то время как чёрная — фактическая. На рис.15 и рис.16 видно, что результат применения рассмотренного подхода доказывает его эффективность.



**Рис. 15:** Динамика движения робота с учетом постоянных возмущений.



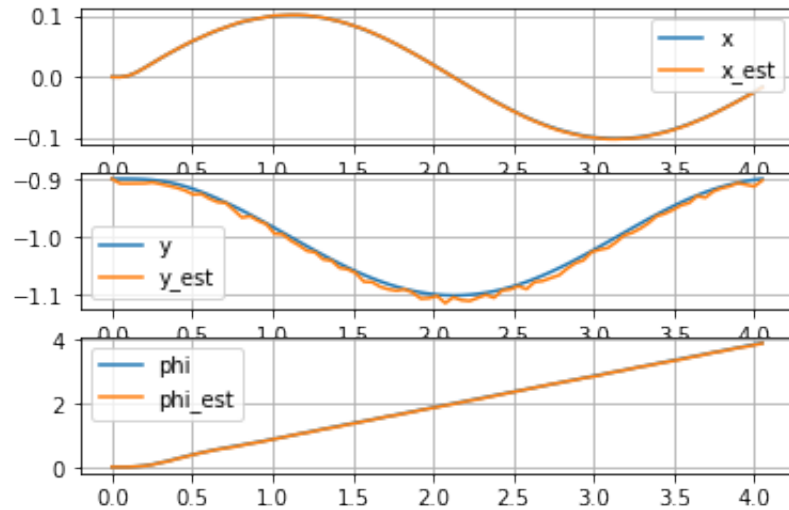
**Рис. 16:** Траектория движения робота с учетом постоянных возмущений.

### 3.2.2 Случай с использованием технического зрения

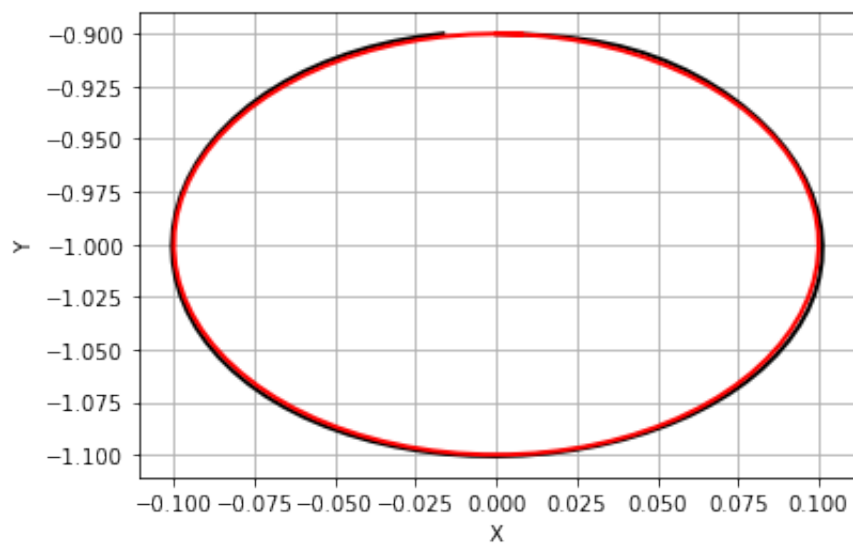
Точность работы технического зрения была рассмотрена в предыдущей главе. Поэтому сразу сравнивается результативность метода в различных условиях.



Теперь проверяется эффективность в случае отсутствия внешних возмущений, а динамический корректор отключен. Красная кривая — желаемая траектория движения, в то время как чёрная — фактическая. Результаты продемонстрированы на рис.17 и рис.18

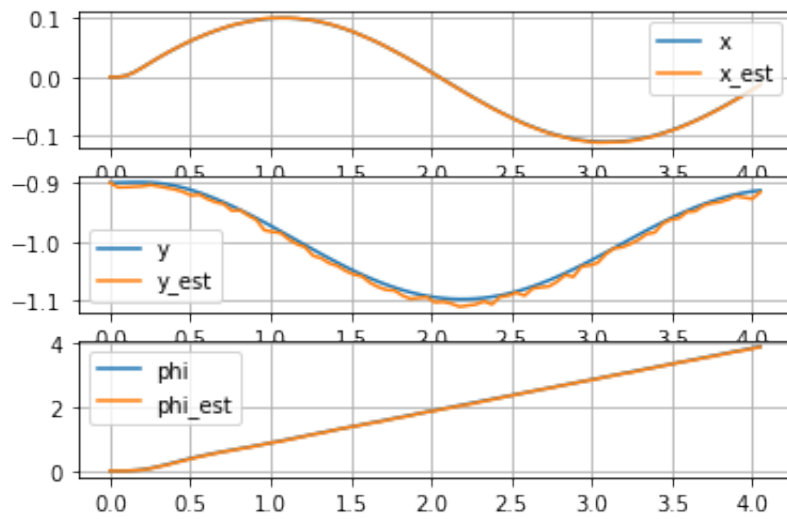


**Рис. 17:** Динамика движения робота без постоянного возмущения и корректора.

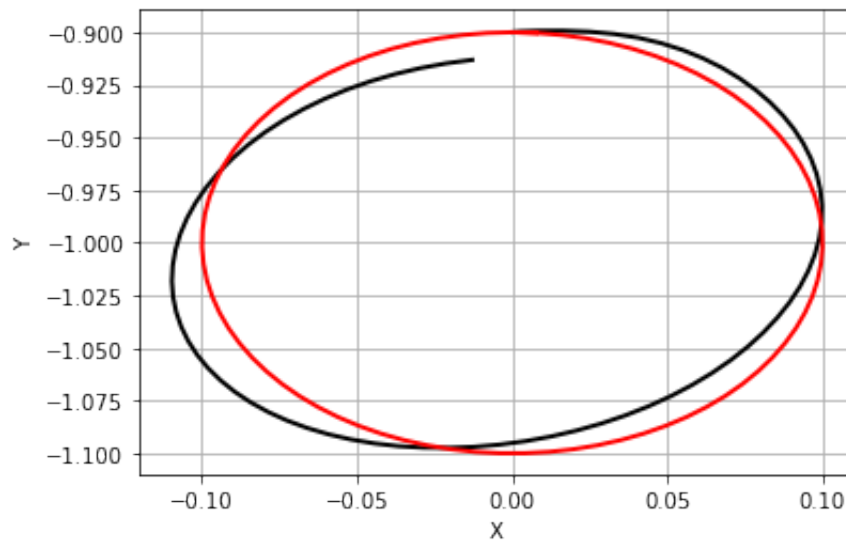


**Рис. 18:** Траектория движения робота без постоянного возмущения и корректора.

Рассматривается случай присутствия внешних возмущений, а динамический корректор отключен. Результаты продемонстрированы на рис.19 и рис.20

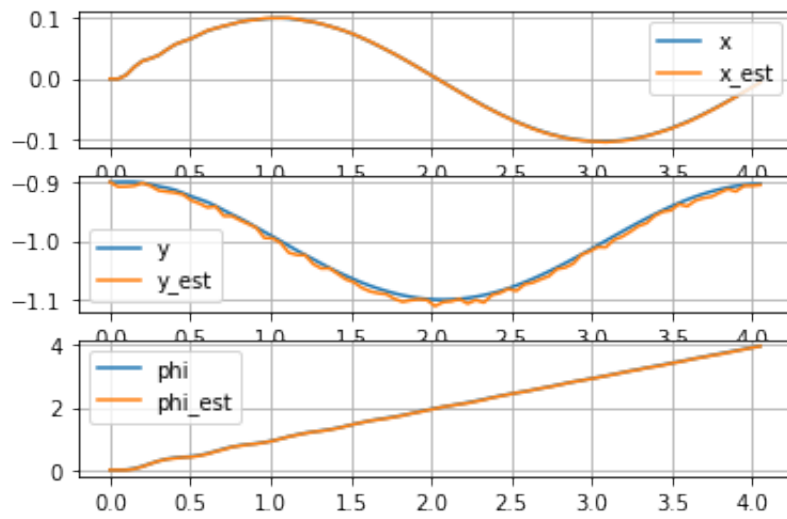


**Рис. 19:** Динамика движения робота с постоянными возмущениями, но без корректора.

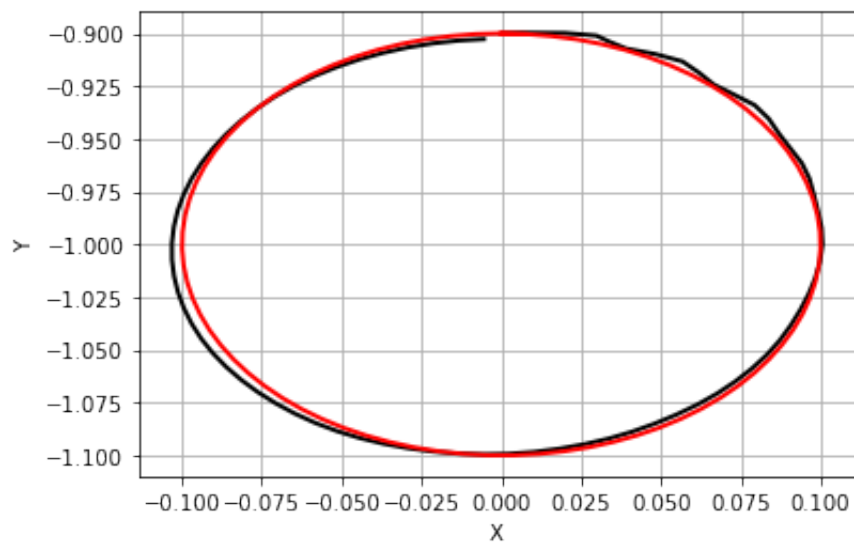


**Рис. 20:** Траектория движения робота с постоянными возмущениями, но без корректора.

И, наконец, случай внешних возмущений с включенным динамическим корректором. Результаты и разница заметны на рис.21 и рис.22.



**Рис. 21:** Динамика движения робота с постоянными возмущениями и корректором.



**Рис. 22:** Траектория движения робота с постоянными возмущениями и корректором.

### **3.3 Вывод**

В данной главе представлен подход к задаче движения робота по кругу с учетом внешних возмущений. Подход обеспечивается обратной связью от корректора и методов технического зрения. Результаты показывают, что поставленная задача выполнена.

## **Выводы**

В результате проделанной работы получены следующие результаты:

1. Синтезирован многоцелевой регулятор для обеспечения движения омниколесного робота по заданной траектории с учетом внешнего постоянного возмущения.

2. Реализован программный комплекс для моделирования движения объекта управления с использованием алгоритмов компьютерного зрения для определения положения робота.

3. Работоспособность представленного метода продемонстрирована на примере экспериментов с компьютерной моделью в различных режимах.

## **Заключение**

В данной работе показано, что многоцелевые регуляторы могут быть эффективным инструментом в задаче динамического позиционирования мобильных роботов, что наглядно демонстрируют результаты компьютерных экспериментов. В ближайшем будущем планируется, в первую очередь, опробовать представленные алгоритмы на реальном омниколесном роботе.

Помимо этого, целесообразным кажется использовать информацию о форме траектории для синтеза более эффективного управления. Также актуальными являются вопросы реакции на периодические возмущения, компенсации запаздывания, а также применения методов управления с прогнозирующими моделями (Model Predictive Control) для явного учета ограничений, наложенных на динамику системы.

## Список литературы

- [1] S. Garrido-Jurado, R. Muñoz-Salinas, F. J. Madrid-Cuevas, and M. J. Marín-Jiménez. Automatic generation and detection of highly reliable fiducial markers under occlusion // *Pattern Recogn.* 2014. 47, 6. P. 2280-2292.
- [2] Veremey E. I. Dynamical correction of positioning control laws // *Proc. of the 9th IFAC Conference on Control Applications in Marine Systems (CAMS-2013)*. Japan. 2013. P. 31—36.
- [3] Oliviera H. P., Sousa A. J., Costa P.J. Dynamical models for omni-directional robots with 3 and 4 wheels // *Proceedings of the 5th International Conference on Informatics in Control, Automation and Robotics*. Portugal. 2008. P. 189—196.
- [4] Bradski G. The OpenCV Library // *Dr Dobb's Journal of Software Tools*. 2000.
- [5] Севостьянов Р. А., // Программная поддержка процессов управления мобильным роботом с визуальной обратной связью // *Системы управления и информационные технологии*. 2019. № 4(78). С. 83—86.
- [6] Б.Т. Поляк, М.В. Хлебников, Л.Б. Рапопорт. Математическая теория автоматического управления. ИПУ РАН, ЛЕНАНД, 2019. — 504 с.
- [7] Mark Lutz, *Learning Python*, 5th Edition. 2003.
- [8] Eli Bressert. *SciPy and NumPy*, 2012.
- [9] Srinivasa Rao Poladi. *Matplotlib 3.0 Cookbook*. 2018.
- [10] L. Felipe Martins, Ruben Oliva Ramos, V Kishore Ayyadevara. *SciPy Recipes*, 2017.

## Приложения

Весь вспомогательный код, использовавшийся в работе, находится в репозитории по ссылке: <https://github.com/NikkiA5/diploma>