



X Научно-техническая конференция РТУ МИРЭА

Система управления динамикой мобильного робота с помощью жестов

Секция «Управление и информатика в технических системах»



Выполнил: студент группы КВБО-05-21
Обухова Анастасия Михайловна
Руководитель: старший преподаватель
Гурьянова Екатерина Олеговна

Москва, 2025

Цель и задачи



Цель: Разработка системы управления динамикой мобильного робота с помощью жестов, включающей алгоритмы компьютерного зрения для распознавания жестов и численную модель робота для оптимизации его динамических характеристик.

Задачи:

- Математическое моделирование динамики робота.
- Проверка адекватности модели.
- Синтез регуляторов.
- Разработка алгоритмов распознавания жестов.
- Разработка модуля принятия решений.
- Интеграция компонентов системы.

Эта работа была представлена на Всероссийском инженерном конкурсе (ВИК) в 2024 году и прошла в полуфинал.

Введение. Актуальность



Основные технологии управления с помощью жестикуляции

Управление роботами с помощью жестов становится всё более актуальным, поскольку оно улучшает взаимодействие человека и робота, предоставляя пользователям интуитивно понятные и естественные методы общения с роботами. Этот подход использует технологии компьютерного зрения и распознавания жестов, делая робототехнические системы более доступными и эффективными в различных областях применения.

Интуитивное взаимодействие

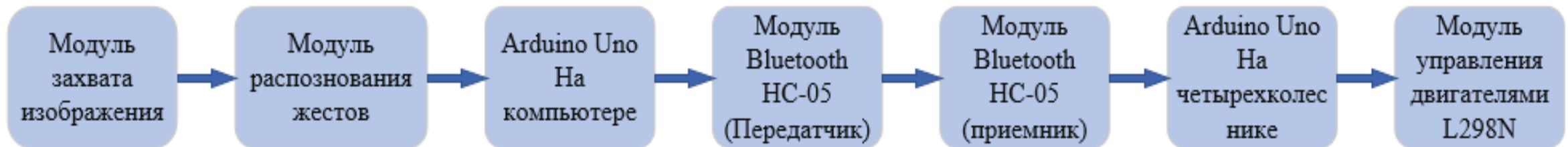
Системы управления на основе жестов позволяют пользователям управлять роботами с помощью простых движений рук, что соответствует естественным методам человеческого общения (Sara, 2024) (Janwe, 2023).

Эффективность и сокращение ошибок в производственных задачах

Управление жестами позволяет значительно сократить время программирования и ошибки операторов: исследования показали, что по сравнению с традиционными методами потери времени сокращаются до 70 % (Andrusyshyn et al., 2024).

Возможность программировать роботов с помощью жестов упрощает кривую обучения для операторов, делая робототехнику более доступной (Andrusyshyn et al., 2024).

Структурная схема



- Модуль захвата изображения: “Web-камера”
- Модуль распознавания жестов: “Python, MediaPipe”
- Arduino Uno на компьютере: “Передача команд”
- Bluetooth HC-05 (Передатчик): “Беспроводная связь”
- Bluetooth HC-05 (Приемник): “Прием команд”
- Arduino Uno на четырехколеснике: “Управление моторами”
- Модуль управления двигателями L298N: “Контроль скорости и направления”

Математическая модель

U Управляющее напряжение

Ke Коэффициент противо-ЭДС

omega_dv Угловая скорость двигателя

Ia Ток якоря

R Сопротивление обмотки якоря

Lia Индуктивность обмотки якоря

kM Коэффициент момента

B Коэффициент вязкого трения

Tc Величина кулоновского трения

J Момент инерции ротора

Rk Радиус колеса

B Ширина корпуса

theta Ориентация робота

Ток якоря: $d(Ia)/dt = (U - Ke * \omega_{dv} - Ia * R) / Lia$

Момент на колесе: $M_{wheel} = kM * Ia * 48$

Суммарный момент на колесе:

$M_{wheel_f} = M_{wheel} - B * \omega_{dv} - Tc$

Угловая скорость вала двигателя:

$d(\omega_{dv})/dt = M_{wheel_f} / J$

Линейная скорость колеса: $V = \omega_{dv} / 48 * Rk$

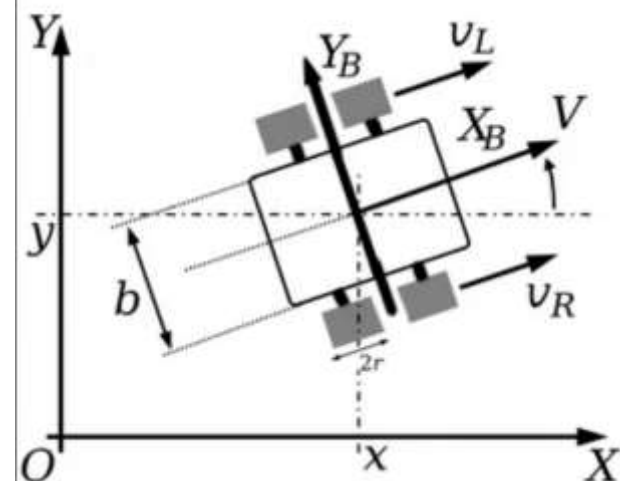
Угловая скорость робота:

$\omega = ((V_{r1} + V_{r2}) - (V_{l1} + V_{l2})) / b$

Перевод в глобальные координаты:

$Vx = V_{cp} * \cos(\theta) \quad Vy = V_{cp} * \sin(\theta)$

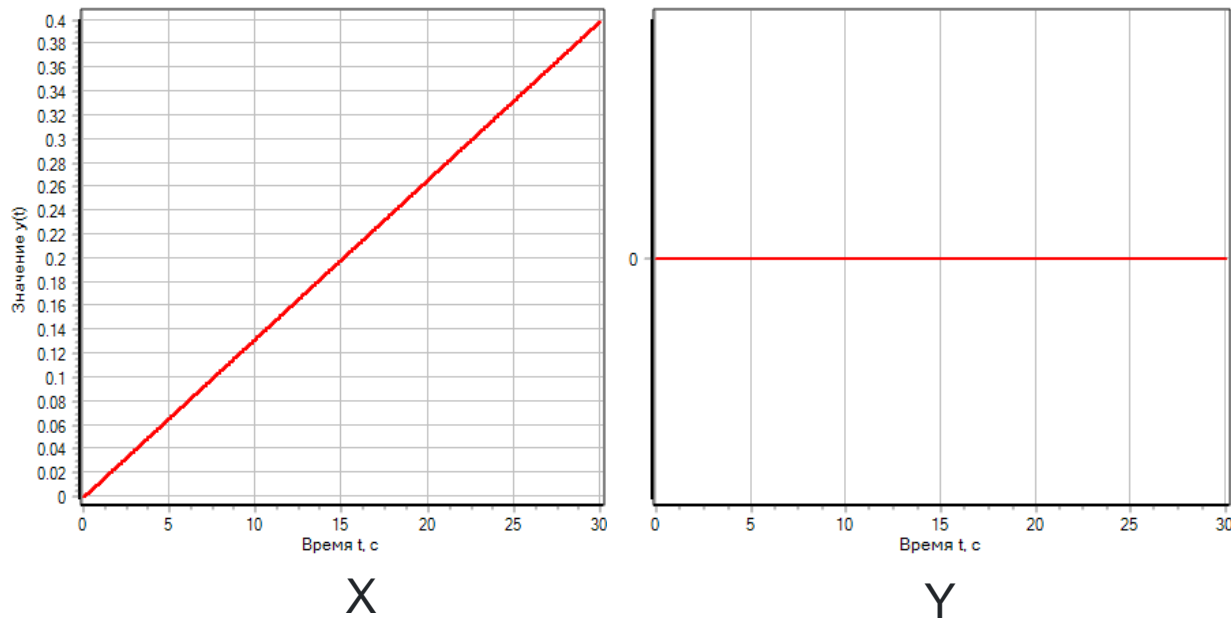
$X_{global} = \int(Vx) + X_{global_0} \quad Y_{global} = \int(Vy) + Y_{global_0}$



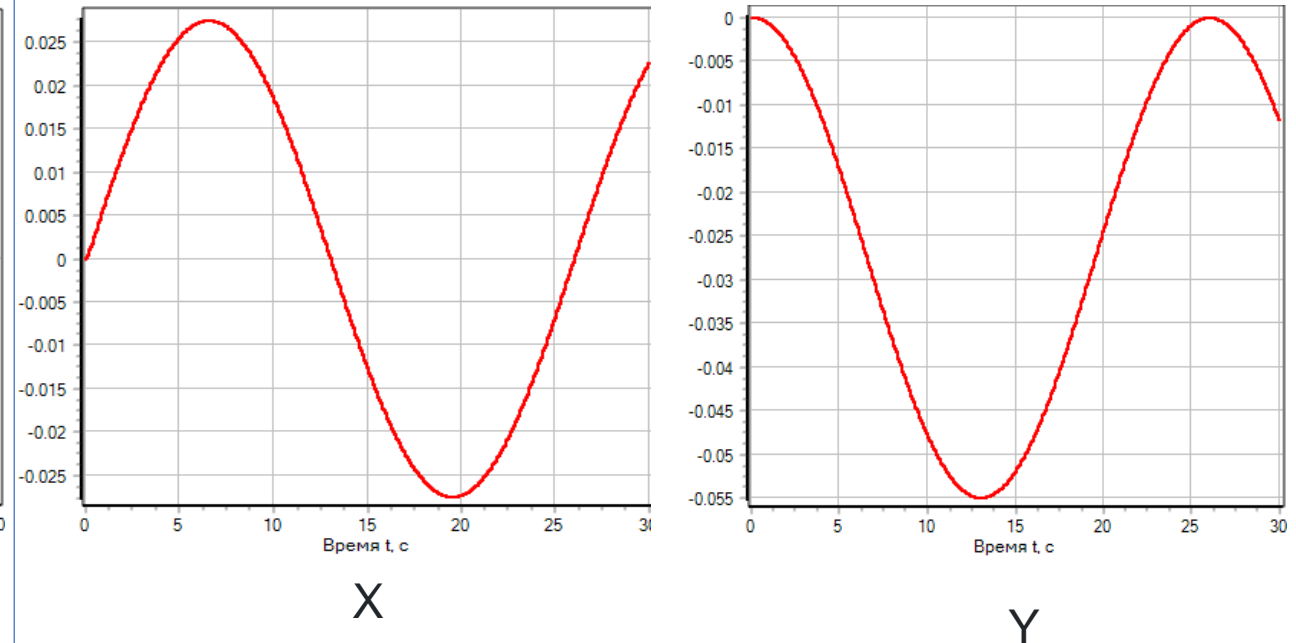
Компьютерное моделирование (SimInTech)



Движение по прямой
($U_{l1} = U_{r1} = U_{l2} = U_{r2} = 5$)



Поворот направо
($U_{l1} = U_{l2} = 0; U_{r1} = U_{r2} = 5$)

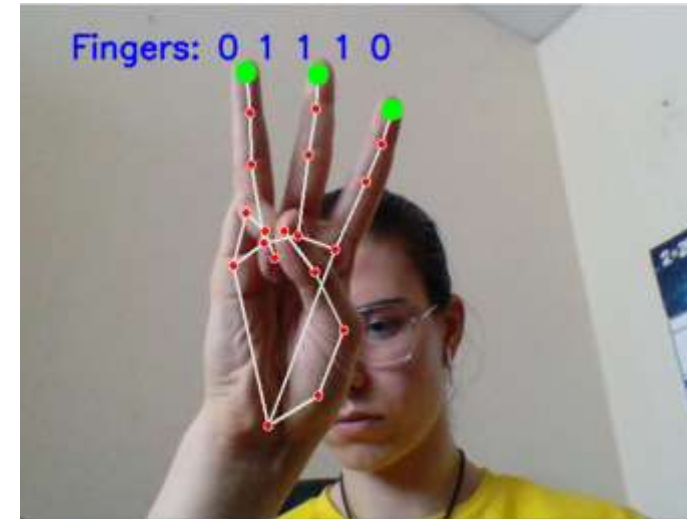


Алгоритм распознавания жестов (Python)

Преобразование видеопотока с камеры в команды управления роботом. Алгоритм на основе анализа координат ключевых точек руки (landmarks). MediaPipe Hands обеспечивает обнаружение рук и отслеживание 21 ключевой точки (landmarks) на руке в реальном времени.

Основные этапы:

- VideoCapture(0): Инициализация захвата видео с камеры.
- cvtColor(frame, COLOR_BGR2RGB): Преобразование цветового пространства.
- hands.process(frame): Обработка кадра MediaPipe Hands.
- hand_landmarks.landmark: Получение координат landmarks
- is_finger_extended(): Определение состояния пальцев.
- match_finger_state_text:: Сопоставление жестов с командами.



```
match finger_state_text:
    case "0 0 0 0 0": value_to_send = 0 # "Стоп"
    case "0 1 1 1 0": value_to_send = 1 # "Вперед"
    case _: value_to_send = -1 # Неизвестный жест
```

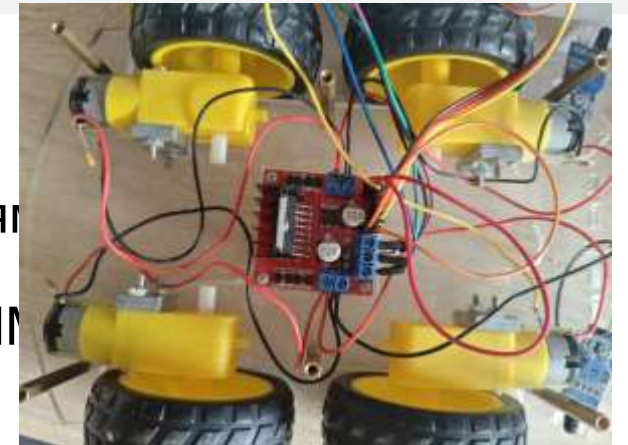

Управление роботом (Arduino)

Программа на Arduino, принимающая команды по Bluetooth и управляющая двигателями.

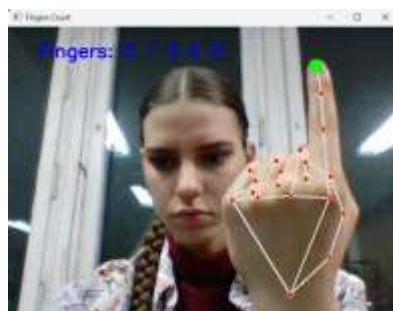
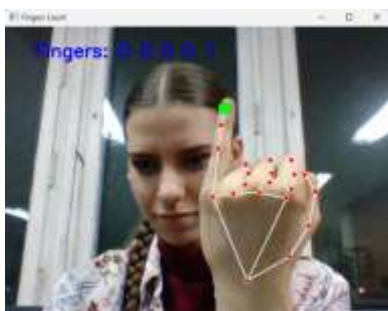
Основные этапы:

- SoftwareSerial BT(4, 7): Инициализация Bluetooth соединения.
- BT.available() > 0: Проверка наличия данных.
- command = BT.read(): Чтение команды.
- switch (command) { ... }: Обработка команды и управление двигателями.
- analogWrite(IN1, speed): Управление скоростью двигателя через ШИМ

```
switch (command) {  
  case '1': moveForward(100); break;  
  case '2': moveBackward(100); break;  
  ...  
  default: stopMotors(); break;  
}
```



Демонстрация работы системы



Спасибо за внимание!