Московский авиационный институт

(Национальный исследовательский университет)

Программирование БПЛА

**Лабораторные работы 1 - 12**

**Лаборатория 1.** Программирование архитектуры беспилотных летательных аппаратов (БПЛА) и их базовые отличия

**Лаборатория 2.** Программирование ПИД-регуляторов и их влияние на архитектуру управления (БПЛА)

**Лаборатория 3.** Взаимодействие элементов БПЛА через сокеты.

**Лаборатория 4.** Взаимодействие Наземной станции и БПЛА Протоколы.

**Лаборатория 5.** Управление Роторами квадрокоптера.

**Лаборатория 6**. Линейная модель квадрокоптера и управление.

**Лаборатория 7**. Нелинейная модель квадрокоптера и управление.

**Лаборатория 8**. Маневры: огибание препятствий, пикирование.

**Лаборатория 9**. Режим автоматического полета и Lidar

**Лаборатория 10.** Управляемый полет с камерой.

**Лаборатория 11**. ---

**Лаборатория 12**. ---

VVVVVVVaaaa \_\_\_\_\_\_\_\_

Подпись

\_\_\_\_\_\_\_\_/СеменовАС./

Подпись: Дата : \_\_ 2024

**Лаборатория 1.** Программирование архитектуры беспилотных летательных аппаратов (БПЛА) и их базовые отличия

**Вопросы:**

1. Каковы основные компоненты архитектуры БПЛА, и какую роль каждый из них играет в его функционировании?

2. В чем основные отличия между фиксированным крылом и многопрофильным БПЛА? Как это влияет на их программирование?

3. Что такое система управления полетом, и как она взаимодействует с другими системами на борту БПЛА?

4. Как алгоритмы маршрутизации и навигации могут различаться для летательных аппаратов разных типов (многороторные, фиксированное крыло и т.д.)?

5. Какова роль программного обеспечения в обеспечении безопасности БПЛА, и какие меры могут быть предприняты для предотвращения киберугроз?

**Задача 1. Проектирование архитектуры управления БПЛА.** Напишите программу на Python, которая моделирует базовую архитектуру системы управления БПЛА. Включите в нее следующие компоненты:

Система управления полетом (Flight Control System, FCS)

* Гироскопы (Gyroscopes): Измеряют вращательное движение.
* Акселерометры (Accelerometers): Измеряют линейное ускорение.
* Магнитометры (Magnetometers): Обеспечивают информацию о направлении.

Система навигации (Navigation System):

* GPS: Обеспечивает данные о местоположении для навигации и позиционирования.
* Инерциальные измерительные устройства (Inertial Measurement Units, IMUs): Помогают отслеживать ориентацию и движение БПЛА

**Системы мониторинга и диагностики (Monitoring and Diagnostic Systems)**

Энергоснабжение (Power Supply):

Система роторных двигателей (Rotor system)

Система связи (Communication System):

* Передатчик и приемник (Transmitter and Receiver): Обеспечивают связь между БПЛА и наземной станцией управления. Это может включать радиосигналы или телеметрические данные.
* Данные связи (Data Link): Для передачи видео, данных с датчиков и команд управления.

Определите классы для каждой из этих систем и реализуйте методы, которые демонстрируют взаимодействие между ними. Например, система управления полетом должна получать данные от сенсорной системы и отправлять команды навигации.

class FlightControlSystem:

def \_\_init\_\_(self):

self.altitude = 0

def receive\_sensor\_data(self, sensor\_data):

# Обрабатывать данные от сенсоров и корректировать высоту

self.altitude = sensor\_data['altitude']

print(f"Flight Control System received altitude: {self.altitude}")

def send\_navigation\_command(self):

# Отправка команды на систему навигации

print("Sendingnavigationcommand.")

class SensorSystem:

def collect\_data(self):

# Симуляция сбора данных от сенсоров

return {'altitude': 100}

class NavigationSystem:

def \_\_init\_\_(self):

self.current\_position = (0, 0)

def update\_position(self, new\_position):

self.current\_position = new\_position

print(f"Navigation System updated position to: {self.current\_position}")

class CommunicationSystem:

def send\_message(self, message):

print(f"Communication System: {message}")

# Примервзаимодействия

fcs = FlightControlSystem()

sensors = SensorSystem()

nav\_system = NavigationSystem()

comm\_system = CommunicationSystem()

sensor\_data = sensors.collect\_data()

fcs.receive\_sensor\_data(sensor\_data)

fcs.send\_navigation\_command()

**Лаборатория 2.** Программирование ПИД-регуляторов и их влияние на архитектуру управления (БПЛА)

**Вопросы:**

1. Что такое ПИД-регулятор, и какие элементы в него входят? Опишите пропорциональную, интегральную и дифференциальную составляющие.

2. Как применение ПИД-регулятора влияет на устойчивость и управление движением БПЛА? Какие преимущества и недостатки могут возникнуть при использовании ПИД-регуляторов в системах управления БПЛА?

3. Как можно протестировать эффективность ПИД-регулятора при управлении БПЛА? Какие методы и инструменты могут быть использованы для симуляции и оценки производительности ПИД-регулятора?

4. Каковы основные параметры, которые нужно настраивать в ПИД-регуляторе для оптимизации управления БПЛА? Обсудите, как изменение коэффициентов P, I и D влияет на реакцию системы.

5. В каких случаях может быть целесообразно использовать ПИД-регулятор вместо других типов регуляторов в системах управления БПЛА? Приведите примеры ситуаций, когда ПИД-регулятор может оказаться наиболее подходящим решением.

**Задачи:**

**1. Задача на численное интегрирование:**Напишите программу на Python, которая моделирует систему управления БПЛА с использованием ПИД-регулятора. Используйте дискретные шаги времени и численные методы (например, метод Эйлера) для интегрирования. Вам нужно реализовать следующее:

- Определите целевую высоту, на которую должен подняться БПЛА.

- Программируйте ПИД-регулятор с возможностью настройки коэффициентов P, I и D.

- Сравните полученную высоту со целевой и выведите график изменения высоты БПЛА во времени.

**2. Задача на анализ устойчивости:**Напишите программу на Python, которая анализирует устойчивость системы управления БПЛА с ПИД-регулятором. Вам нужно реализовать следующее:

Дана модель динамики БПЛА в простейшем приближении:Модель учитывает

подъем БПЛА и управляет его высотой на основе заданной целевой высоты.

БПЛА имеет простейшую динамику: его высота изменяется по времени в зависимости от разности между целевой и текущей высотой, а также от текущей скорости подъема.

**КодмоделидинамикиБПЛА**

import numpy as np

import matplotlib.pyplot as plt

# Параметры

target\_height = 100.0 # Целевая высота (в метрах)

initial\_height = 0.0 # Начальная высота

Kp = 1.0 # Коэффициент пропорциональной части

Ki = 0.1 # Коэффициент интегральной части

Kd = 0.5 # Коэффициент дифференциальной части

# Константы

dt = 0.1 # Шаг времени (в секундах)

total\_time = 20.0 # Общее время моделирования (в секундах)

num\_steps = int(total\_time / dt)

# Массивы для хранения значений

heights = np.zeros(num\_steps)

errors = np.zeros(num\_steps)

integral = 0.0

previous\_error = 0.0

# Начальнаявысота

heights[0] = initial\_height

# Главныйциклмоделирования

for step in range(1, num\_steps):

# Вычислениеошибки

current\_height = heights[step - 1]

error = target\_height - current\_height

errors[step] = error

integral += error \* dt #Интеграл

derivative = (error - previous\_error) / dt #Производная

# ПИД-регулятор

control\_signal = Kp \* error + Ki \* integral + Kd \* derivative

# Обновлениевысоты

heights[step] = current\_height + control\_signal \* dt

previous\_error = error

# Визуализациярезультатов

time = np.arange(0, total\_time, dt)

plt.figure(figsize=(12, 6))

plt.plot(time, heights, label='ВысотаБПЛА', color='blue')

plt.axhline(y=target\_height, color='red', linestyle='--', label='Целеваявысота')

plt.title('Динамика высоты БПЛА с использованием ПИД-регулятора')

plt.xlabel('Время (с)')

plt.ylabel('Высота (м)')

plt.legend()

plt.grid()

plt.show()

Описание кода

1. Параметры: Определены величины для целевой высоты и начальной высоты, а также коэффициенты ПИД-регулятора: `Kp`, `Ki` и `Kd`.

2. Инициализация: Создаются массивы для хранения текущей высоты, ошибки и интеграла.

3. Цикл моделирования: В каждом шаге времени:

- Вычисляется ошибка между текущей и целевой высотой.

- Обновляется интеграл и вычисляется производная.

- Рассчитывается сигнал управления с помощью формулы ПИД-регулятора.

- Обновляется высота БПЛА на основе значения сигнала управления.

4. Визуализация: Результаты выводятся на графике, где отображается изменяющаяся высота БПЛА и целевая высота.

Программируйте ПИД-регулятор и протестируйте систему на различных значениях коэффициентов P, I и D, меняя один коэффициент за раз.

Оцените устойчивость системы по времени ответа и величине колебаний в зависимости от настроек. Выведите результаты в виде графиков и сделайте выводы о том, как выбранные параметры влияют на устойчивость системы управления.

**Лаборатория 3.** Взаимодействие элементов БПЛА через сокеты.

**Вопросы:**

1. Что такое сокет и какова его роль в взаимодействии компонентов БПЛА? Ответ должен содержать определение сокета, его основные функции и примеры использования в системе БПЛА.

2. Объясните разницу между клиентом и сервером в контексте сокетов. Как оба этих компонента могут взаимодействовать в системе БПЛА?Ответ должен описывать основные различия, а также примеры реализации на Python.

3. Какие протоколы передачи данных могут использоваться при взаимодействии элементов БПЛА через сокеты? Приведите примеры. Важно упомянуть TCP и UDP, их характеристики и случаи применения в БПЛА.

4. Как можно обрабатывать исключения при работе с сокетами в Python? Приведите пример кода, иллюстрирующий это. Ответ должен включать краткое объяснение обработки исключений и пример кода с использованием `try` и `except`.

5. Какие меры безопасности следует предпринимать при передаче данных между элементами БПЛА с использованием сокетов?Должны быть указаны основные подходы к обеспечению безопасности: шифрование данных, аутентификация и авторизация.

**Задача 1:** Реализация простого сервера и клиента.

- Напишите серверное приложение на Python, которое будет принимать сообщения от клиента и отвечать на них. Клиент должен подключаться к серверу и отправлять строку сообщения. Сервер должен ответить, добавив префикс "Server: " к полученному сообщению.

**Указания:**

- Используйте модуль `socket`.

- Сервер должен слушать на заданном порту и принимать только одно соединение.

- Клиент должен работать в бесконечном цикле, позволяя пользователю отправлять несколько сообщений подряд.

**Задача 2:** Передача команд от управляющего устройства к беспилотному летательному аппарату.

- Реализуйте простую систему, состоящую из сервера (БПЛА) и клиента (управляющее устройство). Клиент должен отправлять команды (например, "MOVE", "STOP", "TAKEOFF") на сервер, а сервер должен их обрабатывать и выводить соответствующие действия в консоль.

**Указания:**

- Сервер и клиент должны обмениваться информацией через сокеты.

- Реализуйте простую логику обработки команд на стороне сервера.

- Добавьте возможность завершения работы клиента по команде "EXIT".

**Лаборатория 4.** Взаимодействие Наземной станции и БПЛА Протоколы.

**Вопросы:**

1. Какие протоколы обычно используются для связи между наземной станцией и БПЛА? Ответ предполагает исследование таких протоколов, как MAVLink, DJI SDK, и других коммуникационных стандартов.

2. Каковы основные компоненты, составляющие архитектуру взаимодействия между БПЛА и наземной станцией? Ожидается описание компонентов, таких как приёмники, передатчики, контроллеры и сам БПЛА.

3. Как реализовать шифрование данных в протоколе связи между БПЛА и наземной станцией? Вопрос нацелен на изучение методов шифрования, таких как AES или RSA, и их применение в коде.

4. Какие меры необходимо принять для обеспечения надежности связи между наземной станцией и БПЛА? Ответ должен включать методы проверки целостности, обработку ошибок и механизмы подтверждения передачи.

5. Какова роль программного обеспечения в управлении коммуникацией между БПЛА и наземной станцией? Ожидается обсуждение важности программного обеспечения, драйверов и библиотек в осуществлении взаимодействия.

**Задача 1:** Реализация простого клиента-сервера для обмена сообщениями между наземной станцией и БПЛА. Напишите программу на Python, которая реализует клиент-серверное взаимодействие с использованием библиотеки `socket`. Сервер будет представлять наземную станцию, а клиент — БПЛА. Реализуйте обмен текстовыми сообщениями, где клиент (БПЛА) отправляет команду, а сервер (наземная станция) отвечает статусом выполнения команды.

# Пример для сервера

importsocket

server\_socket = socket.socket(socket.AF\_INET, socket.SOCK\_STREAM)

server\_socket.bind(('localhost', 65432))

server\_socket.listen()

print("Сервер запущен, ожидание соединения...")

conn, addr = server\_socket.accept()

print('Подключенок', addr)

while True:

data = conn.recv(1024)

if not data:

break

print('Получено сообщение:', data.decode())

conn.sendall(b'Команда принята')

conn.close()

**Задача 2:** Создание системы мониторинга состояния БПЛА с использованием протокола MAVLink. Используя библиотеку `pymavlink`, создайте скрипт, который будет подключаться к БПЛА по протоколу MAVLink и получать состояние системы (например, уровень заряда батареи, координаты и статус подключения). Обработайте полученные данные и выведите их на экран.

frompymavlinkimportmavutil

# Установка соединения с БПЛА

master = mavutil.mavlink\_connection('udp:127.0.0.1:14550')

while True:

msg = master.recv\_match(type='HEARTBEAT', blocking=True)

print("Получен HEARTBEAT от БПЛА")

# Получение состояния системы

battery\_status = master.recv\_match(type='BATTERY\_STATUS', blocking=True)

print(f"Уровеньзаряда: {battery\_status.battery\_remaining}%")

**Лаборатория 5.**Управление Роторами квадрокоптера.

**Вопросы:**

1. Какова роль каждого из четырех роторов в управлении квадрокоптером ? Ответ: Объясните, как каждый ротор (передний левый, передний правый, задний левый, задний правый) влияет на движение квадрокоптера (подъем, спуск, повороты, движение вперед и назад).

2. Что такое PWM и как он используется для управления роторами квадрокоптера? Ответ: Опишите, что такое широтно-импульсная модуляция (PWM) и как она используется для регулирования скорости роторов.

3. Как изменение скорости отдельных роторов влияет на ориентацию квадрокоптера в пространстве? Ответ: Объясните, как увеличение или уменьшение скорости конкретных роторов может вызвать наклон или поворот квадрокоптера.

4. Какие алгоритмы управления используются в квадрокоптерах для стабилизации полета? - Ответ: Назовите и опишите алгоритмы, такие как PID-регулирование, и их применение в контроле полета.

5. Какие сенсоры могут быть использованы для получения информации о положениях и ориентации квадрокоптера? Ответ: Укажите сенсоры, такие как гироскопы, акселерометры и барометры, и их роль в управлении движением.

**Задача 1.** Управление скоростью роторов: Напишите функцию на Python, которая принимает в качестве входных данных желаемую высоту и текущую высоту квадрокоптера. Функция должна вычислять необходимую скорость для каждого из четырех роторов, чтобы достичь целевой высоты с учетом заданного времени подъема.

def calculate\_motor\_speeds(target\_height, current\_height, time\_to\_reach):

# Примернаяреализация

height\_difference = target\_height - current\_height

motor\_speed = height\_difference / time\_to\_reach #Скоростьизменениявысоты

# Пример распределения скорости между роторами

speeds = [motor\_speed] \* 4 # Упрощенная модель, все моторы работают одинаково

returnspeeds

**Задача 2.** Регулирование ориентацией квадрокоптера:Напишите код, который имитирует поворот квадрокоптера на 90 градусов влево. Используйте скорости роторов, чтобы достичь этого поворота. Каждому переднему и заднему ротору нужно будет увеличить/уменьшить скорость для создания вращающего момента.

def rotate\_left(rotor\_speeds):

# Увеличиваем скорость задних ротор и уменьшаем передние

rotor\_speeds[0] -= 50 # Уменьшаем скорость переднего левого

rotor\_speeds[1] -= 50 # Уменьшаем скорость переднего правого

rotor\_speeds[2] += 50 # Увеличиваем скорость заднего левого

rotor\_speeds[3] += 50 # Увеличиваем скорость заднего правого

returnrotor\_speeds

**Лаборатория 6**. Линейная модель квадрокоптера и управление.

**Вопросы:**

1. Что такое линейная модель квадрокоптера, и какие основные допущения делаются в процессе ее разработки?

2. Каково значение каждого коэффициента в PID-контроллере для управления квадрокоптером, и как их настройка влияет на поведение системы?

3. Как линейная модель квадрокоптера может быть улучшена для учета влияния внешних возмущений, таких как порывы ветра или изменения в весе нагрузки?

4. Как формулируется задача управления высотой квадрокоптера с использованием линейной модели, и какие уравнения нужно решить для достижения стабильной высоты?

5. Какие преимущества и недостатки имеет использование линейной модели квадрокоптера по сравнению с нелинейной моделью?

**Задача 1.** Оценка стабильной высоты

У вас есть линейная модель квадрокоптера с массой 1 кг и подъемной силой, которая может достигать 15 Н. Настройте PID-контрольную систему для управления высотой, чтобы достичь целевой высоты 10 м. Известно, что коэффициенты PID равны Kp = 8, Ki = 0.5 и Kd = 2. Моделируйте систему в течение 20 секунд и постройте график высоты над временем, учитывая воздействие силы тяжести (g = 9.81 м/с²).

Для решения данной задачи необходимо написать симуляцию, используя формулировку движения квадрокоптера в зависимости от управляющего воздействия, и построить график.

**Задача 2.** Анализ чувствительности систем управления

Рассмотрите линейную модель квадрокоптера, где высота зависит от трех параметров: Kp, Ki и Kd. Измените значение Kp на 20% выше и 20% ниже, сохранив остальные параметры постоянными. Рассчитайте, как изменение Kp влияет на время, необходимое для достижения целевой высоты 5 м, и на устойчивость системы.У вас есть линейная модель квадрокоптера с массой 1 кг и подъемной силой, которая может достигать 15 Н. Настройте PID-контрольную систему для управления высотой, чтобы достичь

**Лаборатория 7**. Нелинейная модель квадрокоптера и управление

**Вопросы:**

1. Что такое нелинейная модель квадрокоптера и как она отличается от линейной модели?
2. Каковы основные факторы, влияющие на динамическое поведение квадрокоптера в нелинейной модели?
3. Какой подход можно использовать для управления квадрокоптером на основе нелинейной модели? Ответ может включать обсуждение различных методов, таких как PID-контроль, линейное управление, управление с помощью нейронных сетей и другие современные методы.
4. Почему важно учитывать нелинейные эффекты при моделировании и управлении квадрокоптером? Здесь ожидается объяснение того, как игнорирование нелинейности может привести к ошибкам в управлении и нестабильности.
5. Какие практические примеры применения нелинейных моделей квадрокоптеров можно привести в реальной жизни?

**Задача 1.** Моделирование высоты.

Рассмотрим квадрокоптер с массой 1.5 кг и моментом инерции 0.15. У него есть целевая высота 10 м. Напишите уравнения движения для модели квадрокоптера с учетом нелинейных сил (например, сопротивление воздуха) и настройте параметры управления для достижения целевой высоты. Защитите свое решение через численное моделирование и построение графиков.

**Задача 2.** Оптимизация управления

У вас есть данные о поведении квадрокоптера в условиях безветренной среды и в условиях сильного ветра. Опишите методы, которые могут быть использованы для адаптации управляющих сигналов в ситуации с сильным ветром. Кроме того, создайте алгоритм, который будет автоматически подстраивать параметры PID-контроллера в зависимости от условий окружающей среды. Обоснуйте свои выборы и предположения**.**

**Лаборатория 8**. Маневры: огибание препятствий, пикирование.

**Вопросы:**

1. Что такое маневр огибания препятствий и в каких ситуациях он применяется в авиации?

2. Какие основные техники пикирования используются пилотами для безопасного снижения высоты?

3. Как влияет скорость самолета на эффективность маневра огибания препятствий?

4. Какие факторы необходимо учитывать при выполнении маневра пикирования?

5. Каковы основные различия между огибанием препятствий и пикированием с точки зрения аэродинамики?

**Задача 1:** Огибание препятствия. БПЛА на высоте 100 метров столкнулся с неожиданным препятствием на расстоянии 200 метров впереди. Скорость БПЛА составляет 20 м/с. Автопилот решает выполнить маневр огибания, снизив скорость до 15 м/с. Рассчитайте, какой минимальный радиус поворота необходим, чтобы успешно обойти препятствие, если угол разворота не должен превышать 30 градусов.

**Задача 2:** Пикирование. БПЛА находится на высоте 300 метров и начинает пикирование под углом 45 градусов для выполнения экстренной посадки. Скорость БПЛА на момент начала пикирования составляет 25 м/с. Определите:

a) сколько времени потребуется для достижения высоты 100 метров,

b) какую вертикальную скорость должен развивать БПЛА во время пикирования, чтобы достичь высоты 100 метров за это время.

**Лаборатория 9**. Режим автоматического полета

**Задачи:**

**Лаборатория 10.** Управляемый полет с камерой.

**Вопросы:**

**Задачи:**

**Лаборатория 11**.--

**Вопросы:**

**Лаборатория 12**.--

**Вопросы:**

**Задачи:**