Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Выборнова Любовь Алексеевна
Должность: Ректор
Дата подписания: 29.10.2026 15:09:53
Уникальный программный ключ:
c3b3b9c625f6c113afa2a2d13baff9e05a38b76e

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное бюджетное образовательное учреждение
высшего образования
«Поволжский государственный университет сервиса» (ФГБОУ ВО «ПВГУС»)

Высшая школа передовых производственных технологий

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

«ОСНОВЫ БЕСПИЛОТНЫХ ТЕХНОЛОГИЙ И СИСТЕМ»

Направление подготовки:
09.03.02 «Информационные системы и технологии»

Направленность (профиль):
«Информационные системы и технологии»

Квалификация выпускника: **бакалавр**

Тольятти 2026

Рабочая программа дисциплины «*Основы беспилотных технологий и систем*» разработана в соответствии с Федеральным государственным образовательным стандартом высшего образования - бакалавриата по направлению подготовки 09.03.03 Прикладная информатика, утверждённым приказом Министерства образования и науки Российской Федерации от 19.09.2017 № 922 (ред. от 08.02.2021)

Составители:

к.т.н., доцент (учёная степень, учёное звание) Т.С. Яницкая (ФИО)

РПД утверждена на заседании Высшей школы передовых производственных технологий *(ВШППТ)*

30.08.2026г. протокол №1

директор *ВШППТ* д.т.н., профессор (учёная степень, учёное звание) В.И. Воловач (ФИО)

1. ПЕРЕЧЕНЬ ПЛАНИРУЕМЫХ РЕЗУЛЬТАТОВ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ, СООТНЕСЕННЫХ С ПЛАНИРУЕМЫМИ РЕЗУЛЬТАТАМИ ОСВОЕНИЯ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ

1.1. Цель освоения дисциплины

Целью освоения дисциплины является:

- формирование у обучающихся общепрофессиональных компетенций в области использования информационно-коммуникационных технологий.

1.2. Перечень планируемых результатов обучения по дисциплине

Код и наименование компетенции	Код и наименование индикатора достижения компетенции	Планируемые результаты обучения по дисциплине
УК-6. Способен управлять своим временем, выстраивать и реализовывать траекторию саморазвития на основе принципов образования в течение всей жизни	ИУК-6.1. Определяет приоритеты собственной деятельности, личностного развития, образовательного и профессионального роста; подбирает способы решения и средства развития, в том числе в цифровой среде ИУК-6.2. Владеет технологиями и навыками управления своей познавательной деятельностью и ее совершенствования на основе самооценки, самоконтроля и принципов самообразования в течение всей жизни	**Знает:** принципы работы беспилотных систем, их классификацию, области применения, правовые и этические аспекты. **Умеет:** анализировать влияние беспилотных технологий на различные сферы деятельности и оценивать их перспективы. **Владеет:** навыками работы с онлайн-симуляторами для базового взаимодействия с моделями беспилотных систем.
ОПК-2 Способен понимать принципы работы современных информационных технологий и программных средств, в том числе отечественного производства, и использовать их при решении задач профессиональной деятельности	ИОПК-2.1. Понимает роль цифровой культуры в информационном обществе и профессиональной деятельности; знает современные информационные технологии и программные средства, возможности их использования профессиональной деятельности ИОПК-2.2. Различает принципы работы бизнес-ориентированных языков программирования с учетом их преимуществ, недостатков, сфер применения	

2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ

Дисциплина относится к обязательной части, Блока 1. Дисциплины (модули) образовательной программы (Общепрофессиональный модуль).

3. СТРУКТУРА И СОДЕРЖАНИЕ ДИСЦИПЛИНЫ

3.1. Объём учебной дисциплины и виды учебной работы

Общая трудоёмкость дисциплины составляет **2 з.е. (72 час.)**, их распределение по видам работ и семестрам представлено в таблице.

Виды учебных занятий и работы обучающихся	**Трудоёмкость, час**
Общая трудоёмкость дисциплины, час	**72**
Контактная работа обучающихся с преподавателем по видам учебных занятий (всего), в т.ч.:	**8**
Занятия лекционного типа (лекции)	**4**
Занятия семинарского типа (семинары, практические занятия, практикумы, коллоквиумы и иные аналогичные занятия)	**4**
Самостоятельная работа всего, в т.ч.:	**60**
Самоподготовка по темам (разделам) дисциплины	**60**
Выполнение курсового проекта /курсовой работы	**-/-**
Контроль (часы на экзамен, зачет)	**-/4**
Промежуточная аттестация	**зачёт**

Примечание: *-/- объем часов соответственно для заочной форм обучения*

Дисциплина реализуется посредством проведения контактной работы с обучающимися (включая проведение текущего контроля успеваемости), самостоятельной работы обучающихся и промежуточной аттестации.

В процессе освоения дисциплины может применяться электронное обучение и дистанционные образовательные технологии.

В процессе освоения дисциплины обучающиеся обеспечены доступом к электронной информационно-образовательной среде и электронно-библиотечным системам.

3.2. Содержание дисциплины, структурированное по темам

Планируемые результаты освоения: код формируемой компетенции и индикаторы достижения компетенций	**Наименование разделов, тем**	**Виды учебной работы ***			**Формы текущего контроля (наименование оценочного средства)**
		Контактная работа			
		Лекции, час	**Практические занятия, час**	**Самостоятельная работа, час**	
УК-6. ИУК-6.1, ИУК-6.2; ОПК-2. ИОПК-2.1, ИОПК-2.2	Тема 1. Основы и классификация беспилотных систем. 1.1. Определения, история развития и терминология беспилотных систем. 1.2. Классификация по среде функционирования (воздушные, наземные, водные). 1.3. Классификация по назначению, массе и степени автономности.	0,5			Внеаудиторное тестирование (тест)
	Самостоятельная работа			10	Самостоятельное изучение учебных материалов
	Тема 2. Структура и подсистемы беспилотных систем. 2.1. Обобщённая архитектура: воздушный и наземный сегменты. 2.2. Основные компоненты: датчики, вычислительные блоки, исполнительные механизмы. 2.3. Взаимодействие подсистем и контур управления.	0,5			Внеаудиторное тестирование (тест)

Планируемые результаты освоения: код формируемой компетенции и индикаторы достижения компетенций	Наименование разделов, тем	Виды учебной работы *			Формы текущего контроля (наименование оценочного средства)
		Контактная работа		Самостоятельная работа, час	
		Лекции, час	Практические занятия, час		
	Практическое занятие №1. Первое знакомство со средой Webots и базовое управление дроном		0,5		Отчёт по практическому занятию
	Самостоятельная работа			10	Самостоятельное изучение учебных материалов
	Тема 3. Базовые технологии беспилотных систем. 3.1. Двигательные установки и системы энергоснабжения. 3.2. Коммуникационные системы и каналы связи (радиоуправление, телеметрия). 3.3. Навигационные системы: GNSS, IMU, барометр, магнитометр.	0,5			Внеаудиторное тестирование (тест)
	Практическое занятие №2. Программирование контроллера -- автоматический взлёт и движение по прямой		0,5		Отчёт по практическому занятию
	Самостоятельная работа			10	Самостоятельное изучение учебных материалов
	Тема 4. Полезные нагрузки и сенсорные системы. 4.1. Виды полезных нагрузок и их назначение. 4.2. Сенсорные системы: видеокамеры, лидары, радары, ультразвуковые сенсоры. 4.3. Стабилизированные подвесы и системы наведения.	0,5			Внеаудиторное тестирование (тест)
	Практическое занятие №3. Реализация PID-контроллера высоты и позиции		0,5		Отчёт по практическому занятию
	Самостоятельная работа			10	Самостоятельное изучение учебных материалов
	Тема 5. Методология проектирования и моделирования. 5.1. Инженерный подход к проектированию: технические требования и CONOPS. 5.2. Моделирование и симуляция беспилотных систем. 5.3. Верификация и валидация (V&V).	1			Внеаудиторное тестирование (тест)
	Практическое занятие №4. Чтение и обработка данных датчиков -- GPS и IMU		0,5		Отчёт по практическому занятию
	Практическое занятие №5. Обнаружение препятствий с помощью дальномеров		0,5		Отчёт по практическому занятию
	Самостоятельная работа			10	Самостоятельное изучение учебных

Планируемые результаты освоения: код формируемой компетенции и индикаторы достижения компетенций	Наименование разделов, тем	Виды учебной работы *			Формы текущего контроля (наименование оценочного средства)
		Контактная работа		Самостоятельная работа, час	
		Лекции, час	Практические занятия, час		
					материалов
	Тема 6. Испытания, надёжность и перспективы развития. 6.1. Виды и методы испытаний беспилотных систем. 6.2. Основы надёжности и отказоустойчивости. 6.3. Нормативно-правовые аспекты и перспективы развития.	1			Внеаудиторное тестирование (тест)
	Практическое занятие №6. Компьютерное зрение -- обнаружение целевых объектов через камеру		0,5		Отчёт по практическому занятию
	Практическое занятие №7. Автономная навигация по пути -- алгоритм A*		0,5		Отчёт по практическому занятию
	Практическое занятие №8. Мультидрон -- симуляция двух дронов и базовая синергия		0,5		Отчёт по практическому занятию
	Практическое занятие №9. Интеграция и демонстрация -- комплексная задача		0,5		Отчёт по практическому занятию
	Самостоятельная работа			10	Самостоятельное изучение учебных материалов
	ИТОГО	**4**	**4**	**60**	Зачёт

Примечание *: «/» - *объем часов соответственно для заочной формы обучения*

4. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОБУЧАЮЩИХСЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ

4.1. Общие методические рекомендации по освоению дисциплины, образовательные технологии

Дисциплина реализуется посредством проведения контактной работы с обучающимися (включая проведение текущего контроля успеваемости), самостоятельной работы обучающихся и промежуточной аттестации.

При проведении учебных занятий по дисциплине обеспечивается развитие у обучающихся навыков командной работы, межличностной коммуникации, принятия решений, лидерских качеств (включая проведение групповых практических работ).

Преподавание дисциплины ведется с применением следующих видов **образовательных технологий:**

- балльно-рейтинговая технология оценивания;
- электронное обучение;

Для оценки знаний, умений, навыков и уровня сформированности компетенции по дисциплине применяется балльно-рейтинговая система контроля и оценки успеваемости студентов. В основу балльно-рейтинговой системы положены принципы, в соответствии с которыми формирование рейтинга студента осуществляется в ходе текущего контроля успеваемости. Максимальное количество баллов в семестре – 100.

4.2. Методические указания для обучающихся по освоению дисциплины на занятиях лекционного типа

Лекционный курс предполагает систематизированное изложение основных вопросов тематического плана. В ходе лекционных занятий раскрываются базовые вопросы в рамках каждой темы дисциплины. Обозначаются ключевые аспекты тем, а также делаются акценты на наиболее сложные и важные положения изучаемого материала.

Лекционные занятия проводятся в поточной аудитории с применением мультимедийного проектора в виде учебной презентации или в ЭИОС университета.

В ходе лекционных занятий рекомендуется конспектирование учебного материала. Возможно ведение конспекта лекций в виде интеллект-карт.

Отдельные темы предлагаются для самостоятельного изучения (конспектируются).

Материалы лекций являются опорной основой для подготовки обучающихся к практическим занятиям / лабораторным работам и выполнения заданий самостоятельной работы, а также к мероприятиям текущего контроля успеваемости и промежуточной аттестации по дисциплине.

4.3. Методические указания для обучающихся по освоению дисциплины на занятиях семинарского типа/ на практических занятиях

Практические (семинарские) занятия представляют собой детализацию лекционного теоретического материала, проводятся в целях закрепления курса и охватывают все основные разделы. Основной формой проведения практических занятий является решение задач по проектированию и программированию беспилотных систем.

Практические занятия обучающихся обеспечивают:

- проверку и уточнение знаний, полученных на лекциях;
- получение умений и навыков проектирования, моделирование, организацию сетей. Поиск и устранение неисправностей;
- подведение итогов занятий по рейтинговой системе, согласно технологической карте дисциплины.

4.4. Методические указания по самостоятельной работе обучающихся

Самостоятельная работа обеспечивает подготовку обучающегося к аудиторным занятиям и мероприятиям текущего контроля и промежуточной аттестации по изучаемой дисциплине. Результаты этой подготовки проявляются в активности обучающегося на занятиях и в качестве выполненных практических заданий и других форм текущего контроля.

Самостоятельная работа студентов включает:

1. Изучение учебной литературы по курсу.
2. Работу с ресурсами Интернет
3. Самостоятельное изучение учебных материалов

При выполнении заданий для самостоятельной работы рекомендуется проработка материалов лекций по каждой пройденной теме, а также изучение рекомендуемой литературы.

Для обучающихся по заочной форме обучения самостоятельная работа является основным видом учебной деятельности.

Для обеспечения самостоятельной работы обучающихся используется электронный учебный курс, созданный в ЭИОС университета http://sdo.tolgas.ru/.

5. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

5.1. Перечень основной и дополнительной учебной литературы, необходимой для освоения дисциплины

Основная литература:

1. **Беспилотные летательные аппараты** : учеб. пособие / С. Н. Денисенко, А. Ю. Смирнов, А. М. Хрусталев, И. Г. Штеренберг ; Санкт-Петербург. гос. технол. ун-т. - Документ read. - Санкт-Петербург : СПбГТИ (ТУ), 2023. - 115 с. - URL: https://reader.lanbook.com/book/365894 (дата обращения: 22.04.2025). - Режим доступа: для авториз. пользователей. - Текст : электронный. URL: https://reader.lanbook.com/book/365894

2. **Фетисов, В. С.** Беспилотные авиационные системы: терминология, классификация, структура : учеб. пособие для вузов / В. С. Фетисов, Л. М. Неугодникова. - Изд. 2-е, стер. - Документ read. - Санкт-Петербург [и др.] : Лань, 2025. - 132 с. - URL: https://reader.lanbook.com/book/441680 (дата обращения: 22.04.2025). - Режим доступа: для авториз. пользователей. - ISBN 978-5-507-50513-5. - Текст : электронный. URL: https://reader.lanbook.com/book/441680.

Дополнительная литература:

3. **Макаренко, С.И.** Противодействие беспилотным летательным аппаратам : монография / С. И. Макаренко. - Документ read. - Санк-Петербург : Наукоемкие технологии, 2020. - 206 с. - URL: https://znanium.ru/read?id=475970 (дата обращения: 11.06.2026). - Режим доступа: для авториз. пользователей. - ISBN 978-5-6044793-6-0. - Текст : электронный. URL: https://znanium.ru/read?id=4759706.

4. Стейпл, Д. Устройство и программирование автономных роботов. Проекты на Python и Raspberry Pi : практическое руководство / Д. Стейпл ; пер. с англ. Е. В. Шевчук ; науч. ред. В. С. Яценков. - Москва : ДМК Пресс, 2022. - 522 с. - ISBN 978-5-97060-989-7. - Текст : электронный. - URL: https://znanium.ru/catalog/product/2109590 (дата обращения: 17.09.2026). – Режим доступа: по подписке.

5.2. Профессиональные базы данных, информационно-справочные системы, интернет-ресурсы

1. eLIBRARY.RU : научная электронная библиотека : сайт. – Москва, 2000 - . - URL: https://elibrary.ru (дата обращения: 09.09.2024). – Режим доступа: для зарегистрир. пользователей. – Текст: электронный.
2. Электронная библиотечная система Поволжского государственного университета сервиса : сайт / ФГБОУ ВО «ПВГУС». – Тольятти, 2010 - . - URL. : http://elib.tolgas.ru (дата обращения: 09.09.2024). - Режим доступа: для авториз. пользователей. - Текст : электронный.
3. Электронно-библиотечная система Znanium.com : сайт / ООО "ЗНАНИУМ". – Москва, 2011 - . - URL: https://znanium.com/ (дата обращения: 09.09.2024). - Режим доступа: для авториз. пользователей. - Текст : электронный.
4. Электронно-библиотечная система Лань : сайт / ООО "ЭБС ЛАНЬ". - Москва, 2011 - . - URL: https://e.lanbook.com/ (дата обращения: 09.09.2024). - Режим доступа: для авториз. пользователей. - Текст : электронный.

5.3. Программное обеспечение

Информационное обеспечение учебного процесса по дисциплине осуществляется с использованием следующего программного обеспечения (лицензионного и свободно распространяемого), в том числе отечественного производства:

№ п/п	Наименование	Условия доступа
1.	Microsoft Windows	из внутренней сети университета (лицензионный договор)
2.	Microsoft Office	из внутренней сети университета (лицензионный договор)
3.	СДО MOODLE	из любой точки, в которой имеется доступ к сети Интернет (лицензионный договор)
4.	Браузер	из любой точки, в которой имеется доступ к сети Интернет (свободно распространяемое)
5.	Webots	из любой точки, в которой имеется доступ к сети Интернет (свободно распространяемое
6.	GitHub	из любой точки, в которой имеется доступ к сети Интернет (свободно распространяемое

6. ОПИСАНИЕ МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЙ БАЗЫ, НЕОБХОДИМОЙ ДЛЯ ОСУЩЕСТВЛЕНИЯ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА ПО ДИСЦИПЛИНЕ

Помещения представляют собой учебные аудитории для проведения учебных занятий, предусмотренных учебным планом и рабочей программой дисциплины, оснащённые оборудованием и техническими средствами обучения.

Занятия лекционного типа. Учебные аудитории для занятий лекционного типа укомплектованы мебелью и техническими средствами обучения, служащими для представления учебной информации (стационарные или переносные наборы демонстрационного оборудования (проектор, экран, компьютер/ноутбук), учебно-наглядные пособия (презентации по темам лекций), обеспечивающие тематические иллюстрации, соответствующие данной программе дисциплины.

Занятия семинарского типа. Учебные аудитории для занятий семинарского типа укомплектованы мебелью и техническими средствами обучения, служащими для представления учебной информации (стационарные или переносные наборы демонстрационного оборудования (проектор, экран, компьютер/ноутбук).

Промежуточная аттестация. Для проведения промежуточной аттестации по дисциплине используются компьютерные классы, оснащённые компьютерной техникой с возможностью подключения к сети Интернет и обеспечением доступа в электронную информационно-образовательную среду университета и/или учебные аудитории, укомплектованные мебелью и техническими средствами обучения.

Самостоятельная работа. Помещения для самостоятельной работы оснащены компьютерной техникой с возможностью подключения к сети «Интернет» и доступом к электронной информационно-образовательной среде университета. Для организации самостоятельной работы обучающихся используются:

- компьютерные классы университета;
- библиотека (медиазал), имеющая места для обучающихся, оснащенные компьютерами с доступом к базам данных и сети Интернет.

Электронная информационно-образовательная среда университета (ЭИОС). Каждый обучающийся в течение всего периода обучения обеспечен индивидуальным неограниченным доступом к электронной информационно-образовательной среде университета (ЭИОС) http://sdo.tolgas.ru/ из любой точки, в которой имеется доступ к информационно-телекоммуникационной сети "Интернет", как на территории университета, так и вне ее.

ЭИОС университета обеспечивает:

– доступ к учебным планам, рабочим программам дисциплин (модулей), программам практик, электронным учебным изданиям и электронным образовательным ресурсам, указанным в рабочих программах дисциплин (модулей), программах практик;

– формирование электронного портфолио обучающегося, в том числе сохранение его работ и оценок за эти работы.

В случае реализации образовательной программы с применением электронного обучения, дистанционных образовательных технологий ЭИОС дополнительно обеспечивает:

– фиксацию хода образовательного процесса, результатов промежуточной аттестации и результатов освоения образовательной программы;

– проведение учебных занятий, процедур оценки результатов обучения, реализация которых предусмотрена с применением электронного обучения, дистанционных образовательных технологий;

– взаимодействие между участниками образовательного процесса, в том числе синхронное и (или) асинхронное взаимодействия посредством сети "Интернет".

7. ОСОБЕННОСТИ ОРГАНИЗАЦИИ ОБУЧЕНИЯ ДЛЯ ЛИЦ С ОГРАНИЧЕННЫМИ ВОЗМОЖНОСТЯМИ ЗДОРОВЬЯ И ИНВАЛИДОВ

При необходимости рабочая программа дисциплины может быть адаптирована для обеспечения образовательного процесса инвалидов и лиц с ограниченными возможностями здоровья, в том числе для дистанционного обучения. Для этого требуется заявление студента (его законного представителя) и заключение психолого-медико-педагогической комиссии (ПМПК).

В случае необходимости, обучающимся из числа лиц с ограниченными возможностями здоровья (по заявлению обучающегося) а для инвалидов также в соответствии с индивидуальной программой реабилитации инвалида, могут предлагаться следующие варианты восприятия учебной информации с учетом их индивидуальных психофизических особенностей, в том числе с применением электронного обучения и дистанционных технологий:

- для лиц с нарушениями зрения: в печатной форме увеличенным шрифтом; в форме электронного документа; в форме аудиофайла (перевод учебных материалов в аудиоформат); в печатной форме на языке Брайля; индивидуальные консультации с привлечением тифлосурдопереводчика; индивидуальные задания и консультации.
- для лиц с нарушениями слуха: в печатной форме; в форме электронного документа; видеоматериалы с субтитрами; индивидуальные консультации с привлечением сурдопереводчика; индивидуальные задания и консультации.
- для лиц с нарушениями опорно-двигательного аппарата: в печатной форме; в форме электронного документа; в форме аудиофайла; индивидуальные задания и консультации.

8. ОЦЕНОЧНЫЕ МАТЕРИАЛЫ (ФОНД ОЦЕНОЧНЫХ СРЕДСТВ) ДЛЯ ТЕКУЩЕГО КОНТРОЛЯ УСПЕВАЕМОСТИ И ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ

8.1. Описание показателей и критериев оценивания компетенций на различных этапах их формирования, описание шкал оценивания

Для оценки знаний, умений, навыков и уровня сформированности компетенции по дисциплине применяется балльно-рейтинговая система контроля и оценки успеваемости студентов. В основу балльно-рейтинговой системы положены принципы, в соответствии с которыми формирование рейтинга студента осуществляется в ходе текущего контроля успеваемости. Максимальное количество баллов в семестре – 100.

Шкала оценки результатов освоения дисциплины, сформированности результатов обучения

Форма проведения промежуточной аттестации	**Шкалы оценки уровня сформированности результатов обучения**		**Шкала оценки уровня освоения дисциплины**		
	Уровневая шкала оценки компетенций	100 бальная шкала, %	100 бальная шкала, %	5-балльная шкала, дифференцированная оценка/балл	недифференцированная оценка
Зачёт	допороговый	ниже 61	ниже 61	«неудовлетворительно» / 2	не зачтено
	пороговый	61-85,9	61-69,9	«удовлетворительно» / 3	зачтено
			70-85,9	«хорошо» / 4	зачтено
	повышенный	86-100	86-100	«отлично» / 5	зачтено

По итогам текущей успеваемости студенту может быть выставлена оценка по промежуточной аттестации в соответствии за набранными за семестр баллами (по накопительному рейтингу). Студентам, набравшим в ходе текущего контроля успеваемости по дисциплине от 61 до 100 баллов и выполнившим все обязательные виды запланированных учебных занятий, по решению преподавателя без прохождения промежуточной аттестации выставляется оценка в соответствии со шкалой оценки результатов освоения дисциплины.

Результат обучения считается сформированным (повышенный уровень), если теоретическое содержание курса освоено полностью; при устных собеседованиях студент исчерпывающе, последовательно, четко и логически стройно излагает учебный материал; свободно справляется с задачами, вопросами и другими видами заданий, требующих применения знаний, использует в ответе дополнительный материал; все предусмотренные рабочей учебной программой задания выполнены в соответствии с установленными требованиями, студент способен анализировать полученные результаты; проявляет самостоятельность при выполнении заданий, качество их выполнения оценено числом баллов от 86 до 100, что соответствует повышенному уровню сформированности результатов обучения.

Результат обучения считается сформированным (пороговый уровень), если теоретическое содержание курса освоено полностью; при устных собеседованиях студент последовательно, четко и логически стройно излагает учебный материал; справляется с задачами, вопросами и другими видами заданий, требующих применения знаний; все предусмотренные рабочей учебной программой задания выполнены в соответствии с установленными требованиями, студент способен анализировать полученные результаты; проявляет самостоятельность при выполнении заданий, качество их выполнения оценено числом баллов от 61 до 85,9, что соответствует пороговому уровню сформированности результатов обучения.

Результат обучения считается несформированным, если студент при выполнении заданий не демонстрирует знаний учебного материала, допускает ошибки, неуверенно, с большими затруднениями выполняет задания, не демонстрирует необходимых умений, качество выполнен-

ных заданий не соответствует установленным требованиям, качество их выполнения оценено числом баллов ниже 61, что соответствует допороговому уровню.

Формы текущего контроля успеваемости

Формы текущего контроля	Количество контрольных точек	Количество баллов за 1 контр. точку	Макс. возм. кол-во баллов
Отчёт по практической работе	9	5	40
Тестирование по темам лекционных занятий	6	3	18
Творческий рейтинг (участие в конференциях, олимпиадах, проектной деятельности и т.п.)	1	32	32
Итого по дисциплине			**100 баллов**

Система оценивания представлена в электронном учебном курсе по дисциплине http://sdo.tolgas.ru/.

8.2. Типовые контрольные задания или иные материалы для ТЕКУЩЕГО КОНТРОЛЯ УСПЕВАЕМОСТИ

8.2.1. Типовые задания к практическим (семинарским) занятиям

Практическая работа 1. Первое знакомство со средой Webots и базовое управление дроном
Цель работы: освоить интерфейс симулятора Webots, изучить базовые принципы работы с виртуальной средой и получить навыки ручного управления беспилотным летательным аппаратом.
Задачи:

1. Изучить структуру интерфейса Webots (окно 3D-визуализации, дерево объектов, панель управления).
2. Загрузить готовую сцену с моделью дрона Crazyflie или Mavic 2 Pro.
3. Освоить навигацию в 3D-пространстве: вращение камеры, масштабирование, перемещение.
4. Изучить управление симуляцией: запуск, пауза, сброс, изменение скорости.
5. Выполнить ручное управление дроном с клавиатуры: взлёт, перемещение по горизонтали, изменение высоты, поворот.

Ход выполнения работы:

1. **Запуск Webots и загрузка сцены.** Запустить Webots. Через меню File → Open Sample World перейти в раздел robots → bitcraze → crazyflie и открыть файл crazyflie.wbt. При использовании модели Mavic 2 Pro выбрать robots → dji → mavic → mavic_2_pro.wbt.
2. **Изучение интерфейса.** Ознакомиться с основными элементами интерфейса:
   - **Дерево объектов (Scene Tree)** — иерархия всех объектов сцены, включая робота, датчики, препятствия.
   - **Панель управления** — кнопки запуска, паузы, сброса симуляции, регулировка скорости.
   - **Окно 3D-визуализации** — основное окно, где отображается сцена.
   - **Редактор контроллера** — область для просмотра и редактирования кода.
3. **Освоение навигации в 3D-пространстве.**
   - Вращение камеры: зажать левую кнопку мыши и перемещать курсор.
   - Перемещение камеры: зажать правую кнопку мыши.
   - Масштабирование: прокрутка колёсика мыши.
   - Выбор объекта: клик левой кнопкой мыши по объекту в сцене или в дереве объектов.
4. **Запуск симуляции.** Нажать кнопку «Play» на панели управления. Дрон автоматически взлетит на небольшую высоту. Кликнуть по окну 3D-визуализации, чтобы активировать управление с клавиатуры.
5. **Ручное управление дроном.** Использовать следующие клавиши для управления:

- **Стрелки вверх/вниз/влево/вправо** — перемещение дрона в горизонтальной плоскости (вперёд/назад/влево/вправо).
- **W/S** — увеличение/уменьшение высоты полёта.
- **Q/E** — поворот вокруг вертикальной оси (рыскание).

Выполнить несколько тренировочных полётов: взлететь, переместиться в заданную точку сцены, облететь препятствие, вернуться в исходную точку и совершить посадку.

6. **Изучение свойств объектов.** Выбрать дрон в дереве объектов. В панели свойств ознакомиться с основными параметрами: translation (координаты), rotation (ориентация), controller (имя контроллера). Попробовать изменить координаты дрона вручную и наблюдать результат.
7. **Сохранение и сброс.** Сохранить сцену через File → Save World As. Изучить работу кнопки сброса симуляции — она возвращает всех роботов в исходное состояние.
8. **Оформление отчёта.** Подготовить отчёт, содержащий: скриншот загруженной сцены, описание выполненных манёвров, скриншот траектории полёта (если доступно) и краткие выводы о работе с интерфейсом Webots.

Практическая работа 2. Программирование контроллера — автоматический взлёт и движение по прямой

Цель работы: освоить написание внешних контроллеров на языке Python в среде Webots для реализации автоматического управления дроном.

Задачи:

1. Изучить структуру контроллера в Webots и механизм его подключения к роботу.
2. Освоить базовые функции Webots Python API для управления моторами (Motor).
3. Написать контроллер, реализующий автоматическую последовательность: взлёт → движение вперёд → посадка.
4. Научиться использовать robot.step() для синхронизации шагов симуляции.

Ход выполнения работы:

1. **Изучение структуры контроллера.** В Webots каждый робот имеет поле controller в дереве объектов, указывающее на имя папки с контроллером. Контроллер — это программа, управляющая роботом в цикле симуляции. Для дрона DJI Mavic 2 Pro в примере mavic_world.wbt уже может быть назначен контроллер по умолчанию.
2. **Создание собственного контроллера.** В директории проекта (controllers/) создать папку с именем mavic_auto_controller. Внутри создать файл mavic_auto_controller.py. В поле controller робота в дереве объектов указать это имя.
3. **Изучение Python API.** Для управления дроном необходимо получить доступ к устройствам через getDevice(). Для квадрокоптера Mavic 2 Pro основные моторы могут иметь имена, соответствующие их расположению. В простейшем случае для автоматического взлёта и полёта можно управлять моторами напрямую через установку скоростей.
4. **Написание базового контроллера.** Примерная структура кода на Python:

```
from controller import Robot

robot = Robot()
timestep = int(robot.getBasicTimeStep())

# Получение устройств
motors = []
motor_names = ["front_left_motor", "front_right_motor", "rear_left_motor", "rear_right_motor"]
for name in motor_names:
    motors.append(robot.getDevice(name))

# Инициализация
for motor in motors:
    motor.setPosition(float('inf'))
    motor.setVelocity(0.0)
```

```
# Параметры
takeoff_speed = 1.0
forward_speed = 0.5
duration_takeoff = 2.0
duration_forward = 3.0

# Взлёт
for _ in range(int(duration_takeoff * 1000 / timestep)):
    for motor in motors:
        motor.setVelocity(takeoff_speed)
    robot.step(timestep)

# Движение вперёд (пример: увеличение скорости задних моторов)
for _ in range(int(duration_forward * 1000 / timestep)):
    motors[0].setVelocity(forward_speed)
    motors[1].setVelocity(forward_speed)
    motors[2].setVelocity(forward_speed * 1.2)
    motors[3].setVelocity(forward_speed * 1.2)
    robot.step(timestep)

# Посадка (постепенное снижение)
for _ in range(int(duration_takeoff * 1000 / timestep)):
    for motor in motors:
        motor.setVelocity(-takeoff_speed * 0.5)
    robot.step(timestep)

# Остановка
for motor in motors:
    motor.setVelocity(0.0)
```

Важно: Имена моторов зависят от конкретной модели дрона в Webots. В готовых примерах (например, для Crazyflie) используются стандартные имена из прототипа робота. Для Mavic 2 Pro необходимо проверить имена устройств в дереве объектов или документации соответствующего прототипа.

5. **Тестирование и отладка.** Запустить симуляцию. Наблюдать поведение дрона. Если дрон не взлетает, проверить имена моторов и направление вращения. Использовать robot.step(timestep) для продвижения симуляции на каждом шаге.
6. **Оформление отчёта.** Включить листинг контроллера, описание логики работы и результаты наблюдений (скриншоты или описание поведения дрона).

Практическая работа 3. Реализация PID-контроллера высоты и позиции

Цель работы: реализовать PID-регулятор для стабилизации дрона на заданной высоте и удержания позиции.

Задачи:

1. Изучить принцип работы ПИД-регулятора (пропорциональная, интегральная, дифференциальная составляющие).
2. Научиться считывать данные с датчиков высоты (GPS, InertialUnit) в Webots.
3. Реализовать контур обратной связи для управления вертикальной скоростью.
4. Настроить коэффициенты PID для достижения стабильного зависания.

Ход выполнения работы:

1. **Изучение датчиков.** В Webots дрон Mavic 2 Pro имеет устройства GPS (для определения координат, включая высоту) и InertialUnit (для определения углов ориентации roll, pitch,

yaw). Для получения высоты используется GPS: gps.getValues()[2] возвращает координату Z.

2. **Изучение принципа PID.** PID-регулятор вычисляет управляющее воздействие как сумму трёх компонент:
   - **P (пропорциональная):** Kp * error, где error = target_altitude - current_altitude
   - **I (интегральная):** Ki * integral_of_error
   - **D (дифференциальная):** Kd * derivative_of_error
3. **Написание контроллера с PID.** Примерная структура:

```
python
from controller import Robot

robot = Robot()
timestep = int(robot.getBasicTimeStep())

gps = robot.getDevice("gps")
gps.enable(timestep)

motors = []
motor_names = ["front_left_motor", "front_right_motor", "rear_left_motor", "rear_right_motor"]
for name in motor_names:
    motors.append(robot.getDevice(name))

for motor in motors:
    motor.setPosition(float('inf'))

# PID параметры
Kp = 2.0
Ki = 0.1
Kd = 0.5
target_altitude = 2.0

integral = 0.0
previous_error = 0.0

while robot.step(timestep) != -1:
    current_altitude = gps.getValues()[2]
    error = target_altitude - current_altitude

    integral += error * (timestep / 1000.0)
    derivative = (error - previous_error) / (timestep / 1000.0)

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

    # Применение управления к моторам
    for motor in motors:
        motor.setVelocity(control)

    previous_error = error
```

4. **Настройка коэффициентов.** Начать с Kp только (установить Ki=0, Kd=0). Увеличивать Kp до появления колебаний. Затем добавить Kd для демпфирования. При необходимости добавить небольшой Ki для устранения статической ошибки.
5. **Расширение на позицию.** Для удержания позиции по X и Y использовать аналогичные PID-контуры, где ошибка вычисляется как разность между целевыми и текущими коорди-

натами из GPS. Выход PID по X и Y преобразуется в дифференциальные скорости моторов (например, наклон дрона в соответствующую сторону).

6. **Тестирование.** Запустить симуляцию. Дрон должен стабилизироваться на заданной высоте (например, 2 м). Изменяя целевую высоту, проверить реакцию системы. Записать переходный процесс (время выхода на режим, перерегулирование).
7. **Оформление отчёта.** Включить листинг контроллера, таблицу с настроенными коэффициентами PID, график переходного процесса (если использовалась запись данных) и выводы о качестве регулирования.

Практическая работа 4. Чтение и обработка данных датчиков — GPS и IMU

Цель работы: освоить чтение данных с датчиков GPS и InertialUnit в Webots, научиться сохранять и обрабатывать измерительную информацию.

Задачи:

1. Научиться получать доступ к устройствам GPS и InertialUnit через Webots API.
2. Освоить чтение значений координат и углов ориентации.
3. Записывать данные в CSV-файл для последующего анализа.
4. Выполнить простую обработку: вычисление скорости по разности GPS-координат.

Ход выполнения работы:

1. **Получение доступа к устройствам.** В контроллере дрона получить устройства:

```
python
gps = robot.getDevice("gps")
gps.enable(timestep)
inertial_unit = robot.getDevice("inertial_unit")
inertial_unit.enable(timestep)
```

Датчик GPS в Webots возвращает координаты в метрах (X, Y, Z) относительно начала координат мира. В некоторых конфигурациях с ROS GPS может публиковать данные в формате NavSatFix, но в чистом Python API Webots используются декартовы координаты.

2. **Чтение данных.** На каждом шаге симуляции:

```
python
gps_values = gps.getValues()  # [x, y, z]
roll, pitch, yaw = inertial_unit.getRollPitchYaw()
```

3. **Запись в CSV.** Использовать модуль csv для сохранения данных:

```
python
import csv

with open('sensor_data.csv', 'w', newline='') as f:
   writer = csv.writer(f)
   writer.writerow(['time', 'x', 'y', 'z', 'roll', 'pitch', 'yaw'])

   while robot.step(timestep) != -1:
      x, y, z = gps.getValues()
      r, p, yw = inertial_unit.getRollPitchYaw()
      t = robot.getTime()
      writer.writerow([t, x, y, z, r, p, yw])
```

4. **Вычисление скорости.** По разности координат GPS между последовательными шагами вычислить линейную скорость:

```
python
dt = timestep / 1000.0
vx = (x - prev_x) / dt
vy = (y - prev_y) / dt
vz = (z - prev_z) / dt
```

Сравнить полученные значения с оценками, которые можно получить другими способами (например, численным дифференцированием данных IMU, если доступны акселерометры).

5. **Визуализация.** Использовать matplotlib для построения графиков траектории (X-Y), высоты от времени и углов ориентации. Сохранить графики в файлы для отчёта.
6. **Анализ данных.** Проверить, соответствуют ли данные физическому поведению дрона: при зависании высота должна быть постоянной, углы — близкими к нулю. При движении вперёд координата X или Y должна линейно изменяться.
7. **Оформление отчёта.** Включить листинг контроллера, CSV-файл (или его фрагмент), графики и краткий анализ полученных данных.

Практическая работа 5. Обнаружение препятствий с помощью дальномеров
Цель работы: освоить использование дальномеров в Webots для обнаружения препятствий и реализовать простое поведение объезда.
Задачи:

1. Изучить устройство DistanceSensor в Webots и его API.
2. Научиться добавлять дальномеры к модели дрона (или использовать встроенные).
3. Реализовать логику обнаружения препятствия по пороговому значению.
4. Запрограммировать простое поведение: при обнаружении препятствия — остановка и поворот.

Ход выполнения работы:

1. **Изучение DistanceSensor.** В Webots дальномеры возвращают значения, пропорциональные расстоянию до объекта. Для инфракрасных датчиков (как на E-puck) значение увеличивается при приближении к препятствию. Для дрона можно использовать встроенные датчики расстояния или добавить их в прототип.
2. **Получение устройств.** Если датчики добавлены в модель, получить их по именам:

```
python
sensors = []
sensor_names = ["ds_front", "ds_left", "ds_right"]
for name in sensor_names:
   sensors.append(robot.getDevice(name))
   sensors[-1].enable(timestep)
```

3. **Чтение значений и обнаружение препятствий.** На каждом шаге считывать значения и сравнивать с порогом:

```
python
threshold = 80.0  # значение зависит от типа датчика и настроек

values = [s.getValue() for s in sensors]
obstacle_detected = any(v > threshold for v in values)
```

4. **Реализация поведения объезда.** Простейшая логика: если препятствие обнаружено спереди, дрон должен приостановить движение вперёд и повернуть (например, вправо, если препятствие слева, и наоборот). Пример для дифференциального управления моторами:

```
python
if obstacle_detected:
   # Поворот: левые моторы быстрее правых
   for motor in motors:
      motor.setVelocity(0.0)
   # Выполнить поворот в течение заданного времени
   # ...
```

5. **Тестирование в сцене с препятствиями.** Загрузить мир с препятствиями (например, mavic_world.wbt содержит сельскую местность с деревьями и зданиями). Разместить дрон так, чтобы на его пути были объекты. Запустить симуляцию и наблюдать поведение. При необходимости скорректировать порог срабатывания и скорость поворота.
6. **Расширение.** Можно реализовать более сложное поведение: не просто поворот, а выбор направления объезда на основе комбинации показаний нескольких датчиков (левый, правый, передний). Это позволит дрону огибать препятствия, а не просто останавливаться.

7. **Оформление отчёта.** Включить листинг контроллера, описание логики обнаружения и объезда, скриншоты или описание поведения дрона в сцене с препятствиями.

Практическая работа 6. Компьютерное зрение — обнаружение целевых объектов через камеру

Цель работы: освоить чтение изображений с камеры в Webots и реализовать простое обнаружение объектов по цвету с помощью OpenCV.

Задачи:

1. Научиться получать изображения с устройства Camera в Webots.
2. Освоить преобразование изображения в массив NumPy для обработки OpenCV.
3. Реализовать цветовую сегментацию для выделения объекта заданного цвета.
4. Управлять дроном для наведения на обнаруженный объект.

Ход выполнения работы:

1. **Получение доступа к камере.** Дрон DJI Mavic 2 Pro в Webots оснащён камерой. Получить устройство:

python
```
camera = robot.getDevice("camera")
camera.enable(timestep)
```

2. **Чтение изображения.** Webots Camera API возвращает изображение в формате BGRA. Для получения массива:

python
```
import numpy as np
import cv2

width = camera.getWidth()
height = camera.getHeight()

image = camera.getImage()  # bytes
# Преобразование в numpy array
img_array = np.frombuffer(image, dtype=np.uint8).reshape((height, width, 4))
bgr_image = img_array[:, :, :3]  # убрать альфа-канал
```

3. **Цветовая сегментация.** Преобразовать изображение в цветовое пространство HSV (лучше для выделения по цвету) и задать диапазоны:

python
```
hsv = cv2.cvtColor(bgr_image, cv2.COLOR_BGR2HSV)
# Например, для красного цвета (два диапазона из-за круговой шкалы Hue)
lower_red1 = np.array([0, 100, 100])
upper_red1 = np.array([10, 255, 255])
lower_red2 = np.array([170, 100, 100])
upper_red2 = np.array([180, 255, 255])

mask1 = cv2.inRange(hsv, lower_red1, upper_red1)
mask2 = cv2.inRange(hsv, lower_red2, upper_red2)
mask = mask1 + mask2
```

4. **Поиск контуров и определение положения объекта.** Найти контуры на маске, выбрать наибольший, вычислить его центр:

python
```
contours, _ = cv2.findContours(mask, cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)
if contours:
    largest = max(contours, key=cv2.contourArea)
    M = cv2.moments(largest)
    if M["m00"] > 0:
        cx = int(M["m10"] / M["m00"])
        cy = int(M["m01"] / M["m00"])
```

```
    # cx, cy — координаты центра объекта в изображении
```

5. **Управление дроном для наведения на цель.** Использовать положение центра объекта относительно центра изображения как ошибку для управления:

```
python
center_x = width // 2
error_x = cx - center_x

# Простейшее пропорциональное управление поворотом
turn_speed = -error_x * 0.01  # знак зависит от ориентации камеры
```

Можно комбинировать с движением вперёд, если объект обнаружен, для приближения к нему.

6. **Тестирование.** Разместить в сцене объект красного (или другого заданного) цвета. Запустить симуляцию. Дрон должен обнаружить объект и начать поворачиваться в его сторону (или двигаться к нему). Аналогичный подход используется в проектах по обнаружению фруктов с помощью OpenCV и Webots.
7. **Оформление отчёта.** Включить листинг контроллера, описание алгоритма цветовой сегментации, скриншоты с выделенным объектом (можно выводить маску на экран) и результаты управления.

Практическая работа 7. Автономная навигация по пути — алгоритм A*

Цель работы: реализовать автономную навигацию дрона по заданному маршруту с использованием алгоритма A* для планирования пути.

Задачи:

1. Изучить принцип работы алгоритма A* для поиска пути на графе.
2. Построить сеточную карту среды на основе данных о препятствиях.
3. Реализовать алгоритм A* для поиска оптимального пути из начальной точки в целевую.
4. Запрограммировать следование по найденному пути с использованием PID-контроллера.

Ход выполнения работы:

1. **Изучение A*.** A* — алгоритм поиска пути на графе, использующий эвристическую функцию для оценки стоимости пути до цели. Он широко применяется в проектах навигации дронов в Webots. Ключевые компоненты: открытый список (open set), закрытый список (closed set), стоимость пути g(n), эвристика h(n), общая оценка f(n) = g(n) + h(n).
2. **Построение карты.** Определить размеры рабочей области и разрешение сетки. Например, область 20×20 метров с шагом 1 метр даёт сетку 20×20 ячеек. Для каждой ячейки определить, является ли она проходимой или содержит препятствие. Данные о препятствиях можно получить из известной карты сцены или с помощью датчиков.
3. **Реализация A*.** Написать функцию поиска пути:

```
python
def astar(start, goal, grid):
    open_set = {start}
    came_from = {}
    g_score = {start: 0}
    f_score = {start: heuristic(start, goal)}

    while open_set:
        current = min(open_set, key=lambda x: f_score.get(x, float('inf')))
        if current == goal:
            return reconstruct_path(came_from, current)

        open_set.remove(current)
        for neighbor in get_neighbors(current, grid):
            tentative_g = g_score[current] + 1
            if tentative_g < g_score.get(neighbor, float('inf')):
                came_from[neighbor] = current
                g_score[neighbor] = tentative_g
```

```
            f_score[neighbor] = tentative_g + heuristic(neighbor, goal)
            open_set.add(neighbor)
    return None
```

4. **Следование по пути.** Найденный путь представляет собой последовательность узлов сетки (координат). Для следования по пути использовать PID-контроллер позиции (из занятия 3). На каждом шаге текущая целевая точка — следующая точка пути. Когда дрон достигает текущей точки (расстояние меньше порога), переключиться на следующую:

```
python
waypoints = astar_path  # список координат
current_waypoint_index = 0

while robot.step(timestep) != -1:
    if current_waypoint_index >= len(waypoints):
        break  # путь пройден

    target = waypoints[current_waypoint_index]
    current_pos = gps.getValues()

    distance = math.sqrt((target[0] - current_pos[0])**2 + (target[1] - current_pos[1])**2)
    if distance < 0.5:
        current_waypoint_index += 1
        continue

    # PID для движения к target
    # ...
```

5. **Обработка динамических препятствий.** Если в процессе движения обнаружено новое препятствие (с помощью дальномеров), необходимо пометить соответствующий узел как заблокированный и перезапустить A* для поиска нового пути.
6. **Тестирование.** Задать начальную и целевую точки в сцене с препятствиями (например, городская среда из mavic_world.wbt). Запустить симуляцию. Дрон должен построить путь и проследовать по нему, избегая препятствия.
7. **Оформление отчёта.** Включить листинг алгоритма A*, описание структуры данных для карты, визуализацию найденного пути (скриншот сцены с отмеченным маршрутом) и результаты прохождения.

Практическое занятие 8. Мультидрон - симуляция двух дронов и базовая синергия (взаимодействие как целое)

Цель работы: освоить создание сцены с несколькими дронами в Webots и реализовать простое единое поведение

Задачи:

1. Научиться добавлять несколько экземпляров дрона в одну сцену Webots.
2. Освоить назначение разных контроллеров для разных экземпляров роботов.
3. Реализовать простое 协同: последовательный взлёт с задержкой или удержание дистанции.
4. Изучить механизмы идентификации роботов для раздельного управления.

Ход выполнения работы:

1. **Добавление второго дрона.** В дереве объектов выбрать корневой узел WorldInfo или Robot. Через контекстное меню (Add New) добавить ещё один экземпляр прототипа Mavic 2 Pro (или Crazyflie). Разместить его рядом с первым, задав координаты translation в дереве объектов.
2. **Назначение контроллеров.** Каждый экземпляр робота в Webots имеет собственное поле controller. Создать два контроллера: drone_a_controller.py и drone_b_controller.py. Назначить их соответствующим дронам. Альтернативный подход — использовать один контроллер, который различает роботов по имени (robot.getName()).

3. **Реализация последовательного взлёта.** В контроллере дрона A реализовать взлёт немедленно. В контроллере дрона B добавить задержку перед взлётом:

```
python
# drone_b_controller.py
# Ожидание 2 секунды, затем взлёт
start_time = robot.getTime()
while robot.step(timestep) != -1:
   if robot.getTime() - start_time > 2.0:
      break
# Затем взлёт
```

4. **Удержание дистанции (расширение).** Для удержания заданного расстояния между дронами дрон B может использовать данные GPS дрона A. Однако в Webots контроллеры разных роботов изолированы. Для обмена данными можно:
   - Использовать Supervisor для доступа к позиции других роботов из контроллера.
   - Использовать внешний контроллер или ROS 2 для связи между дронами.

Простейший вариант: дрон B следует за дроном A, используя GPS дрона A, полученный через Supervisor API:

```
python
supervisor = robot.getSupervisor()
drone_a_node = supervisor.getFromDef("DRONE_A")
if drone_a_node:
   a_position = drone_a_node.getField("translation").getSFVec3f()
```

5. **Тестирование.** Запустить симуляцию. Убедиться, что оба дрона появляются в сцене. Дрон A взлетает первым, дрон B — с задержкой. При реализации удержания дистанции проверить, что расстояние между дронами остаётся близким к заданному.
6. **Изучение мультиагентных примеров.** Существующий проект patrickpbarroso/drone-simulation демонстрирует мультидронную симуляцию в Webots с ROS 2, где контроллеры запускаются для каждого дрона отдельно с указанием ID. Этот подход можно изучить для более сложных сценариев.
7. **Оформление отчёта.** Включить листинги контроллеров, описание архитектуры мультиагентной системы, скриншоты сцены с двумя дронами и результаты наблюдений за их поведением.

Практическая работа 9. Интеграция и демонстрация — комплексная задача

Цель работы: интегрировать навыки, полученные на предыдущих занятиях, для выполнения комплексной автономной миссии.

Задачи:

1. Объединить управление высотой (PID), обнаружение препятствий (дальномеры) и навигацию (A*) в единую систему.
2. Реализовать сценарий: взлёт → облёт препятствий → обнаружение целевого объекта (камера) → посадка у цели.
3. Провести полный цикл тестирования и документирования результатов.
4. Подготовить демонстрационные материалы (видео, логи, отчёт).

Ход выполнения работы:

1. **Постановка задачи.** Определить конкретный сценарий. Например: дрон должен взлететь на высоту 2 м, проследовать через среду с препятствиями (здания, деревья) к заданной целевой области, обнаружить объект заданного цвета с помощью камеры, совершить посадку рядом с ним. Среда может быть основана на mavic_world.wbt с добавлением целевого объекта.
2. **Архитектура контроллера.** Построить конечный автомат (state machine) с состояниями:
   - TAKEOFF: взлёт на заданную высоту с использованием PID (занятие 3).
   - NAVIGATE: следование по пути A* к целевой области (занятие 7).
   - AVOID: объезд препятствий при их обнаружении (занятие 5).
   - SEARCH: поиск целевого объекта с помощью камеры (занятие 6).

- LAND: посадка при обнаружении цели.
- EMERGENCY: аварийная посадка по команде.

3. **Интеграция модулей.** Собрать контроллер, объединяющий функции из предыдущих занятий. Каждый модуль должен быть оформлен как отдельная функция или класс. Основной цикл:

```
python
state = "TAKEOFF"

while robot.step(timestep) != -1:
    if state == "TAKEOFF":
        # PID по высоте
        if abs(current_altitude - target_altitude) < 0.1:
            state = "NAVIGATE"

    elif state == "NAVIGATE":
        # Следование по пути A*
        # Проверка препятствий
        if obstacle_detected:
            state = "AVOID"
        # Проверка камеры на наличие цели
        if target_detected:
            state = "LAND"

    elif state == "AVOID":
        # Логика объезда
        # После объезда вернуться в NAVIGATE
        pass

    elif state == "LAND":
        # Посадка
        pass
```

4. **Обработка исключительных ситуаций.** Предусмотреть аварийные сценарии: потеря связи, низкий заряд батареи (если моделируется), обнаружение препятствия на слишком близком расстоянии. Реализовать аварийную посадку по команде с клавиатуры (например, клавиша L).
5. **Логирование и визуализация.** Записывать все ключевые события и данные датчиков в CSV-файл. Сохранять видео симуляции (функция записи в Webots). Строить графики траектории и временных характеристик.
6. **Тестирование и оптимизация.** Провести несколько прогонов в разных начальных условиях. Оценить:
   - Время выполнения миссии.
   - Количество столкновений или критических сближений с препятствиями.
   - Точность обнаружения цели.
   - Плавность полёта (отсутствие резких колебаний).
7. **Демонстрация.** Подготовить финальную демонстрацию: запустить симуляцию в режиме реального времени, показать полный цикл миссии. Подготовить презентацию с описанием архитектуры, ключевых алгоритмов и результатов.
8. **Оформление отчёта.** Включить полный листинг контроллера, описание архитектуры, логи данных, графики, видео (ссылку или файл) и анализ результатов с выводами о работоспособности системы и возможных улучшениях.

8.3. Типовые контрольные задания или иные материалы для проведения промежуточной аттестации

Форма проведения промежуточной аттестации по дисциплине) зачет (по результатам накопительного рейтинга или в форме компьютерного тестирования).

Устно-письменная форма по экзаменационным билетам предполагается, как правило, для сдачи академической задолженности

Примерный перечень вопросов и заданий для подготовки к зачёту

№	Содержание вопроса
УК-6. Способен управлять своим временем, выстраивать и реализовывать траекторию саморазвития на основе принципов образования в течение всей жизни	
1.	Понятие беспилотной системы. Отличия беспилотных, автоматических и автономных систем.
2.	История развития беспилотных технологий: основные этапы и ключевые достижения.
3.	Классификация беспилотных систем по среде функционирования: воздушные, наземные, водные, космические.
4.	Классификация беспилотных систем по назначению, массе и степени автономности.
5.	Обобщённая архитектура беспилотной системы: основные подсистемы и их назначение.
6.	Контур управления беспилотной системы: принцип обратной связи и роль человека-оператора.
7.	Информационно-измерительные подсистемы беспилотных систем: назначение и виды измерительных преобразователей.
ОПК-2 Способен понимать принципы работы современных информационных технологий и программных средств, в том числе отечественного производства, и использовать их при решении задач профессиональной деятельности	
8.	Первичная обработка измерительной информации в беспилотных системах.
9.	Исполнительные подсистемы беспилотных систем: приводы и механизмы.
10.	Двигательные установки беспилотных систем: типы и особенности применения.
11.	Навигационные системы беспилотных систем: GNSS, INS и их взаимодействие.
12.	Инерциальные навигационные системы: принцип работы, состав и области применения.
13.	Сенсорные системы беспилотных систем: видеокамеры, лидары, радары, ультразвуковые сенсоры.
14.	Стабилизированные подвесы и системы наведения полезных нагрузок.
15.	Коммуникационные системы беспилотных систем: каналы связи и телеметрия.
16.	Радиоуправление и автономные режимы полёта: сравнительная характеристика.
17.	Полезные нагрузки беспилотных систем: виды и назначение.
18.	Основы моделирования беспилотных систем: понятие модели и виды моделирования.
19.	Методология проектирования беспилотных систем: этапы и технические требования.
20.	Понятие CONOPS (концепция операций) и его роль в проектировании беспилотных систем.
21.	Верификация и валидация беспилотных систем: сущность и методы.
22.	Основы теории надёжности применительно к беспилотным системам.
23.	Виды испытаний беспилотных систем: наземные, лётные, имитационные.
24.	Методы обработки результатов испытаний беспилотных систем.
25.	Эксплуатация беспилотных систем: организация и регламенты технического обслуживания.
26.	Хранение и транспортировка беспилотных систем: требования и особенности.
27.	Показатели эффективности беспилотных систем: критерии и методы оценки.
28.	Технико-экономическая оценка беспилотных систем.
29.	Нормативно-правовое регулирование эксплуатации беспилотных систем: основные по-

№	Содержание вопроса
	ложения.
30.	Регулирование беспилотных систем в гражданском и военном секторе: сравнительный анализ.

Примерный тест для итогового тестирования

№	Вопрос
УК-6. Способен управлять своим временем, выстраивать и реализовывать траекторию саморазвития на основе принципов образования в течение всей жизни	
1.	1. Какое утверждение наиболее точно отражает различие между автоматической и автономной беспилотной системой? А) Автоматическая система не имеет датчиков, а автономная имеет Б) Автономная система способна принимать решения в условиях неопределённости без вмешательства оператора В) Автоматическая система всегда управляется дистанционно Г) Автономная система обязательно использует GNSS
2.	2. Какая характеристика отличает беспилотную систему от дистанционно пилотируемой? А) Наличие полезной нагрузки Б) Отсутствие пилота на борту В) Наличие контура автоматического управления Г) Использование радиоканала
3.	3. Что является определяющим признаком информационно-измерительной подсистемы? А) Преобразование энергии в механическое движение Б) Получение и первичная обработка данных о состоянии системы и среды В) Передача данных на наземный пункт управления Г) Обеспечение электропитанием бортовых потребителей
4.	4. Какое свойство контура управления с обратной связью является принципиальным? А) Наличие оператора в контуре Б) Сравнение текущего состояния с заданным и выработка корректирующего воздействия В) Использование только цифровых сигналов Г) Обязательное наличие GNSS
5.	5. Что означает аббревиатура CONOPS в контексте проектирования беспилотных систем? А) Concept of Operations Б) Control of Navigation and Positioning Systems В) Communication and Navigation Operational Protocol Г) Certification of New Operational Procedures
6.	6. В чём заключается основное назначение верификации беспилотной системы? А) Проверка соответствия системы требованиям технического задания Б) Проверка соответствия системы условиям эксплуатации В) Оценка экономической эффективности Г) Оценка эргономики рабочего места оператора
7.	7. Что является определяющим признаком имитационных испытаний беспилотной системы? А) Проведение испытаний на реальном аппарате Б) Проведение испытаний на математической или физической модели В) Проведение испытаний в реальных климатических условиях Г) Проведение испытаний с участием оператора
8.	8. Какое свойство отличает алгоритм A* от других алгоритмов поиска пути? А) Использование только жадного поиска Б) Использование эвристической функции для оценки стоимости пути В) Обязательное использование случайного выбора Г) Отсутствие необходимости в карте среды

ОПК-2 Способен понимать принципы работы современных информационных технологий и программных средств, в том числе отечественного производства, и использовать их при решении задач профессиональной деятельности	
9.	9. Что является определяющим признаком мультиагентной беспилотной системы? А) Наличие более одного беспилотного аппарата, взаимодействующих для достижения общей цели Б) Наличие более одного оператора В) Использование более одного канала связи Г) Наличие более одной полезной нагрузки
10.	10. В чём заключается основное отличие SITL от HITL? А) SITL использует реальное оборудование, HITL — модели Б) SITL использует модели, HITL — реальное оборудование В) SITL применяется только для наземных систем Г) HITL применяется только для воздушных систем
11.	11. Какие из перечисленных признаков относятся к классификации беспилотных систем по степени автономности? А) Дистанционно пилотируемые Б) Автоматические В) Автономные Г) Гидравлические
12.	12. Какие из перечисленных подсистем относятся к информационно-измерительным? А) GNSS-приёмник Б) Инерциальный измерительный блок В) Дальномер Г) Рулевая машинка
13.	13. Какие из перечисленных функций выполняет инерциальная навигационная система? А) Определение углов ориентации Б) Определение ускорений В) Определение координат Г) Обнаружение препятствий
14.	14. Какие из перечисленных протоколов относятся к беспроводным каналам связи беспилотных систем? А) Wi-Fi Б) LoRa В) Zigbee Г) USB
15.	15. Какие из перечисленных состояний могут входить в конечный автомат комплексной миссии? А) TAKEOFF Б) NAVIGATE В) LAND Г) COMPILE
16.	16. Какие из перечисленных методов относятся к обработке измерительной информации? А) Фильтрация Б) Сглаживание В) Экстраполяция Г) Рекристаллизация
17.	17. Какие из перечисленных критериев относятся к показателям эффективности беспилотной системы? А) Точность выполнения задачи Б) Время выполнения задачи В) Надёжность Г) Цветовая гамма корпуса
18.	18. Какие из перечисленных функций выполняет наземный пункт управления?

	А) Планирование миссии Б) Приём телеметрии В) Передача команд управления Г) Механическая обработка деталей
19.	**19.** Установите соответствие между подсистемой беспилотного аппарата и её функцией: Подсистема — *Функция* 1. Информационно-измерительная — А. Преобразование команд в механическое движение 2. Исполнительная — Б. Получение данных о состоянии системы и среды 3. Коммуникационная — В. Обеспечение электропитанием 4. Энергетическая — Г. Передача данных и команд
20.	**20.** Установите соответствие между датчиком и измеряемой величиной: Датчик — Измеряемая величина 1. Акселерометр — А. Угловая скорость 2. Гироскоп — Б. Линейное ускорение 3. Магнитометр — В. Расстояние до объекта 4. Дальномер — Г. Напряжённость магнитного поля
21.	**21.** Установите соответствие между алгоритмом и его назначением: Алгоритм — Назначение 1. PID — А. Планирование пути на графе 2. A* — Б. Стабилизация параметров движения 3. Фильтр Калмана — В. Оценка состояния в условиях шума 4. SLAM — Г. Одновременная локализация и картографирование
22.	**22.** Установите соответствие между видом испытаний и их содержанием: Вид испытаний — Содержание 1. Верификация — А. Проверка соответствия условиям эксплуатации 2. Валидация — Б. Проверка соответствия требованиям 3. Наземные испытания — В. Проверка на стенде без полёта 4. Лётные испытания — Г. Проверка в реальном полёте
23.	23. Расположите этапы жизненного цикла беспилотной системы в правильном порядке: А) Эксплуатация Б) Проектирование В) Испытания Г) Производство

24.	24. Расположите этапы обработки данных в контуре управления в правильном порядке: А) Выработка управляющего воздействия Б) Измерение параметров В) Сравнение с заданием Г) Передача на исполнительные механизмы
25.	25. Расположите этапы проектирования беспилотной системы в правильном порядке: А) Разработка технического задания Б) Разработка концепции операций В) Моделирование и симуляция Г) Верификация и валидация
26.	26. Расположите этапы выполнения комплексной миссии в правильном порядке: А) LAND Б) TAKEOFF В) SEARCH Г) NAVIGATE
27.	27. Как называется подсистема, преобразующая команды управления в механическое движение?
28.	28. Как называется алгоритм поиска пути, использующий эвристическую функцию?
29.	29. Как называется документ, определяющий концепцию операций беспилотной системы?
30.	30. Как называется режим испытаний, при котором модель автопилота работает на реальном оборудовании?