



МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение
высшего образования
«Дальневосточный федеральный университет»
(ДФУ)

ШКОЛА ЦИФРОВОЙ ЭКОНОМИКИ

СОГЛАСОВАНО
Руководитель ОП

 Е.В. Пустовалов

« 24 » июня 2018 г.

УТВЕРЖДАЮ»
Проректор по развитию

 Д.И. Земцов

« 27 » июня 2018 г.



РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

**«ТЕХНОЛОГИИ ВИРТУАЛЬНОЙ И ДОПОЛНЕННОЙ РЕАЛЬНОСТИ В
РОБОТОТЕХНИКЕ»**

направления 09.04.01 Информатика и вычислительная техника

Магистерская программа «Технологии виртуальной и дополненной реальности»

Форма подготовки очная

курс 2 семестр 3,4
лекции 18 час.
практические занятия 0 час.
лабораторные работы 54 час.
всего часов аудиторной нагрузки 72 час.
самостоятельная работа 144 час.
контрольные работы программой не предусмотрены
курсовая работа/проект – не предусмотрено
зачет с оценкой – 3,4 семестр
экзамен - не предусмотрено учебным планом

Рабочая программа составлена в соответствии с требованиями федерального государственного образовательного стандарта высшего образования по направлению подготовки 09.04.01 – Информатика и вычислительная техника, утвержденного приказом Министерства образования и науки Российской Федерации от 30.10.2014 № 1420

Рабочая программа рассмотрена и утверждена на заседании Дирекции Школы цифровой экономики 24 июня 2018 г., протокол №2

Составитель(и): к.т.н. Змеу В.К.; асс. Спорышев М.С.

Оборотная сторона титульного листа РПД

I. Рабочая программа пересмотрена на заседании Дирекции Школы цифровой экономики:

Протокол от «_____» _____ 20__ г. № _____

Заместитель директора ШЦЭ

по учебной и воспитательной работе _____
(подпись) (И.О. Фамилия)

II. Рабочая программа пересмотрена на заседании Дирекции Школы цифровой экономики:

Протокол от «_____» _____ 20__ г. № _____

Заместитель директора ШЦЭ

по учебной и воспитательной работе _____
(подпись) (И.О. Фамилия)

АННОТАЦИЯ

Рабочая программа учебной дисциплины «Технологии виртуальной и дополненной реальности в робототехнике» предназначена для студентов, обучающихся по направлению подготовки 09.04.01 Информатика и вычислительная техника (уровень магистратуры), профиль «Технологии виртуальной и дополненной реальности».

Рабочая программа разработана на основе макета рабочей программы учебной дисциплины для образовательных программ высшего образования – программ бакалавриата, специалитета, магистратуры ДВФУ, утверждённого приказом ректора ДВФУ от 08.05.2015 № 12-13-824.

Дисциплина «Технологии виртуальной и дополненной реальности в робототехнике» входит в вариативную часть блока «Дисциплины (модули) по выбору» (Б1.В.ДВ.02) учебного плана подготовки магистров.

Общая трудоемкость освоения дисциплины составляет 6 зачетных единиц или 216 часов. Дисциплина реализуется на 2 курсе в 3 и 4 семестре.

Семестр	Аудиторные занятия			Самостоя- тельная работа	Контроль	Всего по дисциплине	
	Лекци и	Лабор аторн ые работ ы	Всего			Часы	Зачетные единицы
3 семестр	–	36	36	72	Зачет оценкой	108	3
4 семестр	18	18	36	72	Зачет оценкой	108	3
Всего	18	54	72	144		216	6

Цель – изучение технологий и алгоритмов виртуальной (VR) и дополненной реальности (AR); получение практических навыков по применению дополненной реальности (AR) в современной робототехнике.

Задачи:

- изучение основных принципов функционирования беспилотных летательных аппаратов (БПЛА): краткая история, физическая модель

летательного аппарата, управление, в том числе решение задач навигации, выбора траектории полета, управление группой БПЛА;

- изучение методов и алгоритмов «компьютерного зрения», то есть методов и алгоритмов в области обработки изображений и поиска объектов, их применение в робототехнике и БПЛА;

- изучение технологических аспектов социальной робототехники, то есть реализации взаимодействия людей и роботов (HRI – human-robot interaction), в частности, изучение алгоритмов технического зрения и машинного обучения.

Для прохождения курса потребуются базовые знания линейной алгебры, дифференциального исчисления, языков программирования: C/C++ 11 или новее и/или Python. Желательно знание ROS (Kinetic, Melodic, Lunar), OpenCV, OpenGL, Blender.

В результате данной дисциплины у обучающихся формируются следующие профессиональные компетенции (элементы компетенций).

Код и формулировка компетенции	Этапы формирования компетенции	
ПК-8 – способность проектировать распределенные информационные системы, их компоненты и протоколы их взаимодействия	Знает	- методы проектирования распределенных информационных систем
	Умеет	- проектировать распределенные информационные системы, их компоненты и протоколы их взаимодействия
	Владеет	- навыками проектирования распределенных систем, их компонентов и протоколов их взаимодействия
ПК-13 – способность к программной реализации распределенных информационных систем	Знает	<ul style="list-style-type: none"> - основные стандарты в области организации доступа к распределенным информационным системам; - основные технологии реализации распределенных систем; - основные технологии поиска информации в распределенных информационных системах; - основные технологии представления и передачи структурированной информации в распределенных информационных системах
	Умеет	<ul style="list-style-type: none"> - проектировать распределенные информационные системы; - разрабатывать серверное и клиентское программное обеспечения распределенных информационных систем; - пользоваться архивами свободно распространяемого программного обеспечения;

		<ul style="list-style-type: none"> - конструировать программные комплексы для распределенных информационных систем; - организовывать преобразование данных на основе стандартных технологий; - создавать пользовательские интерфейсы для доступа к распределенным информационным системам.
	Владеет	навыками: <ul style="list-style-type: none"> - программной реализации распределенных информационных систем; - конструирования программных комплексов для распределенных информационных систем; - создания пользовательских интерфейсов для доступа к распределенным информационным системам.
ПК-16 – способность к созданию служб сетевых протоколов	Знает	<ul style="list-style-type: none"> - сетевые технологии и протоколы; - принципы построения сетевого взаимодействия
	Умеет	<ul style="list-style-type: none"> - программировать службы сетевых протоколов
	Владеет	<ul style="list-style-type: none"> - навыками создания служб сетевых протоколов
ПК-19 – способность к применению современных технологий разработки программных комплексов с использованием CASE-средств, контролировать качество разрабатываемых программных продуктов	Знает	<ul style="list-style-type: none"> - современные средства автоматизации процесса разработки информационных систем и программного обеспечения (CASE-средства); - основные функциональные, технические и эксплуатационные характеристики, предъявляемые к разрабатываемым программным продуктам
	Умеет	<ul style="list-style-type: none"> - разрабатывать программные комплексы с использованием современных CASE-средств; - контролировать качество разрабатываемых программных продуктов
	Владеет	<ul style="list-style-type: none"> - навыками использования CASE-средств; - методами анализа и повышения качества программных продуктов

Для формирования вышеуказанных компетенций в рамках дисциплины «Технологии виртуальной и дополненной реальности в робототехнике» применяются следующие методы активного/интерактивного обучения: лекция-беседа, метод автоматизированного обучения.

При выполнении различных видов работ используются следующие технологии:

1. *Проблемное обучение* – стимулирование обучающихся к самостоятельному приобретению знаний, необходимых для решения конкретной проблемы.

2. *Контекстное обучение* – мотивация студентов к усвоению знаний путём выявления связей между конкретным знанием и его применением.

3. *Обучение на основе опыта* – активизация познавательной деятельности обучающихся за счёт ассоциации и собственного опыта с предметом обучения.

I. СТРУКТУРА И СОДЕРЖАНИЕ ТЕОРЕТИЧЕСКОЙ ЧАСТИ КУРСА

4 семестр

Лекции (18 часов)

<p>Тема 1. Летающие роботы UAV (БПЛА)</p> <p>Основные принципы, заложенные в БПЛА: краткая история, физическая модель летательного аппарата, управление. А также такие продвинутые топики как сенсоры, навигация, поиск пути, а также полет в группах.</p>	<p>2 часа</p>
<p>Тема 2. Обработка изображений, дополненная реальность (AR)</p> <p>Основы обработки изображений и AR: проекция, фильтры, отслеживание движения объектов и пр.</p> <p>Более сложные алгоритмы обработки изображений и AR и их применение в робототехнике, в частности БПЛА.</p>	<p>2 часа</p>
<p>Тема 3. Взаимодействие людей и роботов (HRI)</p> <p>Современные исследования HRI и технологические аспекты необходимые для их реализации. В частности, алгоритмы компьютерного зрения и, возможно, машинного обучения.</p>	<p>2 часа</p>
<p>Тема 4. Симуляторы</p> <p>Современное ПО для симуляции миссий автономных роботов. Gazebo. AirSim</p>	<p>2 часа</p>
<p>Тема 5. 3d реконструкция. Одновременная локализация и построение карты</p> <p>Основные методы навигации мобильных роботов. Simultaneous localization and mapping. Алгоритмы построения карты</p>	<p>2 часа</p>
<p>Тема 6. Архитектура ПО для робота</p> <p>Архитектура ПО для автономных роботов. Встраиваемые системы. Операционные системы реального времени (RTOS). Системы обмена сообщениями. Robot operating system (ROS).</p>	<p>2 часа</p>
<p>Тема 7. Обработка и фильтрация данных</p> <p>Фильтр Калмана, фильтр частиц. Data fusion.</p>	<p>2 часа</p>
<p>Тема 8. Планирование движения</p> <p>Алгоритмы построения оптимальных маршрутов и траекторий для роботов. Алгоритм A*. Vehicle Routing Problem. Задачи оптимизации для построения траекторий для роботов с большим</p>	<p>2 часа</p>

количеством степеней свободы.	
Тема 9. Обучение с подкреплением для задач управления аппаратом Алгоритмы обучения с подкреплением. TD(λ), Sarsa, Q-learning. Методы Actor-critic. Proximal Policy Optimization.	2 часа

II. СТРУКТУРА И СОДЕРЖАНИЕ ПРАКТИЧЕСКОЙ ЧАСТИ КУРСА

3 семестр

Лабораторные работы (36 часов)

Лабораторная работа №1. Основы Robot Operating System (ROS) Требуется установить ROS, собрать первый проект. Запустить turtle-bot и модуль управления с клавиатуры.	5 часов
Лабораторная работа №2. Основы Gazebo В качестве практического задания предполагается задачи в симуляторе Gazebo. Использовать любую существующую модель робота и среды. Сделать управление роботом с клавиатуры.	5 часов
Лабораторная работа №3. Алгоритмы планирования движением Реализовать алгоритм обхода всей доступной области в среде Gazebo и протестировать на любой модели робота.	5 часов
Лабораторная работа №4. SLAM Реализовать построение карты используя LIDAR. Реализовать построение карты используя карту глубин с камеры.	5 часов
Лабораторная работа №5. Летящие роботы UAV (БПЛА) В качестве практического задания предполагается задачи в симуляторе Gazebo, а также рассматривается вариант с выполнением на реальном роботе.	6 часов
Лабораторная работа №6. Обработка изображений, дополненная реальность (AR) В качестве практического задания рассматривается отслеживание камеры, отслеживание движущихся маркеров	5 часов
Лабораторная работа №7. Взаимодействие людей и роботов (HRI) Реализовать распознавание жестов	5 часов

4 семестр
Лабораторные работы (18 часов)

Лабораторная работа №1. Взаимодействие людей и роботов, AR. Требуется реализовать визуализацию решений, принимаемых роботом с помощью дополненной реальности.	6 часов
Лабораторная работа №2. Интерфейс управления роботом с помощью виртуальной реальности Реализовать визуализацию данных с камеры с помощью виртуальной реальности. Управление летательным аппаратом с помощью контроллеров HTC Vive или аналогичных.	6 часов
Лабораторная работа №3. Реализация на реальном роботе Реализация алгоритма обхода области на реальном летательном аппарате (дроне). Визуализация в VR.	6 часов

III. УЧЕБНО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ САМОСТОЯТЕЛЬНОЙ РАБОТЫ ОБУЧАЮЩИХСЯ

Учебно-методическое обеспечение самостоятельной работы обучающихся по дисциплине «Технологии виртуальной и дополненной реальности в робототехнике» представлено в Приложении 1 и включает в себя:

- план-график выполнения самостоятельной работы по дисциплине, в том числе примерные нормы времени на выполнение по каждому заданию;
- характеристика заданий для самостоятельной работы обучающихся и методические рекомендации по их выполнению;
- требования к представлению и оформлению результатов самостоятельной работы;
- критерии оценки выполнения самостоятельной работы.

IV. КОНТРОЛЬ ДОСТИЖЕНИЯ ЦЕЛЕЙ КУРСА

№ п/п	Контролируемые разделы / темы дисциплины	Коды и этапы формирования компетенций		Оценочные средства	
				текущий контроль	промежуточная аттестация
1	Летающие роботы UAV (БПЛА)	ПК-8	знает	Выполнение лабораторных работ	Зачет с оценкой
		ПК-13	умеет		
		ПК-16	владеет		
		ПК-19			
2	Обработка изображений, дополненная реальность (AR)	ПК-8	знает	Выполнение лабораторных работ	Зачет с оценкой
		ПК-13	умеет		
		ПК-16	владеет		
		ПК-19			
3	Взаимодействие людей и роботов (HRI)	ПК-8	знает	Выполнение лабораторных работ	Зачет с оценкой
		ПК-13	умеет		
		ПК-16	владеет		
		ПК-19			

1. устный опрос (УО): собеседование (УО-1), коллоквиум (УО-2); итоговая презентация (УО-3); круглый стол (УО-4);
2. технические средства контроля (ТС);

3. письменные работы (ПР): тесты (ПР-1), контрольные работы (ПР-2), эссе (ПР-3), рефераты (ПР-4), курсовые работы (ПР-5), научно-учебные отчеты по практикам (ПР-6), конспект (ПР-7), проект (ПР-9). Разноуровневые задачи и задания (ПР-11) и т.п.

Типовые контрольные задания, методические материалы, определяющие процедуры оценивания знаний, умений и навыков и (или) опыта деятельности, а также критерии и показатели, необходимые для оценки знаний, умений, навыков и характеризующие этапы формирования компетенций в процессе освоения образовательной программы, представлены в Приложении 2.

V. СПИСОК УЧЕБНОЙ ЛИТЕРАТУРЫ И ИНФОРМАЦИОННО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

Основная литература

(электронные и печатные издания)

1. Рэндал У. Биард Малые беспилотные летательные аппараты [Электронный ресурс]: теория и практика/ Рэндал У. Биард, Тимоти У. МакЛэйн— Электрон. текстовые данные.— М.: Техносфера, 2015.— 312 с.— Режим доступа: <http://www.iprbookshop.ru/36871.html>.— ЭБС «IPRbooks»

2. Филлипс Ч., Харбор Р. Системы управления с обратной связью. – Лаборатория базовых знаний, 2001.

3. Carbone G., Gomez-Bravo F. Motion and operation planning of robotic systems //Springer International Publishing. Switzerland. DOI. – 2015. – Т. 10. – С. 978-3.

4. Джонатан, Л. Виртуальная реальность в Unity [Электронный ресурс] / Л. Джонатан ; пер. с англ. Р.Н. Рагимов. — Электрон. дан. — Москва : ДМК Пресс, 2016. — 316 с. — Режим доступа: <https://e.lanbook.com/book/93271>. — Загл. с экрана.

5. Иванцовская Н.Г. Перспектива. Теория и виртуальная реальность [Электронный ресурс]: учебное пособие/ Иванцовская Н.Г.— Электрон. текстовые данные.— Новосибирск: Новосибирский государственный технический университет, 2010.— 197 с.— Режим доступа: <http://www.iprbookshop.ru/44820.html>.— ЭБС «IPRbooks»

6. Маров М. Н. 3ds max. Реальная анимация и виртуальная реальность/ Санкт-Петербург: Питер, 2006. – 414 с.

Дополнительная литература
(печатные и электронные издания)

1. Красильников, М.Н. Современные информационные технологии. В задачах навигации и наведения беспилотных маневренных летательных аппаратов [Электронный ресурс] : учебное пособие / М.Н. Красильников, Г.Г. Серебряков. — Электрон. дан. — Москва : Физматлит, 2009. — 557 с. — Режим доступа: <https://e.lanbook.com/book/2688>. — Загл. с экрана.
2. Шалыгин, А.С. Методы моделирования ситуационного управления движением беспилотных летательных аппаратов [Электронный ресурс] : учебно-методическое пособие / А.С. Шалыгин, Л.Н. Лысенко, О.А. Толпегин. — Электрон. дан. — Москва : Машиностроение, 2012. — 584 с. — Режим доступа: <https://e.lanbook.com/book/5807>. — Загл. с экрана.
3. Мак-Илврайд, Б. Распределенное управление беспилотными летательными аппаратами / Боб Мак-Илврайд. 2009. Современные технологии автоматизации . - N 3 (2009), С. 64-65 stav09_no3
4. Торн, А. Основы анимации в Unity [Электронный ресурс] / А. Торн ; пер. с англ. Р. Рагимова. — Электрон. дан. — Москва : ДМК Пресс, 2016. — 176 с. — Режим доступа: <https://e.lanbook.com/book/73075>. — Загл. с экрана.
5. Торн, А. Искусство создания сценариев в Unity [Электронный ресурс] : руководство / А. Торн ; пер. с англ. Р. Н. Рагимова. — Электрон. дан. — Москва : ДМК Пресс, 2016. — 360 с. — Режим доступа: <https://e.lanbook.com/book/82812>. — Загл. с экрана.
6. Дикинсон, К. Оптимизация игр в Unity 5 [Электронный ресурс] / К. Дикинсон. — Электрон. дан. — Москва : ДМК Пресс, 2017. — 306 с. — Режим доступа: <https://e.lanbook.com/book/90109>. — Загл. с экрана.
7. Вдовин А.С. Дизайн игр и медиаиндустрии. Персонажная графика и анимация [Электронный ресурс]: учебное пособие/ Вдовин А.С.— Электрон. текстовые данные.— Саратов: Саратовский государственный технический университет имени Ю.А. Гагарина, ЭБС АСВ, 2015.— 267 с.— Режим доступа: <http://www.iprbookshop.ru/76480.html>.— ЭБС «IPRbooks»
8. Смолин А.А., Жданов Д.Д., Потемин И.С., Меженин А.В., Богатырев В.А. Системы виртуальной, дополненной и смешанной реальности Учебное пособие. – Санкт- Петербург: Университет ИТМО. 2018 . – 59 с.
<https://books.ifmo.ru/file/pdf/2321.pdf>

9. Фореман Н. ., Коралло Л. Прошлое и будущее 3D-технологий виртуальной реальности. Научно-технический вестник ИТМО. ноябрь-декабрь 2014. [Электронный ресурс]. Режим доступа http://ntv.ifmo.ru/ru/article/11182/proshloe_i_budushee_3D_tehnologiy_virtualnoy_realnosti.htm
10. Виртуальная реальность. Единая коллекция цифровых образовательных ресурсов 2017 [Электронный ресурс]. Режим доступа <http://files.schoolcollection.edu.ru/dlrstore/39131517-5991-11da-8314-0800200c9a66/index.htm>
11. Полное погружение в виртуальную реальность: настоящее и будущее. 2017 [Электронный ресурс]. Режим доступа <https://habrahabr.ru/company/miip/blog/330754/>
12. Виртуальная реальность (VR): прошлое, настоящее и будущее 2017 [Электронный ресурс]. Режим доступа <http://vrmania.ru/stati/virtualnaya-realnost.html>
13. 12 платформ разработки приложений дополненной реальности 2017 [Электронный ресурс]. Режим доступа <https://apptractor.ru/info/articles/12-platform-razrabotki-prilozheniy-dopolnennoyrealnosti.html>

Перечень ресурсов информационно-телекоммуникационной сети «Интернет»

1. Robot Operating System (ROS) — это гибкая среда для написания программного обеспечения для роботов. Это набор инструментов, библиотек и соглашений, призванных упростить задачу создания сложного и надежного поведения робота на самых разных роботизированных платформах: www.ros.org
2. <http://gazebosim.org/>
3. www.generationrobots.com/en/content/83-carry-out-simulations-and-make-your-darwin-op-walk-with-gazebo-and-ros
4. <https://robotacademy.net.au/> – **Открытые лекции** от автора курса «Введение в робототехнику», Квинслендский технологический университет.

VI. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ

VII. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

<p>Компьютерный класс:</p> <p>Проектор DLP, 3000 ANSI Lm, WXGA 1280x800, 2000:1 EW330U Mitsubishi; Системный блок с монитором. Процессор: Intel I5-8600k 3.6Ghz, оперативная память: 32gb, жесткий диск: 1ТБ, графический ускоритель: Nvidia GTX 1080 Беспроводные ЛВС для обучающихся обеспечены системой на базе точек доступа 802.11a/b/g/n 2x2 MIMO(2SS).</p> <p>Специализированное ПО: Visual Studio 2019, Unity 2019, STEAM VR, Anaconda, Oculus Rift SDK, HTC VIVE SDK</p>	<p>690922, Приморский край, г. Владивосток, о. Русский, п. Аякс, 10, г. Владивосток, о. Русский, п. Аякс, корпус G, ауд. G464</p>
---	---

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение
высшего профессионального образования
«Дальневосточный федеральный университет»
(ДВФУ)

ШКОЛА ЦИФРОВОЙ ЭКОНОМИКИ

**УЧЕБНО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ САМОСТОЯТЕЛЬНОЙ
РАБОТЫ ОБУЧАЮЩИХСЯ**

по дисциплине

ТЕХНОЛОГИИ ВИРТУАЛЬНОЙ И ДОПОЛНЕННОЙ РЕАЛЬНОСТИ В РОБОТОТЕХНИКЕ

Направление подготовки

09.04.01 Информатика и вычислительная техника

магистерская программа

«Технологии виртуальной и дополненной реальности»

Форма подготовки очная

Владивосток

2018

Учебно-методическое обеспечение самостоятельной работы обучающихся по дисциплине «Технологии виртуальной и дополненной реальности в робототехнике» включает в себя:

- план-график выполнения самостоятельной работы по дисциплине, в том числе примерные нормы времени на выполнение по каждому заданию;
- характеристика заданий для самостоятельной работы обучающихся и методические рекомендации по их выполнению;
- требования к представлению и оформлению результатов самостоятельной работы;
- критерии оценки выполнения самостоятельной работы.

План-график выполнения самостоятельной работы по дисциплине

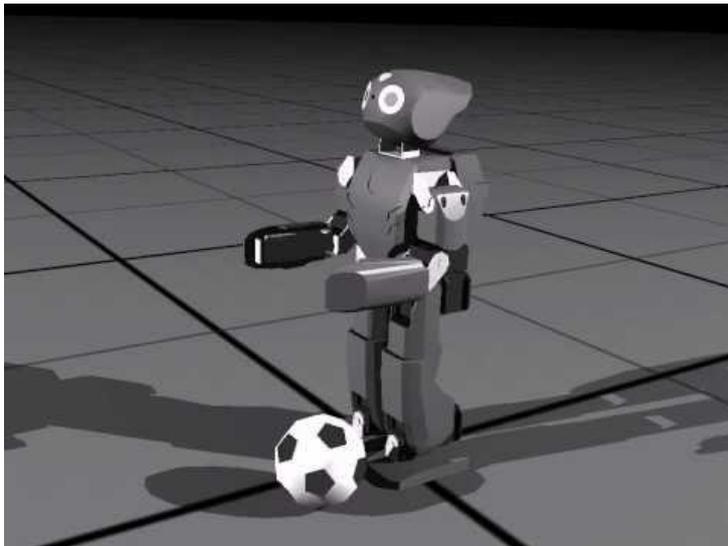
№ п/п	Дата/сроки выполнения	Вид самостоятельной работы	Примерные нормы времени на выполнение	Форма контроля
1 семестр			72 часа	
1	1 – 17 недели	Подготовка к лабораторным работам	66 часов	Выполнение лабораторных работ
2	18 неделя	Подготовка к зачету	6 часов	Зачет с оценкой
2 семестр			72 часа	
1	1 – 17 недели	Подготовка к лабораторным работам	30 часов	Выполнение лабораторных работ
2	18 неделя	Подготовка к зачету	6 часов	Зачет с оценкой

Примеры заданий для самостоятельной работы

Задание 1. Использование симуляторов в робототехнике на примере симулятора Gazebo и робота DARwIn-OP, играющего в футбол

Попробуем разобраться с использованием возможностей современных физических симуляторов и программных платформ для облегчения разработки и отладки ПО при создании сложных робототехнических систем.

В качестве симулятора мы возьмем пакет Gazebo 3D, а человекоподобный DARwIn-OP, которого мы заставим играть в футбол, будет использован в качестве универсальной робототехнической платформы.



DARwin-OP в симуляторе Gazebo

Для изучения уроков, использования примеров кода, а также написания собственных алгоритмов вам не потребуется приобретение дорогостоящего оборудования или ПО. Все программное обеспечение бесплатно, а отладка и демонстрация алгоритмов будет проводиться в симуляторе. Законченный проект будет портирован на реального робота для демонстрации результатов.

Gazebo 3D, разрабатываемый некоммерческой организацией OSRF (Open Source Robotics Foundation), имеет ряд преимуществ по сравнению с другими робототехническими симуляторами. Во-первых, он бесплатный и имеет открытый код. Во-вторых, он очень популярен среди мирового робототехнического сообщества и является официальным симулятором соревнований DARPA. В-третьих, Gazebo отлично интегрируется с программной платформой ROS (Robot Operating System), а значит разработанную вами программу управления виртуальным роботом в Gazebo и ROS будет относительно несложно перенести на реального робота.



Робот PR2 в симуляторе Gazebo

В этом уроке мы изучим, как установить последнюю версию Gazebo, подключить ее к программной платформе **ROS**, загрузить физическую модель робота DARwin и заставить ее

ходить. В конце мы настроим простую систему телеуправления движением робота с клавиатуры. Параллельно будем осваивать инфраструктуру ROS.

Установка ROS

Первым и самым трудоемким шагом является установка ROS и Gazebo. Для установки ROS нам понадобится ПК с Linux (например Ubuntu 15.04). Если вы пользователь Windows или Mac OS, то можно воспользоваться виртуальной машиной, например Virtual Box с установкой Linux Ubuntu. Сразу хочется отметить, что мне не удалось настроить нормальную поддержку 3D под виртуальной машиной, Gazebo либо вылетал, либо работал с тормозами. Поэтому в качестве тренировки, можно воспользоваться виртуальной машиной, но для реальной разработки целесообразно все-таки использовать Linux.

ROS можно установить двумя способами. Первый способ относительно простой — это установка уже готовых Debian пакетов. Данный способ подходит в том случае, если у вас установлен Ubuntu Saucy (13.10) или Trusty (14.04), или, например, Debian Wheezy. Второй способ более сложный и запутанный — это установка из исходников. С помощью него можно произвести установку ROS в более свежих версиях Ubuntu или других дистрибутивов Linux.

Пример установки приведенный ниже осуществляется в Linux Ubuntu 15.04.

Итак, приступим!

Откройте bash terminal через меню программ или нажав комбинацию клавиш ctrl+alt+t. Сначала необходимо установить утилиты для самозагрузки и инициализации ROS. Если вы используете Ubuntu или Debian это можно сделать добавив в репозиторий ссылку на packages.ros.org.

```
$ sudo sh -c 'echo "deb http://packages.ros.org/ros/ubuntu $(lsb_release -sc) main" > /etc/apt/sources.list.d/ros-latest.list'
$ wget https://raw.githubusercontent.com/ros/rosdistro/master/ros.key -O - | sudo apt-key add -
$ sudo apt-get update
$ sudo apt-get install python-rosdep python-rosinstall-generator python-wstool python-rosinstall build-essential
```

Замечание

Инициализируем rosdep. Rosdep — инструмент командной строки для установки системных зависимостей ROS.

```
$ sudo rosdep init
$ rosdep update
```

Теперь приступим к сборке пакетов ROS с помощью утилиты catkin. Catkin — это набор макросов для сборки и управления инфраструктурой ROS.

Создадим каталог для установки ROS и загрузим ключевые пакеты.

```
$ mkdir ~/workspace/ros -p
$ cd ~/workspace/ros
```

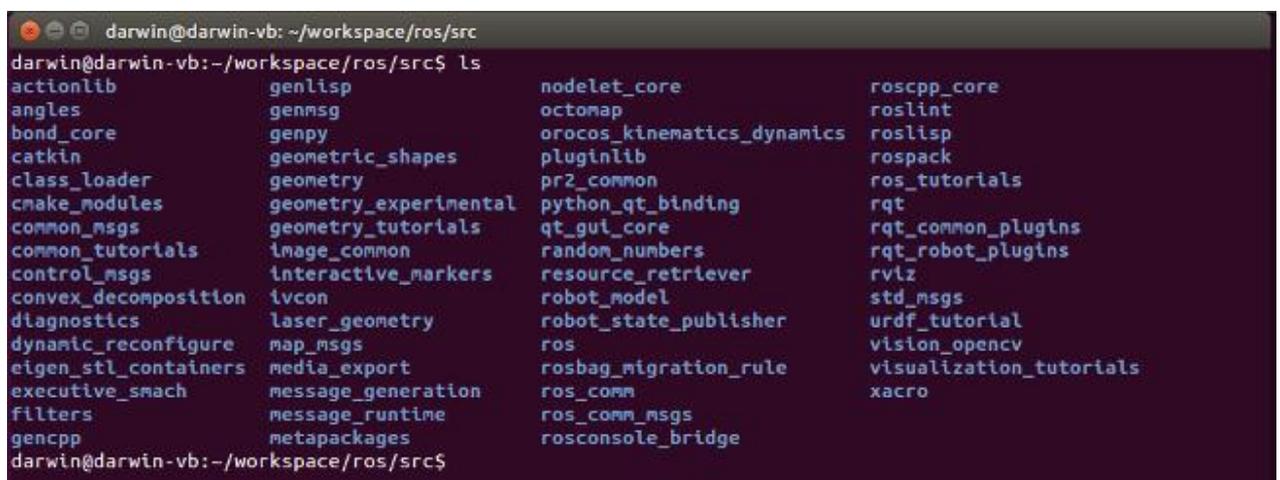
Команда `roscpp_core` создает список пакетов необходимых для установки ROS, в качестве параметров может быть указана минимальная установка `ros_comm`, базовая `desktop` или полная `desktop_full`. Установка `desktop_full` включает версию Gazebo 2.0. Эта версия уже устарела, текущей является версия 5.x. Поэтому мы выберем конфигурацию `desktop`, а Gazebo установим отдельно.

```
$ roscpp_core desktop --rosdistro indigo --deps --wet-only --tar > indigo-desktop-wet.roscpp_core
```

После того как список пакетов для установки сгенерирован в файл `indigo-desktop-wet.roscpp_core`, воспользуемся утилитой `wstool` для инициализации рабочей области и загрузки пакетов.

```
$ wstool init src indigo-desktop-wet.roscpp_core
```

Замечание



```
darwin@darwin-vb: ~/workspace/ros/src
darwin@darwin-vb:~/workspace/ros/src$ ls
actionlib          genlisp            nodelet_core      roscpp_core
angles            genmsg            octomap           roslint
bond_core         genpy             orocos_kinematics_dynamics  roslisp
catkin            geometric_shapes  pluginlib         rospack
class_loader      geometry          pr2_common        ros_tutorials
snake_modules     geometry_experimental  python_qt_binding  rqt
common_msgs       geometry_tutorials  qt_gui_core       rqt_common_plugins
common_tutorials  image_common       random_numbers    rqt_robot_plugins
control_msgs      interactive_markers  resource_retriever  rviz
convex_decomposition  ivcon             robot_model       std_msgs
diagnostics       laser_geometry     robot_state_publisher  urdf_tutorial
dynamic_reconfigure  map_msgs          ros                vision_opencv
eigen_sfl_containers  media_export       rosbag_migration_rule  visualization_tutorials
executive_smach     message_generation  ros_comm          xacro
filters           message_runtime    ros_comm_msgs
gencpp            metapackages       rosconsole_bridge
```

После установки в каталоге `src` можно найти скачанные пакеты

Теперь переместимся в корневой каталог `ros` и выполним команду `rosdep` для установки системных зависимостей для скачанных пакетов.

```
$ cd ~/workspace/ros
$ rosdep install --from-paths src --ignore-src --rosdistro indigo -y
```

После успешной установки всех зависимостей, наконец можно откомпилировать пакеты с помощью системы автоматизации сборки `catkin`. Это делается командой `catkin_make_isolated` или командой `catkin_make`. `Catkin_make_isolated` следует использовать, если часть ваших пакетов использует `CMake` (другая популярная система автоматизации сборки), а часть `catkin`, если же все пакеты используют `catkin`, следует применять `catkin_make`. Базовый репозиторий имеет смешанный тип, поэтому используем команду `catkin_make_isolated`.

```
$ ./src/catkin/bin/catkin_make_isolated --install -DCMAKE_BUILD_TYPE=Release
```

Для того, чтобы инфраструктура ROS заработала, необходимо настроить переменные окружения и прописать все необходимые пути. Для этого следует выполнить скрипт

setup.bash.

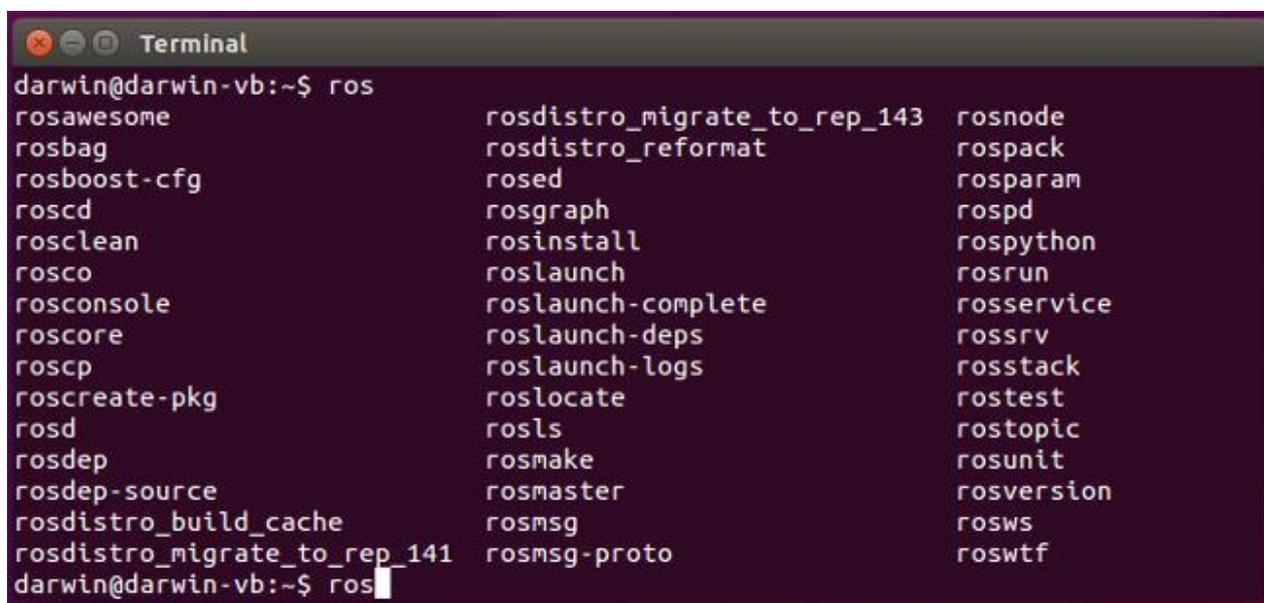
```
$ source ~/workspace/ros/install_isolated/setup.bash
```

Это команда настроит переменные окружения только для текущей bash консоли. Если Вы хотите добавить переменные окружения ROS в bash по умолчанию, то можно записать команду source в файл .bashrc. Этот файл исполняется при открытии новой bash консоли.

```
$ echo "source ~/workspace/ros/install_isolated/setup.bash" > ~/.bashrc
```

Поздравляю! На этом установка ROS окончена, давайте выполним несколько команд, чтобы проверить правильность работы.

Для того чтобы вывести список доступных команд для управления инфраструктурой ROS, наберите в консоли команду ros и нажмите клавишу Tab два раза. Так как все команды ROS начинаются с префикса ros, bash выдаст их названия как результат автодополнения.



```
darwin@darwin-vb:~$ ros
rosawesome          rosdistro_migrate_to_rep_143  rosnode
rosbag              rosdistro_reformat            rospack
rosboost-cfg       rosed                          rosparam
roscd               rosgraph                       rospd
rosclean           rosinstall                     rospython
rosco               roslaunch                     rosrun
rosconsole         roslaunch-complete            rosservice
roscore            roslaunch-deps                rossrv
roscp              roslaunch-logs                rosstack
roscreate-pkg      roslocate                      rostest
rosd               rosls                          rostopic
rosdep             rosmake                        rosunit
rosdep-source      rosmaster                      rosversion
rosdistro_build_cache
rosdistro_migrate_to_rep_141  rosmsg                          rosxs
rosdistro_migrate_to_rep_143  rosmsg-proto                   rosxtf
darwin@darwin-vb:~$ ros
```

Список команд ROS. Все команды начинаются с префикса ros

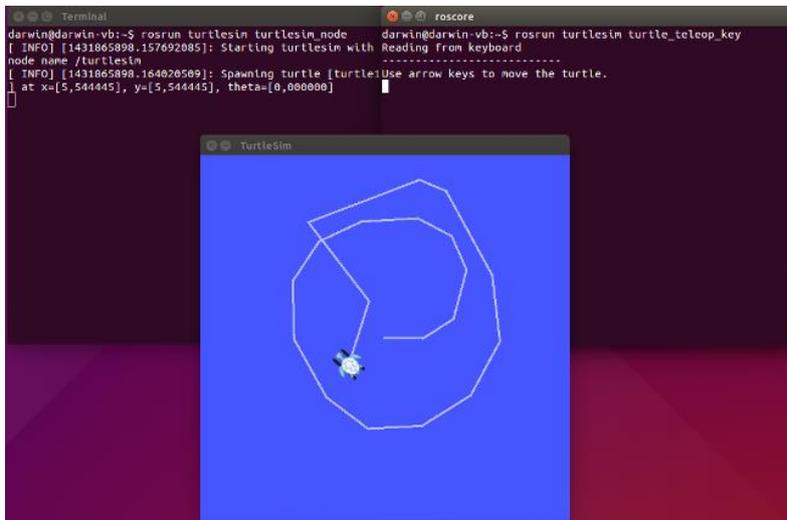
Давайте что-нибудь запустим, например простое телеуправление. Для начала нам нужен объект которым мы хотим управлять, пускай им будет черепаха из turtlesim. Итак, запускаем сервер turtlesim с помощью команды rosrun. Для этого открываем новую консоль и выполняем команду.

```
$ rosrun turtlesim turtlesim_node
```

На экране появится окно с черепашкой. Теперь давайте запустим модуль телеуправления, для этого откройте новую консоль и выполните команду.

```
$ rosrun turtlesim turtle_teleop_key
```

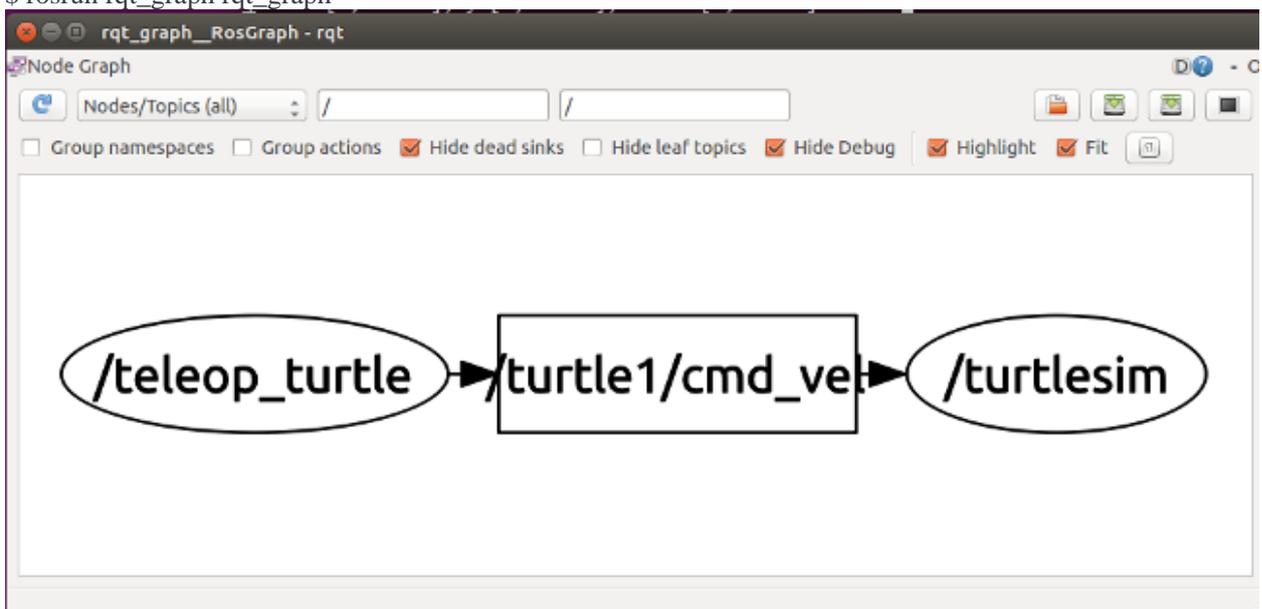
А теперь с помощью клавиш курсора попробуйте подвигать черепаху. Например, вот так.



Пример телеуправления черепахой

Итак, только что мы построили распределенную систему для телеуправления, где turtlesim и teleop являются независимыми процессами коммуницирующими через связующее программное обеспечение ros_comm. Процессы turtlesim и teleop могут находиться как на одном, так и на различных ПК, объединенных в сеть. Отобразить взаимосвязь этих процессов можно с помощью утилиты rqt_graph.

\$ rosrun rqt_graph rqt_graph



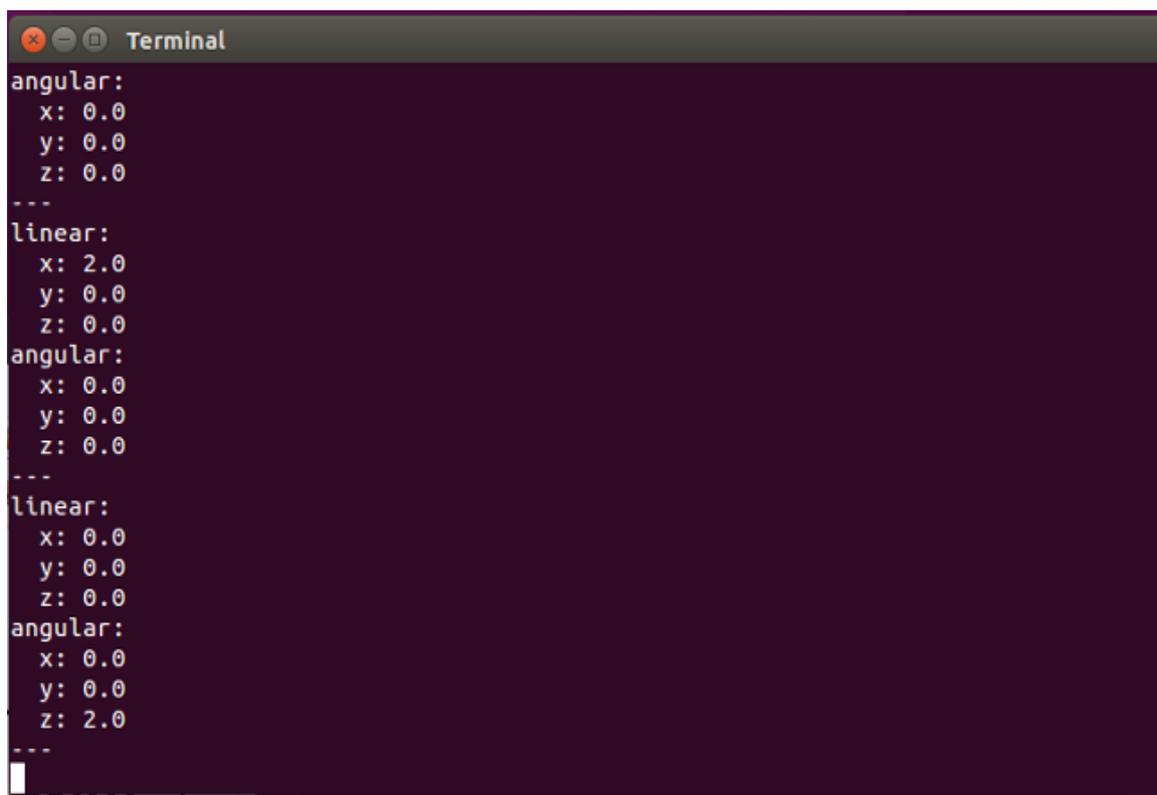
Утилита rqt_graph служит для отображения топологии распределенной системы управления роботом

В rqt_graph овалами обведены различные процессы (ноды) нашей системы управления, а прямоугольниками типы данных (топики) служащие для передачи информации между процессами. Стрелки показывают направление передачи данных. Таким образом, мы видим, что нод teleop публикует топик cmd_vel, содержащий заданную скорость робота, а turtlesim подписан на данный топик.

Заглянуть в содержимое топика можно с помощью команды rostopic. Параметр list выдаст все активные топики системы, а команда echo выведет на экран содержимое указанного топика. Чтобы отобразить содержимое cmd_vel, не забудьте поручить черепахой из

соседней консоли, иначе данные не будут передаваться.

```
$ rostopic list  
$ rostopic echo /turtle1/cmd_vel
```



```
angular:  
  x: 0.0  
  y: 0.0  
  z: 0.0  
---  
linear:  
  x: 2.0  
  y: 0.0  
  z: 0.0  
angular:  
  x: 0.0  
  y: 0.0  
  z: 0.0  
---  
linear:  
  x: 0.0  
  y: 0.0  
  z: 0.0  
angular:  
  x: 0.0  
  y: 0.0  
  z: 2.0  
---
```

Содержимое /turtle1/cmd_vel — заданная линейная и угловая скорость черепахи

После успешной установки ROS приступим к установке симулятора Gazebo.

Установка Gazebo

Gazebo мы не будем компилировать из исходников, а просто установим последнюю версию с помощью менеджера пакетов Ubuntu. Для этого мы подключим репозиторий с Gazebo, добавим ключ и произведем установку.

```
$ sudo sh -c 'echo "deb http://packages.osrfoundation.org/gazebo/ubuntu `lsb_release -cs` main" >  
/etc/apt/sources.list.d/gazebo-latest.list'  
$ wget http://packages.osrfoundation.org/gazebo.key -O - | sudo apt-key add -  
$ sudo apt-get update  
$ sudo apt-get install gazebo5 libgazebo-dev
```

Теперь необходимо настроить конфигурацию `rosdep` так, чтобы система игнорировала зависимость пакетов от более старой версии Gazebo, являющейся частью ROS Indigo, а использовала бы последнюю версию, которую мы только что установили. Для этого мы перейдем в каталог конфигурационных файлов `rosdep`, создадим файл `local.yaml`, в котором укажем игнорировать пакет `gazebo`. Далее перейдем в каталог `sources.list.d` и добавим только что созданный скрипт в список `10-local.list`. Далее выполним команду `update` для того, чтобы новая конфигурация вступила в действие.

```
$ cd /etc/ros/rosdep
$ sudo bash -c 'echo "gazebo: { ubuntu: { vivid: [] } }" > local.yaml'
$ cd ./sources.list.d
$ sudo bash -c 'echo "yaml file:///etc/ros/rosdep/local.yaml" > 10-local.list'
$ rosdep update
```

На финальном этапе необходимо установить пакеты связующие симулятор Gazebo с инфраструктурой ROS, а также пакеты с описанием 3D модели и контроллеров робота DARwIn-OP. Для упрощения этого процесса мы подготовили небольшой скрипт, который произведет загрузку всех необходимых зависимостей.

Перейдем в рабочий каталог и загрузим скрипт из нашего репозитория на github.

```
$ cd ~/workspace
$ git clone https://github.com/robotgeeks/ros_gazebo_darwin.git
```

Далее заходим в каталог `ros_gazebo_darwin`, как было описано ранее, инициализируем рабочую область `catkin` и загружаем пакеты перечисленные в файле `indigo-gazebo-darwin.rosinstall`.

```
$ cd ~/workspace/ros_gazebo_darwin
$ wstool init src indigo-gazebo-darwin.rosinstall
```

Далее производим компиляцию и установку только что скачанных пакетов.

```
$ catkin_make install
```

Теперь добавим переменные окружения и пути для нашей рабочей области. Для этого выполним следующие команды.

```
$ source ~/workspace/ros_gazebo_darwin/install/setup.bash
$ echo "source ~/workspace/ros_gazebo_darwin/install/setup.bash" > ~/.bashrc
```

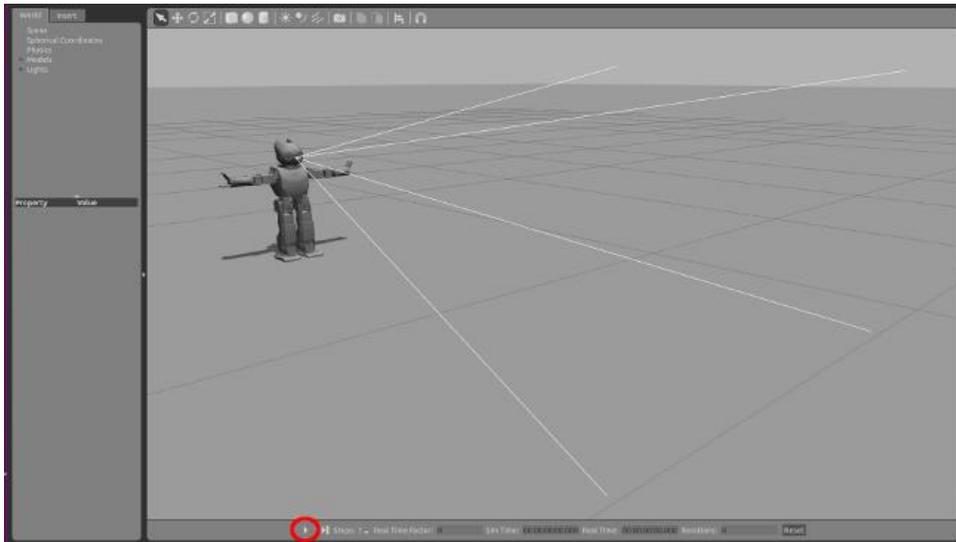
Отлично! Установка Gazebo и модели DARwIn-OP успешно завершена, давайте поскорее запустим результат.

Работа в симуляторе Gazebo

Наконец, после долгих страданий, пришло время запустить симулятор с моделью робота. Для этого с помощью команды `roslaunch` выполним скрипт `darwin_gazebo.launch`.

```
$ roslaunch darwin_gazebo darwin_gazebo.launch
```

Результатом выполнения скрипта должен стать запуск окна Gazebo и загрузка 3D модели робота с соответствующими виртуальными контроллерами и сенсорами. Для запуска симуляции необходимо нажать кнопку `play`, расположенную внизу окна на панели виртуального таймера.



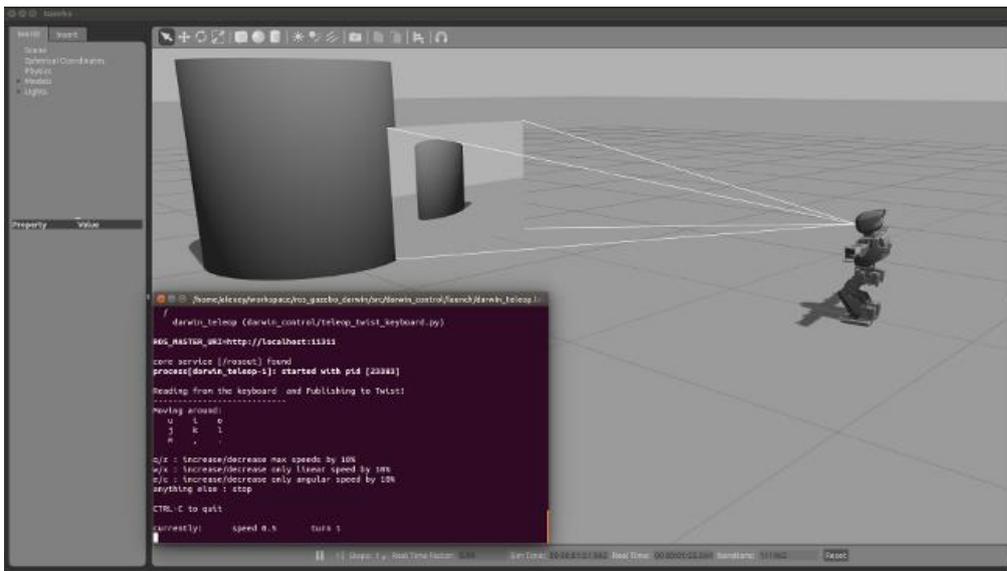
Симулятор Gazebo с загруженной моделью робота DARwIn-OP. (Не забудьте нажать на кнопку play, помеченную красным)

Теперь откроем новую консоль и запустим программу управления движением робота. Она преобразует команды скорости из декартовых координат в движение сервомоторов, расположенных в ногах, торсе и руках робота.

```
$ roslaunch darwin_gazebo darwin_walker.launch
```

Наконец, в третьей консоли запустим программу телеуправления. Ее задача считывать команды с клавиатуры, преобразовать их в заданную скорость и передавать программе управления движением.

```
$ roslaunch darwin_control darwin_teleop.launch
```

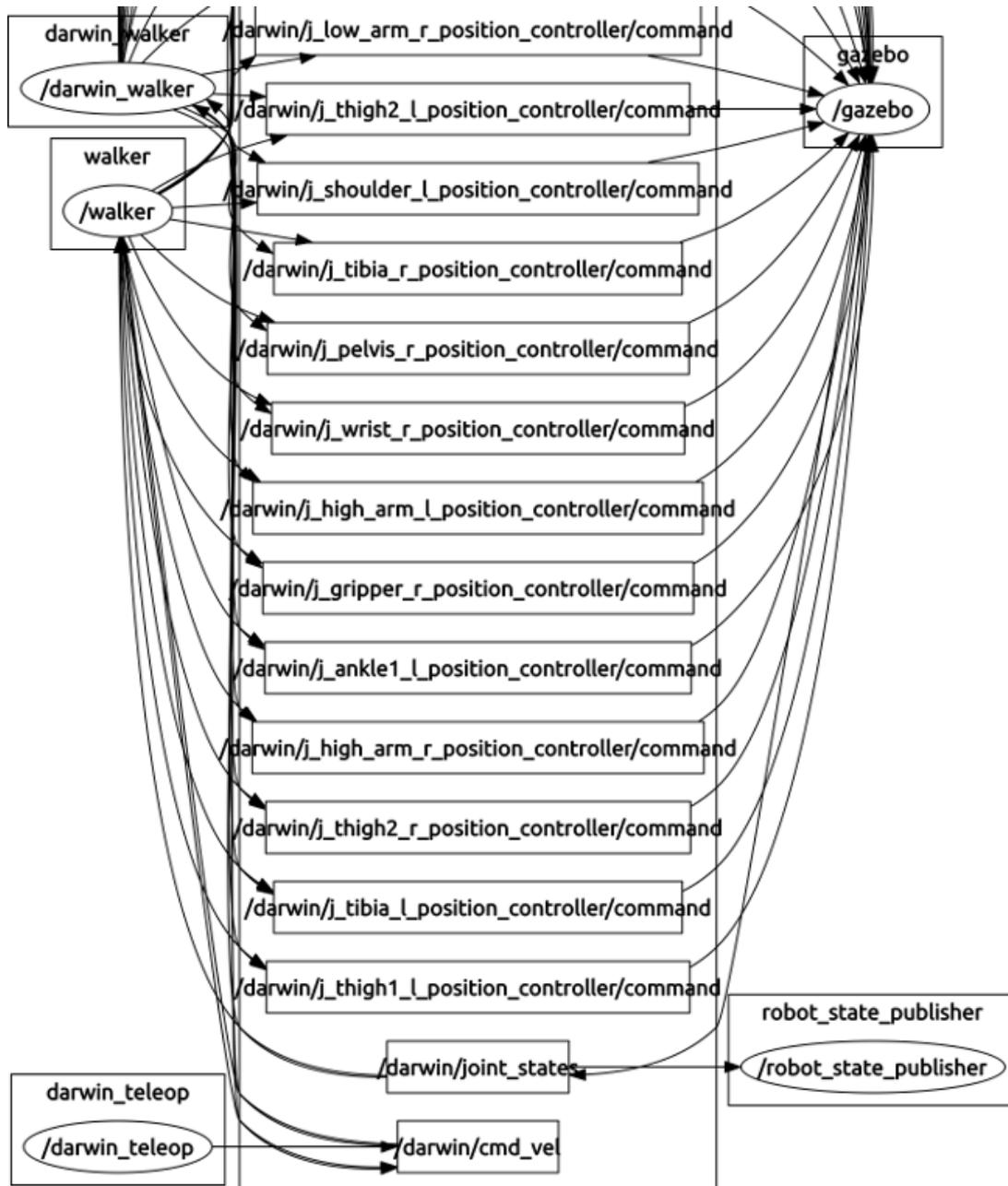


Телеуправление роботом DARwIn-OP в симуляторе Gazebo

Те, кому интересно как выглядит структура нашей системы телеуправления, могут воспользоваться утилитой `rqt_graph` для графического отображения запущенных ROS процессов и каналов связи.

```
$ rosrn rqt_graph rqt_graph
```

Как видите, по сравнению с примером turtlesim, данная система управления гораздо сложнее и имеет больше элементов. Такое количество элементов, обусловлено тем, что модель Darwin-OP имеет 24 виртуальных сервомотора, которыми управляют программы darwin_walker и walker, получая заданную скорость от программы darwin_teleop.



Устройство системы управления роботом DARwIn-OP в симуляторе Gazebo

Подведем итог. В данном уроке мы изучили процесс установки и компиляции ROS Indigo из исходного кода. Сделали мы это с образовательной целью, чтобы немного разобраться с устройством системы ROS и ее команд, а также для того, чтобы подключить последнюю версию симулятора Gazebo 3D. Далее мы загрузили модель человекоподобного робота DARwIn-OP, производимого компанией Robotis, и настроили простую систему управления. Благодаря возможностям ROS наша системы способна работать как на одном, так и на нескольких ПК, объединенных в сеть.

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение
высшего профессионального образования
«Дальневосточный федеральный университет»
(ДВФУ)

ШКОЛА ЦИФРОВОЙ ЭКОНОМИКИ

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ

по дисциплине

ТЕХНОЛОГИИ ВИРТУАЛЬНОЙ И ДОПОЛНЕННОЙ РЕАЛЬНОСТИ В РОБОТОТЕХНИКЕ

Направление подготовки

09.04.01 Информатика и вычислительная техника

магистерская программа

«Технологии виртуальной и дополненной реальности»

Форма подготовки очная

Владивосток

2018

Фонд оценочных средств по дисциплине «Технологии виртуальной и дополненной реальности в робототехнике» включает в себя:

- типовые контрольные задания,
- методические материалы, определяющие процедуры оценивания знаний, умений и навыков и (или) опыта деятельности,
- а также критерии и показатели, необходимые для оценки знаний, умений, навыков и характеризующие этапы формирования компетенций в процессе освоения образовательной программы

Критерии оценивания результатов контрольно-оценочных мероприятий текущей и промежуточной аттестации по дисциплине

Система критериев оценивания, как и при проведении промежуточной аттестации по модулю, опирается на три уровня освоения компонентов компетенций: пороговый, повышенный, высокий.

Результаты обучения (компетенции из ФГОС)	Знает	Умеет	Владеет
ПК - 8 ПК – 13 ПК – 16 ПК – 19	- основные принципы функционирования беспилотных летательных аппаратов (БПЛА); - методы и алгоритмы «компьютерного зрения»; - технологические аспекты социальной робототехники	решать задачи: - навигации, выбора траектории полета, управление группой БПЛА; - обработки изображений и поиска объектов в робототехнике и БПЛА	практическими навыками применения VR и AR в современной робототехнике
Эталонный	Основной и дополнительный материал, предусмотренный компетенциями ПК-8, ПК-13, ПК-16 и ПК-19 без ошибок и погрешностей	Умеет в полном объеме ... решать задачи: - навигации, выбора траектории полета, управление группой БПЛА; - обработки изображений и поиска объектов в робототехнике и БПЛА	всеми навыками применения VR и AR в современной робототехнике, демонстрируя их не только в стандартных ситуациях, но и при решении нестандартных задач
Продвинутый	основной материал, предусмотренный компетенциями ПК-8, ПК-13, ПК-16 и ПК-19,	Умеет с незначительными погрешностями ... решать задачи:	основными навыками применения VR и AR в современной робототехнике,

	без ошибок и погрешностей	- навигации, выбора траектории полета, управление группой БПЛА; - обработки изображений и поиска объектов в робототехнике и БПЛА	демонстрируя их в стандартных ситуациях, в том числе при решении дополнительных задач
Пороговый	большинство основных понятий, изучаемых в рамках дисциплины	Умеет с погрешностями ... решать задачи: - навигации, выбора траектории полета, управление группой БПЛА; - обработки изображений и поиска объектов в робототехнике и БПЛА	некоторыми основными навыками применения VR и AR в современной робототехнике, демонстрируя их в стандартных ситуациях

Критерии выставления оценки студенту на зачете

Порядок начисления рейтинговых баллов по предмету:
выполнение всех лабораторных работ – 100 баллов.

Баллы (рейтинговой оценки)	Оценка экзамена	Требования к сформированным компетенциям
85-100	«отлично»	Оценка «отлично» выставляется студенту, если он глубоко и прочно усвоил программный материал, исчерпывающе, последовательно, четко и логически стройно его излагает, умеет тесно связывать теорию с практикой, свободно справляется с задачами, вопросами и другими видами применения знаний, причем не затрудняется с ответом при видоизменении заданий, использует в ответе материал монографической литературы, владеет разносторонними навыками и приемами выполнения лабораторных работ.
70-84	«хорошо»	Оценка «хорошо» выставляется студенту, если он твердо знает материал, грамотно и по существу излагает его, не допуская существенных неточностей в ответе на вопрос, правильно применяет теоретические положения при решении лабораторных работ вопросов и задач, владеет необходимыми навыками и приемами их выполнения.

50-69	«удовлетворительно»	Оценка «удовлетворительно» выставляется студенту, если он имеет знания только основного материала, но не усвоил его деталей, допускает неточности, недостаточно правильные формулировки, нарушения логической последовательности в изложении программного материала, испытывает затруднения при выполнении лабораторных работ.
0-49	«неудовлетворительно»	Оценка «неудовлетворительно» выставляется студенту, который не знает значительной части программного материала, допускает существенные ошибки, неуверенно, с большими затруднениями выполняет практические работы. Как правило, оценка «неудовлетворительно» ставится студентам, которые не могут продолжить обучение без дополнительных занятий по соответствующей дисциплине.

Правила аттестации для студентов, не набравших необходимый минимум баллов по дисциплине

Если студент, в ходе изучения дисциплины набрал 50 и более баллов, то он имеет право на выставление зачета с соответствующей оценкой без его сдачи.

Если студент набрал менее 50 баллов, то он должен сдавать зачет (выполнение дополнительной лабораторной работы). Данный тест оценивается в диапазоне от 0 до 30 баллов. Полученные баллы суммируются к уже набранным и студенту выставляется итоговая оценка.