Модуль 05 - Robot Operating System

URDF

Резюме: Пора познакомиться с системами визуализации и создания программных двойников - RViz и Gazebo. Задачами для вас будут создание вашего первого робота в ROS и его передвижение.

# Глава I

# Преамбула

Моделирование робота в симуляторе и визуализация его состояния в 3D является важной частью робототехники. Не всегда есть физический доступ к роботу, иногда тестирование разработанных программ на реальном роботе осложнено многими причинами. Поэтому робототехники делают копию робота в симуляторе и разрабатывают для него программы тестируя их в симуляторе. Также вам при проверке работы робота понадобиться визуализировать его состояние и посмотреть что видят его сенсоры в 3D. Все эти задачи в ROS можно решать с помощью симулятора Gazebo и визуализатора RViz.

# 

# Глава II

# Инструкции

• Используйте эту страницу как единственное описание задач. Не слушайте никаких слухов и домыслов о том, как приготовить свой программное решение.

• Здесь и далее мы используем ROS2 Humble и C++/Python.

• Обратите внимание на тип ваших файлов и каталогов.

• Для оценки ваше решение должно находиться в вашем репозитории GitHub.

• Вы не должны оставлять никаких дополнительных файлов в своем каталоге, кроме тех, которые явно указаны в теме. Рекомендуется изменить ваш .gitignore, чтобы избежать конфликтных случаев.

• Когда вам нужно получить точный результат в вашей программе, запрещается отображать предварительно рассчитанный результат вместо правильного выполнения упражнения.

• Есть вопрос? Спросите у соседа справа. В противном случае спросите вашего соседа слева.

• Ваш справочник: коллеги/интернет/google.

• Вы можете задавать вопросы в telegram.

• Внимательно прочитайте примеры. Они вполне могут прояснить детали, которые явно не упомянуты в теме.

Глава III

# Упражнение 01

Упражнение 01: Изучение URDF

Каталог для хранения решения: ex01/

Файлы, которые должны быть в каталоге: robot.urdf, robot\_display.launch

Разрешенные функции:

Комментарии:

В этом упражнении вам нужно будет изучить документацию tf2:

[urdf/XML/link - ROS Wiki](http://wiki.ros.org/urdf/XML/link) (URDF описание из ROS1 для контекста)

[Setting Up Transformations — Nav2 1.0.0 documentation](https://navigation.ros.org/setup_guides/transformation/setup_transforms.html) (концепт расположения link относительно друг друга)

[Setting Up The URDF — Nav2 1.0.0 documentation](https://navigation.ros.org/setup_guides/urdf/setup_urdf.html) (пример робота с дифф. приводом)

[Building a visual robot model from scratch — ROS 2 Documentation](https://docs.ros.org/en/humble/Tutorials/Intermediate/URDF/Building-a-Visual-Robot-Model-with-URDF-from-Scratch.html)

[Building a movable robot model — ROS 2 Documentation: Humble documentation](https://docs.ros.org/en/humble/Tutorials/Intermediate/URDF/Building-a-Movable-Robot-Model-with-URDF.html)

[Adding physical and collision properties — ROS 2 Documentation](https://docs.ros.org/en/humble/Tutorials/Intermediate/URDF/Adding-Physical-and-Collision-Properties-to-a-URDF-Model.html)

Создайте URDF модель робота с дифференциальным приводом. Колеса должны вращаться через GUI Joint state publisher. У робота должны быть установлены physical and collision properties.

Размеры робота должны вписываться в следующие характеристики:

Габариты робота не должны выходить за пределы цилиндра диаметром 210 сантиметров и высотой 250 сантиметров. Форма робота может быть любая.

Также создайте файл запуска с именем robot\_display.launch, который загрузит описание вашего робота и покажет его в RViz вместе с GUI Joint state publisher.

Затем сохраните файл robot.urdf и robot\_display.launch в папку ex01.

# Упражнение 02

Упражнение 02: Использование Xacro для очистки URDF-файла

Каталог для хранения вашего решения: ex02/

Файлы, которые должны быть в каталоге: robot.urdf.xacro, robot\_display.launch

Разрешенные функции:

Комментарии:

В этом упражнении вам нужно будет изучить:

[Using Xacro to clean up your code — ROS 2 Documentation: Humble documentation](https://docs.ros.org/en/humble/Tutorials/Intermediate/URDF/Using-Xacro-to-Clean-Up-a-URDF-File.html)

Переведите URDF файл из предыдущего задания в XACRO.

Ваша модель XACRO должна запускаться через robot\_display.launch, который загрузит описание вашего робота и покажет его в RViz вместе с GUI Joint state publisher.

Затем сохраните файл robot.urdf.xacro и robot\_display.launch в папку ex02.

# Упражнение 03

Упражнение 03: Использование URDF в Gazebo

Каталог для хранения вашего решения: ex03/

Файлы, которые должны быть в каталоге: robot\_gazebo.launch, robot.urdf.xacro, robot.gazebo.xacro

Разрешенные функции:

Комментарии: обратите внимание, что мы используем Gazebo Garden для лучшей совместимости с WSL2

В этом упражнении вам необходимо изучить:

[Gazebo - Docs: ROS 2 Integration](https://gazebosim.org/docs/garden/ros2_integration)

[ros\_gz\_sim\_demos](https://github.com/gazebosim/ros_gz/tree/ros2/ros_gz_sim_demos) (примеры запуска моделей робота Robot description publisher и Diff drive)

[GitHub - gazebosim/ros\_gz\_project\_template: A template project integrating ROS and Gazebo simulator](https://github.com/gazebosim/ros_gz_project_template) (пример создания пакета для взаимодействия с Gazebo)

Создайте ROS пакет запускающий вашего робота с дифференциальным приводом из предыдущего задания в Gazebo.

Ваш робот должен ездить в Gazebo управляемый через панель rqt\_robot\_steering и топик /cmd\_vel с помощью команд ros2 topic pub.

Проверьте управление вашим роботом с помощью клавиатуры через ros2 run teleop\_twist\_keyboard teleop\_twist\_keyboard.py, робот должен двигаться в Gazebo и RViz.

Сохраните все файлы вашего ROS пакета в папку ex03.

# Упражнение 04

Упражнение 04: Программа управления роботом в Gazebo

Каталог для хранения вашего решения: ex04/

Файлы, которые должны быть в каталоге: пакет ROS с роботом, circle\_movement.cpp/.py, circle\_movement.launch

Разрешенные функции:

Комментарии:

Напишите свой ROS пакет с узлом, который должен управлять роботом только через публикацию сообщений в топик /cmd\_vel. В результате робот должен в Gazebo и RViz автономно двигаться по кругу. tf робота должны публиковаться, колеса крутиться, в том числе tf колес.

Сохраните все файлы вашего ROS пакета в папку ex04.

# Упражнение 05

Упражнение 05: Программа управления уникальным роботом в Gazebo

Каталог для хранения вашего решения: ex05/

Файлы, которые должны быть в каталоге: пакет ROS с роботом, \*.cpp/.py, \*.launch

Разрешенные функции:

Комментарии:

Доработайте робота из предыдущего задания так чтобы у него был уникальный дизайн, геометрическая форма, придумайте программу уникальных движений робота. Уникальный значит что не должен повторяться или быть похожим на работы ваших коллег.

Сохраните все файлы вашего ROS пакета в папку ex05.