

# Модуль 04 - Robot Operating System

## Преобразование системы координат

Резюме: Библиотека tf была разработана для обеспечения стандартного способа отслеживания систем координат и преобразования данных в координатах всей системы, чтобы пользователи отдельных компонентов могли быть уверены, что данные находятся в нужной им системе координат, не требуя знания всех систем координат в системе.

# Глава I

## Преамбула

Во время ранней разработки Robot Operating System (ROS) отслеживание систем координат было определено как общая больная точка для разработчиков. Сложность этой задачи делала ее общим местом для ошибок, когда разработчики неправильно применяли преобразования к данным. Проблема также является сложной из-за часто распределенных источников информации о преобразованиях между различными наборами систем координат.

Роботизированная система обычно имеет множество трехмерных систем координат которые изменяются во времени, например, мировая система координат, базовая система координат, система координат захвата, система координат головы и т.д. tf2 отслеживает все эти системы координат во времени и позволяет задавать такие вопросы, как:

- Где находилась система координат головы относительно мировой системы координат 5 секунд назад?
- Какова позиция объекта в моем захвате относительно базы?
- Какова текущая позиция базовой системы координат в системе координат карты?

# 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: Введение в tf2

Каталог для хранения решения: ex01/

Файлы, которые должны быть в каталоге: frames.pdf, transform.txt, last\_transform.txt

Разрешенные функции:

Комментарии:

В этом упражнении вам нужно будет изучить документацию tf2:

<https://docs.ros.org/en/humble/Tutorials/Intermediate/Tf2/Tf2-Main.html>

<https://docs.ros.org/en/humble/Tutorials/Intermediate/Tf2/Introduction-To-Tf2.html>

В первом терминале запустите turtle\_tf2\_demo.launch.

Во втором терминале с помощью утилиты tf2\_tools сохраните дерево tf в файл frames.pdf.

Используя утилиту tf2\_echo сохраните вывод результатов трансформации координат между фреймами turtle1 и turtle2 в файл transform.txt за несколько секунд. При этом передвиньте черепаху с помощью клавиатуры.

Откройте текстовым редактором файл transform.txt, посмотрите последнее преобразование координат между фреймами turtle1 и turtle2 по времени и впишите руками в файл last\_transform.txt три числа через запятую: смещение по оси X, смещение по оси Y, угол поворота в градусах(!) по оси Z.

Затем сохраните файлы frames.pdf, transform.txt, last\_transform.txt в папку ex01.

## Упражнение 02

Упражнение 02: Написание вещателя и слушателя tf2

Каталог для хранения вашего решения: ex02/

Файлы, которые должны быть в каталоге: Все файлы ROS пакета и carrot.rviz

Разрешенные функции:

Комментарии:

В этом упражнении вам нужно будет изучить:

<https://docs.ros.org/en/humble/Tutorials/Intermediate/Tf2/Writing-A-Tf2-Static-Broadcaster-Py.html>

(Python)

<https://docs.ros.org/en/humble/Tutorials/Intermediate/Tf2/Writing-A-Tf2-Static-Broadcaster-Cpp.html>

(C++)

<https://docs.ros.org/en/humble/Tutorials/Intermediate/Tf2/Writing-A-Tf2-Broadcaster-Py.html> (Python)

<https://docs.ros.org/en/humble/Tutorials/Intermediate/Tf2/Writing-A-Tf2-Broadcaster-Cpp.html> (C++)

<https://docs.ros.org/en/humble/Tutorials/Intermediate/Tf2/Writing-A-Tf2-Listener-Py.html> (Python)

<https://docs.ros.org/en/humble/Tutorials/Intermediate/Tf2/Writing-A-Tf2-Listener-Cpp.html> (C++)

<https://docs.ros.org/en/humble/Tutorials/Intermediate/Tf2/Adding-A-Frame-Py.html> (Python)

<https://docs.ros.org/en/humble/Tutorials/Intermediate/Tf2/Adding-A-Frame-Cpp.html> (C++)

Разработать ROS пакет который будет запускать двух черепах в turtlesim, делать их tf2 фреймы, к первой черепахе добавить фрейм морковки. Морковка должна вращаться вокруг первой черепахи на заданное в параметрах launch файла расстояние (назовите аргумент "radius") и направление по часовой или против часовой стрелки (назовите аргумент "direction\_of\_rotation", он должен принимать только два значения "1" - если по часовой стрелке или "-1" против часовой). Вторая черепаха должна следовать самостоятельно за морковкой. У первой черепахи должна быть возможность управлять с клавиатуры через turtle\_teleop\_key.

Создайте файл настроек carrot.rviz, так чтобы в rviz2 было видно фреймы world, turtle1, turtle2, carrot.

Сохраните все файлы вашего ROS пакета в папку ex02.

## Упражнение 03

Упражнение 03: Путешествие во времени с помощью tf2

Каталог для хранения вашего решения: ex03/

Файлы, которые должны быть в каталоге:

Разрешенные функции:

Комментарии:

В этом упражнении вам необходимо изучить:

<https://docs.ros.org/en/humble/Tutorials/Intermediate/Tf2/Learning-About-Tf2-And-Time-Py.html>

(Python)

<https://docs.ros.org/en/humble/Tutorials/Intermediate/Tf2/Learning-About-Tf2-And-Time-Cpp.html>

(C++)

<https://docs.ros.org/en/humble/Tutorials/Intermediate/Tf2/Time-Travel-With-Tf2-Py.html> (Python)

<https://docs.ros.org/en/humble/Tutorials/Intermediate/Tf2/Time-Travel-With-Tf2-Cpp.html> (C++)

Напишите ROS пакет который запустит в симуляторе двух черепах. Сделайте так чтобы вторая черепаха следовала за местом где была первая черепаха некоторое время назад. Сделайте так чтобы в аргументах launch файла можно было задать задержку во времени (аргумент назовите “delay” с мерой измерения в секундах).

Сохраните все файлы вашего ROS пакета в папку ex03.