Robotics &
Internet-of-Things

ROS

Udemy Course
Robot Operating System (Part II)
Navigation and SLAM

Prof. Anis Koubaa

TF Package in ROS

https://www.udemy.com/user/anis-koubaa/

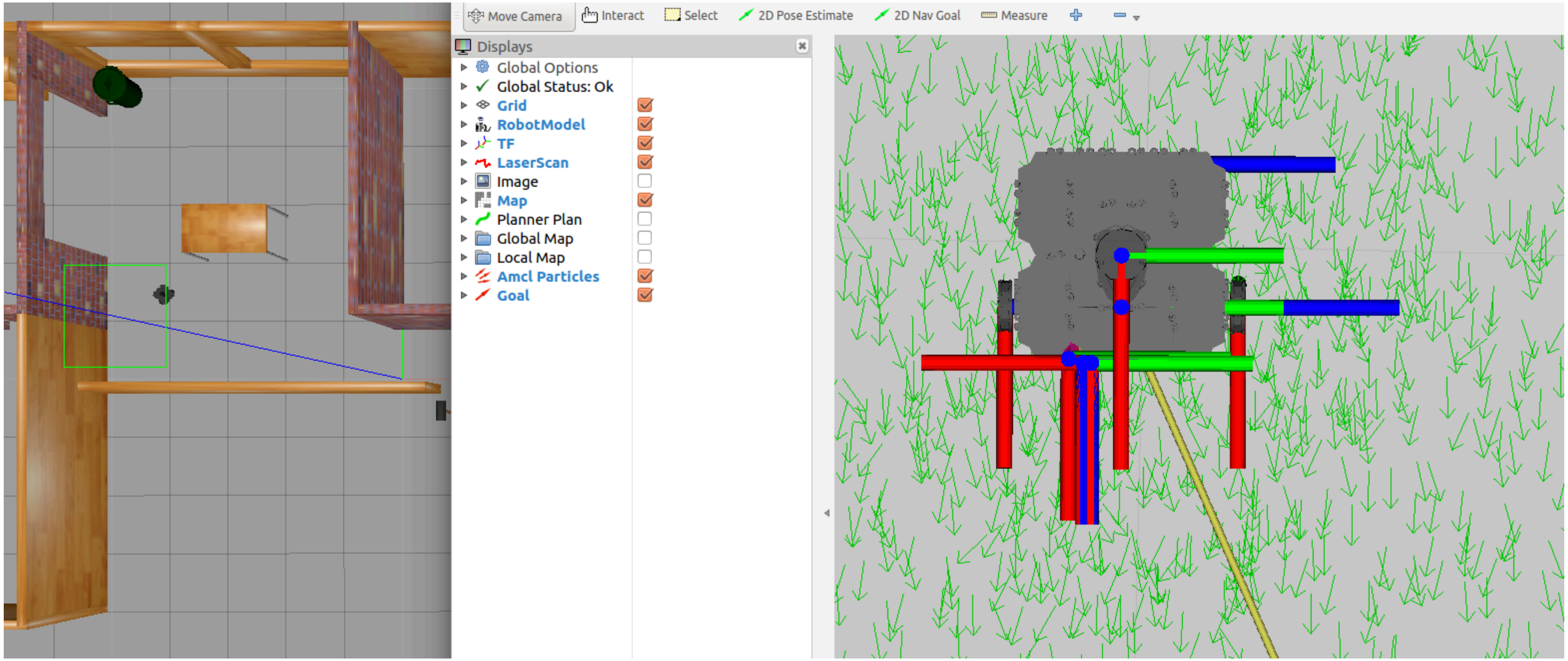
LEARNING OUTCOMES

- In this section, you will:
  - Understand the importance of the TF package
  - Manipulate frames in ROS
  - Perform transformations using tf package

WHAT IS TF?

- TF stands for transformation library in ROS
- It performs computation for transformations between frames.
- It allows to find the pose of any object in any frame using transformations
- A robot is a collection of frames attached to its different joints

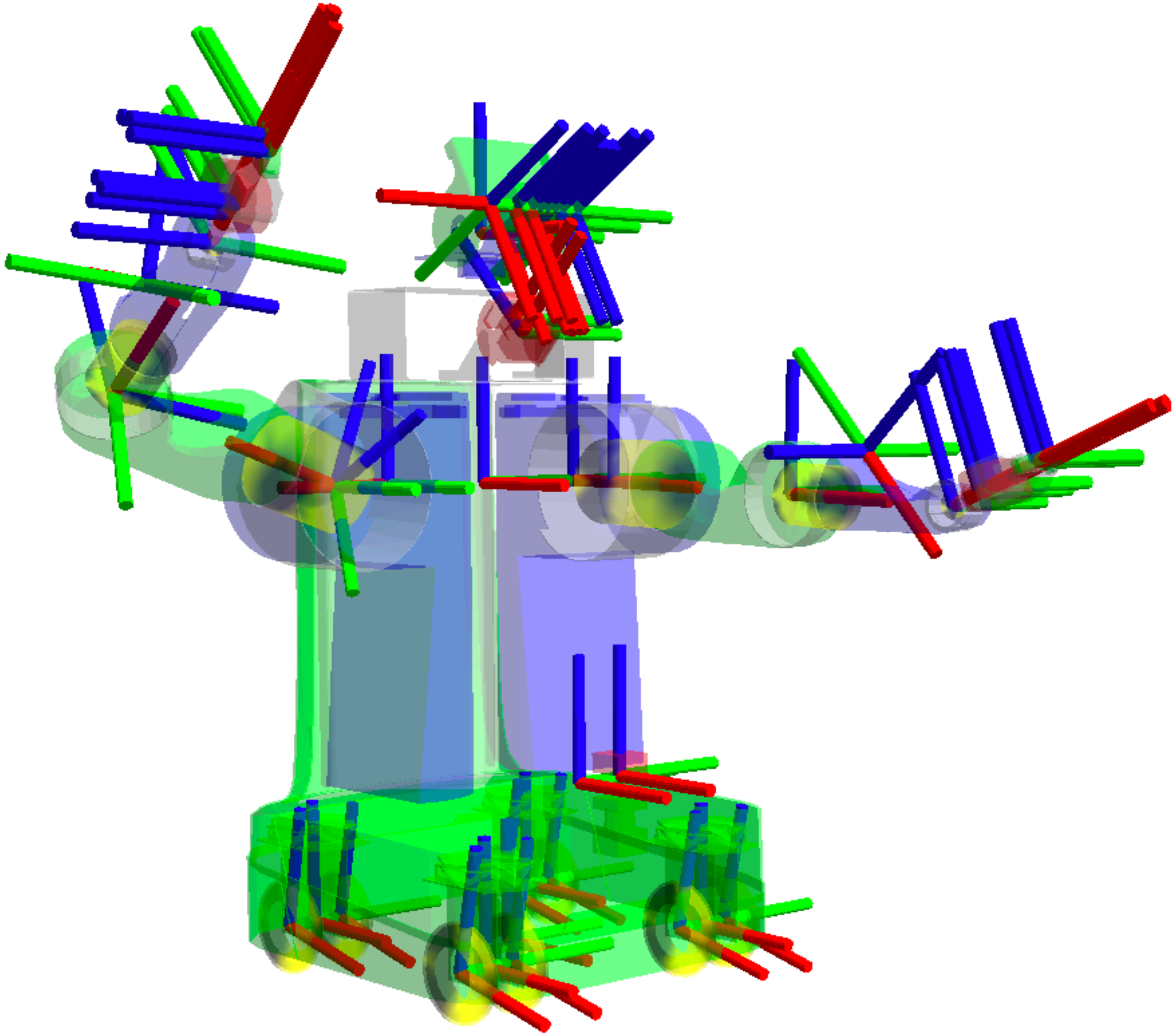
How a robot looks like?



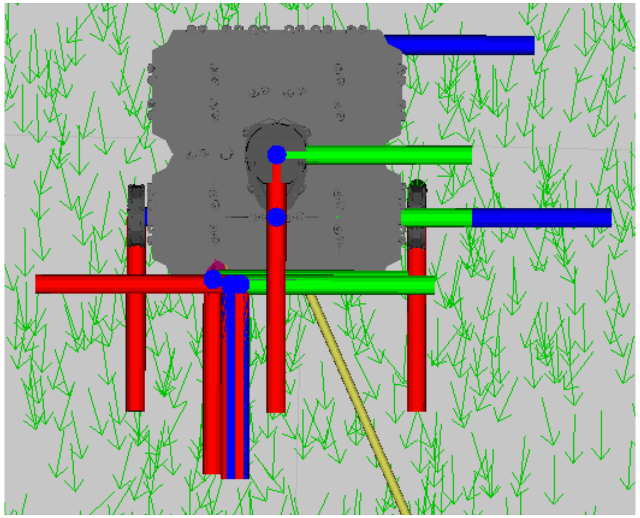
Displays

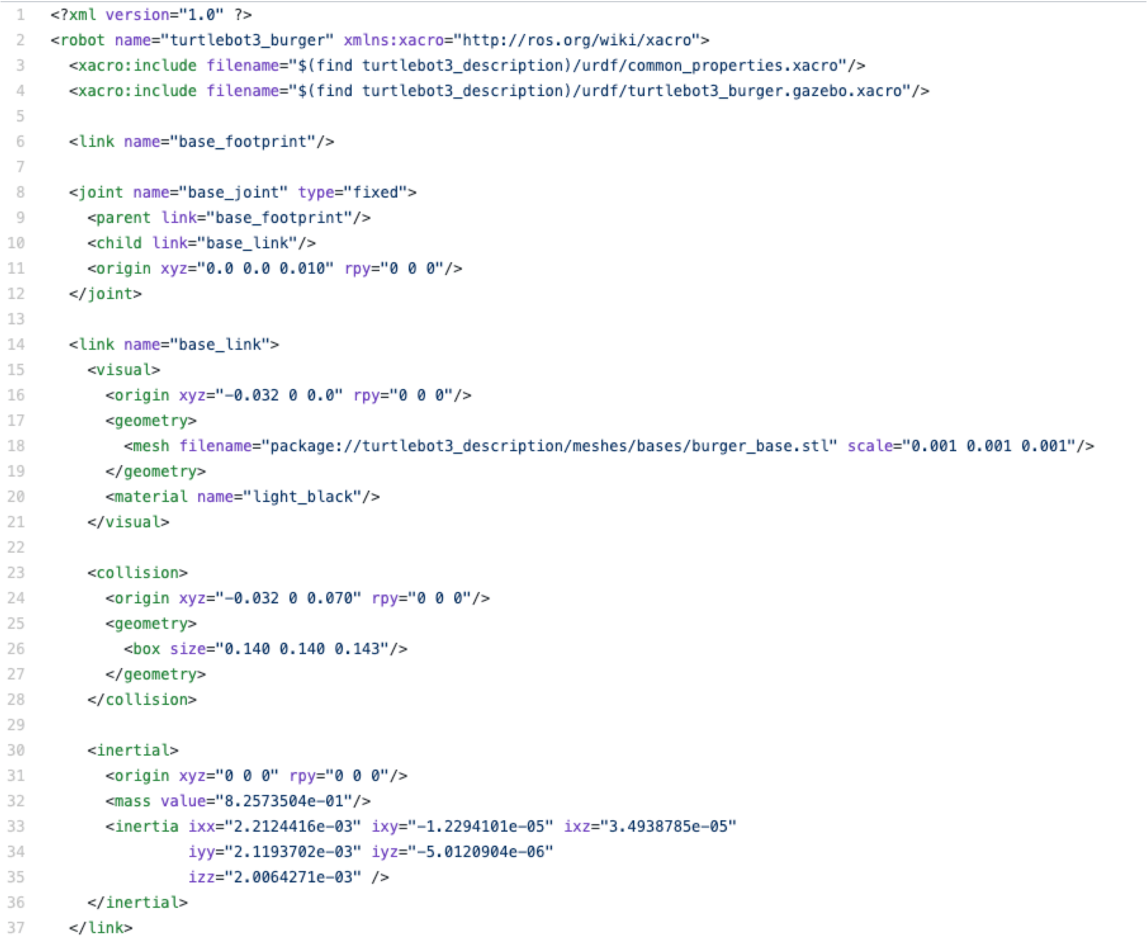
- Global Options
- Global Status: Ok
- Grid ☑
- RobotModel ☑
- TF ☑
- LaserScan ☑
- Image □
- Map ☑
- Planner Plan □
- Global Map □
- Local Map □
- Amcl Particles ☑
- Goal ☑

How a robot looks like?



URDF: Robot Description Language.

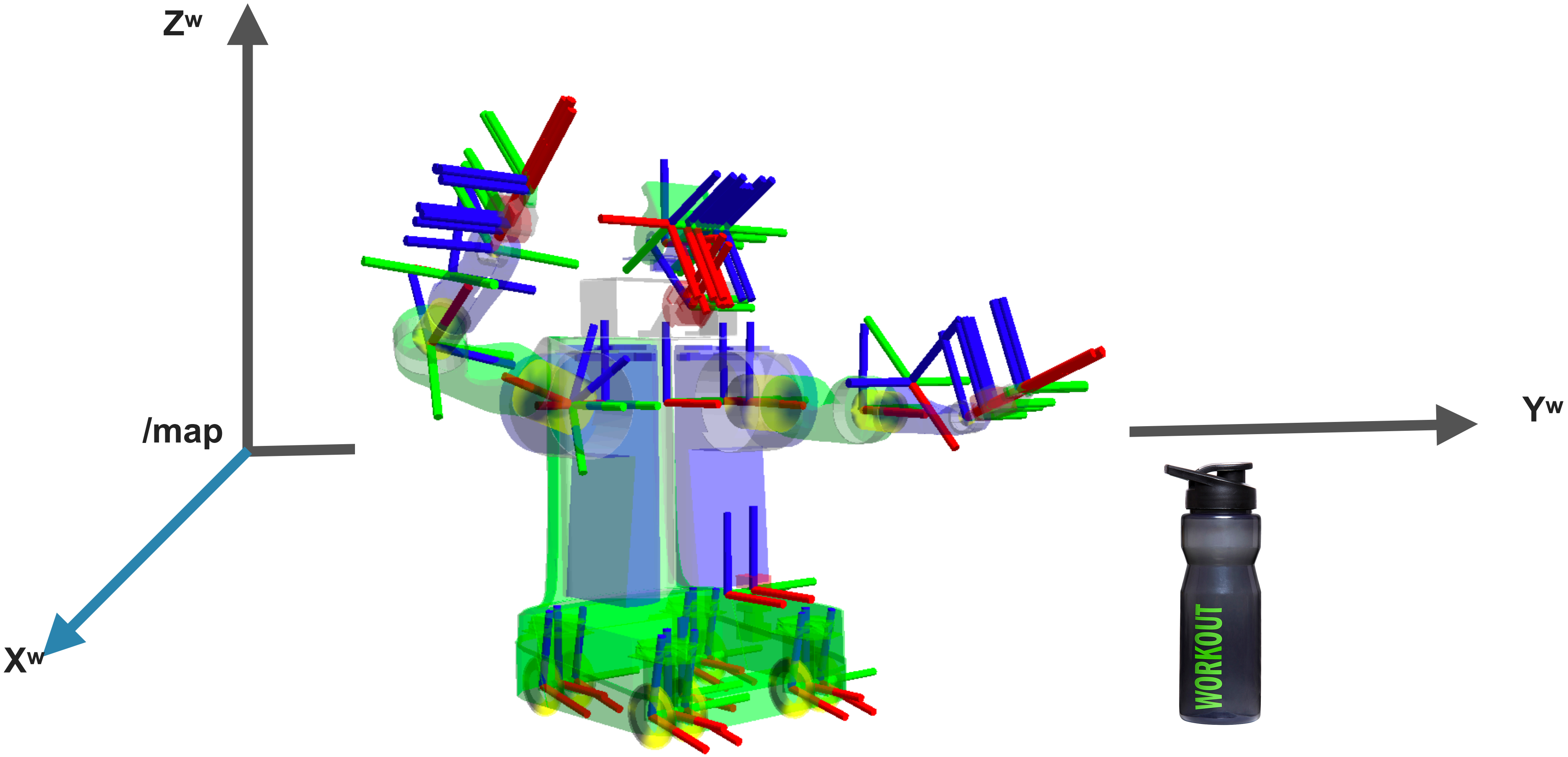




```
<?xml version="1.0" ?>
<robot name="turtlebot3_burger" xmlns:xacro="http://ros.org/wiki/xacro">
  <xacro:include filename="$(find turtlebot3_description)/urdf/common_properties.xacro"/>
  <xacro:include filename="$(find turtlebot3_description)/urdf/turtlebot3_burger.gazebo.xacro"/>

  <link name="base_footprint"/>

  <joint name="base_joint" type="fixed">
    <parent link="base_footprint"/>
    <child link="base_link"/>
    <origin xyz="0.0 0.0 0.010" rpy="0 0 0"/>
  </joint>

  <link name="base_link">
    <visual>
      <origin xyz="-0.032 0 0.0" rpy="0 0 0"/>
      <geometry>
        <mesh filename="package://turtlebot3_description/meshes/bases/burger_base.stl" scale="0.001 0.001 0.001"/>
      </geometry>
      <material name="light_black"/>
    </visual>

    <collision>
      <origin xyz="-0.032 0 0.070" rpy="0 0 0"/>
      <geometry>
        <box size="0.140 0.140 0.143"/>
      </geometry>
    </collision>

    <inertial>
      <origin xyz="0 0 0" rpy="0 0 0"/>
      <mass value="8.2573504e-01"/>
      <inertia ixx="2.2124416e-03" ixy="-1.2294101e-05" ixz="3.4938785e-05"
               iyy="2.1193702e-03" iyz="-5.0120904e-06"
               izz="2.0064271e-03" />
    </inertial>
  </link>
```

http://wiki.ros.org/urdf/Tutorials

WHY TF?



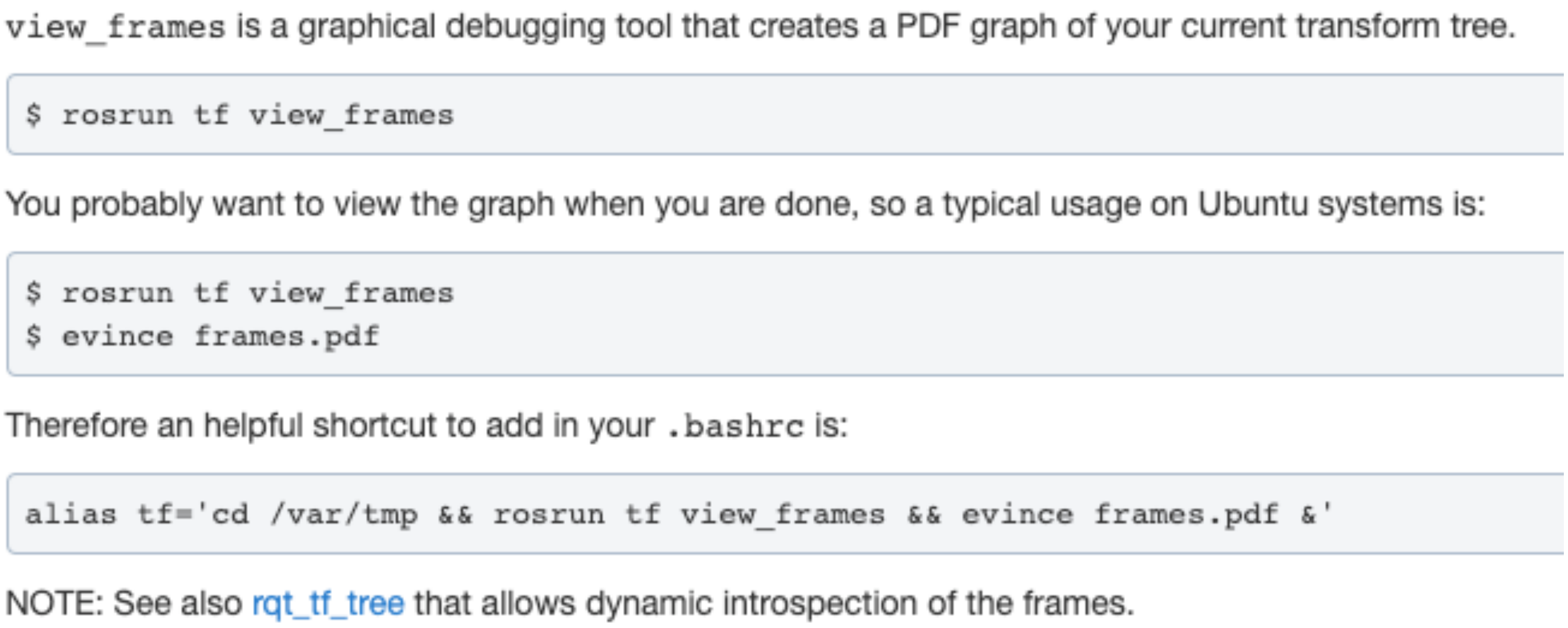
BENEFITS OF TF?

- Performs transformation easily
- The user does not need to worry about frames
- Provides built-in functions to publish and listen to frames in ROS

TF Package Nodes

- The TF Package has several ROS nodes that provide utilities to manipulate frames and transformations in ROS
  - **view_frames:** visualizes the full tree of coordinate transforms.
  - **tf_monitor:** monitors transforms between frames.
  - **tf_echo:** prints specified transform to screen
  - **roswtf:** with the tfwtf plugin, helps you track down problems with tf.
  - **static_transform_publisher** is a command line tool for sending static transforms

View frames (valid before ROS Noetic)



`view_frames` is a graphical debugging tool that creates a PDF graph of your current transform tree.

```
$ rosrun tf view_frames
```

You probably want to view the graph when you are done, so a typical usage on Ubuntu systems is:

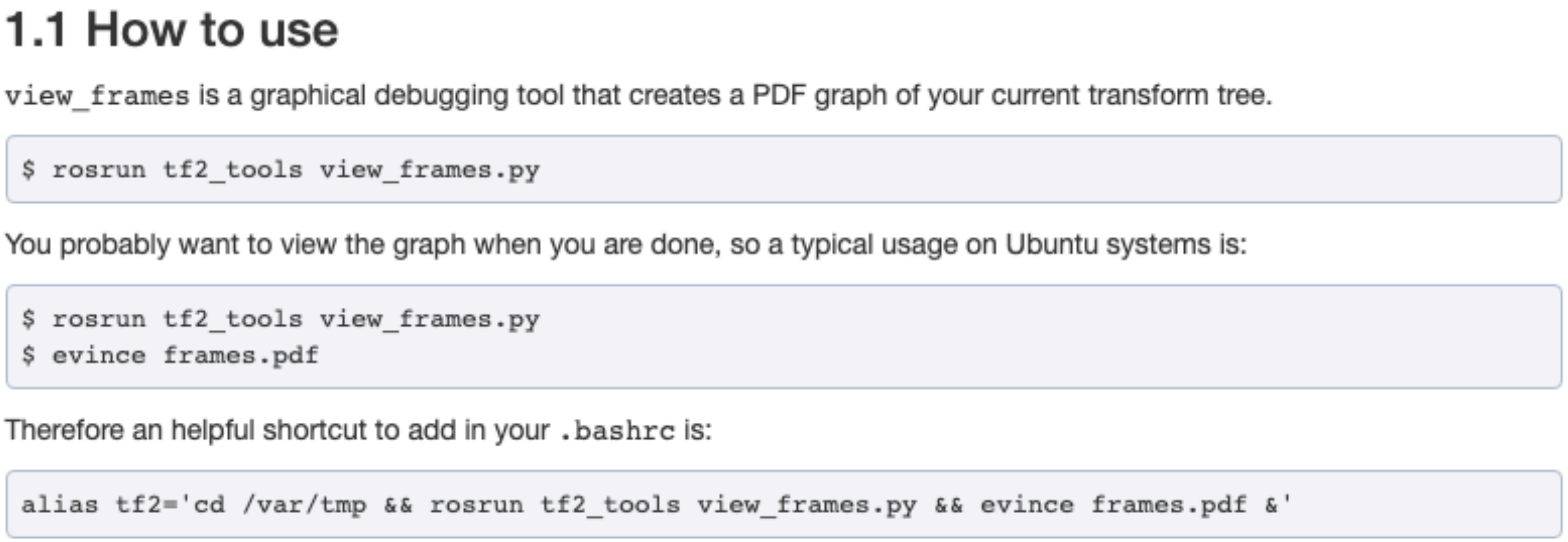
```
$ rosrun tf view_frames
$ evince frames.pdf
```

Therefore an helpful shortcut to add in your `.bashrc` is:

```
alias tf='cd /var/tmp && rosrun tf view_frames && evince frames.pdf &'
```

NOTE: See also rqt_tf_tree that allows dynamic introspection of the frames.

View frames (For ROS Noetic)



1.1 How to use

`view_frames` is a graphical debugging tool that creates a PDF graph of your current transform tree.

```
$ rosrun tf2_tools view_frames.py
```

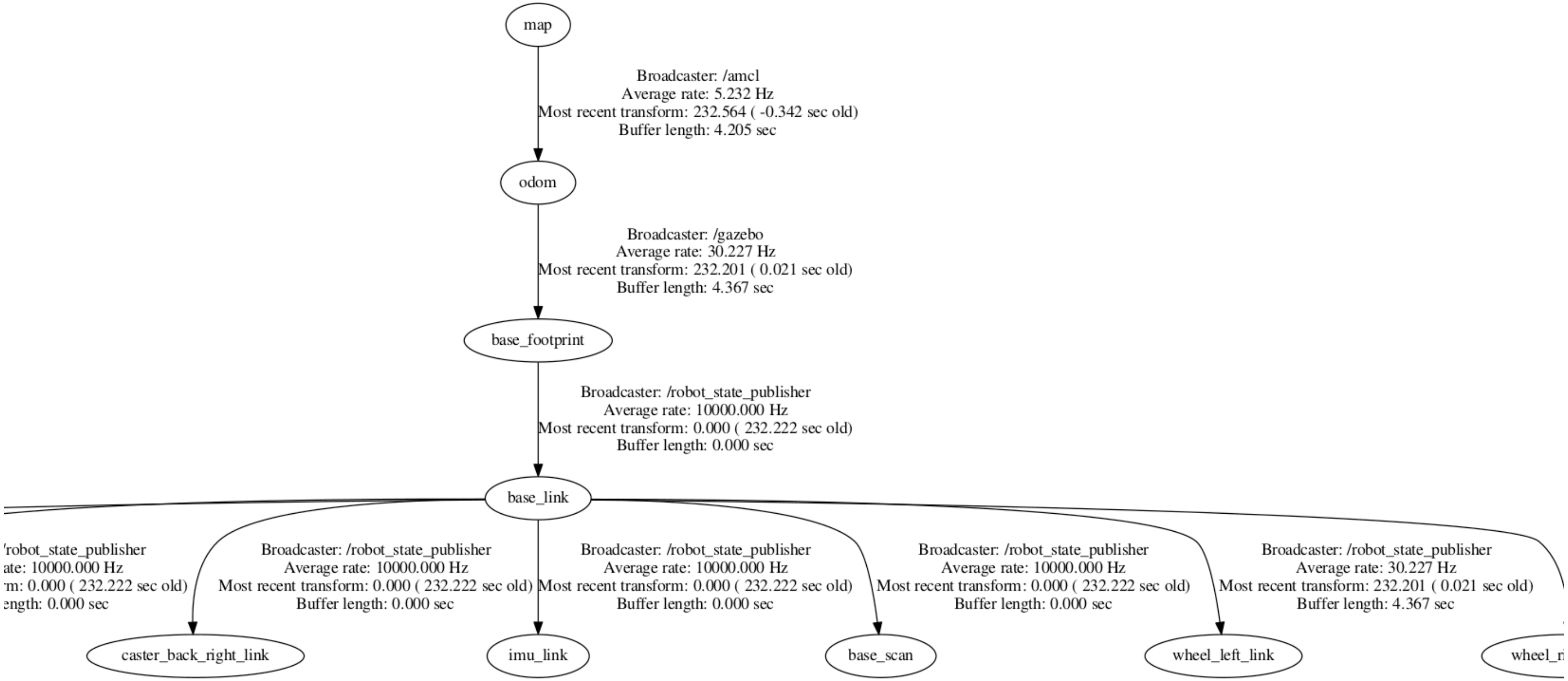
You probably want to view the graph when you are done, so a typical usage on Ubuntu systems is:

```
$ rosrun tf2_tools view_frames.py
$ evince frames.pdf
```

Therefore an helpful shortcut to add in your `.bashrc` is:

```
alias tf2='cd /var/tmp && rosrun tf2_tools view_frames.py && evince frames.pdf &'
```

VIEW FRAMES



map

Broadcaster: /amcl
Average rate: 5.232 Hz
Most recent transform: 232.564 (-0.342 sec old)
Buffer length: 4.205 sec

odom

Broadcaster: /gazebo
Average rate: 30.227 Hz
Most recent transform: 232.201 (0.021 sec old)
Buffer length: 4.367 sec

base_footprint

Broadcaster: /robot_state_publisher
Average rate: 10000.000 Hz
Most recent transform: 0.000 (232.222 sec old)
Buffer length: 0.000 sec

base_link

/robot_state_publisher
ate: 10000.000 Hz
rm: 0.000 (232.222 sec old)
ength: 0.000 sec

Broadcaster: /robot_state_publisher
Average rate: 10000.000 Hz
Most recent transform: 0.000 (232.222 sec old)
Buffer length: 0.000 sec

caster_back_right_link

Broadcaster: /robot_state_publisher
Average rate: 10000.000 Hz
Most recent transform: 0.000 (232.222 sec old)
Buffer length: 0.000 sec

imu_link

Broadcaster: /robot_state_publisher
Average rate: 10000.000 Hz
Most recent transform: 0.000 (232.222 sec old)
Buffer length: 0.000 sec

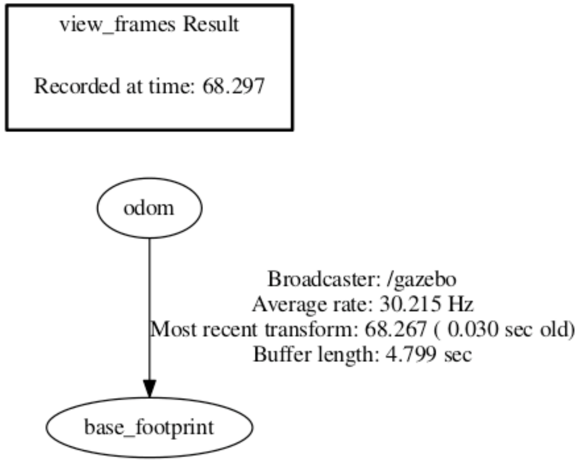
base_scan

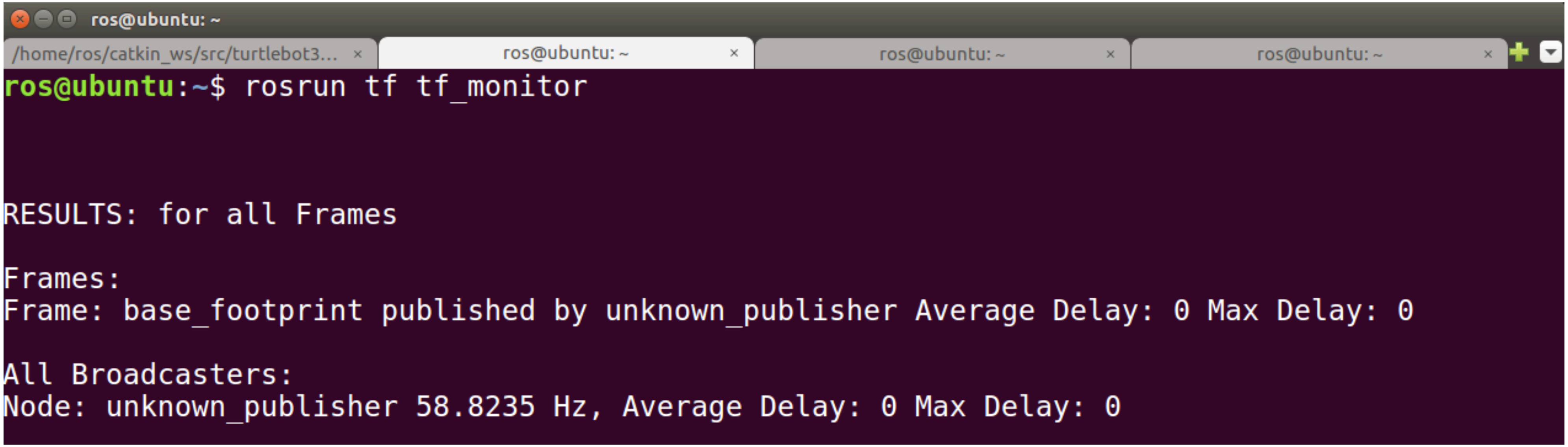
Broadcaster: /robot_state_publisher
Average rate: 30.227 Hz
Most recent transform: 232.201 (0.021 sec old)
Buffer length: 4.367 sec

wheel_left_link

wheel_r

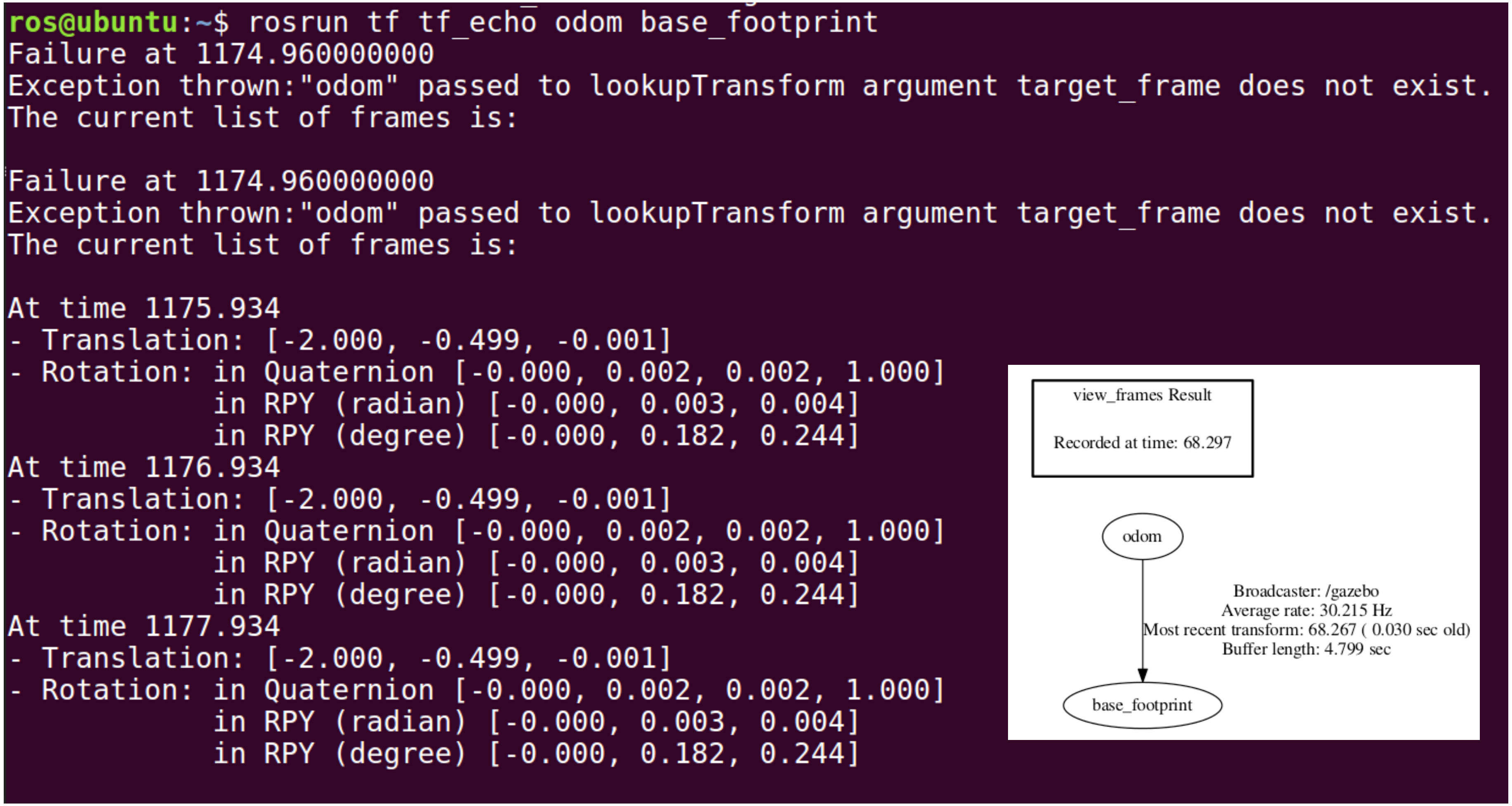
TF Monitor





```
ros@ubuntu:~$ rosrun tf tf_monitor



RESULTS: for all Frames

Frames:
Frame: base_footprint published by unknown_publisher Average Delay: 0 Max Delay: 0

All Broadcasters:
Node: unknown_publisher 58.8235 Hz, Average Delay: 0 Max Delay: 0
```

TF ECHO



```
ros@ubuntu:~$ rosrun tf tf_echo odom base_footprint
Failure at 1174.960000000
Exception thrown:"odom" passed to lookupTransform argument target_frame does not exist.
The current list of frames is:

Failure at 1174.960000000
Exception thrown:"odom" passed to lookupTransform argument target_frame does not exist.
The current list of frames is:

At time 1175.934
- Translation: [-2.000, -0.499, -0.001]
- Rotation: in Quaternion [-0.000, 0.002, 0.002, 1.000]
            in RPY (radian) [-0.000, 0.003, 0.004]
            in RPY (degree) [-0.000, 0.182, 0.244]
At time 1176.934
- Translation: [-2.000, -0.499, -0.001]
- Rotation: in Quaternion [-0.000, 0.002, 0.002, 1.000]
            in RPY (radian) [-0.000, 0.003, 0.004]
            in RPY (degree) [-0.000, 0.182, 0.244]
At time 1177.934
- Translation: [-2.000, -0.499, -0.001]
- Rotation: in Quaternion [-0.000, 0.002, 0.002, 1.000]
            in RPY (radian) [-0.000, 0.003, 0.004]
            in RPY (degree) [-0.000, 0.182, 0.244]
```

TF Package Nodes

- In TF, the frames can be either:
  - Published by a broadcaster node
  - Subscribed by a ROS node that listen to the frames.

Robotics &
Internet-of-Things

ROS

Udemy Course
Robot Operating System (Part II)
Navigation and SLAM

Prof. Anis Koubaa

Conversion of Orientations with ROS tf package

https://www.udemy.com/user/anis-koubaa/

Conversion of Orientation

- Code Explaining Orientation Conversion

Robotics &
Internet-of-Things

ROS

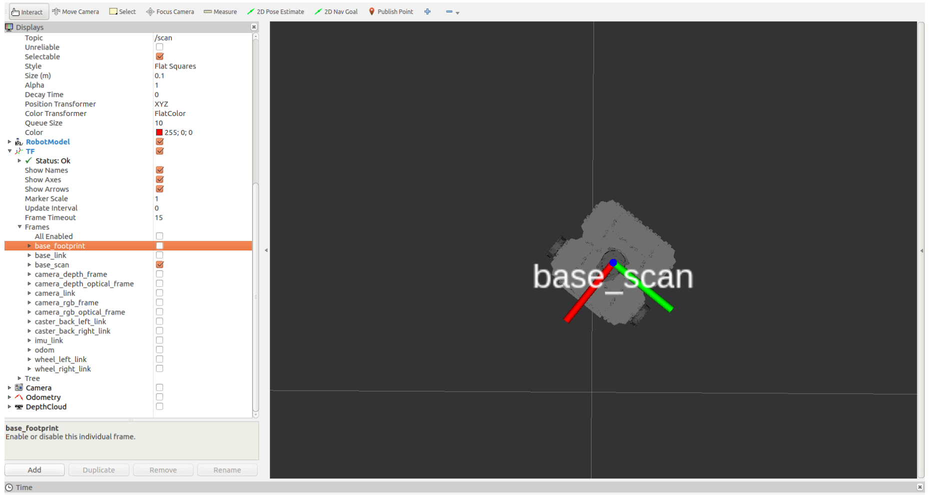
UDEMY COURSE
ROBOT OPERATING SYSTEM
(PART II)
NAVIGATION AND SLAM

PROF. ANIS KOUBAA

Reading the Orientation of a Robot from its Pose

https://www.udemy.com/user/anis-koubaa/

TURTLEBOT3 ROTATION



Topic	/scan
Unreliable	☐
Selectable	☑
Style	Flat Squares
Size (m)	0.1
Alpha	1
Decay Time	0
Position Transformer	XYZ
Color Transformer	FlatColor
Queue Size	10
Color	🟥 255; 0; 0
▸ RobotModel	☑
▾ TF	☑
▸ ✓ Status: Ok	
Show Names	☑
Show Axes	☑
Show Arrows	☑
Marker Scale	1
Update Interval	0
Frame Timeout	15
▾ Frames	
All Enabled	☐
▸ base_footprint	☐
▸ base_link	☐
▸ base_scan	☑
▸ camera_depth_frame	☐
▸ camera_depth_optical_frame	☐
▸ camera_link	☐
▸ camera_rgb_frame	☐
▸ camera_rgb_optical_frame	☐
▸ caster_back_left_link	☐
▸ caster_back_right_link	☐
▸ imu_link	☐
▸ odom	☐
▸ wheel_left_link	☐
▸ wheel_right_link	☐
▸ Tree	
▸ Camera	☐
▸ Odometry	☐
▸ DepthCloud	☐

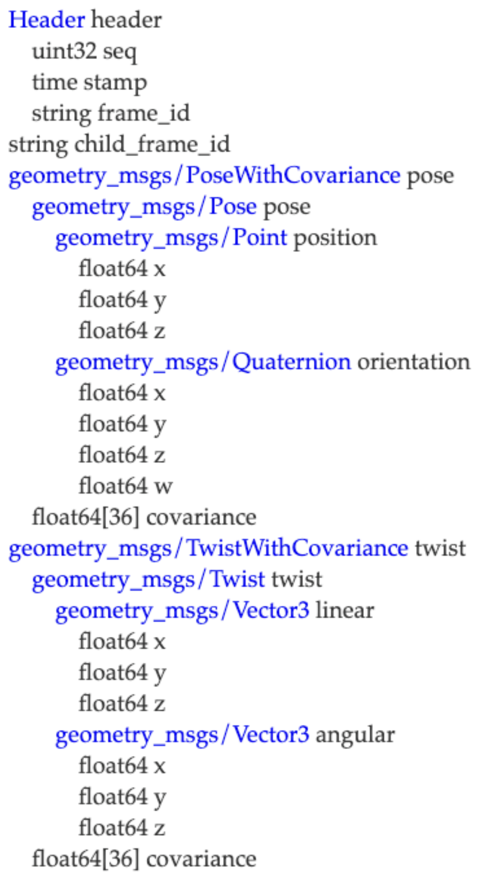
base_footprint
Enable or disable this individual frame.

Add | Duplicate | Remove | Rename

- Start Turtlebot3 Waffle
- List all topics
- Check info of /odom topic and amcl_pose topic
- Understand how position and orientation are presented
- Write a script that print the x,y coordinate and yaw angle of the Turtlebot3 robot.

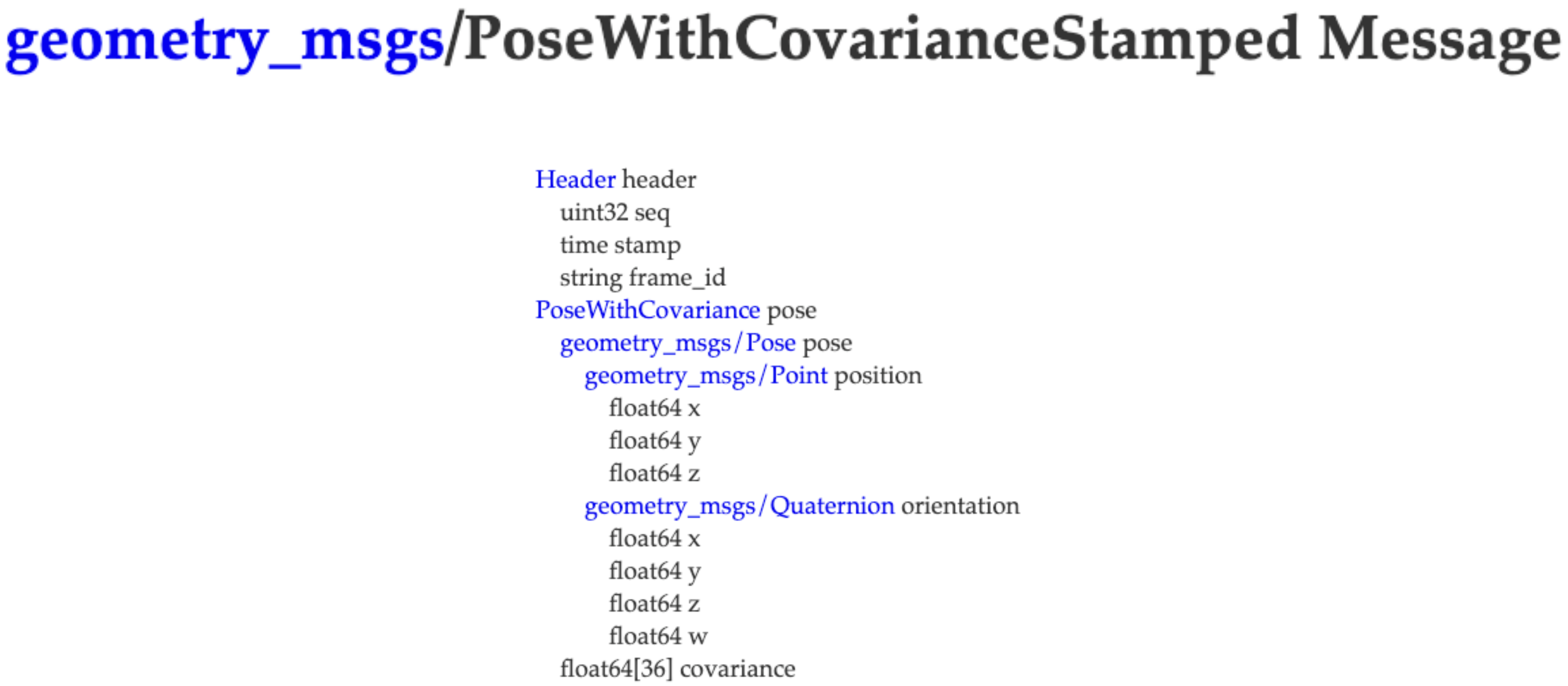
ODOM POSITION MESSAGE FORMAT

nav_msgs/Odometry Message



```
Header header
  uint32 seq
  time stamp
  string frame_id
string child_frame_id
geometry_msgs/PoseWithCovariance pose
  geometry_msgs/Pose pose
    geometry_msgs/Point position
      float64 x
      float64 y
      float64 z
    geometry_msgs/Quaternion orientation
      float64 x
      float64 y
      float64 z
      float64 w
  float64[36] covariance
geometry_msgs/TwistWithCovariance twist
  geometry_msgs/Twist twist
    geometry_msgs/Vector3 linear
      float64 x
      float64 y
      float64 z
    geometry_msgs/Vector3 angular
      float64 x
      float64 y
      float64 z
  float64[36] covariance
```

AMCL_POSE POSITION MESSAGE FORMAT



geometry_msgs/PoseWithCovarianceStamped Message

```
Header header
  uint32 seq
  time stamp
  string frame_id
PoseWithCovariance pose
  geometry_msgs/Pose pose
    geometry_msgs/Point position
      float64 x
      float64 y
      float64 z
    geometry_msgs/Quaternion orientation
      float64 x
      float64 y
      float64 z
      float64 w
  float64[36] covariance
```

Robotics &
Internet-of-Things

ROS

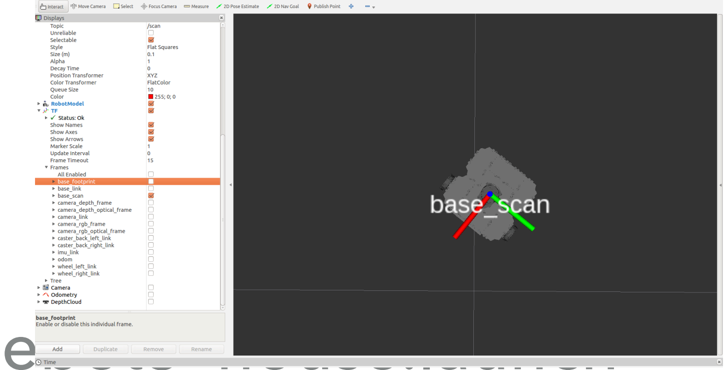
Udemy Course
Robot Operating System
(Part II)
Navigation and SLAM

Prof. Anis Koubaa

The tf package command line and utilities

https://www.udemy.com/user/anis-koubaa/

TURTLEBOT3 ROTATION



Topic	/scan
Unreliable	□
Selectable	☑
Style	Flat Squares
Size (m)	0.1
Alpha	1
Decay Time	0
Position Transformer	XYZ
Color Transformer	FlatColor
Queue Size	10
Color	🟥 255; 0; 0
RobotModel	☑
TF	☑
Status: Ok	
Show Names	☑
Show Axes	☑
Show Arrows	☑
Marker Scale	1
Update Interval	0
Frame Timeout	15
Frames	
All Enabled	□
base_footprint	□
base_link	□
base_scan	☑
camera_depth_frame	□
camera_depth_optical_frame	□
camera_link	□
camera_rgb_frame	□
camera_rgb_optical_frame	□
caster_back_left_link	□
caster_back_right_link	□
imu_link	□
odom	□
wheel_left_link	□
wheel_right_link	□
Tree	
Camera	□
Odometry	□
DepthCloud	□

base_footprint
Enable or disable this individual frame.

Add | Duplicate | Remove | Rename

- Start Turtlebot3 Waffle
  - roslaunch turtlebot3_gazebo turtle
- open rviz
  - roslaunch turtlebot3_gazebo turtlebot3_gazebo_rviz.launch
- show all frames
- Monitor frames
- Echo the frames

/http://emanual.robotis.com/docs/en/platform/turtlebot3/simulation

Robotics & Internet-of-Things

ROS

Udemy Course
Robot Operating System (Part II)
Navigation and SLAM

Prof. Anis Koubaa

Broadcast a transformation in a ROS Node

https://www.udemy.com/user/anis-koubaa/

Robotics &
Internet-of-Things

ROS

Udemy Course
Robot Operating System (Part II)
Navigation and SLAM

Prof. Anis Koubaa

Listen to a transformation in a ROS Node

https://www.udemy.com/user/anis-koubaa/

STATIC TRANSFORM PUBLISHER

```
static_transform_publisher x y z yaw pitch roll
frame_id child_frame_id period_in_ms
```

- Publish a static coordinate transform to tf using an x/y/z offset in meters and yaw/pitch/roll in radians. (yaw is rotation about Z, pitch is rotation about Y, and roll is rotation about X).

- The period, in milliseconds, specifies how often to send a transform. 100ms (10hz) is a good value.

```
rosrun tf static_transform_publisher 1.0 2.1 3.2 0.1 0.11 0.23 odom map 10
```

Static Transform Publisher: Launch Fine

```
<launch>

 <node pkg="tf" type="static_transform_publisher"
name=“odom_to_map”
args="1.0 0 -2 0.1 0.2 0.3 odom map 10" />

</launch>
```

Robotics &
Internet-of-Things

ROS

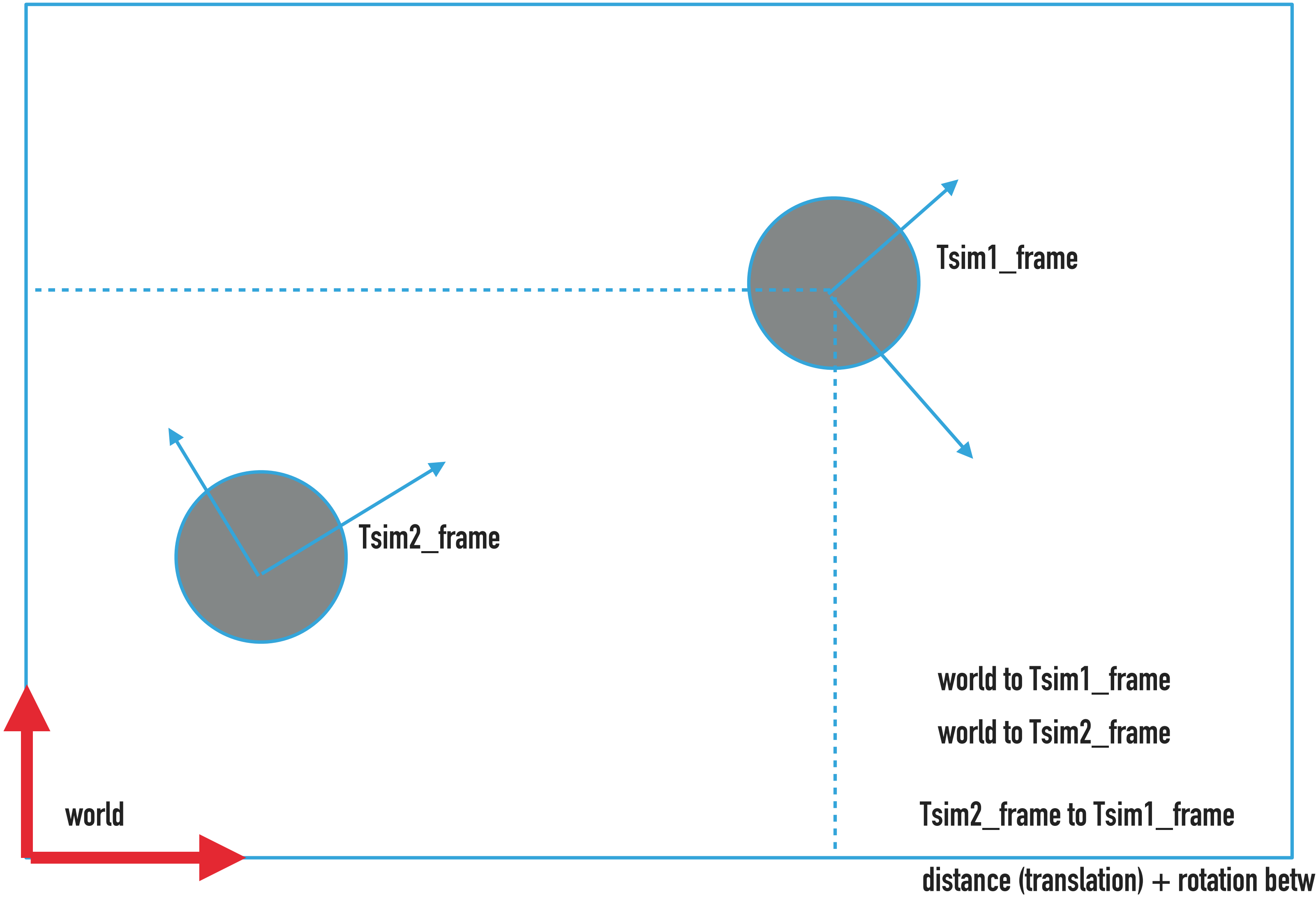
Udemy Course
Robot Operating System (Part II)
Navigation and SLAM

Prof. Anis Koubaa

Publishing and Subscribing to Transforms in ROS Nodes

https://www.udemy.com/user/anis-koubaa/

Introduction to TF



world to Tsim1_frame

world to Tsim2_frame

Tsim2_frame to Tsim1_frame

distance (translation) + rotation between the two robots

Robotics &
Internet-of-Things

ROS

Udemy Course
Robot Operating System
Basics, Motion, and OpenCV

Prof. Anis Koubaa

Control of Mobile Robots

https://www.udemy.com/user/anis-koubaa/