Experiment-4

Introduction to TF

Objectives

1. Understand the concept of TF.
2. Broadcast the state of the robot to TF
3. Get access to frame transformation Theory

tf is a package that lets the user keep track of multiple coordinate frames over time. tf maintains the relationship between coordinate frames in a tree structure buffered in time, and lets the user transform points, vectors, etc between any two coordinate frames at any desired point in time.

Set Up the Demo

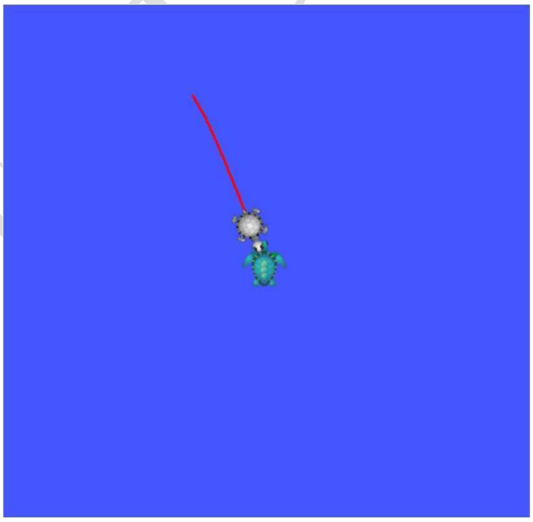
1. The nodes for this tutorial are released for Ubuntu, so go ahead and install them:

```
$ sudo apt-get install ros-noetic-ros-tutorials ros-noetic-geometry-tutorials ros-noetic-rviz ros-noetic-rosbash ros-noetic-rqt-tf-tree
```

2. Now that we're done getting the turtle_tf tutorial package, let's run the demo.

```
roslaunch turtle_tf turtle_tf_demo.launch
```

The display will be as follows;



3. This demo is using the tf library to create three coordinate frames: a world frame, a turtle1 frame, and a turtle2 frame. This tutorial uses a tf broadcaster to publish the turtle coordinate frames and a tf listener to compute the difference in the turtle frames and move one turtle to follow the other.
   tf Tools: view_frames creates a diagram of the frames being broadcast by tf over ROS. $ rosrun tf view_frames

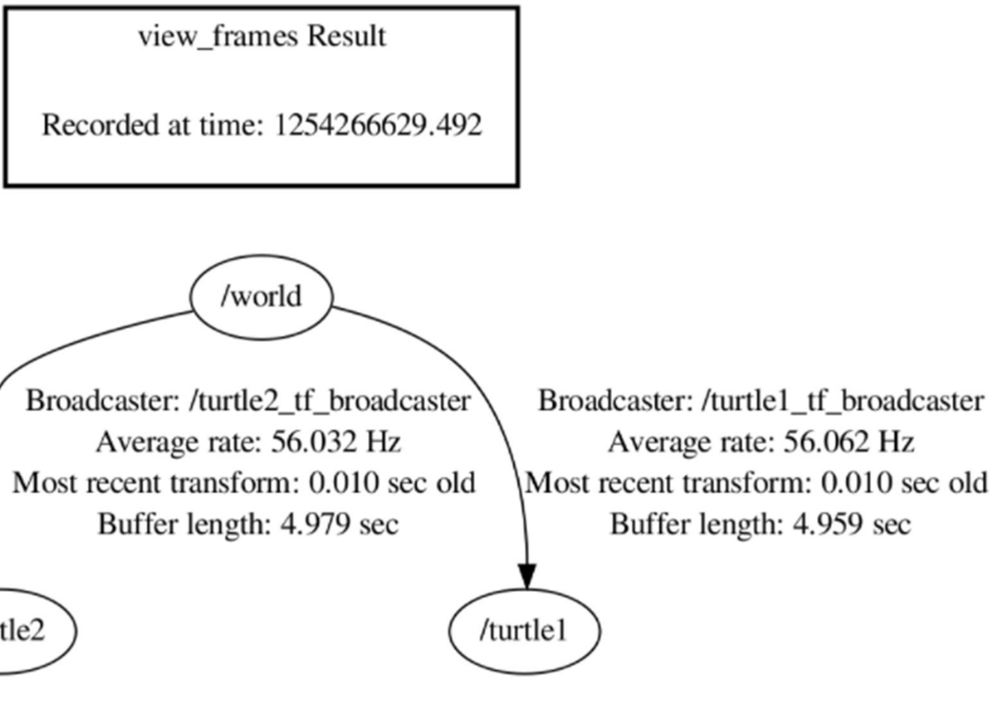
You will see:

```
Transform Listener initing

Listening to /tf for 5.000000 seconds

Done Listening

dot - Graphviz version 2.16 (Fri Feb  8 12:52:03 UTC 2008)


Detected dot version 2.16

frames.pdf generated
```

4. To view the tree:

```
evince frames.pdf
```

The result is


view_frames Result
Recorded at time: 1254266629.492
/world
Broadcaster: /turtle2_tf_broadcaster
Average rate: 56.032 Hz
Most recent transform: 0.010 sec old
Buffer length: 4.979 sec
Broadcaster: /turtle1_tf_broadcaster
Average rate: 56.062 Hz
Most recent transform: 0.010 sec old
Buffer length: 4.959 sec
/turtle2
/turtle1

5. To visualize tf in Rviz rviz is a visualization tool that is useful for examining tf frames. Let's look at our turtle frames using rviz. Let's start rviz with the turtle_tf configuration file using the d option for rviz:

```
$ rosrun rviz rviz -d `rospack find turtle_tf`/rviz/turtle_rviz.rviz
```