

<http://wiki.ros.org/tf/Tutorials/Introduction%20to%20tf>

Introduction to tf

Description: This tutorial will give you a good idea of what tf can do for you. It shows off some of the tf power in a multi-robot example using turtlesim. This also introduces using tf_echo, view_frames, rqt_tf_tree, and rviz.

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

```
harman@Laptop-M1210:~$ sudo apt-get install ros-indigo-ros-tutorials ros-indigo-geometry-
tutorials ros-indigo-rviz ros-indigo-roscat ros-indigo-rqt-tf-tree
```

2. Running the Demo

```
$ roslaunch turtle_tf turtle_tf_demo.launch
```

```
SUMMARY
```

```
=====
```

```
PARAMETERS
```

```
* /rostdistro: indigo
* /rosversion: 1.11.13
* /scale_angular: 2.0
* /scale_linear: 2.0
* /turtle1_tf_broadcaster/turtle: turtle1
* /turtle2_tf_broadcaster/turtle: turtle2
```

```
NODES
```

```
/
```

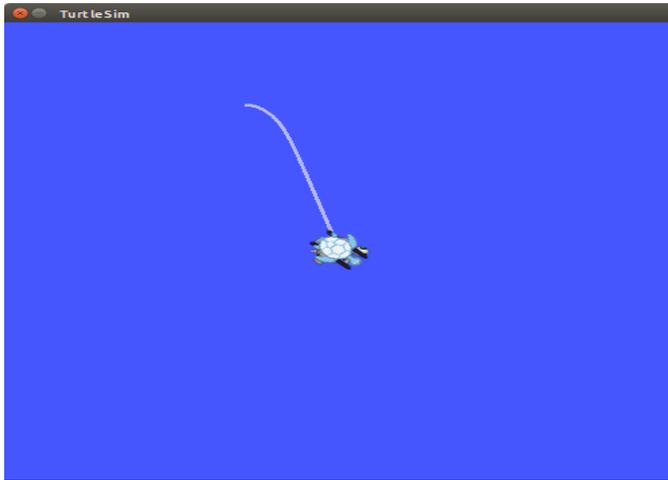
```
  sim (turtlesim/turtlesim_node)
  teleop (turtlesim/turtle_teleop_key)
  turtle1_tf_broadcaster (turtle_tf/turtle_tf_broadcaster.py)
  turtle2_tf_broadcaster (turtle_tf/turtle_tf_broadcaster.py)
  turtle_pointer (turtle_tf/turtle_tf_listener.py)
```

Reading from keyboard

```
-----
```

Use arrow keys to move the turtle.

(Keep this window Active)



What is Happening

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

4.1 Using view_frames

```
harman@Laptop-M1210:~$ rosrun tf view_frames
```

```
Listening to /tf for 5.000000 seconds
```

```
Done Listening
```

```
dot - graphviz version 2.36.0 (20140111.2315)
```

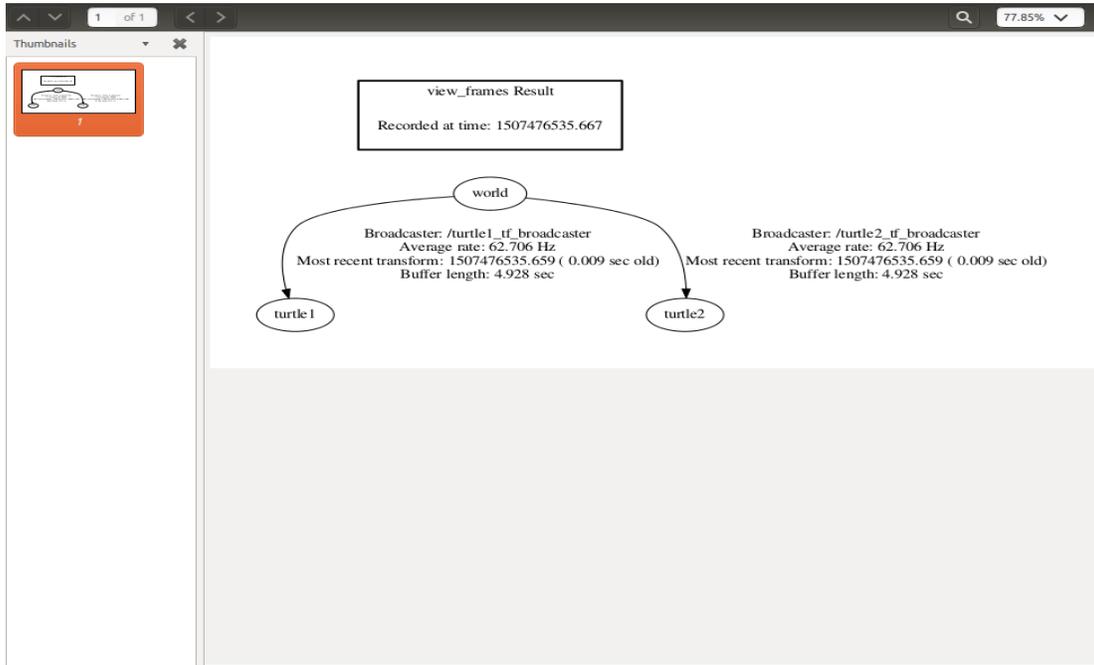
```
Detected dot version 2.36
```

```
frames.pdf generated
```

```
harman@Laptop-M1210:~$ evince frames.pdf
```

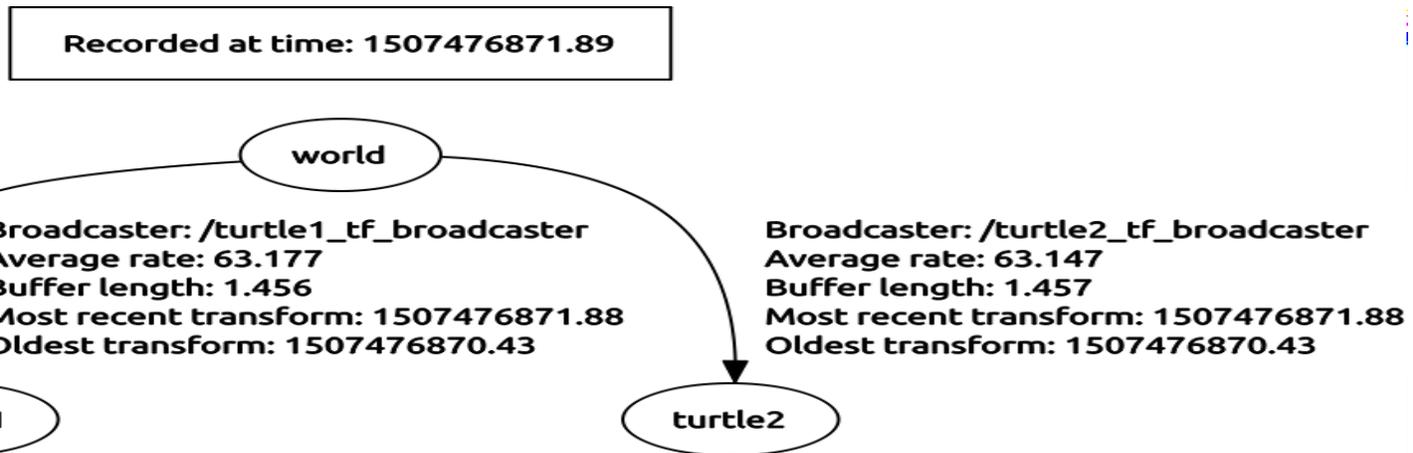
4.2

Using



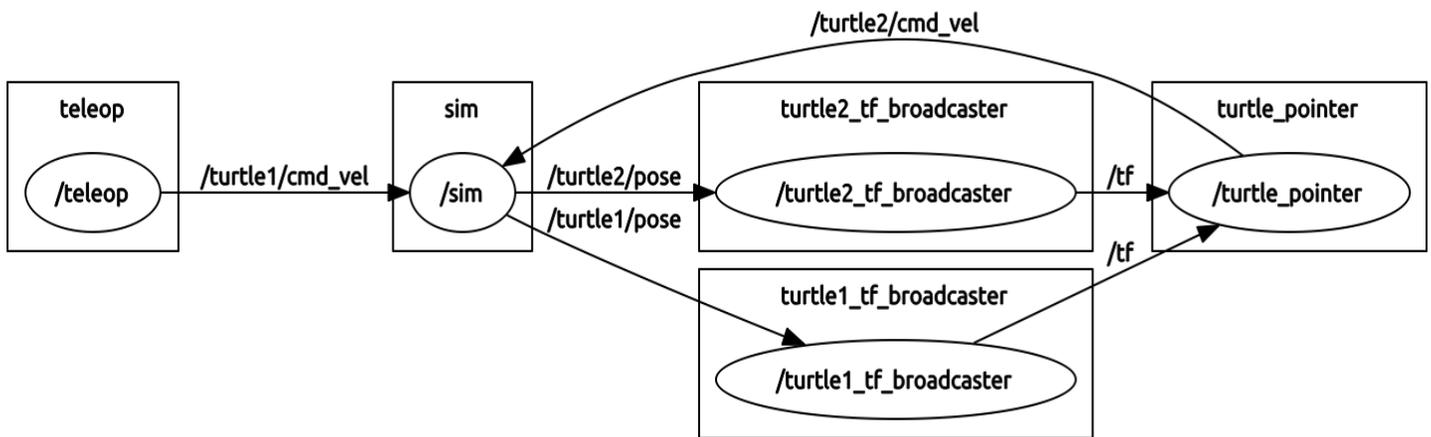
rqt_tf_tree

harman@Laptop-M1210:~\$ rosrund rqt_tf_tree rqt_tf_tree

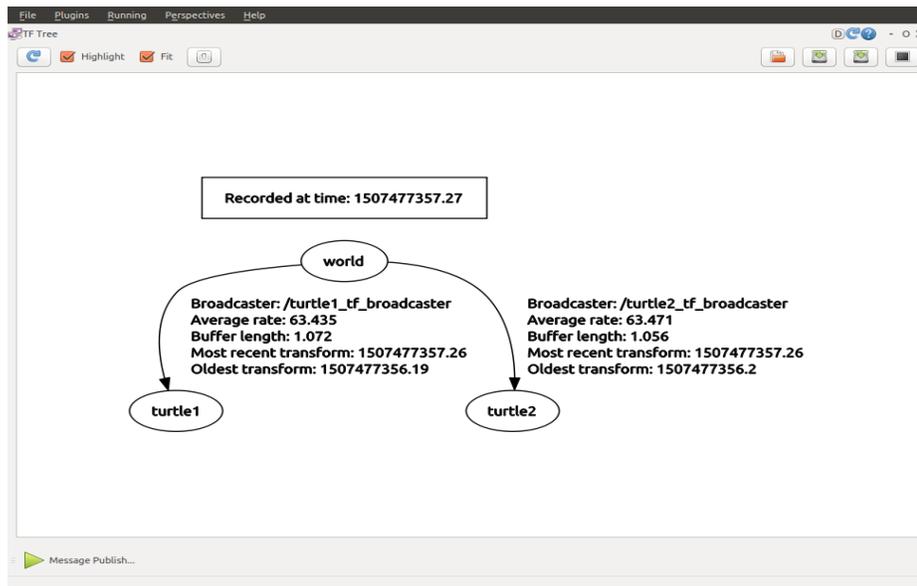


harman@Laptop-M1210:~\$ rqt &
[1] 3537

Choose Plugin > Introspection > Node Graph



Then choose `rqt_tf_tree` from Plugins tab



4.3 tf_echo

`tf_echo` reports the transform between any two frames broadcast over ROS.

(Initial Position Turtle1 in center facing East $q = 0, 0, 0, 1$)
harman@Laptop-M1210:~\$ rosrun tf tf_echo turtle1 world

```
harman@Laptop-M1210:- Translation: [-5.544, -5.544, 0.000]
- Rotation: in Quaternion [0.000, 0.000, 0.000, 1.000]
            in RPY (radian) [0.000, -0.000, 0.000]
            in RPY (degree) [0.000, -0.000, 0.000]
```

General Format

```
$ rosrun tf tf_echo [reference_frame] [target_frame]
```

Let's look at the transform of the turtle2 frame with respect to turtle1 frame which is equivalent to

```
harman@Laptop-M1210:~$ rosrun tf tf_echo turtle1 turtle2
```

At time 1507480878.688

- Translation: [0.000, 0.000, 0.000]

- Rotation: in Quaternion [0.000, 0.000, -0.726, 0.688]

in RPY (radian) [0.000, 0.000, -1.625]

in RPY (degree) [0.000, 0.000, -93.122]

5.0 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:

```
harman@Laptop-M1210:~$ rosrun rviz rviz -d `rospack find turtle_tf`/rviz/turtle_rviz.rviz
```

```
[ INFO] [1507481076.118430054]: rviz version 1.11.18
```

```
[ INFO] [1507481076.118541171]: compiled against Qt version 4.8.6
```

```
[ INFO] [1507481076.118574765]: compiled against OGRE version 1.8.1 (Byatis)
```

```
[ INFO] [1507481077.183342985]: Stereo is NOT SUPPORTED
```

```
[ INFO] [1507481077.183488464]: OpenGL version: 3 (GLSL 1.3).
```

Use top down camera view; close Right pane in rviz

At time 1507482140.159

- Translation: [0.000, 2.307, 0.000]

- Rotation: in Quaternion [0.000, 0.000, 0.883, 0.469]

in RPY (radian) [-0.000, -0.000, 2.165]

in RPY (degree) [-0.000, -0.000, 124.035]

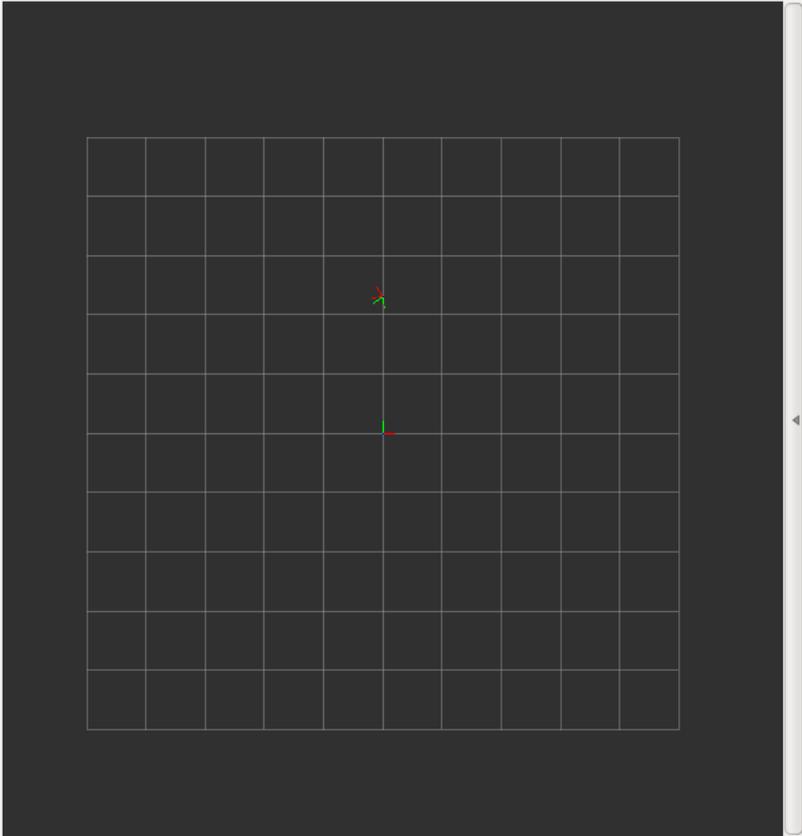
Interact Move Camera Select Focus Camera Measure 2D Pose Estimate 2D Nav Goal Publish Point

Displays

Global Options	
Fixed Frame	world
Background Color	48; 48; 48
Frame Rate	30
Global Status: Ok	
Grid	<input checked="" type="checkbox"/>
TF	<input checked="" type="checkbox"/>
Status: Ok	
Show Names	<input type="checkbox"/>
Show Axes	<input checked="" type="checkbox"/>
Show Arrows	<input type="checkbox"/>
Marker Scale	1
Update Interval	0
Frame Timeout	15
Frames	
All Enabled	<input type="checkbox"/>
turtle1	<input checked="" type="checkbox"/>
turtle2	<input checked="" type="checkbox"/>
world	<input checked="" type="checkbox"/>
Tree	

Fixed Frame
Frame into which all data is transformed before being displayed.

Add Duplicate Remove Rename



Time

ROS Time: 1507482354.98 ROS Elapsed: 140.21 Wall Time: 1507482355.02 Wall Elapsed: 140.21 Experimental

Reset 30 fps

