

06. TF 개념 익히기

AI ROBOT

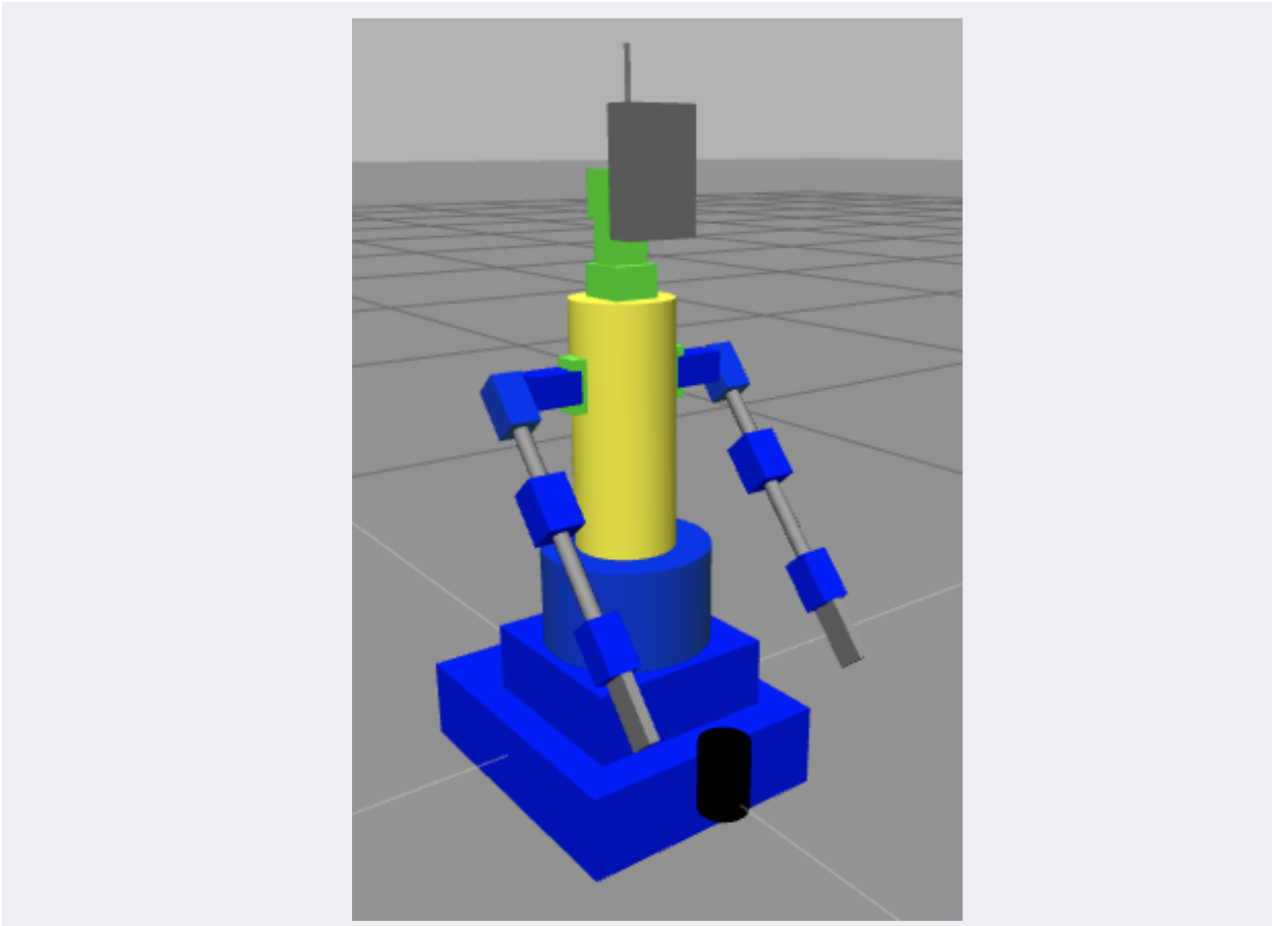
Exported on 08/04/2021

Table of Contents

1	TF	3
1.1	Transformation	3
1.2	간단하지 않은 좌표의 해석.....	4
1.3	TF의 개념을 간단하게 실습을 해보자	5
1.4	roslaunch turtle_tf turtle_tf_demo.launch	5
1.5	teleop_key 실행.....	6
1.6	추적하는 turtlesim	6
1.7	현재 토픽의 흐름	7
1.8	현재 tf의 상황을 기록.....	7
1.9	evince frames.pdf.....	7
1.10	같은 역할로 rosrn rqt_tf_tree rqt_tf_tree.....	8
1.11	tf topic을 한 번만 보자.....	8
1.12	rosrn tf tf_echo turtle1 turtle2	9
1.13	turtle1에서본 turtle2의 좌표	9
1.14	rosrn rviz rviz	10
1.15	frame의 상황과 함께 관찰할 수 있다.....	10
2	TF - pub and sub.....	11
2.1	설치	11
2.2	jupyter notebook 실행	11
2.3	python2로 새문서를 열어서 rospy가 import 되면 OK.....	11
2.4	현재 turtle_tf_demo를 실행하고 Jupyter notebook을 실행해둬	12
2.5	간단히 원리만 확인하는 차원에서 import 모듈.....	12
2.6	turtle2에서 바라본 turtle1의 상대 좌표와 자세를 받고	12
2.7	거리를 계산해본다면	13
2.8	이렇게 해볼 수 있을듯	13
2.9	robot state publisher	13

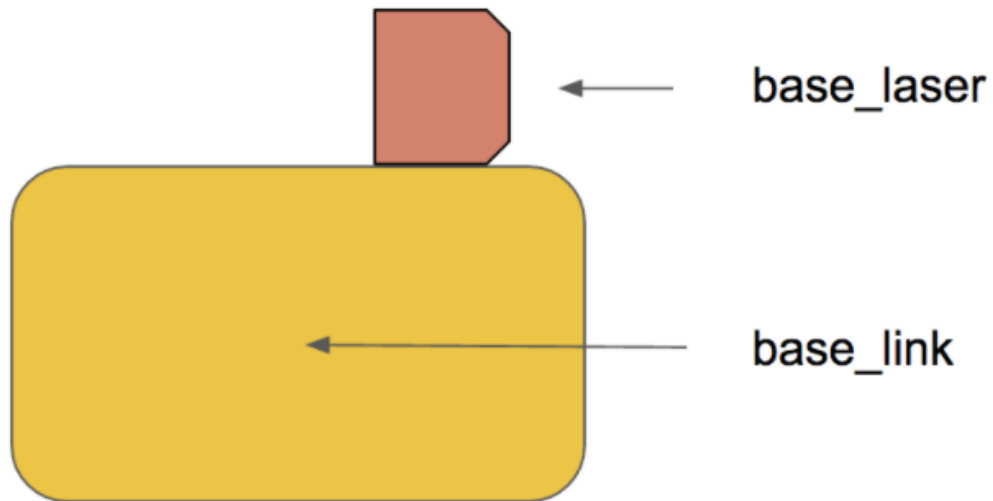
1 TF

1.1 Transformation



- 로봇은 3D 좌표를 아주 많이 가지고 있는 경우가 많다

1.2 간단하지 않은 좌표의 해석



- laser가 읽은 정보를 base_link 입장에서 해석하려면?

1.3 TF의 개념을 간단하게 실습을 해보자

1.4 `roslaunch turtle_tf turtle_tf_demo.launch`

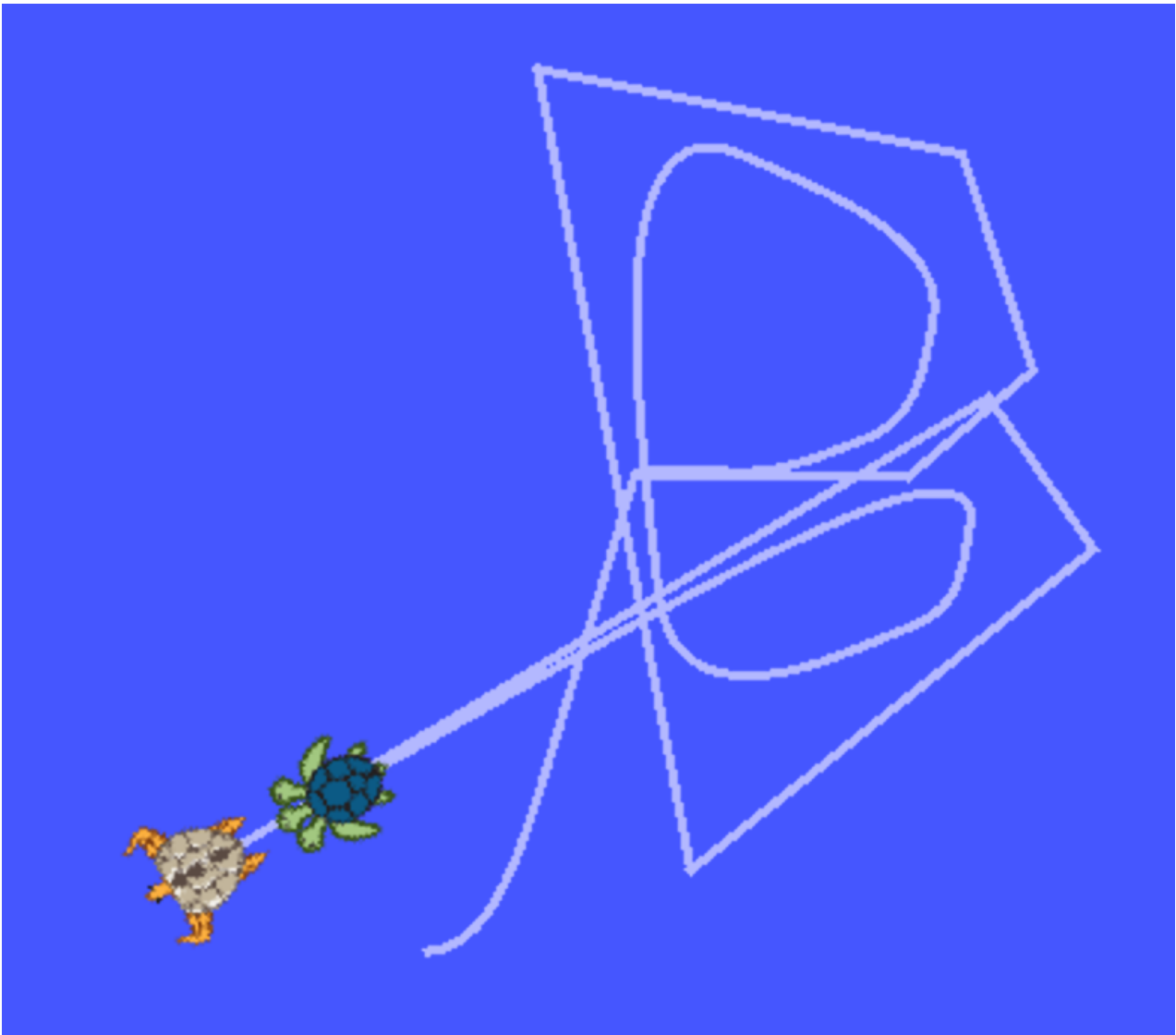


1.5 teleop_key 실행

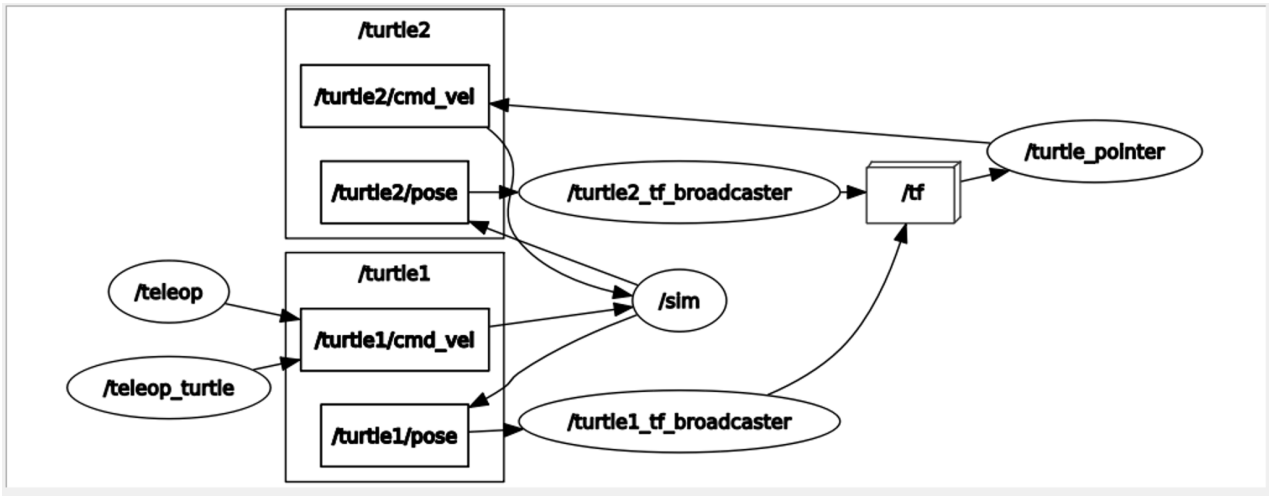
```
pw@melodic:~$ rosrun turtlesim turtle_teleop_key
Reading from keyboard
-----
Use arrow keys to move the turtle. 'q' to quit.

```

1.6 추적하는 turtlesim



1.7 현재 토픽의 흐름



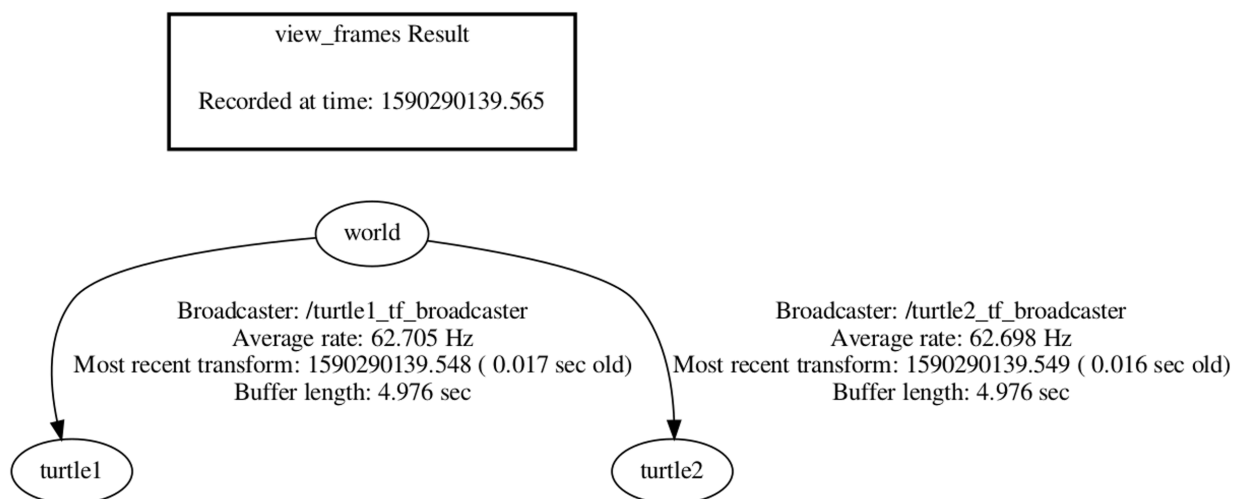
1.8 현재 tf의 상황을 기록

```

pw@melodic:~$
pw@melodic:~$ rosrund tf view_frames
Listening to /tf for 5.0 seconds
Done Listening
dot - graphviz version 2.40.1 (20161225.0304)

Detected dot version 2.40
frames.pdf generated
  
```

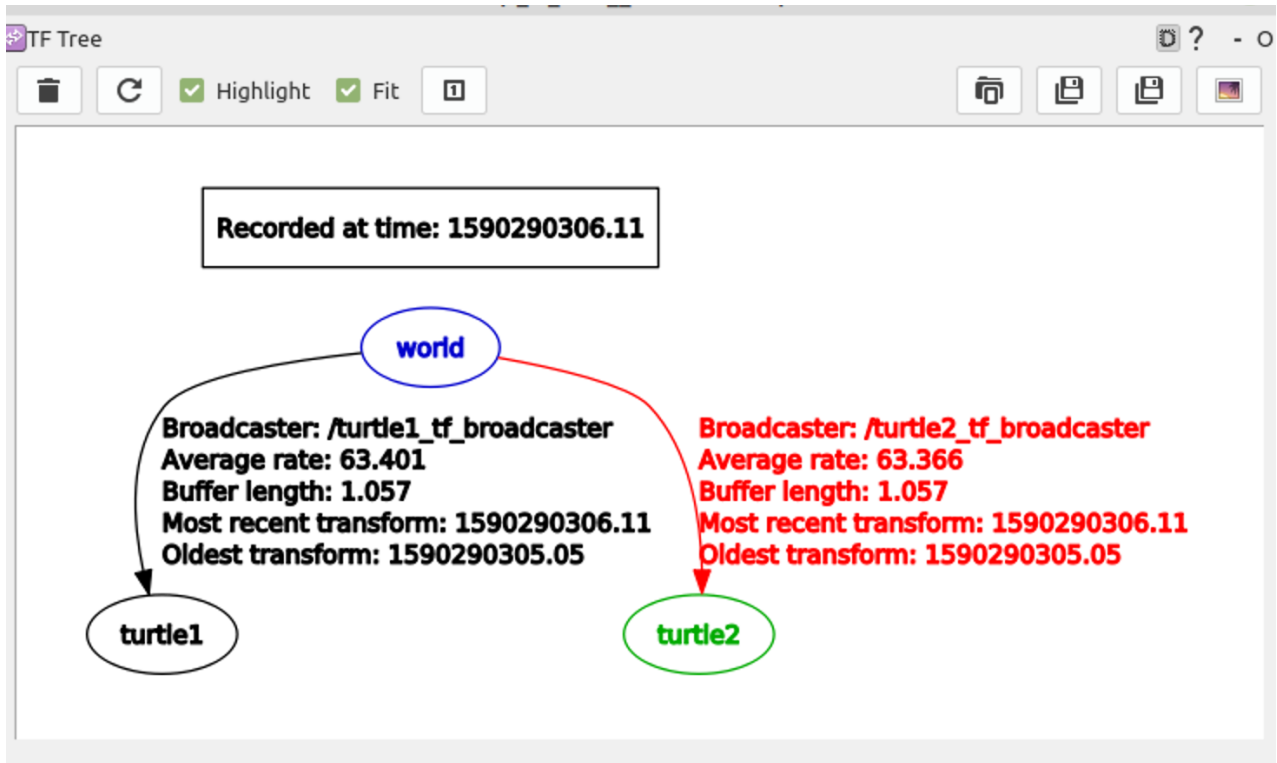
1.9 evince frames.pdf



- / turtle1 및 / turtle2라는 두 개의 프레임이 브로드 캐스트
- 공통 부모는 / world 프레임

- **Broadcaster:** TF 데이터의 브로드 캐스터 이름

1.10 같은 역할로 `roslaunch rqt_tf_tree rqt_tf_tree`



1.11 `tftopic`을 한 번만 보자

```

pw@melodic:~$ rostopic echo -n1 /tf
transforms:
  header:
    package: std_msgs
    seq: 0
    stamp:
      secs: 1590290388
      nsecs: 524173021
    frame_id: "world"
  child_frame_id: "turtle2"
  transform:
    translation:
      x: 6.00517559052
      y: 3.0319609642
      z: 0.0
    rotation:
      x: 0.0
      y: 0.0
      z: 0.953937220198
      w: 0.300006299801

```


1.12 rosrun tf tf_echo turtle1 turtle2

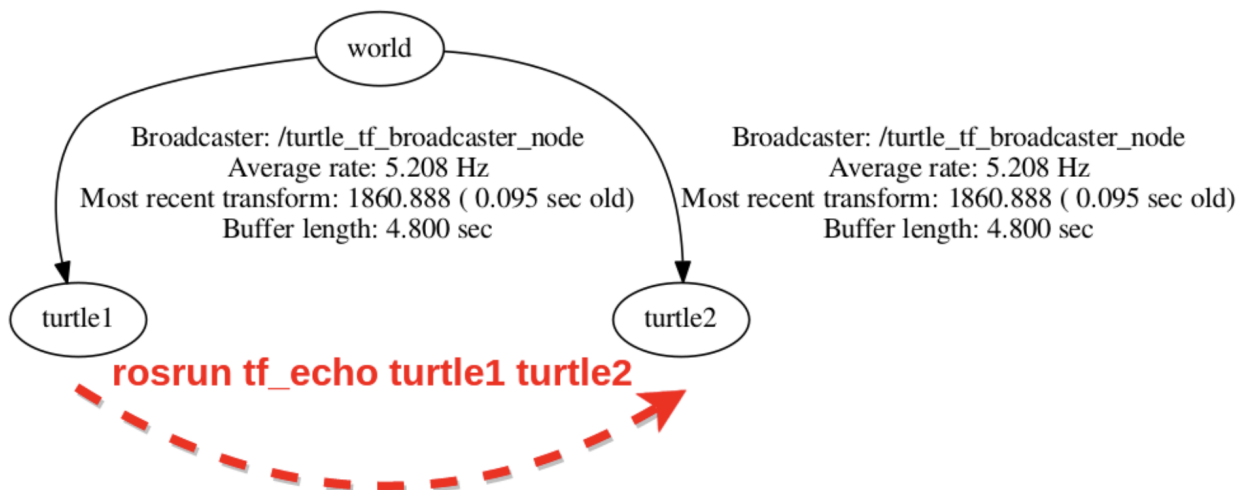
```

pw@melodic:~$ rosrun tf tf_echo turtle1 turtle2
At time 1590290451.660
- Translation: [0.000, 0.000, 0.000]
- Rotation: in Quaternion [0.000, 0.000, 0.996, 0.094]
            in RPY (radian) [0.000, -0.000, 2.953]
            in RPY (degree) [0.000, -0.000, 169.195]
At time 1590290452.381
- Translation: [0.000, 0.000, 0.000]
- Rotation: in Quaternion [0.000, 0.000, 0.996, 0.094]
            in RPY (radian) [0.000, -0.000, 2.953]
            in RPY (degree) [0.000, -0.000, 169.195]
At time 1590290453.373
- Translation: [0.000, 0.000, 0.000]
- Rotation: in Quaternion [0.000, 0.000, 0.996, 0.094]
            in RPY (radian) [0.000, -0.000, 2.953]
            in RPY (degree) [0.000, -0.000, 169.195]
^Cpw@melodic:~$

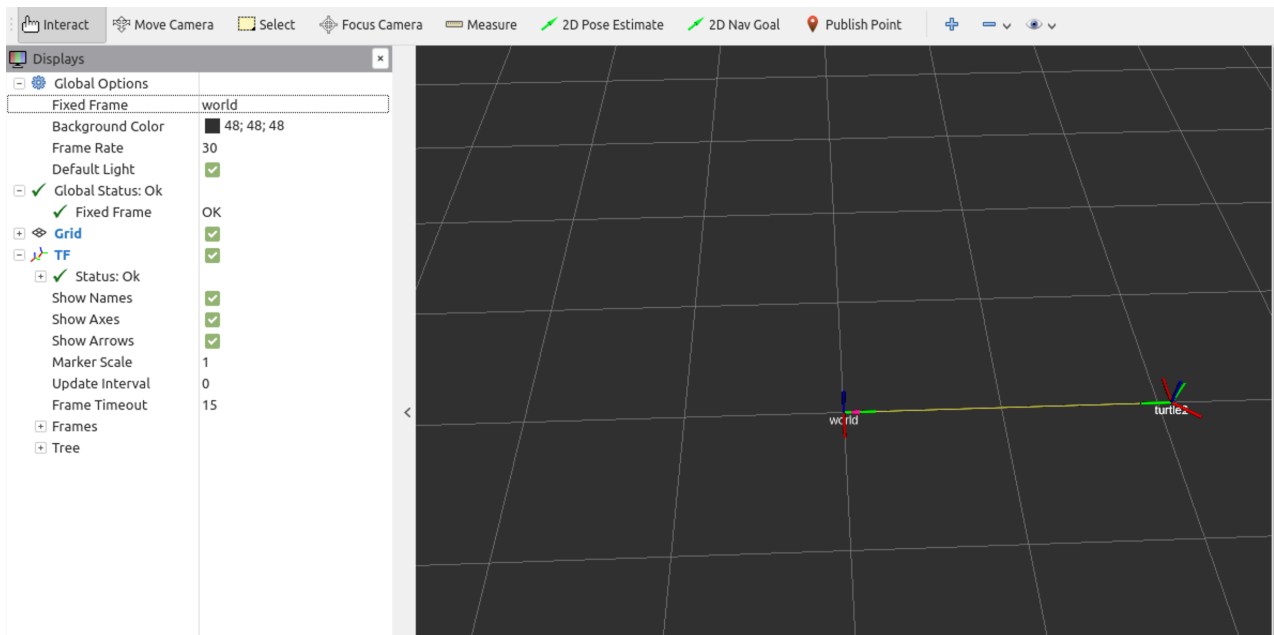
```

- 특정 frame간의 데이터만 볼 수 있다

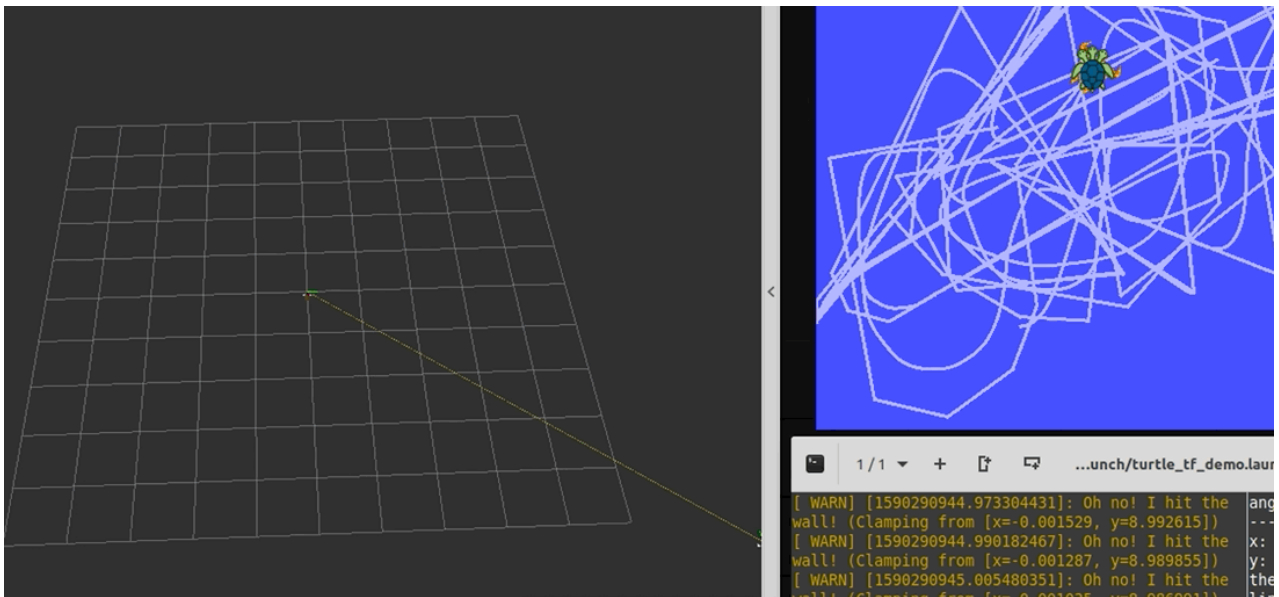
1.13 turtle1에서본 turtle2의 좌표



1.14 rosrn rviz rviz



1.15 frame의 상황과 함께 관찰할 수 있다



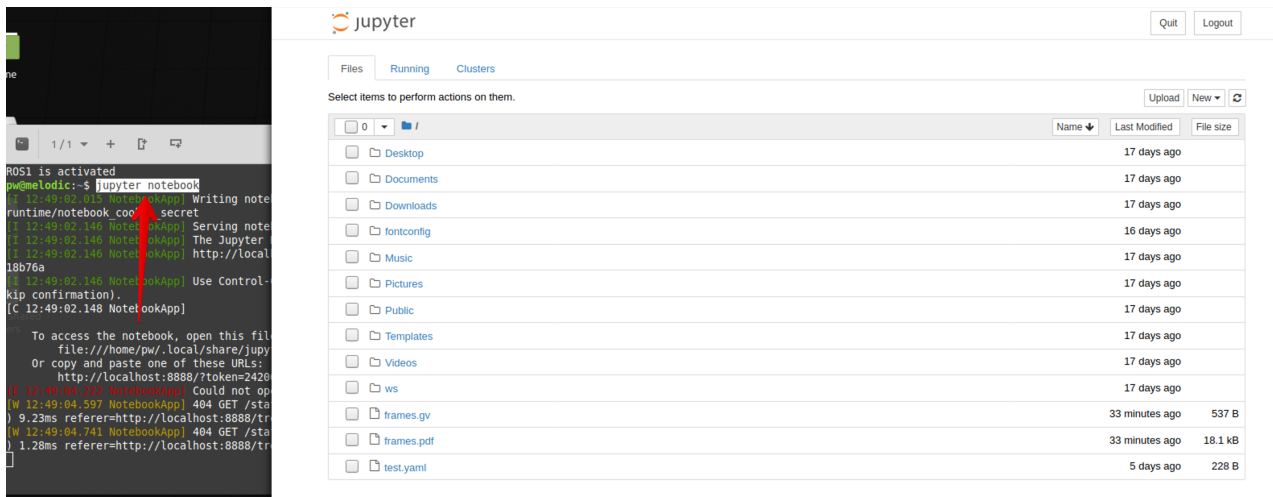
2 TF - pub and sub

2.1 설치

```
sudo apt install python-pip
pip install --upgrade pip
pip install jupyter
pip install ipython
```

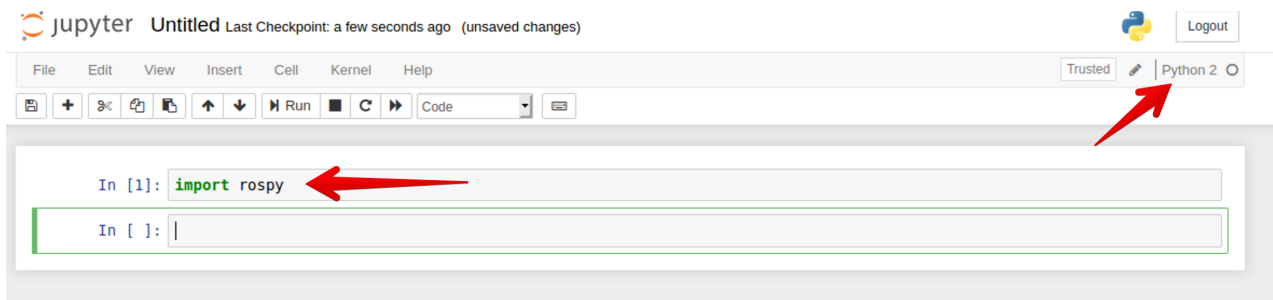
- 그리고 reboot

2.2 jupyter notebook 실행



The image shows a terminal window on the left and a Jupyter Notebook interface on the right. The terminal displays the command `jupyter notebook` being executed, and the Jupyter interface shows a file browser with various folders and files.

2.3 python2로 새문서를 열어서 rospy가 import 되면 OK



The image shows a Jupyter Notebook interface with a new file named 'Untitled' open. The interface includes a menu bar (File, Edit, View, Insert, Cell, Kernel, Help) and a toolbar with icons for file operations and execution. The first cell contains the command `import rospy`, which is highlighted by a red arrow. The second cell is empty and ready for input.

2.4 현재 turtle_tf_demo를 실행하고 Jupyter notebook을 실행해둠

```
[W 14:51:30.579 NotebookApp] 404 GET /static/components/react/react-dom.production.min.js (127.0.0.1) 1.17ms referer=http://localhost:8888/notebooks/Documents/Untitled.ipynb?kernel_name=python2
[W 14:51:30.698 NotebookApp] 404 GET /static/components/react/react-dom.production.min.js (127.0.0.1) 0.96ms referer=http://localhost:8888/notebooks/Documents/Untitled.ipynb?kernel_name=python2
[I 14:51:31.191 NotebookApp] Kernel started: 5432a713-52cc-4826-bcee-a221f278c41a
[W 14:51:31.202 NotebookApp] 404 GET /nbextensions/widgets/notebook/js/extension.js?v=20200524145106 (127.0.0.1) 1.51ms referer=http://localhost:8888/notebooks/Documents/Untitled.ipynb?kernel_name=python2
[I 14:51:31.564 NotebookApp] Adapting to protocol v5.1 for kernel 5432a713-52cc-4826-bcee-a221f278c41a
start_debug_demo.launch turtle_tf_demo.launch turtle_tf_sensor.launch
pw@melodic:~$ roslaunch turtle_tf turtle_tf_demo.launch
... logging to /home/pw/.ros/log/8bb2c752-9d82-11ea-8ee9-001c420be203/roslaunch-melodic-6520.log
Checking log directory for disk usage. This may take a while.
Press Ctrl-C to interrupt
Done checking log file disk usage. Usage is <1GB.

started roslaunch server http://melodic:39993/

SUMMARY
```

2.5 간단히 원리만 확인하는 차원에서 import 모듈

```
In [1]: #!/usr/bin/env python

import rospy
import time
import tf
from turtlesim.msg import Pose
import geometry_msgs.msg
```

```
In [2]: rospy.init_node('turtle_tf_listener')
listener = tf.TransformListener()
turtle_vel = rospy.Publisher('turtle2/cmd_vel',
                             geometry_msgs.msg.Twist, queue_size=1)
```

2.6 turtle2에서 바라본 turtle1의 상대 좌표와 자세를 받고

```
In [3]: listener.lookupTransform('/turtle2', '/turtle1', rospy.Time(0))
Out[3]: ([0.0, 0.0, 0.0], [0.0, 0.0, -0.04248841695728023, 0.9990969594711337])
```

```
In [4]: (trans, rot) = listener.lookupTransform('/turtle2', '/turtle1',
                                             rospy.Time(0))
trans
```

```
Out[4]: [0.0, 0.0, 0.0]
```

2.7 거리를 계산해본다면

```
In [5]: import numpy as np
        np.sqrt((np.array([1, 2, 3]) ** 2).mean())
```

```
Out[5]: 2.1602468994692869
```

```
In [6]: np.sqrt((np.array(trans) ** 2).mean())
```

```
Out[6]: 0.0
```

2.8 이렇게 해볼 수 있을듯

```
In [9]: for n in range(10):
        time.sleep(0.5)
        (trans,rot) = listener.lookupTransform('/turtle2', '/turtle1',
                                                rospy.Time(0))
        print(np.sqrt((np.array(trans) ** 2).mean()))
```

```
0.0
0.0
0.0
0.0
0.202933505379
0.682560812822
1.04779234707
1.33895284011
1.17638973613
0.703204783523
```

2.9 robot state publisher

- URDF를 작성하면 자동으로 TF를 broadcasting 함