

ROS Assignment

Vasileios Papadopoulos

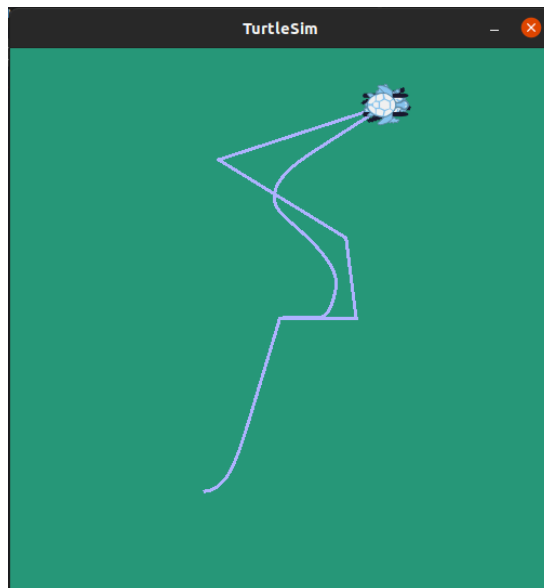
Binaries

ROS Noetic is installed in following directory: `/opt/ros/noetic/bin`

Launch turtle bot

A good practice when working with ROS is to launch the *roscore* underline system, which provides communication among ROS nodes. Though, in our case it is redundant since *roslaunch* launches it automatically if detects that has been stopped. We continue with turtle demo project.

1. `roscore`
2. `roslaunch turtle_tf turtle_tf_demo.launch`



Change background

In order to change the background color of turtlesim we use the *rosparam* command. Particularly, we set RGB colors individually and arbitrarily and then we use *rosservice* command to apply the changes. In order find the available parameters, we first run the *rosparam list*

1. rosparam list
2. rosparam set /sim/background_r 0
3. rosparam set /sim/background_g 100
4. rosparam set /sim/background_b 150
5. rosservice call /clear #apply changes

```

src> M CMakeLists.txt
41   endif()
42   endif()
43   endif()
44   # list of /roslaunch/urls/host_vassllispapadop_virtualbox_33237
45   # list of /roslaunch/urls/host_vassllispapadop_virtualbox_35433
46   set(catin /roslaunch/urls/host_vassllispapadop_virtualbox_41139
47   foreach(pat/roslaunch/urls/host_vassllispapadop_virtualbox_43493
48   if(EXISTS/rosversion
49   list(FI/run_id
50   if(inc/scale.angular
51   list(/scale.linear
52   list(/sin/background_b
53   endif() /sin/background_g
54   endforeach(/sin/background_r
55   /turtle1/background_b
56   /turtle1/background_r
57   # search for
58   set(CATKIN /turtle2_tf_broadcaster/turtle
59   find package
60   NO_POLICY /turtles_tf/background_b
61   NO_POLICY /turtles_tf/background_r
62   PATHS $(catkin/find/background_b
63   NO_DEFAULT /turtlesin/background_g
64   unset(CATKIN/turtlesin/background_r
65   vassllispapadop@vassllispapadop-VirtualBox: /opt/ros/noetic/bin$ rosparam list
66   if(NOT cat/vassllispapadop@vassllispapadop-VirtualBox: /opt/ros/noetic/bin$ rosparam set /sim/background_b 120
67   if(NOT cat/vassllispapadop@vassllispapadop-VirtualBox: /opt/ros/noetic/bin$ rosparam set /sim/background_r 45
68   message(vassllispapadop@vassllispapadop-VirtualBox: /opt/ros/noetic/bin$ rosparam set /sim/background_g 150
69   endif() vassllispapadop@vassllispapadop-VirtualBox: /opt/ros/noetic/bin$ rosservice call /clear
70   vassllispapadop@vassllispapadop-VirtualBox: /opt/ros/noetic/bin$ rosparam set /sim/background_g 100
71   vassllispapadop@vassllispapadop-VirtualBox: /opt/ros/noetic/bin$ rosparam set /sim/background_r 0
72   Found PythonInter/vassllispapadop@vassllispapadop-VirtualBox: /opt/ros/noetic/bin$ rosparam set /sim/background_b 0
73   Using PYTHON EXEC/vassllispapadop@vassllispapadop-VirtualBox: /opt/ros/noetic/bin$ rosparam set /sim/background_b 150
74   Using Debian Pyt/vassllispapadop@vassllispapadop-VirtualBox: /opt/ros/noetic/bin$ rosservice call /clear
75   Using empy: /usr/vassllispapadop@vassllispapadop-VirtualBox: /opt/ros/noetic/bin$
76   call enable Test
77   Using CATKIN TEST
78   Forcing gtest/gm
79   Found gtest sour
80   Found gmock sour
81   Found PythonInter
82   Using Python nose
83   catkin 0.8.9
84   -- BUILD_SHARED_LIBS
85   -- BUILD_SHARED_LIBS
86   -- -- traversing 1
87   -- -- - ros_assign
88   --
89   -- +++ processing cr
90   -- -- add subdirect
91   -- -- Installing devel
92   -- -- le distance.py to /
93   -- -- Installing devel
94   -- -- publisher.py to /h
95   -- -- Configuring done
96   -- -- Generating done
97   -- -- Build files have
98   ####
99   #### Running comman
100  ####
101  vassllispapadop@vass:

```

Background changes to blue-ish, though we notice that previous line path has also been changed/cleared.

Inspect tf tree

In order to run `tf view_frames` command I had to change line 89 in `/opt/ros/noetic/lib/view_frames` to avoid **TypeError: cannot use a string pattern on a bytes-like object**

line 89 replaced with :

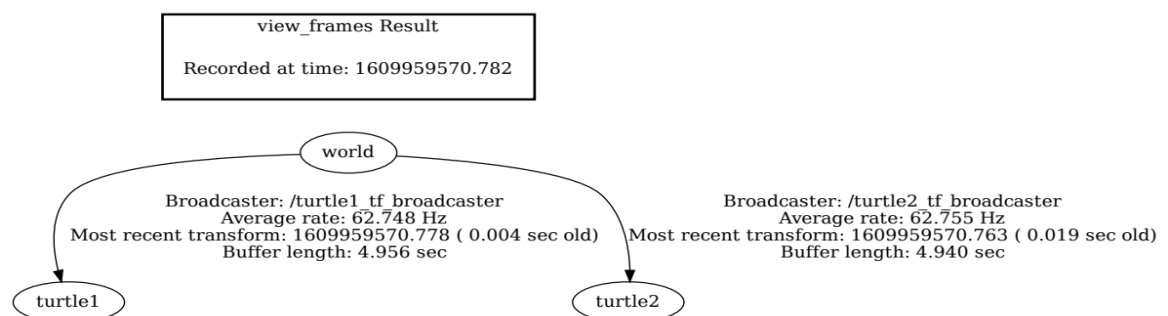
```
decoded = vstr.decode('utf-8')
```

```
m = r.search(decoded)
```

```
$ lsb_release -a
```

```
Distributor ID:    Ubuntu
Description:      Ubuntu 20.04.1 LTS
Release:          20.04
Codename:         focal
```

1. `roslaunch tf view_frames` #OR
2. `roslaunch rqt_tf_tree rqt_tf_tree`
3. `evince frames.pdf` #view pdf



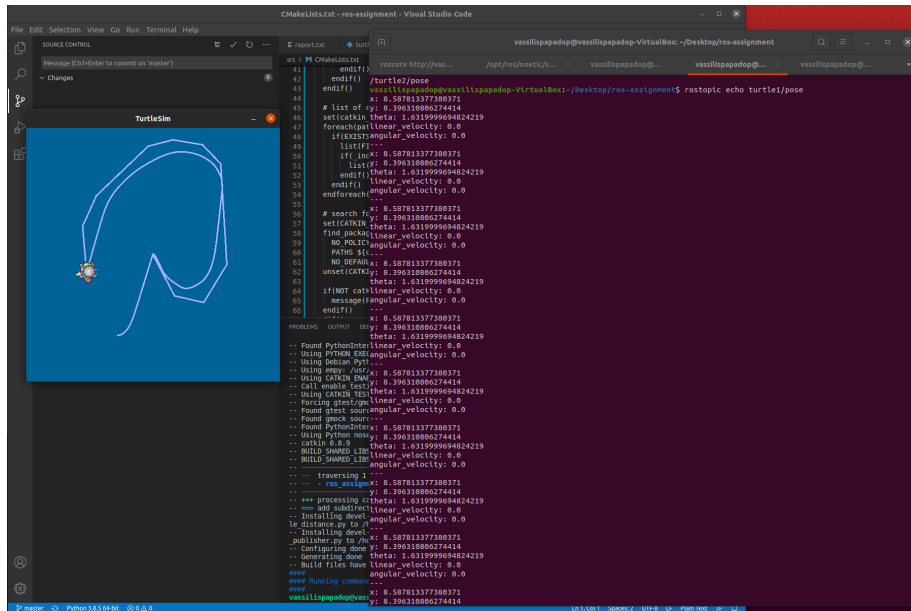
Display topic messages

`rostopic` command displays information of all active ROSTopics. It lists all subscribers/publishers, messages and more. With second command we subscribe to `turtle1/pose` topic as shown below.

1. `rostopic list`
2. `rostopic echo turtle1/pose`

Launch teleoperation and record moves

`roslaunch turtlesim turtle_teleop_key` command allows us to navigate the turtle with keyboard arrows. In a separate terminal, we record and save a rosbag file which contains the navigation/path.



1. `roslaunch turtlesim turtle_teleop_key`
2. `roslaunch turtlesim turtle_teleop_key`

Create catkin package

In order to create a package in ROS environment we install the build system *catkin*, which allows us to generate targets(executables, scripts, libraries) from source code. We navigate to desired folder and run *catkin_create_pkg*. This will generate *package.xml* and *CMakeLists.txt* files which contain information about the package. Then we build the package with *catkin_make* and add the workspace to ROS environment, `. ~/desktop/ros-assignment/devel/setup.bash`

1. `cd ~/desktop/ros-assignment/src`
2. `catkin_create_pkg ros_assignment std_msgs rospy roscpp`
3. `cd ~/desktop/ros-assignment`
4. `catkin_make`
5. `. ~/desktop/ros-assignment/devel/setup.bash`
6. `rospack depends1 ros_assignment`

The last command lists the dependencies the *ros-assignment* package relies on.

Calculate Distance(subscribe/publish)

Below is a simple python script that plays the role of both subscriber and publisher. Having the *ros-bag play subset.bag* command running in a separate terminal we launch the script. We give the *rospy* node the name “turtle_subscriber” and another *rospy.Publisher* instance is created with name “turtle_publisher”. *rospy.Subscriber* subscribes to */turtle1/pose* topic and a callback is set to be triggered for every new incoming message. From within the *poseCallback* function we emit/publish the current travelled distance which is calculated as the Euclidean distance within *calculate_distance* method. *poseCallback* method has the previously created *rospy.Publisher* instance as an argument. Finally, current travelled distance is published and logged.

```
cat ~/desktop/ros-assignment/src/ros-assignment/scripts/turtle_distance.py
```

```
import rospy
from turtlesim.msg import Pose
from std_msgs.msg import String
from math import sqrt
from threading import Thread, Lock
```

```
prev_x = 0.0
prev_y = 0.0
total_distance = 0.0
```

```

# Euclidean distance
def calculate_distance(p1_x, p1_y, p2_x, p2_y):
    return sqrt((p1_x - p2_x)**2 + (p1_y - p2_y)**2)

def poseCallback(msg, publisher):
    #rospy.loginfo("turtle pose: x:%06f, y:%0.6f",msg.x , msg.y)
    global prev_x
    global prev_y
    global total_distance
    step = calculate_distance(prev_x, prev_y, msg.x, msg.y)
    total_distance += step
    prev_x = msg.x
    prev_y = msg.y

    #publish distance
    publish_msg = "Total distance %s" % total_distance
    rospy.loginfo(publish_msg)
    publisher.publish(publish_msg)

def subscriber():
    rospy.init_node('turtle_subscriber', anonymous=True)

    publisher = rospy.Publisher('turtle_publisher', String, queue_size=10)

    rospy.Subscriber('/turtle1/pose', Pose, poseCallback, publisher)

    rospy.spin()

# print
print("Total travelled distance", total_distance)

if __name__ == '__main__':
    subscriber()

```

```
turtle_distance.py - ros-assignment - Visual Studio Code

src > ros_assignment > scripts > turtle_distance.py > poseCallBack > publisher

1 import rosp
2 from turtlesim.msg import Pose
3 from std_msgs.msg import String
4 from math import sqrt
5 from threading import Thread, Lock
6
7 prev_x = 0.0
8 prev_y = 0.0
9 total_distance = 0.0
10
11 # Euclidean distance
12 def calculate_distance(p1_x, p1_y, p2_x, p2_y):
13     return sqrt((p1_x - p2_x)**2 + (p1_y - p2_y)**2)
14
15 def poseCallBack(msg, publisher):
16     #rospy.loginfo("turtle pose: x:%06f, y:%0.6f", msg.x, msg.y)
17     global prev_x
18     global prev_y
19     global total_distance
20     step = calculate_distance(prev_x, prev_y, msg.x, msg.y)
21     total_distance += step
22     prev_x = msg.x
23     prev_y = msg.y
24
25     #publish distance
26     publish_msg = "Total distance %s" % total_distance

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL 2: Python Debug Consc v

[INFO] [1610271751.480827]: Total distance 8127.536539796619
[INFO] [1610271751.540532]: Total distance 8132.159941005036
[INFO] [1610271751.543018]: Total distance 8136.783342213453
[INFO] [1610271751.571417]: Total distance 8141.375081615563
[INFO] [1610271751.643941]: Total distance 8145.966821017674
[INFO] [1610271751.696090]: Total distance 8150.526903388945
[INFO] [1610271751.704948]: Total distance 8155.0869857602165
[INFO] [1610271751.738353]: Total distance 8159.6154154012875
[INFO] [1610271751.744629]: Total distance 8164.148279599378
[INFO] [1610271751.781812]: Total distance 8168.649477119035
[INFO] [1610271751.788166]: Total distance 8173.155167865061
[INFO] [1610271751.808372]: Total distance 8177.629177545402
[INFO] [1610271751.808761]: Total distance 8182.107739564352
[INFO] [1610271751.827552]: Total distance 8186.554606387629
[INFO] [1610271751.828177]: Total distance 8191.006085564217
[INFO] [1610271751.893593]: Total distance 8195.42585545497
[INFO] [1610271751.907681]: Total distance 8199.850298170002
[INFO] [1610271751.931517]: Total distance 8204.243017325143
[INFO] [1610271751.938343]: Total distance 8208.640470704015
[INFO] [1610271751.959120]: Total distance 8213.006187229625
[INFO] [1610271751.965043]: Total distance 8217.376699847953
[INFO] [1610271751.985959]: Total distance 8221.715462158705
[INFO] [1610271751.989432]: Total distance 8226.059083385328
[INFO] [1610271752.006654]: Total distance 8230.37094092352
[INFO] [1610271752.012488]: Total distance 8234.687720684897
[INFO] [1610271752.023165]: Total distance 8238.972723702891
[INFO] [1610271752.026881]: Total distance 8243.262713232383
[INFO] [1610271752.091020]: Total distance 8247.520912595743
[INFO] [1610271752.109838]: Total distance 8251.784163258124
[INFO] [1610271752.127718]: Total distance 8256.015611604538
[INFO] [1610271752.129243]: Total distance 8260.252177045444
```

Test publisher

cat ~/desktop/ros-assignment/src/ros-assignment/scripts/test_published.py

```
import rospy
from std_msgs.msg import String

def callback(data):
    rospy.loginfo(rospy.get_caller_id() + 'received %s', data.data)

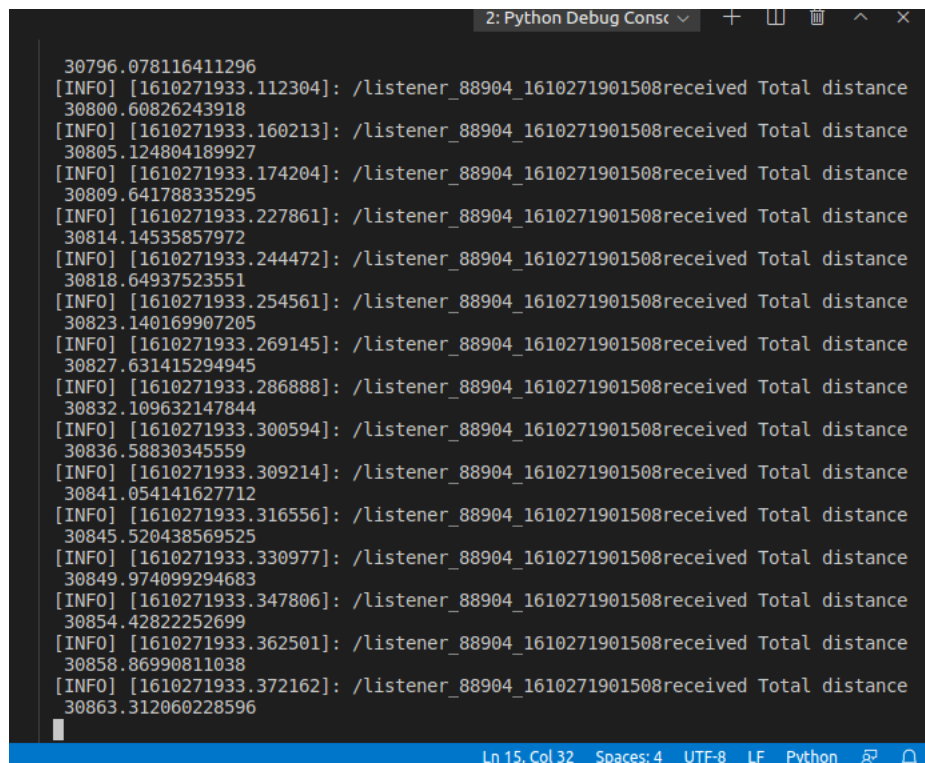
def listener():
    rospy.init_node('listener', anonymous=True)
```

```
rospy.Subscriber('turtle_publisher', String, callback)
```

```
# spin() simply keeps python from exiting until this node is stopped
```

```
rospy.spin()
```

```
if __name__ == '__main__':  
    listener()
```



The screenshot shows a Python Debug Console window with the title '2: Python Debug Console'. The console displays a series of log messages from a ROS node. Each message is an INFO log with a timestamp, a topic name, and a total distance value. The messages are as follows:

```
30796.078116411296  
[INFO] [1610271933.112304]: /listener_88904_1610271901508received Total distance  
30800.60826243918  
[INFO] [1610271933.160213]: /listener_88904_1610271901508received Total distance  
30805.124804189927  
[INFO] [1610271933.174204]: /listener_88904_1610271901508received Total distance  
30809.641788335295  
[INFO] [1610271933.227861]: /listener_88904_1610271901508received Total distance  
30814.14535857972  
[INFO] [1610271933.244472]: /listener_88904_1610271901508received Total distance  
30818.64937523551  
[INFO] [1610271933.254561]: /listener_88904_1610271901508received Total distance  
30823.140169907205  
[INFO] [1610271933.269145]: /listener_88904_1610271901508received Total distance  
30827.631415294945  
[INFO] [1610271933.286888]: /listener_88904_1610271901508received Total distance  
30832.109632147844  
[INFO] [1610271933.300594]: /listener_88904_1610271901508received Total distance  
30836.58830345559  
[INFO] [1610271933.309214]: /listener_88904_1610271901508received Total distance  
30841.054141627712  
[INFO] [1610271933.316556]: /listener_88904_1610271901508received Total distance  
30845.520438569525  
[INFO] [1610271933.330977]: /listener_88904_1610271901508received Total distance  
30849.974099294683  
[INFO] [1610271933.347806]: /listener_88904_1610271901508received Total distance  
30854.42822252699  
[INFO] [1610271933.362501]: /listener_88904_1610271901508received Total distance  
30858.86990811038  
[INFO] [1610271933.372162]: /listener_88904_1610271901508received Total distance  
30863.312060228596
```

The status bar at the bottom of the window indicates 'Ln 15, Col 32', 'Spaces: 4', 'UTF-8', 'LF', 'Python', and icons for search and help.