

Lab report #1

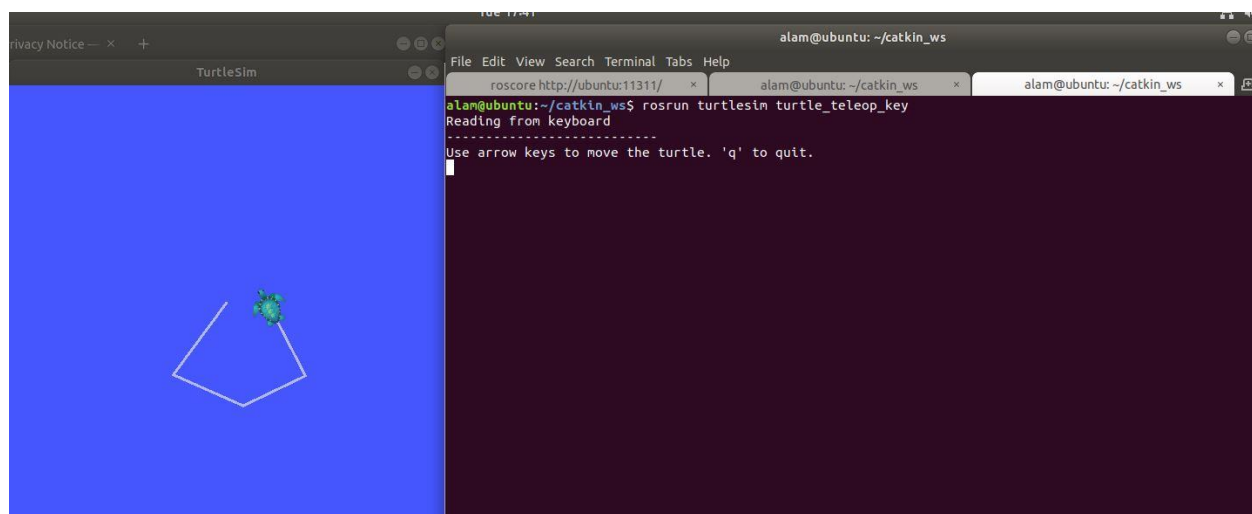
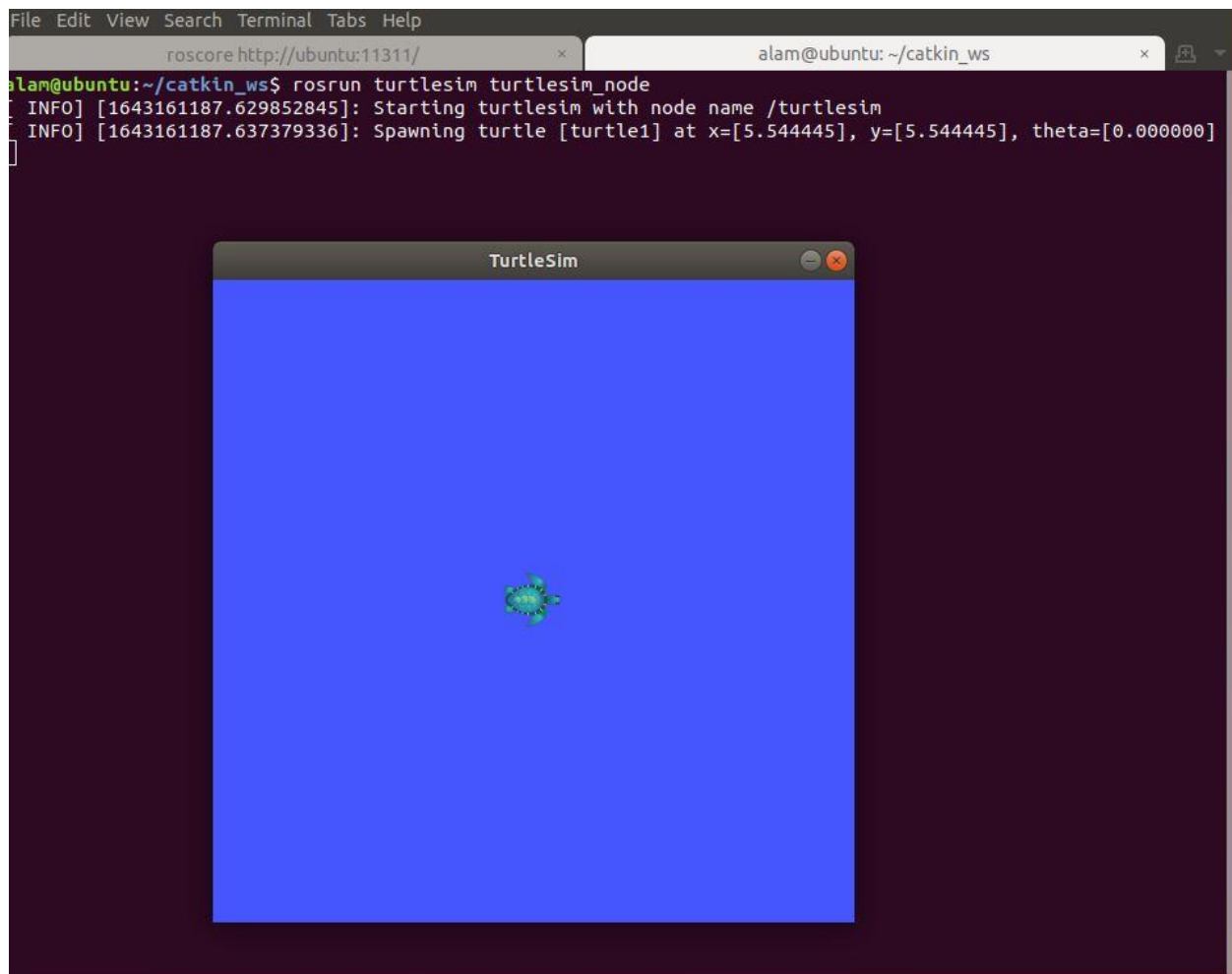
Student ID: 21060007

TASK 1

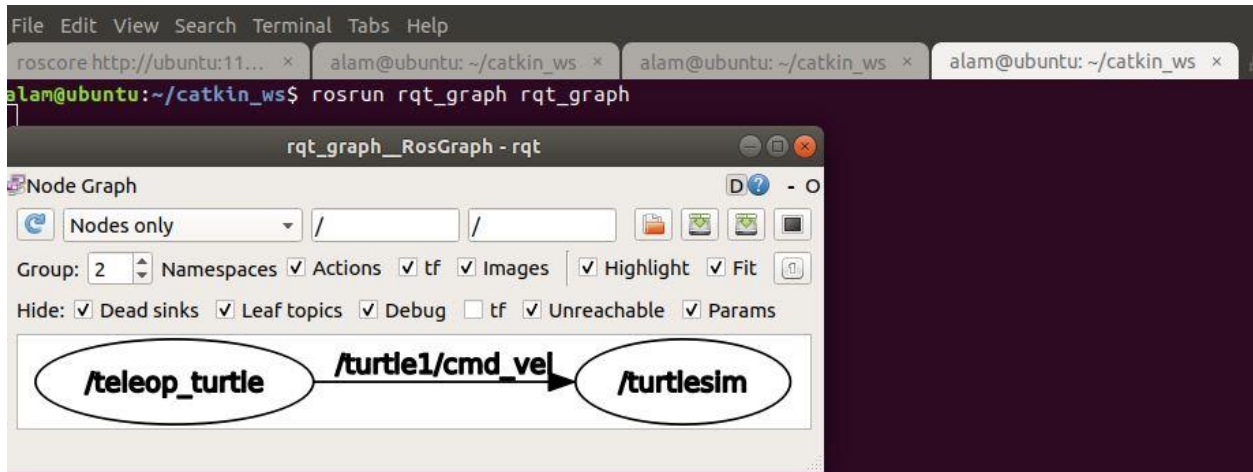
1. Setup catkin workspace on your assigned lab PC or personal laptop

```
alam@ubuntu:~/catkin_ws$ catkin_make
Base path: /home/alam/catkin_ws
Source space: /home/alam/catkin_ws/src
Build space: /home/alam/catkin_ws/build
Devel space: /home/alam/catkin_ws/devel
Install space: /home/alam/catkin_ws/install
Creating symlink "/home/alam/catkin_ws/src/CMakeLists.txt" pointing to "/opt/ros/melodic/share/catkin/cmake/toplevel.cmake"
####
#### Running command: "cmake /home/alam/catkin_ws/src -DCATKIN_DEVEL_PREFIX=/home/alam/catkin_ws/devel -D
CMAKE_INSTALL_PREFIX=/home/alam/catkin_ws/install -G Unix Makefiles" in "/home/alam/catkin_ws/build"
####
-- The C compiler identification is GNU 7.5.0
-- The CXX compiler identification is GNU 7.5.0
-- Check for working C compiler: /usr/bin/cc
-- Check for working C compiler: /usr/bin/cc -- works
-- Detecting C compiler ABI info
-- Detecting C compiler ABI info - done
-- Detecting C compile features
-- Detecting C compile features - done
-- Check for working CXX compiler: /usr/bin/c++
-- Check for working CXX compiler: /usr/bin/c++ -- works
-- Detecting CXX compiler ABI info
-- Detecting CXX compiler ABI info - done
-- Detecting CXX compile features
-- Detecting CXX compile features - done
-- Using CATKIN_DEVEL_PREFIX: /home/alam/catkin_ws/devel
-- Using CMAKE_PREFIX_PATH: /opt/ros/melodic
-- This workspace overlays: /opt/ros/melodic
-- Found PythonInterp: /usr/bin/python2 (found suitable version "2.7.17", minimum required is "2")
-- Using PYTHON_EXECUTABLE: /usr/bin/python2
-- Using Debian Python package layout
-- Using empy: /usr/bin/empy
-- Using CATKIN_ENABLE_TESTING: ON
-- Call enable_testing()
-- Using CATKIN_TEST_RESULTS_DIR: /home/alam/catkin_ws/build/test_results
-- Found gtest sources under '/usr/src/gtest': gtests will be built
-- Found gmock sources under '/usr/src/gtest': gmock will be built
-- Found PythonInterp: /usr/bin/python2 (found version "2.7.17")
```

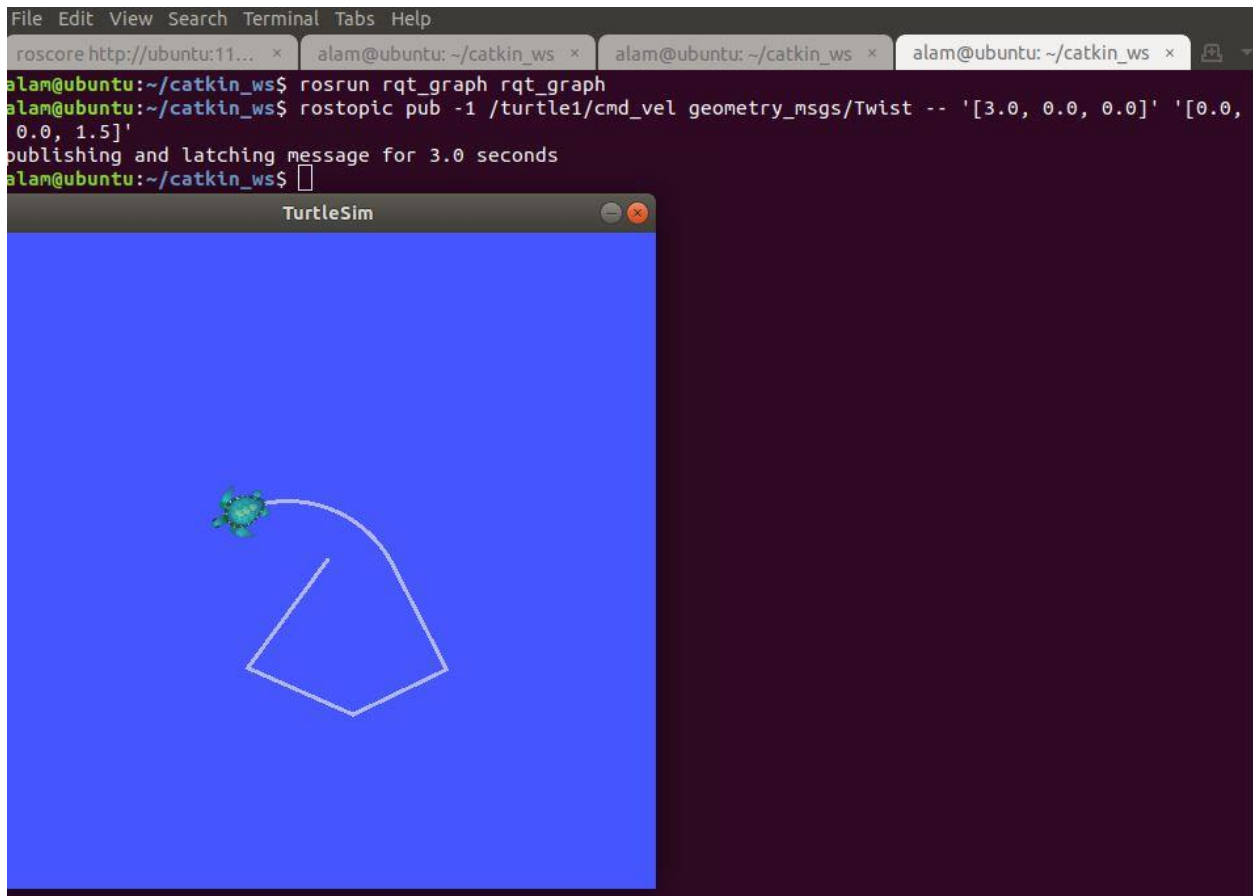
2. Run the ROS nodes `turtlesim_node` and `turtle_teleop_key` and demonstrate that you can control the movement of the turtle using your keyboard



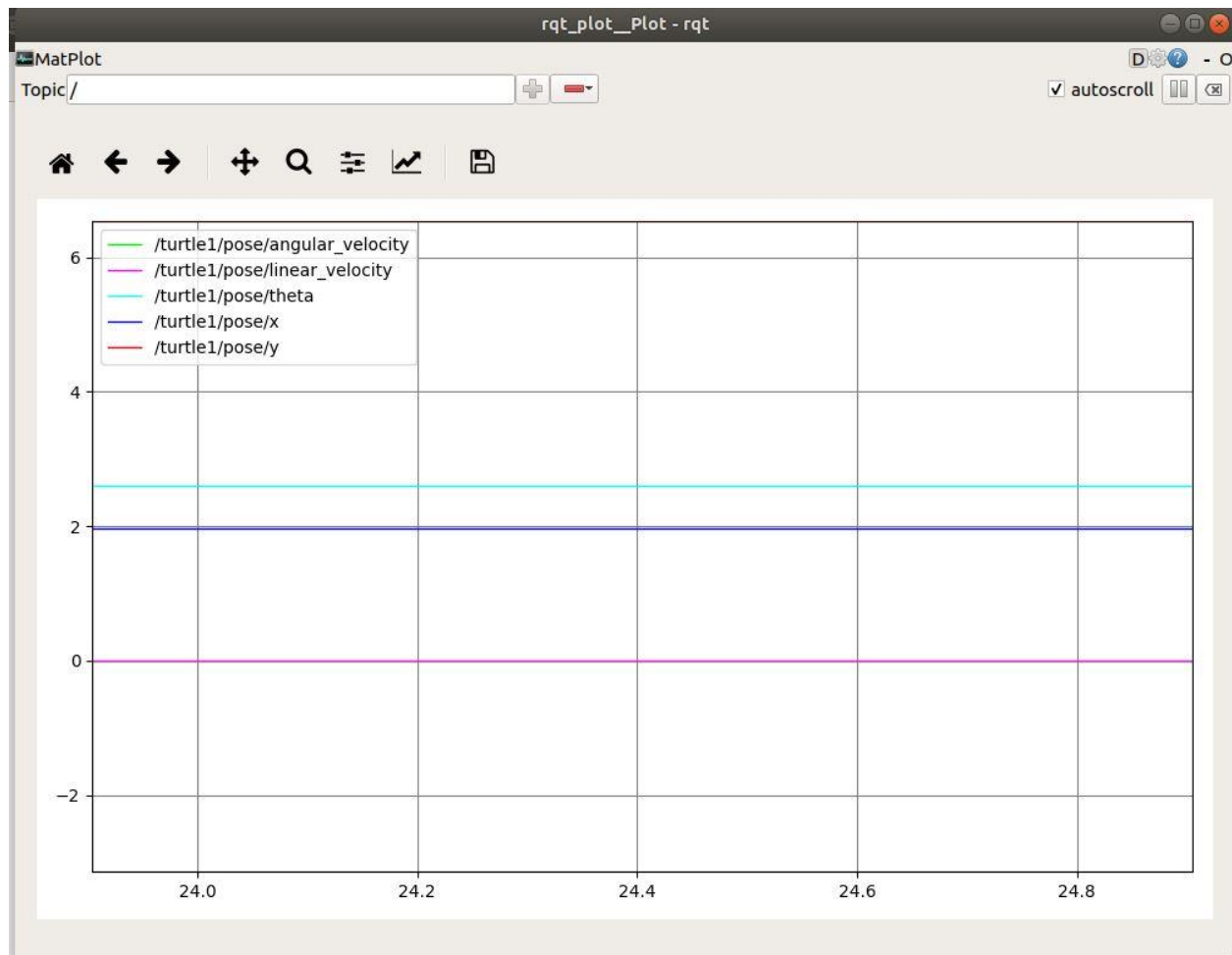
3. Display the dynamic graph of your ROS setup using `rqt_graph`, identify and topics of your system



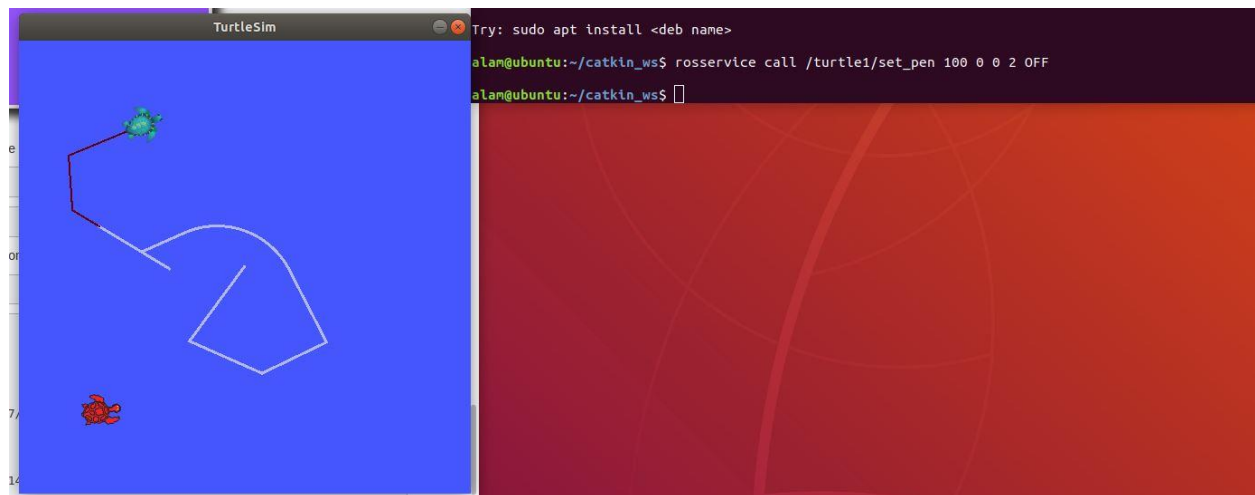
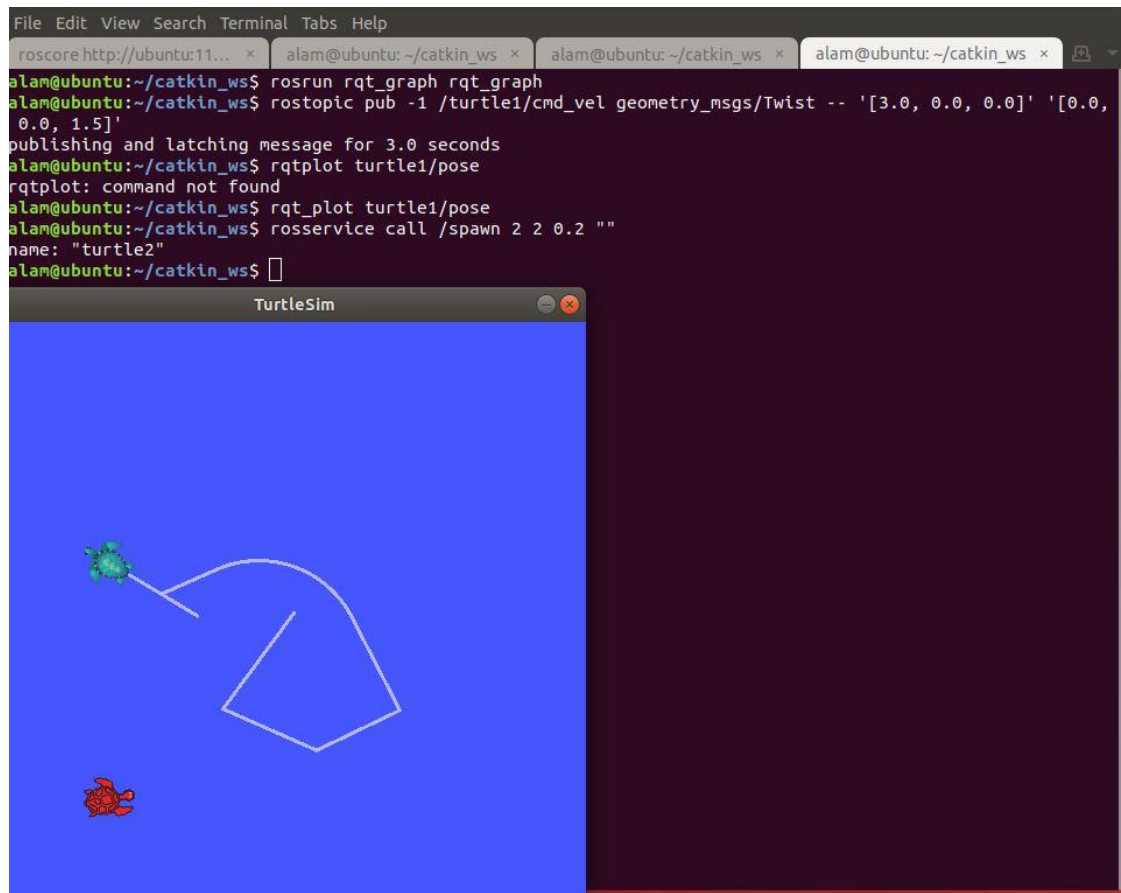
4. Command the turtle robot to move with a linear velocity of 3 and an angular velocity of 1.5. The command parameters should be set to ensure a continuous motion of the turtle.

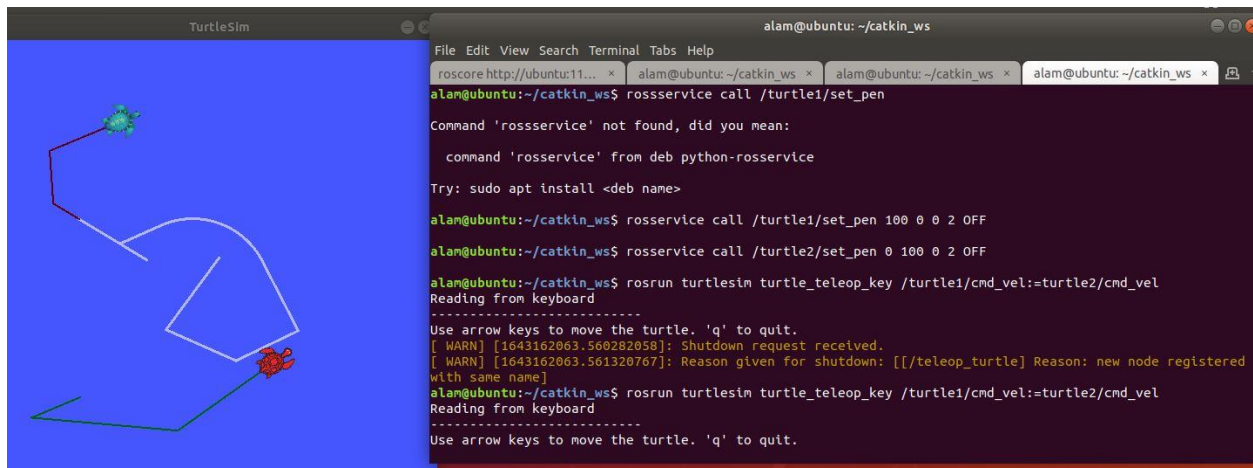


5. Observe turtle pose (x,y,theta) using rqt plot



Spawn two turtles at different locations. Set the first turtle's track line as red while the other one as green.





Task 2

1. Create a new ROS package “lab1_yourName” and configure it as a catkin package. You will have to use `catkin_create_pkg` command to create the package and `catkin_make` command to make it a catkin package.

```
alam@ubuntu:~/catkin_ws$ cd ~/catkin_ws/src
alam@ubuntu:~/catkin_ws/src$ catkin_create_pkg alam_lab1 std_msgs roscpp
Created file alam_lab1/CMakeLists.txt
Created file alam_lab1/package.xml
Created folder alam_lab1/include/alam_lab1
Created folder alam_lab1/src
Successfully created files in /home/alam/catkin_ws/src/alam_lab1. Please adjust the values in package.xml
.
alam@ubuntu:~/catkin_ws/src$ cd ~/catkin_ws
alam@ubuntu:~/catkin_ws$ catkin_make
Base path: /home/alam/catkin_ws
Source space: /home/alam/catkin_ws/src
Build space: /home/alam/catkin_ws/build
Devel space: /home/alam/catkin_ws/devel
Install space: /home/alam/catkin_ws/install
####
#### Running command: "cmake /home/alam/catkin_ws/src -DCATKIN_DEVEL_PREFIX=/home/alam/catkin_ws/devel -D
CMAKE_INSTALL_PREFIX=/home/alam/catkin_ws/install -G Unix Makefiles" in "/home/alam/catkin_ws/build"
####
-- Using CATKIN_DEVEL_PREFIX: /home/alam/catkin_ws/devel
-- Using CMAKE_PREFIX_PATH: /opt/ros/melodic
-- This workspace overlays: /opt/ros/melodic
```

2. Write a Launch File that should initialize two turtlesim nodes “turtle_one” and “turtle_two” such that turtle_one mimics the actions of turtle_two. Command the turtles to move with a linear velocity of 3 and an angular velocity of 1.5. The command parameters should be set to ensure a continuous motion of the turtle


```
lam@ubuntu:~/catkin_ws$ . ~/catkin_ws/devel/setup.bash
lam@ubuntu:~/catkin_ws$ roscd alam_lab1
lam@ubuntu:~/catkin_ws/src/alam_lab1$ mkdir launch
lam@ubuntu:~/catkin_ws/src/alam_lab1$ cd launch
lam@ubuntu:~/catkin_ws/src/alam_lab1/launch$ gedit

command 'geddit' not found, did you mean:

  command 'gedit' from deb gedit

Try: sudo apt install <deb name>

lam@ubuntu:~/catkin_ws/src/alam_lab1/launch$ gedit

gedit:3581): Gtk-WARNING **: 18:00:25.040: Failed to fetch network locations: Timeout was reached
lam@ubuntu:~/catkin_ws/src/alam_lab1/launch$ gedit
```

```
Open  turtlemimic.launch  Save  [Menu]  [Close]  [Exit]
~/catkin_ws/src/alam_lab1/launch

<launch>

  <group ns="turtlesim1">
    <node pkg="turtlesim" name="sim" type="turtlesim_node"/>
  </group>

  <group ns="turtlesim2">
    <node pkg="turtlesim" name="sim" type="turtlesim_node"/>
  </group>

  <node pkg="turtlesim" name="mimic" type="mimic">
    <remap from="input" to="turtlesim1/turtle1"/>
    <remap from="output" to="turtlesim2/turtle1"/>
  </node>
```

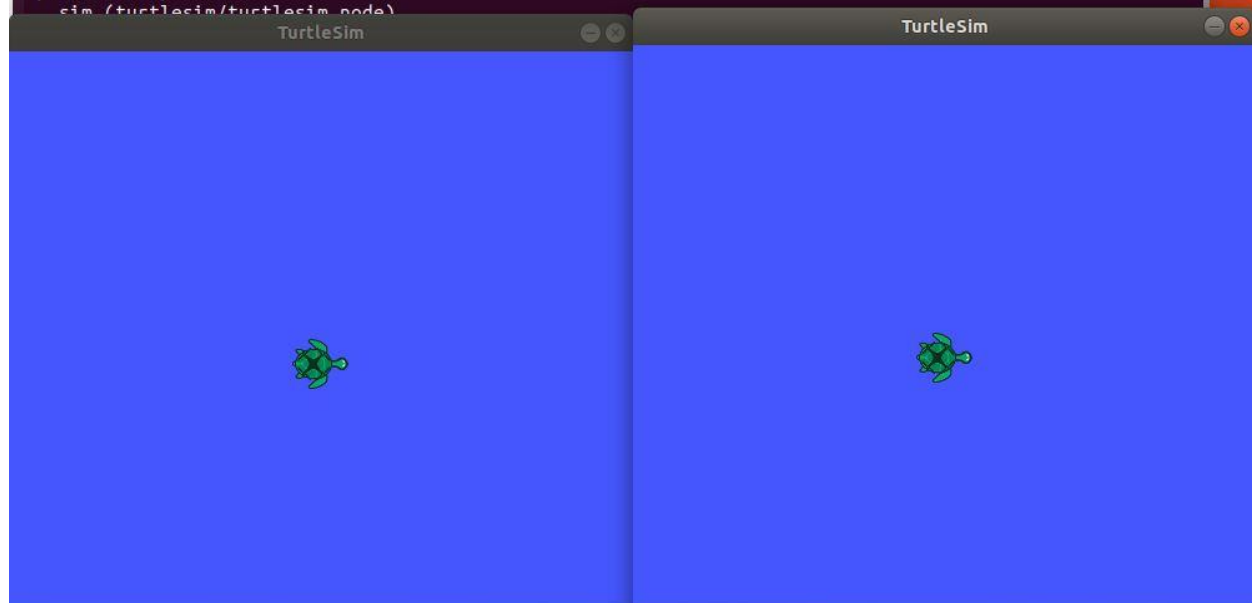
```
lam@ubuntu:~/catkin_ws/src/alam_lab1/launch$ roslaunch alam_lab1 turtlemimic.launch
... logging to /home/alam/.ros/log/cce0b632-7e48-11ec-9005-000c29a712d9/roslaunch-ubuntu-3736.log
Checking log directory for disk usage. This may take a while.
Press Ctrl-C to interrupt
Done checking log file disk usage. Usage is <1GB.

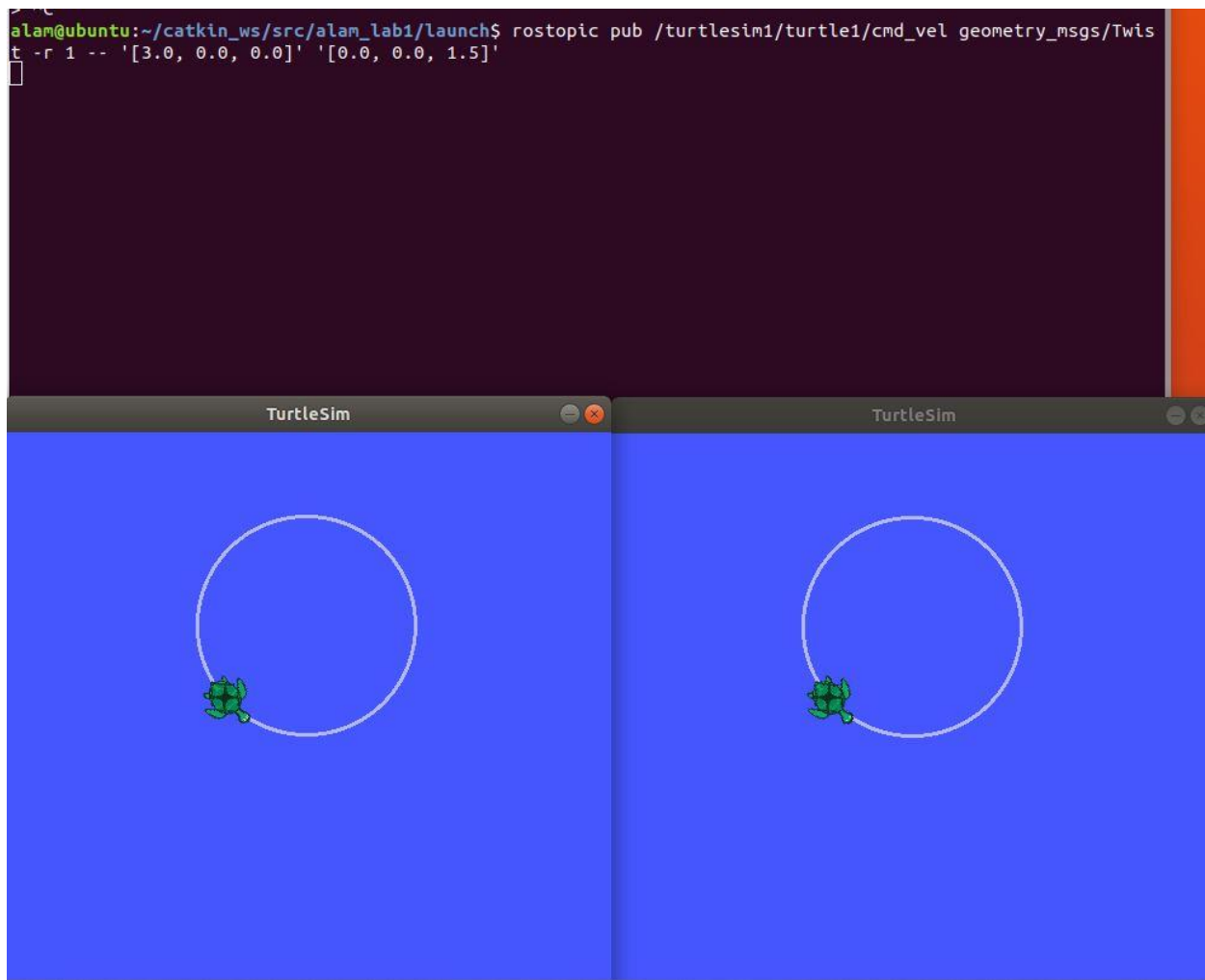
started roslaunch server http://ubuntu:42869/

SUMMARY
=====

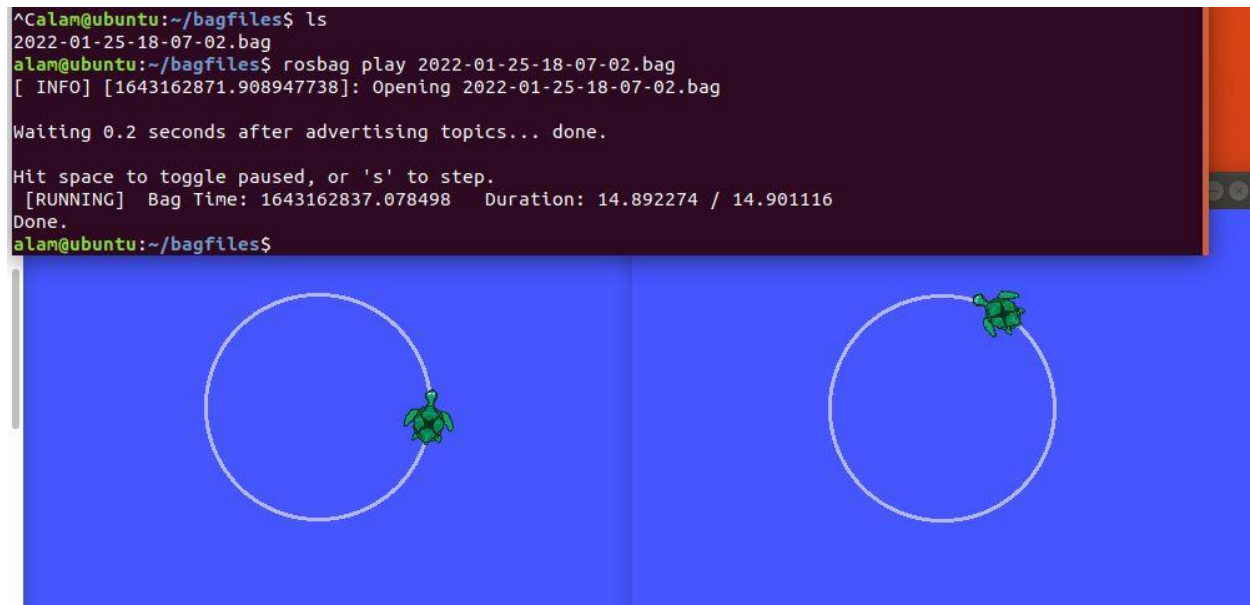
PARAMETERS
* /rostdistro: melodic
* /rosversion: 1.14.12

NODES
/
  mimic (turtlesim/mimic)
/turtlesim1/
  sim (turtlesim/turtlesim_node)
```





Store the turtle topics in ROS bag and you should be able to playback these ROS bag files.



Task 3

1. Create two python scripts labeled "publisher.py" and "subscriber.py" respectively.


```
^Calam@ubuntu:~/catkin_ws/src/alam_lab1/scripts$ rosrn alam_lab1 publisher.py
[INFO] [1643164629.975608]: 4
[INFO] [1643164630.476685]: 4
[INFO] [1643164630.976521]: 4
[INFO] [1643164631.476854]: 4
[INFO] [1643164631.975933]: 4
[INFO] [1643164632.476575]: 4
[INFO] [1643164632.976275]: 4
[INFO] [1643164633.476148]: 4
[INFO] [1643164633.976476]: 4
[INFO] [1643164634.475736]: 4
```

```
^Calam@ubuntu:~/catkin_ws/src/alam_lab1/scripts$ rosrn alam_lab1 subscriber.py
16
32
64
128
256
512
1024
2048
4096
8192
16384
32768
65536
131072
262144
524288
1048576
2097152
```

2. Create a new service for counting the number of words in a string. For this task, the first step will be to write a service-definition file

```
alam@ubuntu:~/catkin_ws/src/alam_lab1$ rosrn alam_lab1 word_count_server.py
Ready to count word
```

```
alam@ubuntu:~/catkin_ws/src/alam_lab1$ rosrn alam_lab1 service_client.py hello
how are you
hello how are you -> 4
```