Made by Sadaan Tahir - 25100224

EE 3002 L1 (Junior Design Studio - Robotics)

Spring 2025 - LAB 1

SOLUTION

Task 1: ROS 1.1 Beginner Tutorials [20 MARKS]

Just follow the steps in the tutorials.

Task 2: Custom Message Publisher and Subscriber in Python [30 MARKS]

Now, let us put our newfound ROS knowledge into practice by developing a custom message publisher and subscriber using Python.

2.1 Setting Up:

```
mkdir -p ~/your_name_ws/src
cd ~/your_name_ws/src
catkin_create_pkg lab1 rospy std_msgs
cd ~/your_name_ws
catkin_make
source devel/setup.bash
```

2.2 Custom Message:

Open the **CMakeLists.txt** file and **update/uncomment** these lines:

```
find_package(catkin REQUIRED COMPONENTS
  rospy
  std_msgs
  message_generation
)

add_message_files(
  FILES
  ComplexNumber.msg
)

generate_messages(
  DEPENDENCIES
  std_msgs
)

catkin_package(
  CATKIN_DEPENDS message_runtime
)
```

Next, in the **package.xml** file in the same directory, add these lines:

```
<build_depend>message_generation</build_depend>
<build_export_depend>message_generation</build_export_depend>
<exec_depend>message_runtime</exec_depend>
```

Lastly, write these commands in **terminal**:

```
cd ~/your_name_ws
catkin_make
source devel/setup.bash
```

2.3 Publisher:

Creating a **publisher.py** file under the scripts folder (if it doesn't exist for you, use mkdir scripts in the same directory to create the scripts folder):

```
cd ~/your_name_ws/src/lab1/scripts
touch publisher.py
chmod +x publisher.py
```

Code for **publisher.py**:

```
#!/usr/bin/env python3
import rospy
from lab1.msg import ComplexNumber

def publish_complex_number():
    pub = rospy.Publisher('/complex_numbers', ComplexNumber, queue_size=10)
    rospy.init_node('complex_number_publisher', anonymous=True)
    rate = rospy.Rate(1)  # 1 Hz

    while not rospy.is_shutdown():
        msg = ComplexNumber()
        msg.real = 3.0
        msg.imaginary = 4.0
        rospy.loginfo(f"Publishing: real={msg.real},
imaginary={msg.imaginary}")
        pub.publish(msg)
        rate.sleep()

if __name__ == '__main__':
    try:
        publish_complex_number()
    except rospy.ROSInterruptException:
        pass
```

2.4 Subscriber:

Similarly, creating a **subscriber.py** python file:

```
cd ~/your_name_ws/src/lab1/scripts
touch subscriber.py
chmod +x subscriber.py
```

Code for **subscriber.py**:

```
#!/usr/bin/env python3
import rospy
from lab1.msg import ComplexNumber
import math

def callback(msg):
    magnitude = math.sqrt(msg.real ** 2 + msg.imaginary ** 2)
    phase = math.atan2(msg.imaginary, msg.real)
    rospy.loginfo(f"Received: real={msg.real}, imaginary={msg.imaginary}")
    rospy.loginfo(f"Magnitude: {magnitude}, Phase: {phase}")

def subscribe_complex_numbers():
    rospy.init_node('complex_number_subscriber', anonymous=True)
    rospy.Subscriber('/complex_numbers', ComplexNumber, callback)
    rospy.spin()

if __name__ == '__main__':
    try:
        subscribe_complex_numbers()
    except rospy.ROSInterruptException:
        pass
```