Section 3 Write-Up: Group 17

Team members: Toni Divic, Emma Schneider, Aliyah Smith, Mason Llewellyn

Problem 1:

```
group17@genbu:~$ rostopic list
/camera/depth/camera_info
/camera/depth/image_raw
/camera/depth/points
/camera/parameter_descriptions
/camera/parameter_updates
/camera/rgb/camera_info
/camera/rgb/image_raw
/camera/rgb/image_raw/compressed
/camera/rgb/image_raw/compressed/parameter_descriptions
/camera/rgb/image_raw/compressed/parameter_updates
/camera/rgb/image_raw/compressedDepth
/camera/rgb/image_raw/compressedDepth/parameter_descriptions
/camera/rgb/image_raw/compressedDepth/parameter_updates
/camera/rgb/image_raw/theora
/camera/rgb/image_raw/theora/parameter_descriptions
/camera/rgb/image_raw/theora/parameter_updates
/clock
/cmd_vel
/gazebo/link_states
/gazebo/model_states
/gazebo/parameter_descriptions
/gazebo/parameter_updates
/gazebo/set_link_state
/gazebo/set_model_state
/gazebo_gui/parameter_descriptions
/gazebo_gui/parameter_updates
/imu
/joint_states
/odom
/rosout
/rosout_agg
/scan
/tf
```

Problem 2:

Message type is nav_msgs/Odometry. The information contained is *pose* (including position, orientation, and covariance) and *twist* (including linear, angular, and covariance)

Problem 3:

vel_publisher.py

```
import rospy
from geometry_msgs.msg import Twist

def publisher():
    pub = rospy.Publisher('cmd_vel', Twist, queue_size=10)
    rospy.init_node('vel_node', anonymous=True)
    rate = rospy.Rate(1)
    while not rospy.is_shutdown():
        twist = Twist()
```
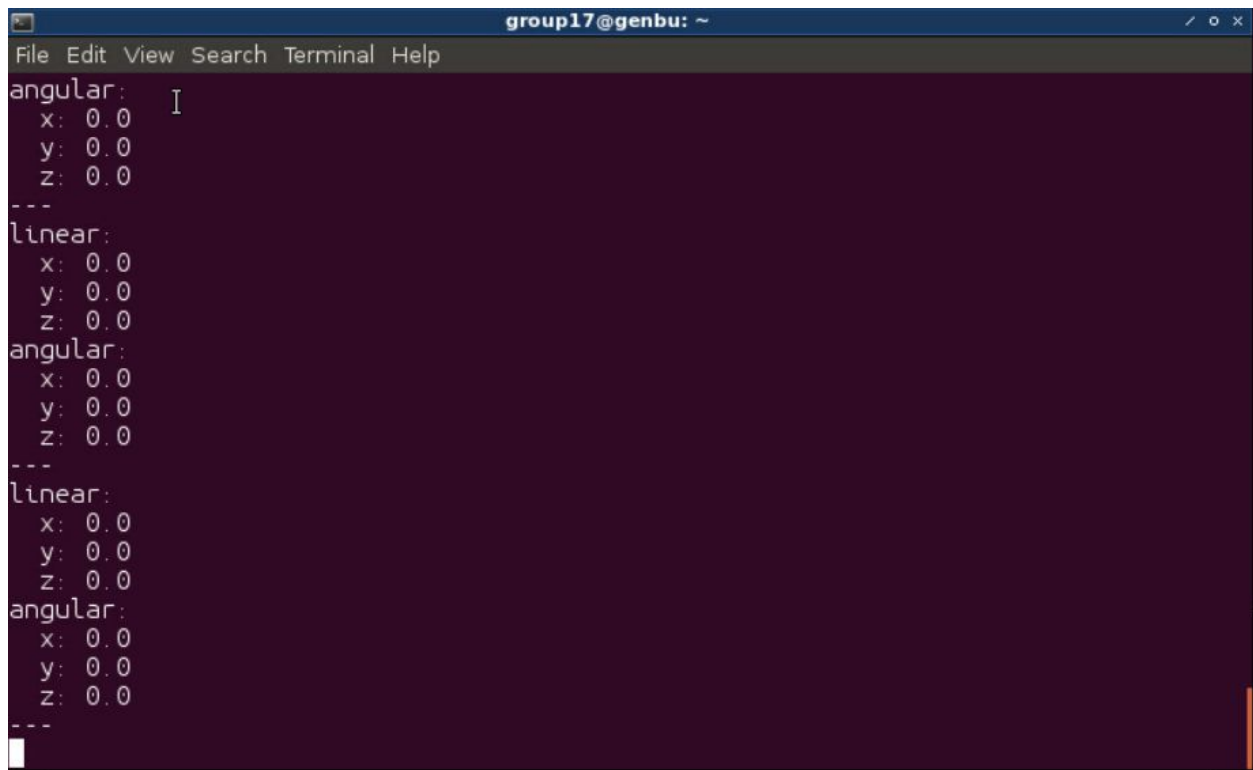
```
            twist.linear.x = 0; twist.linear.y = 0; twist.linear.z = 0
            twist.angular.x = 0; twist.angular.y = 0; twist.angular.z = 0
            pub.publish(twist)
            rate.sleep()


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

Output (via rostopic echo /cmd_vel):

Output (from python vel_publisher.py window):

Problem 4:

Odometry_subscriber.py
```python
import rospy
from nav_msgs.msg import Odometry

def callback(data):
    rospy.loginfo(data.pose)
    rospy.loginfo(data.twist)

def subscriber():
    rospy.init_node('odom_subscriber', anonymous=True)
    rospy.Subscriber("/odom", Odometry, callback)
    rospy.spin()

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

Output:

```
File  Edit  View  Search  Terminal  Help
0, 0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.
0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0]
[INFO] [1602095413.412255, 2798.210000]: pose:
  position:
    x:  -0.144139267082
    y:  0.163588859665
    z:  -0.00100766690611
  orientation:
    x:  -0.00057344269331
    y:  -0.00148317756357
    z:  0.360760014275
    w:  -0.932657323698
covariance: [1e-05, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1e-05, 0.0, 0.0, 0.0, 0.0, 0.0
, 0.0, 1000000000000.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1000000000000.0, 0.0, 0.0,
 0.0, 0.0, 0.0, 0.0, 1000000000000.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.001]
[INFO] [1602095413.414133, 2798.211000]: twist:
  linear:
    x:  -8.04176464139e-06
    y:  -3.11502743014e-07
    z:  0.0
  angular:
    x:  0.0
    y:  0.0
    z:  6.11559537492e-05
covariance: [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.
0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.
0, 0.0, 0.0, 0.0, 0.0, 0.0]
```