Lab Report 1

Daniel Engelsman - 300546173

Tom Smadar - 203374061

Question 2

Part (a)

To find the Pioneer1 robot default topics, we used the following command in Matlab:

>> rosinit
The value of the ROS_MASTER_URI environment variable, http://localhost:11311, will be used to connect to the ROS master.
Initializing global node /matlab_global_node_17123 with NodeURI http://aevadim-08:35744/
>> rostopic list

And got the following (Highlighted in green are the default topics for Pioneer1):

```
/clock
/gazebo/link_states
/gazebo/model_states
/gazebo/parameter_descriptions
/gazebo/parameter_updates
/gazebo/set_link_state
/gazebo/set_model_state
/pioneer1/cmd_vel
/pioneer1/joint_states
/pioneer1/laser/scan
/pioneer1/odom
/pioneer1/pioneer_gazebo_keyop/motor_power
/pioneer1/pioneer_gazebo_keyop/teleop
/pioneer2/cmd_vel
/pioneer2/joint_states
/pioneer2/laser/scan
/pioneer2/odom
/pioneer2/pioneer_gazebo_keyop/motor_power
/pioneer2/pioneer_gazebo_keyop/teleop
/rosout
/rosout_agg
/tf
/tf_static
```

Part (b)

We've created a publisher so we could sent velocity command to the Pioneer1 robot. First, we checked the message type we'll need to send to the corresponding topic we examined in part (a):

```
>> rostopic info /pioneer1/cmd_vel
Type: geometry_msgs/Twist

Publishers:
* /pioneer1/pioneer_gazebo_keyop (http://aevadim-08:43102/)

Subscribers:
* /gazebo (http://aevadim-08:44280/)

And created a publisher:

>> speedcom = rospublisher('pioneer1/cmd_vel', 'geometry_msgs/Twist')

speedcom =

  Publisher with properties:

      TopicName: '/pioneer1/cmd_vel'
      IsLatching: 1
      NumSubscribers: 0
      MessageType: 'geometry_msgs/Twist'
```

Part (c)

To receive odometry readings, we'll need a subscription to the topic in which they are published. We found that topic from Part (a), and checked what kind of messages it uses:

```
>> rostopic info /pioneer1/odom
Type: nav_msgs/Odometry

Publishers:
* /gazebo (http://aevadim-08:44280/)

Subscribers:
```

We've created a subscriber:

```
>> odoread = rossubscriber('pioneer1/odom' , 'nav_msgs/Odometry')
odoread =

  Subscriber with properties:

      TopicName: '/pioneer1/odom'
      MessageType: 'nav_msgs/Odometry'
      LatestMessage: [0×1 Odometry]
      BufferSize: 1
      NewMessageFcn: []
```

Question 3

Part (a)

Hereafter is the structure of the Pioneer ROS velocity message (explanations are highlighted)

```
>> rosmsg info geometry_msgs/Twist
% This expresses velocity in free space broken into its Linear and Angular parts.
Vector3  Linear - Describes linear velocity
Vector3  Angular - Describes angular velocity
```

And in greater detail:

```
>> v1 = rosmessage('geometry_msgs/Twist');

>> showdetails(v1)

  Linear
      X :  0 - Linear velocity in x-axis (body)
      Y :  0 - Linear velocity in y-axis (body)
      Z :  0 - Linear velocity in z-axis (body)
  Angular
      X :  0 - Angular velocity in x-axis (body)
      Y :  0 - Angular velocity in y-axis (body)
      Z :  0 - Angular velocity in z-axis (body)
```

Part (b)

To describe how this message would look, we've converted to conventional units and inserted the data into a velocity-message structure:

```
>> v1.Angular.Z = degtorad(2)

v1 =

  ROS Twist message with properties:

       MessageType: 'geometry_msgs/Twist'
       Linear: [1×1 Vector3]
       Angular: [1×1 Vector3]

  Use showdetails to show the contents of the message

>> v1.Linear.X = 1*5/18

v1 =

  ROS Twist message with properties:

       MessageType: 'geometry_msgs/Twist'
       Linear: [1×1 Vector3]
       Angular: [1×1 Vector3]

  Use showdetails to show the contents of the message
```

And so, it would look like this:

```
>> showdetails(v1)

  Linear
       X :  0.2777777777777778
       Y :  0
       Z :  0
  Angular
       X :  0
       Y :  0
       Z :  0.03490658503988659
```

Question 4

Part (a)

The structure of a Pioneer ROS Odometry message is as follows (Explanations are highlighted):

```
 ChildFrameId :  - Defines the frame of reference
  Header    - Message header, includes several characteristics such as timestamps etc.
       Seq    :  100942
       FrameId :
       Stamp
       Sec  :  0
       Nsec :  0
  Pose          - Defines the pose of the robot
       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]  - Defines the uncertainty of the pose
       Pose
       Position       - Position of the robot in its defined frame of reference [m]
       X :  0
       Y :  0
       Z :  0
       Orientation     - Orientation of the robot in its defined frame of reference [quaternion]
       X :  0
       Y :  0
       Z :  0
       W :  0
  Twist   - Defines the robot's velocity (similar to the explanation in Question 3)
       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]
       Twist
       Linear
       X :  0
       Y :  0
       Z :  0
       Angular
       X :  0
       Y :  0
       Z :  0
```

Part (b)

To describe how this message would look, we've converted to conventional units (also from Euler angles to quaternion) and inserted the data into an odometry-message structure:

```
>> od1.Pose.Pose.Position.X = 10;
>> od1.Pose.Pose.Position.Y = 15;
>> od1.Pose.Pose.Position.Z = 0.5;
>> ang = eul2quat([degtorad(30) 0 0])

ang =

       0.9659       0      0      0.2588

>> od1.Pose.Pose.Orientation.X = ang(1);
>> od1.Pose.Pose.Orientation.W = ang(4);
```

And the message would look like this:

```
>> showdetails(od1)
  ChildFrameId :
  Header
       Seq    :  100942
       FrameId :
       Stamp
       Sec  :  0
       Nsec :  0
  Pose
       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]
       Pose
       Position      - Here we can see the position data we've inserted
       X :  10
       Y :  15
       Z :  0.5
       Orientation     - Here we can see the orientation data we've inserted,
       X :  0.9659258262890683
       Y :  0
       Z :  0
       W :  0.2588190451025207
```

```
  Twist
       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]
       Twist
       Linear
       X :  0
       Y :  0
       Z :  0
       Angular
       X :  0
       Y :  0
       Z :  0
```