Practice Lab Questions

Valmik Prabhu

October 30, 2018

1 Core ROS

1. What's the difference between a topic, a service, and an action server? When would you use each?
2. What's the difference between a topic and a message?
3. `rostopic info /mobilebase/commands/velocity` says that the topic type is `geometry_msgs/Twist`. What is `geometry_msgs`? What is `Twist`?
4. What is a ros node?
5. What's the difference between rosrun and roslaunch?
6. What's the difference between a package and a workspace?
7. How should a ros workspace be organized? Where does each type of file go?
8. What do `package.xml` and `CMakeLists.txt` do?
9. How is a `.msg` file organized?

2 Lab 3/4

1. Imagine you are working on a Baxter. What would `rosrun tf tf_echo /base /base` print (paraphrasing is ok)?
2. You've designed a controller to make your state converge to some desired value. You are able to measure the full state. You implement the controller using the following pseudocode:

```
function controller (desired_q)
{
    while not measure_q() == desired_q
    {
        u = controller(measure_q(), desired_q)
        move_robot(u)
    }
}
```

   Will this function ever terminate? Why?

3. The homography matrix is

$$H = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \tag{1}$$

   where

$$\begin{bmatrix} \alpha u \\ \alpha v \\ \alpha \end{bmatrix} = H \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \tag{2}$$

Is the transformation from $[x, y]$ to $[u, v]$ linear or not. Why?

4. What benefit would a nonlinear homography matrix provide in comparison to an affine matrix (h_{31} and h_{32} in the matrix above are zero)?

5. What is an AR tag? Why might one use one?

3 Labs 5/6

1. How should you select which inverse kinematics solution to use from the set of possible IK solutions?

2. What is a URDF, and why is it used?

3. Does MoveIt return unique solutions for inverse kinematics or path planning? Why?

4. What is SLAM?

5. The kinect on a Turtlebot is broken, so you put a webcam on top of the robot to detect AR tags. You want to know the transform from the AR tag to the robot's base frame. What additional information must you provide to `tf2` in order to get this transform? What will your `tf` tree look like?

6. A proportional controller for a Turtlebot looks like this:

$$\begin{bmatrix} \dot{x} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} K_1 & 0 \\ 0 & K_2 \end{bmatrix} * \begin{bmatrix} x_d - x \\ y_d - y \end{bmatrix} \tag{3}$$

What signs should K_1 and K_2 be for the controller to converge?

7. You have a cart that can move back and forth on a track. The cart's position is x, and you want to bring it to the origin $x = 0$. You can control the system's acceleration $\ddot{x}$. You decide to design a proportional controller for the system, where $\ddot{x} = K_p * (x_d - x)$ Since $x_d = 0$, $\ddot{x} = -K_p x$. Assuming ideal conditions (no measurement error, no disturbances, and perfect actuation), draw the cart's x position over time. What changes might you make to this controller to improve performance?

4 Labs 7/8

1. Which ROS communication method do you use to plan with MoveIt?

2. What are potential uses of orientation constraints?

3. What's the difference between the open-loop controller you implemented in lab 3 and the one you implement in lab 7?

4. Why might you want a closed-loop controller rather than an open-loop controller?