CS 4758/6758: Robot Learning: Homework 1 (Basics)

Due: Feb 9, 2012 (In class)

1 Forward and Inverse Kinematics (60 pts)

Figure 1 shows the schematic of a robot arm lying in the $X-Y$ plane. The robot has three links each of length $l_1 = 80$ cm, $l_2 = 50$ cm, $l_3 = 20cm$. The first link of the robot is located at point (0,0) on the $X-Y$ plane. The angles at each of these joints are $\theta_1, \theta_2, \theta_3$.

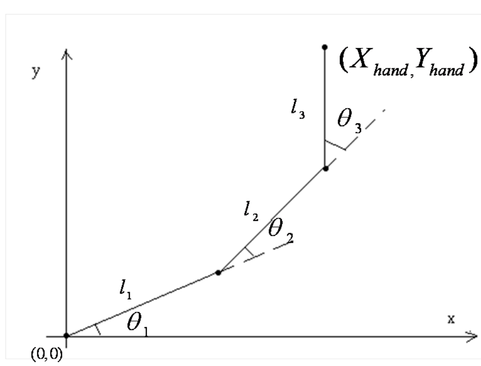
1. **Forward Kinematics 15 pts**: Given $\theta_1, \theta_2, \theta_3$, compute the position of robot's hand (X_{hand}, Y_{hand}). (a) Make rotation matrix[1] for each of these links. (b) Write the homogenous transformation matrices T_1^0, T_2^1 and T_3^2. (3x3 matrix: rotation matrix appended with translation) (c) Write the expression for obtaining (X_{hand}, Y_{hand}) as a function of homogenous tranformation matrices. (d) Evaluate the expression for $\theta_1 = 45^o, \theta_2 = 30^o, \theta_3 = 30^o$, and report the (X_{hand}, Y_{hand}).
(15 points)

2. **Inverse Kinematics (Programming assignment) 35 pts**: Given: (X_{hand}, Y_{hand}), find $\theta_1, \theta_2, \theta_3$ (there could be one solution, many solutions, or no solutions. In addition to the joint angles $\theta_1, \theta_2, \theta_3$ for cases with at least one solution, make sure to specify if there is one solution, multiple solutions, or no solutions. In case of multiple solutions, providing only one solution is sufficient) The angles are constrained to lie in the following ranges: $\theta_1 \in [-30, 30], \theta_2 \in [-30, 30], \theta_3 \in [-45, 45]$.

   Implement a simple inverse kinematics solver to compute poses for our system. You are not allowed to use an existing kinematics solver. You can use any algorithm, but it should be extensible to a different arm. For example, you can use nearest neighbor (with linear interpolation or gradient descent).

   Use your program to compute the angles for the (X_{hand}, Y_{hand}) given in the following file:
   http://www.cs.cornell.edu/courses/CS4758/2012sp/materials/hw1ik.txt
   Submit your code, together with the printout of the computed angles. Make sure that in the text, you also briefly explain your approach.

[1]Hint: Rotation matrix would be 2x2.



h

Figure 1: A planar robot arm with three links.

3. **Inverse Kinematics (ROS) 10 pts**:
We will now run an inverse kinematics package in ROS for the PR-2 robot. PR-2's robot arm consists of 7 degrees of freedom (specified as r_shoulder_pan, r_shoulder_lift,r_upper_arm_roll, r_elbow_flex, r_forearm_roll, r_wrist_flex, r_wrist_roll). Use the ros-electric-pr2-arm-navigation stack to solve for the joint angles given a desired position and orientation of the tip of the hand ($x = 0.75, y = -0.188, z = 0.0$ and $(0, 0, 0, 1)$ in quaternions). ROS, Gazebo and the PR-2 stack are all installed and ready to use on the PCs in the robotics lab (UPSON 317, UPSON 319)

2 Logistic Regression (40 pts)

Suppose we have $x \in \Re^2$, $y \in \{0, 1\}$ and a training set as shown.

x_1	x_2	y
-1	-1	0
-1	1	0
1	-1	1
1	1	1

Table 1: Training set

(a) If a logistic classifier is trained on these points, what is its decision boundary? Write an expression in terms of x_1 and x_2.(10 pts)

(b) Now add an extra point—$(100, 0)$ labelled as $y = 1$—to the training set . What is the decision boundary when the classifier is trained on all five points? Does it change? Justify. (12 pts)

(c) A lazy student decided to use linear instead of logistic regression for parts (a) and (b) above. What were his answers? How do they compare to the correct ones? Explain the difference. (18 pts)

Note: Use a bias term/intercept for linear regression. That is, the model should be of the form $\hat{y} = \beta_0 + \beta_1 x_1 + \beta_2 x_2$. Also, since linear regression makes predictions in $\Re$ instead of $\{0, 1\}$, label a point as 1 if $\hat{y} > .5$ and 0 otherwise.