CS4421.03
Assignment.

Due Date: Feb. 11.

Robotics Assignment.

1. Preliminaries

This lab is about statistics and robot arms; no surprise here. You are going to write some programs to hone your skills in statistics and to control the robots in the Lab. Here are a few things that you should know about the lab:

- It is accesible every day. Sometimes the robot will be booked for a labtest and will be unavailable.
- When you are done for the day, you turn off the robots, take the cables out of the way and logout.
- You have to follow all the reasonable safety precautions. Do not put your head (especially if you have long hair) within the range of any of the robots. Never use speed higher than 30% of the maximum (in fact during the debugging use much less that 30%). One precaution that is not obvious is that you never turn on the robots without being connected to the line that the robot is hooked. The reason is that somebody else might (by mistake or on purpose) take control of it remotely. If you connect to the port the operating system will not allow anybody else to connect. This is easier to happen than it seems.
- Avoid working in the lab alone. Undergraduate students have to work in pairs anyway. Graduates can ask each other for help with the cutoff switches etc.
- In case the robot malfunctions, turn the power off and send me mail. If I am around I can usually fix it if I have the spare parts in a couple of hours. Otherwise expect a one or two day (sometimes up to a week) downtime. There will be no extension for downtime since this is a fact of life when working with real hardware.

2. Simple Statistics

Read the file `varianceplay.mm` into MediaMath (if you want to work in another language, ask me for the source code and you can easily translate). This will give you access to a simple function *grand_vecs*. As you can see from the documentation it returns a set of random vectors (2-D by default) but we do not know the vital statistical properties of these vectors.

(1) Find the mean and variance of the vectors produced by this function. Use empirical methods (like running the function and taking averages etc) rather than looking at the source code. Automate this process by creating a few functions to do it.

(2) Create a function that takes as arguments a mean vector and a covariance matrix and some reasonable optional arguments and returns a set of vectors that have the givel mean and variance. (Hint: if you need to take the square root of a matrix then consider the function *chol_dcmp* that does Cholesky decomposition).

(3) Write a simple tool to visualize the vectors produced by the above function. (You can get ideas by looking at the files `EM_cluster_demo.mm` and `clust_util.mm` in the MediaMath directory)

(4) Generate two sets of 30 random vectors each with different variances. Use your visualization tool to display both sets (in different color) and their pairwise waited averages.

3. Forward and Inverse kinematics

Write a program that controls the robot arm. Solve the kinematics using the Denavit Hartenberg convention. The robot, under the control of your program, should then grab a little object from a known position on the table and place it on another known position on the table. The positions have to be specified in cartesian coordinates.

4. Kinematics on Papaer

Solve the forward and inverse kinematics of a robot wrist that has all parameters equal to zero, except for $\alpha_1 = \phi$ and $\alpha_2 = \phi - 180$. What is the workspace of the robot? Why would someone want to use such a wrist?

5. Understanding clouds

Write a small program that generates elipsoid clouds of random 2D vectors. The output, if the option `plot` is true is plotted on an image and the image displayed. If the option is not true, the default, then the vectors are returned as a 2xN matrix. The input is the expected value of the vectors, the length of the major and minor axis of the 3σ elipse and the angle of the major axis of the elipse with respect to the x-axis. The number of points [default: 100] as well as the displaying parameters are options.

Develop your program in three stages. In stage one, create and display circular clouds. Make sure you have the variance right. In stage two, create eliptical clouds aligned with the axes and in the third include arbitrary rotation.

6. Guidelines for Programs

These programs are not that difficult once you have become familiar with the routines to move the robot and read the sonar. You can use C, MediaMath or anything else for your programming. You have to demo the assignment to the instructor and hand in documentation and a listing of the program. The program should be as professional as possible: comments brief and to the point, on line documentation, appropriate headers, meaningful names, reasonable command line options, well structured etc. The documentation should be short and to the point and cover the program design as well as the relavant math.

7. Documentation

A soft copy of MediaMath manual is in the course web page. If you have any problems with MediaMath (bug report, undocumented behavior etc) send mail to minas@cs.yorku.ca.