University of Washington

Computer Science and Engineering

Winter 2007

CSE 490 I: Design in Neurobotics

Lab 3 Date: 1/23/2007 or 1/25/2007

Write-up Due: 1/30/2007 or 2/1/2007

Submit electronically

Goals:

1. To learn using dataglove to control a robot

2. To understand how the simplest mapping works and how to make it work better

Assignment:

In this lab, you will use the dataglove to control the robot's movement. There will be two parts to this lab. In the first part, you will map each finger to each joint of the robot. You will be asked to execute some tasks with this mapping. In the second part, you will be asked to make this program a little bit smarter (without changing the mapping) so that you can execute tasks better.

Mapping each finger to each joint of the robot.

1. Define the motion range of each individual finger of your dataglove, and the motion range of each robot joint. Refer to the "detailed information" section below for this task.

2. Define the one-to-one linear mappings between the five glove fingers and the five robot joints. There are four different combinations as listed.

1. Finger1 (F1))->Joint 1 (J1), F2 -> J2, ..., F5 -> J5, with min mapping to min, max mapping to max

2. F1 -> J5, F2 -> J4, ..., F 5 -> J1

3. same as b), only with min/max flipped (so F1 max maps to J5 min, etc)

4. same as a), only with min/max flipped (so F1 max maps to J1 min, etc)

3. Write a C++ code to implement the mappings in 2. Takes the glove's sensor readings, calculate the corresponding robot joint angle based on the linear mapping, and send out the command to the robot. Once done, execute the program and test if you can move the robot around reasonably. You should try all the

four mappings and pick the one you like the best.

4. With the selected mode in 3, pick up the screw on the table and drop it into the cup. Record this trial into YourLastnameMap.txt (record at 20Hz, 5 columns of glove sensor data and 5 columns of robot data (so 10 columns total)). Record different trials into separated text files (e.g., YourLastNameMap2.txt). Make sure both partners get to do this. The goal is not to execute this (or learn to execute this) perfectly, but to understand the advantage and disadvantage of this mapping.

Same mapping, only smarter

5. Now be creative. Without changing the mapping you selected above, add a few extra features in your program to make it easier to execute the task. For example, you probably noticed that you cannot move your ring finger too independently from other fingers resulting in other joints to move at the same time. Think of a way to fix this problem without changing the finger to joint mapping. You can also adjust the ranges of some fingers/joints for a better control.

You should come up with minimum of three ideas that improve the chosen mapping. Implement them. For each idea, record the trial into a file named YourLastnameSmarteri.txt (i =1, 2, 3, ...). You should end up with at least three of these files per person. You can use your partner's and/or your own smarter algorithms.

Detailed Information:

A. Download the updated interface code for the dataglove and the robot. The changes include:

1. A new protected member of class Lynx is added, which is of type Lynx::LynxRange, as defined inside the class declaration. It restrains the range of the movement of robot arm joints.

2. This class member is to store the safe ranges of the robot joints. It can be accessed by called getRange() and setRange() class methods. It is initialized in the class Lynx initialization method Lynx::Lynx() in Lynx.cpp.

3. Read C below very carefully BEFORE you change the values in Lynx::Lynx() or call setRange().

4. You can call the new class method demo() to see the current setting for the joints' ranges.

5. In class Glove, the offset field in Glove::GloveCalib is NO longer informative. You still call calibrate() to perform calibration. Use getCalib() and setCalib() to access the minimum and maximum of sensor readings after calibration.

6. You'll notice that the readings from the dataglove sensors look different from the previous labs. Before the readings are mapped to a [0, 255] range. Now they are the raw data within [min, max] defined through calibration.

B. For the dataglove, you want to define the motion range so that your fingers can reach it comfortably all the time. So you do NOT want to extend your fingers to extreme during calibration. Note this is different from what you did in lab 1.

C. For the robot arm, you want to define the range of the joints so that none of the robot joints will exceed its physical limits, or make the robot to be unstable. We suggested a very strict range in Lynx::Lynx() (see A.b). Use them for your first trial of the control mapping. AFTER you feel comfortable with the mapping between your dataglove motion, and the motion of the robot joints, you can adjust these ranges for better control. If you do so, for safety, decrease the final range you choose to its 90%. So, say the range is [jmin, jmax], adjust it to a safer range as [jmin+(jmax-jmin)*0.05, jmax-(jmax-jmin)*0.05].

D. From the information you gather in B and C, fill out the following table for your record.

Table 1: Sensor/Joint Range

	Sensor/Joint 1		Sensor/Joint 2		Sensor/Joint 3		Sensor/Joint 4		Sensor/Joint 5	
	Min	Max	Min	Max	Min	Max	Min	Max	Min	Max
Dataglove										
Robot										

E. If you use modest finger motion while calibrating the dataglove, the calibration should be stable; this allows you to take off the dataglove and put it back on multiple times without the need for recalibration.

F. The motion ranges of the dataglove fingers will be different for users. Thus, each partner in your group should fill in her(his) own Table 1.

Questions:

Please submit individually. However, you CAN work together with your partner and submit the same solutions as your partner's. Write your partner's name on top of your assignment.

I. Plot all your (but not your lab partner's) .txt files saved from the lab in the following format. The suggested Matlab function for each step is colored light blue.

    i. Split the page into two plots

    ii. In the top subplot(subplot(2,1,1)), Plot five finger data with different colors (with legend box), against time in seconds.

    iii. In the bottom subplot(subplot(2,1,2)), Plot five robot data with different colors (with legend box), against time in seconds.

    iv. The time along x-axis should be an array with the same length as your trace data, with the interval 1/20 seconds, because the data is recorded at 20HZ. To generate such an array with length, say 10, use command "timestep=0.05 : 0.05 : 0.05*10; ", or "timestep = (1:10) * 0.05; ".

    v. Label the x and y axes (xlabel/ylabel).

    vi. Put an appropriate title on the top figure so that the graders can tell what this trial was about. (title)

    vii. Paste these Matlab figures into word (or any editor) and make notations on the graph with arrows to describe what happened. Simple thing you can do is to mark on the robot graph where the screw was successfully grasped or dropped into the cup. Also mark as many events as you remember and worth noting on the dataglove graph (such as couldn't hold the index finger in position any more, etc.). Include information that differentiates one trial to another (such as learning, effect of changes you made, etc). Aim to have 3-5 annotations per figure.

II. Write down the specific mapping you ended up using in 3 and explain your choice (in 2 – 3 sentences).

III. Write down 2-3 reasons that made it difficult to execute the task in 4.

IV. Write down 3 or more improvements you made in 5, and describe what you did, why you did it, and the effect of it. Make sure to make the effect of these changes in your plots.

V. What are other techniques you would have tried to implement if you were given more time? List 2-3.

VI. Identify the total time (in seconds) it took to execute the task for all files you recorded. Plot these time in the order of trial in one figure (this is a crude learning curve). Explain the shape of this trend. How much of it is your own hand control learning and how much of is the actual algorithm improvement? Elaborate on this in 3-5 sentences.