Lab 0: Lab Orientation
August 29/30, 2007
BIOEN 5201-Introduction to Biomechanics
Instructor: Jeff Weiss
TA: Shawn Reese

Purpose

Making experimental measurements with a computer controlled system is important in nearly every scientific field. This lab is intended to introduce the student to the hardware and software that will be used during the semester for this purpose. This will include the use of a motion control system, force transducers and Labview data acquisition software. Additionally, a brief introduction to Matlab (which will be necessary for manipulating the experimental data obtained in future labs) will also be given.

Background

In order for any measurement to be made with a computer, a number of things must happen. First, the signal of interest (the output of a load cell, for example) must be “conditioned”, which generally refers to amplification and filtering. A conditioned signal is then received by a computer via a DAQ (data acquisition) device which converts analog signals to digital signals by using a built in AD (analogue to digital) converter. Finally, the digitized signal is saved and then manipulated via a specific piece of software written for that purpose. In this lab the signal conditioner and DAQ device are combined into one unit—the NI SL-2345 signal conditioner. Labview, the industry standard for DAQ applications, is the software that will be used. Once the data has been obtained, generally other software applications (such as Matlab or Excel) are then used to process the data.

![Diagram of laboratory setup](image)

Figure 1. Laboratory Setup for mini materials testing machine. Note how the software (Labview) is used to both receive and send signals to peripheral devices.
In addition to recording a signal, computers are also used to generate signals and control peripheral devices. The signals can often be generated using the same DAQ hardware that acquires signals. The number of peripheral devices that are controlled via a PC is nearly limitless, however common devices include signal generators, oscilloscopes, motion controllers and temperature regulators. Figure 1 shows a schematic of the computer controlled acquisition system that you will be using for several of the experiments in this laboratory course.

For this and the following labs, there are two primary pieces of equipment that will be used: a load cell and a linear actuator. A load cell is a device that outputs a voltage or current that is proportional to the applied load. A linear actuator is a device that generates linear displacements in response to an input signal. In this lab the linear actuator is a servo motor, which is an electric motor with a feedback loop that allows it to be placed in a very precise position.

The load cells used in this lab are (essentially) a strain gauge that is inserted into a Wheatstone bridge. The load cells are driven and amplified by the aforementioned NI signal conditioner. This device creates the proper excitation voltage (2.5 volts) and amplifies the output by a gain of 100. The load cells used in this lab (and in general) are a sensitive piece of equipment and great care should be exercised in their use. Handle them gently, and please DO NOT OVERLOAD THEM! The load cells that will be used are built to measure loads with a maximum force of 9.8 N (equivalent to 1 Kg) and will be damaged by forces greater than this.

The servo motor (which controls the linear displacement of the mini materials testing system) is driven by what is generally referred to as a motion control system. The system used for this lab consists of a servo motor driver (Axiom DV10) which generates the pulse required to move the motor through a certain arc, a universal interface (NI UMI-7764) which takes the computer output and converts it to a usable signal by the motor driver, and a software control package (Labview in this case).

As an introduction, this lab will be broken down into three components: a motion control exercise, force transduction exercise and a Matlab programming exercise. The motion control exercise will give the student a chance to familiarize themselves with the control of the mini materials testing apparatus, while the force transduction exercise will familiarize them with the use of the load cells and Labview. The Matlab programming exercise will be a simple introduction to Matlab and will demonstrate how to accomplish simple tasks.

Exercise I. Motion Control

Introduction
As previously discussed, the linear actuator used for these labs is a servo motor. A servo motor is motor in which current “pulses” or “steps” are sent into the motor in order to generate a specific angular rotation. Coupled with this motor is a so called encoder, which is a plate with small lines printed on it that are optically read. The encoder generates feedback that provides information on the current angular position of the motor shaft. Since servo motors are rotary devices, a mechanism is needed to convert the rotational motion into linear motion. In the system used in this lab, a screw system is used for this purpose. The servo motors for this lab can be controlled in two ways: Manual control and Labview control. Both methods will be explored in this exercise.

Method 1: Manual control

The first method you will use to control the servo motor is to use the Labview Measurement and Automation Explorer, which provides an interactive environment for testing peripheral devices. To use this tool, perform the following steps:

1. Make certain that the motor driver and NI controller are powered up.
2. Double click the desktop icon labeled “Measurement and Automation” to open the Measurement and Automation Explorer.
3. Move the mouse over the “+” next to the Devices and Interfaces icon (shown in Figure 2.). This will expand the selection to all the external devices that have been installed.
4. Click on the “+” by the PCI-7344 icon—this is the motion controller.
5. Click the “+” on the Interact icon, and then click the 1D Interactive icon. This will display the interactive environment that allows for the control of the servo motor.
6. Look over the different fields and get a feel for what options exist. Now set the operation mode to “relative” and enter a positive number (like 1000) into the target position box.
7. Hit “apply” and then the play icon. This will cause the motor to move 1000 steps (equal to ¼ rotation of the motor) in the positive direction, which in this case is down. **NOTE: When moving the clamp, it is important not to run it into the bottom of the apparatus, as this will damage the load cell. If it looks like the clamp will hit the bottom or something has gone wrong, press down the red emergency stop button.**
8. Now repeat, only putting -1000 as the target position. The clamp should now move up.
9. Play around with things a bit, varying the number and direction of steps taken. For the fun of it, hit the emergency stop button to convince yourself that it works. To get the system to run again, pull out the stop button.
Method II: Labview Control

The second method you will use to control the servo motor is with a Labview VI. VI stands for “virtual instrument”, and is a program that controls both data acquisition and peripheral devices. A Labview VI consists of two parts: a front panel, which is the user interface, and a block diagram, which is what controls the program (refer to Figure 2). You will not be required to write any of your own VIs for this course, however it is worthwhile to take a look at the block diagrams and get an idea of how Labview works. To access a motion control VI, perform the following steps:

1. On the desktop is a folder labeled “Biomech07”. Inside this is a folder titled “Lab1”—open it and then double click the file titled “motion_controlV1.vi”.
2. Once the VI is opened, go to the top and click on “window”, and then on “block diagram”. This will show the block diagram associated with the front panel. You can now close the block diagram, as this will not be needed.
3. To run the VI, click the play icon (see Figure 3) in the upper left hand corner. Now mess around with the program, moving the clamp up and down.

Figure 2. Screenshot of the NI Measurement and Automation Explorer for the Motion Controller. The important parts of the window are circled in red.
Exercise II. Load Cell Calibration

A load cell converts a mechanical load into an electrical signal. For these lab stations, a load cell is connected to the mini materials testing apparatus and is driven and amplified by the NI Signal Conditioner. As with the motion controller, there are two ways to access the Load cell—through the Measurement and Automation explorer and a Labview VI. For this exercise you will calibrate the load cell by hanging weights of different mass and then measuring the output voltage in the Measurement and Automation Explorer. To do this, perform the following steps:

1. If it is not open already, open the Measurement and Automation Explorer.
2. Go to the devices and interfaces icon and click the “+” symbol next to it. Now click on the PCI-6052 icon—this is the DAQ device that amplifies and digitizes the load cell signal.
3. Click on the “Test Panel” icon (See Figure 4.). This will create a window in which the output signal of the load cell (in Volts) will be displayed. If it is not already installed, screw in the weight attachment device (if you can’t find it, ask the TA). Lightly tug on the transducer a few times to see the response.
4. At this point the voltage should be around 0 volts with no load. If it is not, ask the instructor for help and he/she will assist you.
5. To obtain proper readings, the load cell must be “zeroed”. This is done by removing all weight from the transducer, taking the cover off of the signal conditioning unit and rotating the screw labeled “gain” with a screwdriver. Rotate this until a value of close to zero volts is obtained.
6. Now screw back in the weight attachment device and make a voltage reading. The weight attachment device weighs 32 grams. This will be your first data point.
7. Generate a number of data points by adding weights of varying mass to the load cell. You should start low (with say 50g) and work you way up to no more than 1000g total weight. Be sure to add the weight of the attachment device for each reading.
8. Now generate a plot in Excel (or some equivalent software) of force vs. voltage. Note: this requires you to convert mass to force via Newton’s second law, \( F=ma \).
9. Perform a linear regression and record the slope—this is the calibration constant for your transducer. Save this value, as you will use it in future labs.
10. Now open the vi named “force measureV1.vi” in the Lab 1 folder. Enter the calibration value and make a few measurements to ascertain that the force transducer is working properly.

![Figure 4. Screenshot of the NI Measurement and Automation Explorer for the Force Transducer. The voltage reading is circled in red.](image)

Exercise III. Introduction to Matlab

Introduction

Matlab is a powerful tool that facilitates both simple and complex data analysis. It allows for functions as simple as addition, more complicated tasks such as statistical analysis and even computationally intensive tasks such as image processing. One of the core strengths of Matlab is that it combines a command prompt (similar to that of MS-DOS) with a programming
development environment (similar to that of Visual Studio). From the command prompt simple calculations can be made, commands can be issued and values can be stored in memory. Using the development environment feature (called the m-file editor) one can write programs (called m-files) that can be compiled and run in a single step. In addition to the command window and m-file editor, there is also a file directory, help menu, workspace and command history. Figure 5 shows a screenshot of Matlab with all the windows open.

![Figure 5. A screenshot of Matlab with all windows open and docked on the screen.](image)

### Opening, closing and Docking Windows

Matlab allows the user to customize their user interface. Upon opening Matlab, the default windows are the command window, command history and file directories. To get use to adding and removing windows, perform the following steps:

1. Click on the small arrow on upper right hand corner of the file directory window to undock it. It is now a free floating window.
2. Now click on the arrow again to re-dock it. If you want you can click and drag the edges to adjust its size.
3. Now close the window to remove it completely.
4. To once again add the window, go to the top of the screen, click on Desktop→Current Directory. This will open the file directory and re-dock it in the window. In this fashion, play around with closing, opening, docking and undocking the different windows.

Using the Command Window

The command window is the most basic feature of Matlab. It allows the user to issue commands, enter variables and perform computations. For a brief demonstration of how it works, perform the following steps:

1. Type “1+1” and “\n”. *\n* indicates hitting the enter key, which causes the statement to be executed. For the remainder of the lab, the *\n* key will be implied after every statement.
2. Type in $2^2$. The carrot is the power function. Type in “sqrt(4)” to get the square root of 4, and “sin(4)” to get the sin of 4.
3. Now type a=2, b=1 and a+b. Notice how variable values are stored in the workspace memory.
4. Now type in a=[1,2,3] and M=[1,2,3;1,2,3;1,2,3]. You have just entered an array (a row vector in this case) and a matrix.
5. Type in c=a*M to perform matrix multiplication.
6. Now type in b=a'. This assigns the transpose of a to the variable b.
7. Now type in c=M*b to perform another matrix multiplication. To access the first entry in the vector c, type in c(1). Access the second and third entry by typing c(2) and c(3).
8. You can access the help menu from the command line. For example, type ‘help dot’. The result is the help listing for the dot product function.
9. At the command line, type “whos”. The output will be a list of all the variables currently stored in the workspace. Now type “clear”. This clears the memory. If you type “whos” again, there should be no variables in storage. To clear the screen, type “clc”.
10. Matlab has an extensive library of functions that can be accessed from the command line. Type “dot(a,b)” to compute the dot product of the arrays a and b that are stored in memory. If you can’t remember what a and b were, type them into the command prompt and hit enter, at which point their values will be displayed.

Using Matlab Help

Matlab has an excellent help feature. The help is searchable in three ways: by content, index and search. To explore all three functions, perform the following:

1. In the help navigator, click the contents tab.
2. Now scroll down to the “Matlab” icon, click on “getting started” and then click on “introduction”. This is the first place to start if you are clueless about Matlab.
3. Go to the index tab. Type in the word “dot”. The entry “dot” will pop up. Click on it and the help entry for the dot product will be displayed. If you know the name of the function you are interested in, this is the most useful feature in the help menu.
4. Now go to the search tab and type in “dot” and hit enter. The help listing for the dot product function pops up.

Writing M files

Matlab has its own programming language similar to that of C where you can write your own code to perform tasks. To introduce you to the concept of writing an m-file, perform the following steps:

1. Open a new m-file by going to the top of the screen and clicking on File→New→M-file.
2. On the first line type ‘hello world’. On the second line type a=[1,2,3], on the third line type b=[1,2,3] and on the fourth line type c=dot(a,b).
3. Now save the file as “test1.m”.
4. Finally, go to Debug and click on Run. This executes the m-file and displays the output in the command window.
5. Now go back to your m-file and add a “%” in front of ‘hello world’. The % deactivates the line of code, thus allowing you to write comments.
6. Now go back and add a semicolon to the end of each line and run it again. Notice how the semicolon suppresses the output from being printed to the command window.

Matlab File system

By default Matlab saves everything in a folder named “work”, which is in the Matlab program folder (which is generally in the root directory such as “C:\”). Spend a little time navigating around in the file directory to get used to how this works.

A simple Matlab Program

Let's say you have a data set that you want to manipulate and then plot—the calibration data for example. By performing the following steps you will generate an M-file that will do this for you:

1. Enter your calibration data as two vectors: mass and volts.
2. Now compute the force from the mass vector.
3. Type in “plot(volts,force)”—this will generate a line plot of the data.
4. Now type in fit= polyfit(volts,force1)—this performs a linear fit on the data.
5. Type in slope=fit(1), which is your slope.
6. Now run the file. The calibration value is the slope.

Figure 6 shows a screenshot of this simple m-file.
More Help on Matlab

This was a very basic introduction to Matlab. For more information on using Matlab, visit one of the following online tutorials:

http://www.engin.umich.edu/group/ctm/basic/basic.html

http://www.cyclismo.org/tutorial/matlab/

http://users.ece.gatech.edu/~bonnie/book/TUTORIAL/tutorial.html

Or better yet, buy a good book. One recommendation is the title Mastering Matlab 7, by Duane Hanselman and Bruce Littlefield. As always, your TA will be more than happy to help students in answering questions regarding Matlab, the lab experiments and the seminar.