

22. LABORATORY GUIDE

- The laboratory work will help enforce the concepts presented in this and previous courses.

- Various labs will require pre- or post-lab work.

- General rules include:

- Unless specified, work is to be done individually.

- All written work is to be clear and accurate.

22.1 Lab 1 - Introduction to Resources and Tutorials

- These tutorials prepare you to use computer and other resources throughout the semester.

22.1.1 Tutorial 1a - Creating Web Pages

- The general steps are:

1. Get a computer account on 'claymore.engineer.gvsu.edu' from Prof. Jack. This account will have a prototype web page that you can edit.
2. Go to a laboratory (EC 616), or home computer and run 'Netscape Communicator'. Go to 'claymore.engineer.gvsu.edu' and look for your account under 'students'. You should be able to find a page that starts with 'YOUR_NAME_GOES_HERE'.
3. In Netscape (with your home page showing), select 'edit' from the tool bar, or under 'file' select 'edit' or 'edit page'. You will be asked if you want to save the page. Create a 'temp' directory on the computer. This directory will be used to temporarily hold your web page files. Make sure that the files will be saved in the 'temp' directory, and then 'save' the files. An editor will start on the screen.
5. The editor behaves much like Microsoft Word, with some subtle differences. At this point add your name, and change your email address to your river account. You can change your email address by clicking on the email link, and then clicking on the chain link near the top of the screen.
6. To upload the changes you have made to the website, select 'publish'. You will need to indicate the file name as 'index.html', the destination as 'ftp://claymore.engineer.gvsu.edu/home/YOUR_NAME/public_html'. You will also

need to enter your user name and your password (DO NOT SAVE THE PASSWORD - SOMEBODY ELSE CAN GET ACCESS TO YOUR ACCOUNT).

You should see a message that indicates files have been uploaded successfully.

7. Use Netscape, not the editor, to see if the changes have occurred. Your changes may not show up on the browser. This is because Netscape does not reload pages every time to look at them. Pages are often stored for up to 1 month on the PC's hard drive, and reused when you look at them. There are two ways to update the screen before this time limit - click on the reload button.
8. Next we will add links to your home page. First, run Mathcad, and create a simple file, and then save it in the same folder/directory you saved. Use a file name that is all lower case such as 'test.mcd' - any upper case letters cause problems in Windows 95.
9. Get your home page back in the Netscape editor. Somewhere type the word 'GVSU'. Use the mouse to select what you just typed, and then click on the link button. For the link name enter 'http://www.gvsu.edu', and apply the change. This will now be a link to the Grand Valley home page. For your Mathcad file type something like 'Mathcad file', highlight it, and add a link to 'test.mcd'. This link will connect to your Mathcad file.
10. Publish the file, but first add the Mathcad file to the list of files at the bottom of the screen.
11. Test the page.

• Some tips are,

- Windows will not allow multiple applications to open the same file at the same time. If you seem to be having trouble opening a file, make sure it is not open in another application.
- As you add other files to your homepage, put them in the 'temp' directory. This will make all of the procedures simpler.
- Try to make your web pages small, and link them together. This will decrease download time and make browsers happier.
- Avoid using excessive images. Anything over 10K will make it very slow downloading over modem. Anything over 100K makes modem downloading painfully slow.
- When putting images on the web page use 'jpg' for photographic images, and 'gif' for line images. 'jpg' images can be compressed more than 'gif', but lines will become blurred.
- To link to other files or web pages there will be a 'link' command. If you want to add a file that is in your 'temp' directory, just put the name of the file in the 'URL' field.
- Watch upper/lower case. This is a major cause of web page problems. It is best to keep to lower case for all file names.

22.1.2 Tutorial 1b - Introduction to Mathcad, Working Model 2D and The Internet

- Objective:

Working Model 2D, Mathcad and the Internet will be used in this course. In some cases students have not been exposed to one or more of these software packages in the past. This session will be used as a refresher for those with little prior exposure, and as a tutorial for those with no experience.

- Theory:

Mathcad is a software package that allows us to do complex calculations both numerically and symbolically. To learn it initially will require a time investment. But, when doing calculations later, it will save a significant amount of time and reduce calculation errors.

Working Model 2D is a software package that allows us to set up systems of multiple rigid bodies. We can then apply forces, moments, etc. and then see how the system dynamics are effected. In comparison, Mathcad will allow longer, precise calculations, whereas Working Model allows faster results with reduced accuracy. Working Model also presents a visual simulation - this allows a more intuitive understanding of a dynamic system.

The Internet is a huge collection of computers providing information and connection on an unprecedented scale. It has become a standard business tool, and continues to evolve.

- Procedure

1. (If needed) Go over the Mathcad tutorial provided.
2. Use Mathcad to calculate the position of a ball that has been held then released just above the surface of the earth, and add the file to your home page.
3. (If needed) Go over the Working Model tutorial provided.
4. Repeat the problem solved in Mathcad with Working Model and add the file to your home page.
5. (If needed) Get a computer account set up, and create a home page.
6. Go to a search engine and find a website for a major business that is related to your co-op position and add a link to it on your home page. Explain how the business is related to your co-op position.

- Post-lab:

None

- Submit:

1. A Mathcad file linked to your home page.
2. A Working Model file linked to your home page.
3. An explanation and a link to a company on your home page.

22.1.3 Presentation 1a - Introduction to Library Searches

- Objective:

To prepare students to use the libraries resources in typical research studies.

- Theory:

The essential purpose of engineering is to apply principles of the arts and sciences to solve real problems. Scientific principles tend to evolve over time, but the essential principles and written works are valid for a number of decades or centuries. As a result, books can be excellent resources for this knowledge. The applications that make use of the basic principles tend to be more revolutionary. As a result, printed books have a value for teaching the fundamentals, but the 'state of the art' must often be found in magazines, journals, etc. To put this in simpler terms, when we look for scientific resources, we will often use sources over a decade old. When using engineering resources, most will be less than five years old. Until recently, print has been the major means of exchanging information, and libraries have been the traditional repositories of printed materials. To deal with the extensive number of publications available in a library, we need to learn how to search for needed information, and what resources are available.

New technology has changed access to library materials. Libraries pool resources and share materials. Internet technology has also helped increase accessibility. In particular internet tools allow the entire library catalog to be examined without visiting the library. There are also a number of resources that can be searched and retrieved over the internet.

- Procedure

1. A presentation will be made by Mr. Lee Lebin, the University Library Director.
2. Use the library resources to identify an application of systems modeling.

- Post-lab:

1. Search for library resource.

• Submit:

1. A copy of the material referenced.

22.2 Lab 2 - Computer Based Data Collection

22.2.1 Prelab 2a - Tutorial for LabVIEW Programming

Objective:

To learn the basic use of LabVIEW.

Theory:

To obtain the greatest computing power and flexibility we need to write computer programs. But, traditional programming languages are not well suited to designing user interfaces and dealing with data flows.

Most computer programs are written with lines of program and compiled to execute. LabVIEW allows you to “write” programs using graphical symbols. This graphical programming approach allows systems to be designed by connecting the symbols with "wires" (i.e., lines).

Equipment:

PC with LabVIEW software

Procedure:

1. Go through the LabVIEW QuickStart Guide provided in the laboratory. This will also be good review for those who have used LabVIEW in previous courses.
2. Write a Labview program that will count from 1 to 100, square the values, and print the results on a strip chart.

Marking:

1. The VI created should be posted to the web.

22.2.2 Prelab 2b - Overview of Labview and the DAQ Cards

- To obtain the greatest computing power and flexibility we need to write computer programs. But, traditional programming languages are not well suited to designing user interfaces and dealing with data flows.

- LabVIEW allows you to “write” programs using graphical symbols. This graphical programming approach allows systems to be designed by connecting the symbols with "wires" (i.e., lines).

- The remainder of the labs will focus on using LabVIEW to write programs to allow a computer to interact with the environment outside the computer.

- The computers we will use all have DAQ (Data Acquisition) boards - National Instruments PCI-1200 DAQ cards. These cards have capabilities that include:

- 24 I/O bits - TTL 0,5VDC, 20mA max.

- 8 single ended or 4 double ended analog inputs - 12 bits

- 3 counters - 16 bits

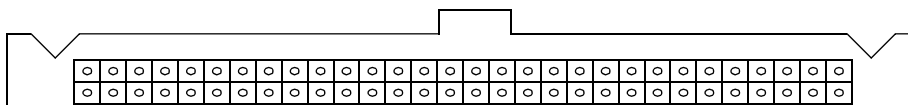
- 2 analog outputs - 12 bits

- The connector for the card can be found on the back of the computer. It will have a connector with pinouts like the one shown below. A ribbon cable will be used to make electrical connection to the connector in the back of the computer.

1	2	ACH0	ACH1
3	4	ACH2	ACH3
5	6	ACH4	ACH5
7	8	ACH6	ACH7
9	10	AISENSE/AIGND	DAC0OUT
11	12	AGND	DAC1OUT
13	14	DGND	PA0
15	16	PA1	PA2
17	18	PA3	PA4
19	20	PA5	PA6
21	22	PA7	PB0
23	24	PB1	PB2
25	26	PB3	PB4
27	28	PB5	PB6
29	30	PB7	PC0
31	32	PC1	PC2
33	34	PC3	PC4
35	36	PC5	PC6
37	38	PC7	EXTTRIG
39	40	EXTUPDATE	EXTCONV
41	42	OUTB1	GATB0
43	44	OUTB2	GATB1
45	46	CLKB1	OUTB2
47	48	GATB2	CLKB2
49	50	+5V	DGND

LEGEND:

- Analog inputs - ACHx
- Analog input ground - AISENSE/AIGND
- Analog outputs - DACxOUT
- Analog output ground - AGND
- Digital inputs and outputs - PAx, PBx, PCx
- Digital input/output ground - DGND
- Control handshaking - EXTTRIG, EXTUPDATE, EXTCONV
- Counter inputs/outputs - OUTBx, GATBx, CLKBx



pin 1

Looking at the connector
(on the back of the computer)

NOTE: LABVIEW MANUALS ARE AVAILABLE ON-LINE, AND CAN BE FOUND ON THE COURSE HOME PAGE - LEAVE THE PAPER MANUALS IN THE LAB.

22.2.3 Experiment 2 - Introduction to LabVIEW and the DAQ Cards

Objective:

Learn to use computers equipped for A/D and digital inputs.

Theory:

The computer reads data at discrete points in time (like a strobe light). We can read the data into the computer and then do calculations with it.

To read the data into a computer we write programs, and use "canned" software to help with the task. LabVIEW allows us to write programs for data collection, but instead of typing instructions we draw function blocks and connect them.

How we connect them determines how the data (numbers) flow. The functions are things like data reads and calculations.

In this lab we will be using Labview to connect to a data acquisition (DAQ) board in the computer. This will allow us to collect data from the world outside the computer, and make changes to the world outside with outputs.

When interfacing to the card using a program such as Labview, there must be ways to address or request information for a specific input or output (recall memory addresses in EGR226). The first important piece of information is the board number. There can be multiple DAQ boards installed in the computer. In our case there is only one, and it is designated device '1'. There are also many inputs and outputs available on the card. For analog outputs there are two channels so we need to specify which one when using the output with 0 or 1. For analog inputs there are 8 channels, and as before, we must specify which one we plan to read from using 0 to 7. For digital I/O there are a total of 24 pin distributed across 3 ports (1 byte each). Therefore when connecting inputs and output we must specify the port (PA=0, PB=1, PC=2) and the channel from 0 to 7. Note is that we can make the ports inputs or outputs, but not mixed - in other words we must pick whether a port will only be used for inputs or for outputs.

The voltage levels for the inputs and outputs are important, and you will need to be aware of these. For the digital outputs they will only ever be 0V or 5V. But the analog inputs and outputs will vary from -5V to 5V. This is build into the board. If we exceed these voltage limits by a few volts on the inputs, the boards have built in protection and should be undamaged. If we exceed the input voltages significantly, there is a potential to permanently damage the board.

Equipment:

PC with LabVIEW software and PCI-1200 DAQ card

Interface cable

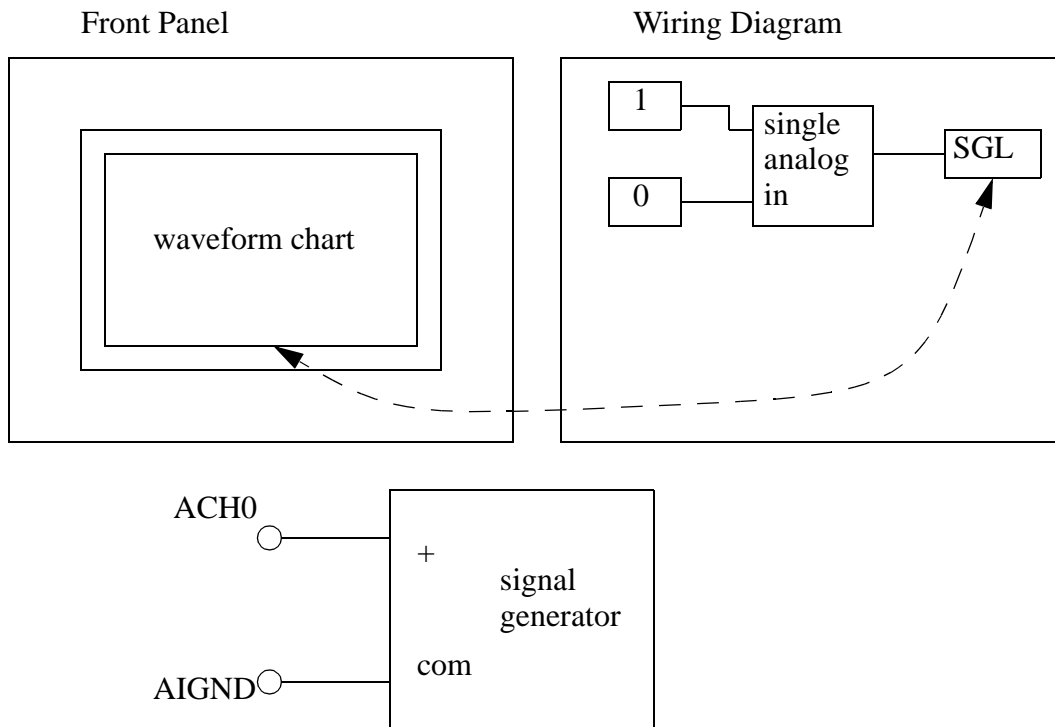
PLC trainer boards

Signal generator

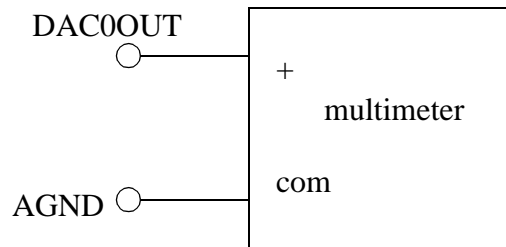
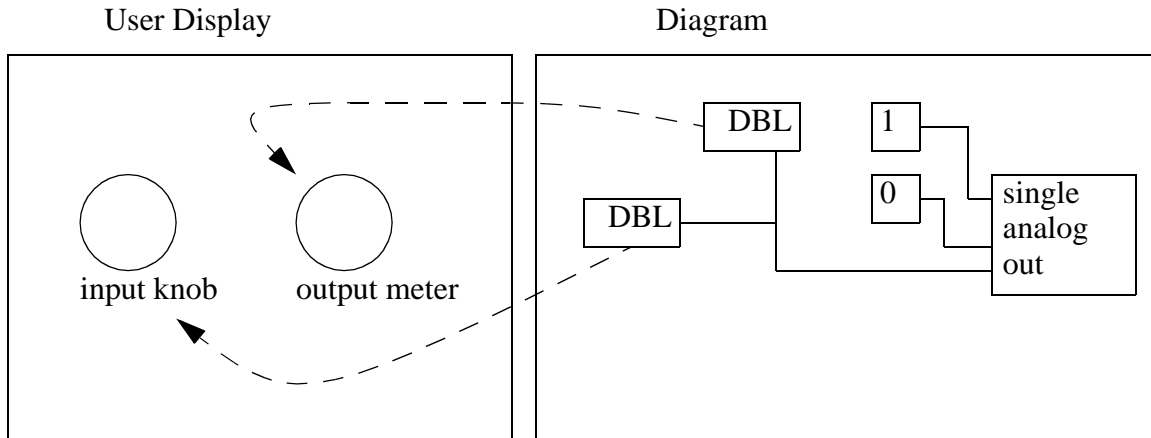
Digital multimeter

Procedure:

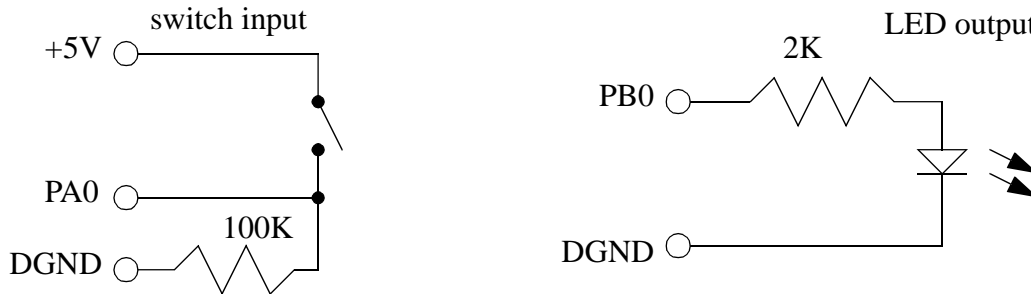
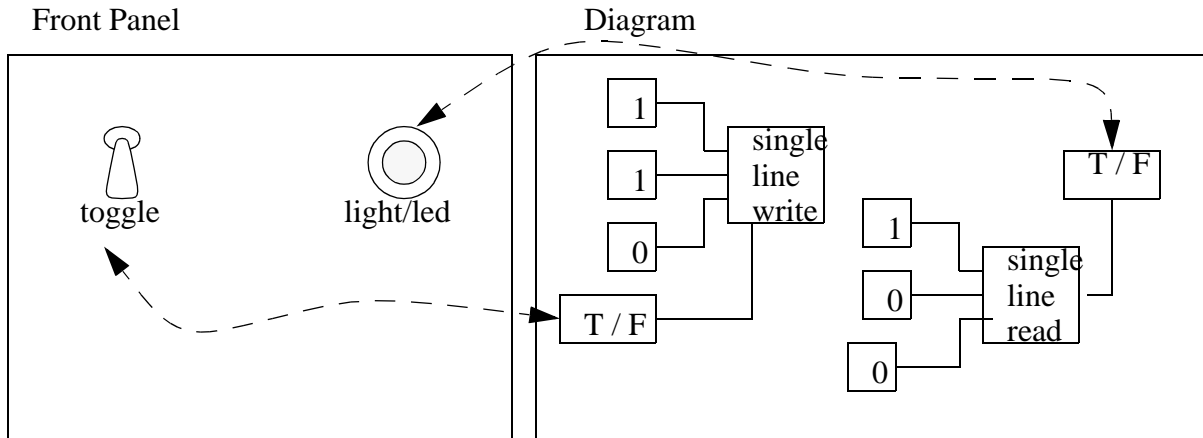
1. Go through the LabVIEW QuickStart Guide provided in the laboratory. This will also be good review for those who have used LabVIEW in previous courses.
2. Enter the LabVIEW program (layout) schematically shown below and connect a signal generator to the analog input (ACH0). (Note: there is a pin diagram for the connector in the Labview tutorial section.) Start the signal generator with a low frequency sinusoidal wave. Use the 'DAQ Configure' software to test the circuitry and verify that your hardware is operational. Then run your Labview program. Record the observations seen on the screen.



- 2b. Use a stop watch to determine the average number of samples per second. Run additional programs such as browsers, spreadsheets, etc. and see how this affects the data collection speed.
3. Connect the multimeter as shown below. Test the circuit using the 'DAQ Configure' utility. Enter the LabVIEW program schematically illustrated below and then run it. You should be able to control the output voltage from the screen using the mouse. Record your observations.



4. Connect the digital input and output circuits to the DAQ card and use the test panel to test the circuits. To do this, run the 'DAQ Configure' utility, double click on the 'PCI-1200', run the test panel window and ensure that the inputs and outputs are working correctly. Create the LabVIEW screen schematically illustrated below. This should allow you to scan an input switch and set an output light. When done, quit the program and run your LabVIEW program.



Marking:

1. A laboratory report should be written, including observations, and posted to the web.
2. The programs (VIs) that use the DAQ card should be posted to the web.

22.3 Lab 3 - Sensors and More Labview

22.3.1 Prelab 3 - Sensors

Theory:

Sensors allow us to convert physical phenomenon to measurable signals, normally voltage or current. These tend to fall into one of two categories, discrete or continuous. Discrete sensors will only switch on or off. Examples of these include, Inductive Proximity Sensors - use magnetic fields to detect presence of

metals

Capacitive Proximity Sensors - use capacitance to detect most objects

Optical Proximity Sensors - use light to detect presence

Contact Switches - require physical contact

Continuous sensors output values over a range. Examples of these are,

Potentiometers - provide a resistance proportional to an angle or displacement

Ultrasonic range sensors - provides a voltage output proportional to distance

Strain Gauges - their resistance changes as they are stretched

Accelerometers - output a voltage proportional to acceleration

Thermocouples - output small voltages proportional to temperature

In both cases these sensors will have ranges of operation, maximum/minimum resolutions and sensitivities.

Prelab:

1. Prepare a Mathcad sheet to relate sensor outputs to the physical phenomenon they are measuring.

22.3.2 Experiment 3 - Measurement of Sensor Properties

Objective:

To investigate popular industrial and laboratory sensors.

Procedure:

1. Sensors will be set up in the laboratory at multiple stations. You and your team should circulate to each station and collect results as needed. Instructions will be provided at each station to clarify the setup. The stations might include,
 - a mass on a spring will be made to oscillate. The mass will be observe by measuring position and acceleration.
 - a signal generator with an oscilloscope to read voltages phenomenon observe should include sampling rates and clipping.
2. Enter the data into Mathcad and develop a graph for each of the sensors relating input and output.

Submit:

1. A full laboratory report with graphs and mathematical functions for each sensor.

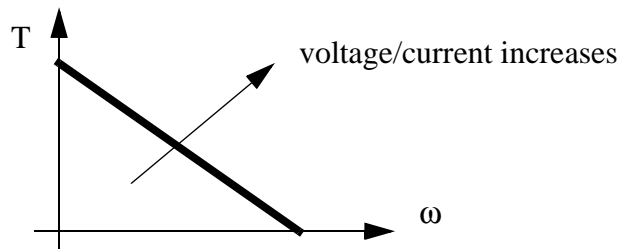
22.4 Lab 4 - Motors

- This set of labs will examine devices that have multiple phenomenon occurring.

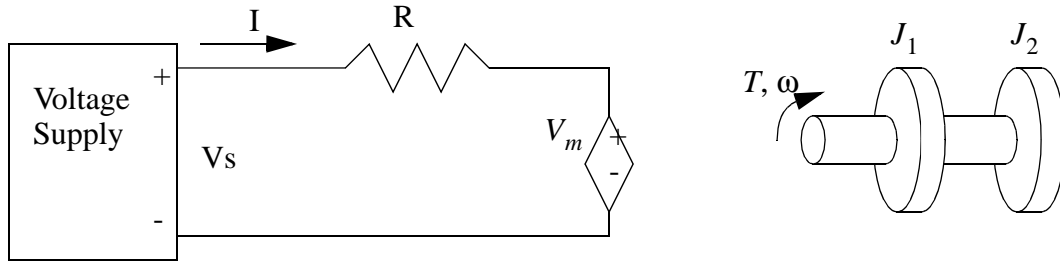
22.4.1 Prelab 4a - Permanent Magnet DC Motors

- Theory:

DC motors will apply a torque between the rotor and stator that is related to the applied voltage or current. When a voltage is applied the torque will cause the rotor to accelerate. For any voltage and load on the motor there will tend to be a final angular velocity due to friction and drag in the motor. And, for a given voltage the ratio between steady-state torque and speed will be a straight line.



The basic equivalent circuit model for the motor is shown below. We can develop equations for this model. This model must also include the rotational inertia of the rotor and any attached loads. On the left hand side is the resistance of the motor and the 'back emf' dependent voltage source. On the right hand side the inertia components are shown. The rotational inertia J_1 is the motor rotor, and the second inertia is an attached disk.



Because a motor is basically wires in a magnetic field, the electron flow (current) in the wire will push against the magnetic field. And the torque (force) generated will be proportional to the current.

$$T = KI \quad \therefore I = \frac{T}{K}$$

Next, consider the power in the motor,

$$P = V_m I = T\omega = KI\omega \quad \therefore V_m = K\omega$$

Consider the dynamics of the rotating masses by summing moments.

$$\sum M = T = J_M \left(\frac{d}{dt} \right) \omega \quad \therefore T = J_M \left(\frac{d}{dt} \right) \omega$$

The model can now be considered as a complete system.

The current-voltage relationship for the left hand side of the equation can be written and manipulated to relate voltage and angular velocity.

$$I = \frac{V_s - V_m}{R}$$

$$\therefore \frac{T}{K} = \frac{V_s - K\omega}{R}$$

XXXXXXAdd friction to the model

$$\therefore \frac{J_M \left(\frac{d}{dt} \right) \omega}{K} = \frac{V_s - K\omega}{R}$$

$$\boxed{\therefore \left(\frac{d}{dt} \right) \omega + \omega \left(\frac{K^2}{J_M R} \right) = V_s \left(\frac{K}{J_M R} \right)}$$

Looking at this relationship we see a basic first-order differential equation. We can measure motor properties using some basic measurements.

• Prelab:

1. Integrate the differential equation to find an explicit function of speed as a function of time.

2. Develop a Mathcad document that will accept values for time constant, supplied voltage and steady-state speed and calculate the coefficients in the differential equation for the motor.
3. In the same Mathcad sheet add a calculation that will accept the motor resistance and calculate values for K and J.
4. Get the data sheets for an LM675 from the web (www.national.com).

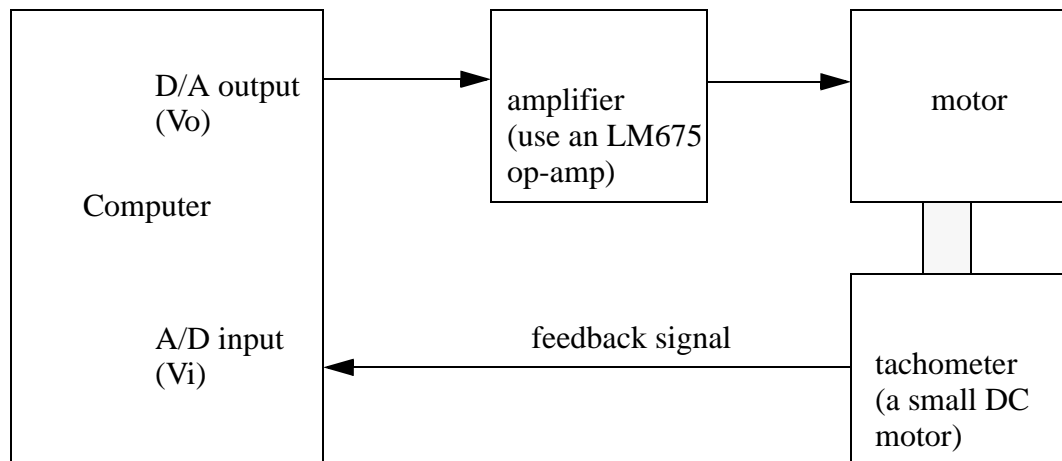
22.4.2 Experiment 4a - Modeling of a DC Motor

- Objective:

To investigate a permanent magnet DC motor with the intention of determining a descriptive equation.

- Procedure

1. With the motor disconnected from all other parts of the circuit, measure the resistance across the motor terminals.
2. Connect the motor amplifier, motor and computer as shown in the figure below.



Note: The motor being tested is the large motor. The small motor will be driven and act as a generator. In this case we will refer to it as a tachometer.

3. Write a Labview program that will output an analog voltage to drive the motor amplifier. An analog input will be used to measure the motor speed from the tachometer.

4. Use a strobe light to find the relationship between the tachometer voltage and the angular speed.
5. Obtain velocity curves for the motor with different voltage step functions.
6. Use a fish scale and a lever arm to determine the torque when the motor is stalled with an input voltage.

• Post-lab:

1. Determine the values of K for the motor. Determine the J for the rotor, and calculate J values for different load masses added.
2. Use the values of R, J and K to compare theoretical to the actual motor response curves found in procedure step #5.
3. Use the values of R, J and K to determine what the stalled torque should be in procedure step #6. Compare this to the actual.
4. Find the time constant of the unloaded motor.

• Submit:

1. All work and results.

22.5 Lab 5 - Basic Control Systems

22.5.1 Prelab 6a - Servomotor Proportional Control Systems

Theory:

DC servomotors typically have a first-order (velocity) response as found in the previous lab.

$$\left(\frac{d}{dt}\right)\omega + \omega\left(\frac{K^2}{JR}\right) = V_s\left(\frac{K}{JR}\right)$$

We can develop a simple control technique for control of the velocity using the equation below. For this form of control, we need to specify a desired velocity (or position) by setting a value 'Vd'. The difference between the desired speed and actual speed is calculated (Vd-Vi). This will give a voltage difference between the two values. This difference is multiplied by a constant 'K'. The value of 'K' will determine how the system responds.

$$V_o = P(V_d - V_i)$$

where,

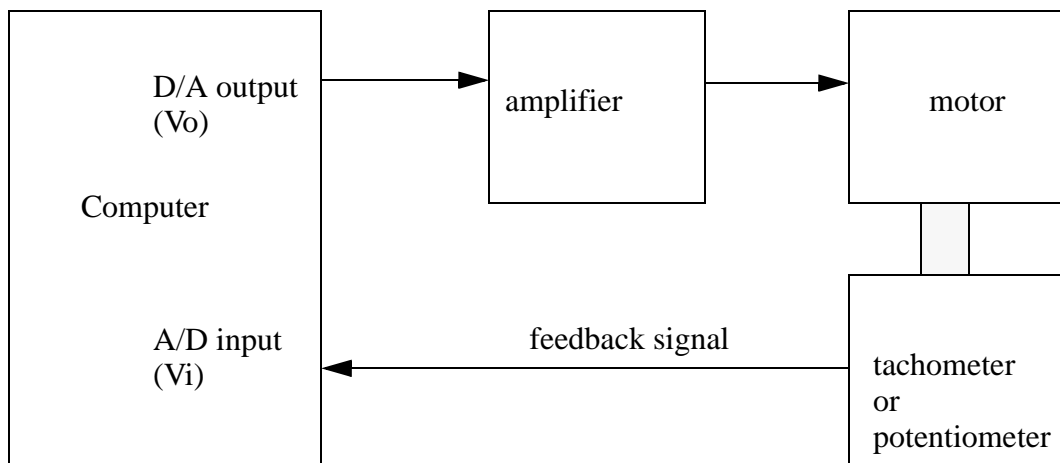
V_o = Voltage to motor amplifier to control speed

V_i = Voltage from tachometer to measure speed or position

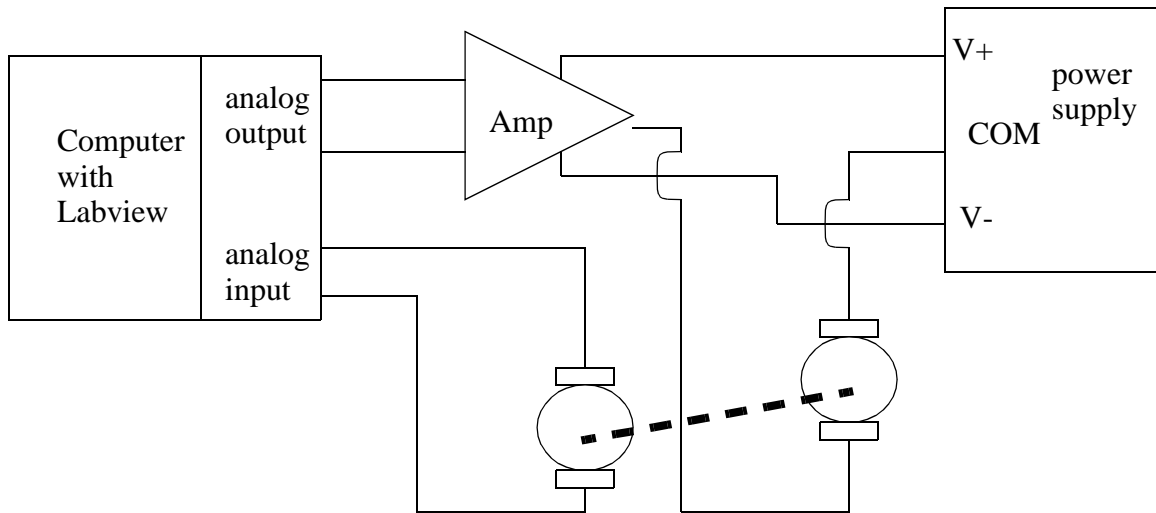
V_d = Desired motor speed voltage (user input)

P = Controller gain

The basic controller is set up as shown in the figure below. We can use a Labview program to implement the basic control equation described above.



The system below shows the components in the laboratory. The power supply may need to be constructed using two power supplies connected together. The analog output from the computer must be amplified (the computer can only output about 20mA maximum). The amplifier is constructed with an LM675 high current op-amp. The amplifier drives the large DC motor, which in turn drives a small DC motor being used as a tachometer. The voltage from the tachometer is input into the computer to determine the speed. A Labview program will subtract the tachometer voltage from the desired tachometer voltage, multiply the difference by a gain constant, and output the result to drive the motor.



Prelab:

1. Develop a Mathcad document that will model the velocity feedback controller given, motor parameters, desired velocity, an inertial load, and a gain constant. This is to be solved two different ways i) with Runge-Kutta integration, ii) integration of differential equations.
2. Test the controller model using a step function.

22.5.2 Experiment 5a - Servomotor Proportional Control Systems

Objective:

To investigate simple proportional servo motor control.

Procedure

1. Construct the equipment described in the
2. Apply a step function input and record the response.
3. For several values of proportional gain 'P', measure the response curves of the motor to a step function.

Post-lab:

1. Compare the theoretical and actual response curves on the same graphs.
2. Find and compare the time constants for experimental and theoretical results.

Submit:

1. All work and results.

22.6 Lab 6 - Basic System Components

22.6.1 Prelab 6a - Mechanical Components

- Theory:

Recall that for a rigid body we can sum forces. If the body is static (not moving), these forces and moments are equal to zero. If there is motion/acceleration, we use d'Alembert's equations for linear motion and rotation.

$$\sum F = Ma$$

$$\sum M = J\alpha$$

If we have a system that is comprised of a spring connected to a mass, it will oscillate. If the system also has a damper, it will tend to return to rest (static) as the damper dissipates energy. Recall that springs ideally follow Hooke's law. We can find the value of the spring constant by stretching the spring and measuring the forces at different points or we can apply forces and measure the displacements.

$$F = K_s \Delta x$$

$$\Delta x = x - x_0$$

$$F = K_s(x - x_0)$$

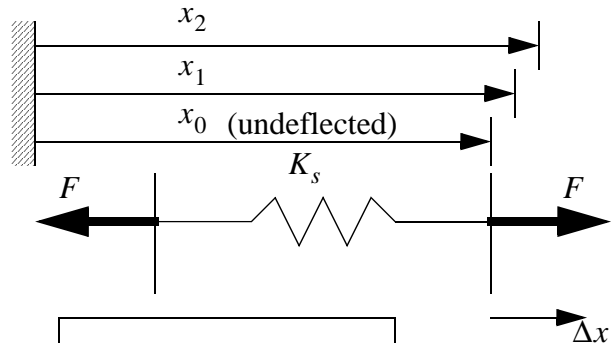
$$K_s = \frac{F_1}{x_1 - x_0} = \frac{F_2}{x_2 - x_0}$$

$$\frac{1}{K_s} = \frac{x_1 - x_0}{F_1} = \frac{x_2 - x_0}{F_2}$$

$$x_0 = x_1 - \frac{F_1}{K_s} = x_2 - \frac{F_2}{K_s}$$

$$\frac{1}{K_s}(F_2 - F_1) = x_2 - x_1$$

$$K_s = \frac{F_2 - F_1}{x_2 - x_1}$$



OR

$$F = K_s x$$

$$\Delta F = K_s \Delta x$$

$$K_s = \frac{\Delta F}{\Delta x}$$

$$K_s = \frac{F_2 - F_1}{x_2 - x_1}$$

In many cases we will get springs and devices that are preloaded. Both of the devices used in this lab have a preloaded spring. This means that when the spring has no force applied and appears to be undeflected, it is already under tension or compression, and we cannot use the unloaded length as the undeflected length. But, we can find the true undeflected length using the relationships from before.

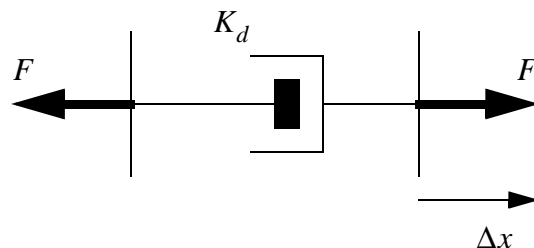
$$x_0 = x_1 - \frac{F_1}{K_s} = x_2 - \frac{F_2}{K_s}$$

Next, recall that the resistance force of a damper is proportional to velocity. Consider that when velocity is zero, the force is zero. As the speed increases, so does the force. We can measure this using the approximate derivatives as before.

$$F = K_d \frac{d}{dt} x$$

$$F = K_d \left(\frac{x(t + \Delta T) - x(t)}{\Delta T} \right)$$

$$K_d = \frac{F \Delta T}{x(t + \Delta T) - x(t)}$$



Now, consider the basic mass-spring combinations. If the applied forces are static,

the mass and spring will remain still, but if some unbalanced force is applied, they will oscillate.

$$+ \uparrow \sum F_y = -F_s - F_g = M \left(\frac{d}{dt} \right)^2 y$$

$$-K_s y - Mg = M \left(\frac{d}{dt} \right)^2 y$$

$$M \left(\frac{d}{dt} \right)^2 y + K_s y = -Mg$$

$$y_h(t) = C_1 \cos \left(\sqrt{\frac{K_s}{M}} t + C_2 \right)$$

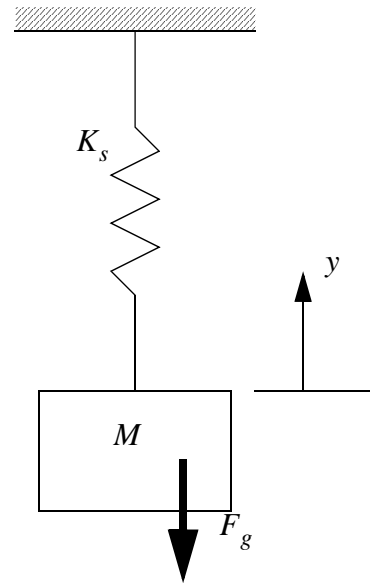
$$y_p(t) = A$$

$$\left(\frac{d}{dt} \right) y_p(t) = 0$$

$$\left(\left(\frac{d}{dt} \right)^2 y \right) y_p(t) = 0$$

$$M(0) + K_s(A) = -Mg \quad \therefore A = \frac{-Mg}{K_s}$$

$$y(t) = y_h(t) + y_p(t) = C_1 \cos \left(\sqrt{\frac{K_s}{M}} t + C_2 \right) + \frac{-Mg}{K_s}$$



Assume we start the mass at rest at the equilibrium height.

$$y(0) = 0 = C_1 \cos\left(\sqrt{\frac{K_s}{M}}(0) + C_2\right) + \frac{-Mg}{K_s}$$

$$\therefore C_1 = \frac{Mg}{K_s \cos(C_2)}$$

$$y'(0) = 0 = C_1 \sqrt{\frac{K_s}{M}} \sin\left(\sqrt{\frac{K_s}{M}}(0) + C_2\right)$$

$$\therefore C_2 = 0$$

$$y(t) = \left(\frac{Mg}{K_s}\right) \cos\left(\sqrt{\frac{K_s}{M}}t\right) + \frac{-Mg}{K_s}$$

The natural frequency is found by completing one time period,

$$2\pi = \sqrt{\frac{K_s}{M}}T = \sqrt{\frac{K_s}{M}}\frac{1}{f}$$

$$\therefore f = \frac{1}{2\pi} \sqrt{\frac{K_s}{M}}$$

In the lab an ultrasonic sensor will be used to measure the distances to the components as they move. The sensor used is an Allen Bradley 873C Ultrasonic Proximity Sensor. It emits sound pulses at 200KHz and waits for the echo from an object that is 30 to 100cm from it. It outputs an analog voltage that is proportional to distance. This sensor requires a 18-30 VDC supply to operate. The positive supply voltage is connected to the Brown wire, and the common is connected to the blue wire. The analog voltage output (for distance) is the black wire. The black wire and common can be connected to a computer with a DAQ card to read and record voltages. The sound from the sensor travels outwards in an 8 degree cone. A solid target will give the best reflection.

• Prelab:

1. Review the theory section.
2. Extend the theory by finding the response for a mass-spring, mass-damper, and mass-spring-damper system (assume values).
3. Set up a Mathcad sheet for the laboratory steps.

22.6.2 Experiment 6a - Mechanical Components

- Objective:

This lab will explore a simple translational system consisting of a spring mass and damper using instrumentation and Labview.

- Procedure

1. Use two masses to find the spring constant or stiffness of the spring. Use a measurement with a third mass to verify. If the spring is pretensioned determine the 'undeformed length'.
2. Hang a mass from the spring and determine the frequency of oscillation. Determine if the release height changes the frequency. Hint: count the cycles over a period of time.
3. Connect the computer to the ultrasonic sensor (an Allen Bradley Bulletin 873C Ultrasonic Range Sensor, see www.ab.com), and calibrate the voltages to the position of the target (DO NOT FORGET TO DO THIS). Write a Labview program to read the voltage values and save them to a data file. In the program set a time step for the voltage readings, or measure the relationship between the reading number and actual time for later calculation.
4. Attach a mass to the damper only and use Labview to collect position as a function of time as the mass drops. This can be used to find the damping coefficient.
5. Place the spring inside the damper and secure the damper. This will now be used as a combined spring damper. In this arrangement the spring will be precompressed. Make sure you know how much the spring has been compressed when the damper is in neutral position.
6. Use the spring-damper cylinder with an attached mass and measure the position of the mass as a function of time.
7. Use Working Model 2D to model the spring, damper and spring-damper responses.

- Post-lab:

1. Determine if the frequency of oscillation measured matches theory.
2. Compare the Labview data to the theoretical data for steps 2, 4 and 6.
3. Compare the Labview data to the working model simulations.

- Submit:

1. All results and calculations posted to a web page as a laboratory report.

22.7 Lab 7 - Oscillating Systems

- Many systems undergo periodic motion. For example, the pendulum of a clock.

22.7.1 Prelab 7a - Oscillating Systems

- Theory:

Suppose a large symmetric rotating mass has a rotational inertia J , and a twisting rod has a torsional spring coefficient K . Recall the basic torsional relationships.

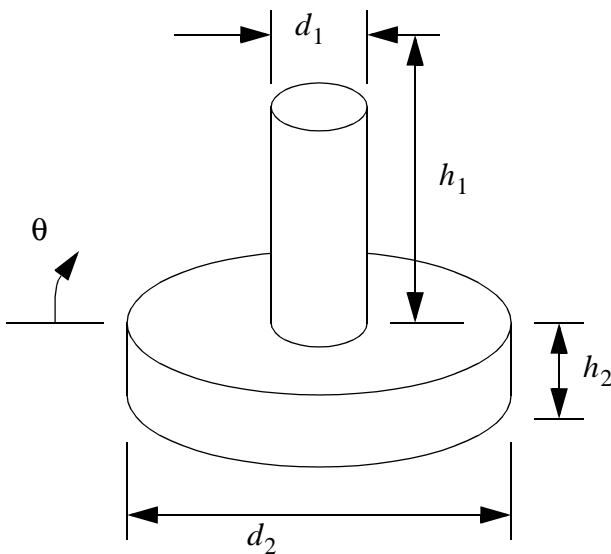
$$\sum T = -T = J\alpha = J\left(\frac{d}{dt}\right)^2 \theta$$

$$T = K\Delta\theta = K(\theta - \theta_0)$$

We can calculate the torsional spring coefficient using the basic mechanics of materials

$$T = \frac{JG\theta}{L}$$

Finally, consider the rotating mass on the end of a torsional rod.



$$\sum T = -\left(\frac{J_1 G}{L}\right)\theta = J\left(\frac{d}{dt}\right)^2 \theta$$

- Prelab:

1. Calculate the equation for the natural frequency for a rotating mass with a torsional spring.
2. Set up a Mathcad sheet that will
 - accept material properties and a diameter of a round shaft and determine the spring coefficient.
 - accept geometry for a rectangular mass and calculate the polar moment of inertia.
 - use the spring coefficient and polar moment of inertia to estimate the natural frequency.
 - use previous values to estimate the oscillations using Runge-Kutta.
 - plot the function derived using the homogeneous and particular solutions.

22.7.2 Experiment 7a - Oscillation of a Torsional Spring

- Objective:

To study torsional oscillation using Labview and computer data collection.

- Procedure

1. Calibrate the potentiometer so that the relationship between and output voltage and angle is known. Plot this on a graph and verify that it is linear before connecting it to the mass.
2. Set up the apparatus and connect the potentiometer to the mass. Apply a static torque and measure the deflected position.
3. Apply a torque to offset the mass, and release it so that it oscillates. Estimate the natural frequency by counting cycles over a long period of time.
4. Set up LabVIEW to measure the angular position of the large mass. The angular position of the mass will be measured with a potentiometer.

- Post-lab:

1. Compare the theoretical and experimental values.

- Submit:

1. All work and observations.
2. Post the laboratory report to a web page.

22.8 Lab 8 - Servo Control Systems

22.8.1 Prelab 5a - Research

- Theory:

This lab explores the use of advanced servo control systems. In particular Allen Bradley 5000 drives will be used. These drives are programmed using C. The drives can be interfaced to digital and analog IO, as well as handle serial communications.

- Prelab:

1. Visit the Allen Bradley web site (www.ab.com) and review the manuals for the Ultra5000 drives. In particular look at the installation manuals and programming manuals.
2. If necessary, review the principles of programming in C. Please note that the programming guide for the Ultra5000 drive contains a brief review of C pro-

gramming practices.

22.8.2 Experiment 5a - Tutorial and programming

- Objective:

To be able to develop programs for an AB Ultra5000 servo control system.

- Procedure

1. Follow the tutorial for the ULTRA5000 drives.
2. Develop a program that will use a voltage input (AIN1) from a potentiometer to position the drive. The drive will use forward and reverse limit switches on INPUT1 and INPUT2. When these switches are off the drive will be permitted to move in the given direction. INPUT3 should be used to stop the motor at any time, and stop the program. Ask the instructor to verify this program before continuing to the next stage.
3. Connect the servo drive to a linear slide. Attach two proximity sensors a few inches from each end of the slide to limit the stage motion. Use the 'Move' function of the drive to verify that the forward and reverse limit switches are on the proper ends.
4. Verify that your program can be used to control the position of the slide using a potentiometer.

- Submit:

1. A lab report including the fully commented C program for the drive.

22.9 TUTORIAL - ULTRA 5000 DRIVES AND MOTORS

1. (If not installed, install the Ultraware software, version 1.1.) Connect the ULTRA 5000 drive to the serial port, set the drive to '01' on the front of the drive, and then apply power.
2. Run the 'Motor Configuration' program and open the 'Motors' file. Select the correct motor type and then select 'New Rotary....'. Notice the motor parameters. Close the program.
3. Run the 'Ultraware' software. Select 'Create new file....' and give it a unique name such as 'lab.udb' (Note: you should put this file on a floppy disk, or zip disk so that you may use it again later.). It will scan the available drives to

- locate the drive. When done, a '5KDrive' should appear under 'On-line drives'.
4. Right click on the '%KDrive' and select properties. For the 'Drive Type' select the correct version indicated on the Ultra 5000 drive. Use 'Setup' to update the drive parameters, then close the window.
 5. Under the '5KDrive' select 'Motor' and then select the motor type. Then select 'Setup' to update the motor parameters. Notice the values and units. Close the window.
 6. Turn the motor shaft by hand, it should turn easily. Enable the motor with 'Commands' 'Enabled' on the tool bar. The motor should now be controlled, and the shaft hard to turn.
 7. For the motor select 'Motion' then 'Jog'. In the window set the values below. After this hold the shaft and hit 'Jog forward' then 'Jog reverse' then 'Stop'. Notice how the values at the bottom of the screen change.
 - Program Velocity 100,000
 - Program Acceleration 100,000
 - Program Deceleration 100,000
 8. Now select 'Motion' then 'Move' and set the values below. When done, click 'Start' to cause the motion. Change the value in 'Profile Distance' and click 'Start' again.
 - Profile Distance 40,000
 - Velocity 20,000
 - Acceleration 20,000
 - Deceleration 20,000
 9. Create a project with 'Insert' then 'Project'. Click on the 'Project' that was created on the tree, then 'Insert Source File'. Type in the following file, and then click 'Build'. If there are any errors fix them before continuing to the next step.

```

#include <motion.h>

int main(void){
    InitMotionLibrary(); // Start the motor control functions

    AxisEnable();        // Enable control of the motor

    MoveSetAcc(2000); // Set the maximum accelerations and velocity
    MoveSetVel(1000);
    MoveSetDec(2000);

    MoveDistance(10000); // Move in the positive direction
    while(MoveInProgress()){ }; // Wait until the move is complete
    Sleep(1000);         // Pause for 1 second
    MoveDistance(-10000); // Move in the negative direction
    while(MoveInProgress()){ };
    Sleep(1000);

    AxisDisable();      // Release the motor drive control

    return 1;
}

```

10. Drag the new file 'Project.exe' to the 'Programs' under the '5KDrive'. Hold the shaft of the motor and right click on the new file just dropped and click 'Run'. The motor shaft should move forward, pause for 1 second, reverse, then stop.
11. Enter the following program and observe the results. It should behave the same as the previous program, but it has more structure. When developing control applications it is important to structure the programs so that they are easier to write and debug.

```
#include <motion.h>

void setup(){
  InitMotionLibrary();
  AxisEnable();
  MoveSetAcc(2000);
  MoveSetVel(1000);
  MoveSetDec(2000);
}

void shutdown(){
  AxisDisable();
}

void jog(int distance){
  MoveDistance(distance);
  while(MoveInProgress()){};
}

int main(void){
  setup();

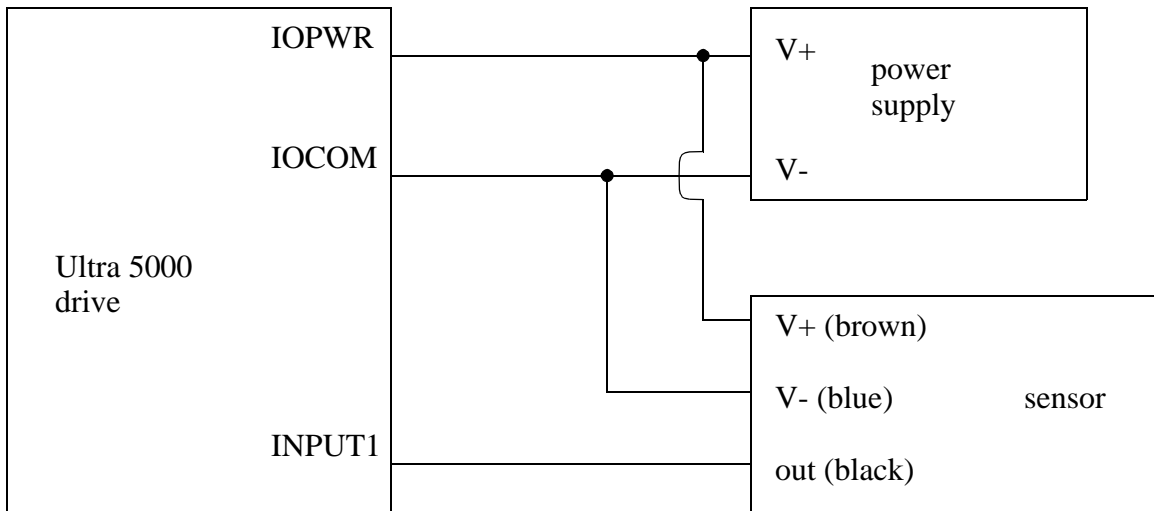
  jog(10000);
  Sleep(1000);
  jog(-10000);
  Sleep(1000);

  shutdown();
  return 1;
}
```

12. Digital inputs and outputs for the drive are located on the upper connector, CN1A. The pin out for this connector is given below. Wires can be connected by pushing a thin slot screwdriver into the slot to release the pressure clamp, and then inserting the wire. The nominal voltages to activate an input are between 12V and 24V - do not exceed the maximum voltage as this may damage the drive.

Connector	Pin	Function
CN1A	1	INPUT 1 - Digital input 1
	2	INPUT 2 - Digital input 2
	3	INPUT 3 - Digital input 3
	4	INPUT 4 - Digital input 4
	5	INPUT 5 - Digital input 5
	6	INPUT 6 - Digital input 6
	7	INPUT 7 - Digital input 7
	8	INPUT 8 - Digital input 8
	9	OUTPUT 1 - Digital output 1
	10	OUTPUT 2 - Digital output 2
	11	OUTPUT 3 - Digital output 3
	12	OUTPUT 4 - Digital output 4
	13	SHIELD - for shielded cable termination
	14	IOPWR - V+ from external supply power for the IO
	15	INPUT 9 - Digital input 9
	16	INPUT 10 - Digital input 10
	17	INPUT 11 - Digital input 11
	18	INPUT 12 - Digital input 12
	19	INPUT 13 - Digital input 13
	20	INPUT 14 - Digital input 14
	21	INPUT 16 - Digital input 15
	22	INPUT 16 - Digital input 16
	23	OUTPUT 5 - Digital output 5
	24	OUTPUT 6 - Digital output 6
	25	OUTPUT 7 - Digital output 7
	26	OUTPUT 8+ - Relay output 8
	27	OUTPUT 8- - Relay output 8
	28	IOCOM - common from external IO power supply

13. Connect an external power supply to the digital IO and connect a proximity sensor to input 1, as shown in the diagram below. Verify that the sensor is connected properly using the Ultraware software. Also connect a multimeter to OUTPUT1 to read the output voltage.



14. Enter the program below and determine what the program does.

```

#include <motion.h>

void setup(){
  InitMotionLibrary();
  AxisEnable();
  MoveSetAcc(2000);
  MoveSetVel(1000);
  MoveSetDec(2000);
}

void shutdown(){
  AxisDisable();
}

void jog(int distance){
  MoveDistance(distance);
  while(MoveInProgress()){};
}

int main(void){
  int    count = 0,
        state_last = OFF,
        state_now;
  setup();

  while(count < 10){
    state_now = InputGetState(1); // get the state of INPUT1
    if( (state_now == ON) && (state_last == OFF) ){
      jog(1000);
      count++;
    }
    state_last = state_now;
    OutputSetState(1, state_now); // set OUTPUT1 to the INPUT1 value
  }

  shutdown();
  return 1;
}

```

15. Analog outputs (and connections for additional encoders) are available on connector CN1B. Connect a variable power supply to act as an analog input across AIN1 and +5VCOM (WARNING: TURN THE POWER SUPPLY OFF AND SET THE VOLTAGE AT THE MINIMUM BEFORE TURNING THE SUPPLY BACK ON). Connect a multimeter across AOUT1 and +5VCOM to read

an analog output voltage. Verify the connection using the Ultraware software.

Connector	Pin	Function
CN1B	1	+5V - internal power supply
	2	AX+ Encoder input/output A+
	3	AX- Encoder input/output A-
	4	BX+ Encoder input/output B+
	5	BX- Encoder input/output B-
	6	IX+ Encoder input/output I+
	7	IX- Encoder input/output I-
	8	+5VCOM internal power supply common
	9	AIN1 - analog input 1
	10	AIN2 - analog input 2
	11	+5VCOM internal power supply common
	12	AOUT1 - analog output 1
	13	AOUT2 - analog output 2
	14	SHIELD - termination point for shielded cables

16. Enter the program below and determine what it does.

```
#include <motion.h>

void setup(){
  InitMotionLibrary();
  AxisEnable();
  MoveSetAcc(2000);
  MoveSetVel(1000);
  MoveSetDec(2000);
}

void shutdown(){
  AxisDisable();
}

void locate(int position){
  MovePosition(position);
  while(MoveInProgress()){};
}

int main(void){
  float a_in;
  int new_location;
  setup();

  while(1 == 1){ // loop forever
    a_in = AnalogInputGetVoltage(1); // Get the voltage in from AIN1
    new_location = a_in * 1000;
    locate(new_location);
    AnalogOutputSetVoltage(1, a_in/2.0); // output half the input voltage
  }

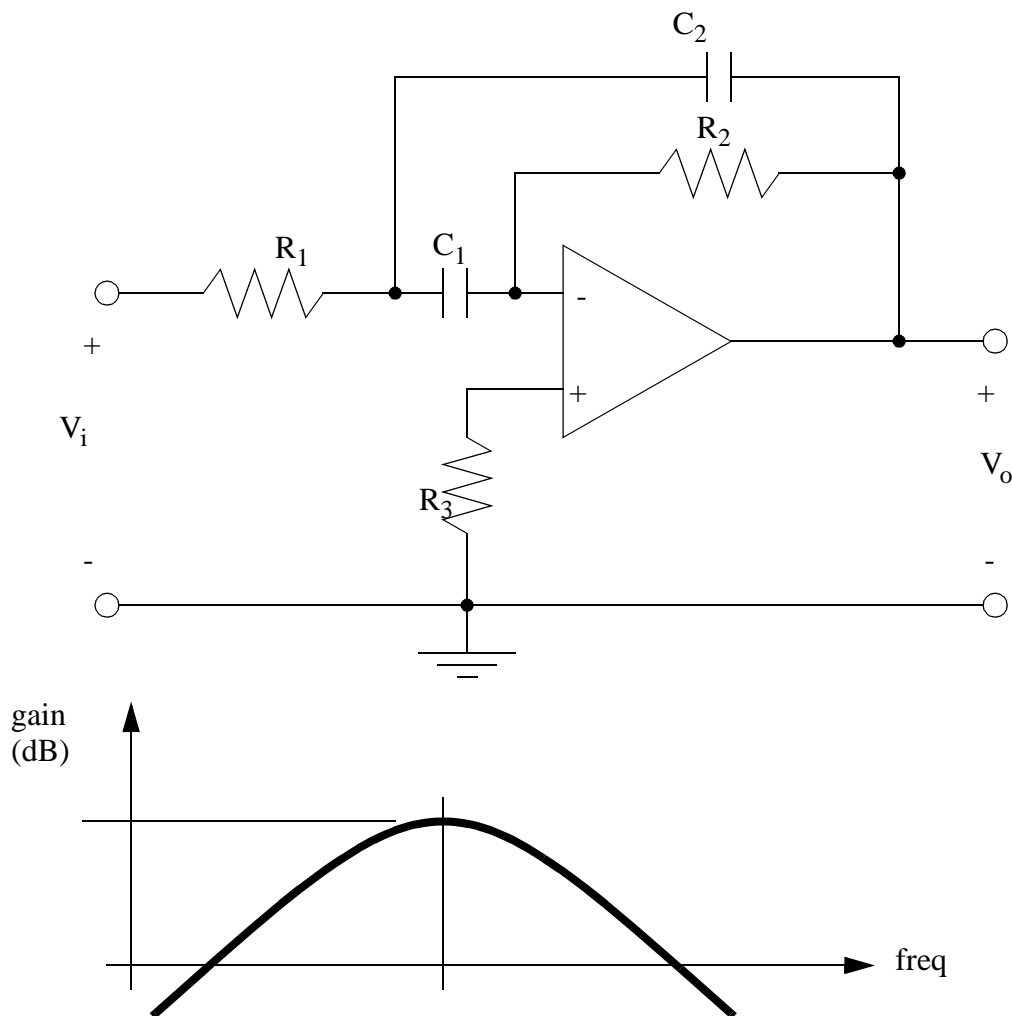
  shutdown();
  return 1;
}
```

22.10 Lab 9 - Filters

22.10.1 Prelab 9 - Filtering of Audio Signals

Theory:

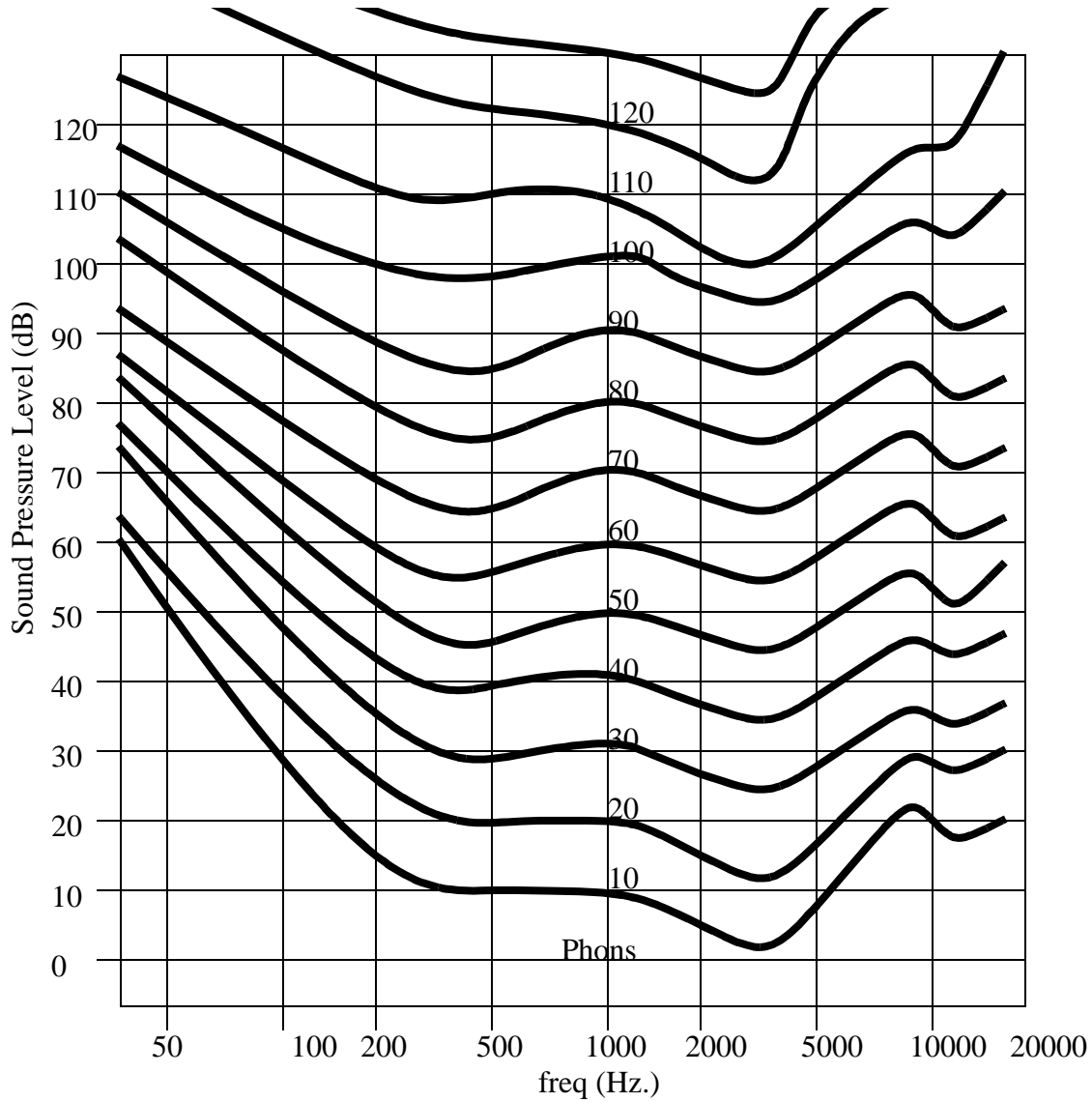
We can build simple filters using op-amps, and off the shelf components such as resistors and capacitors. The figure below shows a band pass filter. This filter will pass frequencies near a central frequency determined by the resistor and capacitor values. By changing the values we can change the overall gain of the amplifier, or the tuned frequency.



The equations for this filter can be derived with the node voltage method, and the final transfer function is shown below.

$$\frac{V_o}{V_i} = \left(-\frac{1}{R_1} \right) \left(\frac{DC_1R_1R_2}{1 + DC_2R_1 + DC_1R_1 + D^2C_1C_2R_1R_2} \right)$$

As dictated by the ear, audio signals have frequencies that are between 10Hz and 16KHz as illustrated in the graph below. This graph shows perceived sound level, with the units of 'phons'. For example, we can follow the 100 phons curve (this would be like a loud concert or very noisy factory requiring ear protection). At much lower and higher frequencies there would have to be more sound pressure for us to perceive the same loudness, or phon value. If the sound were at 50Hz and 113 dB it would sound as loud as 100dB sound at 1KHz.



You may appreciate that these curves are similar to transfer functions, but they are non-linear. For this lab it is important to know how the ear works because you will be using your ear as one of the experimental devices today.

Prelab:

1. Derive the transfer function given in the theory section.
2. Draw the Bode plot for the circuit given $R1=R2=1000\text{ohms}$ and $C1=C2=0.1\mu\text{F}$.
You are best advised to use Mathcad to do this.

22.10.2 Experiment 9 - Filtering of Audio Signals

Objective:

To build and test a filter for an audio system.

Procedure:

1. Set up the circuit shown in the theory section. Connect a small speaker to the output of the amplifier.
2. Apply a sinusoidal input from a variable frequency source. Use an oscilloscope to compare the amplitudes and the phase difference. Also record the relative volumes you notice.
3. Supply an audio signal, from a radio, CD player, etc. Record your observations about the sound.

Submit:

1. Bode plots for both theory and actual gain and phase angle.
2. A discussion of the sound levels you perceived.

22.11 Lab 10 - Stepper Motors

22.11.1 Prelab 10 - Stepper Motors

Theory:

Stepper motors move to positions, steps, but don't rotate continuously. To achieve continuous rotation the motor is moved, or stepped, continuously in one direction. A full rotation of the motor is often divided into 200 steps (of 1.8 degrees each), although other resolutions are common. The motors we will use in the lab have 400 steps per revolution. When rotating the motor it can be stepped in the clockwise (CW) or counterclockwise (CCW) direction.

The equipment to be used in the lab is comprised of a stepper motor, a drive unit, a motion controller (indexer), and a computer running terminal software for programming. The motor is connected to the drive unit which will power the coils and step in the positive or negative directions, as indicated by pulses from the motion controller. The motion controller will accept motion commands, or run programs which generate the overall motion. The motion controller commands the motor drive to move by sending it pulses. The motor controller varies the period of the pulses. When motion is starting or ending a longer period corresponds to a slower angular velocity. At the midpoint of motion the pulse period is the shortest, at the maximum velocity.

The motion is normally described with a velocity profile. This can be given by defining the maximum acceleration/deceleration and maximum velocity. An alternative way to define these limits is to provide an acceleration and deceleration time, with a total motion time.

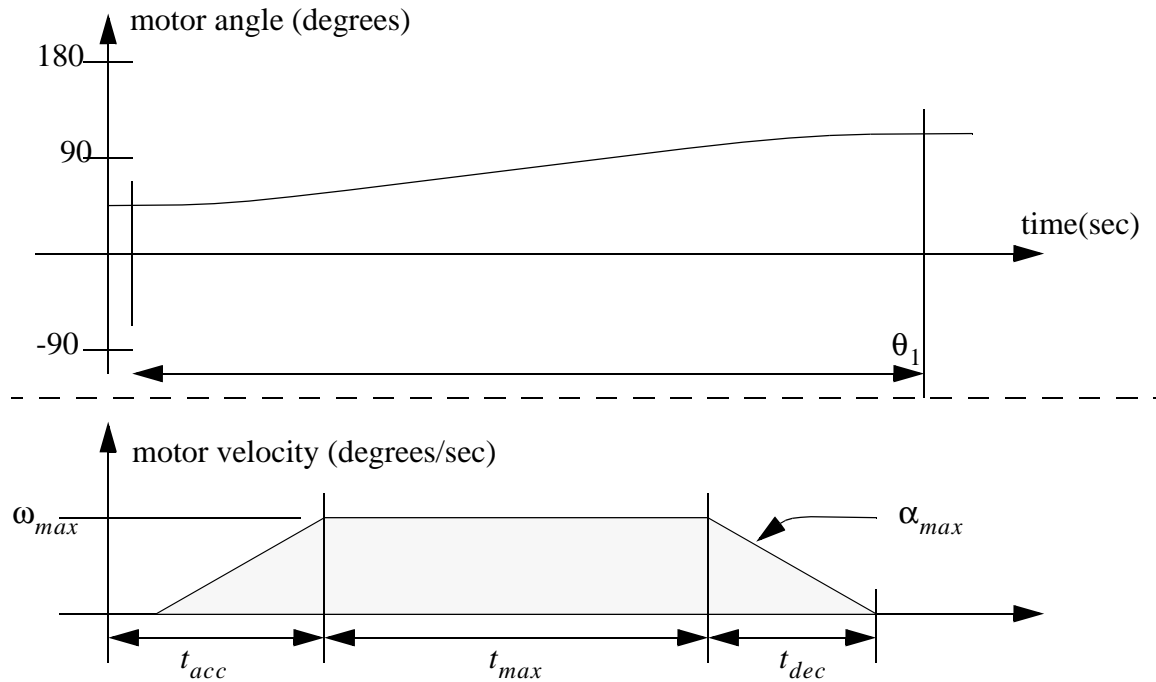


Figure 427 Velocity curves for motor control

Prelab:

1. In the textbook review the sections on stepper motors and motion control. Make sure you are familiar with the basic operation of a stepper motor and the structure of a velocity profile.
2. Review the manuals for the stepper motor on the course home page on claymore.

22.11.2 Experiment 9 - Stepper Motors

Objective:

To develop a stepper motor control program.

Procedure:

1. Follow the attached tutorial
2. Develop a program to accept an input to the motion controller and then move the

motor to different positions.

Submit:

1. The questions answered during the tutorial.
2. The program developed (with comments).

22.12 TUTORIAL - STEPPER MOTOR CONTROLLERS (by A. Blauch and H. Jack)

1. Examine the manuals for the stepper motor and controller.
2. Obtain and examine the stepper motor equipment. There are three main components: stepper motor, drive unit, and indexer. Identify each unit. Verify the components are connected together properly based on the diagram shown below. Note that the drive unit obtains power from an AC source (outlet) while the indexer receives power from a DC source (power supply). Once all of the connections have been verified, apply power to the system. If set up properly, the power LEDs on the drive unit and indexer will turn on and the stepper motor will have holding torque (the shaft won't turn freely).

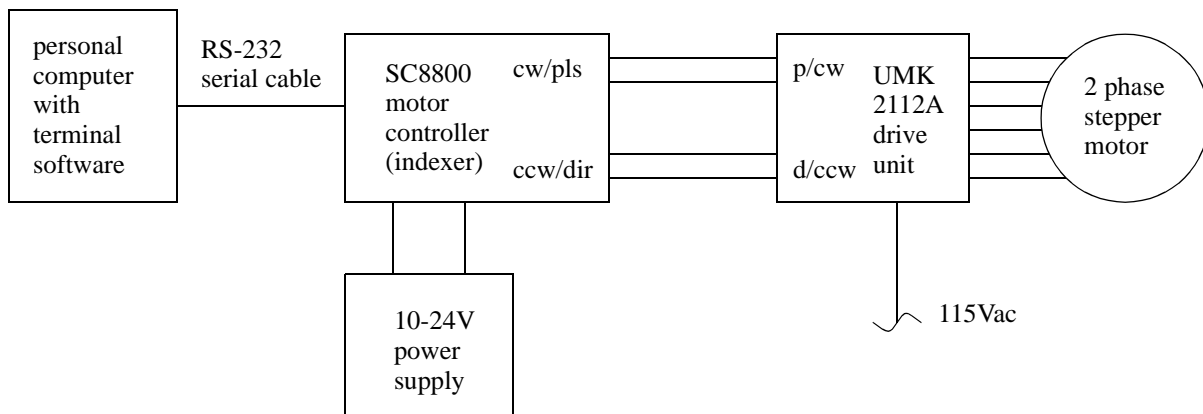


Figure 428 The stepper motor control system components

3. The indexer communicates with the PC via text transfers across a serial port. Connect a serial cable from the indexer to the COM port on the back of the PC. On the PC, run HyperTerminal (located under **Programs->Accessories->Communications**) or any other terminal emulation program. Set it to go "Direct to Com 1". Check to make sure the communication parameters are set to 9600 baud, 8 data bits, 1 stop bit, no parity, and no flow control/handshaking. Once connected the prompt **0>** will appear on the screen. If nothing appears, press enter several times. Cycling power on the indexer will cause a reset ban-

ner to appear. To display all of the available indexer commands, type **HELP**.

4. The indexer commands of interest are the ones used to generate and execute different velocity profiles. The list below highlights the commands used in this lab. Typing **HELP2**, **HELP3**, **HELP4**, **etc...** will display a more detailed lists of these and other profiling commands

Command	Description
H<+/->	Sets the direction of motion
T<xx.xx>	Sets the acceleration/deceleration time in seconds
VS<xxxxxx>	Sets the starting velocity in pulses per second
V<xxxxxx>	Sets the run velocity in pulses per second
D<xxxxxx>	Sets the distance in pulses for the MI command
MC	Starts continuous motion
MI	Starts incremental motion
R	Displays the status of system parameters
RESET	Resets the system
S	Controlled stop of current motion
STOP	Immediate stop of current motion

5. To see what the current velocity profile parameters are, type the parameter name followed immediately by a question mark (i.e. **T?** will display the current acceleration/deceleration time).
6. Type in the following series of commands. Observe how the stepper motor responds and how the response relates to the parameter values entered.

```
VS0
V2000
T0.5
D10000
MI
```

7. Change the acceleration and deceleration times to be longer and shorter and observe the effects using the values below. As part of your observations, hold the shaft while the motion is in progress.

acc/dec	vel	distance
0.01	20000	50000
0.5	10	100
0.01	2000	6000

8. Generate and execute a velocity profile that will cause the motor to rotate at one revolution/second using the continuous motion command. Based on the run velocity parameter value, determine what angle one pulse (step) corresponds to. Compare your number to the resolution value on the stepper motor label.
9. Set the acceleration/deceleration parameter to several seconds. Using the run velocity parameter and the continuous motion command, make the stepper motor rotate at various velocities. Determine the maximum speed of rotation for the stepper motor (i.e. at what point does the motor lose synchronization with the input pulses).
10. Decrease the run velocity to about one half the maximum speed determined previously. Using the acceleration/deceleration parameter and the continuous motion command, make the stepper motor accelerate at various rates. Determine the maximum acceleration for the stepper motor (i.e. at what point does the motor lose synchronization with the input pulses). Make a note of any strange phenomena that may occur.
11. Determine the parameter values for a velocity profile with an acceleration of 18000 rpm^2 , maximum velocity of 600 rpm, and a total run time of 10 seconds. Generate, execute, and verify the profile. For this profile you will need to use the incremental motion command.
12. Based on your experiences in lab with motors, discuss some advantages and disadvantages of using a stepper motor compared to a DC motor. Some issues to think about are torque, position control, velocity control, and interface.
13. Enter a program with the following commands. Notice that the editor has commands to alter 'Ax' line 'x', delete 'Dx' line 'x' or insert 'Ix' a line 'x'. Each program will be assigned a number between 0 and 49. The program can be run with 'RUN TEST'.

```
EDIT TEST  
PC0  
TA0.5  
TD0.5  
VS2000  
D6000  
MI  
Q
```

NOTE: After this point you should use the manual for a source of examples.

14. You can see a list of programs in the memory using 'LIST'. Programs can be deleted with 'DELETE'.

15. The motion controller has a limited set of variables available for user programs. Unlike normal programming variables, there are a fixed set available, as indicated below. The integers values are all 4 bytes and can range in value between +/-2,147,483,647. Write a simple program that sets a general variable value.

W, X, Y, Z - general purpose integer

V- velocity

VS -

LOOP -

D - distance

IN_x - input bit 'x'

PC - position counter, the current motor position in steps

16. Outputs can be set using the 'OUT_x' command, where 'x' is the output bit number. Connect a voltage supply to the inputs and a multimeter to the outputs and create a program to read an input and set an output as shown below. Refer to the manual for pinouts and electrical specifications.

17. Logical operations can be performed using 'IF' and 'WHILE' statements. The operations can include =, !=, <, <=, >, >=. The if statements can include 'ELSE' conditions and should terminate with 'ENDIF' statements. The while loops must end with 'ENDW' statements. The equivalent to a for-loop is the 'LOOP x' statement. It will loop the code 'x' times to the 'ENDL' statement. Write a simple program that will loop until an input becomes true.

18. It is possible to branch within a program using 'JMP'. To jump to another program (subroutine) the 'JMP_SEQ' command can be used, the program can then return using the 'RET' command. Write a program that has a subroutine.

22.13 Lab 11 - Variable Frequency Drives

22.13.1 Prelab 11 - Variable Frequency Drives

Theory: AC induction motors are designed with motor winding on the stator (outside) of the motor. The AC current in the stator coils sets up an alternating magnetic field. This field induces currents in the conductors (squirrel cage) in the rotor. This current creates a magnetic field that opposes the field from the stator. As a result a torque is created. In actuality the rotor must rotate somewhat slower than the field changes in the stator, this difference is called slip. For example a 3 phase motor (with two poles) that has a 60Hz power applied will with absolutely no rotational resistance rotate at 60 times per second. But in use it might rotate at 58 or 59Hz. As the number of poles in the motor rises, the speed of rotation decreases. For example a motor with four poles would rotate at half the speed of a two pole motor. The speed of the motor can be controlled by changing the frequency of the AC power supplied to the motor. The motor that we will use in the lab is a 3 phase AC motor made by Marathon Electric. It is a Black Max model number 8VF56H17T2001. The motor drives are Allen Bradley model 160 and 161 motor drives.

Prelab:

1. Visit the Marathon Electric and Allen Bradley web sites and review the manuals for the motor and controllers. Don't print these, but make a note of the web address so that you can find the manuals easily during the lab.

22.13.2 Experiment 11 - Variable Frequency Drives

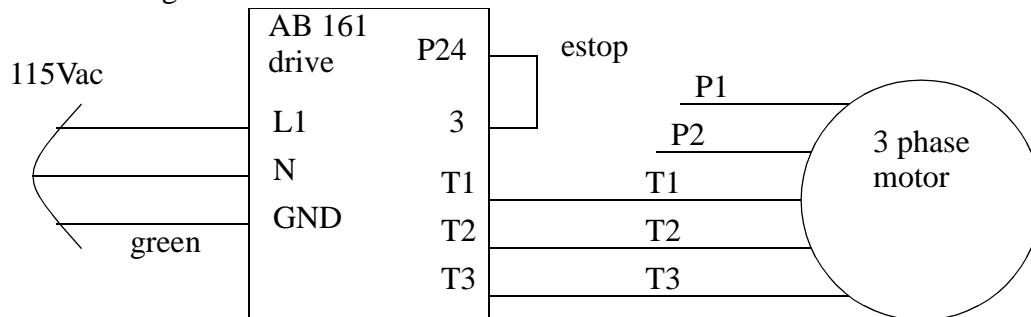
Objective: To learn the basic operation of Variable Frequency Drives (VFD) and motors.

Procedure:

1. Follow the attached tutorial. (and make notes and observations.)
2. Connect the 3 phase motor to a tachometer for a velocity feedback. Using the manual determine how to change the PID parameters. Run a number of tests to see what it does.
3. Determine how to use an analog input to control the speed. Set it up and find the relationship between the input voltage and the motor speed.
4. Replace the 161 controller with the 160 controller, and determine how this controller operates.

22.14 TUTORIAL - ALLEN BRADLEY 161 VARIABLE FREQUENCY DRIVES

- The motor should be connected to the drive as shown in the figure below. The three phase power lines are T1, T2 and T3. These can be connected to the drive by loosening the screw on the front face of the drive. (WARNING: make sure the power is off before opening the drive, and reseat the drive when done) The two other lines P1 and P2 from the drive are for a thermal overload relay, we will not use these. For general caution the ends of the wires should be covered with electrical tape to prevent accidental contact with other conductors. The wire from P24 (a 24V power source) and input 3 are for an emergency stop and must be connected for the drive to work. These terminals can be found under a flip up panel on the bottom front of the drive that can be opened by pulling on the right side of the face.

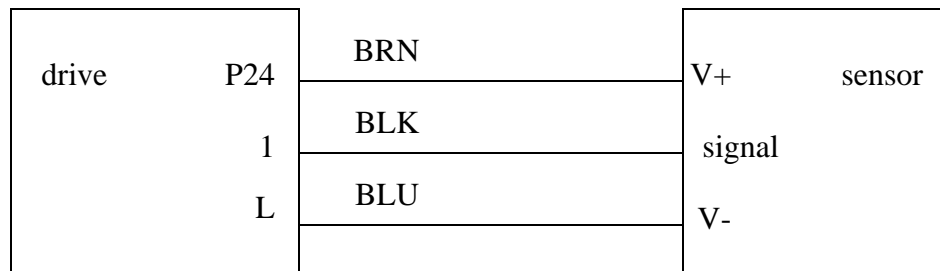


- Notice the display and lights on the front face of the drive. When a program is running the 'RUN' light is on. The program can be started using the green '1' button, and stopped with the red '0' button.
- Program the unit using the buttons on the front panel, following the steps below (from page 17 in the manual). After the steps have been followed press the run button and turn the potentiometer to vary the speed. Try holding the shaft (cover the shaft to avoid cuts) at high and low speeds. What do you notice at very low speeds?

button(s)	result	description
sel	d01	move the the parameter number, from frequency
up or down	A--	move to the 'A' parameters
sel	A01	select the 'A' parameter group (use up or down if not A01)
up or down	00	select the velocity input source
sel	01	display the current setting
up or down	00	select the potentiometer (on the controller) as the input
enter	A01	accept the new value
up or down	A02	move the the start button selector
sel	01	move to parameter
up or down	02	select the value for start to the 'run' button
enter	A02	accept the value

- Connect a proximity to the controller as shown in the diagram below. Display

the input values using 'D05'. The bits on the screen should move up for an active input, and down for an inactive input. Input 3 will always be active because it is being used for the emergency stop (a factory default).



5. Various display parameters are listed below. Try these to see what information they show.

- D02 - display the motor current (try holding the shaft while turning slowly)
- D01 - display the frequency of rotation
- D03 - the direction of rotation
- D04 - PID parameters (when in use)
- D05 - input status
- D06 - output status
- D16 - total drive run time in 10 hour blocks

6. Function parameters can be set with the 'F' locations. Change the acceleration and deceleration times to 1 second using the 'F02' and 'F03' locations.

7. Restore the controller to the factory defaults using the sequence below from page 16 of the manual.

button(s)	result	description
up or down	b--	move to the 'B' parameter groups
sel	b01	enter the
up or down	b84	move to the reset function
sel	01	select the function, make sure the value is '01'
sel	b84	select the value
sel+up+down+0	b84	hold down the keys for 3 seconds
sel+up+down	00 then 0.0	release the '0' key and continue to hold the others until the display blinks.

22.15 TUTORIAL - ULTRA 100 DRIVES AND MOTORS

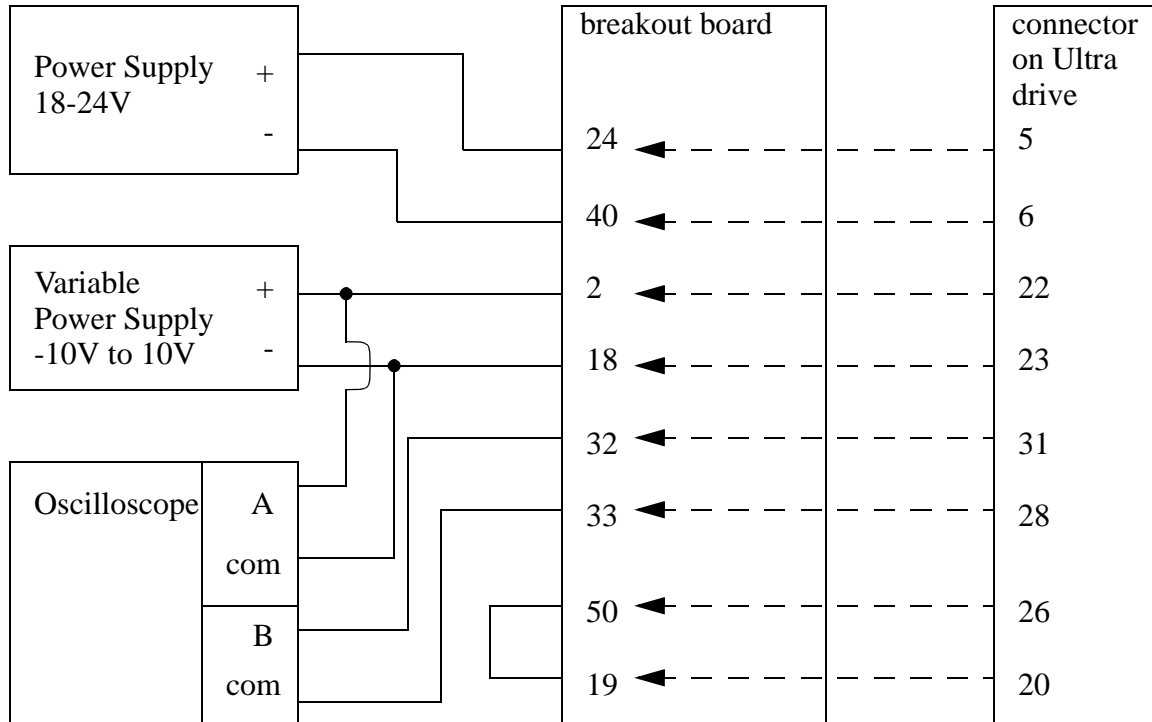
1. Obtain the motor and controller sets. This should include a brushless servo motor (Y-1003-2H), an Ultra 100 Controller and miscellaneous cables. The cables are described below. Locate and verify they are connected as indicated in the sequence given below.
 - a) The motor is connected to the controller with two cables, one for the power to drive the motor, and the other a feedback from the encoder. The encoder is used to measure the position and velocity of the motor. If not connected already, connect these to the motor controller.
 - b) The controller should also have a serial cable for connecting to a PC serial port. Connect this to a PC.
 - c) A breakout board is provided with that will connect to the motor controller through a ribbon cable to J1. The pin mapping from the motor controller to the terminal numbers on the motor controller are given below.

controller pin	breakout pin	controller pin	breakout pin
1	9	26	50
2	25	27	17
3	41	28	33
4	8	29	49
5	24	30	16
6	40	31	32
7	7	32	48
8	23	33	15
9	39	34	31
10	6	35	47
11	22	36	14
12	38	37	30
13	5	38	46
14	21	39	13
15	37	40	29
16	4	41	45
17	20	42	12
18	36	43	28
19	3	44	44
20	19	45	11
21	35	46	27
22	2	47	43
23	18	48	10
24	34	49	26
25	1	50	42

- d) A power cord to be connected to a normal 120Vac power source. DO

NOT CONNECT THIS YET.

2. The controller needs to be connected as shown below. In this case it will require that the breakout board be connected to the motor controller by a ribbon cable. Note: the pins on the breakout board don't have the same numbers as on the motor controller. Connect the wires as shown below but do not turn on the devices yet.



- Plug in the power cord for the Ultra 100 drive and look for a light on the front to indicate that it is working. **WARNING:** The motor might begin to turn after the next step --> Turn on the power supplies and the oscilloscope.
- Run the 'Ultramaster' software on the PC. When prompted select 'drive 0'. At this point the software should find the drive and automatically load the controller parameters from the controller. When it is done a setup window will appear.
- In the "drive setup" window ensure the following settings are made. When done close the setup window.
 - Motor Model: Y-1003-2-H - this is needed so that the proper electrical and mechanical properties of the motor will be used in the control of the motor (including rotor inertia).
 - Operation Mode Override: Analog Velocity Input - This will allow the voltage input (C+ and C-) to control motor velocity.
- Select the 'IO Configuration' window and make the following settings.
 - Analog Output 1: Velocity - Motor Feedback - this will make the analog voltage output from the controller (R+ and R-) indicate the velocity of the motor, as measured by the encoder.
 - Scale: 500 RPM/VOLT - this will set the output voltage so that every

500RPM will produce an increase of 1V.

7. Open the "Oscilloscope" on the computer screen and make the following settings. When done position the oscilloscope to the side of the window so that it can be seen later.
 - Channel A: Velocity - command
 - Channel B: Velocity - motor feedback
 - Time Base: 12.5ms
8. Open the control panel and change the speed with the slider, or by typing it in. Observe the corresponding changes on the oscilloscope on the screen, and the actual oscilloscope. Notice that the velocity change follows a ramp. The slope of the ramp is a function of the maximum acceleration.
9. Hold the motor shaft while it is turning slowly and notice the response on the oscilloscopes.
10. Open the tuning window and change the 'I' value to 0. Change the motor speed again and notice the final error. It should be larger than in previous trials.
11. Leave the 'I' value at '0' and change the 'P' value to 50 and then change the speed again. This time the error should be larger, and the response to the change should be slower. Hold the shaft again, it should be easier to stop.
12. Try other motor parameters and see how the motor behaves. Note that very small or large values of the parameters may lead to controller faults - if this occurs set more reasonable 'P' or 'I' values and then reset the fault.
13. Notice that the left hand side of the tuning window has an autotune function. Select this to automatically pick the controller values. To perform this function click the 'stop' then 'start' buttons. The motor will move, and then reverse direction to determine the dynamic load. It will then pick parameters to leave the motor controller critically damped.
14. Set the parameters back to 'P' = 200 and 'I' = 66.
15. Stop the 'Control Panel', this will allow the motor to be controlled by the external voltage supply. Change the voltage supply and notice how the motor responds. Reverse the voltage supply and notice that the motor turns in the other direction - remember to return it to normal when done.
16. Connect proximity sensors to digital inputs 1 and 2. The pins on the controllers are 32 and 33 respectively. The pins on the breakout board are 48 and 15 respectively. The wires for the proximity sensors are standardized as brown for V+, blue for the common and black for the signal.
17. Once the proximity sensors are attached test them with the "Display Digital IO" window. After this change the "IO Configuration window to set input 1 as "forward enable" and input 2 as the reverse enable". Test these to see that they actually stop the motor in both directions when not on.
18. Attach the motor to a ball screw slide and attach the proximity sensors. Change the motor controller to follow a position command. You should be able to control the position of the motor by changing the variable power supply voltage.

22.16 TUTORIAL - DVT CAMERAS (UPDATE FROM 450???)

(*****To be done outside the lab without the hardware)

1. If the DVT software is not installed yet, install it now.
2. In either case put the DVT CD Rom in the computer and follow the tutorials on the CD.
3. Run the 'Framework' software. When it starts to run a window called 'PC Communications' will appear. In this case we will only use the simulator, so select 'Emulator' then 'Model 630', then 'Connect'. The emulator will also start as another program.
4. At this point we can set up a simple set of vision tests for the camera. The camera is capable of supporting tests for more than one product. You can select which product the tests will be for by first selecting 'Products' then 'Product Management...'. This will put up a window to define the product details. Select 'New' and then enter 'part1'. Use 'OK' to dismiss the windows and get back to the main programming window.
5. We need to now set up the emulator to feed images to the programming software. We can do this by loading a set of test images. To do this open the 'DVT SmartImage Sensor Emulator' window. Then select 'Configure' then 'Image sequence...'. Select 'Browse for Image' and then look for images on the CD in the 'emulator\images\Measurement' directory. Pick the file 'Measurement001.bmp'. The next window will ask for the last image so select 'Measurement010.bmp'. Select 'OK' to get back to the main emulator window.
6. Go back to the Framework programming screen. The emulator can be started by selecting the right pointing black triangle near the red circle. You should see images appear in a sequence. Pause the images using the single vertical line.
7. Next we will add a test to the product. On the left hand side of the screen select 'Measurement', select the small line with two red arrows and then move the mouse over the picture and draw a line over a single section of the part, for example the right hand most side by the large circle.
8. In the pop-up window enter the test name 'test1'. Apply the changes and dismiss the window and run the camera again. Notice that as the images flip by the result of the test is shown at the bottom of the screen. As the images are captured the part may move around. It is common to delete and recreate the sensor a number of times to find the right position for the line. Double click on the sensor and look at some of the options possible. Try changing these and see what happens.
9. Delete the previous sensor and create a sensor (rotational positioning: find edge in parallelogram), and a translation sensor (translational positioning: fiducial rectangle). For the translation sensor, under the 'Threshold' select 'Dark Fiducial'. Run the software and note the rotation and translation data at the bottom of the screen as the images are processed.
10. Delete the sensors. Go to the emulator and load the two '1dBarCode' images in

the 'emulator\images\1DReader'. In Framework run the camera so that these appear on the screen, then pause the camera. On the left hand side of the screen select 'Readers' then 'Barcode Sensor: Line for 1D codes'. Draw the sensor line over the middle of the barcode.