Distributed Laboratory Modules for System Dynamics and Controls Courses

A Plan A Master’s Thesis
Submitted by: David Waletzko
Advised by: Dr. William Durfee

In partial fulfillment of the requirements for the degree of Master of Science in Mechanical Engineering

At the
University of Minnesota, Institute of Technology
December 2005
UNIVERSITY OF MINNESOTA

This is to certify that I have examined this copy of a master’s thesis by

David Donald Waletzko

and have found that it is complete and satisfactory in all respects, and that any and all revisions required by the final examining committee have been made.

_______________________________________
Name of Faculty Adviser

_______________________________________
Signature of Faculty Advisor

_______________________________________
Date

Graduate School: Department of Mechanical Engineering
Acknowledgements

First, I would like to thank God for blessing me with so many talents as well giving me this life.

Dr. William Durfee, my advisor, invested large quantities of time, effort, and instruction into this project. He was always accessible to ask a question, and never short on ideas for new things the project could do. I wouldn’t be the engineer I am today without his guidance and aid.

Then there are the other members of my committee, Dr. Lee and Dr. Terveen, for their feedback and review of this document.

My lovely wife Diana, whose love and support was a great boost when things seemed hopeless.

Finally there is my family, who always encouraged me to excel academically and even more so in life.
Abstract

The distributed laboratory system is for engineering students in introductory system dynamics and controls courses that need to gain intuitive feel for physical systems. It is a way to explore basic mechanical engineering concepts through a hands-on experience that uses inexpensive, computer-controlled hardware kits. Unlike traditional laboratory experiences, the distributed lab module is brought home by students and tackled on a self-paced schedule in much the same manner as a homework assignment, thus allowing the student to customize the laboratory experience to their learning style. Over the two years of its development, the laboratory modules changed from an open ended design problem to a final module combing both controls and frequency response lessons. This module gives the students all the hands-on experiences that traditional labs on these topics would provide, while at the same time providing all the benefits of a self-paced problem set. Improvements in accuracy, reliability, manufacturability, ruggedness, and easy of use were made continuously during the design process, resulting in the creation of a powerful tool for the instruction of students in system dynamics and controls courses and beyond.
Table of Contents

1. Introduction
   1.1. Purpose
   1.2 Other Teaching Methods
   1.3 Types of Laboratories
   1.4 Distributed Laboratories’ Design Requirements

2. Generation 0
   2.1. Overview
   2.2. Module Selection Process

3. Generation 1
   3.1. Overview
   3.2. ¼ Car Module
   3.3. Audio Filtering Module
   3.4. Evaluation
   3.5. Lessons Learned

4. Generation 2
   4.1. Overview
   4.2. MSD/PCM Module
   4.3. Audio Filtering Module
   4.4. Component Characterization Tests

5. Evaluation

6. Recommendations and Conclusions

Appendix A: Photographs

Appendix B: Technical Documentation
List of Tables and Figures

Table 1.1: The design requirements
Figure 2.1: A) Team generated concepts and modules. B) Professor generated concepts and modules
Figure 2.8a: The speaker (audio filtering) module
Figure 2.8b: The position control module
Figure 2.8c: The mass spring damper module
Figure 2.8d: The PIC Control box. This box was the link between the modules and the computer, and housed the PIC chip, motor controller, and other electronics.
Figure 3.1: The block diagram of a quarter car that the module is based on.
Figure 3.2a: Drawing of Mass Spring Damper Module
Figure 3.2b: Close up of linkage
Figure 3.3: Photograph of the Mass Spring Damper module
Figure 3.4: Mass Spring Damper Visual Basic user interface
Figure 3.5: Picture of Audio Filtering Module
Figure 3.6: Drawing of Audio Filtering Module
Figure 3.7: Audio Filtering Module Visual Basic user interface.
Figure 3.8: Student test results
Figure 3.9: Student confidence data
Figure 3.10: The means and standard deviation for the test data.
Figure 3.11: The means and standard deviation for the student confidence data.
Table 3.12: The results of the student survey. The average is computed by assigning a value to each answer, i.e. 1 for strongly disagree, 2 for disagree, etc, and then averaging those numbers.
Table 3.13: Student Comments
Figure 4.1: Diagram of the Frequency Response/ PID Module
Figure 4.2: The Frequency Response/ PID Module
Figure 4.3: A string is connected to the inertia wheel, wrapped around the disk several times, with a bag hanging down the other end of the string. The pennies are then placed in the bag, one at a time, and the rotation of the inertia is recorded via the user interface.
Figure 4.4: C Code (Always running)
Figure 4.5: C Code functions
Figure 4.6: VB Code: DLABForm (Runs at startup)
Figure 4.7: VB PID code
Figure 4.8: VB PID Code Continued
Figure 4.9: VB FR Code
Figure 4.10: VB FR Code
Figure 4.11: The system model, assuming that the rubber band is slightly stretched to begin with, i.e L>Lund.
Figure 4.12: The system equations.
Figure 4.13: The torque vs. theta relationship for the spring linearity test shows a linear relationship.

Figure 4.14: The data generated during the spring constant determination test. As can be seen in the graph, the spring constant of this particular rubber band was 0.0141 N/Deg.

Figure 4.15: Calculation of Inertia

Table 5.1: The design metrics for the project, along with their final status.

Table 5.2: Durability/ruggedness/accuracy tests.

Figure 5.3: Before and after data for test 1. Black dots were taken before the hour long test, the red circles were taken after.

Figure 5.4: Before and after data for engineering test 2. The dots were data taken before the test, the “X”s were taken after.

Figure 5.5: The module and all its required parts inside the storage box.

Figure 5.6: Before and after data for engineering test 5. The dots were data taken before the test, the “X”s were taken after.

Figure 5.7: The concrete staircase where test 6 was performed

Figure 5.8: Initial Proportional Controller

Figure 5.9: Medium Gain Proportional controller

Figure 5.10: Large Gain Proportional Controller

Figure 5.11: Initial Proportional + Derivative Controller
(Same proportional gain as figure 5.9)

Figure 5.12: Medium Gain Proportional + Derivative Controller

Figure 5.13: Large Kd Proportional Derivative Controller

Figure 5.14: Initial Proportional Integral Derivative Controller

Figure 5.15: Critically Damped Proportional Integral Derivative Controller

Figure 5.16: A large Ki value Proportional Integral Derivative Controller

Figure 5.17: The data collected for test 9

Figure A.1: The final FR/PID Module

Figure A.2: The final FR/PID Module (Front View)

Figure A.3: The final FR/PID Module (Top View)

Figure A.4: The final FR/PID Module (Back View, with rubber band (PID Mode))

Figure A.5: The final FR/PID Module (Back View, without rubber band (PID Mode))

Figure A.6: The final FR/PID Module (Top View, with rubber band (PID Mode))

Figure A.7: The Mass Spring Damper Module

Figure A.8: The Mass Spring Damper Module Mass Production

Figure A.9: The Audio Filtering Module

Figure A.10: The Audio Filtering Module Mass Production
1. Introduction

1.1 Purpose

Distributed Laboratory Modules are take-home laboratory kits that enhance student learning by allowing them to experience first hand the concepts taught in system dynamics and controls courses. Unlike traditional laboratories, students take the module home with them, and, much like a homework assignment, complete it at their own pace; there is no need to devote a university classroom for several hours a day to allow students to complete the labs. The modules are designed to be low cost (<$60), durable enough to survive abuse in a student backpack, and accurate to give the results that students see in their textbooks. While the laboratories require the use of a windows PC, all students have easy access to these machines in a computer lab, if not at home.

1.2 Other Teaching Methods

Laboratories certainly can’t provide everything for every student, and universities usually strive to cater to as many learning styles as possible. Lectures, where a professor or teaching assistant stands in front of a class and instructs them on a certain topic, remains a popular teaching method, and is an excellent way of covering new material both quickly and effectively. As an added benefit students have the option of asking a question of the lecturer. The disadvantage to a lecture is that different students assimilate information at different rates, and lectures can only meet so often during the school year. Also, while some lectures will have a demonstration, providing hands on material for students in the middle of a lecture is simply not feasible.
Reading and homework assignments are common for many courses and provide a student with the opportunity to go at their own pace (within reason) through the material. This allows students to experience the material on a much more personal level, and focus on the areas they need more help in. It is common for a student’s grade to be based on their homework performance. Most engineering courses encourage students to work together on homework assignments, though asking a question of a professor or teaching assistant can be difficult at times. While homework assignments can be valuable learning tools, they do not provide a hands-on experience with real equipment.

1.3 Types of Laboratories

Traditional laboratories have long served an important function in university settings. They allow student to experience first hand what they hear in lecture. Most laboratories focus on a series of steps that a student, either alone or in a group, goes through, to complete the lab. A teaching assistant is present to guide students through the lab, answer questions as they come up, and make sure the experiment is done properly. For most of the day, the room where the lab is done remains empty, and students cannot work on the experiments. If a student needs more time, they must set up a special work session with the teaching assistant, or if that is not possible, then the student may not get a chance to complete the work at all, lowering not only their grade, but losing out on the lessons they could learn.

Simulated labs are much like the traditional laboratory described above. Instead of performing the experiment on actual equipment, the tests, and possibly even the data,
are simulated on a computer. This allows students to experience experiments that may be
to time consuming or expensive to actually perform.

There are also web-based labs. These labs are performed by the student at home,
or in a computer lab, where they connect to the laboratory computer over the internet.
These labs may either be simulated entirely on the labs computer, with the student just
remotely connecting, or they may allow for remote control of physical lab equipment,
with cameras and other sensors giving the student feedback\footnote{1}.

The distributed lab system is different from all of these. Here the lab kits are
stored at the university until they are ready to be distributed out to the students, one lab
kit per student. The student then performs the lab at home or in a computer lab, similar to
the web-based method described above, but they also have the hardware connected
directly to the computer they are at. They can see feel, hear, and touch the equipment
they are interacting with. Students determine the pace they will perform the lab with, and
how much they want to deviate from the instructions given to them. There is no need for
the university to maintain a lab full of equipment, or to keep a server running. Teaching
assistants or other staff may still need to answer questions, but they do not hover over the
students as they perform the lab. Finally, students can perform the lab at any time that
works well for them.

1.4 Distributed Laboratories’ Design Requirements

The design requirements are shown in Table 1.1
<table>
<thead>
<tr>
<th>Metric</th>
<th>Reason</th>
<th>Goal</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accurate</td>
<td>To compare to theory results</td>
<td>Results from calculations and simulations match</td>
</tr>
<tr>
<td>Demonstrates Concept</td>
<td>Module designed for learning</td>
<td>95% of students understand concept after performing the lab</td>
</tr>
<tr>
<td>Easy to Manufacture</td>
<td>Manufacturing is done in house</td>
<td>100 modules made in under two weeks by undergraduate students</td>
</tr>
<tr>
<td>Final Cost</td>
<td>Low cost is always desirable</td>
<td>&gt;$100 to make in lots of 100</td>
</tr>
<tr>
<td>Lab Experiment Documentation</td>
<td>Aids student and instructors performing the labs</td>
<td>Exists</td>
</tr>
<tr>
<td>Light</td>
<td>Ease of transport</td>
<td>&lt;5lbs</td>
</tr>
<tr>
<td>Low weight</td>
<td>Ease of transport and storage</td>
<td>&lt;1900 in³</td>
</tr>
<tr>
<td>Mass Production Cost</td>
<td>Resource efficient, no large loss if students loose them</td>
<td>&lt;$100 parts cost</td>
</tr>
<tr>
<td>Multiple Concepts Taught in 1 Module</td>
<td>Adjusts to different faculty, and allows students more areas to explore. Lowers cost and number of required modules.</td>
<td>&gt;3 topics can be taught by a module</td>
</tr>
<tr>
<td>PC Compatibility</td>
<td>Needs to work in a variety of environments</td>
<td>Communicates with Windows 98 or newer PCs and laptops, over USB</td>
</tr>
<tr>
<td>Power</td>
<td>Needs to get power in many locations (dorms, homes, computer labs, etc)</td>
<td>Uses standard wall outlet</td>
</tr>
<tr>
<td>Rugged</td>
<td>Survive trips in backpack</td>
<td>*See Sect. 6 “Evaluation”</td>
</tr>
<tr>
<td>Safe</td>
<td>Keeps students safe</td>
<td>Low voltages/currents, no sharp edges, no injuries</td>
</tr>
<tr>
<td>Short Setup Time</td>
<td>Assembling the labs is not the focus</td>
<td>&lt;10 minutes to set up lab</td>
</tr>
<tr>
<td>Simple</td>
<td>Operation easy to understand</td>
<td>95% of students report no trouble performing the lab</td>
</tr>
<tr>
<td>Simple Student Interface</td>
<td>Easy to use and change</td>
<td>Visual Basic interface</td>
</tr>
<tr>
<td>Standardized Parts</td>
<td>To easily obtain replacements</td>
<td>100% of materials from suppliers, not surplus</td>
</tr>
<tr>
<td>User Friendliness</td>
<td>Encourages student use</td>
<td>=&gt;95% of students found labs enhance their learning experience</td>
</tr>
</tbody>
</table>

**Table 1.1:** The design requirements
2. Generation 0

2.1. Overview

The first phase, or Generation 0, for the distributed labs, was a Senior Design project for the ME 4054 course at the University of Minnesota from January to May, 2003. The objective was “to design and construct lab module kits that students enrolled in the ME3281 System Dynamics and Controls course can use at home or in the university computer labs to learn course concepts. The student would then be able to perform the lab on ‘their’ time, not worrying about lab periods being too short or inconvenient, and leaving the course with some physical interaction with systems and concepts they learn in the classroom”\(^2\). A team of five mechanical engineering seniors, Jason Halpern, Jason Havemeier, Jon Knutson, Jason McVay, and David Waletzko, along with Prof. Durfee as advisor, worked on, and completed three modules which were demonstrated at the Design Show. These modules were: a Mass, Spring, Damper (MSD) module for showing how system models and equations related to real world systems; an Audio Filtering Module (AFM) to demonstrate both visually and audibly the effects of high and low pass filters on audible sine waves; and a Position Control Module (PCM) to allow students to try their hand at applying the PID control gains they calculated on a real system.

2.2. Module Selection Process

A large part of the Generation 0 work went into selecting which modules to construct. To begin with, the design team generated a list of topics that the modules could teach, as well as what sort of modules could teach them. The course textbook\(^3\), the
original NSF proposal, and the description of the project provided by the course were used to generate these lists. Also, a survey of the professors who teach ME 3281 was conducted to find what modules they would like to see, and the topics they cover when they teach the course. These lists are tabulated below in Figure 2.1.

A) Team Generated Lists

<table>
<thead>
<tr>
<th>Preliminary Concepts</th>
<th>Preliminary Modules</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Translational Systems</td>
<td>• Inverted Pendulum</td>
</tr>
<tr>
<td>• Mechanical Systems</td>
<td>• Spring and Damper</td>
</tr>
<tr>
<td>• Electrical Systems</td>
<td>• Cantilever Beam</td>
</tr>
<tr>
<td>• Thermal Systems</td>
<td>• Speaker</td>
</tr>
<tr>
<td>• Fluid Systems</td>
<td>• Axis-Gear System</td>
</tr>
<tr>
<td>• State Variables Equations</td>
<td>• Pulley System</td>
</tr>
<tr>
<td>• Input-Output Equations</td>
<td>• Position Control with Feedback Response</td>
</tr>
<tr>
<td>• Matrix Equations</td>
<td></td>
</tr>
<tr>
<td>• Block Diagrams</td>
<td></td>
</tr>
<tr>
<td>• Simulink</td>
<td></td>
</tr>
<tr>
<td>• PID Control</td>
<td></td>
</tr>
<tr>
<td>• Time Constants</td>
<td></td>
</tr>
<tr>
<td>• Step, Impulse, Sinusoidal, and Constant Inputs</td>
<td></td>
</tr>
<tr>
<td>• Filtering</td>
<td></td>
</tr>
<tr>
<td>• Laplace Transforms</td>
<td></td>
</tr>
<tr>
<td>• Motor Sizing</td>
<td></td>
</tr>
</tbody>
</table>

B) Professor Generated Lists

<table>
<thead>
<tr>
<th>Preliminary Concepts</th>
<th>Preliminary Modules</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Control Theory</td>
<td>• Dual Clock Control System</td>
</tr>
<tr>
<td>• Relationship Between Physics and Laplace Transformations</td>
<td>• Tanks and Pumps</td>
</tr>
<tr>
<td>• Application of Poles and Zeros</td>
<td>• Mystery Box with Known Inputs and Outputs</td>
</tr>
<tr>
<td>• Frequency Response</td>
<td>• Degrees of Freedom</td>
</tr>
<tr>
<td>• Natural Frequency</td>
<td>• D.C. Servo Motor</td>
</tr>
<tr>
<td>• Stability and Instability in Systems</td>
<td>• Heaters and Thermocouple Temperature</td>
</tr>
<tr>
<td>• Filtering</td>
<td>• Mass Spring Damper</td>
</tr>
<tr>
<td>• Time Constants</td>
<td>• Filtering with a Speaker</td>
</tr>
<tr>
<td>• Operational Amplifiers</td>
<td>• Excited Cantilever Beam and Choosing Appropriate System Poles</td>
</tr>
<tr>
<td>• Motors</td>
<td></td>
</tr>
<tr>
<td>• P, PI, PID control</td>
<td></td>
</tr>
<tr>
<td>• Rotational Systems</td>
<td></td>
</tr>
</tbody>
</table>

**Figure 2.1:** A) Team and B) Professor generated concepts and modules
With the lists generated, the team began the process of concept narrowing. The concepts were put through multiple rounds of screening, including comparison with the design metrics, what the module teaches, the required interface, the setup required, safety concerns, ability to deliver a prototype on time, and team preference. A complete description of the process can be found in the ME4054 Design Report.

Once the screening was done, the team was left with three modules. They were: a speaker module for filtering sounds and sine waves, a mass spring and damper system for system modeling, and a position control module for PID control.

The rest of the semester was devoted to building and testing the speaker module (renamed to audio filtering module), position control module, and mass spring damper module (Fig. 2.8). They were presented with great success at the ME4054 design show. Again, more information on them can be found in the ME4054 Design Report. These three modules became the basis for all future work that was to come.

**Figure 2.8a:** The speaker (audio filtering) module
Figure 2.8b: The position control module

Figure 2.8c: The mass spring damper module
Figure 2.8d: The PIC Control box. This box was the link between the modules and the computer, and housed the PIC chip, motor controller, and other electronics.

3. Generation 1

3.1. Overview

At this point, the three modules underwent extensive review. The most dramatic change was the postponement of work on the position control module till generation two, in order to focus efforts on the audio filtering and mass spring damper modules, which needed more development than the controls module. Also at this point, distributed labs changed from a team project with a group of seniors in mechanical engineering, to this Masters Thesis project.

3.2. ¼ Car Module

As for the mass spring damper module, the largest change was the shift from an exited cantilever beam to a system of sliding weights to model a ¼ car (Fig. 3.1).
was done to help students relate to the system they were modeling, and also because it is a classic system to model in lectures and textbooks.

Figure 3.1: The block diagram of a quarter car that the module is based on.

A drawing and a photograph of the module are shown in Figures 3.2 and 3.3 respectively. The module is driven by a motor spinning a disk to which an offset link is attached. This link is attached to a slider on an aluminum rod. As the motor spins, the disk turns and causes the link and slider to move up and down 1cm vertically. The frequency the motor is driven at can be changed by the student. This frequency is determined via a photo reflector which detects the metal link as it passes by. Thus, each pass of the link is one revolution of the disk, and the frequency can be determined. The slider driven by the link acts to simulate the road that the car drives over, and the faster the motor is driven, the more bumpy the simulated road. Next comes the first spring, which represents the springiness of the tires. Above that is the first mass, which represents the weight of the tires and suspension system. That first mass consists of two
cylinders which are connected via a press fit and adhesive. The second spring simulates
the spring in the suspension, while the top mass is the mass of the car. A piece of foam is
affixed to the top of the second mass in order to represent the damping elements in the
suspension. The amount of damping can be adjusted by placing rubber bands on the
foam to control how tightly the foam presses against the top cylinder of the first mass.
Connected to each mass is a ring magnet. Using Hall Effect sensors mounted to the PCB,
the position of the masses can be determined, and then sent to the computer for the
student to see. Based on the instructions, or simply to experiment on their own, various
combinations of springs, masses, and damping elements can be added or removed by the
students, and then excited by the motor.
Figure 3.2a: Drawing of Mass Spring Damper Module

Figure 3.2b: Close up of linkage
Figure 3.3: Photograph of the Mass Spring Damper module
The user interface for the Generation 1 Mass Spring Damper Module is shown in Figure 3.4. The Motor Control section of the interface allows the student to control the speed of the motor, and hence the frequency of the input to the system, and to turn the motor on or off. This section also displays the frequency of the input in hertz.

![Mass Spring Damper Visual Basic user interface.](image)

**Figure 3.4:** Mass Spring Damper Visual Basic user interface.

Below the “Motor Control” section is the “Positions” section. Here the student can view real time displays of the positions of the two masses, as well as the input slider. Much like an oscilloscope, students can adjust slider bars to move green lines over the displays to take measurements. Based upon where the students places the lines, the software will display the magnitudes of the top and bottom graph, and also the phase
difference between them. Students can choose to display any combination of the input, mass 1 or mass 2 on either graph, as suits their needs or preference. They can also start and stop the real time feed to the displays if they wish to examine a particular waveform more closely. The largest section of the interface is for the student to view and create bode plots. By selecting a 2\textsuperscript{nd} or 4\textsuperscript{th} order system and entering in the values for the various system elements, the students can have the software display an ideal bode plot, just as they would see in their textbooks. Then, as the student took measurements of the real system, they can enter the frequency and magnitudes they measure, and plot them on top of the ideal plot. Screen shots of their bode plots were often used as a means of grading the labs, and making sure students completed them.

The lower right section of the interface displays the average position of both masses over the past four seconds. While not terribly useful while the system is being excited by the motor, these can be used to measure static values, such as the deflection in the system caused by adding an additional mass.

The schematic, PCB layout, bill of materials, and additional photos of this version of the Mass Spring Damper module can be found in appendix A-2.

3.3. Audio Filtering Module

The Audio Filtering Module had several improvements made to it as well. For instance, the pre-made, and expensive, filter cards were replaced by a breadboard upon which students could construct a wide range of filters. An amplifier and volume knob was added to improve the sound coming out of the headphones. Also, as can be seen in
Figures 3.5 and 3.6, the module was greatly reduced in size to aid in storage and transportation.

**Figure 3.5:** Picture of Audio Filtering Module

**Figure 3.6:** Drawing of Audio Filtering Module
In addition to the power jack, there are three audio jacks on the module. The first of these is connected to the “Line Out” jack on the student’s computer. This takes the audio signal from the PC and brings it to the board. After passing through the filter, which students construct on the provided breadboard, the signal is sent back to the computer via the “Line In” jack. In addition, the signal is sent to the headphone jack, allowing the student to hear the sound on a pair of headphones or speakers. By toggling the switch just below the jack, they can select if they wish to hear either the filtered or non-filtered sound. In all cases, the PC is sent both the filtered and unfiltered sound via the line in jack, by sending one signal over the left channel, and the other over the right. This allows the user interface to displace both sound waves simultaneously.

The user interface for this module is similar in format to the Mass Spring Damper module, allowing a student familiar with one to quickly grasp the other (Fig. 3.7). The bode plot area works exactly in the same fashion as before, the only difference being the choices of high or low pass filter instead of 2nd or 4th order systems. The “Scope” section displays the input and output signals, and students can take measurements in the same way as before.

The Sound Generator section is where students can start and stop two independent sine waves which the software will generate. They can also control the volume and the frequency of the generated sine waves. For students with multiple sound cards, this section also allows them to select the one they wish to use to play their sound files from. In the FFT section a Fast Fourier Transform of the sine wave is displayed. Here the student is able to select whether they wish to view the input or output wave, and adjust the gain. The combination of audio, sinusoidal, and FFT displays give the students a
variety of ways to learn about, and experiment with, filters. More pictures, along with the bill of materials, PCB layout, and schematic for the Audio Filtering Module can be found in appendix A-3.

Figure 3.7: Audio Filtering Module Visual Basic user interface.

3.4. Evaluation

Thirty modules were constructed and loaned to students for evaluation. This evaluation took place in the summer semester of 2004 in the ME 3281 System Dynamics and Controls course taught by Prof. Kelso.

First, students were given a short test (see appendix B-6) on the subject matter found in the lab. This was done before the instructor had even gone over the material, to gauge how much of the subject matter students knew before lecture, homework or module instruction.
There were 27 students in the class, which would have allowed for every student to take home a module. However, in order to compare the results between students who did and did not use the module, the class was randomly split in half. One half of the students then received the modules, and over the next week performed them. During that week, the entire class was instructed by the teacher and given homework assignments. After that week, the students took the same quiz, indicating on the quiz whether or not they had done the module. This allowed a comparison between students who go through the class in the traditional way, and those who had a lab module as well as the lecture and homework. Next, the other half of the students were given the lab module and one week later the entire class took the quiz for the third time. The results of all three tests are shown below (Fig. 3.8). Students were also asked to rank how confident they were in their answers with results shown in Figure 3.9. Means and standard deviations of this data is shown in Figures 3.10 and 3.11.

![Student Score Data](image)

**Figure 3.8:** Student test results
Figure 3.9: Student confidence data

Figure 3.10: The means and standard deviation for the test data.
Figure 3.11: The means and standard deviation for the student confidence data.

The results were conclusive. An unpaired t test performed on the test scores before and after using the module give a miniscule 0.00000859% chance that there was no significant difference between the means of the two groups.

As was expected, scores varied wildly before the students received any sort of formal instruction. Some students, admitting in their comments to having prior knowledge of the subject, did well, while those who had never heard of the concepts before did poorly. Confidence data also varied wildly.

Perhaps the most striking of the results comes from comparing the mid-test scores of students who had and had not done the lab. While at the time of the test, all students had been instructed by the teacher in the same manner, the scores of those with the labs is higher. Confidence data also shows that for the students with modules not only are they correct in their answers, but they are far more certain of their answers.
The post test also shows that the first randomly selected half of the class wasn’t a fluke, and that once all the students received instruction from the lab module, scores and confidence were higher across the board.

Once the module distribution was over, and the modules turned in, students were asked a series of questions regarding what they thought of the labs. Those results are tabulated in Table 3.12.

<table>
<thead>
<tr>
<th>Question</th>
<th>Strongly Disagree</th>
<th>Disagree</th>
<th>Neutral</th>
<th>Agree (4)</th>
<th>Strongly Agree (5)</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>This assignment helped me understand the concept of filtering</td>
<td>0</td>
<td>0</td>
<td>7</td>
<td>10</td>
<td>8</td>
<td>4.04</td>
</tr>
<tr>
<td>This assignment helped me understand the concept of cutoff frequency</td>
<td>0</td>
<td>0</td>
<td>9</td>
<td>12</td>
<td>4</td>
<td>3.8</td>
</tr>
<tr>
<td>This assignment helped me understand the concept bode plots</td>
<td>0</td>
<td>5</td>
<td>5</td>
<td>11</td>
<td>4</td>
<td>3.56</td>
</tr>
<tr>
<td>This assignment helped me understand the readings in my textbook</td>
<td>1</td>
<td>8</td>
<td>16</td>
<td>0</td>
<td>0</td>
<td>2.6</td>
</tr>
<tr>
<td>This assignment helped me understand the lecture notes better</td>
<td>0</td>
<td>7</td>
<td>11</td>
<td>7</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>This assignment helped me with the problem sets related to these concepts.</td>
<td>0</td>
<td>8</td>
<td>12</td>
<td>4</td>
<td>1</td>
<td>2.92</td>
</tr>
<tr>
<td>The questions in this assignment were clear</td>
<td>0</td>
<td>4</td>
<td>5</td>
<td>10</td>
<td>6</td>
<td>3.72</td>
</tr>
<tr>
<td>The audio filtering apparatus was simple and easy to use</td>
<td>2</td>
<td>1</td>
<td>3</td>
<td>11</td>
<td>8</td>
<td>3.88</td>
</tr>
<tr>
<td>The software was easy to setup and use</td>
<td>1</td>
<td>1</td>
<td>5</td>
<td>11</td>
<td>7</td>
<td>3.88</td>
</tr>
</tbody>
</table>

Table 3.12: The results of the student survey. The average is computed by assigning a value to each answer, i.e. 1 for strongly disagree, 2 for disagree, etc, and then averaging those numbers.

These first three questions deal with concepts directly taught by the modules. As can be seen from the average, students tended to agree with this, especially in regard to filtering and cutoff frequency, where there were no negative responses.
On the next three questions, regarding how the modules related to other aspects of the course, the student average tended toward neutral. While students were evenly split in regard to the module’s ability to help them understand the lecture material, they tended to disagree or be neutral toward the idea that it helped them do the homework or understand the book. A more rigorous look at what the book has to say on these topics, as well as what homework questions were given to the student, would be required to shed further light on this.

The final set of questions asked about how easy the various parts of the module were to use. Overall students indicated that everything was clear, and they understood what they were required to do, with the averages falling in the neutral-agree range.

In addition to the questions shown above, students were asked to comment on the good and bad aspects of the modules. Those comments are shown in Table 3.13.

<table>
<thead>
<tr>
<th>Biggest Strength</th>
<th>Biggest Weakness</th>
</tr>
</thead>
<tbody>
<tr>
<td>Provides hand on practice on Bode plots and interactive features allow better understanding</td>
<td>Limit possible errors</td>
</tr>
<tr>
<td>We do it at home</td>
<td>Nothing. I am too lazy to read, this helps bunches</td>
</tr>
<tr>
<td>Ease of use and the more you play the more you LEARN</td>
<td>More complicated looking gadgets. Had computer/software related problems. Had to beg CompSci buddies for help.</td>
</tr>
<tr>
<td>It looks scientific</td>
<td>It would be nice to vary the RC values</td>
</tr>
<tr>
<td>The ability to experience a hands on method of audio filtering</td>
<td>Could use a &quot;print lab&quot; in the menu. My computer wouldn't print it.</td>
</tr>
<tr>
<td>Easy to set up</td>
<td>FFT explanation—there's little in the bode from what I could find (but I didn't look that hard)</td>
</tr>
<tr>
<td>Being able to hear first hand filtered and unfiltered sounds especially in the music part</td>
<td>The technology—I was unable to get my equipment to work properly on my own-making it easier to get help would have made a big difference.</td>
</tr>
<tr>
<td>The packet was well written and clear</td>
<td>The module wouldn't uninstall easily</td>
</tr>
<tr>
<td>Easy to set up</td>
<td>It's relevancy could be clearer</td>
</tr>
<tr>
<td>Helps the students understand filtering well</td>
<td>The software was very finicky. Didn't work on Windows 2000</td>
</tr>
<tr>
<td>It's simplicity</td>
<td>Better instruction and illustration of how it works</td>
</tr>
<tr>
<td>It exposed me to something real rather than just words and pictures in a book. I really enjoyed it.</td>
<td>It's dumb to have everyone fill out the post test 3 times. Do it once before and once after. That's not that difficult to arrange.</td>
</tr>
<tr>
<td>Clearly shows how to use Bode plots and cutoff frequency and filtering. A good idea</td>
<td>Needs to be friendly to all types of computers laptops and desktops alike. I don't have time to come down to school in my off time to work on a lab. Luckily I was able to install it at work.</td>
</tr>
<tr>
<td>Seeing results easily without getting lost in too much mumbo jumbo</td>
<td>Software needs to be easier to set up</td>
</tr>
<tr>
<td>Good learning aid if done with or after frequency response</td>
<td>Clarity on what is expected in write-up and more explanation of how the circuit works</td>
</tr>
<tr>
<td>Relating the filter concept to an important class of signals</td>
<td>I wish we had longer to work on this assignment. People's schedules are busy in the summer.</td>
</tr>
<tr>
<td>It was easy to use</td>
<td></td>
</tr>
</tbody>
</table>

23
Table 3.13: Student Comments

As can be seen from the comments, students were particularly excited about the hands-on aspects of the lab. Students even enjoyed the lab to the point of telling their friends about it, and when those friends didn’t get to do the lab when they took the course (due to the labs not being distributed that semester), they complained to the professor. It should be noted, however, that while students reported that the labs were easy to use, setting up the hardware proved to be a challenge for some. Of particular difficulty was the plugging of cables into the wrong jack, and configuring the volume settings.

3.5. Lessons Learned

In the end, a great deal was learned from Generation 1. From determining which concepts to focus on, to designing inexpensive modules that still performed like ideal systems, distributing cross-platform code, assembling and distributing 30 modules, and seeing how students interacted with them, the value cannot be understated. What follows is a list of the key lessons that were learned during that entire generation 1 process.

1) Extra modules need to be manufactured, as some will not work, and debugging them is a diminishing returns game, i.e. you can fix a few in a couple hours, but some of them would take days to determine what is going wrong on.

2) It takes time and manpower to assemble large quantities of modules, even when done assembly line style. (It took over 500 man hours to assemble 30 of each module).
3) Using pre-made parts, and putting as many holes on the PCB board as possible can greatly shave time off of construction.

4) Some modules will work better than others for reasons that are difficult to determine.

5) Staying away from sensors that require precise placement, such as Hall Effect sensors, is advisable.

6) Automating as much as possible about the setup and installation will save tons of headaches when students are trying it themselves.

7) Modules returned by students were in surprisingly good condition, so they do appear to care for their modules.

The changes that were made in generation 2 that respond to generation 1 weakness are discussed in this report.

4. Generation 2

4.1. Overview

Generation 2 modules attempted to take the lessons learned in making the Generation 1 modules to create the final set of modules which would be used by System Dynamics and Controls courses for years to come. A major component of the Generation 2 modules was the number of prototype modules that were built before the final version was approved. Making the prototypes, seeing how they performed, and then changing what needed to be changed, rather than trying to predict what would happen in advance, allowed for a large quantity of iteration to take place in a short time. Simulink models were used to verify that the modules were performing as they should while they were still
being prototyped. This allowed the proper changes to be made early on, rather then just using Simulink as a check at the end, when it was too late to change anything.

Due the desire to optimize the FR/PID module, work on the audio filtering module was put off for a later time. Suggestions for future work on this module are described in chapter 6.

The largest change was that the MSD module was combined with the Position Control Module, and the ¼ car dropped due to its complexity, to create the Frequency Response/Proportional-Integral-Derivative Control (FR/PID) module. A diagram of the FR/PID module is shown in Figure 4.1 and a photograph in Figure 4.2.

4.2. FR/PID Hardware

To do this, frequency response data would be gathered from a rotational system, rather then a linear one as was the case in previous generations. Also, the adjustable damper was removed, as the inherent friction in the system would suffice, and it wasn’t felt that a great deal of educational value would be added by having an adjustable damping element.

The spring is a rubber band which wraps around the inertia disk and a post. By making a complete revolution around the post, the band is prevented from acting like a belt on a pulley. As the rubber band is sized to be stretched when it is installed, and the disk programmed to only rotate a small amount, the rubber band acts as a linear spring. See Test 1 in Section 4.4.
Figure 4.1: Diagram of the Frequency Response/ PID Module
The spring constant varies greatly between rubber bands, even of the same size, brand, or out of the same box. Thus, as one of the first steps when conducting the lab, students will measure the spring constant of their rubber band. To accomplish this, students will turn the module on its side, wrap a string around the inertia, and connect a bag to the other end of the string, as shown in Figure 4.3. Students will then place pennies, or any other object of known weight, into the bag, and record the rotation of the disk. Using this information, students will be able to calculate the spring constant for their rubber band. An example of this is shown in test 2.
Figure 4.3: A string is connected to the inertia wheel, wrapped around the disk several times, with a bag hanging down the other end of the string. The pennies are then placed in the bag, one at a time, and the rotation of the inertia is recorded via the user interface.

The steel inertia wheel is held in place with a magnet. This brings several useful features to the module. First, the ability to quickly remove and reconnect the inertia makes it simple for students to change the inertia of their system without a large time commitment or extra tools. The magnet will also allow for future expansion of the module in the form of other inertias, which can be any magnetically attracted object. In this way not only different sized disks could be used, but any object a student wishes, from paperclips to steel cans. Finally, the magnet acts as an emergency disconnect in the event that the model is dropped. Any large force, such as from a fall, that the model
endures will simply separate the inertia from the motor shaft, protecting the shaft from bending.

The gears used in the module are inexpensive and easily machinable, allowing further customization if desired by an instructor for a classroom demo. The gears, being press fit onto the bushing and the potentiometer are easily removed and different sizes or ratios added if desired. It is also possible to add an entire gear chain next to the module if desired, with the motor driving the system and the potentiometer taking the final measurement. These extra gear chains would allow topics such as shaft misalignment to be taught.

Around the motor is an area purposely kept clear of traces and components. This will allow for the motor to be changed in the future should that become desirable. Also, this clear space is filled with a solid plate which helps with transferring heat away from the motor. This plate runs all the way to one of the legs, further increasing the surface area over which heat can be transferred.

The module stands on four suction cup feet, which use electrical standoffs to accommodate the motor. These feet provide stability during resonance. When engaged, the suction cups remain active for days at a time.

If it ever becomes desirable to change the code on the PIC18F2455, an in-circuit programming jack is present (J1). Thus, any modifications desired can be made to the PIC chip without any need for soldering or installing a new chip.

The L293D H-bridge motor controller is placed such that it is exposed to the open air, allowing for greater heat transfer than would be possible under the gears. The
rotation of the gears also helps to stir up the air above the motor controller, aiding convection.

Each of the plugs fit into only their respective jacks, which eliminates the problem of plugging in components incorrectly. As an added debugging feature, the LED lights up when the module is powered correctly.

For ease of transportation, the module, along with all required cords and parts, fits into a sturdy 5 x 5 x 4" corrugated mailing box. As this box is designed to go through the mail system, it is well suited for journeys in a student backpack.

Appendix A contains photographs showing the final FR/PID module from various angles.

4.3. FR/PID Software

There are two pieces of software that keep the FR/PID module running: the C code firmware on the PIC chip and the Visual Basic user interface running on the student’s computer. The firmware comes preinstalled on the module when the student receives it. It can be altered via the ICD port by the course staff if desired. The user interface however, must be downloaded by the student and then installed on their computer. This code can be altered by the course staff in the standard way.

The software flow charts appear on the following pages in Figures 4.4-4.10. The code as well as a more in-depth analysis can be found in Appendix B-7.
Setup and Initializations

While True

Wait for Timer 0 to Fire

Get Potentiometer Position

Get and Send USB data

Check Command Value

Figure 4.4: C Code (Always running)
Figure 4.5: C Code functions
While True

Wait for USB Data

Is PIDForm Visible?

Is FRForm Visible?

Does PIDForm.BGraph.Caption = "Freeze Plot"?

Graph Data

Does FRForm.BGraph.Caption = "Freeze Plot"?

Graph Data

Does FRForm.Zoom.Caption = "UnZoom"?

Average 100 data points

Set Axis and Magnitude

Is CollectData = "True"?

Send Data to file

Figure 4.6: VB Code: DLABForm (Runs at startup)
Figure 4.6: VB Code: DLABForm (Runs at startup)
Figure 4.7: VB PID code
Figure 4.8: VB PID Code Continued
Figure 4.8: VB PID Code Continued
Send nsamp and Command 5

Send Command 2

Figure 4.9: VB FR Code
Press "Update"

Theoretical Bode Plot

Clear Observed Data*

Press a keyboard arrow key

Send nsamp and Command 5

Increase/Decrease Frequency Scrollbar

Increase/Decrease Frequency Label

Current Input = Pending Input

nsamp = HzLookup (Frequency Caption)

Press "Plot Point"

Plot New Point

Add New Data to Output Magnitude and Frequency Array

Increment TotalPoints by 1

Press "Clear Last Point"

\[ i = \text{TotalPoints} - 1 \]

Remove Point i from the Bode Plot

TotalPoints = i

Press "Collect Data"

Caption="Stop Collecting"

Collectdata=True

Press "Stop Collecting"

Caption="Collect Data"

Collectdata=False

\[ \text{Collectdata} = \text{True} \]

\[ \text{Collectdata} = \text{False} \]

**Figure 4.10:** VB FR Code
Press
“Zoom”
zoomtotal = 0
zoompoint = 0
Caption=
“UnZoom”
ZoomEnabled=False
ZoomFlag=1

Press
“UnZoom”
zoomtotal = 0
zoompoint = 0
ZoomFlag=0
Caption=
“Zoom”

Reset Axis and Magnitude

Figure 4.10: VB FR Code
Figure 4.10: VB FR Code
4.4. Component Characterization Tests

Test 1: Spring Linearity Test

Objective: To determine if the rubber band configuration is a linear spring.

Methods: Create a mathematical model of the spring and inertia. Using this model, test various amount of rotation in the disk. Next, record the force on the rubber band for various amounts of rotation using the equations in Figure 4.12. Plot the data, and verify that the resulting line is straight, which indicates a linear spring.

\[
\begin{align*}
\theta &= \text{Angle the inertia moves through} \\
R &= \text{Radius of the inertia} \\
L &= \text{Distance from post to inertia along rubber band} \\
L_{\text{ud}} &= \text{Undeﬂected length of the rubber band along the post-inertia edge line} \\
a &= \text{Arc made by the inertia as it moves through } \theta \\
T &= \text{Torque caused by the rubber band} \\
F &= \text{Force of the rubber band} \\
d &= \text{Distance the rubber band is stretched}
\end{align*}
\]

Figure 4.11: The system model, assuming that the rubber band is slightly stretched to begin with, ie \( L > L_{\text{ud}} \).
Calculation for one side of the inertia:

\[ a = r \theta \]
\[ F = K d \]
\[ F = K (L - Lund) + r \theta \]
\[ T = r F \]
\[ T = K ((L - Lund) r + r^2 \theta) \]
\[ T = \text{Const} + \text{Const} \theta \]

Calculation for both sides of the inertia:

\[ F = (K (L - Lund) + r \theta) - (K (L - Lund) + r \theta) \]
\[ F = 2 K r \theta \]
\[ T = 2 K r^2 \theta \]
\[ T = \text{Const} \theta \]

**Figure 4.12**: The system equations.

Thus, the torque created by the rubber band is simply a constant term multiplied by the angle \( \Theta \).

Results: Figure 4.13 shows a graph of the various values of \( \Theta \) and the corresponding

**Figure 4.13**: The torque vs. theta relationship for the spring linearity test shows a linear relationship.
torque from the rubber band (spring) for a fixed value of $K$ (in this case 0.1 lbs/in).

Discussion and Conclusions: It was found that using a rubber band, wrapped once around the inertia disk, results in a linear spring. This assumes that the rubber band is slightly stretched to begin with ($L>L_{und}$). It is also interesting to note that so long as both sides of the rubber band stay taut, $L$ and $L_{und}$ have no effect on the generated torque.

Test 2: Student Calculation of the Spring Constant

Objective: To calculate the spring constant for a rubber band using a method students can do on their own at home.

Methods: Turn the module on its side, wrap a string around the inertia, and connect a bag to the other end of the string, as shown in Figure 4.3. Next add washers, pennies, or any other object of known weight, to the bag, and record the rotation of the disk using the measuring bars in the frequency response software. Continue to add weights, recording the deflection each time. It is helpful to hold the bag up, allowing the string to go slack some, add the weight, and then lower the bag. This prevents the impact of the weights from over-deflecting the inertia wheel. Once a set of deflections and weights has been recorded, calculate the spring constant for the rubber band by entering the data on a spreadsheet, cross-plotting, fitting a line, and then finding the slope of that line. Once this is done, the student will have a series of data containing the net force on the disk (the added weight times gravity) and the corresponding deflection of the disk. The student can then graph this data, and plot a trend line to the graph. Once the trend line is plotted,
they will have the spring constant in Newtons/degree. Note that this is a linear spring
constant as defined in Test 1.

Results: The above method was performed on a rubber band, and the results are shown
below in Figure 4.14.

<table>
<thead>
<tr>
<th>Force (N)</th>
<th>Deflection (deg)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.196</td>
<td>1.5</td>
</tr>
<tr>
<td>0.2952</td>
<td>3</td>
</tr>
<tr>
<td>0.3744</td>
<td>6.2</td>
</tr>
<tr>
<td>0.3136</td>
<td>8.4</td>
</tr>
<tr>
<td>0.3628</td>
<td>19.8</td>
</tr>
<tr>
<td>0.362</td>
<td>20.6</td>
</tr>
<tr>
<td>0.4312</td>
<td>22.6</td>
</tr>
<tr>
<td>0.4704</td>
<td>23.7</td>
</tr>
<tr>
<td>0.5096</td>
<td>25</td>
</tr>
<tr>
<td>0.5408</td>
<td>26</td>
</tr>
<tr>
<td>0.568</td>
<td>27</td>
</tr>
<tr>
<td>0.6272</td>
<td>30</td>
</tr>
<tr>
<td>0.6664</td>
<td>35</td>
</tr>
<tr>
<td>0.7056</td>
<td>39</td>
</tr>
<tr>
<td>0.7448</td>
<td>40</td>
</tr>
<tr>
<td>0.784</td>
<td>42.7</td>
</tr>
</tbody>
</table>

**Figure 4.14:** The data generated during the spring constant determination test. As can be
seen in the graph, the spring constant of this particular rubber band was 0.0141 N/Deg.

Discussion and Conclusions: As rubber bands are not made of consistent quality, their
spring constants are not uniform, even if two bands are from the same box. Also, as a
band ages, it tends to become more brittle, which greatly alters its K value. Hence, the
above method can be used by the student to determine the current spring coefficient of
their rubber band, using the lab kit itself, and simple materials found around a typical
dorm or home.
Test 3: Student Calculation of Inertia

Students will call upon their knowledge of physics to find the inertia of their collar. They will be told that the collar is made of steel, and possibly its dimensions, and with that information they should be able to calculate the inertia as follows:

\[
\text{Inertia} = \frac{1}{2} \times \text{Mass} \times (R1^2 + R2^2) \\
\text{Mass} = \text{Volume} \times \text{Density} \\
\text{Volume} = H \times \pi \times (R1^2 - R2^2) \\
\text{Inertia} = \frac{1}{2} \times \left[\frac{5}{8} \times \pi \times (0.75^2 - 0.5^2)\right] \times 0.284 \times \left[(0.75^2 + 0.5^2)\right] \\
\text{Inertia} = 0.071 \text{lb/in}^2 = 207 \text{ g/cm}^2
\]


\[
\text{Density} = 0.284 \text{ lb/in}^3 \\
R1 = 0.75 \text{ in} \\
R2 = 0.5 \text{ in} \\
H = 5/8 \text{ in}
\]

Figure 4.15: Calculation of Inertia

Test 4: Student Calculation of Damping

The system is designed to minimize friction and other damping effects. One of the key specifications for selecting the potentiometer and motor were models with low running friction. The pilot holes for the gears are precision drilled into the circuit board itself during creation, which greatly aids in ensuring a good mesh without binding.

However, in all systems, even those designed to minimize friction and damping, some is always present, and the FR/PID module is no different. Thus, the small amount of friction in the module should still be accounted for. As it is difficult for a student to calculate the kinetic friction of the module at home, they will be provided with a good
estimate of it, and encouraged to change this value slightly if their results seem to indicate they have more or less damping in their system.

5. Evaluation

The FR/PID was evaluated against the metrics laid out in section 1.5 (Table 5.1):

<table>
<thead>
<tr>
<th>Metric</th>
<th>Reason</th>
<th>Goal</th>
<th>FR/PID Spec</th>
<th>Pass / Fail</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accurate</td>
<td>To compare to equation results</td>
<td>*See Below</td>
<td>*See Below</td>
<td>Pass</td>
</tr>
<tr>
<td>Demonstrates Concept</td>
<td>Module designed for learning</td>
<td>95% of students understand concept after performing the lab</td>
<td>TBD</td>
<td>TBD</td>
</tr>
<tr>
<td>Easy to Manufacture</td>
<td>Manufacturing is done in house</td>
<td>100 modules made in under two weeks by undergrads</td>
<td>TBD</td>
<td>TBD</td>
</tr>
<tr>
<td>Lab Experiment Documentation</td>
<td>Aids student and instructors performing the labs</td>
<td>Exists</td>
<td>TBD</td>
<td>TBD</td>
</tr>
<tr>
<td>Light</td>
<td>Ease of transport</td>
<td>&lt;5lbs</td>
<td>1lb 14oz</td>
<td>Pass</td>
</tr>
<tr>
<td>Low Assembly Time</td>
<td>Assembling the lab is not the focus</td>
<td>&lt;10 minutes to setup, including code installation</td>
<td>1 Minute</td>
<td>Pass</td>
</tr>
<tr>
<td>Mass Production Cost</td>
<td>Resource efficient, no large loss if students loose them</td>
<td>&lt;$100 parts cost</td>
<td>$57.57</td>
<td>Pass</td>
</tr>
<tr>
<td>Multiple Concepts Taught in 1 Module</td>
<td>Adjusts to different faculty, and allows students more areas to explore. Lowers cost and number of required modules.</td>
<td>&gt;3 topics can be taught by a module</td>
<td>16</td>
<td>Pass</td>
</tr>
<tr>
<td>PC Compatibility</td>
<td>Needs to work in a variety of environments</td>
<td>Communicates with Windows 98 or newer PCs and laptops, over USB</td>
<td>Communicates with Windows 98 or newer PCs and laptops, over USB</td>
<td>Pass</td>
</tr>
<tr>
<td>Power</td>
<td>Needs to get power in many locations (dorms, homes, computer labs, etc)</td>
<td>Uses standard wall outlet</td>
<td>Uses standard wall outlet</td>
<td>Pass</td>
</tr>
<tr>
<td>-------</td>
<td>-----------------------------------------------------------------</td>
<td>--------------------------</td>
<td>--------------------------</td>
<td>------</td>
</tr>
<tr>
<td>Rugged</td>
<td>Survive trips in backpack</td>
<td>*See Below</td>
<td>*See Below</td>
<td>Pass</td>
</tr>
<tr>
<td>Safe</td>
<td>Keeps students healthy</td>
<td>Low voltages/currents, no sharp edges, no injuries</td>
<td>12V, 500mA</td>
<td>Pass</td>
</tr>
<tr>
<td>Simple</td>
<td>Operation easy to understand</td>
<td>95% of students report no trouble performing the lab</td>
<td>TBD</td>
<td>TBD</td>
</tr>
<tr>
<td>Simple Student Interface</td>
<td>Easy to use and change</td>
<td>Visual Basic interface</td>
<td>Visual Basic interface</td>
<td>Pass</td>
</tr>
<tr>
<td>Small</td>
<td>Ease of transport and storage</td>
<td>&lt;1900 in³</td>
<td>100 in³</td>
<td>Pass</td>
</tr>
<tr>
<td>Standardized Parts</td>
<td>To easily obtain replacements</td>
<td>100% of materials from suppliers, not surplus</td>
<td>100% of materials from suppliers</td>
<td>Pass</td>
</tr>
<tr>
<td>User Friendliness</td>
<td>Encourages student use</td>
<td>=&gt;95% of students found labs enhance their learning experience</td>
<td>TBD</td>
<td>TBD</td>
</tr>
</tbody>
</table>

**Table 5.1:** The design metrics for the project, along with their final status.

*To verify we had a durable, accurate and overall well built module, the following evaluation tests were performed (Table 5.2):

<table>
<thead>
<tr>
<th>Test #</th>
<th>Description</th>
<th>Results</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Module can run at resonance for 1 hour with no damage</td>
<td>Pass</td>
</tr>
<tr>
<td>2</td>
<td>Module can run continuously for 12 hours without significant part wear</td>
<td>Pass</td>
</tr>
<tr>
<td>3</td>
<td>Suction cups hold for 12 hours</td>
<td>Pass</td>
</tr>
<tr>
<td>4</td>
<td>Module fits inside storage box</td>
<td>Pass</td>
</tr>
<tr>
<td>5</td>
<td>Module functions after the storage box containing the module goes through a rigorous backpack trip</td>
<td>Pass</td>
</tr>
<tr>
<td>6</td>
<td>Module functions without damage after falling 5 feet onto concrete, then down a flight of concrete stairs</td>
<td>Pass</td>
</tr>
<tr>
<td>7</td>
<td>The module runs on 5 random windows 98 or newer machines</td>
<td>Pass</td>
</tr>
<tr>
<td>8</td>
<td>PID responses match theory</td>
<td>Pass</td>
</tr>
<tr>
<td>9</td>
<td>Bode Plot matches theory</td>
<td>Pass</td>
</tr>
</tbody>
</table>

**Table 5.2:** Durability/ruggedness/accuracy tests.
Evaluation Test 1:

Objective: Verify that the module can run at resonance for 1 hour with no damage or appreciable wear to any parts.

Methods: First, three data sets were collected, just as if the student was doing the lab. Then the module was run at its resonant frequency of ~4 Hz for an hour. After that time, data was again collected, in the same manner as before. If the results fell within 2% of the values recorded earlier, then the module was considered to have passed this test.

Data: The four sets of data are shown below in Figure 5.3. The dots are the three initial sets of data, and the “X”s are for the data taken after running at resonance for 1 hour. The data can be found in appendix B-6.

Discussion/Conclusion: The data collected after running at resonance is within 2% of the three data sets collected ahead of time. The module passed this test.

Figure 5.3: Before and after data for test 1. The dots were taken before the hour long test, the X’s were taken after.
Evaluation Test 2:

Objective: Verify that the module can run continuously for 12 hours without significant part wear or damage.

Methods: As in the previous test, three sets of data were collected, just as if the student was doing the lab. Then the module was run for over 12 hours (the actual run time was over 16 hours). After that time, data was collected again, in the same manor as before. If the results fell within 2% of the original values recorded earlier in test 1, then the module was considered to have passed this test.

Data: The four sets of data are shown below in Figure 5.4. The dots are the three initial sets of data, and the “X”s are for the data taken after running for over 12 hours. The data can be found in appendix B-6.

![Figure 5.4: Before and after data for engineering test 2. The dots were data taken before the test, the “X”s were taken after.](image-url)
Discussion/Conclusion: The data collected after running is within 2% of the three data sets collected ahead of time, thus, the module has passed this test.

**Evaluation Test 3:**

Objective: Verify that the suction cups hold for 12 hours.

Methods: The suction cups were moistened with water to ensure a good seal, and then the module pressed onto the table. After 12 hours had passed, the module was checked to verify that it was still connected to the table via the suction cups. If this was accomplished, then the module was considered to have passed this test.

Data: After 12 hours, the module was still firmly connected to the table. All four suction cups were still engaged, and the module was in no danger of being knocked over.

Discussion/Conclusion: The module has passed this test.

**Evaluation Test 4:**

Objective: Verify that the module fits inside the storage box.

Methods: The module, along with power supply and USB cable was placed inside the box. The box was then securely closed. If this was accomplished, then the module was considered to have passed this test.

Data: As can be seen in the picture below (Fig. 5.5), the module fits inside the box.
Figure 5.5: The module and all its required parts inside the storage box.

Discussion/Conclusion: The module has passed this test.

Evaluation Test 5:

Objective: Verify that the module functions after the storage box containing the module goes through a rigorous backpack trip

Methods: The module was placed in its storage container, and then the container placed into a backpack. In addition several textbooks and other typical school supplies will be placed in the backpack. The backpack and its contents were then subjected to a rigorous journey including dropping under chairs, banging against floors and doors, and other hazards a backpack may be subject to. If the results fell within 2% of the original values recorded in test 1, then the module was considered to have passed this test.

Data: The four sets of data are shown below in Figure 5.6. The dots are the three initial sets of data, and the “X”s are for the data taken after the backpack trip. The actual numbers can be found in appendix B-6.
Discussion/Conclusion: The data collected after running is within 2% of the three data sets collected ahead of time, thus, the module has passed this test.

**Evaluation Test 6:**

Objective: Verify that the module functions without damage after falling 5 feet onto concrete, then down a flight of concrete stairs.

Methods: In an effort to recreate a worst case scenario test, the module was dropped from chest height, (roughly five feet) onto a concrete platform, landing on its side. This platform was at the top of a flight of concrete stairs. The module was dropped such that it proceeded to bounce down the flight of stairs, coming to rest after striking the steel barricade at the bottom of the staircase (Fig. 5.7). When that was complete, the module
was examined for damage, and any components that had gotten loose or misaligned. If no parts required replacement, then the module was considered to have passed this test.

Figure 5.7: The concrete staircase where test 6 was performed

Data: After the fall, all components of the module were in good condition, and correctly aligned. The inertia collar did separate from the magnet, but no damage was sustained and the collar connected back onto the magnet without any problem.

Discussion/Conclusion: The separation of the inertia collar from the magnet (and hence motor shaft) served to protect the motor shaft from any high impact forces and possible misalignment. As all components were in good condition after the fall, the module was considered to have passed this test.

Evaluation Test 7:

Objective: Verify that the module runs on five random windows 98 or newer machines

Methods: The completed module was plugged into five different windows machines, ranging from home computers to computer lab computers. The compiled Visual Basic
code was downloaded onto these machines. If all five of the computers successfully
detect the module and correctly run the code, then the module was considered to have
passed this test.

Data: The module ran on five random windows 98 or newer machines.
Discussion/Conclusion: The module has passed this test.

**Evaluation Test 8:**

Objective: Verify that the module can generate PID responses that match theory.

Methods: A series of PID responses were generated by the module, and screen shots
taken. These responses were overlaid with theoretical, linear, PID responses generated
by Matlab. The module’s response and the linear Matlab response should be similar. If
they were, then this module was considered to have passed this test.

Data: To begin, a simple P response was analyzed (Fig. 5.8).

![Small Kp](image)

**Figure 5.8:** Initial Proportional Controller
This is a typical under damped P response. The rise time is relatively quick, there
are several overshoot peaks, and then the response levels off with some steady state error.
The unmodeled nonlinearities in the system (especially friction) cause the module to
separate from the linear Matlab model. Next, the value of Kp was increased (Fig. 5.9).

![Medium Kp](image)

**Figure 5.9**: Medium Gain Proportional controller

As expected, we see more and larger overshoot spikes, along with an increase in
the settling time. As always with pure proportion control, the steady state error remains.
To complete the study of proportional control, the Kp gain was again increased (Fig.
5.10)
Here again an increase in the number of oscillations was observed. Also, in both the theoretical and real cases, the oscillations clearly are decreasing in amplitude as time goes on, as was the case with the previous values of Kp.

Next, PD responses were analyzed. The value of Kp was held constant at the middle value used before, and only Kd was varied.
These results should be compared with Figure 5.9, which uses the same Kp value, but doesn’t have the additional effects of Kd. As was expected with the addition of derivative control, a marked decrease in the number and size of overshoot oscillations was noted. The settling time was also greatly decreased as a result. Figure 5.12 shows the results of increasing the Kd value.
Figure 5.12: Medium Gain Proportional + Derivative Controller

Now all but the first overshoot hump was eliminated. The settling time was again decreased, as the controller draws nearer to having no overshoot. The final increase in Kd is shown in Figure 5.13 below.

Figure 5.13: Large Kd Proportional Derivative Controller
With this increased Kd value, the module has become over damped. No oscillations were seen, and the steady state error was evident. With the addition of integral control, it was hoped that the steady state error could then be eliminated. Below in Figure 5.14, integral control is added.

![Small Ki Graph](image)

**Figure 5.14:** Initial Proportional Integral Derivative Controller

While the addition of integral control didn’t eliminate the steady state error, it did reduce it. Also, the rise time of the system was improved. The Ki gain was then further increased, and the results shown in Figure 5.15.
With this increase in integral control, a critically damped system was achieved. There is no steady state error or overshoot, and the rise time is short and crisp. For completeness, the Ki gain was further increased, as shown in Figure 5.16.

**Figure 5.15:** Critically Damped Proportional Integral Derivative Controller

**Figure 5.16:** A large Ki value Proportional Integral Derivative Controller
As was expected, higher values of Ki caused the overshoot to reappear, and the settling time to dramatically increase.

Discussion/Conclusion: As the module, even with its inherent nonlinearities, followed the linear Matlab plots and, even more importantly, the expected behaviors associated with changing the control gains, the module was considered to have passed this test.

**Evaluation Test 9:**

Objective: Verify that the module can generate frequency responses that match theory.

Methods: A magnitude bode plot for a range of frequencies (2-20 Hz) was generated by the module, and a screen shot taken. This response is plotted on top of the theoretical response generated by the module. These two graphs should be similar. If they were, the module was considered to have passed this test.

Data: The results of this test are shown below in Figure 5.17. As can be seen, the data points collected align with the theoretical model.

![Bode Magnitude Plot](image)

**Figure 5.17:** The data collected for test 9
Discussion/Conclusion: As the collected date points matched the theoretical graph of the magnitude bode plot, the module was considered to have passed this test.

6. Recommendations and Conclusions

The FR/PID module performs well, meeting all design requirements. It is recommended that the modules proceed to the next stage of development, which is the: construction of 100 modules, creation of lab assignments, distribution to System Dynamics and Control students, and the formal evaluation of student learning.

Further improvements and features could be added to the module as well. Automating the calculation of the spring constant, adding additional inertias, and gear ratio experiments have all been suggested. Expanding the data export features of the module, and incorporation with Matlab could also be explored. Developing a non-linear model in Matlab would also be fruitful in studying the module and ways in which to improve it. Student and professor surveys given after module distribution will likely yield additional improvements.

The Audio Filtering Module must be re-designed. Most needed is a way for the module to utilize the microphone jack, which is far more common and easier to find on PCs then the line-in jack. Also, the dependence of this module on a students sound card configuration and volume settings is a cause for concern. With those two concerns addressed, the module will be ready for manufacture and distribution.

Entirely new modules can also be created. Several professors have requested strain gauge, haptic knob, thermal, and fluids modules. The model set forth by the
Distributed Laboratory System can also be applied to other classes at the University, both in Mechanical Engineering and other departments.

Also, it was found that many professors were eager to have classroom demonstrations of the modules, even when the modules were not finished, and had known bugs. This excitement should not be discarded, and the development of modules solely for demonstration purposes could be investigated further.

Finally, the existing FR/PID module can still be improved with the feedback and the input of students and staff as they use, enjoy, and learn from the module for years to come.
References

Appendix A: Photographs
Appendix A-1: FR/PID Module

Figure A.1: The final FR/PID Module
Figure A.2: The final FR/PID Module (Front View)
**Figure A.3:** The final FR/PID Module (Top View)

**Figure A.4:** The final FR/PID Module (Back View, with rubber band (PID Mode))
Figure A.5: The final FR/PID Module (Back View, without rubber band (PID Mode))

Figure A.6: The final FR/PID Module (Top View, with rubber band (PID Mode))
Appendix A-2: Generation One Modules

Figure A.7: The Mass Spring Damper Module

Figure A.8: The Mass Spring Damper Module Mass Production
Appendix A-2: Generation One Modules

**Figure A.9:** The Audio Filtering Module

**Figure A.10:** The Audio Filtering Module Mass Production
Appendix B: Technical Documents
## BILL OF MATERIALS --FR/PID

<table>
<thead>
<tr>
<th>Ref</th>
<th>Part</th>
<th>Vendor</th>
<th>Vendor PN</th>
<th>Cost (1)</th>
<th>Cost @ QTY 100</th>
<th>QTY Needed per Board</th>
<th>QTY In Pack</th>
<th>Cost per Board @ QTY 100</th>
<th>Order QTY</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>U1</td>
<td>PIC18F2455 Flash MCU's SPDIP-28 24kB 20KRM FSUSB2</td>
<td>Mouser</td>
<td>PIC18F2455-455-I/SP</td>
<td>$5.55</td>
<td>$5.16</td>
<td>1 1</td>
<td>$5.16</td>
<td>110</td>
<td>$567.60</td>
<td></td>
</tr>
<tr>
<td>U2</td>
<td>Microchip Amplifiers - Operational PDIP-14 Quad 1.8V 1MHz MCP6004</td>
<td>Mouser</td>
<td>579-MCP600-4-E/P</td>
<td>$0.46</td>
<td>$0.42</td>
<td>1 1</td>
<td>$0.42</td>
<td>110</td>
<td>$46.20</td>
<td></td>
</tr>
<tr>
<td>U3</td>
<td>L293D H-Bridge</td>
<td>Digi-Key</td>
<td>296-9518-5-ND</td>
<td>$2.13</td>
<td>$1.28</td>
<td>1 1</td>
<td>$1.28</td>
<td>110</td>
<td>$140.25</td>
<td></td>
</tr>
<tr>
<td>C2, C6, C9</td>
<td>Capacitor, monolithic ceramic, 0.1uF, 50V, 20%</td>
<td>Jameco</td>
<td>25523</td>
<td>$0.07</td>
<td>$0.06</td>
<td>6 1</td>
<td>$0.06</td>
<td>660</td>
<td>$39.60</td>
<td></td>
</tr>
<tr>
<td>C1,7</td>
<td>Capacitor, radial electrolytic, 100 uF, 50V</td>
<td>Jameco</td>
<td>29962</td>
<td>$0.07</td>
<td>$0.06</td>
<td>2 1</td>
<td>$0.06</td>
<td>220</td>
<td>$13.20</td>
<td></td>
</tr>
<tr>
<td>C8</td>
<td>CAP MONO .22 uF .50V 20%</td>
<td>Jameco</td>
<td>25540</td>
<td>$0.17</td>
<td>$0.12</td>
<td>1 1</td>
<td>$0.12</td>
<td>110</td>
<td>$13.20</td>
<td></td>
</tr>
<tr>
<td>D1</td>
<td>LED HI-EFF GREEN 565 NM, T-1</td>
<td>Jameco</td>
<td>333201C-J</td>
<td>$0.08</td>
<td>$0.06</td>
<td>1 1</td>
<td>$0.06</td>
<td>110</td>
<td>$6.93</td>
<td></td>
</tr>
<tr>
<td>R1</td>
<td>47K 1/4 Carbon Film 5%</td>
<td>Mouser</td>
<td>660-CF1/4L73J</td>
<td>$0.05</td>
<td>$0.03</td>
<td>1 1</td>
<td>$0.03</td>
<td>110</td>
<td>$3.30</td>
<td></td>
</tr>
<tr>
<td>R2</td>
<td>330 1/4 Carbon Film 5%</td>
<td>Mouser</td>
<td>660-CF1/4L31J</td>
<td>$0.05</td>
<td>$0.03</td>
<td>1 1</td>
<td>$0.03</td>
<td>110</td>
<td>$3.30</td>
<td></td>
</tr>
<tr>
<td>R3</td>
<td>Potentiometer 7/8&quot; 1KOhm Vishay 157-11102</td>
<td>Newark</td>
<td>98F7374</td>
<td>$16.16</td>
<td>$11.19</td>
<td>1 1</td>
<td>$11.19</td>
<td>100</td>
<td>$1,119.00</td>
<td></td>
</tr>
<tr>
<td>R4, R6, R7</td>
<td>10K 1/4 Carbon Film 5%</td>
<td>Mouser</td>
<td>660-CF1/4L103J</td>
<td>$0.05</td>
<td>$0.03</td>
<td>3 1</td>
<td>$0.09</td>
<td>330</td>
<td>$9.90</td>
<td></td>
</tr>
<tr>
<td>R5</td>
<td>RESISTOR 1/4 W 5%, 33K OHM</td>
<td>Mouser</td>
<td>660-CF1/4L33J</td>
<td>$0.05</td>
<td>$0.03</td>
<td>1 1</td>
<td>$0.03</td>
<td>110</td>
<td>$3.30</td>
<td></td>
</tr>
<tr>
<td>X1</td>
<td>RESONATOR 20.00MHZ CERAMIC W/CAP</td>
<td>Digi-Key</td>
<td>X909-ND</td>
<td>$0.54</td>
<td>$0.36</td>
<td>1 1</td>
<td>$0.36</td>
<td>110</td>
<td>$39.60</td>
<td></td>
</tr>
<tr>
<td>Item</td>
<td>Description</td>
<td>Part Number</td>
<td>Supplier</td>
<td>Price</td>
<td>Quantity</td>
<td>Total</td>
<td>Per Unit</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
<td>-------------</td>
<td>----------</td>
<td>--------</td>
<td>----------</td>
<td>-------</td>
<td>----------</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>J1</td>
<td>ICP Machine Contact Single-In-Line Female Sockets</td>
<td>78641C H</td>
<td>Jameco</td>
<td>$0.89</td>
<td>18</td>
<td>$14.40</td>
<td>$0.80</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>J2</td>
<td>DC Power jack, 2.1mm, male, pc board mount, Switchcraft RAPC722</td>
<td>SC1153-ND</td>
<td>Digi-Key</td>
<td>$1.16</td>
<td>1</td>
<td>$0.73</td>
<td>$0.73</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>J3</td>
<td>Connector - USB Type B</td>
<td>787780-1-ND</td>
<td>Digi-Key</td>
<td>$1.15</td>
<td>1</td>
<td>$0.76</td>
<td>$0.76</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>J4-0, J4-1</td>
<td>3 Holes on PCB board for Potentiometer wires</td>
<td>NA NA</td>
<td>NA</td>
<td>$0.00</td>
<td>1</td>
<td>$0.00</td>
<td>$0.00</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>J5-0</td>
<td>3 Holes on PCB board for future expansion</td>
<td>NA NA</td>
<td>NA</td>
<td>$0.00</td>
<td>1</td>
<td>$0.00</td>
<td>$0.00</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>J6-0, J6-1</td>
<td>2 Holes on PCB board for motor wires</td>
<td>NA NA</td>
<td>NA</td>
<td>$0.00</td>
<td>1</td>
<td>$0.00</td>
<td>$0.00</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>J7-0</td>
<td>4 Holes on PCB board for serial port expansion</td>
<td>NA NA</td>
<td>NA</td>
<td>$0.00</td>
<td>1</td>
<td>$0.00</td>
<td>$0.00</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>H1</td>
<td>Motor:DC,24V/0.9A,2680RPM,178GM-CM,0.09&quot;SHFT DIA</td>
<td>237382C L</td>
<td>Jameco</td>
<td>$7.19</td>
<td>110</td>
<td>$640.20</td>
<td>$5.82</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>H2</td>
<td>8-32 Bolts: MACHINE PHILLIPS 8-32X1/2</td>
<td>H384-ND</td>
<td>Digi-Key</td>
<td>$2.85</td>
<td>1</td>
<td>$2.85</td>
<td>$0.14</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>H3</td>
<td>Gears: Molded Nylon 14-1/2 Deg Angle Spur Gear, 48 Pitch, 96 Teeth, 2&quot; Pitch Diameter, 1/4&quot; Bore</td>
<td>57655K3 T</td>
<td>McMaster</td>
<td>$4.31</td>
<td>220</td>
<td>$948.20</td>
<td>$8.62</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>H4</td>
<td>Inertia: Zinc-Plated Steel Set Screw Shaft Collar 1&quot; Bore, 1-1/2&quot; Outside Diameter, 5/8&quot; Width</td>
<td>6432K25</td>
<td>McMaster</td>
<td>$1.61</td>
<td>120</td>
<td>$193.20</td>
<td>$1.61</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>H5</td>
<td>Bushing: Black Delrin Rod 1/4&quot; Diameter</td>
<td>8576K11</td>
<td>McMaster</td>
<td>$2.05</td>
<td>1</td>
<td>$2.05</td>
<td>$0.03</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>H6</td>
<td>Rubber Band: Natural, 2&quot; Lg X 1/16&quot; WD, Size 14</td>
<td>12205T5 2</td>
<td>McMaster</td>
<td>$4.77</td>
<td>338</td>
<td>$160.84</td>
<td>$4.27</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>H7</td>
<td>Legs: Aluminum Threaded 8-32 Round Standoffs 1/4&quot; OD, 2&quot; Length</td>
<td>534-3498</td>
<td>Mouser</td>
<td>$0.69</td>
<td>4</td>
<td>$2.40</td>
<td>$0.60</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>H8</td>
<td>Spring Pole: Aluminum 8-32 Threaded Round Standoffs 1/4&quot; OD, 1.5&quot; Length</td>
<td>534-3497</td>
<td>Mouser</td>
<td>$0.69</td>
<td>1</td>
<td>$0.60</td>
<td>$0.60</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Item</td>
<td>Description</td>
<td>Vendor</td>
<td>SKU</td>
<td>Price</td>
<td>Quantity</td>
<td>Unit Price</td>
<td>Total</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
<td>--------</td>
<td>-----</td>
<td>-------</td>
<td>----------</td>
<td>------------</td>
<td>-------</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>H10</td>
<td>Suction Cups: Rubber With Threaded Post, 1&quot;DIA, 5/16&quot;H, 8-32 Machine Screw, 1/4&quot;SCREW Proj</td>
<td>McMaster</td>
<td>53535A1</td>
<td>$6.09</td>
<td>4</td>
<td>$6.09</td>
<td>$2.44</td>
<td>42</td>
<td>$255.78</td>
<td></td>
</tr>
<tr>
<td>H11</td>
<td>Magnet: Flexible Strip With Adhesive Back 1/16&quot; Thick, 2&quot; Width</td>
<td>McMaster</td>
<td>5759K82</td>
<td>$1.64</td>
<td>1</td>
<td>$1.64</td>
<td>$0.27</td>
<td>20</td>
<td>$32.80</td>
<td></td>
</tr>
<tr>
<td>H12</td>
<td>Motor Bolts 18-8 SS Metric Pan Head Phil Machine Screw M3 Size, 6mm Length, .5mm Pitch</td>
<td>McMaster</td>
<td>92000A1</td>
<td>$6.17</td>
<td>2</td>
<td>$6.17</td>
<td>$0.12</td>
<td>2</td>
<td>$12.34</td>
<td></td>
</tr>
<tr>
<td>H13</td>
<td>Pot Wires: Jumper Wire 22 AWG solid, tinned copper, PVC insulated Stripped 1/4&quot;, bent 90° on both ends</td>
<td>Mouser</td>
<td>517-923345-20</td>
<td>$8.44</td>
<td>3</td>
<td>$8.44</td>
<td>$0.25</td>
<td>4</td>
<td>$33.76</td>
<td></td>
</tr>
<tr>
<td>H14</td>
<td>8-32 Nuts: Zinc-Plated Steel Machine, 11/32&quot; Width, 1/8&quot; Height</td>
<td>McMaster</td>
<td>90480A0</td>
<td>$1.07</td>
<td>4</td>
<td>$1.07</td>
<td>$0.04</td>
<td>4</td>
<td>$4.28</td>
<td></td>
</tr>
<tr>
<td>E1</td>
<td>Power Supply TRANS, WALL REG, 12V DC/500mA, F2 2.1mmX5.5mm</td>
<td>Jameco</td>
<td>162996</td>
<td>$9.95</td>
<td>1</td>
<td>$9.95</td>
<td>$7.25</td>
<td>1</td>
<td>$7.25</td>
<td>105</td>
</tr>
<tr>
<td>E2</td>
<td>Circuit Board PCB Board</td>
<td>Express PCB</td>
<td>NA</td>
<td>$59.00</td>
<td>1</td>
<td>$5.03</td>
<td>$5.03</td>
<td>1</td>
<td>$5.03</td>
<td>110</td>
</tr>
<tr>
<td>E3</td>
<td>USB 2.0 Cable, A Male To B Male, Black, 6ft</td>
<td>CyberGuys</td>
<td>1310980</td>
<td>$2.19</td>
<td>1</td>
<td>$1.99</td>
<td>$1.99</td>
<td>1</td>
<td>$1.99</td>
<td>110</td>
</tr>
<tr>
<td>E4</td>
<td>Box 5 x 5 x 4&quot;, Corrugated Mailers</td>
<td>Packaging Price</td>
<td>M554</td>
<td>$18.50</td>
<td>1</td>
<td>$18.50</td>
<td>$0.37</td>
<td>2</td>
<td>$7.74</td>
<td></td>
</tr>
<tr>
<td>E5</td>
<td>Box Label 3 1/3 x 4&quot;</td>
<td>Avery</td>
<td>5160</td>
<td>$0.00</td>
<td>1</td>
<td>$0.00</td>
<td>$0.00</td>
<td>1</td>
<td>$0.00</td>
<td>$0.00</td>
</tr>
<tr>
<td>E6</td>
<td>Module Label 1x 2 5/8&quot;</td>
<td>Avery</td>
<td>5164</td>
<td>$0.00</td>
<td>1</td>
<td>$0.00</td>
<td>$0.00</td>
<td>1</td>
<td>$0.00</td>
<td>$0.00</td>
</tr>
<tr>
<td>E7</td>
<td>Wire Ties, 5.75&quot; long, 1.3&quot; Max Diameter</td>
<td>Digi-Key</td>
<td>RP213C-ND</td>
<td>$0.04</td>
<td>1</td>
<td>$3.87</td>
<td>$0.04</td>
<td>2</td>
<td>$7.74</td>
<td></td>
</tr>
<tr>
<td>E8</td>
<td>Adhesive: Permabond Standard 910 FS, 1 oz Bottle, 5 Sec Set Time</td>
<td>McMaster</td>
<td>7580A65</td>
<td>$14.76</td>
<td>0</td>
<td>$14.76</td>
<td>$0.00</td>
<td>1</td>
<td>$14.76</td>
<td></td>
</tr>
</tbody>
</table>

**Totals:** $180.64 | $118.03 | $57.90 | $6,205.63
Appendix B-2: Express SSH (Electrical Schematic)
Appendix B-3: Express PCB (Circuit Board)
Appendix B-4: Matlab

This is the Matlab file used in Engineering Test 8 to simulate a PID system a student might find in a textbook or homework problem.

```matlab
function pidcon (kp,ki,kd)
% P-I-D controller
clc %Clears the command window

%You can comment these to set their values at runtime
kp=500; % Small Value=70 Medium Value=500 Large Value=2000
ki=0; % Small Value=10 Medium Value=50 Large Value=2000
kd=400; % Small Value=50 Medium Value=100 Large Value=400

Gp=tf(1,[20 40 1]) % Plant
Gc=tf([kd kp ki],[1 0]) %Controller
Gcl=feedback(Gc*Gp,1) % Closed loop transfer function

poles=pole(Gcl) %Prints out the poles
zeros=zero(Gcl) %Prints out the zeros
dcgain=dcgain(Gcl) %Prints out the DC gain

figure(1) %Sets focus to figure 1
clf %Clears the figure
hold on
AXIS([0 10 0 80]) %Keeps the plot size constant
step(45*Gcl) %Plots the step
plot([0 100],[45 45],'r') %Plots the input step
hold off
```

Appendix B-5: Pro-E
This Pro-E model was created for one of the later prototypes to check for tolerance issues. The files are included on the companion CD.

Appendix B-6: Data Tables and Sheets
This test was given to ME 3281 students during Summer Semester 2004.

Students were given the same test three times: before anyone had done the lab, when half the class had done the lab, and when everyone had completed the lab. The Pre, Mid, and Post tests all asked the same questions. The secret code was generated by the student, and they used that code on all three tests, allowing their scores to be tracked without knowing their name.

Post Test 1 for ME3281

*Secret Code Name ____________*

*I Have Done the Lab (Yes/No) ____________*

**Purpose:**
These Pre/Mid/Post tests will be used to help gauge the effectiveness of the Distributed Laboratory program. These test scores will not count towards your overall grade and individual students cannot be identified by their randomly chosen secret code name. Thanks for your cooperation!

- Please complete each problem to the best of your ability. Please use complete sentences.
- After you complete each problem, rank your confidence in your answer using the following scale.

<table>
<thead>
<tr>
<th>5 = Very Confident</th>
<th>4 = Confident</th>
<th>3 = Neutral</th>
<th>2 = Unconfident</th>
<th>1 = Very Unconfident</th>
</tr>
</thead>
</table>

**Questions:**
1. What is a Bode plot and what is it used for?

   Confidence Ranking =

2. What are the differences between high pass and low pass filters

   Confidence Ranking =

(continued on next page)

Appendix B-6: Data Tables and Sheets
3. If a low pass filter has a cut-off frequency of 1000 Hz, what effect would this filter have on the following frequencies?
   a. 10 Hz
   b. 1000 Hz
   c. 10000 Hz

Confidence Ranking =
Appendix B-6: Data Tables and Sheets

Data calculated for $\Theta$ and corresponding torque using the equations of Figure 4.12 and plotted in Figure 4.13

<table>
<thead>
<tr>
<th>$\Theta$ (deg)</th>
<th>T (lb-in)</th>
<th>15</th>
<th>0.01309</th>
<th>-17</th>
<th>-0.01484</th>
</tr>
</thead>
<tbody>
<tr>
<td>45</td>
<td>0.03927</td>
<td>14</td>
<td>0.012217</td>
<td>-18</td>
<td>-0.01571</td>
</tr>
<tr>
<td>44</td>
<td>0.038397</td>
<td>13</td>
<td>0.011345</td>
<td>-19</td>
<td>-0.01658</td>
</tr>
<tr>
<td>43</td>
<td>0.037525</td>
<td>12</td>
<td>0.010472</td>
<td>-20</td>
<td>-0.01745</td>
</tr>
<tr>
<td>42</td>
<td>0.036652</td>
<td>11</td>
<td>0.009599</td>
<td>-21</td>
<td>-0.01833</td>
</tr>
<tr>
<td>41</td>
<td>0.035779</td>
<td>10</td>
<td>0.008727</td>
<td>-22</td>
<td>-0.0192</td>
</tr>
<tr>
<td>40</td>
<td>0.034907</td>
<td>9</td>
<td>0.007854</td>
<td>-23</td>
<td>-0.02007</td>
</tr>
<tr>
<td>39</td>
<td>0.034034</td>
<td>8</td>
<td>0.006981</td>
<td>-24</td>
<td>-0.02094</td>
</tr>
<tr>
<td>38</td>
<td>0.033161</td>
<td>7</td>
<td>0.006109</td>
<td>-25</td>
<td>-0.02182</td>
</tr>
<tr>
<td>37</td>
<td>0.032289</td>
<td>6</td>
<td>0.005236</td>
<td>-26</td>
<td>-0.02269</td>
</tr>
<tr>
<td>36</td>
<td>0.031416</td>
<td>5</td>
<td>0.004363</td>
<td>-27</td>
<td>-0.02356</td>
</tr>
<tr>
<td>35</td>
<td>0.030543</td>
<td>4</td>
<td>0.003491</td>
<td>-28</td>
<td>-0.02443</td>
</tr>
<tr>
<td>34</td>
<td>0.029671</td>
<td>3</td>
<td>0.002618</td>
<td>-29</td>
<td>-0.02531</td>
</tr>
<tr>
<td>33</td>
<td>0.028798</td>
<td>2</td>
<td>0.001745</td>
<td>-30</td>
<td>-0.02618</td>
</tr>
<tr>
<td>32</td>
<td>0.027925</td>
<td>1</td>
<td>0.000873</td>
<td>-31</td>
<td>-0.02705</td>
</tr>
<tr>
<td>31</td>
<td>0.027053</td>
<td>0</td>
<td>0</td>
<td>-32</td>
<td>-0.02793</td>
</tr>
<tr>
<td>30</td>
<td>0.02618</td>
<td>-1</td>
<td>-0.00087</td>
<td>-33</td>
<td>-0.0288</td>
</tr>
<tr>
<td>29</td>
<td>0.025307</td>
<td>-2</td>
<td>-0.00175</td>
<td>-34</td>
<td>-0.02967</td>
</tr>
<tr>
<td>28</td>
<td>0.024435</td>
<td>-3</td>
<td>-0.00262</td>
<td>-35</td>
<td>-0.03054</td>
</tr>
<tr>
<td>27</td>
<td>0.023562</td>
<td>-4</td>
<td>-0.00349</td>
<td>-36</td>
<td>-0.03142</td>
</tr>
<tr>
<td>26</td>
<td>0.022689</td>
<td>-5</td>
<td>-0.00436</td>
<td>-37</td>
<td>-0.03229</td>
</tr>
<tr>
<td>25</td>
<td>0.021817</td>
<td>-6</td>
<td>-0.00524</td>
<td>-38</td>
<td>-0.03316</td>
</tr>
<tr>
<td>24</td>
<td>0.020944</td>
<td>-7</td>
<td>-0.00611</td>
<td>-39</td>
<td>-0.03403</td>
</tr>
<tr>
<td>23</td>
<td>0.020071</td>
<td>-8</td>
<td>-0.00698</td>
<td>-40</td>
<td>-0.03491</td>
</tr>
<tr>
<td>22</td>
<td>0.019199</td>
<td>-9</td>
<td>-0.00785</td>
<td>-41</td>
<td>-0.03578</td>
</tr>
<tr>
<td>21</td>
<td>0.018326</td>
<td>-10</td>
<td>-0.00873</td>
<td>-42</td>
<td>-0.03665</td>
</tr>
<tr>
<td>20</td>
<td>0.017453</td>
<td>-11</td>
<td>-0.0096</td>
<td>-43</td>
<td>-0.03752</td>
</tr>
<tr>
<td>19</td>
<td>0.016581</td>
<td>-12</td>
<td>-0.01047</td>
<td>-44</td>
<td>-0.0384</td>
</tr>
<tr>
<td>18</td>
<td>0.015708</td>
<td>-13</td>
<td>-0.01134</td>
<td>-45</td>
<td>-0.03927</td>
</tr>
<tr>
<td>17</td>
<td>0.014835</td>
<td>-14</td>
<td>-0.01222</td>
<td></td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>0.013963</td>
<td>-15</td>
<td>-0.01309</td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>0.01309</td>
<td>-16</td>
<td>-0.01396</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## Appendix B-6: Data Tables and Sheets

<table>
<thead>
<tr>
<th>Hz</th>
<th>Deg</th>
<th>Deg</th>
<th>Deg</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.5</td>
<td>54</td>
<td>54.1</td>
<td>56</td>
</tr>
<tr>
<td>3.5</td>
<td>59</td>
<td>59.7</td>
<td>59.7</td>
</tr>
<tr>
<td>4</td>
<td>62</td>
<td>61.5</td>
<td>61.5</td>
</tr>
<tr>
<td>4.5</td>
<td>61.5</td>
<td>58.9</td>
<td>58.5</td>
</tr>
<tr>
<td>5</td>
<td>52</td>
<td>50.9</td>
<td>51.5</td>
</tr>
<tr>
<td>6</td>
<td>36</td>
<td>36.6</td>
<td>32.9</td>
</tr>
<tr>
<td>7.5</td>
<td>22</td>
<td>22.7</td>
<td>20.5</td>
</tr>
<tr>
<td>10</td>
<td>11.6</td>
<td>11.1</td>
<td>9.8</td>
</tr>
<tr>
<td>15</td>
<td>4.5</td>
<td>4.5</td>
<td>5.2</td>
</tr>
<tr>
<td>20</td>
<td>2.4</td>
<td>2.3</td>
<td>3</td>
</tr>
</tbody>
</table>

Evaluation Test Initial Data:
- Taken before any tests for comparison purposes

<table>
<thead>
<tr>
<th>Hz</th>
<th>Deg</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.5</td>
<td>53.4</td>
</tr>
<tr>
<td>3.5</td>
<td>60</td>
</tr>
<tr>
<td>4</td>
<td>62.2</td>
</tr>
<tr>
<td>4.5</td>
<td>60.7</td>
</tr>
<tr>
<td>5</td>
<td>52.7</td>
</tr>
<tr>
<td>6</td>
<td>33.7</td>
</tr>
<tr>
<td>7.5</td>
<td>20.5</td>
</tr>
<tr>
<td>10</td>
<td>10.3</td>
</tr>
<tr>
<td>15</td>
<td>4.5</td>
</tr>
<tr>
<td>20</td>
<td>2.3</td>
</tr>
</tbody>
</table>

Evaluation Test 1 Data: Taken after running for 1 hour

<table>
<thead>
<tr>
<th>Hz</th>
<th>Deg</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.5</td>
<td>55.6</td>
</tr>
<tr>
<td>3.5</td>
<td>60.3</td>
</tr>
<tr>
<td>4</td>
<td>63</td>
</tr>
<tr>
<td>4.5</td>
<td>57.8</td>
</tr>
<tr>
<td>5</td>
<td>49.9</td>
</tr>
<tr>
<td>6</td>
<td>35.9</td>
</tr>
<tr>
<td>7.5</td>
<td>22</td>
</tr>
<tr>
<td>10</td>
<td>11.6</td>
</tr>
<tr>
<td>15</td>
<td>5.1</td>
</tr>
<tr>
<td>20</td>
<td>3</td>
</tr>
</tbody>
</table>

Evaluation Test 2 Data: Taken after running for 12 hours

<table>
<thead>
<tr>
<th>Hz</th>
<th>Deg</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.5</td>
<td>55.1</td>
</tr>
<tr>
<td>3.5</td>
<td>59.2</td>
</tr>
<tr>
<td>4</td>
<td>60.9</td>
</tr>
<tr>
<td>4.5</td>
<td>58.4</td>
</tr>
<tr>
<td>5</td>
<td>51.1</td>
</tr>
<tr>
<td>6</td>
<td>36.8</td>
</tr>
<tr>
<td>7.5</td>
<td>22.1</td>
</tr>
<tr>
<td>10</td>
<td>11.7</td>
</tr>
<tr>
<td>15</td>
<td>5.2</td>
</tr>
<tr>
<td>20</td>
<td>2.3</td>
</tr>
</tbody>
</table>

Evaluation Test 5 Data: Taken after backpack trip
Appendix B-7: Code

PICchip firmware for Gen 2 FR/PID

/******************************************************************************
FILE: dlab.c
VERSION: 2.20
DESCRIPTION: PICchip firmware for UMN DLAB project. Runs
frequency response (FR) experiment and PID control experiment.
Connects to host by USB. Uses UMN PIC18 USB driver code.
See companion readme.txt file for additional details
The reference "readme xx" means that Note xx in readme.txt
has an expanded comment.
Code developed by Waletzko, modified by Durfee and Waletzko
*******************************************************************************/

//--Includes
#include <18f2455.h>
//--Setup commands
#define LED_PIN B5
#define IEMAX 150 // integral windup saturation
#define UMAX 1023 // max pwm cmd 1023
#define USB_SPEED LOW_SPEED // low speed USB (readme 03)
#include "usb_driver_pic18.h"
#include "dlab_descriptors.h"
#include "usb_driver_pic18.c"
//--Other includes
//--Definitions
#include <18f2455.h>
#include <18f2455.h>
#include "dlab_descriptors.h"
#include "usb_driver_pic18.c"
//--Register bits and bytes
#include <18f2455.h>
#include <18f2455.h>
#include "dlab_descriptors.h"
#include "usb_driver_pic18.c"
//--Global variables
#include <18f2455.h>
#include <18f2455.h>
#include "dlab_descriptors.h"
#include "usb_driver_pic18.c"
//--Function declarations
#include <18f2455.h>
#include <18f2455.h>
#include "dlab_descriptors.h"
#include "usb_driver_pic18.c"

---Main

void main(void)
{
    unsigned char i;
    //----Initialize hardware
    set_tris_a(0xFF); // a is all inputs for adc (readme 05)
    set_tris_b(0x00); // b is all outputs

    //----Initialize hardware
    set_tris_a(0xFF); // a is all inputs for adc (readme 05)
    set_tris_b(0x00); // b is all outputs
```c
set_tris_c(0x80); // c7 in for serial, rest out
setup_timer_0(RTCC_8_BIT|RTCC_INTERNAL|RTCC_DIV_1); // overflows every .043 ms
setup_adc(ADC_CLOCK_DIV_32); // (readme 06)
setup_adc_ports(AN0|VREF_VREF); // RA0 with RA2/RA3 as low/high refs (readme 07)
setup_ccp1(CCP_PWM); // configure as pwm (pin rc2)
setup_ccp2(CCP_PWM); // configure as pwm (pin rc1)
setup_timer_2(T2_DIV_BY_4, 255, 1); // pwm period = 170 usec (readme 08)

//----Initialize variables
for (i=0; i<8; i++) {
    in_buffer[i]=0;
    out_buffer[i]=0;
}
pidflag=0; frflag=0; incflag=0; ninc=0; scale=0;
r=436; y=0; samp=0; nsamp=0;
e=0; ie=0; eml=0; ieml=0; u=0; wave=4;
K1=0; K2=0; K3=0;

//----Initialize USB then wait for enumeration (readme 16)
InitUSB();
while (!ConfiguredUSB()) { ServiceUSB(); // poll USB
    motor_off(); // turn motor off
}
//----Do task control forever
while (TRUE) {

    **************************************
    Task control
    Wait for next sample period, sample sensor, send data to host, query host for data, switch on host command, run current task (idle, pid, or fr)
    **************************************

    while (!T0IF); // wait for timer0 overflow (readme 11) (the ; immediately after the while will hold the program here)
    T0IF=0;
    output_low(LED); // flash led pin for viewing w/ scope
    output_high(LED); // and leave it on so you know the module has power
    ServiceUSB(); // regular polling of USB required
    y=read_adc(); // get pot position
    in_buffer[0]=0; // load up data frame for host (readme 12)
    in_buffer[1]=(unsigned char)(r&0xFF); // low byte of reference
    in_buffer[2]=(unsigned char)((r&0xFF00)>>8); // high byte
    in_buffer[3]=(unsigned char)(y&0xFF); // low byte of position
    in_buffer[4]=(unsigned char)((y&0xFF00)>>8); // high byte
    if (ConfiguredUSB()) { // only access EP1 if configured
        PutEP1(8, in_buffer); // send data to host (readme 13)
        if (GetEP1(out_buffer)==8) { // get new data from host, if any
            switch (out_buffer[0]) { // switch on mode command (byte 0)
                case 0: // idle mode
                    pidflag=0; // stop pid/fr if running
                    frflag=0;
                    motor_off(); // turn motor off
                    break;
                case 1: // pid mode
                    pidflag=1;
                    frflag=0;
                    break;
                case 2: // freq response mode (readme 14)
                    frflag=1;
                    pidflag=0;
                    break;
                case 3: // set pid gains
                    K1=out_buffer[1]+(((int16)out_buffer[2])<<8);
                    K2=out_buffer[3]+(((int16)out_buffer[4])<<8);
                    K3=out_buffer[5]+(((int16)out_buffer[6])<<8);
                    break;
                case 4: // set pid ref position
                    r=out_buffer[1]+(((int16)out_buffer[2])<<8);
                    break;
                case 5: // set freq response param (readme 17)
                    nsamp=out_buffer[1]+(((int16)out_buffer[2])<<8);
            }
        }
    }

Appendix B-7: Code
```
scale=out_buffer[3];
ninc=nsamp>>4;  // divide by 16
break;
case 20:            // flash LED (readme 18)
  for (i=0;i<out_buffer[1];i++){
    output_low(LED);
    delay_ms(100);
    output_high(LED);
    delay_ms(300);
  }
  out_buffer[0]=0;  // to prevent repeated flashing
  pidflag=0;      // stop pid/fr if running
  frflag=0;
  r=436;          // reset ref position (436=90)
  motor_off();    // turn motor off
  default:
    break;
  }
} // end of switch
} // end of GetEP1 test
if (pidflag) // do pid algorithm once
  pid();
if (frflag)  // do freq resp algorithm once
  fr();
}
} // end of task control, go back and do again
} // end of main

/*********************************************************
id()
Compute pid control law, once (readme 15)
*********************************************************/
void pid(void)
{
  //Count causes the PID routine to only be run about every 13 ms
  // giving the inertia time to move, thus allowing for error-previous_error (e-em1)
  // to be non-zero, and preventing the integral control from winding up as fast (ie=ie+e)
  if (count++>320)
  {
    count=0;
    e=y-r;      // error
    ie=ie+e;     // integral error term
    if (ie>IEMAX) ie=IEMAX;  // cap to prevent integrator wind-up
    if (ie<-IEMAX) ie=-IEMAX;
    u=K1*e-K2*em1+K3*ie;    // control law
    em1=e;                  // update past values
    drive_motor();         // update motor drive
  }
}

/*********************************************************
fr()
Compute freq resp command, once
*********************************************************/
void fr(void)
{
  if (++samp>=nsamp) // if at peak, toggle inc/dec flag
    samp=0;
  if (incflag) incflag=0;
  else incflag=1;
  if (++inc>=ninc) // if time to inc/dec, inc/dec wave
    inc=0;
  if (incflag) wave++;
  else wave--;
  if (wave>8) wave=8;    // cap to +/- 8
  if (wave<8) wave=-8;
  u=wave<<scale; // form motor command
  drive_motor(); // command motor
}

/*********************************************************
drive_motor()
Update motor pwm driver with current motor command u
*********************************************************/
void drive_motor()
{
if (u>UMAX) u=UMAX;     // cap output
if (u<-UMAX) u=-UMAX;

if (u<0) {              // spin one way
    set_pwm1_duty(-u);
    set_pwm2_duty(0);
} else {                 // the other way for non-negative u
    set_pwm1_duty(0);
    set_pwm2_duty(u);
}

/*********************************************************
motor_off()
Turns motor off with braking (see L293D data sheet)
*********************************************************/
void motor_off(void)
{
    set_pwm1_duty(0);
    set_pwm2_duty(0);
}

Appendix B-7: Code

Visual Basic DLABForm.frm software

Private Sub BQuit_Click()
'quit
    'Idle mode
    OutputReportData(0) = 0
    WriteReport
    'and quick
End
End Sub
Private Sub Form_Load()
Dim i As Integer
Dim sensor As Single
Dim result As Boolean
Dim r(2) As Double          'data array for graphs
Dim p As Integer

'--Initialize variables
For i = 0 To 7
    OutputReportData(i) = 0
    InputReportData(i) = 0
Next

'--See if Dlab usb device is attached
If (FindTheHid) Then  'success
    Else
        MsgBox ("Could not find Dlab hardware. Check connections.")
End
End If
Me.Show

    'Idle mode (Make sure nothing is running on the pic)
    OutputReportData(0) = 0
    WriteReport
    FRForm.BodePlot.AxisTitleFontSize(X_AXIS1) = 12
    FRForm.BodePlot.AxisTitleColor(X_AXIS1) = black
    Dim SysTime As SYSTEMTIME
    '--Infinite loop reading data and displaying device sensor reading.
    While (True)
        DoEvents
If (ReadReport) Then 'get data from device
    sensor = CSng((InputReportData(3) + InputReportData(4) * 256)) / 4.84 '4.84 is a ROUGH Conversion factor
    picinput = (InputReportData(1) + InputReportData(2) * 256) / 4.84
        If PIDForm.Visible = True Then
            PIDForm.InputCurrent.Caption = Round(picinput, 0)
            r(0) = picinput
            r(1) = sensor
            If (PIDForm.BGraph.Caption = "Freeze Plot") Then
                PIDForm.TOutput.Caption = Round(sensor, 1)
                Call PIDForm.PIDGraph.UpdateDynData(0, r(0))
                Call PIDForm.PIDGraph.UpdateDynData(1, r(1))
        End If
    End If
End If

If FRForm.Visible = True Then
    r(1) = sensor
        If (FRForm.BGraph.Caption = "Freeze Plot") Then
            Call FRForm.MSDGraph.UpdateDynData(1, r(1))
        End If
End If

Appendix B-7: Code

    If FRForm.Zoom.Caption = "UnZoom" Then
        If zoompoint < 100 Then 'average 100 points of data
            zoomtotal = zoomtotal + sensor
            zoompoint = zoompoint + 1
        Else
            If zoomflag = True Then
                FRForm.MSDGraph.AxisMax(Y_AXIS1) = CInt(zoomtotal / 100 + 15)
                FRForm.MSDGraph.AxisMin(Y_AXIS1) = CInt(zoomtotal / 100 - 15)
                FRForm.MSDGraph.UpdateGraph
                zoomflag = False
                FRForm.Zoom.Enabled = True
                FRForm.Omag.Caption = Format((FRForm.MaxUnfiltered.Top - FRForm.MinUnfiltered.Top) / (4290 / 30), "#0.0")
            End If
            zoompoint = 0
        End If
    End If
Else
    MsgBox("Hardware disconnected, quitting application")
End If

If collectdata = True Then
    GetSystemTime SysTime
    'output to text file
    Open "c:\pid.csv" For Append As #1
    Write #1, r(0), r(1), SysTime.wSecond & "." & SysTime.wMilliseconds
    Close #1
End If
Wend
End Sub
Private Sub FR_Click()
    'Load FR VB Interface
    DLABForm.Hide
    FRForm.Show
    FRForm.freqsetScroll.SetFocus
End Sub
Private Sub PIDButton_Click()
    'Load PID VB Interface
    DLABForm.Hide
    PIDForm.Show
    PIDForm.InputScroll.SetFocus
End Sub

Visual Basic PIDForm.frm code

Private Sub BDefaultParameters_Click()
    'Reset the default values
    TP.Text = 7
    TD.Text = 7
    TI.Text = 1
    TInput.Text = 90
    Dim k1 As Integer, k2 As Integer, k3 As Integer
k1 = Val(TP.Text)
k2 = Val(TD.Text)
k3 = Val(TI.Text)

k1 = k1 + k2 'combined kp and kd
OutputReportData(0) = 3
OutputReportData(1) = (k1 And &HFF) 'low byte
OutputReportData(2) = (k1 And &HFF00) \ 256 'hi byte
OutputReportData(3) = (k2 And &HFF)
OutputReportData(4) = (k2 And &HFF00) \ 256

Appendix B-7: Code

OutputReportData(5) = (k3 And &HFF)
OutputReportData(6) = (k3 And &HFF00) \ 256
WriteReport

'update PID reference position
Dim r As Integer
r = Val(TInput.Text) * 4.84 '4.84 is a ROUGH Conversion factor
OutputReportData(0) = 4
OutputReportData(1) = (r And &HFF) 'low byte
OutputReportData(2) = (r And &HFF00) \ 256 'hi byte
WriteReport

'update the current control values as we just sent them
PCurrent.Caption = Val(TP.Text)
ICurrent.Caption = Val(TI.Text)
DCurrent.Caption = Val(TD.Text)
InputCurrent.Caption = Val(TInput.Text)
BUpdate.Enabled = False 'Disable the update button

'Sync scroll with new input value
InputScroll.Value = Val(TInput.Text)
InputScroll.SetFocus

Private Sub BGraph_Click()
'Toggle Graphing
If BGraph.Caption = "Freeze Plot" Then 'stop graphing
BGraph.Caption = "UnFreeze Plot"
Collectdatabutton.Caption = "Collect Data"
collectdata = False
Else 'start graphing
BGraph.Caption = "Freeze Plot"
End If
InputScroll.SetFocus
End Sub

Private Sub BStart_Click()
'Start or Stop PID Controller
If BStart.Caption = "Start Controller" Then

'update PID gains
'K1=Kp, K2=Kd, K3=Ki, assuming T=1.0 for convenience
Dim k1 As Double, k2 As Double, k3 As Double
k1 = Val(TP.Text)
k2 = Val(TD.Text)
k3 = Val(TI.Text)
flag = 1

'this clears out letters and characters and sets them to 0
TP.Text = k1
TD.Text = k2
TI.Text = k3

'check for negative gains which cannot be sent over USB
If k1 < 0 Then
flag = 0
End If
If k2 < 0 Then
flag = 0
End If
If k3 < 0 Then
flag = 0
End If

Appendix B-7: Code
'check for large gains which cannot be sent over USB
If k1 > 32767 Then
  flag = 0
End If
If k2 > 32767 Then
  flag = 0
End If
If k3 > 32767 Then
  flag = 0
End If
If flag = 1 Then 'all values are positive
  k1 = k1 + k2 'combined kp and kd
  OutputReportData(0) = 3
  OutputReportData(1) = (k1 And &HFF) 'low byte
  OutputReportData(2) = (k1 And &HFF00) \ 256 'hi byte
  OutputReportData(3) = (k2 And &HFF)
  OutputReportData(4) = (k2 And &HFF00) \ 256
  OutputReportData(5) = (k3 And &HFF)
  OutputReportData(6) = (k3 And &HFF00) \ 256
  WriteReport
  'update the current control values as we just sent them
  PCurrent.Caption = Val(TP.Text)
  ICurrent.Caption = Val(TI.Text)
  DCurrent.Caption = Val(TD.Text)
  'Start PID mode
  OutputReportData(0) = 1
  WriteReport
  BStart.Caption = "Stop Controller"
Else ' at least one gain is negative
  MsgBox ("Values for Kp, Ki and Kd must be between 0 and 32767")
End If
Else
  'Idle mode
  OutputReportData(0) = 0
  WriteReport
  BStart.Caption = "Start Controller"
End If
InputScroll.SetFocus
End Sub
Private Sub BStep_Click()
  If Val(InputCurrent.Caption) <= 45 Then
    TInput.Text = 90
  ElseIf Val(InputCurrent.Caption) >= 90 Then
    TInput.Text = 45
  Else
    TInput.Text = 90
  End If
  'update PID reference position
  Dim r As Integer
  r = Val(TInput.Text) * 4.84 '4.84 is a ROUGH Conversion factor
  OutputReportData(0) = 4
  OutputReportData(1) = (r And &HFF) 'low byte
  OutputReportData(2) = (r And &HFF00) \ 256 'hi byte
  WriteReport
Appendix B-7: Code

'update the current control values as we just sent them
InputCurrent.Caption = Val(TInput.Text)
InputScroll.Value = InputCurrent.Caption
InputScroll.SetFocus
End Sub
Private Sub BUpdate_Click()
  'Update Gains and Input Position
'update PID gains
'K1=Kp, K2=Kd, K3=Ki, assuming T=1.0 for convenience
'this also clears out letters and characters and sets them to 0
Dim k1 As Double, k2 As Double, k3 As Double
k1 = Val(TP.Text)
k2 = Val(TD.Text)
k3 = Val(TI.Text)
TP.Text = k1
TD.Text = k2
TI.Text = k3
TInput.Text = Val(TInput.Text)
flag = 1
inputflag = 1
If TInput.Text > 210 Then
inputflag = 0
End If
If TInput.Text < 1 Then
inputflag = 0
End If
If (inputflag) Then
'update PID reference position
Dim r As Integer
r = Val(TInput.Text) * 4.84 '4.84 is a ROUGH Conversion factor
OutputReportData(0) = 4
OutputReportData(1) = (r And &HFF) 'low byte
OutputReportData(2) = (r And &HFF00) \ 256 'hi byte
WriteReport
InputCurrent.Caption = Val(TInput.Text)
'Sync scroll with new input value
InputScroll.Value = Val(TInput.Text)
Else
MsgBox("Input must be between 1 and 210")
End If
'check for negative gains which cannot be sent over USB
If k1 < 0 Then
flag = 0
End If
If k2 < 0 Then
flag = 0
End If
If k3 < 0 Then
flag = 0
End If
'check for large gains which cannot be sent over USB
If k1 > 32767 Then
flag = 0
End If
If k2 > 32767 Then
flag = 0
End If
If (flag) Then 'all values are positive
k1 = k1 + k2 'combined kp and kd
OutputReportData(0) = 3
OutputReportData(1) = (k1 And &HFF) 'low byte
OutputReportData(2) = (k1 And &HFF00) \ 256 'hi byte
OutputReportData(3) = (k2 And &HFF)
OutputReportData(4) = (k2 And &HFF00) \ 256
OutputReportData(5) = (k3 And &HFF)
OutputReportData(6) = (k3 And &HFF00) \ 256
WriteReport
'update the current control values as we just sent them
PCurrent.Caption = Val(TP.Text)
TCurrent.Caption = Val(TI.Text)
DCurrent.Caption = Val(TD.Text)
Else ' at least one gain is negative

Appendix B-7: Code

flag = 0
End If
If k1 > 32767 Then
flag = 0
End If
If k2 > 32767 Then
flag = 0
End If
If flag = 1 Then 'all values are positive
k1 = k1 + k2 'combined kp and kd
OutputReportData(0) = 3
OutputReportData(1) = (k1 And &HFF) 'low byte
OutputReportData(2) = (k1 And &HFF00) \ 256 'hi byte
OutputReportData(3) = (k2 And &HFF)
OutputReportData(4) = (k2 And &HFF00) \ 256
OutputReportData(5) = (k3 And &HFF)
OutputReportData(6) = (k3 And &HFF00) \ 256
WriteReport
'update the current control values as we just sent them
PCurrent.Caption = Val(TP.Text)
TCurrent.Caption = Val(TI.Text)
DCurrent.Caption = Val(TD.Text)
Else ' at least one gain is negative

93
MsgBox ("Values for Kp, Ki and Kd must be between 0 and 32767")
End If

BUpdate.Enabled = False
InputScroll.SetFocus
End Sub

Private Sub Collectdatabutton_Click()
If Collectdatabutton.Caption = "Collect Data" Then
Collectdatabutton.Caption = "Stop Collecting"
collectdata = True
Else
Collectdatabutton.Caption = "Collect Data"
collectdata = False
End If

End Sub

Private Sub fPrint_Click()
    Set PrinterFormPicture1.Picture = CaptureActiveWindow()
    Me.SetFocus
    Call PrintPictureToFitPage(Printer, PrinterFormPicture1.Picture)
    Printer.EndDoc
    InputScroll.SetFocus
End Sub

Private Sub fQuit_Click()
    'Quit
    'Idle mode
    OutputReportData(0) = 0
    WriteReport
End
End Sub

Private Sub InputScroll_Change()
    TInput.Text = InputScroll.Value
    'update PID reference position
    p = Val(PIDForm.TInput.Text) * 4.84 '4.84 is a ROUGH Conversion factor
    OutputReportData(0) = 4
    OutputReportData(1) = (p And &HFF) 'low byte
    OutputReportData(2) = (p And &HFF00) \ 256 'hi byte

Appendix B-7: Code

    WriteReport
End Sub
Private Sub TD_Gotfocus()
BUpdate.Enabled = True 'Enable the update button to submit any changes made
End Sub
Private Sub TI_gotfocus()
BUpdate.Enabled = True 'Enable the update button to submit any changes made
End Sub
Private Sub TInput_gotfocus()
BUpdate.Enabled = True 'Enable the update button to submit any changes made
End Sub
Private Sub TP_gotfocus()
BUpdate.Enabled = True 'Enable the update button to submit any changes made
End Sub

Visual Basic FRForm.frm code

Private Sub BGraph_Click()
    'Toggle Graphing
    If BGraph.Caption = "Freeze Plot" Then 'stop graphing
        BGraph.Caption = "UnFreeze Plot"
        Collectdatabutton.Caption = "Collect Data"
        collectdata = False
    Else 'start graphing
        BGraph.Caption = "Freeze Plot"
        'timerlabel.Caption = 0
        'TimerNav.Enabled = True
        End If
        freqsetScroll.SetFocus
End Sub
Private Sub BStart_Click()
    If BStart.Caption = "Start" Then 'start sine wave
        BStart.Caption = "Stop"
End Sub

94
Frequency.Caption = (freqsetScroll.Value) / 2

'update freq resp parameters
'Hz gets converted to cycles per 1/2 period
'scale is power of 2 for multiplying std +/-8 waveform
Dim nsamp As Integer, ampscale As Integer
Dim Hz As Single
Hz = Val(Frequency.Caption)
nsamp = HzLookup(Hz) 'nsamp = CInt((1 / Hz) / 4.26666666666667E-05) / 2) - 4
'0.001365 0.010923
ampscale = 6 '7 This is the scaling factor
OutputReportData(0) = 5
OutputReportData(1) = (nsamp And &HFF) 'low byte
OutputReportData(2) = (nsamp And &HFF00) \
 OutputReportData(3) = ampscale
WriteReport

'freq resp mode
OutputReportData(0) = 2
WriteReport
Else 'stop sine
BStart.Caption = "Start"
'Idle mode
OutputReportData(0) = 0
WriteReport
End If
freqsetScroll.SetFocus
End Sub
Private Sub ClearButton_Click()
'Clear all the plotted data points
Appendix B-7: Code
i = 0
While i <= Val(totpoints.Text)
   BodePlot.XDataValues(1, i) = 0
   BodePlot.YDataValues(1, i, 0) = 0
   i = i + 1
Wend
totpoints.Text = 0
BodePlot.UpdateGraph
freqsetScroll.SetFocus
End Sub
Private Sub Collectdatabutton_Click()
If Collectdatabutton.Caption = "Collect Data" Then
   Collectdatabutton.Caption = "Stop Collecting"
   collectdata = True
Else
   Collectdatabutton.Caption = "Collect Data"
   collectdata = False
End If
End Sub
Private Sub DCGainScroll_Change()
   DCGain.Caption = DCGainScroll.Value
   Plot_Click
End Sub
Private Sub DCGainScroll_Scroll()
   DCGain.Caption = DCGainScroll.Value
   Plot_Click
End Sub
Private Sub DummyScroll_Change()
   'This is the pretty, non-blinking scroll bar
   'when it is changed, change the real scroll bar
   freqsetScroll.Value = DummyScroll.Value
End Sub
Private Sub DummyScroll_Scroll()
   'This is the pretty, non-blinking scroll bar
   'when it is changed, change the real scroll bar
   freqsetScroll.Value = DummyScroll.Value
End Sub
Private Sub fdefaultparameters_Click()
   'Reset parameters to default settings
   Mass1.Text = 250
Appendix B-7: Code

freqsetScroll.SetFocus

Private Sub fQuit_Click()
' quit
' Idle mode
OutputReportData(0) = 0
WriteReport
End
End Sub

Private Sub OscScroll_Change()
MaxUnfiltered.Top = OscScroll + 1395 'moves the osc. bar. The 1395 is the top of the graph
' find the difference between the osc. bars, then convert it to degrees:
' there are 4290 values in the scroll bar, then divide by 210 degrees
If Zoom.Caption = "Zoom" Then
Omag.Caption = Format((MinUnfiltered.Top - MaxUnfiltered.Top) / (4290 / 210), "#0.0")
Else
Omag.Caption = Format((MinUnfiltered.Top - MaxUnfiltered.Top) / (4290 / 30), "#0.0")
End If
If (Omag.Caption < 0) Then
Omag.Caption = -(Omag.Caption)
End If
freqsetScroll.SetFocus
End Sub

Private Sub OscScroll2_Change()
MinUnfiltered.Top = OscScroll2 + 1395
If Zoom.Caption = "Zoom" Then
Omag.Caption = Format((MinUnfiltered.Top - MaxUnfiltered.Top) / (4290 / 210), "#0.0")
Else
Omag.Caption = Format((MinUnfiltered.Top - MaxUnfiltered.Top) / (4290 / 30), "#0.0")
End If
If (Omag.Caption < 0) Then
Omag.Caption = -(Omag.Caption)
End If
freqsetScroll.SetFocus
End Sub

Private Sub OscScroll_Scroll()
MaxUnfiltered.Top = OscScroll + 1395
If Zoom.Caption = "Zoom" Then
Omag.Caption = Format((MinUnfiltered.Top - MaxUnfiltered.Top) / (4290 / 210), "#0.0")
Else
Omag.Caption = Format((MinUnfiltered.Top - MaxUnfiltered.Top) / (4290 / 30), "#0.0")
End If
If (Omag.Caption < 0) Then
Omag.Caption = -(Omag.Caption)
End If
freqsetScroll.SetFocus
End Sub

Private Sub OscScroll2_Scroll()
MinUnfiltered.Top = OscScroll2 + 1395
If Zoom.Caption = "Zoom" Then
Omag.Caption = Format((MinUnfiltered.Top - MaxUnfiltered.Top) / (4290 / 210), "#0.0")
Else
Omag.Caption = Format((MinUnfiltered.Top - MaxUnfiltered.Top) / (4290 / 30), "#0.0")
End If
If (Omag.Caption < 0) Then
Omag.Caption = -(Omag.Caption)
End If
freqsetScroll.SetFocus
End Sub
End If
If (Omag.Caption < 0) Then
  Omag.Caption = -(Omag.Caption)
End If
freqsetScroll.SetFocus
End Sub
Friend Sub Plot_Click()

Appendix B-7: Code

'Update the Bode Plot
j = Val(Mass1.Text)
k = Val(Spring1.Text)
b = Val(damp.Text)

'this changes letters and characters inputted to be 0
Mass1.Text = j
Spring1.Text = k
damp.Text = b
flag = 1

'test for values resulting in a division by 0
If j = 0 Then
  flag = 0
End If
If k = 0 Then
  flag = 0
End If
If flag = 1 Then 'no 0 values that will cause problems are present
  i = 0
  While i <= 10000 'plot theoretical bode plot
    BodePlot.XDataValues(0, i) = (i + 1) * 0.1591549
    BodePlot.YDataValues(0, i, 0) = 20 * Log10(((k / j) ^ 2) ^ (0.5) / ((-i ^ 2 + k / j) ^ 2 + (b / j * i) ^ 2) ^ 0.5))
    i = i + 1
  Wend

 ' Clear all the plotted data points
  i = 0
  While i <= Val(totpoints.Text)
    BodePlot.XDataValues(1, i) = 0
    BodePlot.YDataValues(1, i, 0) = 0
    i = i + 1
  Wend

 'Plot the plotted data points
  i = 0
  While i < Val(totpoints.Text)
    BodePlot.XDataValues(1, i) = freqarray(i)
    BodePlot.YDataValues(1, i, 0) = 20 * Log10(pointarray(i) / Val(DCGain.Caption))
    i = i + 1
  Wend
BodePlot.UpdateGraph
Elseif flag = 0 Then 'something was 0 that can't be
  MsgBox("Values for J, K and 'Output Scaling Factor' cannot be 0")
Else ' the y axis values were equal, thus the graph couldn't be made
  MsgBox("The max and min for the y axis cannot be the same")
End If
If FRForm.Visible Then
  freqsetScroll.SetFocus
End If
End Sub
Static Function Log10(X)
'defines Log10
Log10 = Log(X) / Log(10#)
End Function

Appendix B-7: Code

97
Private Sub Submitbutton_Click()
    'Plot a new data point
    flag = 1
    If Val(Omag.Caption) = 0 Then
        flag = 0
    End If
    If flag = 1 Then 'no 0 values that will cause problems are present
        BodePlot.XDataValues(1, Val(totpoints.Text)) = Frequency.Caption
        BodePlot.YDataValues(1, Val(totpoints.Text), 0) = 20 * Log10(Omag.Caption / DCGain.Caption)
    End If
    If flag = 1 Then 'no 0 values that will cause problems are present
        BodePlot.UpdateGraph
        pointarray(totpoints) = Omag.Caption
        freqarray(totpoints) = Frequency.Caption
        totpoints.Text = Val(totpoints.Text) + 1
    Else 'something was 0 that can't be
        MsgBox("The Output Magnitude cannot be 0")
    End If
End Sub

Private Sub Undo_Click()
    'Undo the last point
    i = Val(totpoints.Text)
    If (i > 0) Then 'to prevent negative point totals
        i = i - 1 'go back to the last point plotted
    End If
    BodePlot.XDataValues(1, i) = 0 'and remove it
    BodePlot.YDataValues(1, i, 0) = 0
    totpoints.Text = i 'update the totalpoints tally
    BodePlot.UpdateGraph
End Sub

Private Sub freqsetScroll_Change()
    Frequency.Caption = (freqsetScroll.Value) / 2
    Dim nsamp As Integer, ampscale As Integer
    nsamp = HzLookup(Val(Frequency.Caption))
    ampscale = 6 'This is the scaling factor
    OutputReportData(0) = 5
    OutputReportData(1) = (nsamp And &HFF) 'low byte
    OutputReportData(2) = (nsamp And &HFF00) \ 256 'hi byte
    OutputReportData(3) = ampscale
    WriteReport
    DummyScroll.Value = freqsetScroll.Value
End Sub

Private Sub freqsetScroll_Scroll()
    Frequency.Caption = (freqsetScroll.Value) / 2
End Sub

Appendix B-7: Code
'freq resp mode
OutputReportData(0) = 2
WriteReport

DummyScroll.Value = freqSetScroll.Value
End Sub

Private Function HzLookup(Hz)
' This looks up the correct nsamp value for a given frequency

Select Case Hz  ' Evaluate Number.
Case 1
  HzLookup = 11712
Case 1.5
  HzLookup = 7808
Case 2
  HzLookup = 5856
Case 2.5
  HzLookup = 4688
Case 3
  HzLookup = 3904
Case 3.5
  HzLookup = 3344
Case 4
  HzLookup = 2928
Case 4.5
  HzLookup = 2608
Case 5
  HzLookup = 2336
Case 5.5
  HzLookup = 2128
Case 6
  HzLookup = 1952
Case 6.5
  HzLookup = 1808
Case 7
  HzLookup = 1680
Case 7.5
  HzLookup = 1568
Case 8
  HzLookup = 1472
Case 8.5
  HzLookup = 1376
Case 9
  HzLookup = 1296
Case 9.5
  HzLookup = 1232
Case 10
  HzLookup = 1168
Case 10.5
  HzLookup = 1120
Case 11
  HzLookup = 1072
Case 11.5
  HzLookup = 1024
Case 12
  HzLookup = 976
Case 12.5
  HzLookup = 944
Case 13
  HzLookup = 896
Case 13.5
  HzLookup = 864
Case 14
  HzLookup = 832
Case 14.5
  HzLookup = 816
Case 15
  HzLookup = 784
Case 15.5
  HzLookup = 752
Case 16
  HzLookup = 736
Case 16.5
  HzLookup = 704
Case 17
  HzLookup = 688
Case 17.5

Appendix B-7: Code
HzLookup = 672
Case 18
 HzLookup = 656
Case 18.5
 HzLookup = 640
Case 19
 HzLookup = 624
Case 19.5
 HzLookup = 608
Case 20
 HzLookup = 576
Case Else  ' Other values:
 ' there shouldn't be
End Select
End Function

Private Sub Zoom_Click()
zoomtotal = 0
zoompoint = 0
If Zoom.Caption = "Zoom" Then
 Zoom.Caption = "UnZoom"
 Zoom.Enabled = False
 zoomflag = 1
Else
 Zoom.Caption = "Zoom"
 zoomflag = 0
MSDGraph.AxisMax(Y_AXIS1) = 210
 MSDGraph.AxisMin(Y_AXIS1) = 0
 MSDGraph.UpdateGraph
Omag.Caption = Format((MaxUnfiltered.Top - MinUnfiltered.Top) / (4290 / 210), "#0.0")
End If
End Sub

Appendix B-8: HzLookup Value Calculation

This is a discussion of how the nsamp values in the HzLookup function were obtained.

When the FR module is running and told to start, every tick of Timer 0 will iterate through the FR subroute once. (Fig. 4.4). With Fosc=24 MHz the base tick rate of Timer 0 is 24/4=6 MHz. Using the full range (255) of Timer 0 causes it to roll over every (1/6)*256=42.67 uS. Prescaling by 1 gives rollovers every 42.67*1=0.04267mS, which is also the sampling period.

As Figure B-8-1 shows, nsamp determines how quickly the sine wave changes direction, hence determining its frequency. Ninc determines how often the magnitude of the motor command changes. (Refer to Figure 4.5 for a code flow chart)
Another way of phrasing it would be that nsamp determines the frequency, and ninc determines the resolution of the generated sine wave (the number of steps). To ensure a good approximation of the sine wave, there are 16 total steps between the peaks of the sine wave. Hence, ninc=nsamp/16. Ninc is capped at 8 both to ensure that it stays in the -8 to 8 range during frequency shifts, but also because in binary the number 8 is 1000. If this number is shifted over by 7, the result is 100 0000 0000 or 1024, which sets the motor to full on. (In practice, this number is only shifted over by 6 in order to reduce the magnitude of the oscillations, and prevent damage to the system during extended on-time.)

There is, however, a catch. If nsamp is not evenly divisible by 16, the resulting rounding performed by the code leads to non-uniform sine waves, with ninc being unable to move throughout its -8 to 8 range before nsamp changes the wave’s direction. The workaround is to calculate the nsamp required for a given frequency, test to see if it is divisible by 16, and if not find the closest number that is and use it instead. Here lies

**Figure B-8-1:** Nsamp and ninc relationship shown over the unscaled motor command
another advantage of a fast sampling time. As shown below in Figure B-8-2, the sampling time is inversely proportional to nsamp. Thus, the faster (smaller) the sampling time, the higher the values of nsamp. With large nsamp values, the adjustment required to achieve a number divisible by 16, isn’t a large percentage of the total number, and the resulting error in frequency is greatly reduced. For instance, instead of generating a 4 Hz sine wave, the module actually generates a 4.0023 Hz wave, an insignificant difference to the measurements students will be taking.

\[
nsamp = \frac{1}{\text{Hz}} \times \frac{2}{\text{SamplingRate}}
\]

**Figure B-8-2:** Formula for Nsamp
<table>
<thead>
<tr>
<th>Desired Hz</th>
<th>Real nsamp</th>
<th>New nsamp</th>
<th>Divisor</th>
<th>New nic</th>
<th>Real Hz</th>
<th>Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>11713.8</td>
<td>11712</td>
<td>16</td>
<td>732</td>
<td>1.0006</td>
<td>0.000576332</td>
</tr>
<tr>
<td>1.5</td>
<td>7812.5</td>
<td>7808</td>
<td>16</td>
<td>488</td>
<td>1.5009</td>
<td>0.000864498</td>
</tr>
<tr>
<td>2</td>
<td>5859.37</td>
<td>5856</td>
<td>16</td>
<td>366</td>
<td>2.0012</td>
<td>0.001152664</td>
</tr>
<tr>
<td>2.5</td>
<td>4678.75</td>
<td>4688</td>
<td>16</td>
<td>293</td>
<td>2.4997</td>
<td>0.00266838</td>
</tr>
<tr>
<td>3</td>
<td>3906.25</td>
<td>3904</td>
<td>16</td>
<td>244</td>
<td>3.0017</td>
<td>0.001728996</td>
</tr>
<tr>
<td>3.5</td>
<td>3348.21</td>
<td>3344</td>
<td>16</td>
<td>209</td>
<td>3.5044</td>
<td>0.004410885</td>
</tr>
<tr>
<td>4</td>
<td>2929.69</td>
<td>2928</td>
<td>16</td>
<td>183</td>
<td>4.0023</td>
<td>0.002305328</td>
</tr>
<tr>
<td>4.5</td>
<td>2604.17</td>
<td>2608</td>
<td>16</td>
<td>163</td>
<td>4.4934</td>
<td>0.006614264</td>
</tr>
<tr>
<td>5</td>
<td>2343.75</td>
<td>2336</td>
<td>16</td>
<td>146</td>
<td>5.0166</td>
<td>0.016588185</td>
</tr>
<tr>
<td>5.5</td>
<td>2130.68</td>
<td>2128</td>
<td>16</td>
<td>133</td>
<td>5.5069</td>
<td>0.00693319</td>
</tr>
<tr>
<td>6</td>
<td>1953.13</td>
<td>1952</td>
<td>16</td>
<td>122</td>
<td>6.0035</td>
<td>0.003457992</td>
</tr>
<tr>
<td>6.5</td>
<td>1802.88</td>
<td>1808</td>
<td>16</td>
<td>113</td>
<td>6.4816</td>
<td>0.018390487</td>
</tr>
<tr>
<td>7</td>
<td>1674.11</td>
<td>1680</td>
<td>16</td>
<td>105</td>
<td>6.9754</td>
<td>0.02453571</td>
</tr>
<tr>
<td>7.5</td>
<td>1562.5</td>
<td>1568</td>
<td>16</td>
<td>98</td>
<td>7.4737</td>
<td>0.026307398</td>
</tr>
<tr>
<td>8</td>
<td>1464.84</td>
<td>1472</td>
<td>16</td>
<td>92</td>
<td>7.9611</td>
<td>0.03892663</td>
</tr>
<tr>
<td>8.5</td>
<td>1378.68</td>
<td>1376</td>
<td>16</td>
<td>86</td>
<td>8.5165</td>
<td>0.01653343</td>
</tr>
<tr>
<td>9</td>
<td>1302.08</td>
<td>1296</td>
<td>16</td>
<td>81</td>
<td>9.0422</td>
<td>0.04224537</td>
</tr>
<tr>
<td>9.5</td>
<td>1233.55</td>
<td>1232</td>
<td>16</td>
<td>77</td>
<td>9.5120</td>
<td>0.011972403</td>
</tr>
<tr>
<td>10</td>
<td>1171.88</td>
<td>1168</td>
<td>16</td>
<td>73</td>
<td>10.0332</td>
<td>0.03317637</td>
</tr>
<tr>
<td>10.5</td>
<td>1116.07</td>
<td>1120</td>
<td>16</td>
<td>70</td>
<td>10.4632</td>
<td>0.036830357</td>
</tr>
<tr>
<td>11</td>
<td>1065.34</td>
<td>1072</td>
<td>16</td>
<td>67</td>
<td>10.9317</td>
<td>0.068330224</td>
</tr>
<tr>
<td>11.5</td>
<td>1019.02</td>
<td>1024</td>
<td>16</td>
<td>64</td>
<td>11.4441</td>
<td>0.055908203</td>
</tr>
<tr>
<td>12</td>
<td>976.562</td>
<td>976</td>
<td>16</td>
<td>61</td>
<td>12.0069</td>
<td>0.00691584</td>
</tr>
<tr>
<td>12.5</td>
<td>937.5</td>
<td>944</td>
<td>16</td>
<td>59</td>
<td>12.4139</td>
<td>0.08609915</td>
</tr>
<tr>
<td>13</td>
<td>901.442</td>
<td>896</td>
<td>16</td>
<td>56</td>
<td>13.0790</td>
<td>0.078962054</td>
</tr>
<tr>
<td>13.5</td>
<td>868.056</td>
<td>864</td>
<td>16</td>
<td>54</td>
<td>13.5634</td>
<td>0.063368056</td>
</tr>
<tr>
<td>14</td>
<td>837.054</td>
<td>832</td>
<td>16</td>
<td>52</td>
<td>14.0850</td>
<td>0.085036058</td>
</tr>
<tr>
<td>14.5</td>
<td>808.19</td>
<td>816</td>
<td>16</td>
<td>51</td>
<td>14.3612</td>
<td>0.138786765</td>
</tr>
<tr>
<td>15</td>
<td>781.25</td>
<td>784</td>
<td>16</td>
<td>49</td>
<td>14.9474</td>
<td>0.052614096</td>
</tr>
<tr>
<td>15.5</td>
<td>756.048</td>
<td>752</td>
<td>16</td>
<td>47</td>
<td>15.5834</td>
<td>0.083444149</td>
</tr>
<tr>
<td>16</td>
<td>732.422</td>
<td>736</td>
<td>16</td>
<td>46</td>
<td>15.9222</td>
<td>0.077785326</td>
</tr>
<tr>
<td>16.5</td>
<td>710.227</td>
<td>704</td>
<td>16</td>
<td>44</td>
<td>16.6460</td>
<td>0.145951705</td>
</tr>
<tr>
<td>17</td>
<td>689.338</td>
<td>688</td>
<td>16</td>
<td>43</td>
<td>17.0331</td>
<td>0.03306686</td>
</tr>
<tr>
<td>17.5</td>
<td>669.643</td>
<td>672</td>
<td>16</td>
<td>42</td>
<td>17.4386</td>
<td>0.061383929</td>
</tr>
<tr>
<td>18</td>
<td>651.042</td>
<td>656</td>
<td>16</td>
<td>41</td>
<td>17.8639</td>
<td>0.13651829</td>
</tr>
<tr>
<td>18.5</td>
<td>633.446</td>
<td>640</td>
<td>16</td>
<td>40</td>
<td>18.3105</td>
<td>0.189453125</td>
</tr>
<tr>
<td>19</td>
<td>616.776</td>
<td>624</td>
<td>16</td>
<td>39</td>
<td>18.7800</td>
<td>0.219951923</td>
</tr>
<tr>
<td>19.5</td>
<td>600.962</td>
<td>608</td>
<td>16</td>
<td>38</td>
<td>19.2743</td>
<td>0.225740132</td>
</tr>
<tr>
<td>20</td>
<td>585.938</td>
<td>576</td>
<td>16</td>
<td>36</td>
<td>20.3451</td>
<td>0.345052083</td>
</tr>
</tbody>
</table>

Total Error = 2.447672336

**Figure B-8-3:** Table of frequencies and their corresponding nsamp values. “Real nsamp” is the value nsamp should be to give exactly the desired frequency. “New nsamp” is the closest number to “Real nsamp” that is evenly divisible by the “Divisor”, resulting in the “New nic”. “Real Hz” is the actual frequency of the sine wave generated by “New nsamp”. Error shows the difference between the “Real Hz and the “Desired Hz”.
It is important to note that in order to give the wide range of Hz values shown above, the timer is set to overflow every 42.67 uS or 23KHz (Note that the PID code has a built in delay to slow it’s rate to ~77Hz), while the USB data transmission to the PC is only 100Hz (the max for low speed USB). Thus, some data is not transmitted up to the student’s PC in frequency response mode. This is only noticeable when the student is zoomed in on the frequency response data, as shown below in Figure B-8-4:

Figure B-8-4: At this high zoom level, the missing data points become evident. However, students can still take accurate data measurements by measuring from peak to peak as shown here.

Thus, students should take all of their measurement data off of the highest visible peaks, as shown in Figure B-8-4.
Appendix B-9: Engineering Diagrams

Bushing

- 5/8”
- 3/8”
- 1/4” dia
- 1/8” (drill)
- #42 drill
- Black Delrin (part H5)
- +/- 0.1”

Magnet

- 1”
- 1.9”
- 1/8”
- (Use 2” holesaw)
- Magnet (part H11)
- +/- 0.1”
- Machine from paper side
PCB Modifications

Pot Hole: Expand with size W drill

Motor Hole: Expand with size X drill