University of Idaho
College of Engineering
Moscow, ID 83844

May 5, 2017

Dr. Herbert Hess, Department of Electrical and Computer Engineering
Moscow, ID 83844

Subject: Low Speed Flywheel – Controls Team Final Report

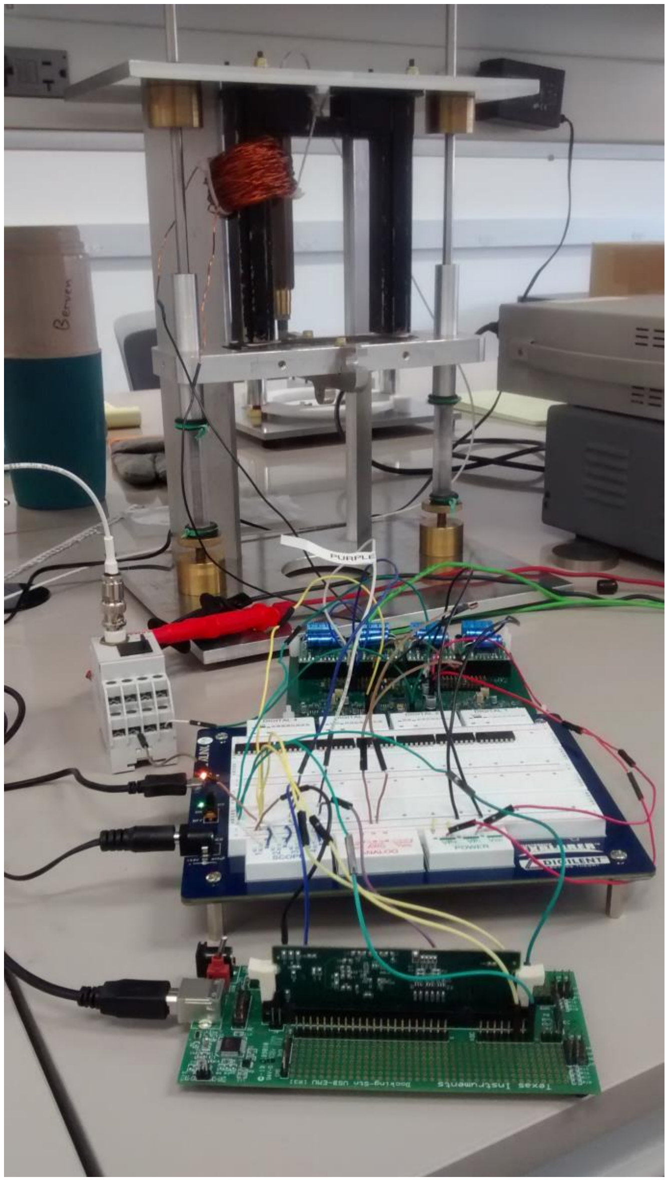
The UIFESS Controls Team is pleased to submit a final report of our findings and results on the low speed flywheel energy storage system.

Our deliverables consist of tested code for operating the stabilization bearing, as yet untested code for operating the field regulated reluctance machine, and details on setting up the single axis single bearing demonstration unit on which the stabilization code was tested. We have also included suggestions and advice on how to move the project forward.

The team encourages you to bring forward comments, concerns, and suggestions as well as any questions you may have. Thank you for your time and the advice given during the project.
Thank you for the opportunity to explore the possibilities that this project has presented.

Sincerely,

Lunar Flywheel Controls Team



Lunar Flywheel Controls

MAY 05, 2017

Team Members

Andrew Jones – Computer Engineering……………………………………..jone1258@vandals.uidaho.edu
Brian Cartwright – Computer Science…………………………………………cart1189@vandals.uidaho.edu
Ian Tanimoto – Computer Science……………………………………………..tani0633@vandals.uidaho.edu
David Arnett - Graduate Advisor………………………………………………arne9249@vandals.uidaho.edu

Table of Contents

List of Acronyms

AMB	Active Magnetic Bearing
FESS	Flywheel Energy Storage System
FlyCAM	Flywheel Control and Monitoring
FRRM	Field Regulated Reluctance Machine
GPIO	General Purpose Input and Output
HSFESS	High-speed FESS
HTS	High Temperature Superconductor(s)
LSFESS	Low-speed FESS
MCU	Microcontroller Unit
PCB	Printed Circuit Board
PID	Proportional-Integral-Derivative
PSU	Power Supply Unit
PWM	Pulse Width Modulation
SASB	Single-Axis Single-Bearing
SB	Stabilization Bearing
TI	Texas Instruments
UIFESS	University of Idaho FESS

Project Summary

The University of Idaho's Flywheel Energy Storage System (UIFESS) in an ongoing project funded by the NASA Steckler Space Grant with the goal of developing an efficient form of energy storage for use in lunar colonization. During the day, power on the Moon can be provided by solar energy. However, the lunar night lasts fourteen days, during which settlements on the Dark Side will need some way to keep the lights on. The UIFESS is intended to fill this role. The idea is that energy, collected by solar panels during the day, will be stored in flywheels and then dispersed during the night.

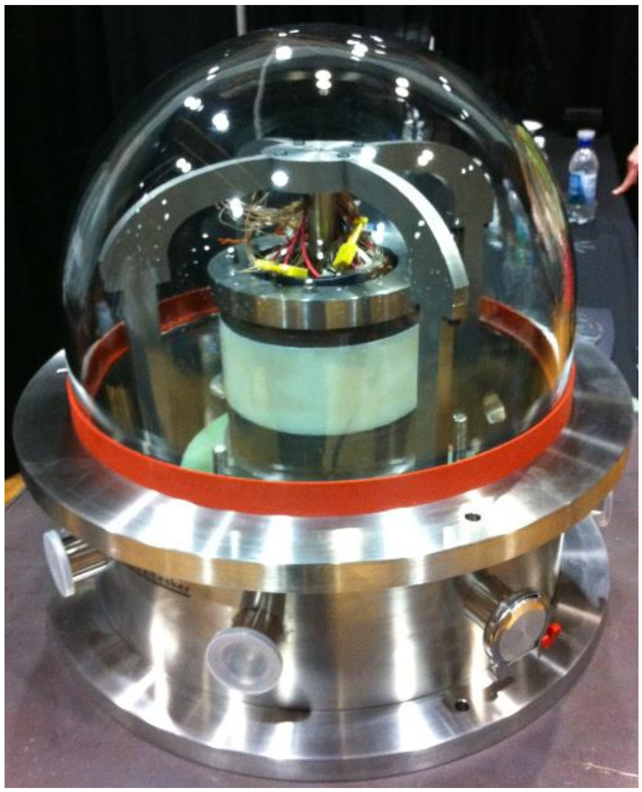


Figure 1: Assembled Flywheel Prototype

Three teams in total worked on the UIFESS project during the 2016-2017 Academic year. A team of computer scientists and electrical engineers developed Flywheel Control and Monitoring (FlyCAM), a GUI that allows for precise control of the FESS. The stator design team, consisting of mechanical and electrical engineers, developed theoretical models and equations defining the operation of a new stator designed to run at high speeds (25,000 - 30,000 RPM). The controls team worked to develop control algorithms for an existing low-speed flywheel prototype, which is intended to act as a proof of concept for the high-speed model.

Background Information

A flywheel energy storage system is a device that stores energy by spinning a rotor at high speeds. The amount of energy that can be stored is represented by the following equation:

$$KE = \frac{1}{2} I\omega^2$$

Where KE is the total kinetic energy stored within the rotor, I is the mass moment of inertia of the flywheel, and ω is the angular velocity of the rotor. The flywheel was chosen over other energy storage methods due to its high energy density.

The UIFESS utilizes a field-regulated reluctance machine (FRRM) mounted to the stator in order to magnetically accelerate and decelerate the flywheel rotor. Having the rotor rotate around the stator, rather than inside of the stator, allows for a larger rotor, increasing its moment of inertia and, therefore, the total energy that can be stored within the FESS.

The rotor is levitated by High Temperature Superconductors (HTS) and a Halbach array of permanent magnets. When liquid nitrogen is pumped through the superconductor array, the rotor will hover at an elevation of about 0.25 inches.

The completed FESS will be put in a vacuum-evacuated chamber. This, combined with the magnetic levitation and acceleration, will minimize frictional and mechanical losses of energy within the system.

Initial Project Goals

The goals of the controls team were as follows:

- Evaluate stabilization code developed by Brent Kisling and Kevin Ramus in 2014
- Evaluate position and current code developed by the Fly Rollers team in 2016
- Wire up sensors and run code on the fully assembled low speed flywheel prototype
- Perform full-speed test (~2000 RPM) on the flywheel

The following items were stretch goals and were conditional on completing the above set of goals:

- Implement more complex stabilization algorithms
- Optimize algorithm and resource usage
    - Use dual-core capability of the 77d MCU to replace one of the 335 MCUs
- Begin adaptation to High Speed FESS

Project Plan

Because the rotor is levitated and accelerated magnetically, it makes no physical contact with any other part of the FESS. The stabilization code was developed to maintain a 1mm airgap between the rotor and the stator, with a tolerance of +/- 0.1mm. Because of the small room for error, and because there is no need for the flywheel to reach its maximum angular velocity as quickly as possible, stabilization is given priority over acceleration. The position of the rotor will be constantly monitored during periods of acceleration. If the position of the rotor falls out of the tolerance band, the code will generate an interrupt and engage the stabilization code until the rotor is back in position.

A summary of the intended and actual schedule can be found in Appendix A.

Concepts Considered

We had to consider options for each of two categories: a Control system type for stabilizing the lateral position of the rotor, and a main rotation control loop, which keeps up with the rotor's rotation and accelerates or stabilizes when needed.

Lateral Stabilization

We needed an algorithm which used signals from four displacement sensors (two sets of two per magnetic bearing) and one current sensor per bearing coil, to generate a stream of signals sent to the MCU, which would direct the power electronics to stabilize the rotor laterally, such that it becomes centered on the stator and ready to resume acceleration.

We decided to go with a conditional PID (Proportional, Integral, Derivative) control, and also considered Bang-bang and Hysteresis for this role.

PID Control

Using PID control (and related combinations), the response f(x) of the system given a position error x is determined by the following equation:

$$f(x) = k_P * x + k_I * \int x \, dt + k_D * \frac{dx}{dt}$$

where k_P, k_I, & k_D are "gain" or "tuning" constants. In our case, the integral and derivative terms are approximated over discrete timesteps instead of continuously. This control scheme has the advantage, assuming proper gain tuning, of being able to avoid overshooting the target position since the response is scalable and can make very small adjustments when the error is small.

A quirk of the inherited power electronics hardware requires that one of two constant current values are supplied to the coil in one direction or the other. So to achieve intermediate current values, pulse-width modulation (PWM) is used during stabilization (also implemented in the software). In pulse-width modulation, the direction is toggled at frequency high enough that the actual coil's current doesn't have time to "catch up" before the direction is switched again. As a result, the actual current averages out to an intermediate value proportional to the duty cycle, i.e. the percentage of time that the direction is positive. This PWM is only needed for PID control, as bang-bang and hysteresis controls do not output intermediate values.

Bang-bang control

In bang-bang control, the response is simply positive if the position is below the target, and negative if the position is above target. We chose not to use this as the main position control loop because it was power-inefficient and involved higher force which is potentially dangerous; potentially overshooting the target, constantly vibrating, and/or causing more stress on the physical components. On the other hand, if the force is not high enough to overshoot, then the range at which it will be effective at all is significantly lower, or nonexistent, due to the nature of magnetic fields. However, bang-bang control is useful for the implementation of PWM mentioned above, since the current doesn't have the problem of physical inertia.

Hysteresis control

In hysteresis control systems, the adjustment response is 0 while the error is within a tolerance band, and fully positive or negative if the error is below the lower limit or above the upper limit, respectively. This is safer than bang-bang control, but wouldn't even try to return the rotor to the centered position. It also cannot be used for PWM like bang-bang can.

Rotation and Acceleration Control

Polling-based

A viable, if not optimal, solution for rotating the frame of reference is to frequently poll the angular position sensor. Then, based on its position the polarity for each coil can be set accordingly. This was used for some time due to its simplicity.

Timer-based

A more efficient solution uses the microcontrollers' timer interrupts to poll at the expected time of the next coil-shift. The timestep per shift is trivially calculated given the number of coils and current speed (which can conveniently be calculated using the length of the last timer, and the difference between the two corresponding angular positions). During these calculations, if the speed is too high for the processor running at one coil shift per minimum timestep, longer timesteps can be used in combination with shifting the coils by more than one.

Concept Selection Methods

As we inherited most of the stabilization code, the previous team had already selected some control systems for the stabilization loops. However, they hadn't explained why, so our team analyzed their work and reconsidered the other options. We found the reasons each of their systems were chosen, and decided to continue using them for the duration of the project as they're all safe. We will also suggest slight changes to the implementations of their systems in the future work section.

Regarding the rotation control systems, our constraints were, in order of priority:
Risk of failure, risk of failure, risk of failure, maximum achievable speed, CPU load, and sexiness.

Resources Used

Hardware

TI Delfino Microcontrollers

The control code runs on three Texas Instruments Delfino Microcontrollers (MCUs): one TMS320F28377d (henceforth referred to as 77d) and two TMS320F28335s (335). The stator is controlled by the 77d and one of the 335s, while the SB is controlled by the second 335. Each MCU plugs into its own C2000 Experimenter Kit dock. These docks provide access to the MCUs' GPIO and ADC pins. Any wiring diagrams showing connections to the MCU refer to the pins on these docks rather than the MCU itself.

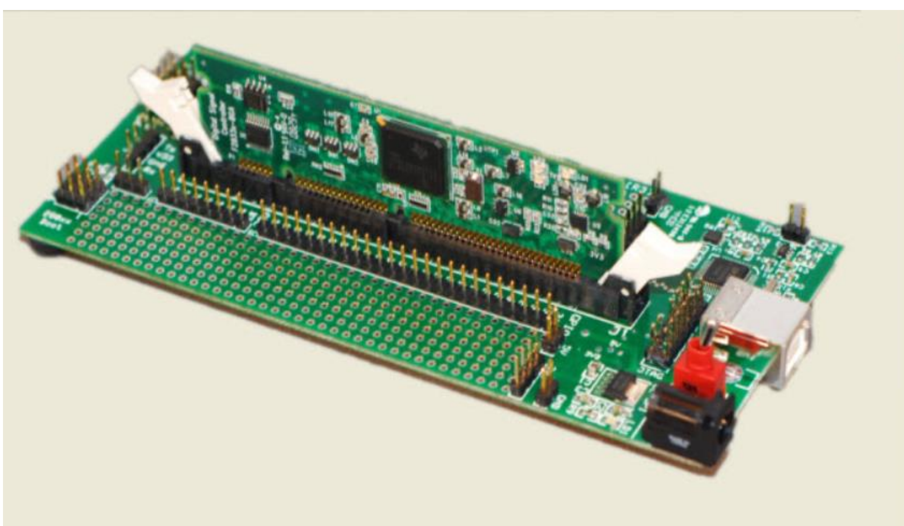


Figure 2: TMS320F28335 MCU plugged into its dock. The MCU is between the white clips.

Kaman Displacement Sensor

The airgap between the rotor and the stator is monitored by four Kaman KD-2306 Eddy Current Displacement Sensors using 9U probes. Two probes, separated by 90°, are aimed at the top of the rotor in order to monitor its movement in two dimensions. The other two probes are aimed at the bottom of the rotor in the same configuration. This setup allows the sensors to monitor both translational movement as well as tilt.

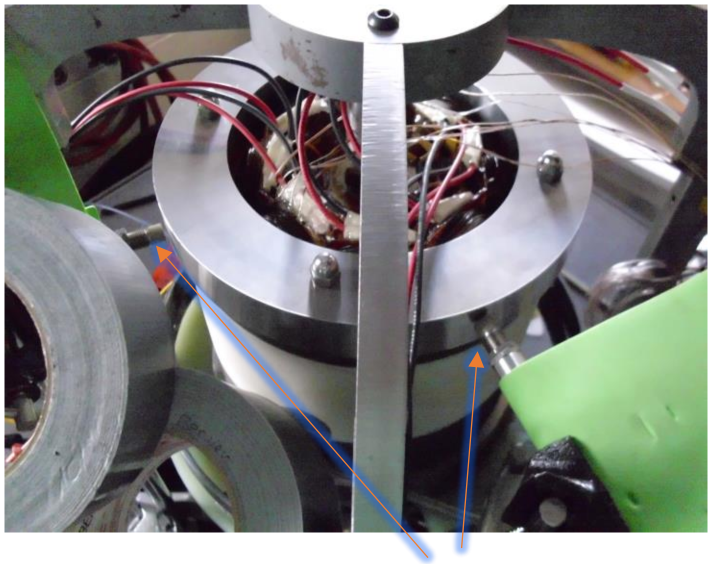


Figure 3: Positioning of 9U probes

Power Electronics PCB

Kevin Ramus, a grad student who worked on the Flywheel project in 2014, designed the print circuit boards (PCBs) that are used for signal conditioning and interfacing the MCU with the FRRM and SB coils and the displacement sensors. There are eight of these boards in total: one designated UIFESS Driver #1 (January 2014) and seven designated UIFESS Driver #2 (March 2014). Each PCB is capable of controlling the current through four coils. As such, only seven of the PCBs are necessary to control the final configuration: six PCBs for controlling the 24 coils in the FRRM, and one PCB for controlling the four coils in the SB. The seven Driver #2 PCBs should be used, with Driver #1 being kept as a backup.

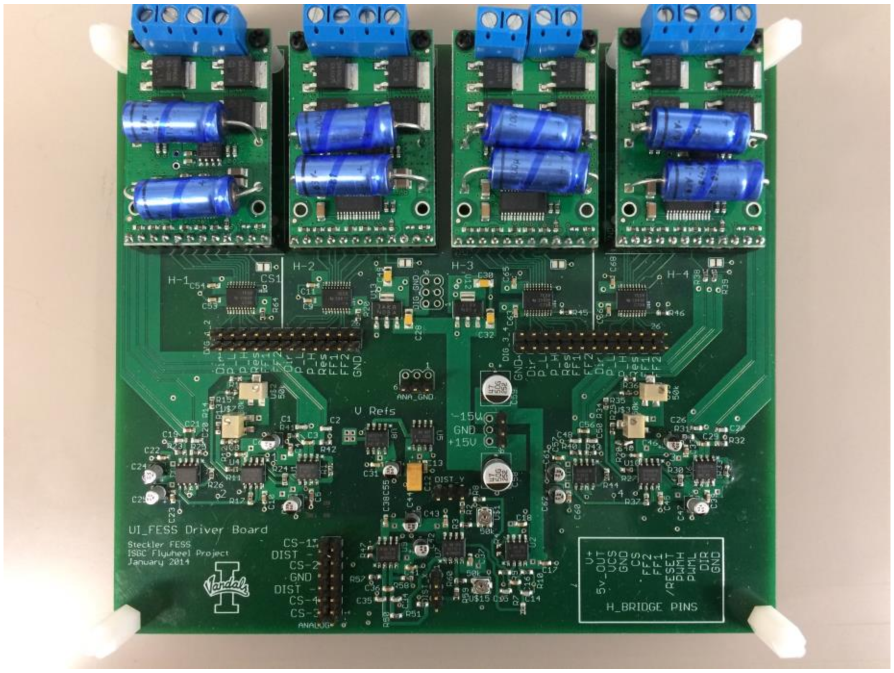


Figure 4: UIFESS PCB Driver #1 (Pololu units along the top)

Each PCB contains four Pololu High-Power Motor Drivers (model 36v20 CS, Pololu item #1457) that control the voltage across the connected magnetic coils. The Pololu uses an H-bridge to rapidly reverse the direction of current flow through the coil. The direction is changed using PWM. The PWM duty cycle changes to generate a moving average of the voltage across the coil, allowing it to change the strength of the magnetic field and keep the rotor in the desired position.

It should also be noted that there exists a prototype PCB, although it can only control one coil at a time, and as such should only be used on the Single-Axis Single-Bearing demonstrator discussed later in this section.

Digilent Explorer Board

The Explorer breadboard, made by Digilent Inc., is typically used for prototyping analog and digital circuits for educational purposes, and is used in 200- and 300-level ECE courses at the University of Idaho. It was used on the UIFESS project as an oscilloscope for monitoring the displacement and current sensor outputs from the power electronics for testing and debugging purposes. It was also used as a wire splitter, as the displacement and current sensors each need to connect a single output from the PCB to two inputs on the MCU. As the project progresses, the Explorer board will become unnecessary, its functions replaced by dedicated wiring and the FlyCAM user interface.

Single-Axis Single-Bearing Demonstration Unit

The single-axis single-bearing (SASB) unit is used to demonstrate the stabilization code's ability to maintain the airgap between the rotor and the stator. It consists of an electromagnet, a flotor (floating rotor analog) containing a mundane iron bar, and a Kaman 9U position/displacement sensor probe. While in operation, the flotor will hover approximately 1mm below the electromagnet, even when a reasonable force is applied to the flotor. Because the SASB uses only one magnet and one displacement sensor, code runs on just one 335 MCU. The aforementioned prototype PCB was designed and built to be used on this SASB; however, we used the #1 driver for our own testing.

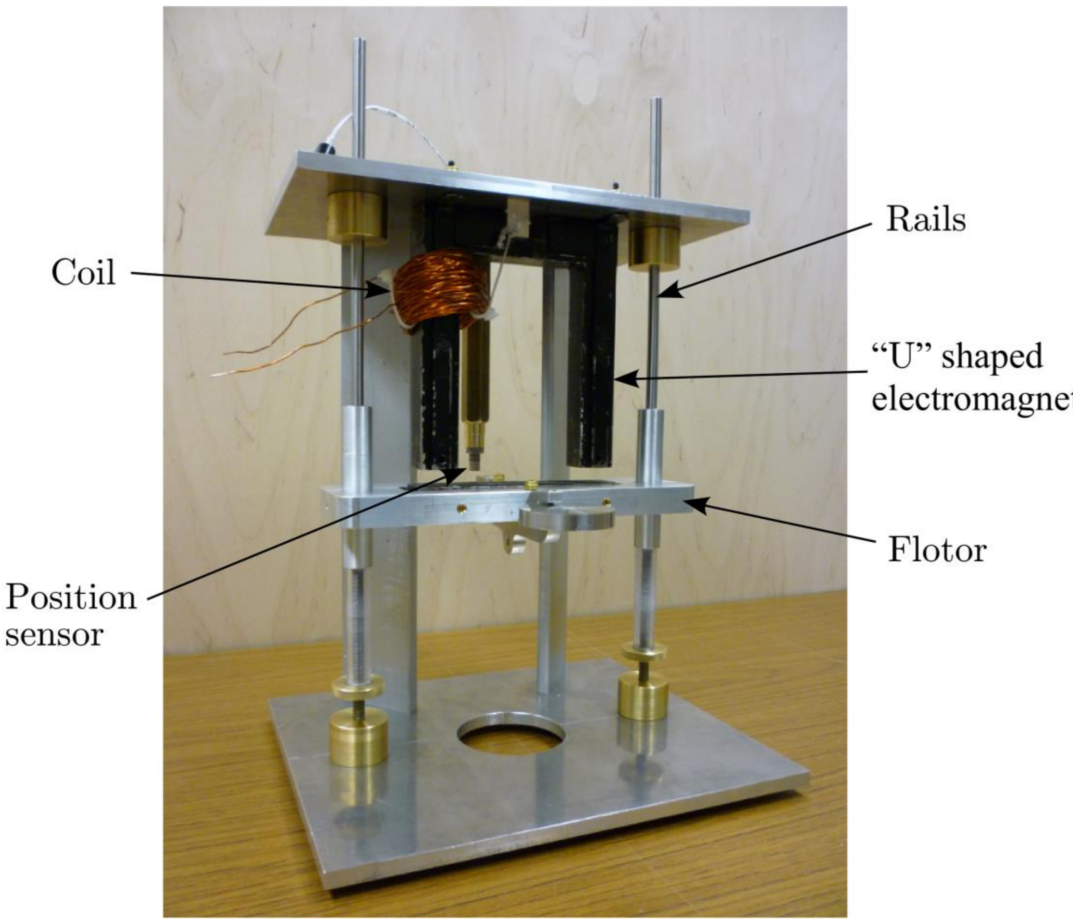


Figure 5: The SASB Demonstrator

When the SASB unit is turned on and the flotor is moved toward the magnet, the initial pull of the magnet causes the flotor to accelerate through its stable 1mm airgap and lock up to the magnet. To prevent this, a spacer needs to be placed between the flotor and the magnet to prevent the lock up (we used guitar picks). Once the initial inertial force is dispelled, the spacers can be removed and the flotor will hover as expected.

To calibrate the position sensor, touch the flotor to the magnet and tighten the locking rings. While monitoring the displacement sensor output with either the Digilent Waveforms software or another oscilloscope, thread the sensor probe in and out of the mounting rod until the sensor output reads zero volts. Note that the guitar picks cannot be present during calibration.

Power Supplies

A total of three power supply units (PSUs) were necessary to successfully run the SASB unit:

- A Hewlett-Packard E3631A provided +/- 15V to the PCB
- An Agilent U8002A provided 24V to the displacement sensor
- A Sorensen XHR 150-7 provided 24V to the magnetic coil via the Pololu on the PCB

These exact PSUs may or may not be available at the time of this reading; they were borrowed from various sources across campus and may need to be returned. The PSU that powers the coil must be in constant voltage mode for the coil to behave properly. Instructions for entering CVM can be found in the user manual of the PSU.

Software

The code editing software used was Texas Instruments (TI)’s Code Composer Studio (v5.5 specifically), with TI’s Control Suite addon for hardware-specific code packages for interfacing with the microcontrollers.

The software Digilent Waveforms (2015) was used to interface with the Explorer board, so that we could use it as an oscilloscope and waveform generator for testing purposes.

Control Code

Brent Kisling and Kevin Ramus wrote and mostly debugged the SB and SASB control code in 2014, and provided some documentation. The 2015-16 Fly Rollers started on writing FRRM control code, but with lacking comments and no testing. Our team wrote readable pseudocode for the FRRM control scheme, to be used to analyze and finish the Fly Rollers’ work.

Results

Due to time constraints and supply issues, we were unable to test the stabilization code on the prototype flywheel itself. However, the stabilization code was successfully tested and demonstrated on the SASB. Thorough documentation has been written and will be given to future teams so they can get started right away without the same delays we dealt with. This documentation includes equipment locations, assembly instructions, concept explanation, and general advice.

Future Work

Tasks

We were not able to test the stabilization on 2 axes with the actual low-speed FESS, since the rotor was being balanced during the timeframe we were around to test it. Once the rotor is balanced, setting up the 2-axis stabilization code should be relatively straightforward, and we deem said code safe to run. Assembly and wiring of the LSFESS may take 1-2 solid days, not including the significant time it takes to collect all the hardware, software, and knowledge.

The Digilent Explorer board still has most of the wires connected from our test setup. However, due to the fickle nature of breadboard wiring, future teams may need to reconnect loose wires. The logic gates plugged into the top of the breadboard are not part of the test setup.

A new 335 MCU (or its successor, if there is one) will need to be purchased from Texas Instruments. Two of the original three 335s got fried at some point before we were assigned to the project. We only needed one to perform our tests, but a second one will be necessary to run the completed flywheel, unless some resource optimization can be done beforehand (read on for more details).

Once the 2-axis code is confirmed to run, consider changing from the current hybrid PID/PD stabilization system to always PID. This should be safe, testing it wasn't a priority. (See lines 592-601 of "SB_Final_Version\main.c ". Ask your mentors first, of course.

Our acceleration code needs to be translated to proper c using the appropriate TI code suites for each MCU model in order to be tested. This is where the most work will be required. The existing SB code can be used as an example of how MCU pins can be set, and the existing Fly Rollers' FRRM code should have a usable framework, in theory. TI's 77d MCU package will need to be found and used.

While the controls are currently designed with three MCUs in mind, it may be possible to optimize the code to run on just two MCUs (the 77d and one of the 335s). The 77d has two cores, as well as two programmable Control Law Accelerators (CLAs) that can respond to peripheral triggers and execute code concurrently with the main CPU. This function may be able to replace one of the 335 MCUs. Research into this kind of optimization may be needed.

Advice for Future Teams

Besides the first teams' theses, We recommend reading the Wikipedia pages for PID control, Pulse Width Modulation, and Reluctance Motors. None of these are exactly the same as our systems, but they're related concepts and a good way to get background knowledge.

One of the time sinks we encountered was the lack of available power supplies. We spent about a month waiting for a third power supply to be made available to us. Be sure to secure your power supplies as early as possible, assuming they are not still in the project's storage cabinet (See Appendix B)

When the prototype FESS is fully assembled, a 24V power supply providing about 300A will be needed to provide parallel voltage to the coils.

Hooking up the Stabilization Bearing and running its code on it early should have the benefits of helping understand the code and systems more quickly, getting practice assembling the components, and accomplishing a portion of the final product assembly. Try to do this as early in the semester as your supervisors will allow.

Appendix A Project Timeline (Original and Final)

Original project schedule

Task	Start	End	Cal. Days	Work Days
High Speed Flywheel Model				
Project Learning	Mon 9/12/16	Thu 11/10/16	60	44
Thesis Reading	Mon 9/12/16	Wed 10/19/16	38	28
Previous Senior Design Projects	Wed 9/21/16	Sun 10/30/16	40	28
Client Interview	Wed 9/21/16	Fri 9/23/16	3	3
Previous Code Review	Thu 9/29/16	Thu 11/10/16	43	31
Software Acquisition	Sat 10/01/16	Fri 10/21/16	21	15
Project Outline	Sat 10/15/16	Fri 11/18/16	35	25
Design Finalization	Wed 10/26/16	Wed 1/18/17	85	61
Graphing required structures	Wed 10/26/16	Wed 1/18/17	85	61
Comparing existing code	Mon 11/28/16	Wed 1/18/17	52	38
Coding	Wed 1/11/17	Thu 2/09/17	30	22
Correcting existing code	Wed 1/11/17	Thu 2/09/17	30	22
Code Review/Verification	Mon 1/23/17	Thu 2/09/17	18	14
Testing	Mon 1/23/17	Tue 4/18/17	86	62
Code testing	Sat 2/04/17	Fri 3/03/17	28	20

Power hardware testing	Mon 1/23/17	Fri 3/03/17	21	30
Very-low speed system testing	Fri 3/03/17	Thu 3/30/17	28	20
Full speed testing	Thu 3/30/17	Tue 4/18/17	20	14
Portfolio Review for Completeness	Wed 4/26/17	Wed 5/10/17	15	11
EXPO Prep	Wed 4/19/17	Wed 5/03/17	15	11

Project schedule as executed

Task	**Start**	**End**	Cal. Days	**% Done**	**Work Days**
High Speed Flywheel Model					
Project Learning	Mon 9/12/16	Thu 11/10/16	60	100%	44
Thesis Reading	Mon 9/12/16	Wed 10/19/16	38	100%	28
Previous Senior Design Projects	Wed 9/21/16	Sun 10/30/16	40	100%	28
Client Interview	Wed 9/21/16	Fri 9/23/16	3	100%	3
Previous Code Review	Thu 9/29/16	Thu 11/10/16	43	100%	31
Software Acquisition	Sat 10/01/16	Fri 10/21/16	21	100%	15
Project Outline	Sat 10/15/16	Fri 11/18/16	35	100%	25
Design Finalization	Wed 10/26/16	Wed 1/18/17	85	100%	61
Graphing control structures	Wed 10/26/16	Fri 3/24/17	150	100%	108
Comparing existing code	Mon 11/28/16	Wed 1/18/17	52	100%	38
Coding	Wed 1/11/17	Thu 2/09/17	30	100%	22

Write pseudocode	Wed 1/11/17	Thu 2/09/17	30	100%	22
Review existing code	Mon 1/23/17	Thu 2/09/17	18	100%	14
Assembly and wiring	Wed 1/11/17	Sat 3/25/17	74	60%	53
SASB	Wed 1/11/17	Fri 2/17/17	38	100%	28
Low-Speed Flywheel	Tue 2/14/17	Sat 3/25/17	40	20%	29
Testing	Fri 2/17/17	Wed 4/12/17	55	83%	39
Code testing	Fri 2/17/17	Thu 3/16/17	28	100%	20
Power hardware testing	Fri 2/17/17	Thu 3/16/17	21	100%	20
Very-low speed system testing	Thu 3/16/17	Wed 4/12/17	28	50%	20
Portfolio Review for Completeness	Wed 4/26/17	Wed 5/10/17	15	99%	11
EXPO Prep	Wed 4/19/17	Wed 5/03/17	15	100%	11

Appendix B: Current Inventory of Equipment

The following items are located in locker 3001 inside the Integrated Research and Innovation Center's first floor Flex Lab (IRIC 120):

- The Single-Axis Single-Bearing Demonstrator Unit
- Texas Instruments C2000 Microcontrollers Development Tools x 4
    - Two of these boxes are marked as "Bad". This referred to the microcontrollers themselves. The marked boxes still contain the MCU docks, which still work as far as we know.
    - One of the remaining boxes contains a working 335 MCU and dock.
    - The last box contains the working 77d MCU and dock.
- Kaman KD2306-9U Displacement Sensor x 4
    - One of the 9U probes is mounted to the SASB.
- Digilent Explorer Board
- Kevin Ramus' Power Electronics PCB x 8
    - March 2014 (Rev 2) Driver x 7
        - The second Pololu on board 5 may have a burnt out capacitor. If so,

only the Pololu needs to be replaced, not the entire board.
    - January 2014 (Rev 1) Driver x 1
- Agilent 54622D Mixed Signal Oscilloscope

There may or may not be power supply units in the cabinet at the time of reading, as they may have been returned to their original sources.

There are several other items in the locker that belong to the UIFESS project as a whole, but were not dealt with by the controls team.

Appendix C: Additional Resources

A complete collection of of our findings, as well as documents from previous teams, can be found on the University of Idaho Shared drive:

\\files.uidaho.edu\shared\Engineering\SeniorDesign\- Group Folders\Flywheel\Year 5\2016-2017\Low Speed