Control Moment Gyroscope Self-Balancing Cube

Srijan Duggal, Kyle Heiss, and Alex Yu

ME 4012 Modeling and Control of Motion Systems, Spring 2021

Abstract

Control moment gyroscopes are used to produce external torque in a system when power efficiency is a necessary consideration, such as attitude control on spacecraft [1]. This torque-producing method can also be used to stabilize systems such as a cube balancing on its side. The system was modeled as a classic inverted pendulum with a fixed point of rotation. The system was stabilized using a PID controller designed using the root locus technique. The idealized design of the controller needed to be tuned during system testing to achieve system stabilization that met the specifications.

Introduction

Control moment gyroscopes are used in many applications, especially attitude control on satellites [1] and spacecraft, as well as gimbals. Control moment gyroscopes contain a flywheel that rotates at a constant angular velocity with a method of changing the orientation of the flywheel. By changing the axis about which the flywheel spins, a moment is produced which can position the system. A demonstration of this ability to produce an external torque is the balancing of a cube on its edge, like the problem of balancing an inverted pendulum. Another physical mechanism to balance the cube is to have a reaction wheel which produces a restorative torque when the speed of the flywheel is changed. Control moment gyroscopes are more power efficient than reaction wheels as they only need to spin the flywheel up to speed once and then change its direction. This requires far less energy than constantly controlling the speed of a potentially high-mass flywheel. This is particularly important in aerospace applications of control moment gyroscopes where energy efficiency is crucial. Control moment gyroscopes are widely used in applications where an external torque needs to be produced efficiently.

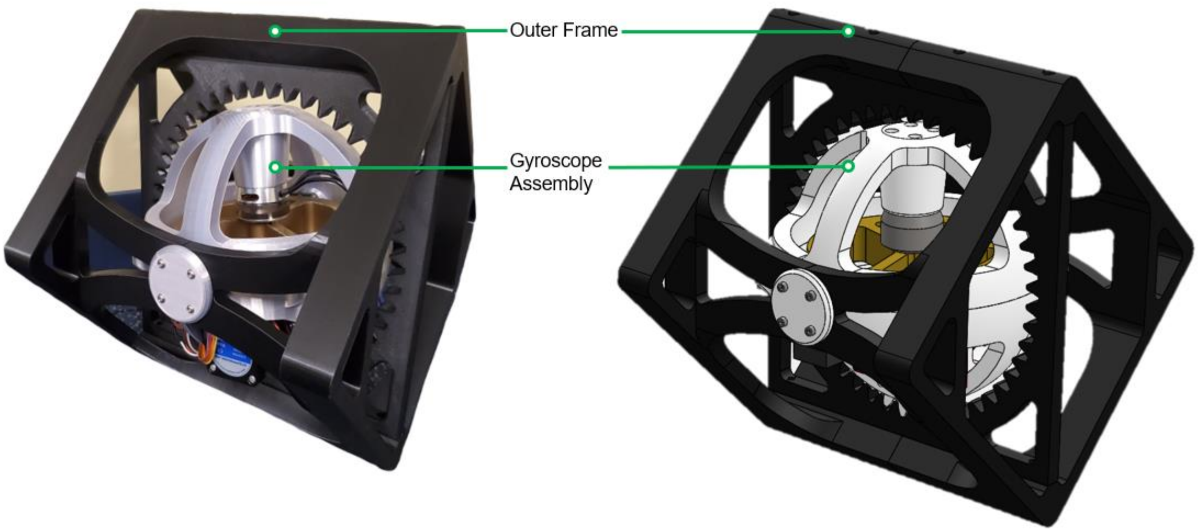
Physical Design

Table 1: List of manufactured components

Part	**Manufacturing Method and Material**
All structural components	3D Printing (PLA)
Ring & pinion gear	3D Printing (PLA)
Flywheel	Waterjet (Aluminum 6061)

Table 2: List of purchased/pre-existing components

Purchased Components	**Pre-existing Components**
Arduino Nano A4988 stepper motor driver Nema 17 stepper motor	MPU 9250 inertial measurement unit AS5047 hall effect encoder Brushless DC motor driver Brushless DC motor 3S 1300mAh LiPo battery Buck switching voltage regulator



0

Figure 1: *Overall view of the system*

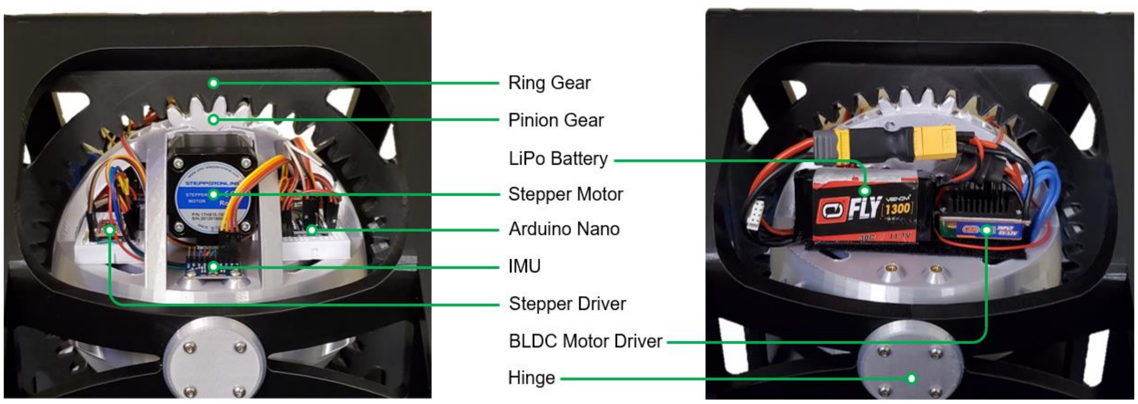


Figure 2: *Upper half of the gyroscope assembly containing the majority of electronics*

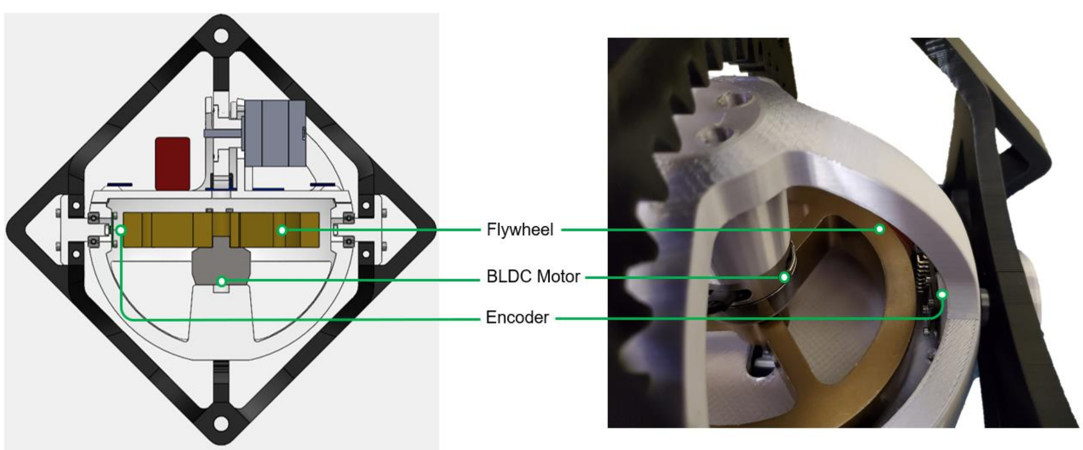


Figure 3: *Lower half of the gyroscope assembly containing the flywheel and encoder*

Important design considerations include:

- Electronics placement: All electronics were placed on the rotating gyroscope assembly. This eliminated the need to cross wires from a rotating body to the stationary outer frame and being caught and cut.

- Gyroscope assembly actuation: By actuating the gyroscope assembly internally with a gear, all electronics were able to be placed within the gyroscope assembly. In addition, a gear ensures no slip (as opposed to a belt) for precise velocity control. Gear actuation also allowed a gear reduction, reducing the torque required of the stepper and improving the velocity resolution of the system.

Modeling

The system was modeled as a classic inverted pendulum with a fixed pivot point at its base, as shown in the free body diagram in Figure 4.

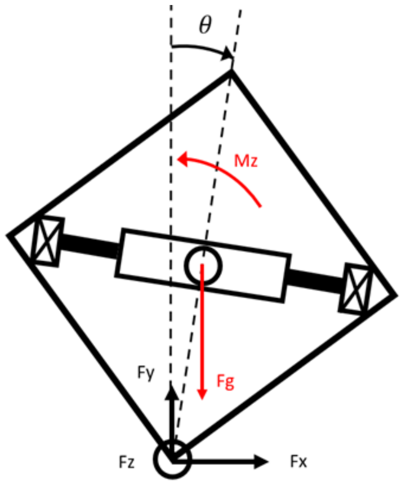


Figure [4]: System Modeled as Inverted Pendulum with Fx,y,z being the reaction at the base, Fg being force from gravity, Mz being restorative torque needed to stabilize the system, and θ being angle from vertical.

The balance of moment from the free body gives the relationship between θ and moment required to stabilize the system.

$$\Sigma M_z\colon\ I_{cube}\ddot{\theta} = M_z - m_c * g * \sin(\theta) * L\ ,$$

where $I_{cube}$is the moment of inertia of the cube, $m_c$is the mass of the cube, and L is the distance to the center of mass of the cube from the pivot point.

Due to the nonlinearity introduced by the sin(θ) term, the system needs to be linearized. The system is defined as θ being the angle from vertical. The team used a small angle approximation of sin(θ) = θ as theta approaches zero. This results in a new moment balance:

$$\Sigma M_z\colon\ I_{cube}\ddot{\theta} = M_z - m_c * g * L\ * \theta$$

Taking the Laplace transform of this system yields

$$\frac{\theta(s)}{M_z(s)} = \frac{-1}{I_{cube}s^2 -\ m_c * g * L}\ .$$

This transfer function relates the required stabilizing moment experienced by the cube with the angle the cube is tilted. However, the controlled variable is the moment produced by the gyroscope, not the one experienced by the cube. Since the cube experience a moment equal and opposite to the one produced by the motor, the transfer function becomes:

$$\frac{\theta(s)}{M_{z,commanded}(s)} = \frac{1}{I_{cube}s^2 -\ m_c * g * L}\ .$$

A figure of the system showing how the flywheel angle relates to the torque along the Z-axis is shown below.

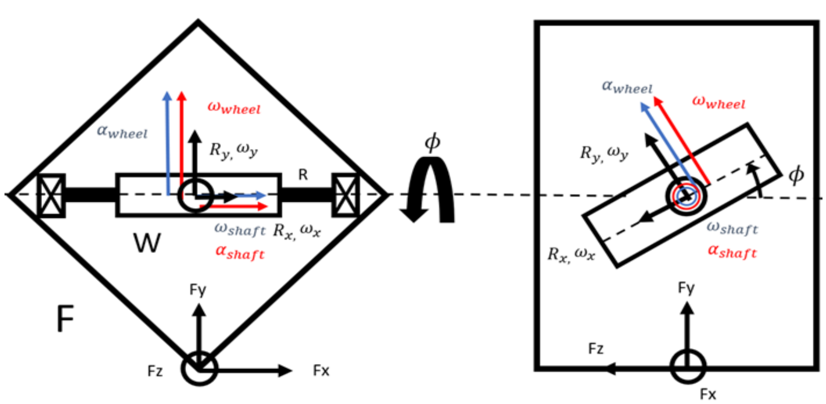


Figure [5]: System Frame of References and Relationship between Flywheel Axis of Rotation and Moment Produced

As shown in the figure, there are three frames of reference relevant to developing the relationship between flywheel angle and moment along the z-axis. The W frame is rotating with the wheel at the center of mass of the. The R frame is rotating with the shaft at the center of mass of the wheel. Finally, the F frame is fixed. Using the derivative theorem, the moment produced at the center of mass of the wheel.

$$\textit{Derivative Theorem – (Express everything in coordinate of R frame)}$$
$$\Sigma M_c * {}^{F}\dot{\overrightarrow{H_c}} = {}^{R}\dot{\overrightarrow{H_c}} + \overrightarrow{\omega_{\frac{R}{F}}} * \vec{H}_c,$$

(where M_c is the moment about the center of mass and H represents angular momentum, and ω is the angular velocity of the wheel).

This shows the relationship between the moment produced at the center of mass expressed in the R frame. The angular momentum of the flywheel can be represented by,

$$^{R}\dot{\overrightarrow{H_c}} = \mathrm{I_{yy}}\alpha_{\mathrm{wheel}}\widehat{J_R},$$

(where $I_{xx,yy,zz}$ is the rotational inertia about a the corresponding axis and $\boldsymbol{\alpha}$ is the angular acceleration of the wheel)

and the angular velocity of the wheel can be represented by the speed of shaft rotation, $\dot{\phi}$

$$\overrightarrow{\omega_{\frac{R}{F}}} = \dot{\phi}\hat{\imath}_R,$$

(where ϕ represents the rotation of the shaft). The moment of the system about its center of mass can be represented as,

$$\vec{H}_c = [\mathrm{I_{wheel}}]\overrightarrow{\omega_{wheel}} = \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix} \begin{bmatrix} \omega_{shaft} \\ \omega_{wheel} \\ 0 \end{bmatrix} = I_{xx}\dot{\phi}\hat{\imath}_R + I_{yy}\,\omega_{wheel}\widehat{J_R}.$$

Using this reduced form, the moment of the system about its center of mass can be shown as,

$$\vec{M}_c = \mathrm{I_{yy}}\alpha_{\mathrm{wheel}}\widehat{J_R} + \dot{\phi}\hat{\imath}_R \times (\mathrm{I_{yy}}\dot{\phi}\hat{\imath}_R + I_{yy}\omega_{wheel}\widehat{J_R}),$$

$$\vec{M}_c = \mathrm{I_{yy}}\alpha_{\mathrm{wheel}}\widehat{J_R} + \dot{\phi}\omega_{wheel}I_{yy}\widehat{K_R},$$

However, we are concerned about the moment produced along the fixed frame, F, not the frame R.

$$\vec{M}_c = \mathrm{I_{yy}}\alpha_{\mathrm{wheel}}(\sin(\phi)\widehat{K_F} + cos(\phi)\widehat{J_F}) + \; +\mathrm{I_{yy}}\dot{\phi}\omega_{wheel}(\cos(\phi)\widehat{K_F} - sin(\phi)\widehat{J_F}),$$

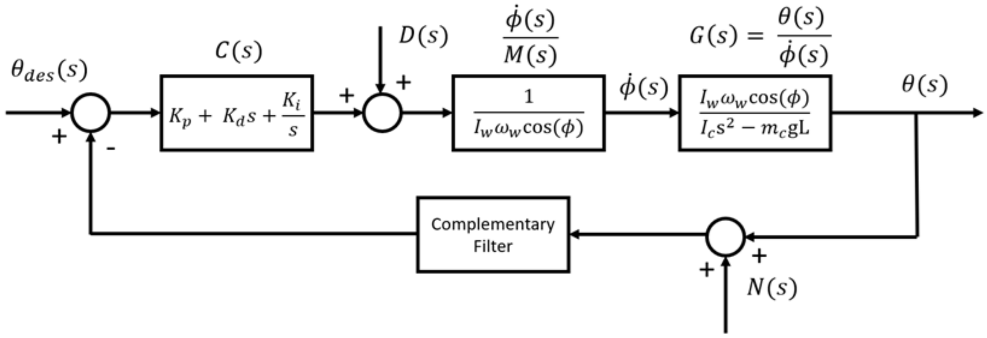
shows the same equations with the unit vectors substituted for the fixed frame representations. The cube is balancing about the z-axis of rotation, so it is necessary to isolate the z-direction:

$$\vec{M}_c = \mathrm{I_{yy}}(\alpha_{\mathrm{wheel}}\sin(\phi) + \dot{\phi}\omega_{wheel}\cos(\phi)).$$

The flywheel is spinning at a constant angular velocity, so its angular acceleration is zero, leading the final equation representing how the rotation of the shaft, ϕ, related to the moment produced about the center of mass of the system along the z-direction:

$$\vec{M}_z = \mathrm{I_{yy}}\dot{\phi}\omega_{wheel}\cos(\phi).$$

Following this derivation, the overall system block diagram, open loop, and closed loop transfer are shown in Figure 6 with a complimentary filter needed for angle estimation and a PID controller needed for stabilization.



$$OLTF = \frac{(K_p + K_d s + \frac{Ki}{s})}{I_c s^2 - m_c g L} * Complementary\ Filter$$

$$CLTF = \frac{(K_p s + K_d s^2 + K_i)}{I_c s^3 + K_d s^2 + (K_p - m_c g L)s + K_i} (with\ complementary\ filter\ modeled\ as\ unity\ gain)$$

Figure [6]: System Block Diagram, Open Loop, and Closed Loop Transfer Functions

The values for these variables such as mass and moment of inertia were determined through estimations using Solidwork's properties. The flywheel speed was determined experimentally, by spinning up the flywheel and observing the frequency of oscillations in the accelerometer data. A Fourier transform of the accelerometer data was computed as shown below, and based on the frequency of vibration, the flywheel was determined to be spinning at approximately 209 radians per second.

Parameter	Value
I_{cube}	$7.24E-3\ kgm^2$
I_{wheel}	$5.24E-4\ kgm^2$
ω_{wheel}	**206 rad/s**
m_{cube}	$1.55\ kg$
m_{wheel}	$0.383\ kg$
L	$0.141\ m$
g	$9.81 \frac{m}{s}$

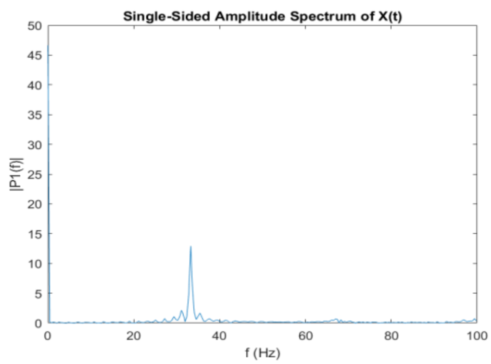


Figure [7]: System Parameters and Fourier Transform for Wheel Speed

Controller Design

The team chose to use root locus as the design method for the controller. The plant function is represented by, $\frac{\theta(s)}{M_{z,commanded}(s)} = \frac{1}{I_{cube}s^2 - m_c * g * L}$, and the root locus of P, PD, and PID controllers are shown below in Figure [8]. The root locus for the P control shows that the system is inherently unstable due to the pole on its right-hand plane. The step response shows that the system cannot be brought stable with just a P gain. The PD controller root locus has loci that extend over to the left-hand plane, showing that the system can be stabilized with a sufficient K gain. This is further reflected in the PD step response. One consideration of control moment gyroscopes is their effective range of use. If the direction of angular momentum becomes in-line with the axis of rotation of the cube, the system is not able to produce enough external torque to stabilize and the system reaches saturation. This means that a PID controller is necessary to avoid steady state error. Steady state error means a constant torque is required to keep the system stabilized which would in turn mean a constant rotation of the flywheel's angular momentum and a higher probability that the system reaches saturation. With the addition of an integral control, the system experiences higher overshoot but steady state error is eliminated as shown in the PID step response.

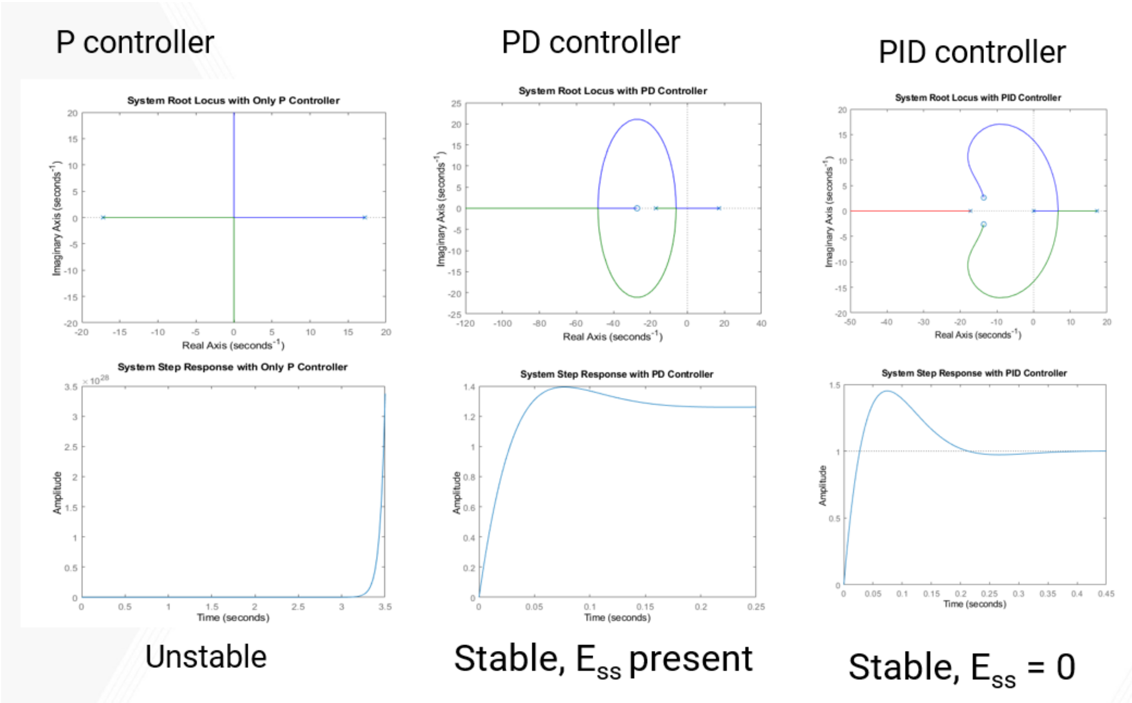
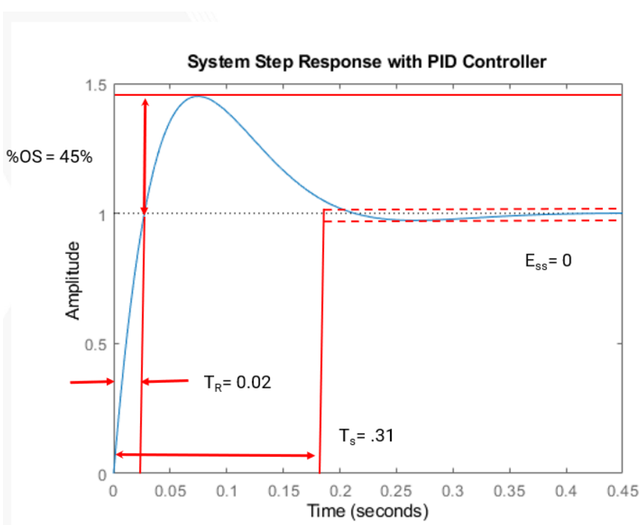


Figure [8]: Plant Function Root Locus under P, PD, and PID control

Using the knowledge that PID is required to stabilize the system with zero steady-state error, the team designed the controller in Figure [9]. The controlled system step response is also shown along with the step response specifications.



$$G(s) = \frac{1}{I_c s^2 - m_c g L}$$

$$C(s) = \frac{0.378(s^2 + 27.23 + 192.2)}{s}$$

System Requirements to a step

- Stable
- Settling Time: $T_s < .4$ s
- Rise Time: $T_R < 0.1$ s
- Steady State Error: $E_{ss} = 0$
- Overshoot: %OS < 50%

Figure [9]: Plant Function Root Locus under P, PD, and PID control

Implementation

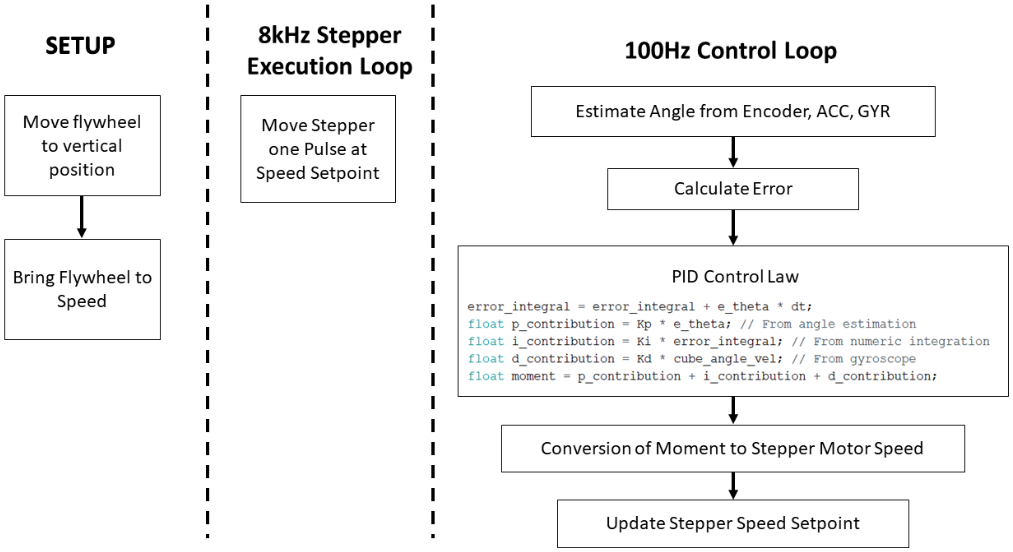


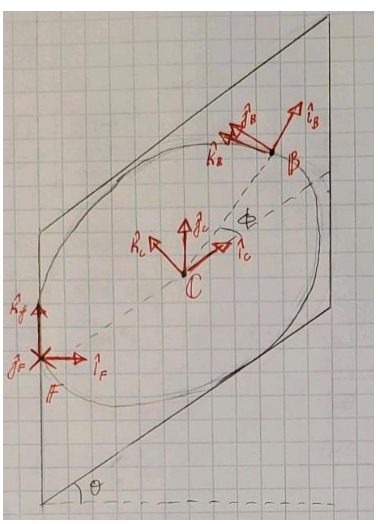
Figure [10]: Software Implementation Flow

The final software implementation follows the diagram (Figure 10). The key aspects are the angle estimation, PID control law, and the conversion of moment to stepper motor speed.

Angle Estimation

To get an accurate angle estimate, various computations on the raw IMU data were necessary, as it was mounted on the rotating gyroscope assembly. This involved taking the raw gyroscope and accelerometer data and creating estimates of the cube angle (θ). These two estimates were then combined using a complementary filter for the final angle estimate.

Angle Estimate from Accelerometer



To derive the angle estimate from the accelerometer, start with the figure to the left. There are three important body-fixed reference frames here: **F** (fixed frame, with coordinate system attached to the table / floor / bed), **C** (the cube frame, with coordinate system located at the center of the cube), **B** (the ball frame, with coordinate system located at the accelerometer). $\hat{k}_c$ is the axis of the cube that goes through the encoder and $\hat{k}_b$ is the axis of the ball that goes through the encoder. $\hat{j}_c$ and $\hat{j}_F$ align (this is the axis about which the cube is rotating).

The accelerometer gives the acceleration of frame **B** with respect to frame **F**, in the coordinates of frame **B**. It is necessary to transform this value into the coordinates of frame **F**, which uses the transformation matrices from the following equations.

The equation

$$ {}^{C}_{F}R = \begin{bmatrix} cos(\theta) & 0 & -sin(\theta) \\ 0 & 1 & 0 \\ sin(\theta) & 0 & cos(\theta) \end{bmatrix}, $$

represents the rotation of the cube about the fixed frame counterclockwise by an angle of theta (the cube tilting by an angle of theta). The equation

$$ {}^{B}_{C}R = \begin{bmatrix} cos(\phi) & -sin(\phi) & 0 \\ sin(\phi) & cos(\phi) & 0 \\ 0 & 0 & 1 \end{bmatrix}, $$

represents the rotation of the ball about the cube frame counterclockwise by an angle of phi (the gyroscope assembly tilting within the cube by an angle of phi). Combining the two transformation matrices, this gives the following transformation matrix from the ball frame to the fixed frame:

$$ {}^{B}_{F}R = \begin{bmatrix} cos(\phi)cos(\theta) & -sin(\phi) & -cos(\phi)sin(\theta) \\ sin(\theta)cos(\phi) & cos(\phi) & -sin(\phi)sin(\theta) \\ sin(\theta) & 0 & cos(\theta) \end{bmatrix}. $$

The acceleration of frame **B** will be a result of two things: constant downward gravitational force and acceleration due to the rotation of the cube / gyroscope assembly. The team is interested in obtaining an estimate for theta using the acceleration due to gravity. Assume that the acceleration is independent of the constant downward gravitational force and will occur at a higher range of frequencies than the acceleration due to the constant downward gravitational force (which is a DC phenomenon since it is constant). Since the system uses a complementary filter (lowpass on the accelerometer estimate) later, derive our angle estimate using only the force of gravity as follows.

The acceleration due to gravity of the ball with respect to the fixed frame, written in the coordinates of the fixed frame, is: $-g\,\widehat{K_F}$. Combining this with the transformation matrix, it is possible to find the acceleration due to gravity of the ball with respect to the fixed frame, written in the coordinates of the ball frame: $\begin{bmatrix} cos(\phi)sin(\theta) \\ sin(\phi)sin(\theta) \\ -cos(\theta) \end{bmatrix} g$.

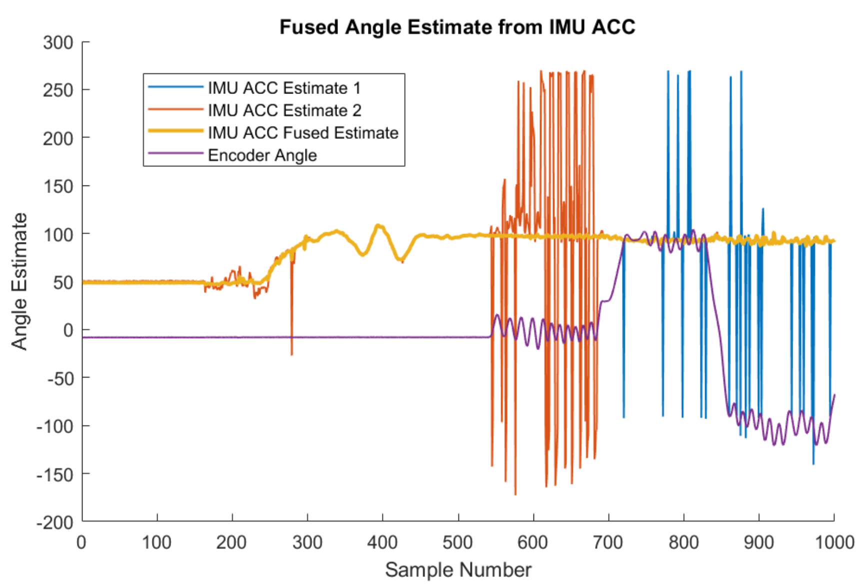
Equate this vector to the readings from the accelerometer, which provide the acceleration of the ball with respect to the fixed frame, written in the coordinates of the ball frame.

$$ acc_i = cos(\phi)sin(\theta)g $$
$$ acc_j = sin(\phi)sin(\theta)g $$
$$ acc_k = -cos(\theta)g $$

Note that in these equations, the left-hand side is measured by the accelerometer, phi is measured by the encoder, and theta is unknown. It is possible solve for theta in each equation as follows:

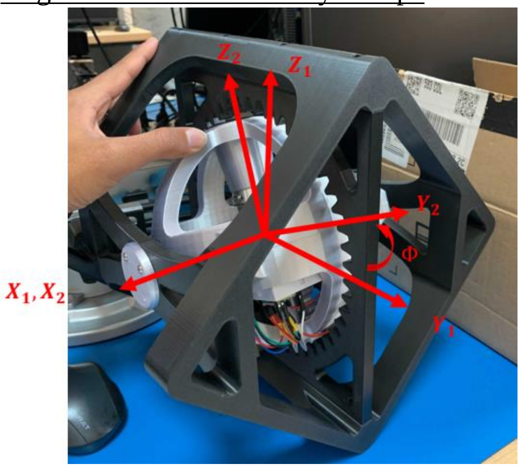
$$ \theta = atan2(acc_i, -cos(\phi)acc_k) $$
$$ \theta = atan2\left(acc_j, -sin(\phi)acc_k\right) $$

This was tested with the flywheel not spinning, while varying the angle of the gyroscope assembly and got the following results:



As shown in this figure, the two estimates track very closely to each other for the most part. When phi is close to 0 or close to 90, the denominator of the argument of the arctangent will become small, causing numerical instability issues in the estimate. So, the team decided to use the first equation when phi was less than 45 degrees, and the second equation when phi was greater than 45 degrees. This gave the results in yellow, which were close to the expected results.

Angle Estimate from the Gyroscope



The angular velocity estimation was simpler. In this figure, coordinate system 1 is aligned with the cube and coordinate system 2 is aligned with the ball. The X direction of these two coordinate systems is aligned along the rotational axis of the encoder. This allows the angle phi to be measured from the encoder.

The gyroscope gives the angular velocity of the ball frame with respect to the fixed frame, in the coordinates of the ball frame. The angular velocity of the ball frame with respect to the fixed frame, in the coordinates of the fixed frame is the desired output. Specifically, the rotation about the Y1 axis, which is aligned with the fixed frame (the cube tilts about this axis) is the interest. To get this value, we used the following equation:

$$\omega_{Y_1} = \omega_{Y_2} cos(\phi) - \omega_{Z_2} sin(\phi),$$

where ω_{Y2} and ω_{Z2} are given by the gyroscope and phi is obtained from the encoder.

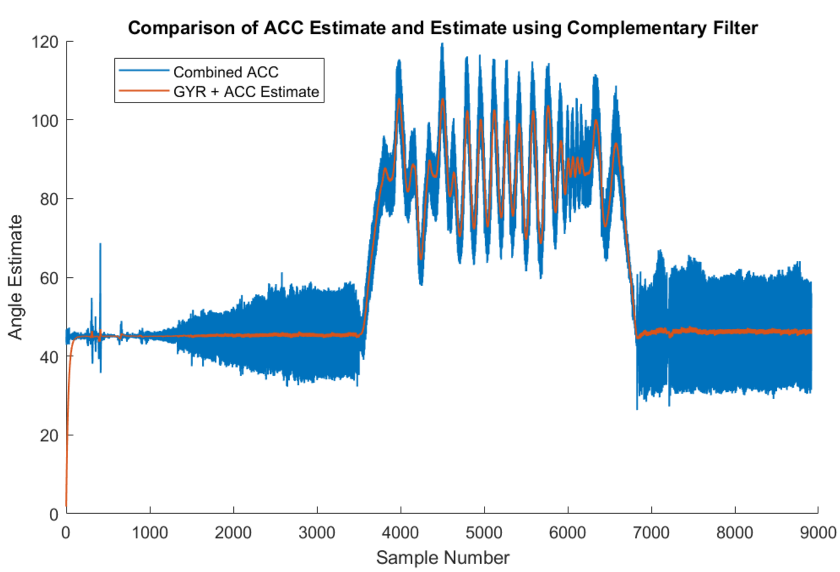
Overall Angle Estimation with Complementary Filter

Our equation for the estimate of theta during iteration n combining the gyroscope and accelerometer with the complementary filter is as follows:

$$\theta_n = \theta_{acc}\alpha + \left(\theta_{n-1} + \omega_{Y_1} * dt\right)\beta$$

Where θ_n is the estimate of theta during iteration n, θ_{n-1} is the previous estimate of theta, dt is the period of the control loop, θ_{acc} is the angle estimate from the accelerometer, ω_{Y_1} is the angular velocity estimate from the gyroscope, and α and β are the filtering constants, which we set to .02 and 0.98 respectively.

The estimate was tested while the flywheel was spinning (causing vibration due to the unbalanced nature of the flywheel) and got the following results:



As shown, the complementary filter helped significantly in reducing the vibration noise in the accelerometer estimate and still maintained some of the higher-frequency oscillations through the gyroscope data.

PID Control Law

The PID Control implementation was straightforward. The code is shown below:

```
error_integral = error_integral + e_theta * dt;
float p_contribution = Kp * e_theta; // From angle estimation
float i_contribution = Ki * error_integral; // From numeric integration
float d_contribution = Kd * cube_angle_vel; // From gyroscope
float moment = p_contribution + i_contribution + d_contribution;
```

The source for the e_theta is the difference between the fused angle estimate and the target angle. The source for cube_angle_vel is the gyroscope estimate (ω_{Y1}). The source for the error_integral term is just numerical integration of the previous error.

Conversion of Moment to Stepper Motor Speed

Since the controller outputs a desired moment, it is necessary to convert this moment to a desired stepper motor speed that can be commanded. The relationship between moment and speed comes from this equation:

$$M_z = I_{yy}(\alpha_{wheel} \sin(\phi) + \dot{\phi}\omega_{wheel} \cos(\phi))$$

Where α_{wheel} is zero, so the first term is 0 since the flywheel is spinning at a constant velocity. Iyy was estimated from CAD, ω_{wheel} was determined experimentally as shown in prior sections, and phi is obtained from the encoder angle. This solves for $\dot{\phi}$ (which the speed of the gyroscope assembly rotation) as a function of desired moment.

The code for this is shown below:

```
float motor_speed = moment / (IW * OMEGA_W * cos(ball_angle * M_PI / 180.0));
if(motor_speed > MAX_MOTOR_SPEED) {
  motor_speed = MAX_MOTOR_SPEED;
}else if(motor_speed < -MAX_MOTOR_SPEED) {
  motor_speed = -MAX_MOTOR_SPEED;
}
setStepperVel(motor_speed);
```

The motor speed is calculated and then is clamped to a specific range of values. Note a mistake that was made is that the target speed, $\dot{\phi}$, is the speed of the gyroscope assembly, and the team forgot to take into account the gearing of the stepper motor to the gyroscope assembly when performing this calculation. Fortunately, the PID tuning took care of this oversight.

Conclusion

The team was able to design and control an inherently unstable system of a cube balancing on its edge using a control moment gyroscope to produce a stabilizing torque. The PID controller used to stabilize the system was designed using root locus. The team was able to use the encoder IMU and encoder to achieve a stable angle reading while the system is rotating. The flywheel was slightly imbalanced about its rotational axis leading to significant vibrations. This led to better system performance on a soft surface such as carpet as opposed to tile or wood. For future iterations, the IMU could be placed on the frame in order to more easily read the angle of the cube, though the team was able to get a consistent angle reading with this internally mounted method. Finally, the gains produced by root locus design were not able to stabilize the system. In the future, the team would like to refine their model and use a more systematic approach to controller design such as LQR.

Acknowledgements

The team would like to thank Dr. Ellen Mazumdar, Roman Balak, and Timothy Brumfiel for their guidance on controller design.

References

[1] C. Gurrisi, R. Seidel, S. Dickerson, S. Didziulis, P. Frantz, and K. Fergurson, “Space Station Control Moment Gyroscope Lessons Learned ,” *ESA*, 2010. [Online]. Available: https://www.esmats.eu/amspapers/pastpapers/pdfs/2010/gurrisi.pdf. [Accessed: 25-Apr-2021].

[2] H. Vathsangam, "Complementary Filter," 29 May 2010. [Online]. Available: https://sites.google.com/site/myimuestimationexperience/filters/complementary-filter. [Accessed 25 April 2021].

ME4012 Grading Rubric for Final Project Report

Team/Project Name: ____

Team Members: ____

Metrics	**Pts**	**Comments**
<u>Design or Simulation Assumptions (5 pts)</u> *Is the design of the system well-described in the text? *What is the quality of the design, is it well thought out and executed well?		
<u>Model (30 pts)</u> *Is the model section well-written and easy to follow? Is the model error-free? *Does the model accurately capture the relevant dynamics? *Are the equations of motion correctly derived? *Does the team do a good job of estimating the parameters of their plant? *Is there a table of derived/estimated parameters?		
<u>Controller Design/Implementation (30 pts)</u> *Is the controller section well-written and easy to follow? *Does the controller design make sense? *Are they able to meet their desired design metric, achieve disturbance rejection, or achieve stability (for unstable systems)? *Does the team do a good job of implementing a working controller (Labview, Arduino, or Matlab)? *If the controller does not work, can they describe why and what they would change?		
<u>Experimental or Simulation Results (15 pts)</u> *Are the results easy to follow and logically written? *Does the performance of the system match expectations from theory (compare nonlinear simulation with linear model, or compare model with experiment)? If not, did the team explain why? *Are there graphs/figures to show performance?		
<u>Final Paper Quality (20 pts)</u> *Is the abstract/introduction/conclusion well written? *Are the figures captioned and readable? *Did the team make the figures/formatting look nice? *Are there references and are they correctly cited? *Is the paper error-free in terms of spelling and grammar? *Is the paper easy to read and easy to follow logically? *Did the team pick an appropriate level of difficulty for their team size? Where they able to implement what they set out to do? *Did the team finish construction and have a mostly working system? *Does the system work well enough for control?		