Oh, What a Bot!

HackRVA.org

Contents

List of Symbols

The following list describes several symbols that will be used later within the body of the document

Physics constants

g	Gravitational constant (m s^{-2})

Transfer Functions

$C(s)$	control function
$M(s)$	motor transfer function
$P(s)$	pendulum transfer function
$R(s)$	rotor transfer function
$T(s)$	open loop transfer function

Other symbols

η_g	gear efficiency
η_m	motor efficiency
ω	motor rotation speed (rad s^{-1})
ϕ	flywheel angle (rad)
θ	pendulum angle (rad)
A	coulomb friction coefficient (N m)
B	rotational friction coefficient(N m rad^{-1} s)
F	environmental forces on the pendulum
i	motor current (A)
I_c	moment of inertia of the pendulum (kg m^2)
I_f	moment of inertia of the reaction wheel (kg m^2)

K_ω	rotor velocity feedback gain
K_t	motor torque constant (N m A^{-1})
K_v	motor voltage constant (V rad^{-1} s)
L	armature inductance (H)
l	distance from the pivot point to the center of pendulum mass (m)
m	total mass of inverted pendulum (kg)
r	armature resistance (Ω)
T	total motor torque (N m)
$T_{friction}$	torque lost to motor and gear friction (N m)
T_{shaft}	torque available to the flywheel (N m)
u	controller output
v	voltage supplied to the motor (V)
x	net torque applied to the pendulum (N m)

1 Introduction

This article describes the development of a control system model for an reaction wheel pendulum, shown in Figure 1. A reaction wheel pendulum is a variation of a simple pendulum, balanced upright, with rotating wheels at the top in place of the pendulum bob. Each wheel is driven by a separate motor. The pendulum is inherently unstable in the upright position - gravity will pull the pendulum over if it is not perfectly balanced. Thus, a control system is needed to accelerate the reaction wheels, applying torques as needed to the pendulum so as to keep it upright. The reaction wheels are mounted at right angles to each other, providing the ability to correct the alignment of the pendulum in any direction.

The pendulum physical structure includes a rod, an accelerometer, and two flywheels. Each flywheel is comprised of a motor mount, a DC brushed motor, and a reaction wheel. The flywheel assemblies are mounted at the top of the rod at right angles to each other and the rod. The DC motors are wired to a motor controller and an Arduino. The Arduino is also wired to the accelerometer, which is located at the top of the rod.

needs a figure number

The basics of the pendulum control system is outlined in Figure **??**. The system can be controlled by adjusting the voltage. An additional input F is included to describe random environmental forces that may affect the pendulum. The random forces may result in the desired angle of the pendulum not being achieved. More accurate control will result from feeding the resulting pendulum angle back into the control system so that the input voltage is continuously adjusted until the desired pendulum angle is achieved. This is shown in Figure 2.

In this case, however, the goal is for the pendulum to remain vertical with as little wobble as possible. The real control comes into play when correcting for the stray forces on the system. Hence, the control system can be reformulated as shown in Figure 3, commonly known as a Regulator Problem. This structure emphasizes the environmental forces as the primary input into the system and that the feedback is primarily intended to compensate for those forces.

Figure 3 is missing a feedback arrow

Figure 1: Sketch of the Pendulum

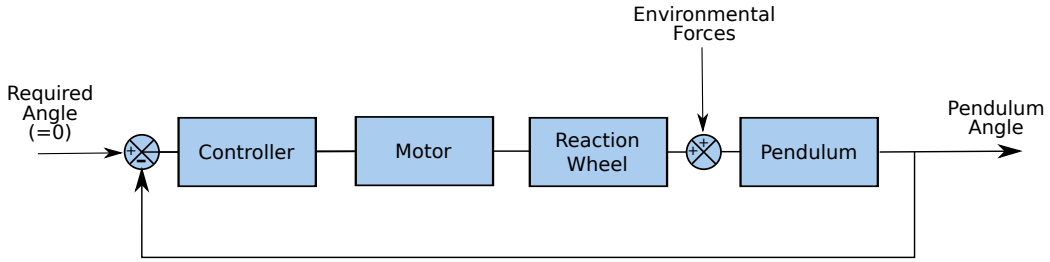


Figure 2: Simplified Pendulum Model

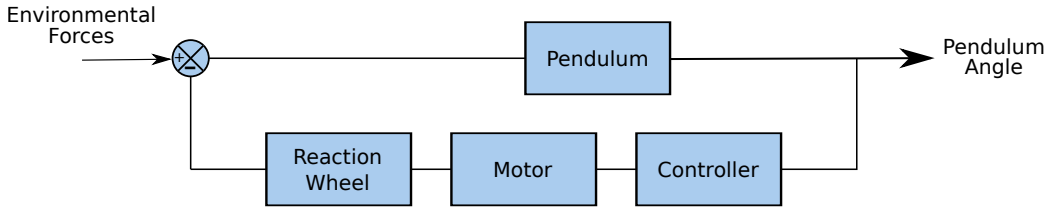


Figure 3: Regulator Model

2 Pendulum Dynamics

More detail can be added to the regulator control system description by building a mathematical model of the the pendulum. Figure 4 shows a simplified sketch of the inverted pendulum. Reference [2] contains a summary of the model dynamics for the balancing pencil. The model dynamics can be summarized as

$$mgl\sin\theta + F - I_c\ddot{\theta} = I_f\ddot{\phi} \tag{1}$$

Other references may have alternate formulations using different definitions for θ and ϕ.[1]

After linearizing the equation by noting that $\sin\theta \approx \theta$ for small angles, the transfer function for the pendulum is then

$$m\,g\,l\theta + F - I_c\ddot{\theta} = I_f\ddot{\phi} \tag{2}$$

The reaction wheel can be continuously revolving, making the reaction wheel angle ϕ less useful as a control parameter. In its place we will use $\omega = \dot{\phi}$.

$$m\,g\,l\theta + F - I_c\ddot{\theta} = I_f\dot{\omega} \tag{3}$$

For ease of use in later analysis, it is beneficial to split (3) into two parts:

$$m\,g\,l\theta + x(t) = I_c\ddot{\theta} \tag{4}$$

$$x(t) = F - I_f\dot{\omega} \tag{5}$$

where $x(t)$ is the net torque applied to the pendulum. The sign convention is important; a positive torque $x(t)$ will accelerate the pendulum in the direction of the positive pendulum angle. However, the sign convention in (5) indicates that a positive acceleration of the rotor will result in a *negative* acceleration of the pendulum.
Note that if the pendulum is not vertical (i.e. $\theta = 0$) gravity will begin to pull the pendulum over. Correcting this requires accelerating the flywheel

[1] Reference [1] has a slightly different formulation for the model dynamics; the sign of the gravitational term is positive rather than negative. This difference is caused by Reference [1] defining the angle from the downward, resting position while Reference [2] defines the angle from the vertical. The cosines of angles differing by π have opposite signs.

to apply torque to the pendulum. A flywheel turning at a constant velocity applies no torque to the pendulum and will not alter the movement of the pendulum.

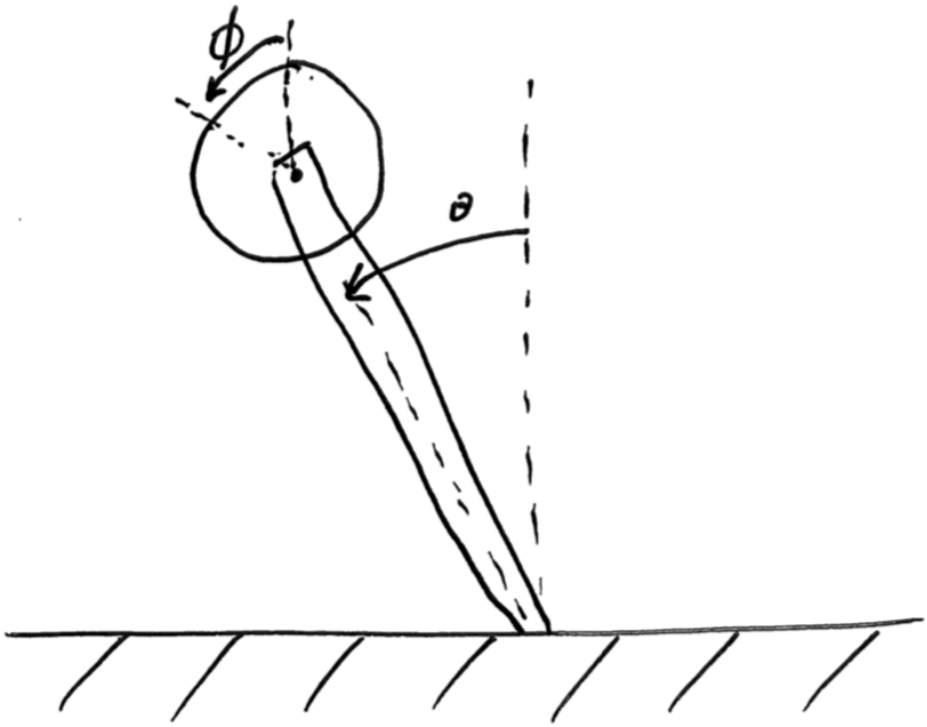


Figure 4: Simplified Pendulum Model

3 Motor and Flywheel Dynamics

Reference [2] models the motor as

$$I_f \, \ddot{\phi} \, \eta_m \, \eta_g = K_t \, i \tag{6}$$

This formulation appears to have problems, as a reduction in the motor and gear efficiencies results in a greater flywheel acceleration for a given motor current. Similarly, a gear ratio above unity would decrease the flywheel acceleration for a specified motor current/torque, which is non-physical. The equation can be altered to correct these issues, resulting in the following:

$$I_f \, \ddot{\phi} = \eta_m \, \eta_g K_t \, i \tag{7}$$

However, this format may not be the easiest to implement, as the motor and gear efficiencies are not known. Reference [1] takes a different approach,[2]

$$I_f \, \ddot{\phi} = T_{shaft} \tag{8}$$

Where the earlier format dealt with motor/gear efficiencies using η_m and η_g, Reference [1] develops a correlation of friction in terms of torque required to maintain a given $\dot{\phi}$. Here the torque will be reformulated as

$$T_{shaft} = T - T_{friction} \tag{9}$$

$$T_{friction} = A \, sgn(\dot{\phi} - \dot{\theta}) + B(\dot{\phi} - \dot{\theta}) \tag{10}$$

$$T = K_t \, i \tag{11}$$

Note the use of $(\dot{\phi} - \dot{\theta})$. Friction is dependent on the relative motion of the motor armature and the motor frame. If the pendulum angular velocity $\dot{\phi}$ is equal to the reaction wheel angular velocity $\dot{\theta}$ there will be no relative motion between the motor armature and frame; no friction will occur.
Combining equations,

$$I_f \, \ddot{\phi} = K_t \, i - A \, sgn(\dot{\phi} - \dot{\theta}) - B(\dot{\phi} - \dot{\theta}) \tag{12}$$

Again, using the notation $\omega = \dot{\phi}$,

$$I_f \, \dot{\omega} = K_t \, i - A \, sgn(\omega - \dot{\theta}) - B(\omega - \dot{\theta}) \tag{13}$$

As will be seen, this equation is easier to implement because K_t, A, and B can be measured experimentally (see Appendix C).

[2] Note that there is no term in (8) that represents torque associated with external work done by the shaft. Hence this formulation is limited to flywheels that are reasonably aerodynamic (e.g. not something that acts as a propeller).

4 Motor Control

The previous chapter determined the relationship between the motor current and the acceleration of the flywheel. However, the motor will be not be controlled via motor current. Instead, a motor controller will be used to control the motor by adjusting the motor voltage via a PWM signal. The equivalent, average voltage will then regulate the motor. Following Reference [1], but recognizing the motor is at the end of a pendulum, the relationship between motor current and voltage is

$$L\,\frac{di}{dt} + r\,i = v - K_v\,(\omega - \dot{\theta}) \tag{14}$$

Normally the inductance of the motor is much lower than the resistance, such that $L/r \sim 0.001$ sec. In such cases it is acceptable to ignore the time dependence of the current and write

$$r\,i = v - K_v\,(\omega - \dot{\theta}) \tag{15}$$

Solving for i:

$$i = \frac{v}{r} - \frac{K_v\,(\omega - \dot{\theta})}{r} \tag{16}$$

Combining with (13)

$$I_f\,\dot{\omega} = K_t\left(\frac{v}{r} - \frac{K_v\,(\omega - \dot{\theta})}{r}\right) - A\,sgn(\omega - \dot{\theta}) - B(\omega - \dot{\theta}) \tag{17}$$

Rearranging,

$$I_f\,\dot{\omega} + \left(B + \frac{K_t K_v}{r}\right)(\omega - \dot{\theta}) + A\,sgn(\omega - \dot{\theta}) = \left(\frac{K_t}{r}\right)v \tag{18}$$

Not all of the voltage results in acceleration of the reaction wheel. As the reaction wheel velocity increases, an increasing amount of the voltage (and the resulting torque) is consumed by friction and the back EMF.

The voltage used to drive the motor is determined by the controller. Here the voltage will be set to the output of the controller

$$v = u \tag{19}$$

Substituting into (18)

$$I_f\,\dot{\omega} + \left(B + \frac{K_v K_t}{r}\right)(\omega - \dot{\theta}) + A\,sgn(\omega - \dot{\theta}) = \left(\frac{K_t}{r}\right) u \tag{20}$$

The simple substitution of u for v does not seem to have much purpose at this point in the analysis, but will be useful when complex controllers are evaluated.

It is useful to assume that the control function can adjust the demand voltage by adding or subtracting a constant, depending on the current spin direction of the reaction wheel. This means that $-A\,sgn(\dot{\phi})$ is folded into u for the purposes of this analysis. Our motor control equation is then

$$I_f\,\dot{\omega} + \left(B + \frac{K_v K_v}{r}\right)(\omega - \dot{\theta}) = \left(\frac{K_t}{r}\right) u \tag{21}$$

5 Modeling in the Frequency Domain

The primary objectives of a control system for the pendulum should be to stabilize the upright position of the pendulum and recovering from external forces on the pendulum. In addition, the control system should be designed to deal with a number of physical limitations of the physical components of the pendulum. Specifically,

- the motor voltage is limited to maximum value,
- the rotor velocity has a practical upper limit.

The stability of the system shown in Figure 3 can be evaluated by examining the open loop transfer function

$$T(s) = P(s)C(s)M(s)R(s) \tag{22}$$

The control system in terms of these transfer functions is shown in Figure 5.

The controller, at this point in the analysis, is assumed to be a pass-through, or

$$u = \theta \tag{23}$$

Hence, the controller transfer function is

$$C(s) = \frac{u(s)}{\theta(s)} = 1 \tag{24}$$

The pendulum transfer function can be derived from (4)

$$m\,g\,l\theta(s) + x(s) = I_c\theta(s)s^2 \tag{25}$$

Ignoring the environmental forces, the pendulum transfer function is

$$P(s) = \frac{\theta(s)}{x(s)} = \frac{1}{I_c s^2 - mgl} \tag{26}$$

The rotor transfer function can be derived from (5) by noting that the torque generated by the flywheel is

$$x(s) = F(s) - I_f\omega(s)s \tag{27}$$

$$R(s) = \frac{x(s)}{\omega(s)} = -I_f s \tag{28}$$

Note that the transfer function for the rotor in (28) is negative. Combined with the summing junction shown in Figure 5, this produces a negative feedback control system.

The motor transfer function must be derived from (21). However, in (21) ω is a function of both θ and u, preventing the determination of a single-input, single-output transfer function. One solution is to make the approximation $\omega \approx (\omega - \dot{\theta})$. The reaction wheel may rotate at hundreds of RPM, while the pendulum is expected to deviate from vertical for only brief periods. With this approximation the motor transfer function can be derived from (21) as

$$I_f\,\omega(s)s + \left(B + \frac{K_v K_t}{r}\right)\omega(s) = \left(\frac{K_t}{r}\right)u(s) \tag{29}$$

$$M(s) = \frac{\omega(s)}{u(s)} = \frac{\left(\frac{K_t}{r}\right)}{I_f s + (\frac{K_v K_t}{r} + B)} \tag{30}$$

Using (24), (26), (28), and (30) results in

$$T(s) = \frac{-(\frac{1}{I_c})(\frac{K_t}{r})s}{(s^2 - \frac{mgl}{I_c})(s + (\frac{K_v K_t}{r I_f} + \frac{B}{I_f}))} \tag{31}$$

Substituting in values from the appendixes, the transfer function for the pendulum is

$$T(s) = \frac{-0.1731s}{s^3 + 3.113s^2 - 34.8s - 108.3} \tag{32}$$

The numerator of the transfer function is the characteristic equation that, when factored, is $(s - 5.899)(s + 5.899)(s + 3.113)$.
Using the root locus technique and the open loop transfer function, Figure 6 shows the behavior of the closed loop poles as a function of feedback gain. Note that one pole is always in the right half of the plane, indicating that the system is unstable regardless of the magnitude of the gain.

The system might be stabilized by using both proportional and derivative, or PD, feedback. The control function would be:

$$u = \theta + p\dot{\theta} \tag{33}$$

thus

$$C(s) = 1 + ps \tag{34}$$

Rearranging,

$$C(s) = p\left(s + \frac{1}{p}\right) \tag{35}$$

indicating a zero at $1/p$ and a gain of p. The results are sensitive to the placement of the zero. Placing the zero to the right of the pole at -5.8993 eliminates any oscillation but does not remediate the pole in the right half of the plane. Placing the zero to the left of the pole at -5.8993 results in a much faster response but can result in oscillations (Figure 7). Again, the pole in the right half of the plane is not remediated.

One additional possibility for resolving the issue of the zero at the origin would be to implement a PID controller.

$$u(t) = \theta(t) + q\int \theta\, dt + p\dot{\theta}$$

then

$$C(s) = 1 + \frac{q}{s} + ps \tag{36}$$

Rearranging,

$$C(s) = p\left(\frac{s^2 + \frac{s}{p} + \frac{q}{p}}{s}\right) \tag{37}$$

This controller would have two zeros that can be placed as required and a pole that would cancel the zero at the origin. If the zeros are placed to the left of the pole at -4.7303, then the system may stabilize. However, in the real world the original zero may not be exactly at the origin and the poles in the left side of the axis may not be exactly where they are expected (due to nonlinearities, etc.) leading to a less-than robust control system.

verify this

Finally, it is possible to include feedback from the other monitored and controlled variable, the rotor velocity. Unfortunately, to continue to evaluate the system using a root locus analysis, the feedback from the rotor velocity cannot be included directly in $C(s)$, as it would introduce a second input into the transfer function. Instead, the definition of the motor voltage v is modified to include the effect of the rotor velocity feedback

$$v = u + K_\omega \omega \tag{38}$$

where u continues to represent the output of a PID controller using θ as an input and $K_\omega\omega$ is an additional feedback from the rotor velocity. This results in (21) transforming into

$$I_f\,\dot{\omega} + \left(B + \frac{K_v^2}{r}\right)\omega + A\,sgn(\omega) = \left(\frac{K_v}{r}\right)(u + K_\omega\omega) \tag{39}$$

The motor transfer function is then

$$I_f\,\omega(s)s + \left(B + \frac{K_v^2}{r}\right)\omega(s) = \left(\frac{K_v}{r}\right)(u(s) + K_\omega\omega(s)) \tag{40}$$

Rearranging terms,

$$M(s) = \frac{\omega(s)}{u(s)} = \frac{\left(\frac{K_v}{I_f r}\right)}{s + (\frac{1}{I_f})(\frac{K_v^2}{r} - \frac{K_v K_\omega}{r} + B)} \tag{41}$$

Using (35), (26), (28), and (41), the open loop transfer function is now

$$T(s) = \frac{-(\frac{Kp}{I_c r})(s + \frac{1}{p})s}{s^3 + (\frac{Br - KK_\omega + K^2}{I_f r})s^2 - (\frac{mgl}{I_c})s - (\frac{Br - KK_\omega + K^2}{I_f r} - \frac{mgl}{I_c})} \tag{42}$$

Figure 8 shows the root locus with $p = 1/4.5$ and $K_\omega = 0.075$. With a controller gain[3] of 370 the controlling close loop poles are at -11.3, -1.33, and -1.12.

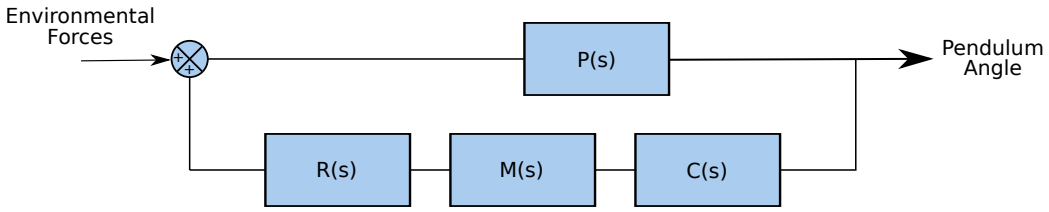


Figure 5: Control System Modeled as Transfer Functions

[3]The gain given here is in addition to the system gain modeled in (42).

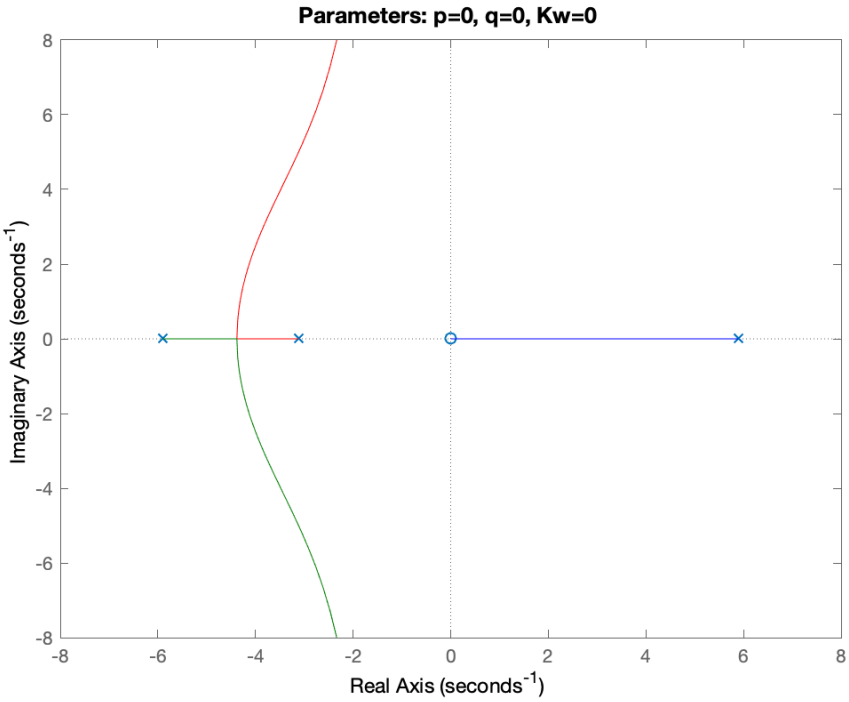


Figure 6: Root Locus Plot With Proportional Feedback

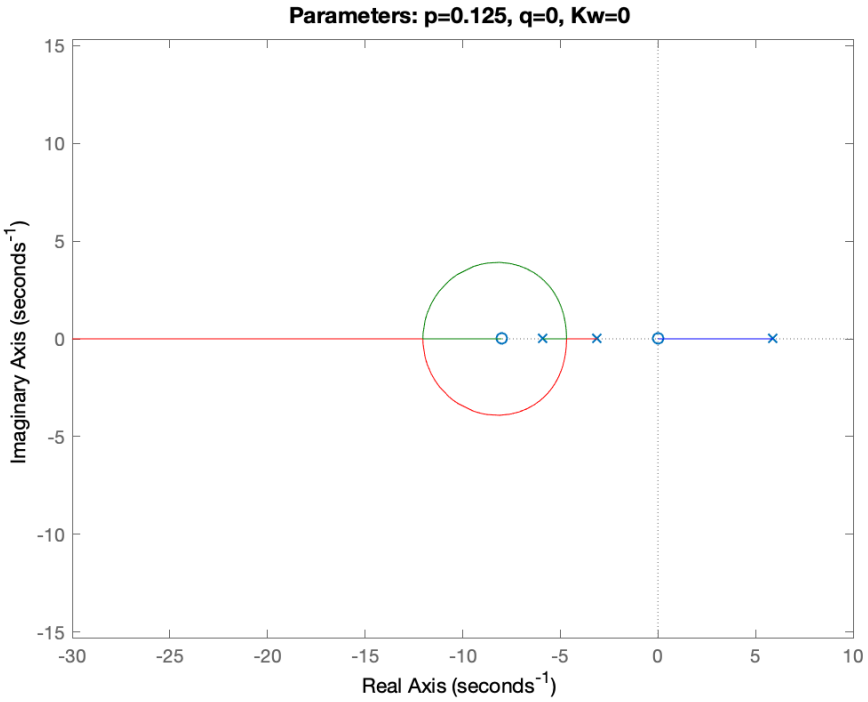


Figure 7: Root Locus Plot With PD Feedback

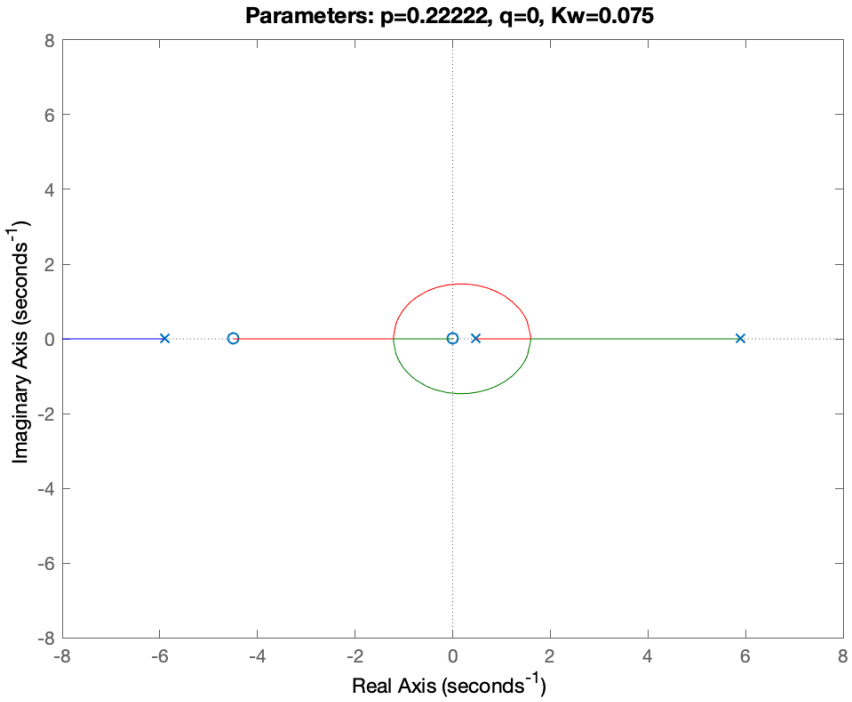


Figure 8: Root Locus Plot With PD and Rotor Velocity Feedback

6 State Space Model

A State Space model can be constructed from equations (18) and (3). The state space variables are ω, θ, and $\dot{\theta}$. The state space model is then

$$\dot{x_m} = A_m x + B_m v \tag{43}$$

where $x_m = (\omega, \theta, \dot{\theta})^T$ and

$$A_m = \begin{bmatrix} -\frac{B}{I_f} - \frac{K_v K t}{I_f r} & 0 & \frac{B}{I_f} + \frac{K_v K t}{I_f r} \\ 0 & 0 & 1 \\ \frac{B}{I_c} + \frac{K_v K t}{I_c r} & \frac{mgl}{I_c} & -\frac{B}{I_c} - \frac{K_v K t}{I_c r} \end{bmatrix} \tag{44}$$

$$B_m = \begin{bmatrix} \frac{K_t}{I_f r} \\ 0 \\ -\frac{K_t}{I_c r} \end{bmatrix} \tag{45}$$

However, measured sensor data from motor encoder signals are proportional to the relative velocity of the reaction wheel, $\omega_r = \omega - \dot{\theta}$. Changing the state space variables to $x = (\omega_r, \theta, \dot{\theta})^T$ is performed using a transformation matrix $x = P x_m$ where

$$P = \begin{bmatrix} 1 & 0 & -1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{46}$$

From Reference [6] the new state space model is then

$$\dot{x} = Ax + Bv \tag{47}$$

$$y = Cx + Dv \tag{48}$$

where $x = (\omega_r, \theta, \dot{\theta})^T$, $y = (\omega_r, \theta)^T$

$$A = P A_m P^{-1} \tag{49}$$

$$B = P B_m \tag{50}$$

resulting in

$$A = \begin{bmatrix} -\frac{B}{I_f} - \frac{B}{I_c} - \frac{K_v K t}{I_f r} - \frac{K_v K t}{I_c r} & -\frac{mgl}{I_c} & 0 \\ 0 & 0 & 1 \\ \frac{B}{I_c} + \frac{K_v K t}{I_c r} & \frac{mgl}{I_c} & 0 \end{bmatrix} \tag{51}$$

$$B = \begin{bmatrix} \frac{K_t}{I_f r} + \frac{K_t}{I_c r} \\ 0 \\ -\frac{K_t}{I_c r} \end{bmatrix} \tag{52}$$

$$C = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \tag{53}$$

$$D = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{54}$$

7 Implementation Details

In section 5 the necessary constants for the control system were developed. However those constants need to be adjusted for the reaction wheel pendulum.
The constants developed in section 5 took the form

$$C(s) = G(1 + \frac{q}{s} + ps) \tag{55}$$

where G is the controller gain, q is the integral constant, and p is the derivative constant. However the software requires constants in the form of

- the proportional coefficient K_p
- the integral coefficient K_i
- the derivative coefficient K_d
- the rotor velocity coefficient K_s

K_s is simply K_ω. To generate the other required constants, the controller equation is first restructured as

$$C(s) = G + \frac{Gq}{s} + Gps \tag{56}$$

From this it can be seen that $K_i = Gq$ and $K_d = Gp$. The proportional constant requires a little more work. Recall from (42) the system has a gain of $Kp/I_c r$. All of these terms, with the exception of p are a inherent to the hardware; p is the derivative constant. K_p is then

$$K_p = \frac{G}{p} \tag{57}$$

Another item to note in the control system implementation is that the feedback should be *positive* as shown in Figure 5. This reflects the fact that the rotor transfer function in (28) is negative. This can also be seen in Figure 4, where accelerating the rotor in the direction of positive θ results in a negative change in ϕ.

Finally, units matter. This documentation was created with SI units, eliminating the need to incorporate a multitude of conversion factors. However, software is frequently constructed in more common units, for example rotor velocity in RPM, angles in degrees, and time in milliseconds or microseconds. Hence numerical values developed in this document may need to be converted to other units prior to use.

References

[1] Block, D. J., Astrom, K. J., and Spong, M. W. (2007). *The Reaction Wheel Pendulum*. Morgan & Claypool Publishers.

[2] Ramm, A., and Sjostedt, M. (2015). Reaction wheel balanced robot, Design and Sensor Analysis of Inverted Pendulum Robot. Technical Report, KTH Royal Institute of Technology.

[3] Hellman, H. and Sunnerman, H, (2015). Two-Wheeled Self-Balancing Robot. Technical report, KTH Royal Institute of Technology.

[4] Chauveau, G., Chazal, D., Nakayama, D., Olsen, E., and Palm, S., (2005). Controlling the Reaction Wheel Pendulum.

[5] "Stick Balances Itself With Reaction Wheels" by Gerrit Coetzee

[6] Iqbal, Kamran (2020). *Introduction to Control Systems*.

Appendices

Appendix A Motor and Gear Set Data

Key to much of the work so far are the constants associated with the motor and gear set, K_t, K_v, R, and L. The following work is based on a Pololu 4.4:1 metal gear motor.
Pololu gives the following parameters for the 4.4:1 gear motor:
weight = 95g
gear ratio = 4.4:1
No-load speed @ 12V = 1700 rpm (178 rad/sec)
No-load current @ 12V = 0.2 A
Stall current @ 12V = 2.1 A
Stall torque @ 12V (at the gearbox shaft) = 11 oz in (0.07768 Nm)
Encoder frequency = 211.2 counts/rev of gearbox shaft[4]

[4]The encoder has a frequency of 48 counts per revolution when counting both edges of both channels. With a gear ratio of 4.4:1, the encoder frequency relative to the shaft rotation is 48*4.4 counts/rev, or 211.2 counts per shaft revolution when counting both

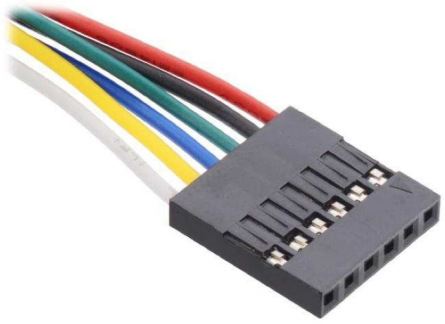
A number of the motor constants can be calculated from the published Pololu parameters. However, it should be noted that, for inexpensive motors, the actual motor parameters may vary from the published data. In any event, the motor constants will be calculated here and then later compared with measured data.

K_t can be calculated directly from the stall current and torque: K_t = 0.07768 Nm / 2.1 A = 0.0370 Nm/A. K_v can be set equal to K_t, or 0.0370 Vs.

The motor winding resistance can be determined from the rated voltage and the stall current, as no back-EMF is occurring at stall conditions. Thus, R = 12 V / 2.1 A = 5.7 ohms.

Motor wiring is effected via the provided 0.1"-pitch connector.

Lead Color	Function
Red	Motor power
Black	Motor power
Green	Encoder ground
Blue	Encoder Vcc (3.5 V to 20 V)
Yellow	Encoder A output
White	Encoder B output



edges of both channels.

Appendix B Pendulum Constants

A multitude of other values were calculated in MATLAB.

needs values for both rotors and move after rotor description

m	0.517327 kg
l	0.319038 m
I_c	0.046512 kg m^2
I_f	0.000168 kg m^2
g	9.81 m s^{-2}

Appendix C Measuring Motor Constants

Measurements were made to determine a number of motor constants, including A, B, r, l, and K_v, remembering that $K_t = K_v$ provided that SI units are used.[5] Not all of these motor constants can be directly measured. The resistance and inductance of the motor windings can be measured with an LCR meter. Motor voltage, current, and RPM can be measured by connecting the motor to a power supply and varying the voltage. The RPM can be measured either with a microcontroller monitoring the encoder or using a tachometer.

$\frac{A}{K_t}$ and $\frac{B}{K_t}$ can be determined from (13) by ensuring that the motor speed is constant. Then

$$i = \frac{A}{K_t} sgn(\omega) + \frac{B}{K_t}\omega \tag{58}$$

Measuring I at various values of ω permits estimating both $\frac{A}{K_t}$ and $\frac{B}{K_t}$ by linear regression. Optimally this should be performed using the rotor mounted on the motor. This ensures that the weight of the rotor is accurately accounted for in the gearbox friction.

The next step is to determine values for K_v (and hence K_t) and R. Substituting (13) into (15) and assuming a constant velocity we get

$$v = r\frac{A}{K_t} sgn(\omega) + (r\frac{B}{K_t} + K_v)\omega \tag{59}$$

Knowing $\frac{A}{K_t}$ and $\frac{B}{K_t}$ from the earlier regression, r and K_v can be determined by a linear regression of v as a function of K_v. Setting K_t equal to K_v,

[5] Provided SI units are used, K_v and K_t will have the same magnitude. This can be seen by equating the mechanical and electrical power of a motor:

$$T\omega = vi$$

where T is the motor torque. Rearranging,

$$\frac{T}{i} = \frac{v}{\omega}$$

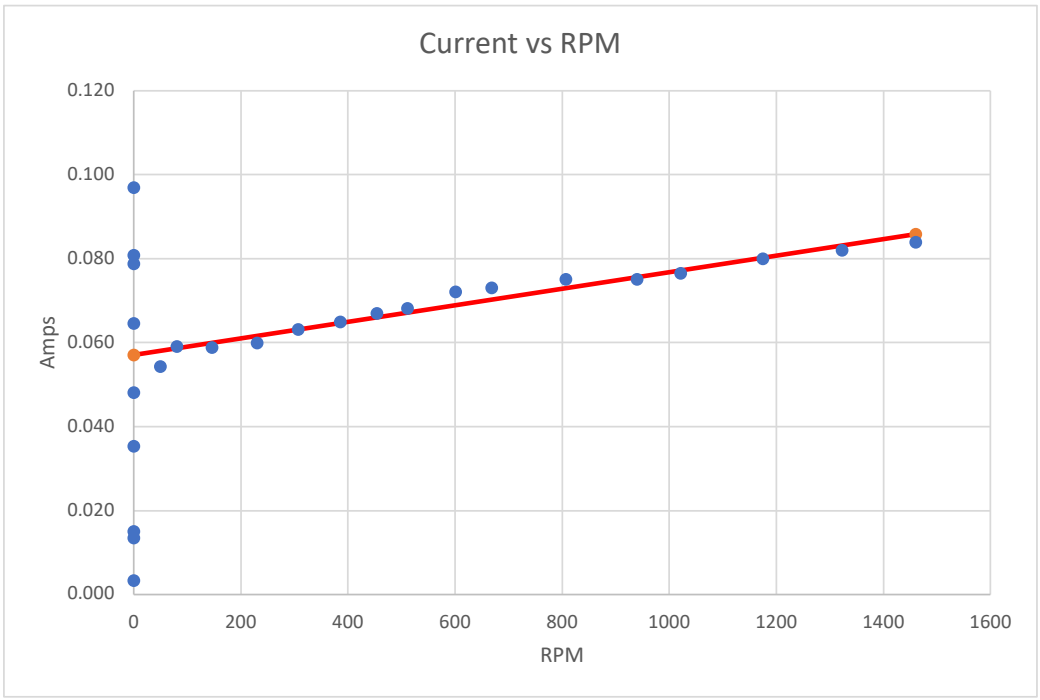
or

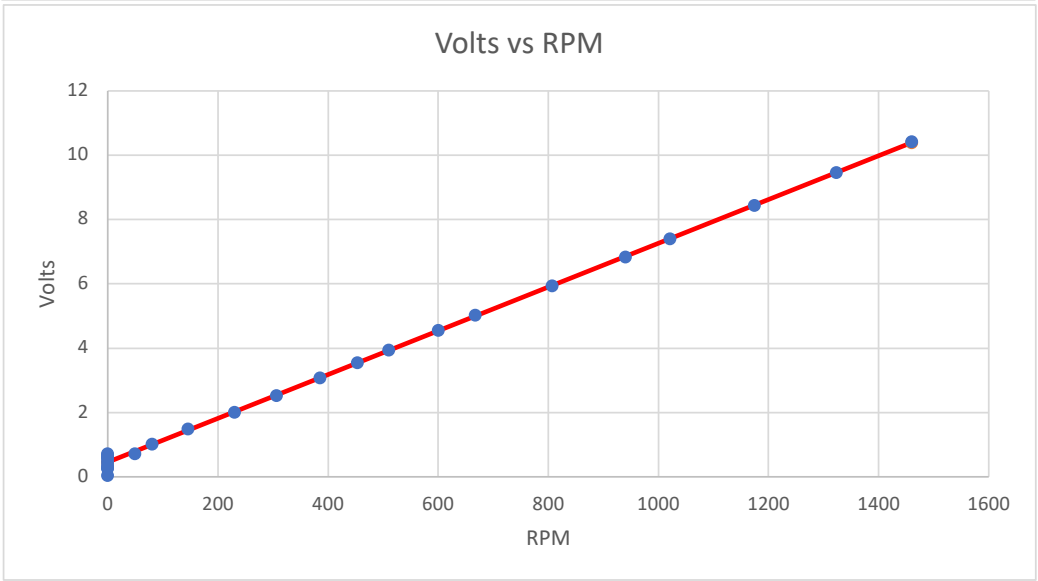
$$K_t = K_v$$

values for A and B can then be resolved.

Measurements of i, v, and ω were performed with the motor driven by a 12 V power supply. Current measurements were made with a a μCurrent DVM adapter and voltage measurements were performed at the motor input leads. Rotor angular velocity was measured with a hand-held tachometer. The following table list the resulting measurements:

Case	v (V)	i (A)	ω (RPM)	ω (rad/s)
1	0.0438	0.0034	0.	0.
2	0.286	0.015	0.	0.
3	0.601	0.0135	0.	0.
4	0.681	0.0788	0.	0.
5	1.02	0.0591	80.4	8.42
6	1.487	0.0589	146.	15.29
7	2.012	0.0599	230.	24.09
8	2.532	0.0631	307.	32.15
9	3.086	0.065	386.	40.42
10	3.549	0.067	454.	47.54
11	3.948	0.0682	511.	53.51
12	4.56	0.0721	601.	62.94
13	5.03	0.073	668.	69.95
14	5.94	0.0751	807.	84.51
15	6.84	0.0751	940.	98.44
16	7.4	0.0765	1021.	106.92
17	8.44	0.08	1175.	123.05
18	9.46	0.082	1323.	138.54
19	10.42	0.0839	1460.	152.89
20	0.2618	0.0354	0.	0.
21	0.3561	0.0481	0.	0.
22	0.476	0.0646	0.	0.
23	0.597	0.0808	0.	0.
24	0.718	0.0969	0.	0.
25	0.722	0.0543	49.6	5.19





Linear regression analysis results in the following values for the motor constants:
$A = 0.00363$ N m
$B =$ 1.2E-5 N m/(radian/sec)

$r = 7.9\ \Omega$
$K_v = 0.0636$ V sec

The coulomb friction represents approximately 5% of the motor stall torque. At the motor's no-load speed the viscous friction would double the total friction to approximately 7% of the motor stall torque. The viscous friction coefficient is significantly larger than the value of 9.06E-7 N m/(radian/sec) measured in Reference [4]. This may be due to the gearbox associated with the motor used in the current work. In any event, the friction will be included in the system modeling.

l was measured with the same instrument as 6.004 mH at 1kHz. This gives a motor electrical time constant l/r of 7.6E-4 sec. This is the time taken by the current in the motor to go from rest to 63% of the final steady-state current. This validates the assumption used in (15) .

Appendix D Arduino Code for Measuring RPM

The following is an Arduino sketch can be used to monitor the speed of a dc motor in lieu of a tachometer. The sketch is designed to work with either an Arduino Uno or an Arduino Mega. The motor wiring described in the comments referrs to the wiring of the Pololu 25D 4.4:1 metal gear motor. The output is in position change per second, where the motor has 211.2 counts/rev of gearbox shaft.

```
#include <Encoder.h>

// 25D motor wiring
// red - motor power
// black - motor power
// green - encoder GND
// blue - encoder VCC
// yellow - encoder A output
// white - encoder B output

/*
 * Encoder library
 * #include <Encoder.h>
 *
 * Change these two numbers to the pins connected to your encoder.
 *   Best Performance: both pins have interrupt capability
 *   Good Performance: only the first pin has interrupt capability
 *   Low Performance:  neither pin has interrupt capability
 * Encoder myEnc(5, 6);
 *   avoid using pins with LEDs attached
 * Interrupt Enabled Pins:
 * Arduino Uno: 2, 3
 * Mega, Mega2560, MegaAdK: 2,3,18,19,20,21
 */


 /*
```

```
  * Motor wiring
  * red - motor power
  * black - motor power
  * green - encoder GND
  * blue - encoder Vcc (3.5 - 20V)
  * Yellow - encoder A output
  * White - encoder B output
  */

/*
 * Hence, the connections
 * motor red - power supply
 * motor black - power supply
 * motor green - gnd pin
 * motor blue - 5V
 * motor white - pin 2
 * motor yellow - pin 3
 */



Encoder myEnc(2, 3);
long newposition = 0;
long oldposition = 0;
unsigned long newtime= 0;
unsigned long oldtime = 0;
long vel;

void setup()
{
  Serial.begin(115200);
  Serial.println("DC motor encoder RPM");
}

void loop()
{
  getVelocity();
  //Serial.print("M1 new: ");
```

```
  //Serial.println(newposition);
  //Serial.print("M1 old: ");
  //Serial.println(oldposition);
  //Serial.print(" delta T:");
  //Serial.println(newtime-oldtime);
  Serial.print("M1 speed: ");
  Serial.println(vel);
  delay(1000);
}

void getVelocity()
{
  oldposition = newposition;
  oldtime = newtime;
  newposition = myEnc.read();
  newtime = millis();
  vel = (newposition-oldposition) * 1000 /(newtime-oldtime);
}
```

Appendix E Reaction Wheel Design Options

The initial reaction wheel design was modeled after Reference [5], with modifications to allow the reaction wheel to mate properly with a Pololu motor hub. The reaction wheel is shown in Figure 9. The reaction wheel body is Aluminum 6061 with an outer diameter of 90 mm, a width of 18 mm, and an annular thickness of 7 mm. Fusion 360 reports the mass of the reaction wheel to be 0.09391 kg (assuming a density of 2.7 g/cc). The moment of inertia around the centerline is 1.562E-4 kg/m2.

An alternate, simpler design (although less elegant) is shown in Figure 10. Fashioned from carbon steel with a density of 7.85 g/cc, the diameter of the wheel is 90 mm and the thickness is 3.175 mm (1/8"). The reaction wheel mass is 0.175 kg and the moment of inertia around the centerline is 1.605E-04 kg/m2. The simpler design with only a small change in the moment of inertia permits an easier fabrication at a cost of a doubling of the reaction wheel mass.

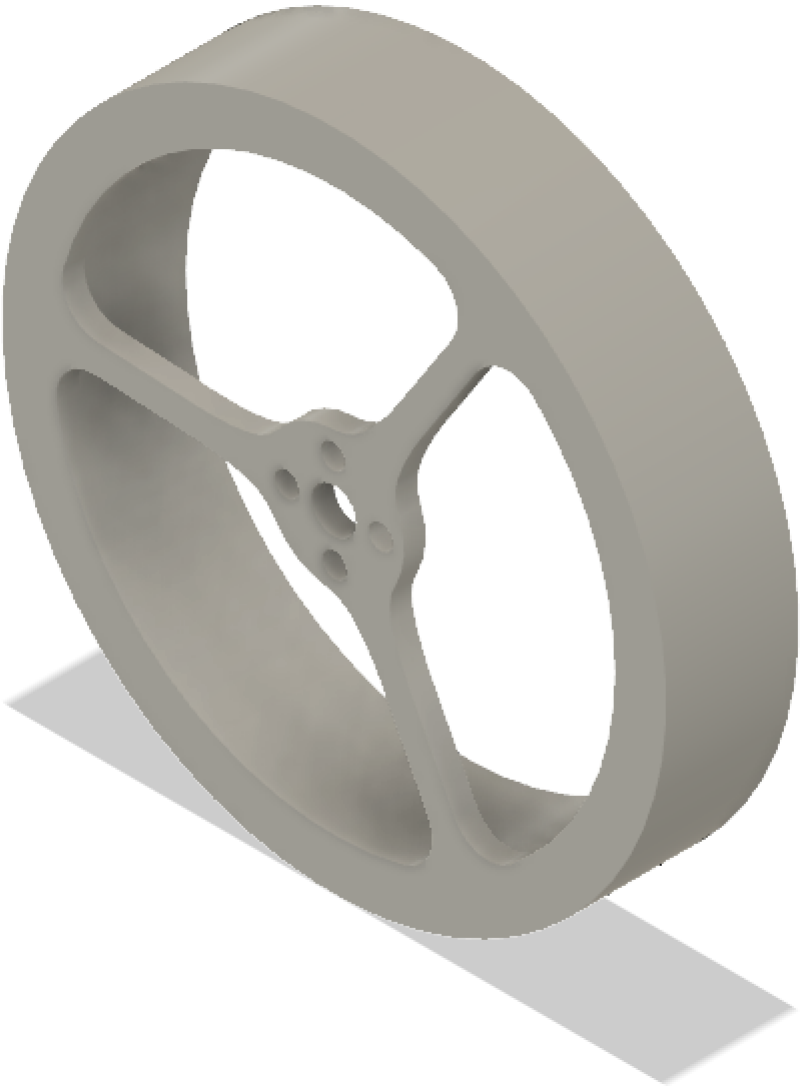


Figure 9: Aluminum Reaction Wheel

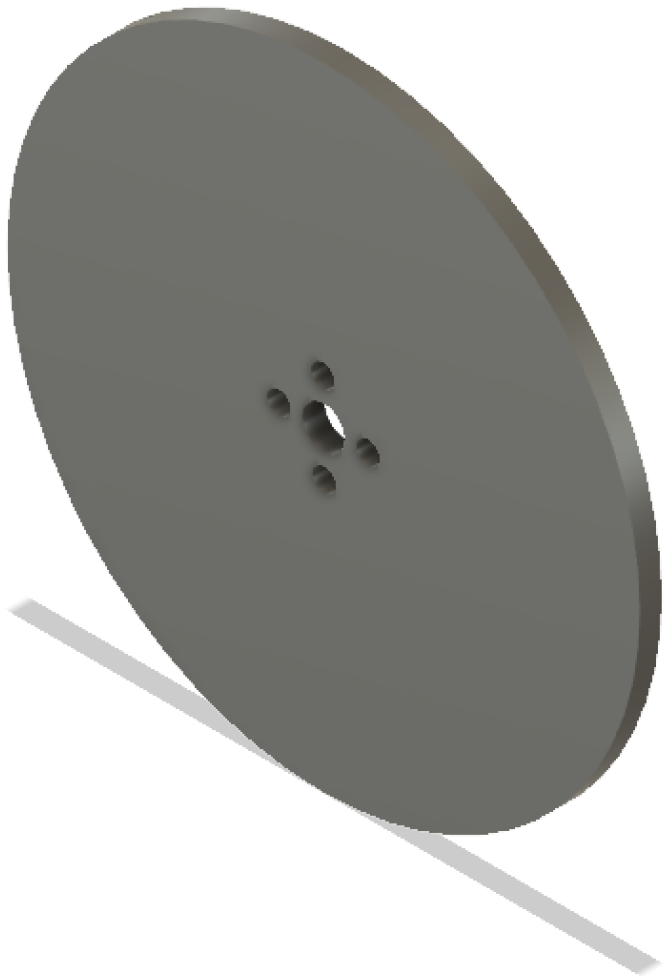


Figure 10: Simplified Carbon Steel Reaction Wheel