Oh, What a Bot!

HackRVA.org

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.

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 1: Sketch of the Pendulum

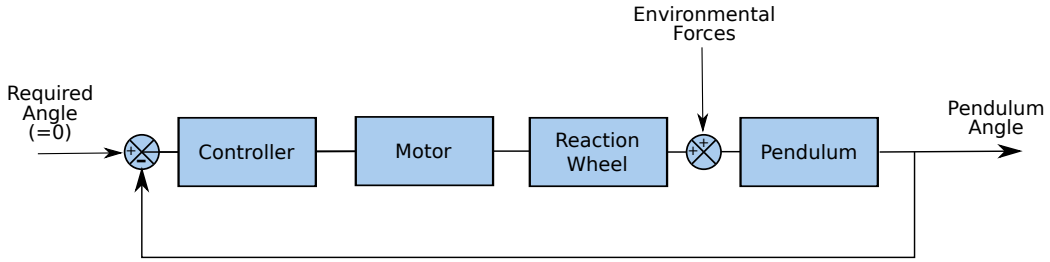

Figure 2: Simplified Pendulum Model

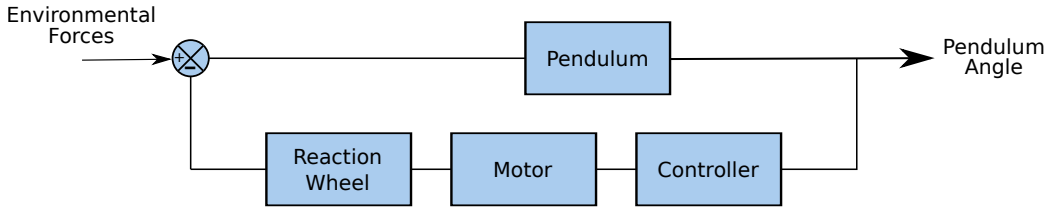

Figure 3: Regulator Model

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

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

where:
m = total mass of inverted pendulum (kg)
g = gravitational constant (m s^{-2})
l = distance from the pivot point to the center of pendulum mass (m)
θ = pendulum angle (rad)
F = environmental forces on the pendulum
I_c = moment of inertia of the pendulum (kg m^2)
I_f = moment of inertia of the reaction wheel (kg m^2)
ϕ = flywheel angle (rad)

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:

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

[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.

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

where $u(t)$ is the net torque applied to the pendulum. The sign convention is important; a positive torque $u(t)$ 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 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.

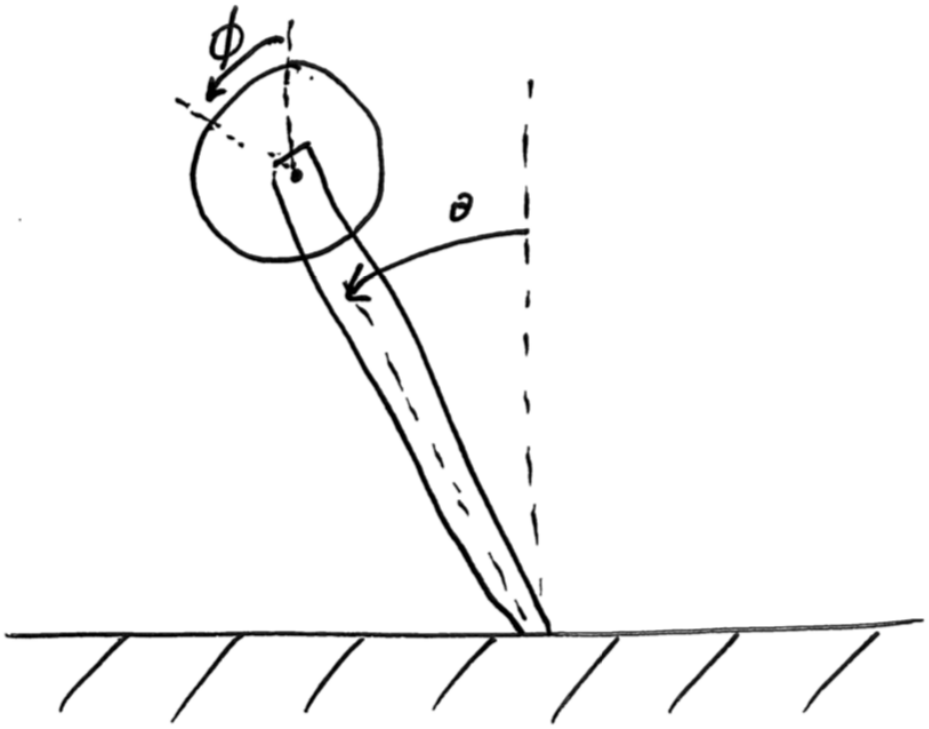

Figure 4: Simplified Pendulum Model

3 Motor and Flywheel Dynamics

Reference [2] models the motor as

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

where:
η_m = motor efficiency
η_g = gear efficiency
K_t = motor torque constant (N m A^{-1})
i = motor current

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, the 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,

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

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

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

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

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

where
T = the total motor torque
T_{shaft} = torque available to the flywheel
$T_{friction}$ = torque lost to motor and gear friction
A = coulomb friction coefficient

B = rotational friction coefficient

Combining equations and adding the reduction gear ratio,

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

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

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

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

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. Reference [1] give the relationship between motor current and voltage as

$$l\,\frac{di}{dt} + r\,i = v - K_v\,\omega \tag{14}$$

where:
l = armature inductance (henry)
r = armature resistance (ohm)
v = voltage supplied to the motor (volt)
K_v = motor voltage constant (V rad^{-1} sec)
ω = motor rotation speed (rad sec^{-1})

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

$$r\,i = v - K_v\,\omega \tag{15}$$

Note that, provided mks units are used, K_v and K_t will have the same magnitude. This can be seen by equating the mechanical and electrical power

$$T\omega = vi \tag{16}$$

where T = is the motor torque. Rearranging,

$$\frac{T}{i} = \frac{v}{\omega} \tag{17}$$

or

$$K_t = K_v \tag{18}$$

Back to the motor equation, we can solve (15) for i:

$$i = \frac{v}{r} - \frac{K_v\,\omega}{r} \tag{19}$$

Combining with (13)

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

Rearranging,

$$I_f \dot{\omega} + \left(B + \frac{K_t K_v}{r} \right) \omega + A\, sgn(\omega) = \left(\frac{K_t}{r} \right) v \tag{21}$$

Recognizing that $K_t = K_v$ and replacing them with K,

$$I_f \dot{\omega} + \left(B + \frac{K^2}{r} \right) \omega + A\, sgn(\omega) = \left(\frac{K}{r} \right) v \tag{22}$$

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.
We will 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 v for the purposes of this analysis.

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 design 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

$$T(s) = P(s)C(s)M(s)R(s) \tag{23}$$

where $P(s)$ is the pendulum transfer function, $C(s)$ is the control function, $M(s)$ is the motor transfer function, and $R(s)$ is the rotor transfer function. 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

$$v = \theta \tag{24}$$

Hence, the controller transfer function is

$$C(s) = \frac{v(s)}{\theta(s)} = 1 \tag{25}$$

The pendulum transfer function can be derived from (4)

$$m\,g\,l\theta(s) + u(s) = I_c\theta(s)s^2 \tag{26}$$

The rotor transfer function can be derived by noting that the torque generated by the flywheel is

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

Ignoring the environmental forces, the pendulum transfer functions are

$$P(s) = \frac{\theta(s)}{u(s)} = \frac{1}{I_c s^2 - mgl} \tag{28}$$

$$R(s) = \frac{u(s)}{\omega(s)} = -I_f s \tag{29}$$

Note that the transfer function for the rotor in (29) is negative. Combined with the summing junction shown in Figure 5, this produces a negative feedback control system.

From (22) the transfer function for the motor is:

$$I_f\,\omega(s)s + \left(B + \frac{K^2}{r}\right)\omega(s) = \left(\frac{K}{r}\right)v(s) \tag{30}$$

$$M(s) = \frac{\omega(s)}{v(s)} = \frac{\left(\frac{K}{r}\right)}{I_f s + (\frac{K^2}{r} + B)} \tag{31}$$

Using (25), (28), (29), and (31) results in

$$T(s) = \frac{-(\frac{1}{I_f I_c})(\frac{K}{r})s}{(s^2 - \frac{mgl}{I_c})(s + (\frac{K^2}{rI_f} + \frac{B}{I_f}))} \tag{32}$$

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

$$T(s) = \frac{-0.2464s}{s^3 + 4.73s^2 - 34.8s - 164.6} \tag{33}$$

The numerator of the transfer function is the characteristic equation that, when factored, is $(s - 5.8993)(s + 5.8993)(s + 4.7303)$.
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.
One possible method of stabilizing the system would be to use both proportional and derivative, or PD, feedback. The control function would be.

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

Rearranging,

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

indicating a zero at $1/p$ and a gain of $1/p$. 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 possibility for resolving the issue of the zero at the origin would be to implement a PID controller.

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

Rearranging,

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

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.

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 will have to be directly factored into the transfer function.

Here we modify (15) to include the feedback from the rotor velocity:

$$r\, i = v + K_\omega \omega - K_v\, \omega \tag{38}$$

where:
K_ω = rotor velocity feedback gain.

Solving for i,

$$i = \frac{v}{r} - \frac{(K - K_\omega)\, \omega}{r} \tag{39}$$

and

$$M(s) = \frac{\omega(s)}{v(s)} = \frac{\left(\frac{K}{I_f r}\right)}{s + (\frac{1}{I_f})(\frac{K^2}{r} - \frac{KK_\omega}{r} + B)} \tag{40}$$

Using (25), (28), (29), and (40), the open loop transfer function is now

$$T(s) = \frac{-(\frac{K}{I_c r})s}{s^3 + (\frac{Br - KK_\omega + K^2}{I_f r})s^2 - (\frac{mgl}{I_c})s - (\frac{Br - KK_\omega + K^2}{I_f r} - \frac{mgl}{I_c})} \tag{41}$$

Figure 8 shows the root locus with $p = 1/4.5$ and $K_\omega = 0.075$. With a system gain of 266 the controlling poles are at -11.5205, -1.5353, and -0.999.

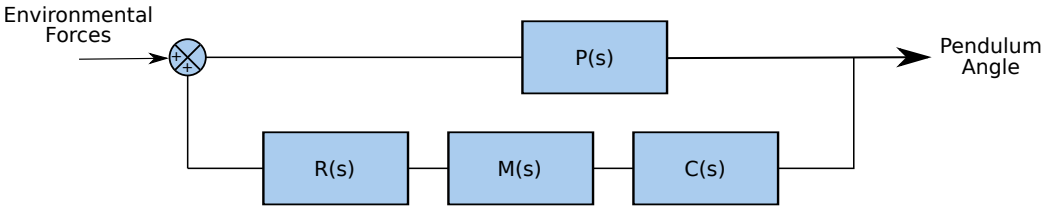

Figure 5: Control System Modeled as Transfer Functions

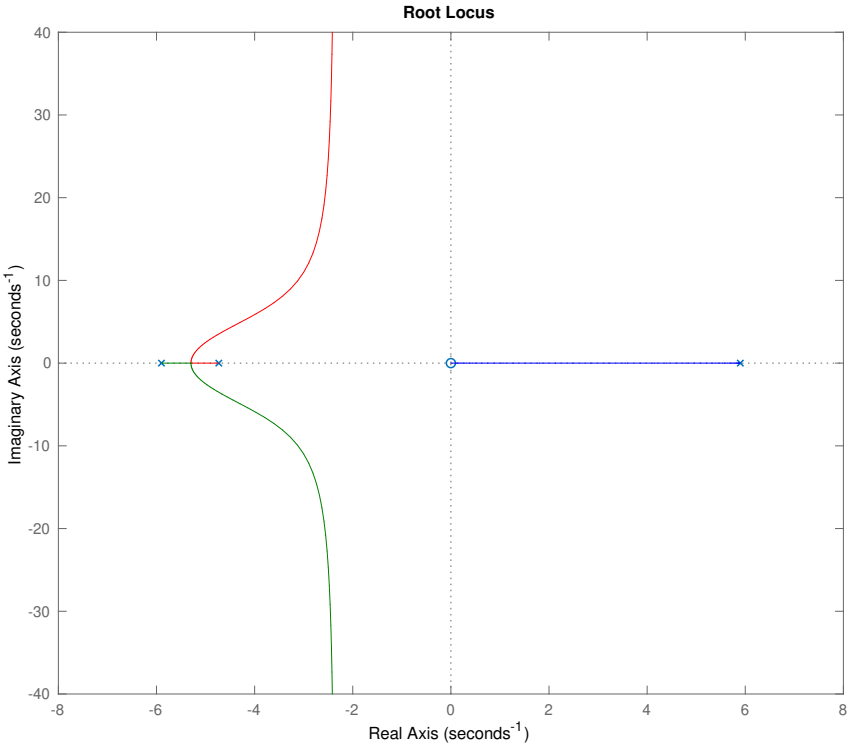

Figure 6: Root Locus Plot With Proportional Feedback

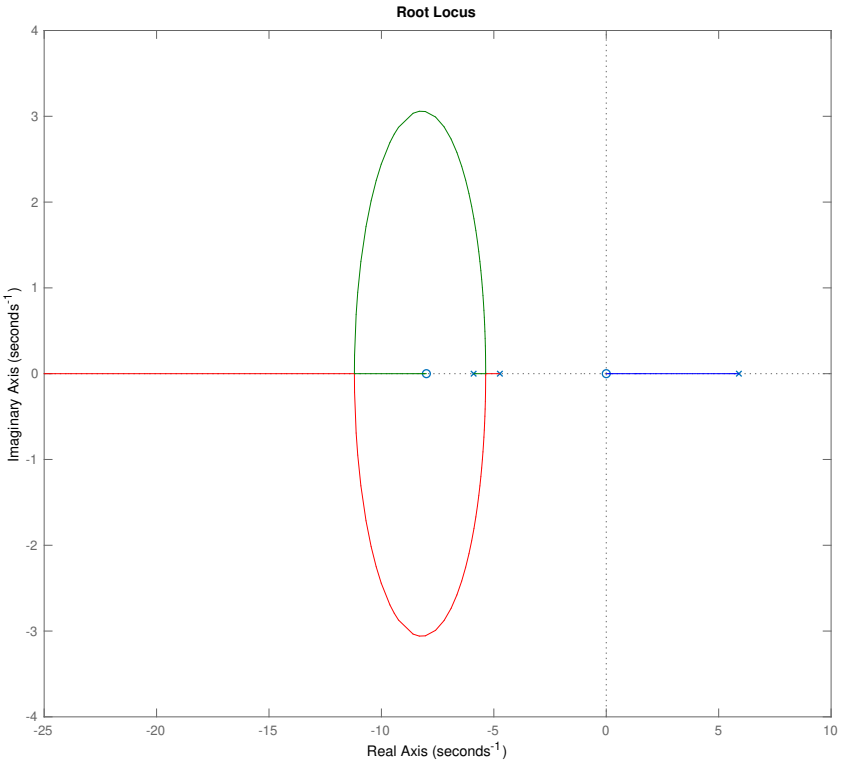

Figure 7: Root Locus Plot With PD Feedback

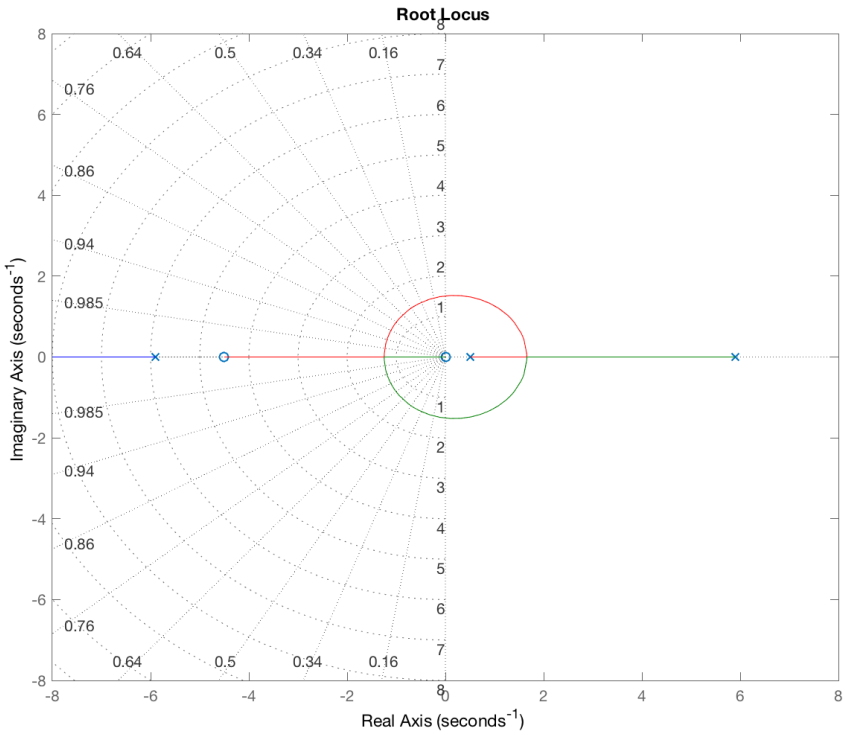

Figure 8: Root Locus Plot With PD and Rotor Velocity Feedback

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.

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. Here we are using 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 = 11 oz in (0.07768 Nm)
Encoder frequency = 211.2 counts/rev of gearbox shaft

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_f can be calculated directly from the stall current, gear ratio, and torque: K_f = 0.07768 Nm / 2.1 A = 0.0370 Nm/A. K_v can be set equal to K_f, 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.

Appendix B Pendulum Constants

A multitude of other values were calculated in MATLAB.

m	0.517327 kg
l	0.319038 m
I_c	0.046508 kg m^2
I_f	0.000164 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 r, l, and K_v.

r was measured using an EXTEC LCR meter, on the R_{DC} setting, as 5.82 Ω. l was measured with the same instrument as 5.994 mH at 1kHz. This gives a motor electrical time constant l/r of 0.001 sec. This is the time taken by the current in the motor to go from rest to 63% of the final steady-state current.

K_v was measured by driving a motors with a constant voltage and varying loads while measuring V, i, and ω.

The motor was driven with a 12 V power supply. The current was measured by the power supply. Measuring current in this manner can have larger uncertainties, but comparing with measurements from a μCurrent DVM adapter indicated that the power supply current measurement was reasonable.

ω was measured using the motor's encoder and an Arduino. The encoder measured the shaft speed, not the gear-set speed. However, a conversion coefficient of 211.2 counts/revolution was used to convert the encoder pulse rate to gear set rotation speed.

The following table shows the results for three different loads. The first load was just rotor friction. The second load added the Pololu hub adapter. The third load included a plastic wheel attached to a propeller to increase the drag. The resulting measured values are as follows:

Case	v (V)	i (A)	speed (counts/sec)	ω (RPM)	ω (sec^{-1})
1	11.95	0.0675	5,942.2	1,688.1	176.74
2	11.95	0.075	5,923.58	1,682.83	176.19
3	11.95	0.153	5,574.88	1583.77	165.82

Calculating K_v can be approached two ways. The first is to solve (15) using the measured value of R. The resulting values of K_v, one for each case, are:

Case	K_v (V sec)
1	0.0654
2	0.0653
3	0.0667

A second approach is to use two cases in (15) and solve simultaneously for R and K_v. Cases 1 and 3 were chosen because of the largest difference in i,

attempting to ensure that the two cases are independent. This results in a values of $r = 8.24\ \Omega$ and $K_v = 0.0645$ V sec.
The value of K_v is fairly consistent with the values determined in the first approach, but the value of r appears to be high. This may be caused by the limited change in i in the data. Use of a larger load torque might result in a larger value of i and a better estimate of r. However, for the purposes of this work the following values will be used:

r (Ω)	K_v (V sec)	K_t (N m A)
5.82	0.0667	0.0667

Appendix D Measuring Motor Friction Constants

Friction in the motor was evaluated by measuring the supply current as a function of motor speed. The friction force is related to the motor current as $F = K_t i$. The measurement was performed by using a power supply with a variable current limit. For various currents the speed of the motor was measured by monitoring the motor encoder output with an Arduino. The current was measured both at the power supply as well as with a μCurrent DVM adapter.

The following data was collected:

Supply Voltage (V)	Supply Current (A)	μCurrent (A)	Speed (pulse/sec)	Speed (rpm)	Speed radians/sec
10.90	0.068	0.0676	5417	1538.9	161.16
10.04	0.063	0.0627	5001	1420.7	148.78
9.09	0.058	0.0596	4520	1284.1	134.47
7.93	0.056	0.0562	3937	1118.5	117.13
6.99	0.055	0.0550	3465	984.4	103.08
6.00	0.051	0.0521	2953	838.9	87.85
5.00	0.047	0.0500	2437	692.3	72.50
4.04	0.046	0.0483	1960	556.8	58.31
3.57	0.046	0.0486	1701	483.2	50.60
3.02	0.047	0.0481	1421	403.7	42.27
2.52	0.045	0.0459	1169	332.1	34.78
2.02	0.042	0.0437	910	258.5	27.07
1.51	0.039	0.0409	662	188.1	19.69
1.02	0.034	0.0366	414	117.6	12.32
0.47	0.031	0.0328	148	42.0	4.40

The encoder speed, in pulses/sec, was converted to RPM using the encoder frequency of 211.2 counts/revolution.

The Excel regression function was used to fit the μCurrent current measurement as a function of the speed in radians/sec. The regression intercept represents the fixed Coulomb friction and the slope represents the viscous friction. The regression results were 2.47E-3 N m and 1.20E-5 N m/(radian/sec), after converting from current using K_t. The coulomb friction represents approximately 3% of the motor stall torque. At the motor's no-load speed the viscous friction would double the total friction to approximately 6% 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.

Appendix E Arduino Code for Measuring RPM

The following is an Arduino sketch used to monitor the speed of a dc motor. 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 F Reaction Wheel Design Options

The initial reaction wheel design was modeled after XXXX, with modifications to allow the reaction wheel to mate properly with a Pololu motor hub. The reaction wheel is shown in Figure **??**. The reaction wheel body is aluminum 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 **??**. 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 1.605E04 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.

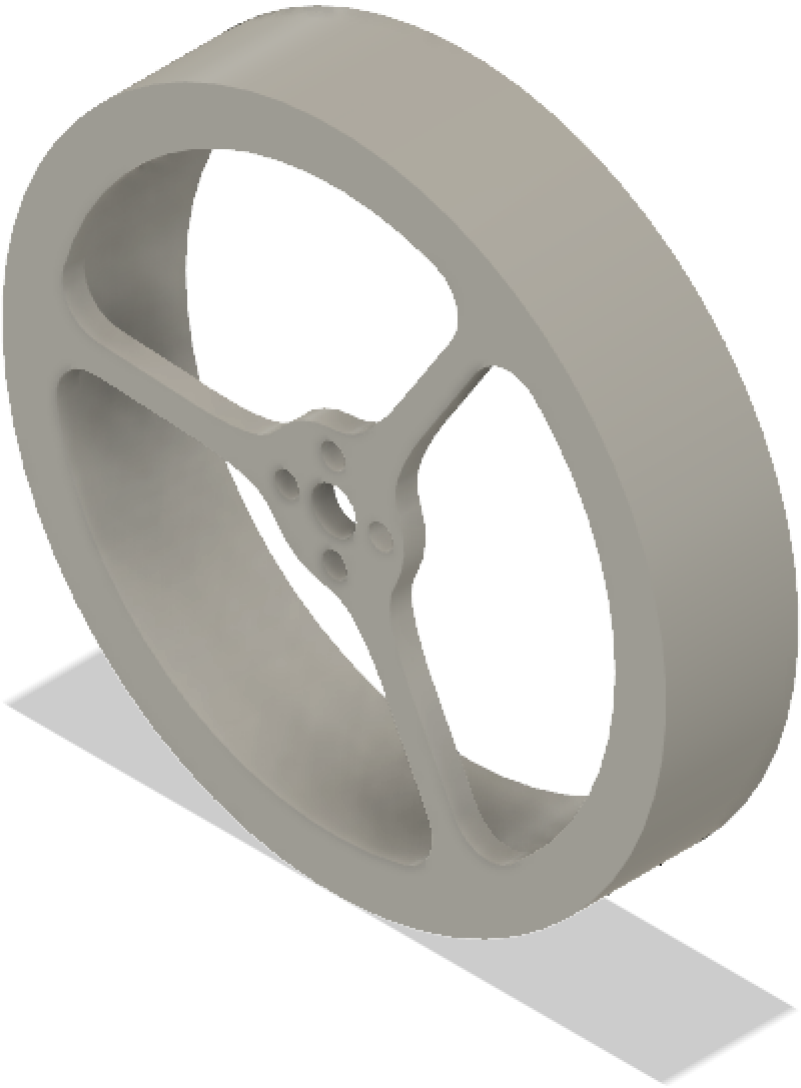

Figure 9: Aluminum Reaction Wheel

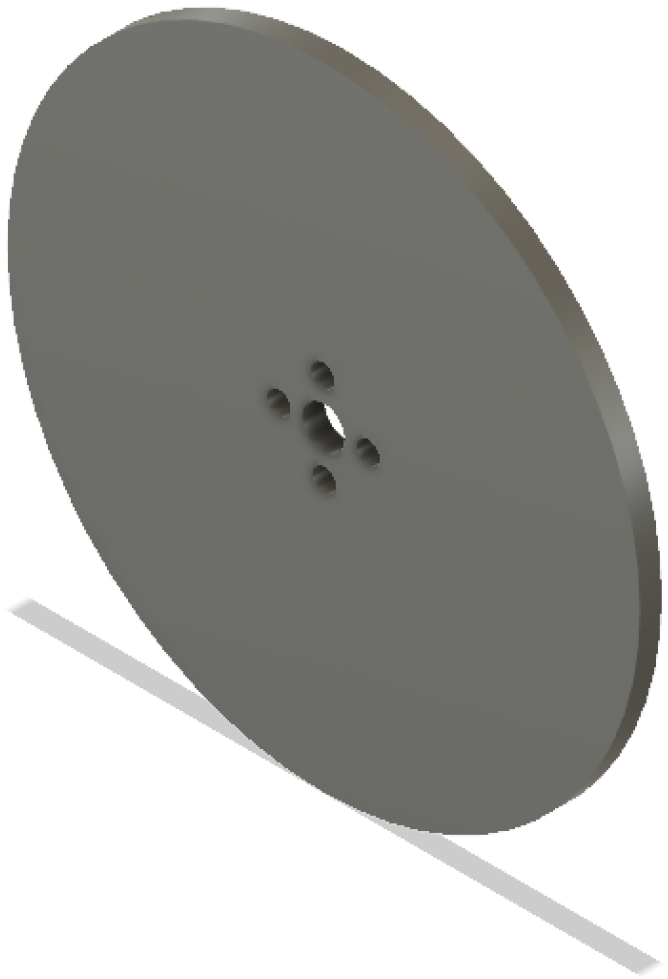

Figure 10: Simplified Carbon Steel Reaction Wheel