BLDC propeller Thrust

INTRODUCTION

The previous model used for the EE471 design project had some issues. The actuators (spinning propellors) were modeled basically as a 1st order lowpass filter for no good reason (intuition does not count as a good reason), and they were not even considered as part of the plant. So the time has come to derive a model based on actual physical principles. This model will include the following significant assumptions:

1. The BLDC can be modeled as a normal 2nd order DC motor. This is acceptable because the only difference between the two types is the method of commutation.
2. Air resistance experienced by the spinning prop is proportional to the speed squared. This is implying that the most significant component of air resistance is turbulent drag, given by: $F_D = \frac{1}{2} \rho v^2 C_D A$. We can simplify this by collecting all the constants like so: $F_D = \mathrm{d} v^2$
3. Model thrust as proportional to the square of the rotational rate. There is a well known propellor thrust equation given by: $T = C_t \rho \mathrm{n}^2 D^4$ (lbf)
4. The aerodynamic thrust responds instantaneously to rotor speed, meaning that the above equation has no frequency dependance. I do not know if this is true, and I suspect it is not, so the prevailing assumption is that the working fluid responds many times faster to changes in rotor speed, than does the rotor speed to changes in input voltage (or duty cycle). I would like to check up on this at a later date.

I would also like to standardize some symbols:

Those applied to motors:

Kv --> BLDC motor constant (rpm/V)
K --> SI motor constnat (V/[rad/s])
ϕ_i --> Angular Position of the ith Rotor (rad)
i_i --> Current through the ith Armature (A)
J --> Moment of Inertia of combined Rotor and Propeller $(\mathrm{kg} \cdot m^2)$
R --> Motor Coil Resistance (Ω)
L --> Motor Coil Inductance (H)
d -->Turbulent Drag Coefficient $(N / [\mathrm{rad}/s]^2)$
V_i --> Effective Voltage applied to ith Motor -- $V_{\mathrm{supply}} \cdot \mathrm{D}$ -- (V)
PW_i --> Pulse Width Input to Ith Motor (μs)

n_i --> Rotational Rate in rps of ith Rotor (rps)
n_{rpm_i} --> Rotational Rate in rpm of ith Rotor (rpm)

Those Related to Propellers and Aerodynamics:

T_{Ni} -->Thrust of ith Actuator (N)
C_t -->Thrust Coefficient (Calculated) (unitless)
D --> Propeller Diameter (feet)
ρ --> Air density (slugs/ft^3)
λ_N --> Combined Thrust Constant $\left(N/[\text{rad}/s]^2\right)$

Those Related to the Plant

θ --> Angle of Pendulum Relative to vertical (rad)
θ_{range} --> Maximum Measureable Angle (rad)
V_{range} --> Measurement Voltage range (V)
V_{center} --> Voltage presented when $\theta = 0$ (rad)
I_C --> Moment of Inertia about pendulum's pivot point $\left(\text{kg}\cdot m^2\right)$
m --> Mass of pendulum (kg)
R_P --> Length of Pendulum (m)
r_C --> Distance from pivot to center of mass (m)
g --> gravitational constant $\left(m/s^2\right)$
α --> Coefficent of angular dependence in acceleration equation $\left(1/s^2\right)$
b --> Velocity Dependent Friction Coeficient (N/[m/s])
β --> Input gain (1/kg·m)
γ --> Measurement Gain (V/rad)
u_i --> Control input (N)

Getting to It

Emax BLDC motor specs

Specified motor constant

```
Kv = 2300;
```

Specified parameters at 12V input with 5X3 (DIameter X Pitch) plastic prop:
Shown: speed (rpm), thrust (grams), current (amps)

```
n12 = 20100;
```

```
T12g = 310;
```

```
I12 = 7.5;
```

Standardization of units

Motor constant in standard units (Vs/rad)

```
K = N[60 / 2 / Pi / Kv]
```

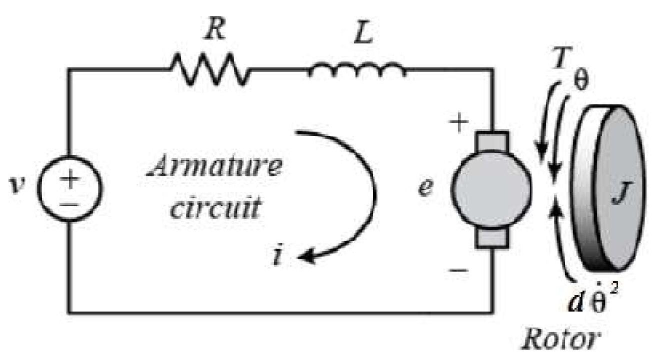
0.00415187

Force conversion factors: grams to newtons (g2N), pounds force to newtons (lbf2N)

```
g2N = .0098;
```

```
lbf2N = 4.448;
```

DC motor with propellor (steady state)



;

The differential equations which descirbe this dynamical DC motor model are as follows:
Note that I'm calling the angle θ ϕ, since I want to use θ for the plant.

$$K i_i = J\ddot{\phi}_i + d\,\dot{\phi}_i^2$$
$$V_i = R\,i_i + L\frac{di_i}{dt} + K\,\dot{\phi}_i$$

In **steady state** (no change of speed) motor torque is equal to the torque due to friction.
Also, the voltage across the Inductor is nothing, because the current doesn't change:

$$\mathbf{K i_i = d\,\dot{\phi}_i^2}$$
$$\mathbf{V_i = R i_i + K\,\dot{\phi}_i}$$
$$\dot{\phi}_i = 2\,\pi n_{rps}$$
$$n_{rps} = n_{rpm}/60$$

According to the EMAX 2204 datasheet, the angular speed (in rad/s) with prop load at

12V input (2000μs pulse width).

```
ω12 = N[2 * Pi * n12 / 60]
```

2104.87

Solving for the Resistance

```
R = (12 - K * ω12) / I12
```

0.434783

Since we already said that thrust is proportional to the square of the rotational rate, we desire to obtain an expression for the speed squared as a function of applied voltage:

$$i_i = \frac{d}{K}\dot{\phi}_i^2$$
$$i_i = \frac{(V_i - K\dot{\phi}_i)}{R}$$
$$\frac{d}{K}\dot{\phi}_i^2 + \frac{K}{R}\dot{\phi}_i - \frac{V_i}{R} = 0$$
$$\dot{\phi}_i = \frac{K}{2d}\left[\frac{-K}{R} \pm \sqrt{\frac{K^2}{R^2} + 4\frac{d V_i}{K R}}\right]$$

Only the positive solution makes sense, so we only take that one. After some simplification, the expression for the speed squared is obtained:

$$\phi_i^2 = \frac{K}{dR}V_i - \frac{K^4}{2d^2R^2}\left[\sqrt{1 + \frac{4dR}{K^3}V_i} - 1\right]$$

The squre of the rotational rate in rps is obtained by dividing the whole thing by 4 π^2.

$$n_i^2 = \frac{K}{4\pi^2 dR}\left[V_i - \frac{K^3}{2dR}\left[\sqrt{1 + \frac{4dR}{K^3}V_i} - 1\right]\right]$$

Lets review this formula: So far, we know the following: K was given, R was calcuated, and n is known for $V_a = 12\ V$. The only thing left to calculate is d. This can be done by comparing with data. I did it by guessing and checking -- evaluating the above equation at the operating condition spec'd in the spec sheet:

$$d = 6.89 X 10^{-9}\ (N / [m/s]^2)$$

$$n_i^2 = 3.517 X 10^4\left[V_i - 11.9672\left[\sqrt{1 + 0.1671\ V_i} - 1\right]\right]$$

Obtaining the thrust funciton is just a matter of multiplying the speed squared by some constant.

Linearized expression for speed squared

The function we got is nonlinear. It will become convenient to make n^2 a linear function of voltage. Let us linearize about the half throttle point: V_i=6V. Since only the speed term in one equation is nonlinear, only that term in the eqs of motion will be modified.

The first order Taylor series expansion is like this:
f(x) = f(a) + (df/dx) $|_{x=a}$(x-a)

$n_i^2\,(6) = 3.6276\,X\,10^4\,(\text{rps}^2)$

$$\frac{\partial n_i^2}{\partial V_i} = \frac{K}{4\,\pi^2\,d\,R}\left[1 - \frac{K^3}{2\,d\,R}\left[\frac{1}{2}\,\frac{4\,d\,R}{K^3}\left(1 + \frac{4\,d\,R}{K^3}\,V_i\right)^{-1/2}\right]\right] = \frac{K}{4\,\pi^2\,d\,R}\left[1 - \left(1 + \frac{4\,d\,R}{K^3}\,V_i\right)^{-1/2}\right]$$
$$\frac{\partial n_i^2}{\partial V_i}\,(6) = 1.0318\,X\,10^4\,(\text{rps}^2/V)$$

so...

$n_i^2\,(V) = 3.6276\,X\,10^4 + 1.0318\,X\,10^4\,(V_i - 6)$

$\therefore\ \boldsymbol{n_i^2(V) = 1.0318\,X\,10^4\,(V_i) - 2.5633\,X\,10^4}$

Linearized Thrust Function

Thrust can be calculated using the following formula:
$\boldsymbol{T_i = C_t\,X\,\rho\,X\,n_i^2\,X\,D^4}$

T_i = Thrust in lbf
C_t = Thrust Coefficient
ρ = air density in slugs/ft^3
n_i = revolution rate in rps
D = diameter in feet

Since we have data for top speed, we can solve for the thrust coefficient to get thrust as a function of speed.

```
ρo = 0.002378;
```

```
n12s = n12 / 60;
```

```
Df = 5 / 12;
```

The thrust specified in the emax datasheet can be converted to lbf:

```
T12lbf = T12g * g2N / lbf2N
```

0.683004

Knowing the rotational rate and the thrust at a known operating point (12V input) allows us to calculate the thrust coefficient:

```
Ct = T12lbf / ρo / n12s^2 / Df^4
```

0.0849115

This is a reasonable value for the thrust coefficient, so we can move on, and rewrite the thrust function as:

$\boldsymbol{T_{Ni} = \lambda_N n_i^2}$
where $\lambda_N = C_t X \rho X D^4 X 4.448 = 2.707 X 10^{-5}$

The factor of 4.448 constitutes the conversion from pounds of force to Newtons.

The complete linearized thrust function is obtained by substituting the linearized expression for n^2 into the above:

$T_{Ni} = 2.707 X 10^{-5}\left[1.0318 X 10^4 (V_i) - 2.5633 X 10^4\right]$
$\boldsymbol{\therefore T_{Ni} = 0.2793 (V_i) - 0.6939}$

Notice that this gives a negative result for voltages below 2.48V, and that does not make sense. So if we agree on this as the model to be used, let us agree to not apply an input that corresponds to anything below 2.48V. Before we decide if this is good enough, let's try to get everything on a plot so we can look at it.

Mapping Pulse Width to Effective Voltage

Generally, ESCs designed for RC hobby applications take a pulse width ranging from 1ms to 2ms. **A 1ms pulse should correspond to a low throttle control signal, and a 2ms pulse should correspond to a full throttle control signal**. For the electric motor, it would make sense if 1ms corresponds to 0V, and 2ms corresponds to 12V, but really, **there is no reason to believe that the person who programmed the ESC didn't just apply some arbitrary monotonic function mapping the input pulse width to the effective output voltage (or duty cycle)**, and in fact, these things have several operating modes, programmable through the throttle. With that in mind, lets assume that alow throttle signal will be *near* 1ms, and a full throttle signal will be *near* 2ms. By adjusting the mapping of min and max pulse width signals to min and max effective voltages, wilst not straying too far from the baselines, we can get our nonlinear thrust function to match the data very closely.

The plot below was generated with the following postulated operating parameters:

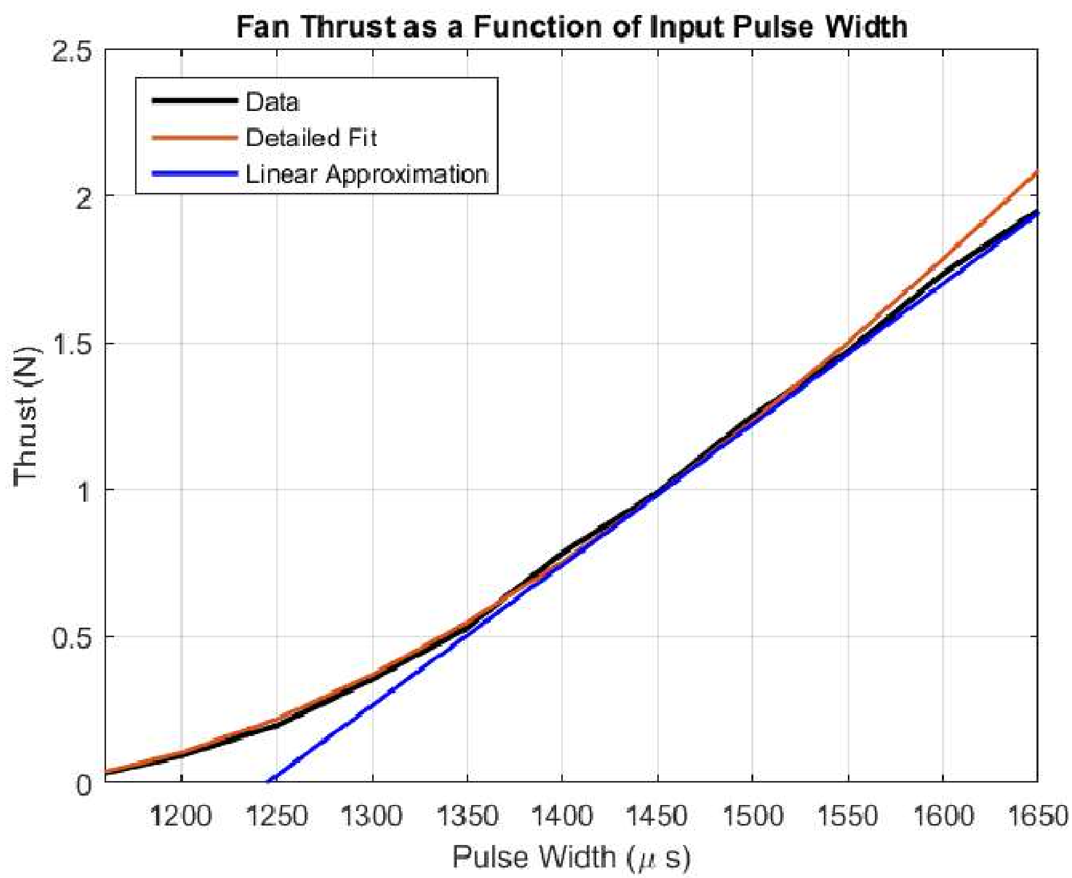
$\mathbf{PW_{min} = 1100\ \mu s}$

$\mathbf{PW_{max} = 1800\ \mu s}$

$V_R = 12\ V$

$V_{eff} = \frac{PW_{input} - PW_{min}}{PW_{max} - PW_{min}}\ X\ V_R$

Note: the above mapping is the only truly dippy mathematical thing going on here. The results below make it easy to defend.



;

This looks pretty good! We can adjust the offset if we want to minimize error, but we might as well just leave it and see how it works. The next thing is to make a linearized dynamical model and see if its steady state response matches this.

DC motor with propeller (dynamical model)

As stated before, the differential equations describing the electric motor with the propoellor are:

$K\,i_i = J\ddot{\phi}_i + d\,\dot{\phi}_i^{\,2}$

$V_i = R\,i_i + L\,\frac{di}{dt} + k\,\dot{\phi}_i$

The nonlinear state space equations for states i and $\dot{\theta}$ can be written as:

$\ddot{\phi}_i = \frac{1}{J}\left[\text{K}i_i - d\,\dot{\phi}_i^{\,2}\right]$
$\frac{\text{d}i_i}{\text{dt}} = \frac{1}{L}\left[V_i - \text{R}i_i - k\,\dot{\phi}_i\right]$

That first equation is nonlinear, so we can linearize it about the same rotational speed as we did earlier... so let’s find out what the rotational speed is when the effective input voltage is 6V. Above, we showed that:

$n_i^2\,(6) = 3.6276 \text{ X } 10^4 \text{ (rps)}^2$

so $n_i(6)$ = 190.46 rps

$\therefore\ \dot{\phi}_i(6) = 2\,\pi\,n(6)\ = 1.1967\,X\,10\text{^}3\ (\text{rad}/s)$

$\frac{\partial\,\ddot{\phi}_i}{\partial\,\dot{\phi}_i}\,(6) = -\frac{2\,d}{J}\,\dot{\phi}_i(6) = -\,\frac{4\,\pi\,d\,n_i(6)}{J} = -\,\frac{1.6491\,X\,10^{-5}}{J}$

Notice that this is the only term that needs to be linearized. We can now make a linear SS model.

Linearized State Space Representation of the Motor and Prop

The state space model looks like this: Note that the Vbias input is required for matching of the linearization about a 6V input. Also I got rid of subscripts for now.

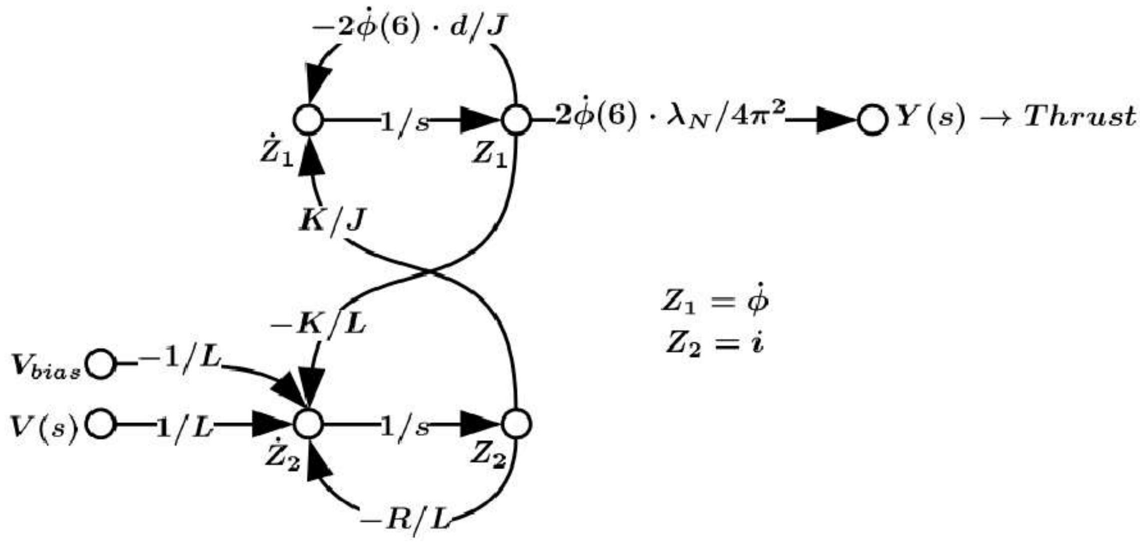
$$\begin{pmatrix} \ddot{\phi} \\ \dot{i} \end{pmatrix} = \begin{pmatrix} \dot{z}_1 \\ \dot{z}_2 \end{pmatrix} = \begin{pmatrix} -\frac{2\,d}{J}\,\dot{\phi}(6) & \frac{K}{J} \\ -\frac{K}{L} & -\frac{R}{L} \end{pmatrix}\begin{pmatrix} z_1 \\ z_2 \end{pmatrix} + \begin{pmatrix} 0 & 0 \\ \frac{1}{L} & -\frac{1}{L} \end{pmatrix}\begin{pmatrix} V_a \\ V_{\text{bias}} \end{pmatrix}$$

$$\dot{\phi}^2 = y = \begin{pmatrix} 2\,\dot{\phi}(6) & 0 \end{pmatrix}\begin{pmatrix} z_1 \\ z_2 \end{pmatrix}$$

The output equation comes from the first equation of motion -- it was solved for $\dot{\theta}^2$, and the linearized expression for $\ddot{\theta}$ was substituted in like so:

$$\dot{\phi}^2 = \frac{1}{d}\left[K\,i - J\left[\frac{1}{J}\left(K\,i - 2\,d\,\dot{\phi}^2\right)\right]\right] = \frac{1}{d}\left[K\,i - K\,i + 2\,d\,\dot{\phi}(6)\,\dot{\phi}\right] = 2\,\dot{\phi}(6)\,\dot{\phi}$$

Converting this output to thrust is just a matter of multiplying the C matrix by some constants. A signal flow graph for the motor/prop combo with thrust as an ouptut is shown below.



;

So, K is known, R was calculated, d was calculated, n(6) is known... but J and L are unmeasured. These parameters determine the speed of the system. I tried to figure them out by looking at pictures of the motor and propellor, and finding out how much they weigh, and guessing things.

```
Am = {{-2 * d * ϕ̇6 / J, Km / J}, {-Km / L, -Ra / L}}
```

$$\begin{pmatrix} -\frac{2\, d\, \dot{\phi}_6}{J} & \frac{\text{Km}}{J} \\ -\frac{\text{Km}}{L} & -\frac{\text{Ra}}{L} \end{pmatrix}$$

```
Bm = {{0, 0}, {1 / L, -1 / L}}
```

$$\begin{pmatrix} 0 & 0 \\ \frac{1}{L} & -\frac{1}{L} \end{pmatrix}$$

```
Cm = {{2 * ϕ̇6, 0}}
```

$$\begin{pmatrix} 2\, \dot{\phi}_6 & 0 \end{pmatrix}$$

```
Dm = {{0}}
```

$$\begin{pmatrix} 0 \end{pmatrix}$$

Thrust/Voltage Transfer Function

It will be convenient to have a transfer function for this motor for analysis of its response. Let’s derive one from the state space model.

```
Φ = IdentityMatrix[2] * s - Am
```

$$\begin{pmatrix} s + \frac{2\,d\,\dot{\phi}_6}{J} & -\frac{\text{Km}}{J} \\ \frac{\text{Km}}{L} & \frac{\text{Ra}}{L} + s \end{pmatrix}$$

```
TF = Simplify[Cm.Inverse[Φ].Bm]
```

$$\left(\frac{2\,\text{Km}\,\dot{\phi}_6}{\text{Km}^2 + J\,s\,(\text{Ra}+L\,s) + 2\,d\,(\text{Ra}+L\,s)\,\dot{\phi}_6} \quad -\frac{2\,\text{Km}\,\dot{\phi}_6}{\text{Km}^2 + J\,s\,(\text{Ra}+L\,s) + 2\,d\,(\text{Ra}+L\,s)\,\dot{\phi}_6} \right)$$

```
Dcoeff = CoefficientList[Denominator[TF[[1, 1]]], s]
```

$$\left\{2\,d\,\text{Ra}\,\dot{\phi}_6 + \text{Km}^2,\ 2\,d\,L\,\dot{\phi}_6 + J\,\text{Ra},\ J\,L\right\}$$

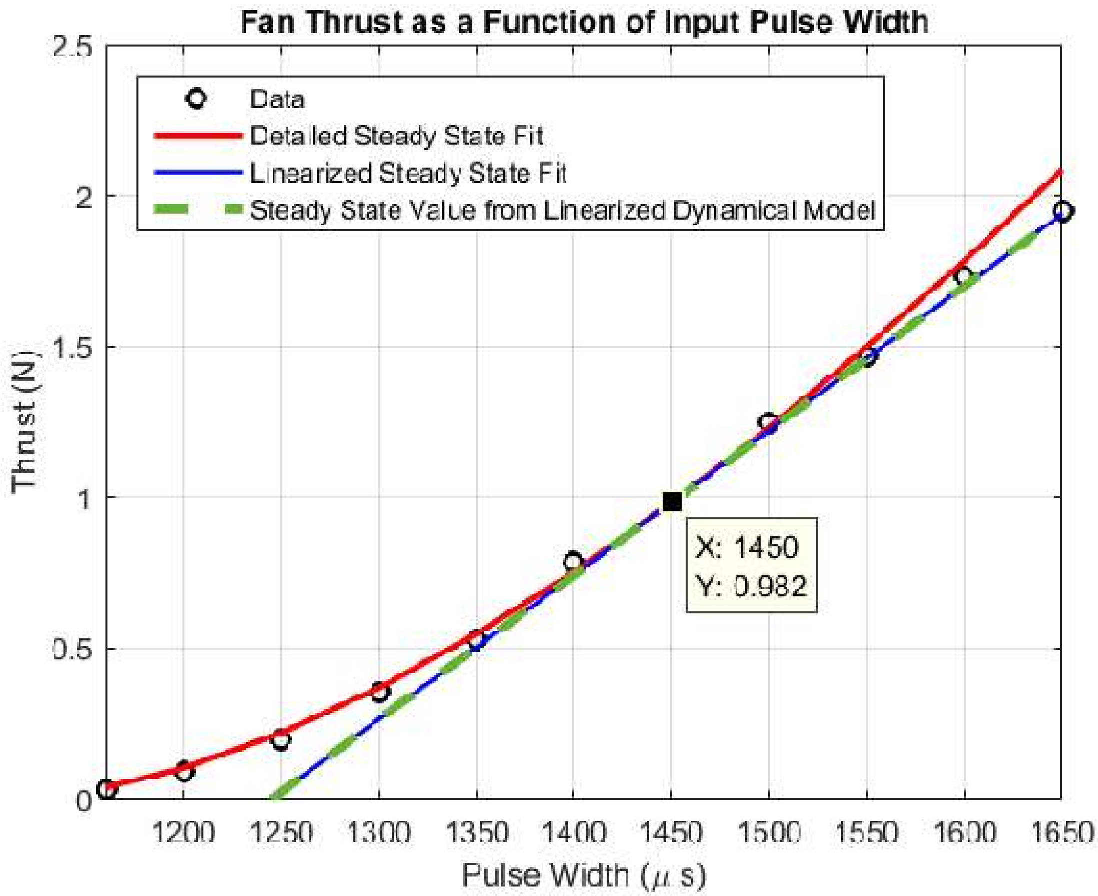
So now that we have two transfer functions for $\frac{\dot{\theta}^2(s)}{V(s)}$, we can easily obtain a transfer function for $\frac{\text{Thrust}(s)}{V(s)}$ by multiplying by a factor of $\frac{\lambda_N}{4\,\pi^2}$. Ignoring that second TF for now (the one for input bias)...

$$\frac{\text{Thrust}(s)}{V(s)} = \frac{\frac{2\,K\,\dot{\theta}(6)}{J\,L}\,\frac{\lambda_N}{4\,\pi^2}}{s^2 + \frac{J\,R + d\,L\,\dot{\theta}(6)}{J\,L}\,s + \frac{2\,d\,R\,\dot{\theta}(6) + K^2}{J\,L}}$$

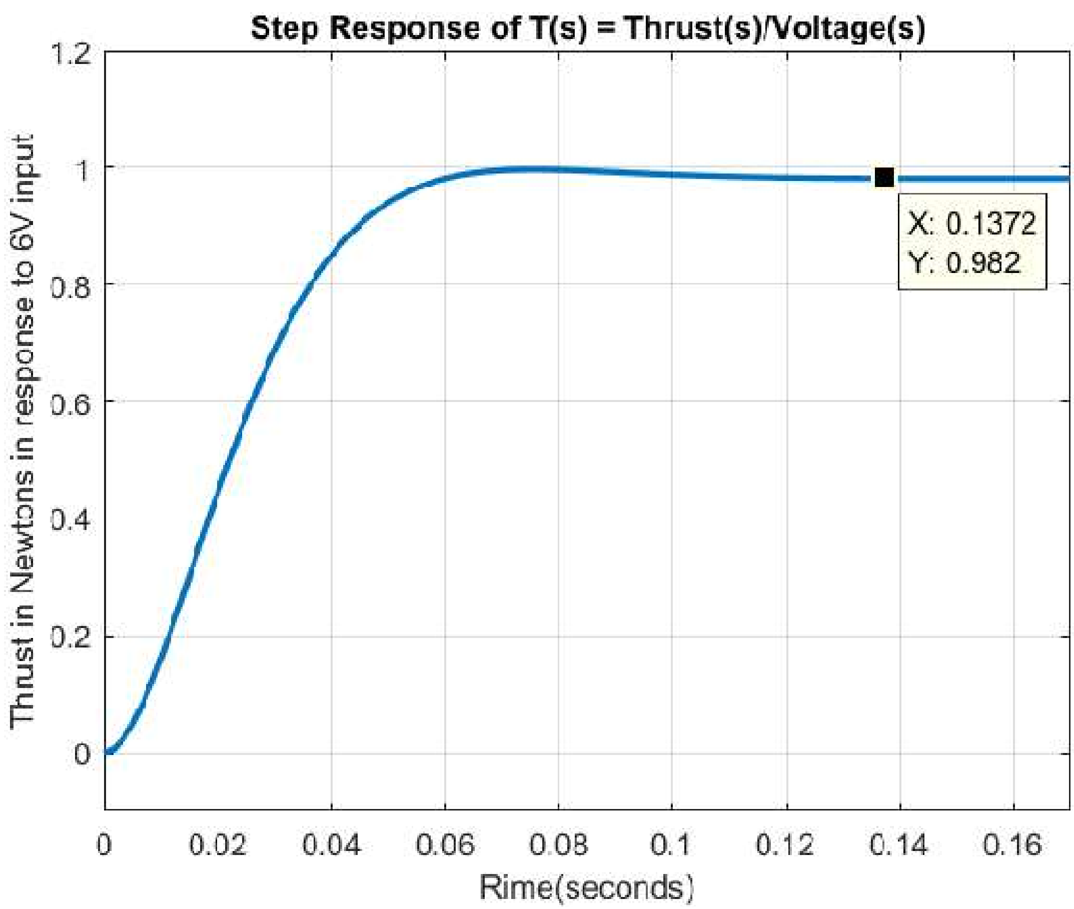
Withe the calculated and approximated values for the specific parameters, this gives:

$$\text{Thrust}(s) = \frac{1346}{s^2 + 111.1\,s + 4819}\,V(s)$$

One important thing to remember is that our linearized steady state model required a voltage offset of negative 2.48V (due to the modificaiton of mapping PWM signal to effective voltage). This is the input into the 2nd TF. When all is said and done, the results are thoroughly satisfying. Note that the data cursor in both figures shows highlights the steady state value of thrust due to an effective input of 6V, which happens to correspond to a pulse width of 1450μs according to the mapping used.


Fan Thrust as a Function of Input Pulse Width
Data
Detailed Steady State Fit
Linearized Steady State Fit
Steady State Value from Linearized Dynamical Model
X: 1450
Y: 0.982
Thrust (N)
Pulse Width (μ s)
2.5
2
1.5
1
0.5
0
1200
1250
1300
1350
1400
1450
1500
1550
1600
1650

;



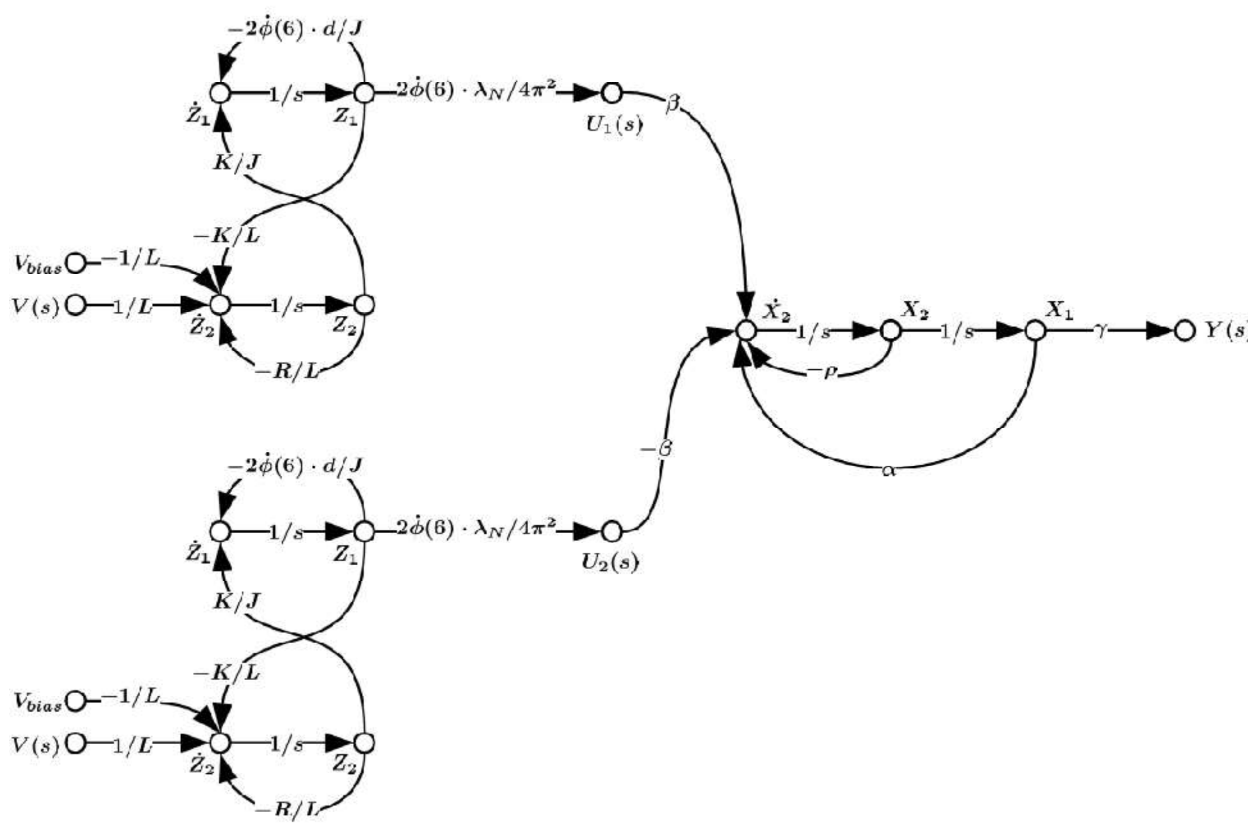
;

Another important thing to remember is that I totally guessed the coil inductance and the moment of inertia of the rotor/prop. Well, not totally guessed: I judged the size, core material and number of turns based on pictures of the motor, and I assumed that the rotor weighted 10 grams and that its moment of inertia can be approximated by a ring of mass at a distance around where the magnets are. Measuring these things could help.

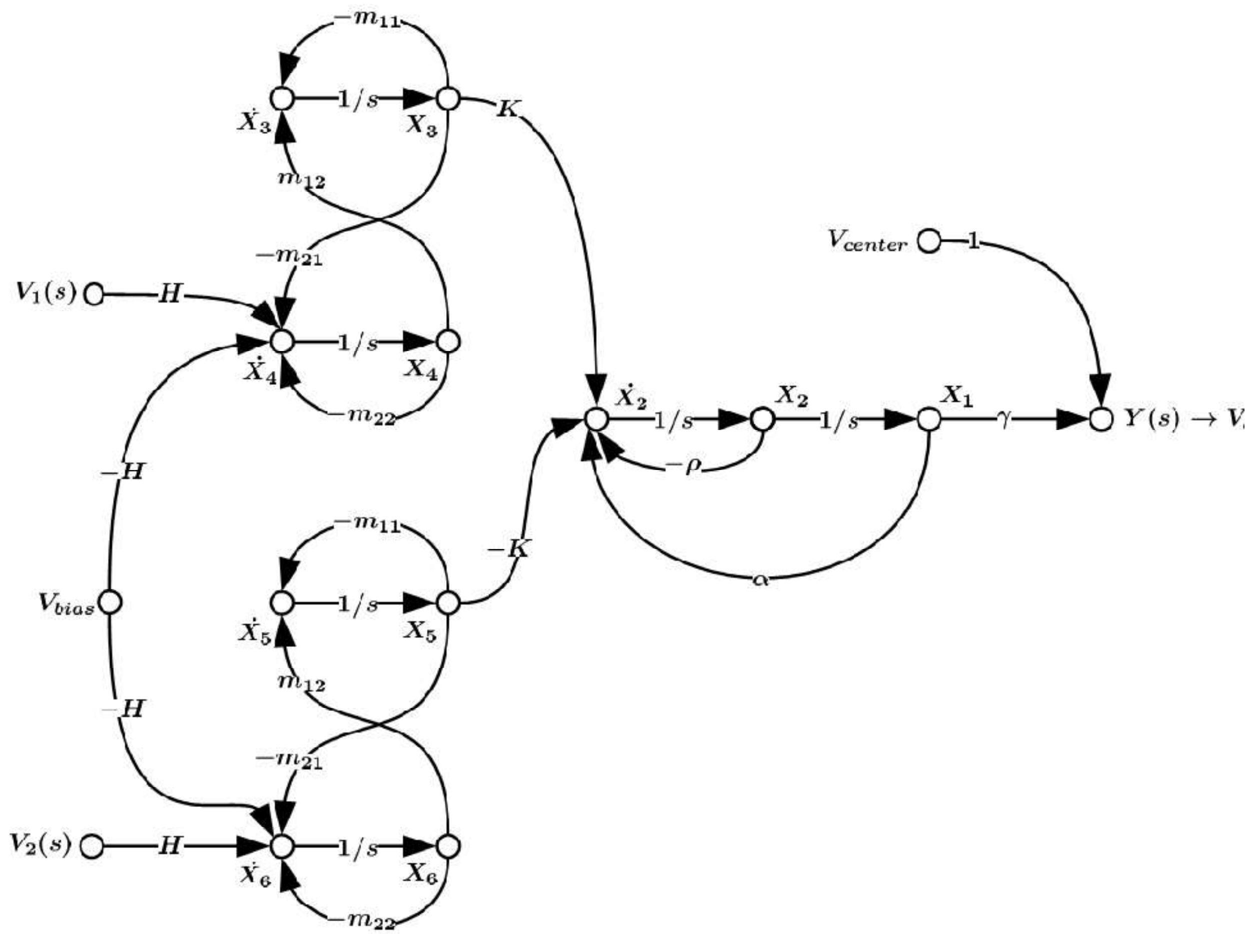
Also, I have no idea if the ESC has some kind of servomechanism to control the speed. All this assumes it does not.

Complete Open Loop State Space Model: Plant + Actuators:

We can take a look at the whole system, actuators and all on a signal flow graph like this:



This picture is kind of rough, so we can do some variable changes, and simplify some connections to arrive at this:



I also added another input V_{center}, which is 2.5V with our current setup.

The states are as follows:

$\boldsymbol{X_1 = \theta_p}$
$\boldsymbol{X_2 = \dot{\theta}_p}$
$\boldsymbol{X_3 = \dot{\phi}_1}$
$\boldsymbol{X_4 = i_1}$
$\boldsymbol{X_5 = \dot{\phi}_2}$
$\boldsymbol{X_6 = i_2}$

The Inputs are:

$\boldsymbol{U_1 = V_1}$
$\boldsymbol{U_2 = V_2}$
$\boldsymbol{U_3 = V_{\text{center}}}$
$\boldsymbol{U_4 = V_{\text{bias}}}$

The variable simplifiactions are thus:

$$H = 1/L$$
$$K = 2\,\dot{\phi}(6)\,\beta\lambda_N\,/\,4\,\pi^2$$
$$m_{11} = 2\,\dot{\phi}(6)\,d/J$$
$$m_{12} = K/J$$
$$m_{21} = K/L$$
$$m_{22} = R/L$$
$$\alpha = r_C\,m\,g/\left[m\,r_C^2 + I_C\right]$$
$$\gamma = V_{\text{range}}/2\,\theta_{\text{range}}$$
$$\rho = \text{air friction coeff}$$

The second one is “Kappa”, but it looks like “K” in this font... :(

We can make a new State Space model for the complete open loop system directly from the SFG:

$$\dot{X}_1 = X_2$$
$$\dot{X}_2 = \alpha X_1 - \rho X_2 + K X_3 - K X_5$$
$$\dot{X}_3 = -m_{11} X_3 + m_{12} X_4$$
$$\dot{X}_4 = -m_{21} X_3 - m_{22} X_4 - H V_{\text{bias}} + H V_1$$
$$\dot{X}_5 = -m_{11} X_5 + m_{12} X_6$$
$$\dot{X}_6 = -m_{21} X_5 - m_{22} X_6 - H V_{\text{bias}} + H V_2$$

$$Y = \gamma X_1 + V_{\text{center}}$$

$$\begin{pmatrix} \dot{X}_1 \\ \dot{X}_2 \\ \dot{X}_3 \\ \dot{X}_4 \\ \dot{X}_5 \\ \dot{X}_6 \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ \alpha & -\rho & K & 0 & -K & 0 \\ 0 & 0 & -m_{11} & m_{12} & 0 & 0 \\ 0 & 0 & -m_{21} & -m_{22} & 0 & 0 \\ 0 & 0 & 0 & 0 & -m_{11} & m_{12} \\ 0 & 0 & 0 & 0 & -m_{21} & -m_{22} \end{pmatrix} \begin{pmatrix} X_1 \\ X_2 \\ X_3 \\ X_4 \\ X_5 \\ X_6 \end{pmatrix} + \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ H & 0 & 0 & -H \\ 0 & 0 & 0 & 0 \\ 0 & H & 0 & -H \end{pmatrix} \begin{pmatrix} U_1 \\ U_2 \\ U_3 \\ U_4 \end{pmatrix}$$

$$Y = \begin{pmatrix} \gamma & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} X_1 \\ X_2 \\ X_3 \\ X_4 \\ X_5 \\ X_6 \end{pmatrix} + \begin{pmatrix} 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} U_1 \\ U_2 \\ U_3 \\ U_4 \end{pmatrix}$$

MATHEMATICA STUFF

```
Aol = {{0, 1, 0, 0, 0, 0}, {α, -ρ, K, 0, -K, 0}, {0, 0, -m11, m12, 0, 0},
  {0, 0, -m21, -m22, 0, 0}, {0, 0, 0, 0, -m11, m12}, {0, 0, 0, 0, -m21, -m22}}
```

$$\begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ \alpha & -\rho & \mathrm{K} & 0 & -\mathrm{K} & 0 \\ 0 & 0 & -m_{11} & m_{12} & 0 & 0 \\ 0 & 0 & -m_{21} & -m_{22} & 0 & 0 \\ 0 & 0 & 0 & 0 & -m_{11} & m_{12} \\ 0 & 0 & 0 & 0 & -m_{21} & -m_{22} \end{pmatrix}$$

```
Bol = {{0, 0, 0, 0}, {0, 0, 0, 0}, {0, 0, 0, 0}, {H, 0, 0, -H}, {0, 0, 0, 0}, {0, H, 0, -H}}
```

$$\begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ H & 0 & 0 & -H \\ 0 & 0 & 0 & 0 \\ 0 & H & 0 & -H \end{pmatrix}$$

```
Col = {{γ, 0, 0, 0, 0, 0}}
```

$$\begin{pmatrix} \gamma & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

```
Dol = {{0, 0, 1, 0}}
```

$$\begin{pmatrix} 0 & 0 & 1 & 0 \end{pmatrix}$$

```
olssm = StateSpaceModel[{Aol, Bol, Col, Dol}]
```

$$\left(\begin{array}{cccccc|cccc} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \alpha & -\rho & \mathrm{K} & 0 & -\mathrm{K} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -m_{11} & m_{12} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -m_{21} & -m_{22} & 0 & 0 & H & 0 & 0 & -H \\ 0 & 0 & 0 & 0 & -m_{11} & m_{12} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -m_{21} & -m_{22} & 0 & H & 0 & -H \\ \hline \gamma & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{array}\right)^{\mathcal{S}}$$

```
MatrixRank[ControllabilityMatrix[olssm]]
```

6

```
MatrixRank[ObservabilityMatrix[olssm]]
```

4

Oh no! The observability matrix is not of full rank! We have to find out which modes are unobservable. Might as well take a look at the transfer functions:

```
Φol = s * IdentityMatrix[6] - Aol
```

$$\begin{pmatrix} s & -1 & 0 & 0 & 0 & 0 \\ -\alpha & s+\rho & -\mathrm{K} & 0 & \mathrm{K} & 0 \\ 0 & 0 & s+m_{11} & -m_{12} & 0 & 0 \\ 0 & 0 & m_{21} & s+m_{22} & 0 & 0 \\ 0 & 0 & 0 & 0 & s+m_{11} & -m_{12} \\ 0 & 0 & 0 & 0 & m_{21} & s+m_{22} \end{pmatrix}$$

```
TFol = Col.Inverse[Φol].Bol;
```

```
Roots[Denominator[TFol][[1, 1]] == 0, s]
```

$$s = \frac{1}{2}\left(-m_{11} - m_{22} - \sqrt{m_{11}^2 - 2\, m_{22}\, m_{11} + m_{22}^2 - 4\, m_{12}\, m_{21}}\right) \bigvee$$
$$s = \frac{1}{2}\left(-m_{11} - m_{22} - \sqrt{m_{11}^2 - 2\, m_{22}\, m_{11} + m_{22}^2 - 4\, m_{12}\, m_{21}}\right) \bigvee$$
$$s = \frac{1}{2}\left(-m_{11} - m_{22} + \sqrt{m_{11}^2 - 2\, m_{22}\, m_{11} + m_{22}^2 - 4\, m_{12}\, m_{21}}\right) \bigvee$$
$$s = \frac{1}{2}\left(-m_{11} - m_{22} + \sqrt{m_{11}^2 - 2\, m_{22}\, m_{11} + m_{22}^2 - 4\, m_{12}\, m_{21}}\right) \bigvee s = \frac{1}{2}\left(-\sqrt{4\,\alpha + \rho^2} - \rho\right) \bigvee s = \frac{1}{2}\left(\sqrt{4\,\alpha + \rho^2} - \rho\right)$$

Looks like we only have one RHP pole... just like the original system. What is interesting to me is that the observability matrix becomes full rank when the motors have different parameters. This bugs me. I don't know what to do about this for sure, but it seems odd to me...

UPDATE! Measure the Current, or Motor Speed!

I also noticed that when you try to make an LQR controller with the model we made, it wants to apply feedback to the inputs V_{center} and V_{bias}. I think I should just get rid of these, and just remember their meanings via software.

```
Bol = {{0, 0}, {0, 0}, {0, 0}, {H, 0}, {0, 0}, {0, H}}
```

$$\begin{pmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ H & 0 \\ 0 & 0 \\ 0 & H \end{pmatrix}$$

```
Col = {{γ, 0, 0, 0, 0, 0}, {0, 0, 0, 1, 0, 0}}
```

$$\begin{pmatrix} \gamma & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix}$$

```
Dol = {{0, 0}, {0, 0}}
```

$$\begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}$$

```
olssm = StateSpaceModel[{Aol, Bol, Col, Dol}]
```

$$\left(\begin{array}{cccccc|cc} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ \alpha & -\rho & K & 0 & -K & 0 & 0 & 0 \\ 0 & 0 & -m_{11} & m_{12} & 0 & 0 & 0 & 0 \\ 0 & 0 & -m_{21} & -m_{22} & 0 & 0 & H & 0 \\ 0 & 0 & 0 & 0 & -m_{11} & m_{12} & 0 & 0 \\ 0 & 0 & 0 & 0 & -m_{21} & -m_{22} & 0 & H \\ \hline \gamma & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \end{array}\right)^{\mathcal{S}}$$

```
MatrixRank[ObservabilityMatrix[olssm]]
```

6

Hurray! this is a great thing. We can measure the current through just one motor, and get a fully observable system. What if our measurement is the sum of currents through both motors?

```
Col = {{γ, 0, 0, 0, 0, 0}, {0, 0, 0, 1, 0, 1}}
```

$$\begin{pmatrix} \gamma & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 \end{pmatrix}$$

```
olssm = StateSpaceModel[{Aol, Bol, Col, Dol}]
```

$$\left(\begin{array}{cccccc|cc} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ \alpha & -\rho & K & 0 & -K & 0 & 0 & 0 \\ 0 & 0 & -m_{11} & m_{12} & 0 & 0 & 0 & 0 \\ 0 & 0 & -m_{21} & -m_{22} & 0 & 0 & H & 0 \\ 0 & 0 & 0 & 0 & -m_{11} & m_{12} & 0 & 0 \\ 0 & 0 & 0 & 0 & -m_{21} & -m_{22} & 0 & H \\ \hline \gamma & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \end{array}\right)^{\mathcal{S}}$$

```
MatrixRank[ObservabilityMatrix[olssm]]
```

6

Haha! what a gift! This is really easy to do with the setup we made. All we need is something like an INA169, and a tenth ohm power resistor. I think we decided to limit the current drawn by each motor to around 3A, so this will work good.

But, now that we have the combined current measurement, we need an observer for both -- and all the other states if we want our control to be optimal. A good experiment could be to see how little of feedback we need. Simple PD worked with the old model, so it will probably work here. I would like to see if having more feedback about the motor speed and current will make it behave differently. I hope it would, because I spent alot of time in this...