Final Project Report

MEEN 431-500

May 2, 2019

"On my honor as an Aggie, I have neither given nor received unauthorized aid on this academic work"

Kathryn Hall

Jacob Hartzer

Yuming Zhao

Dynamic Model

Three sets of origins and axes were established relating to the PIGA. The first frame, $\{x_0, y_0, z_0\}$, is located at the center of mass of the disk with the x_0 axis always orthogonal to the face of the disk and the y_0z_0 plane is coplanar with the yoke. The next frame, $\{x_1, y_1, z_1\}$, was located in the center of the yoke bearing with the y_1 axis parallel to the motor axis and the z_1 axis along the yoke axis of rotation. A set reference axes were located in the center of mass of the disk when $\theta = 0$. This frame is denoted as $\{x_2, y_2, z_2\}$. The rotation of the yoke and pendulum about the motor was denoted as positive ϕ in the counterclockwise direction when viewed from the top. The rotation of the pendulum was denoted as positive θ from horizontal as shown in *Figure 1*.

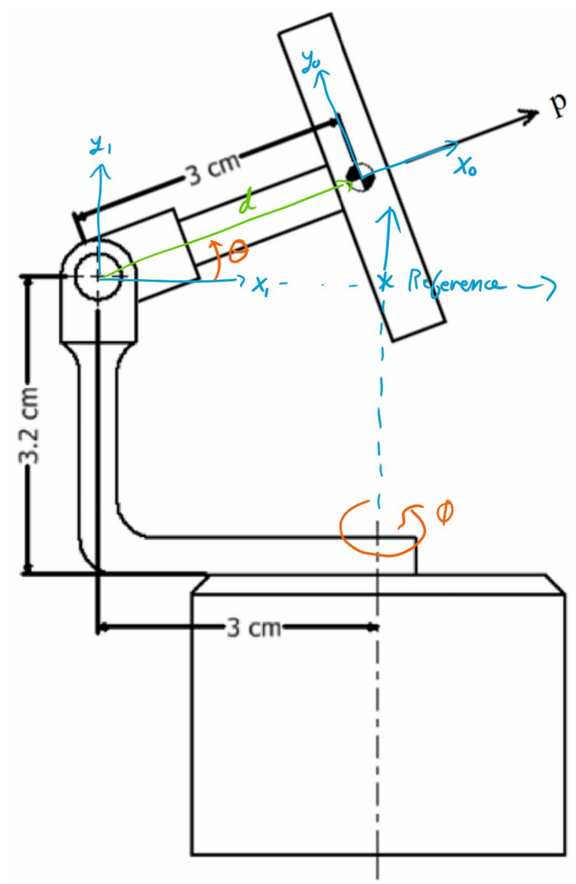

Figure 1. Axis and State Variables of PIGA

To begin calculations relations between the axes, variables for inertia terms, and rotational velocity terms were created about the principle axes. Using the rotating frame formula, the displacement vector of the center of mass from the reference point and the rotation about the motor axis were used to determine the velocity vector of the center of mass. This process was repeated to find the acceleration vector with respect to the reference frame.

The angular momentum was calculated by multiplying the inertia matrix and the angular velocity vector with respect to the $\{x_0, y_0, z_0\}$ frame. By using this angular momentum vector with the rotating frame formula, the change in angular momentum can be equated to the net torque on the system.

This torque was equated to the sum of the moments from reaction forces, while the reaction forces were found from force balancing and the acceleration vector. Using these six equations, two equations of motion can be written. These equations were inputted in a MATLAB script to solve within the *ode15s()* function. All of the previously mentioned calculations can be found in *Appendix A*.

The motor dynamics were calculated by adding armature current as a state variable in the ODE function. The state equation for current included the the electrical dynamics of the motor. The rotational moment of inertia *J* was also added into the angular momentum matrix and the equations of motion were solved again including this value.

When observing the derived differential equations of motions for the system dynamics, it was deemed too complicated to linearize the system. Thus, the equations of motion were left in state - space form.

Torque Control

The open-loop simulation, based off of the given system parameters shown in *Figure 2* below, demonstrates the expected behavior of the PIGA with an instantaneous change in frame acceleration of ± 1 m/s^2. θ represents the rotation of the pendulum and Φ represents the rotation of the motor. Under uniform acceleration, the pendulum gyrates as expected. In this case, the angle of the pendulum arm θ reaches steady state at around -27.5° and the angle of the motor Φ continues to increase indefinitely at a constant rate. This continuous increase of Φ caused by precession in the system and is observable by the constant angular velocity of Φ even after θ has stabilized a constant value. This is the result of the constant angular velocity of the disk at 12000 rpm.

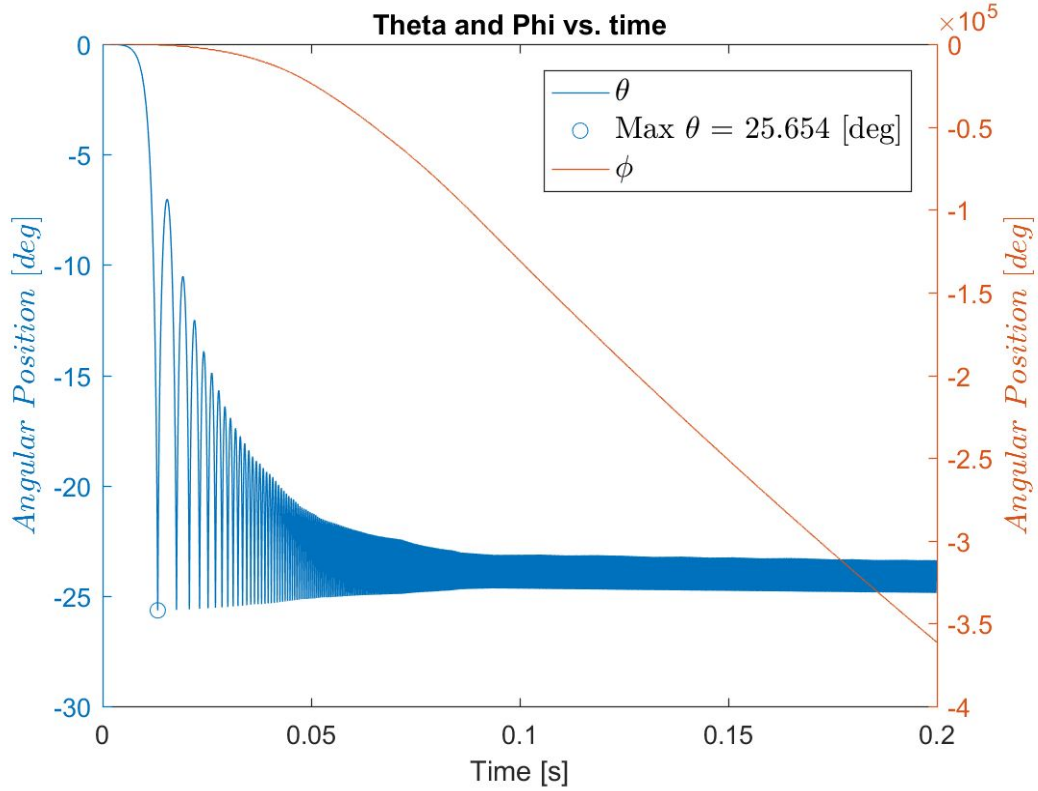

Figure 2. Open Loop Simulation of EoM

In order to control the open-loop simulation to meet the desired design specifications of -3°< θ <3°, no steady state error and a small settling time, the following negative feedback controller gains were implemented shown below as *Table 1*.

Table 1. Torque PD Controller Gains

Controller	**Gain**
P (θ)	70
D (θ)	1.2
P (Φ)	0
D (Φ)	1

The proportional controller gain is needed within the system as the angle of the pendulum arm θ is roughly proportional to the angular acceleration of the motor when implemented as shown in *Figure 3*. Thus, a large proportional gain reduces the steady state error due to a step acceleration input. However, using just a P controller is relatively unstable and causes high-frequency oscillations. To correct for these oscillations, a derivative gain was added, making the controller a PD feedback controller. This derivative gain reduces overshoot and affects the settling time. While the PD torque-controller with the found gains satisfies the maximum error requirement with a 20 m/s step input, there is still a steady-state error as shown in *Figure 3a*. To correct this, an integral gain would be required. The integral controller is unnecessary for the step response and motor dynamics of the system but is vital when modeling tracking ascent profile of the Saturn V rocket.

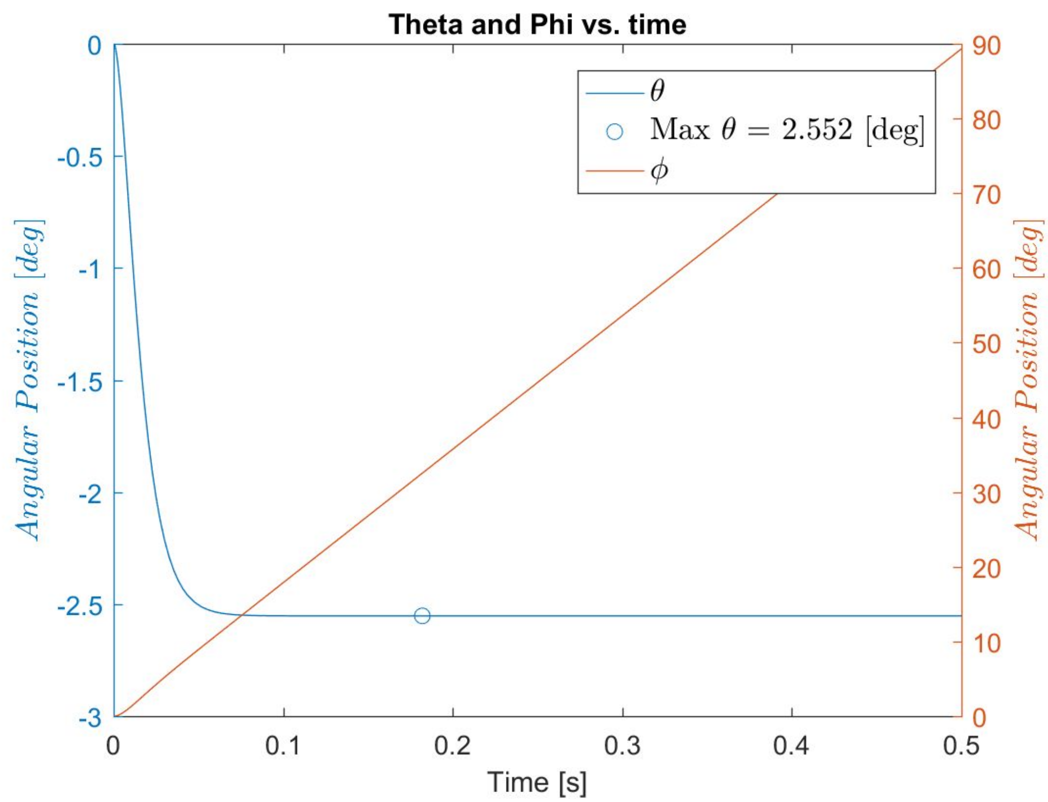

Figure 3a. 20 m/s Step Responses with Torque Controller

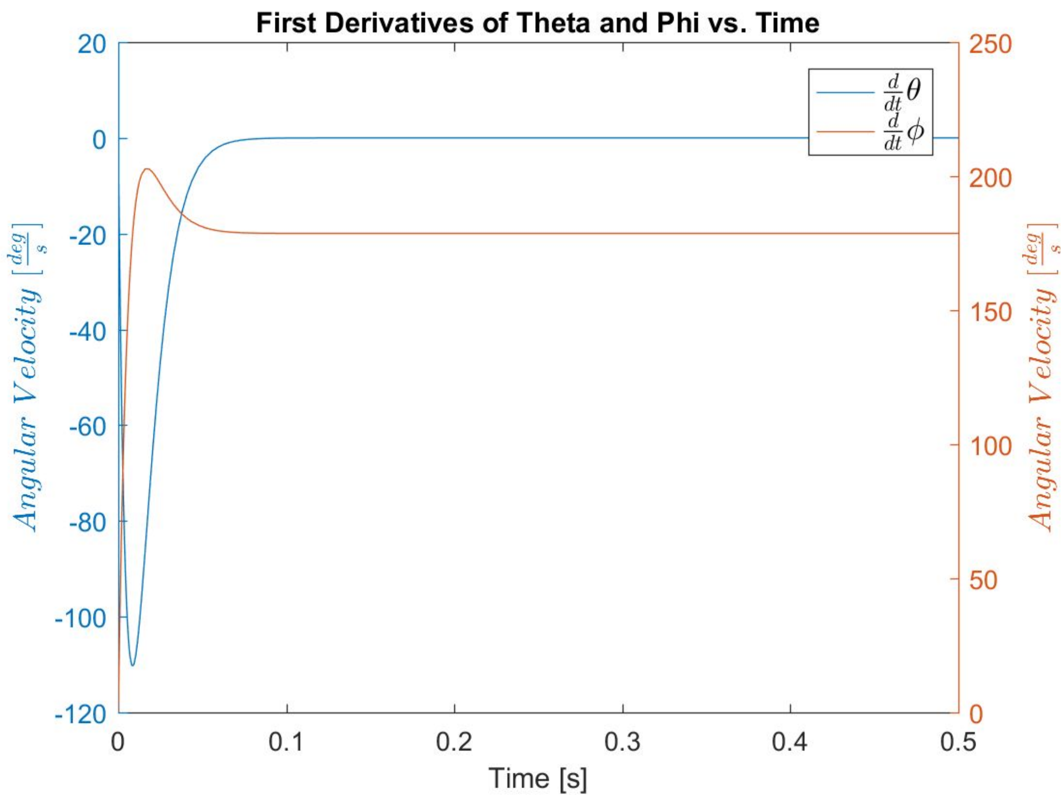

Figure 3b. 20 m/s Step Responses with Torque Controller Derivative

Motor Dynamics

In the previous simulations, torque was the input of the system. By observing the output of the torque controller controller with respect to the equations of motions and the step input, it was possible to find the maximum torque and rotational velocity requirements for the motor. These equations can be seen in the appendix. From these simulations, it was determined the motor for this system should be able to at least provide a torque of 0.029 Nm and voltage of 0.887 V.

From the listed motors, the DP20-10 brush motor from ElectroCraft can be implemented to meet the desired output response of the PIGA controller as it can reach a peak torque of 0.3742 Nm and the required 30.44 rpm. This is far greater than the necessary torque for the system at varying speeds, and therefore gives the system a factor of safety for the motor's abilities. With this in mind, no gear ratios are needed to increase or decrease the rotation between the motor and the pendulum. If a lower torque value is needed, the voltage can be controlled to match the desired torque. A detailed graph demonstrating the torque response of the motor is shown below as *Figure 4*.

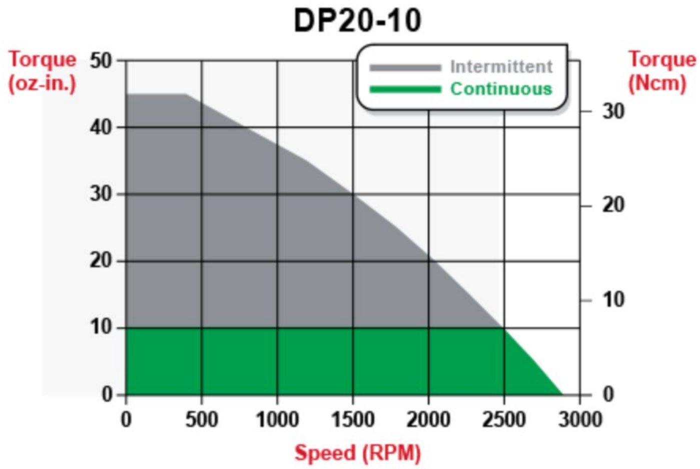

Figure 4. DP20-10 Speed Torque Curve

The DP20-10 has the characteristics listed in *Table 2*. These characteristics were accounted for when augmenting the state-space model to include the motor dynamics. Motor inertia and motor inductance were taken into account with calculations found in the appendixes.

Table 2. DP20-10 Characteristics

Motor Characteristic	**Symbol**	**Value**
Moment of Inertia of the Rotor	J	$1.20*10^{-5}$ kg.m^2
Motor Viscous Friction Constant	b	0 N.m.s
Electromotive Force Constant	k_b	0.035 V/rad/sec
Motor Torque Constant	k_t	0.03531 N.m/Amp
Electric Resistance	R	0.70 Ω
Electric Inductance	L	0.00110 H

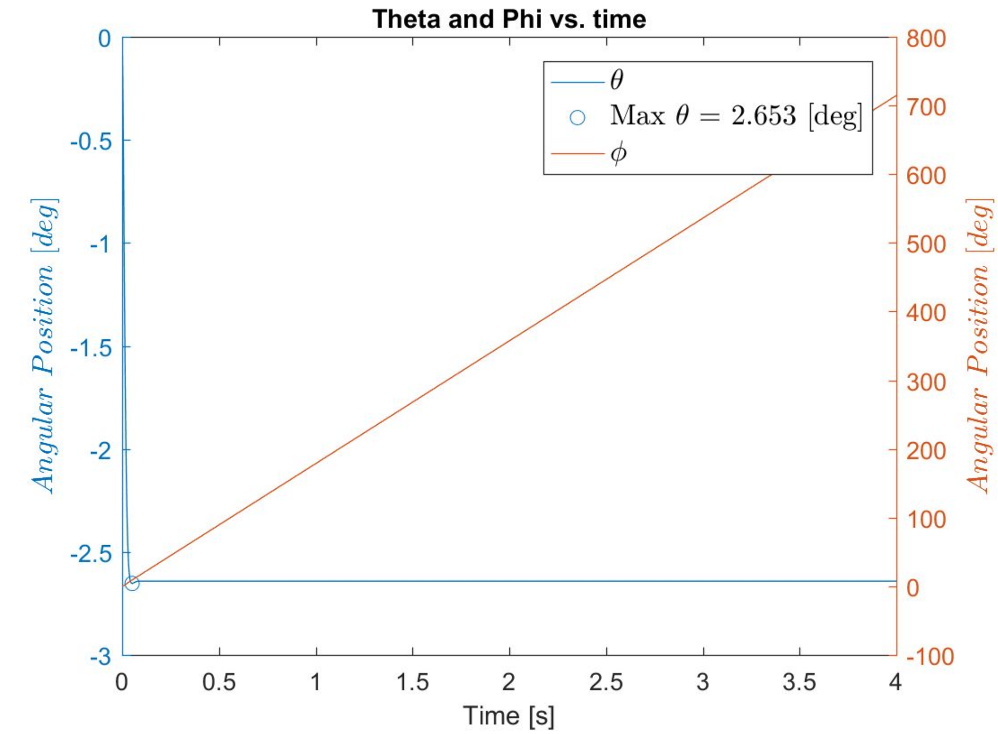

Figure 5a. Step Response with Motor Controller and Motor Dynamics

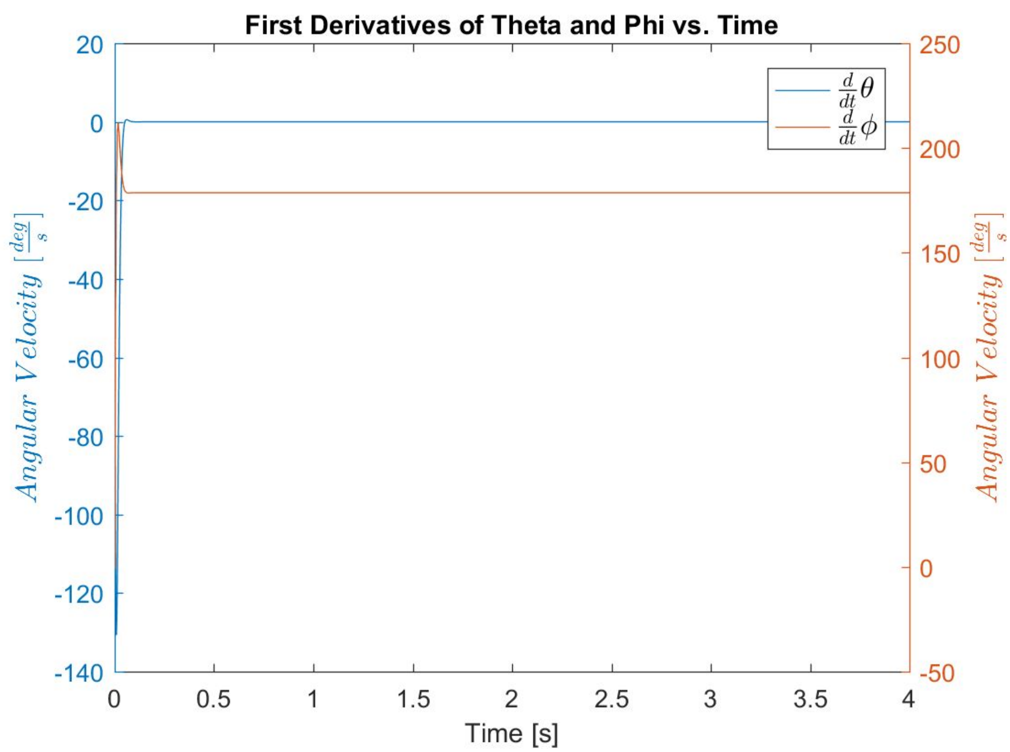

Figure 5b. Step Response with Motor Controller and Motor Dynamics Derivative

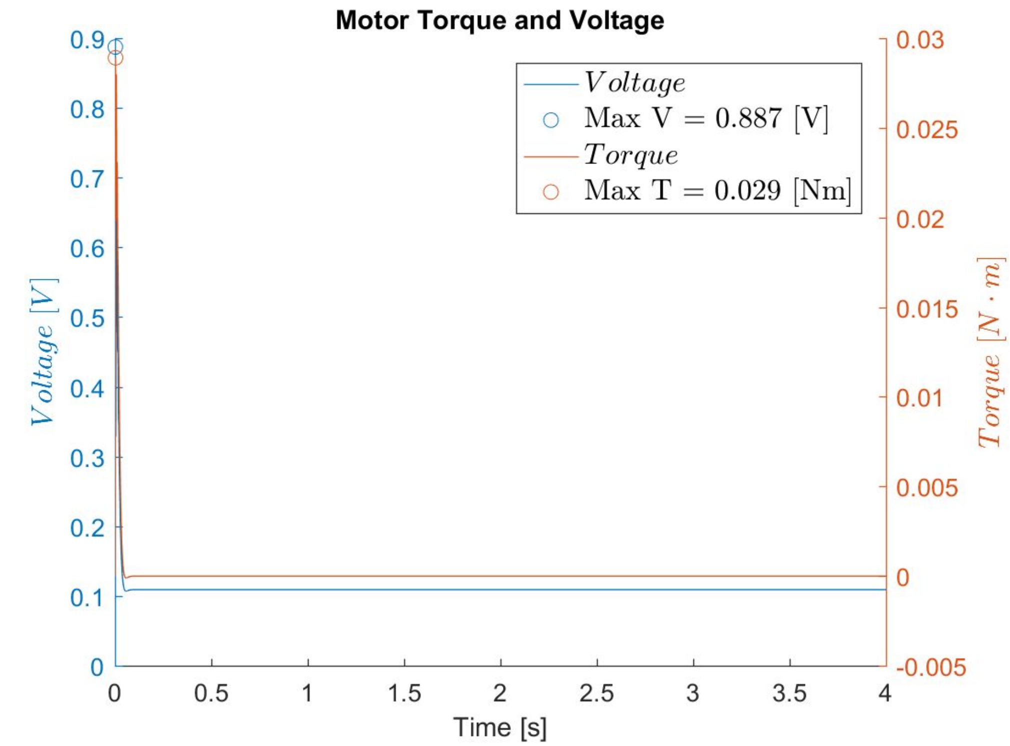

Figure 5c. Step Response with Motor Controller and Motor Dynamics Voltage and Torque

With the augmented state-space model, it was possible to implement a simulation of the voltage-control system and tune the gains in order to meet the set requirements. Despite the need to tune the gains, using the same controller gains as before still satisfies the problem specifications as shown in *Figure 5*. Therefore, the gains used in *Table 1* satisfy the problem specifications when used in a closed loop voltage control system.

Validity and Limitations

The simulations verify that the implemented controller is capable of meeting the problem specifications. The angle of the pendulum arm does not exceed 3° in either rotational direction (maximum of -2.5°), and the settling time is practical. While there is a slight steady state error with respect to the horizontal, it is within the bounds of 3°. The motor control inputs do not exceed the manufacturer limitations as only a rotation of 30 RPM is needed to control this system whereas the motor limitations can reach values greater than 2000 RPM. It should be noted that even though the motor can reach speeds far greater than 30 RPM, it should not be done in excess due to the fact that it will obstruct the effectiveness of the PIGA. The motor should only be done to mitigate the effects of precession to remain within 3° of the horizontal. Gear ratios are not necessary for speed or torque purposes so a 1:1 ratio should be used.

The only significant improvement that can be noted about the controller is the settling time. While a settling time of 4 seconds is not inoperable, the controller would be improved by decreasing this value. This can be done by altering the derivative gain of the controller but would alter the steady state error and angle response of the pendulum armature angle. Additionally, the implementation of a I gain can be used to remove steady state error, as was done in the following section.

Bonus

When modeling the ascent profile of the Saturn V rocket with the prior feedback controller (PD controller), the simulation did not meet the specification of no steady-state error. To counteract this, a PID controller was used instead of a PD controller with gains shown in *Table 3*:

Table 3. PID Controller of Saturn V Rocket Ascent

Controller	Gain
P (θ)	70
I (θ)	150
D (θ)	1.2
P (Φ)	0
D (Φ)	1

Figure 6 shows the simulated response of the Saturn V in its ascent with the PID controller on the PIGA. The figure demonstrates the response of $\frac{d}{dt}\theta$ and $\frac{d}{dt}\Phi$, θ and Φ, and the motor torque and voltage of the PIGA on the Saturn V in its ascent. With the PID controller, it was possible to meet the no steady-state error requirement while also keeping the pendulum angle error low (maximum of 2.5°).

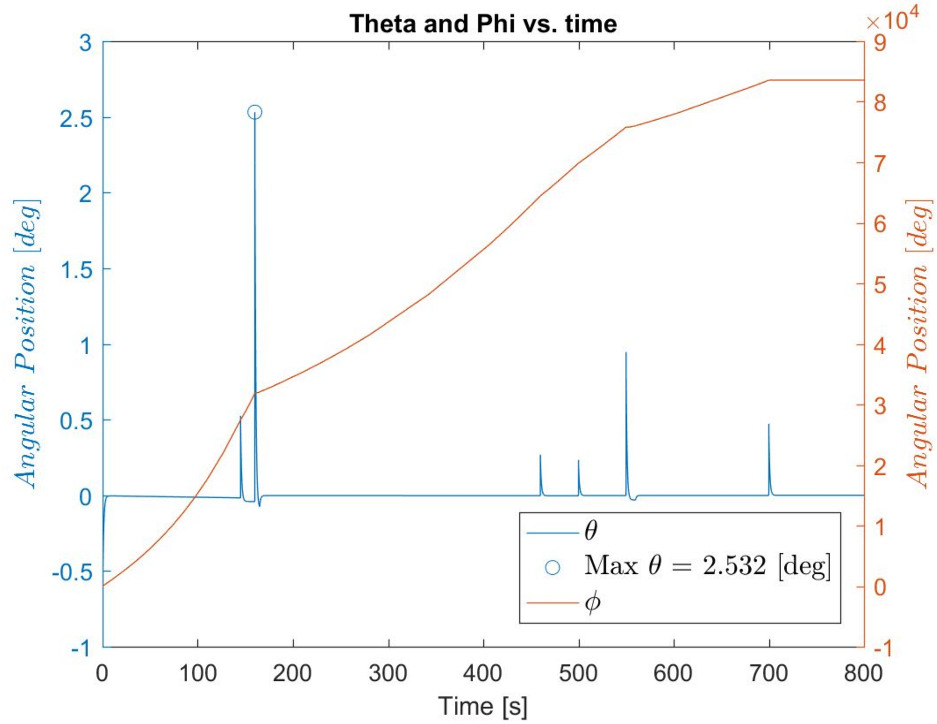

Figure 6a. Ascent Profile of Saturn V Torque and Voltage

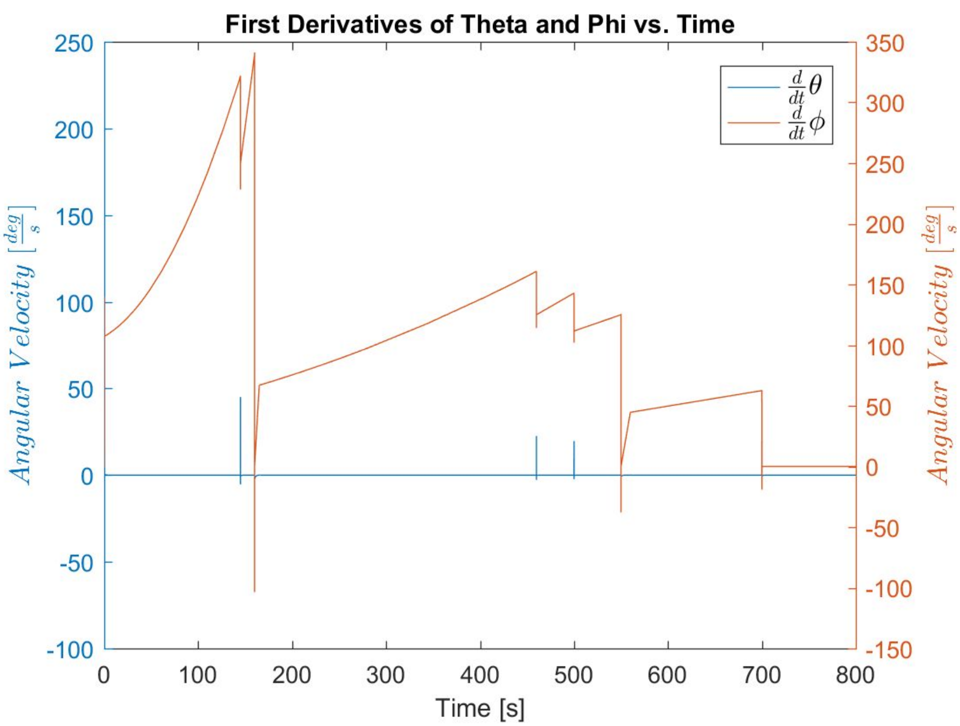

Figure 6b. Ascent Profile of Saturn V Torque and Voltage Derivative

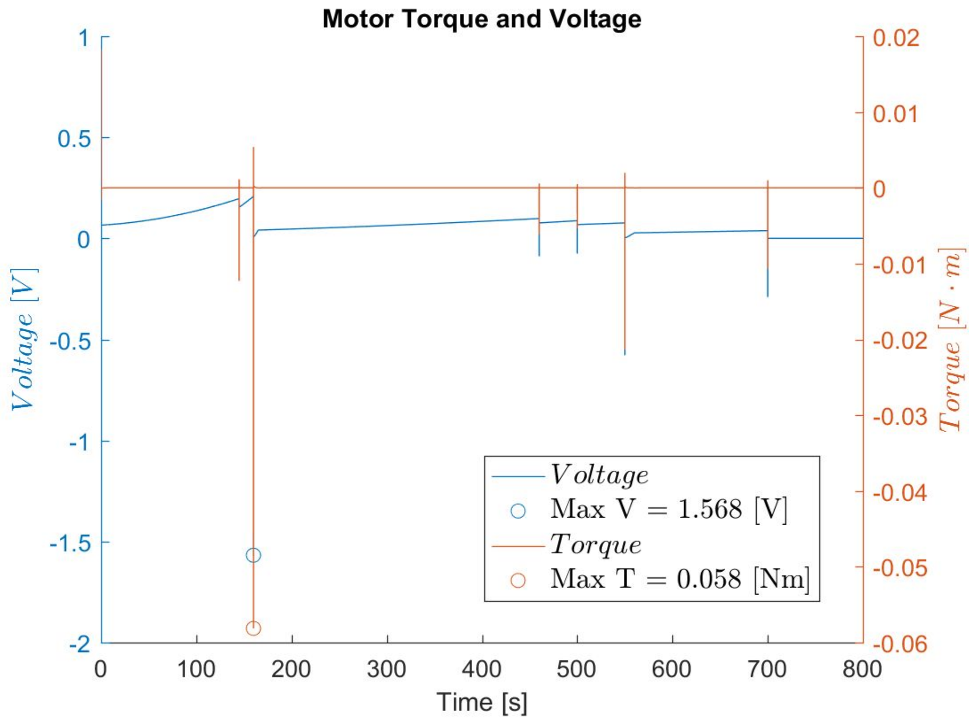

Figure 6c. Ascent Profile of Saturn V Torque and Voltage

Appendices

Appendix A: Derivation Steps

Unit Vectors:

$$\hat{\imath}_1 = \cos(\theta)\,\hat{\imath}_0 - \sin(\theta)\,\hat{\jmath}_0 + 0\hat{k}_0$$
$$\hat{\jmath}_1 = \sin(\theta)\,\hat{\imath}_0 + \cos(\theta)\,\hat{\jmath}_0 + 0\hat{k}_0$$
$$\hat{k}_1 = 0\hat{\imath}_0 + 0\hat{\jmath}_0 + 1\hat{k}_0$$

$$I = I_{xx} = \frac{1}{8}mb^2$$
$$I_0 = I_{yy} = I_{zz} = \frac{1}{16}mb^2$$

Acceleration Vector:

$$\underline{\Omega} = \dot{\phi}\hat{\jmath}_1 + \dot{\theta}\hat{k}_0$$

$$\underline{\Omega} = \dot{\phi}\sin(\theta)\,\hat{\imath}_0 + \dot{\phi}\cos(\theta)\,\hat{\jmath}_0 + \dot{\theta}\hat{k}_0$$

$$\underline{\omega} = \left(\dot{\phi}\sin(\theta) + p\right)\hat{\imath}_0 + \dot{\phi}\cos(\theta)\,\hat{\jmath}_0 + \theta\dot{\hat{k}}_0$$

$$\underline{Q} = \dot{\phi}\hat{\jmath}_0$$

$$\underline{r_0} = \begin{bmatrix} d\cos(\theta) - d \\ d\sin(\theta) \\ 0 \end{bmatrix}$$

$$\underline{\dot{r_0}} = \frac{d}{dt}\left(\underline{r_0}\right) + \underline{Q} \times \underline{r_0}$$

$$\frac{d}{dt}\left(\underline{r_0}\right) = \begin{bmatrix} -\dot{\theta}d\sin(\theta) \\ \dot{\theta}d\cos(\theta) \\ 0 \end{bmatrix}$$

$$\underline{Q} \times \underline{r_0} = \begin{vmatrix} \hat{\imath}_0 & \hat{\jmath}_0 & \hat{k}_0 \\ 0 & \dot{\phi} & 0 \\ d\cos(\theta) - d & \mathrm{d}\sin(\theta) & 0 \end{vmatrix} = \begin{bmatrix} 0 \\ 0 \\ -\dot{\phi}(d\cos(\theta) - \mathrm{d}) \end{bmatrix}$$

$$\underline{\dot{r_0}} = \begin{bmatrix} -\dot{\theta}d\sin(\theta) \\ \dot{\theta}d\cos(\theta) \\ -\dot{\phi}(d\cos(\theta) - \mathrm{d}) \end{bmatrix}$$

$$\underline{\ddot{r_0}} = \frac{d}{dt}\left(\underline{\dot{r_0}}\right) + \underline{Q} \times \underline{\dot{r_0}}$$

$$\frac{d}{dt}\left(\underline{\dot{r_0}}\right) = \begin{bmatrix} -\ddot{\theta} d\sin(\theta) - \dot{\theta}^2 d\cos(\theta) \\ \ddot{\theta} d\cos(\theta) - \dot{\theta}^2 d\sin(\theta) \\ -\ddot{\phi} d(\cos(\theta) - 1) + \dot{\theta}\dot{\phi} d\sin(\theta) \end{bmatrix}$$

$$\underline{Q} \times \underline{\dot{r_0}} = \begin{vmatrix} \hat{\imath}_0 & \hat{\jmath}_0 & \hat{k}_0 \\ 0 & \dot{\phi} & 0 \\ -\dot{\theta} d\sin(\theta) & \dot{\theta} d\cos(\theta) & -\dot{\phi}(d\cos(\theta) - \mathrm{d}) \end{vmatrix} = \begin{bmatrix} \dot{\phi}^2 d(\cos(\theta) - 1) \\ 0 \\ \dot{\phi}\dot{\theta} d\sin(\theta) \end{bmatrix}$$

$$\underline{\ddot{r_0}} = \begin{bmatrix} -\ddot{\theta} d\sin(\theta) - \dot{\theta}^2 d\cos(\theta) + \dot{\phi}^2 d(\cos(\theta) - 1) \\ \ddot{\theta} d\cos(\theta) - \dot{\theta}^2 d\sin(\theta) \\ -\ddot{\phi} d(\cos(\theta) - 1) + 2\dot{\theta}\dot{\phi} d\sin(\theta) \end{bmatrix}$$

Time Derivative of Angular Velocity:

$$\underline{H^o} = \underline{I_0 \omega} = \begin{bmatrix} I & 0 & 0 \\ 0 & I_0 & 0 \\ 0 & 0 & I_0 \end{bmatrix} \begin{bmatrix} \dot{\phi}\sin(\theta) + p \\ \dot{\phi}\cos(\theta) \\ \theta \end{bmatrix} = \begin{bmatrix} I(\dot{\phi}\sin(\theta) + p) \\ I_0\dot{\phi}\cos(\theta) \\ I_0\dot{\theta} \end{bmatrix}$$

$$\frac{d}{dt}\left(\underline{H^o}\right) = \frac{d}{dt}\begin{bmatrix} I(\dot{\phi}\sin(\theta) + p) \\ I_0\dot{\phi}\cos(\theta) \\ I_0\theta \end{bmatrix} = \begin{bmatrix} \ddot{\phi} I\sin(\theta) + \dot{\theta}\dot{\phi} I\cos(\theta) \\ \ddot{\phi} I\cos(\theta) - \dot{\theta}\dot{\phi} I\sin(\theta) \\ I_0\ddot{\theta} \end{bmatrix}$$

$$\underline{\Omega} \times \underline{H^o} = \begin{vmatrix} \hat{\imath}_0 & \hat{\jmath}_0 & \hat{k}_0 \\ \dot{\phi}\sin(\theta) & \dot{\phi}\cos(\theta) & \dot{\theta} \\ I(\dot{\phi}\sin(\theta) + p) & I_0\dot{\phi}\cos(\theta) & I_0\theta \end{vmatrix} = \begin{bmatrix} 0 \\ \dot{\theta}\dot{\phi}(I - I_0)\sin(\theta) - \dot{\theta} Ip \\ \dot{\phi}^2(I_0 - I)\sin(\theta)\cos(\theta) - \dot{\phi} Ip\cos(\theta) \end{bmatrix}$$

$$\underline{\dot{H}^o} = \frac{d}{dt}\left(\underline{H^o}\right) + \underline{\Omega} \times \underline{H^o} = \begin{bmatrix} \ddot{\phi} I\sin(\theta) + \dot{\theta}\dot{\phi} I\cos(\theta) \\ \ddot{\phi} I\cos(\theta) + \dot{\theta}\dot{\phi}(I - 2I_0)\sin(\theta) - \dot{\theta} Ip \\ I_0\ddot{\theta} + \dot{\phi}^2(I_0 - I)\sin(\theta)\cos(\theta) - \dot{\phi} Ip\cos(\theta) \end{bmatrix}$$

Force and Moment Balance:

$$\sum F = ma$$

$$R_x = m\ddot{x} = m(-\ddot{\theta} d\sin(\theta) - \dot{\theta}^2 d\cos(\theta) + \dot{\phi}^2 d(\cos(\theta) - 1))$$

$$R_y = m\ddot{y} + ma = m(a + \ddot{\theta} d\cos(\theta) - \dot{\theta}^2 d\sin(\theta))$$

$$R_z = m\ddot{z} = m(-\ddot{\phi} d(\cos(\theta) - 1) + 2d\,\dot{\theta}\dot{\phi}\sin(\theta))$$

$$\sum M = \underline{\dot{H}}^o$$

$$\underline{\dot{H}}^o{}_x = M_x\cos(\theta) + M_y\sin(\theta) = \ddot{\phi}I\sin(\theta) + \dot{\theta}\dot{\phi}I\cos(\theta)$$

$$\underline{\dot{H}}^o{}_y = -M_x\sin(\theta) + M_y\cos(\theta) + R_z d = \ddot{\phi}I\cos(\theta) + \dot{\theta}\dot{\phi}(I - 2I_0)\sin(\theta) - \dot{\theta}Ip$$

$$\underline{\dot{H}}^o{}_z = R_x d\sin(\theta) - R_y d\cos(\theta) = I_0\ddot{\theta} + \dot{\phi}^2(I_0 - I)\sin(\theta)\cos(\theta) - \dot{\phi}Ip\cos(\theta)$$

Equations of Motion:

$$\begin{aligned} m(-\ddot{\theta}d\sin(\theta) - \dot{\theta}^2 d\cos(\theta) + \dot{\phi}^2 d(\cos(\theta) - 1))d\sin(\theta) - m(a + \ddot{\theta}d\cos(\theta) - \dot{\theta}^2 d\sin(\theta))d\cos(\theta) \\ = I_0\ddot{\theta} + \dot{\phi}^2(I_0 - I)\sin(\theta)\cos(\theta) - \dot{\phi}Ip\cos(\theta) \end{aligned}$$

$$\ddot{\theta} = \dot{\phi}^2\frac{md^2\sin(\theta)(\cos(\theta) - 1) - \sin(\theta)\cos(\theta)(I_0 - I)}{I_0 + md^2} + \dot{\phi}\frac{Ip\cos(\theta)}{I_0 + md^2} - a\frac{md\cos(\theta)}{I_0 + md^2}$$

$$\tan(\theta)\underline{\dot{H}}^o{}_x = M_x\sin(\theta) + M_y\sin(\theta)\tan(\theta) = \ddot{\phi}I\sin(\theta)\tan(\theta) + \dot{\theta}\dot{\phi}I\sin(\theta)$$

$$\tan(\theta)\underline{\dot{H}}^o{}_x + \underline{\dot{H}}^o{}_y$$

$$\begin{aligned} M_y(\sin(\theta)\tan(\theta) + \cos(\theta)) + R_z d \\ = \ddot{\phi}I\sin(\theta)\tan(\theta) + \dot{\theta}\dot{\phi}I\sin(\theta) + \ddot{\phi}I\cos(\theta) + \dot{\theta}\dot{\phi}(I - 2I_0)\sin(\theta) - \dot{\theta}Ip \end{aligned}$$

$$\begin{aligned} M_y(\sin(\theta)\tan(\theta) + \cos(\theta)) + md(-\ddot{\phi}d(\cos(\theta) - 1) + 2d\,\dot{\theta}\dot{\phi}\sin(\theta)) \\ = \ddot{\phi}I\sin(\theta)\tan(\theta) + \dot{\theta}\dot{\phi}I\sin(\theta) + \ddot{\phi}I\cos(\theta) + \dot{\theta}\dot{\phi}(I - 2I_0)\sin(\theta) - \dot{\theta}Ip \end{aligned}$$

$$\begin{aligned} \ddot{\phi} = M_y\frac{\sin(\theta)\tan(\theta) + \cos(\theta)}{I\sin(\theta)\tan(\theta) + I_0\cos(\theta) + md^2(\cos(\theta) - 1)} \\ + \dot{\theta}\dot{\phi}\frac{2(\mathrm{m}d^2\sin(\theta) - \sin(\theta)(I - I_0))}{I\sin(\theta)\tan(\theta) + I_0\cos(\theta) + md^2(\cos(\theta) - 1)} \\ + \dot{\theta}\frac{Ip}{I\sin(\theta)\tan(\theta) + I_0\cos(\theta) + md^2(\cos(\theta) - 1)} \end{aligned}$$

Motor Dynamics:

$$\ddot{\phi} = K_T i \frac{\sin(\theta)\tan(\theta) + \cos(\theta)}{(I+J)sin(\theta)tan(\theta) + (I_0+J)cos(\theta) + md^2(cos(\theta)-1)} +$$
$$\dot{\theta}\dot{\phi}\frac{2(md^2 sin(\theta) - sin(\theta)(I-I_0))}{(I+J)sin(\theta)tan(\theta) + (I_0+J)cos(\theta) + md^2(cos(\theta)-1)} +$$
$$\dot{\theta}\frac{Ip}{(I+J)sin(\theta)tan(\theta) + (I_0+J)cos(\theta) + md^2(cos(\theta)-1)} +$$

$$\frac{d}{dt}i = -\frac{K_e}{L}\dot{\phi} - \frac{R}{L}i + \frac{1}{L}V$$

Appendix B: Simulation Code

Main.m

```
y0 = zeros(6,1);
tspan = [0 5];
[t,y] = ode15s(@odefun_PIGA,tspan,y0);
```

Output Variables

```
Kt  = 0.03531;                        % [ N m / A]
V = zeros(size(y,1),1);
T = zeros(size(y,1),1);
for i = 1:size(y,1)
    V(i) = controller_PIGA(y(i,:)');    % [ V ]
    T(i) = y(i,5) * Kt;                % [ A ] * [ N m / A ]
end
[maxV,idV] = max(abs(V));
[maxT,idT] = max(abs(T));
[maxTheta,idTheta] = max(abs(y(:,1)*180/pi));
```

Plotting

Plots the results of the ODE solver

```
figure(1)
yyaxis left
hold on
plot(t,y(:,1)*180/pi);
scatter(t(idTheta),y(idTheta,1)*180/pi)
```

```
hold off
MTheta = sprintf('Max $\\theta$ = %0.3f [deg]',maxTheta);
title('Theta and Phi vs. time')
xlabel('Time [s]')
ylabel('$Angular\ Position\ [deg]$','Interpreter','latex','FontSize',12)

yyaxis right
plot(t,y(:,3)*180/pi);
ylabel('$Angular\ Position\ [deg]$','Interpreter','latex','FontSize',12)
legend({'$\theta$',MTheta,'$\phi$'},'Interpreter','latex','FontSize',12)

figure(2)
yyaxis left
plot(t,y(:,2)*180/pi);
title('First Derivatives of Theta and Phi vs. Time')
xlabel('Time [s]')
ylabel('$Angular\ Velocity\ [\frac{deg}{s}]$','Interpreter','latex','FontSize',12)

yyaxis right
plot(t,y(:,4)*180/pi);
ylabel('$Angular\ Velocity\ [\frac{deg}{s}]$','Interpreter','latex','FontSize',12)
legend({'$\frac{d}{dt}\theta$','$\frac{d}{dt}\phi$'},'Interpreter','latex','FontSize',12)

figure(3)
yyaxis left
hold on
plot(t,V);
scatter(t(idV),V(idV));
hold off
title('Motor Torque and Voltage')
xlabel('Time [s]')
ylabel('$Voltage\ [V]$','Interpreter','latex','FontSize',12)

yyaxis right
hold on
plot(t,T);
scatter(t(idT),T(idT));
hold off
MV = sprintf('Max V = %0.3f [V]',maxV);
MT = sprintf('Max T = %0.3f [Nm]',maxT);
ylabel('$Torque\ [N \cdot m]$','Interpreter','latex','FontSize',12)
legend({'$Voltage$',MV,'$Torque$',MT},'Interpreter','latex','FontSize',12)
```

controller_PIGA.m

```
function [ V ] = controller_PIGA( y )
    % odefun_PIGA: Controller function for input voltage
    % INPUTS:
    %       y(1)        = theta             [ rad ]
    %       y(2)        = d_theta/dt        [ rad / s ]
    %       y(3)        = phi               [ rad ]
    %       y(4)        = d_phi/dt          [ rad / s ]
    %       y(5)        = i                 [ amps ]
    %       y(6)        = theta_int         [ rad * s ]
    %
    % OUTPUTS:
    %       V           = motor voltage     [ V ]

    K = zeros(6,1);

    K(1)    =   -70;
    K(2)    =  -1.2;
    K(3)    =     0;
    K(4)    =    -1;
    K(5)    =     0;
    K(6)    =  -150;

    V = sum(y.*K);
end
```

odefun_PIGA.m

```
function dydt = odefun_PIGA(t,y)
    % odefun_PIGA: Solves the ODE functions of the PIGA system
    % INPUTS:
    %       y(1)        = theta             [ rad ]
    %       y(2)        = d_theta/dt        [ rad / s ]
    %       y(3)        = phi               [ rad ]
    %       y(4)        = d_phi/dt          [ rad / s ]
    %       y(5)        = i                 [ amps ]
    %       y(6)        = theta_int         [ rad * s ]
    %
    % OUTPUTS:
    %       dydt(1)     = d_theta/dt        [ rad / s ]
    %       dydt(2)     = d^2 theta /dt^2   [ rad / s^2 ]
```

```
    %       dydt(3)     = d_phi/dt          [ rad / s ]
    %       dydt(4)     = d^2 phi /dt^2     [ amps ]
    %       dydt(5)     = d_i/dt            [ A / s ]
    %       dydt(6)     = theta             [ rad ]
```

Ascent Profiles

```
    a   = 20;                       % [m/s]
%     a   = SV_Ascent(t);             % [m/s]
```

Pendulum Constants

```
    m   = 50./1000;                 % [kg]
    b   = 3.5./100;                 % [m]
    p   = 12000.*2.*pi./60;         % [rad/s]
    d   = 3./100;                   % [m]
```

Motor Constants

```
    Ke  = 0.035;                    % [ VDC / RPM]
    Kt  = 0.03531;                  % [ N m / A]
    R   = 0.70;                     % [ Ohms ]
    L   = 0.00110;                  % [ H ]
```

Derived Values

```
    Io  = 1/16*m*b^2;               % [kg m^2]
    I   = 1/8*m*b^2;                % [kg m^2]
    J   = 1.201e-05;                % [kg m^2]
```

Controller

```
    V  = controller_PIGA( y );      % [N*m]
```

State-Space Equations

```
    dydt = zeros(size(y,1),size(y,2));

    dydt(1) = y(2);
    dydt(2) =  (y(4).^2.*(m.*d.^2.*sin(y(1)).*(cos(y(1))-1)-sin(y(1)).*cos(y(1)).*(Io-I))+...
                y(4).*(I.*p.*cos(y(1)))-...
                a.*(m.*d.*cos(y(1)))).*/...
                (Io+m.*d.^2);
```

```
    dydt(3) = y(4);
    dydt(4) =  (y(5)*Kt*(cos(y(1))+sin(y(1))*tan(y(1)))+...
                y(2)*y(4)*2*(m*d^2*sin(y(1))-sin(y(1))*(I-Io))+...
                y(2)*(I*p))/...
                ((I+J)*sin(y(1))*tan(y(1))+(Io+J)*cos(y(1))+m*d^2*(cos(y(1))-1));

    dydt(5) = -Ke/L*y(4) - R/L*y(5) + V/L;
```

Added Integral Control

```
    dydt(6) = y(1);

end
```

SV_Ascent.m

```
function g=SV_Ascent(t)
  if t<0
    g=0;
  elseif t<145
    g=(797E-6).*t.^2+(0.05).*t+12;
  elseif t<160
    g=(0.667).*(t-145)+28;
  elseif t<165
    g=(1.5).*(t-160);
  elseif t<460
    g=(32.9E-6).*(t-165).^2+(25.9E-3).*(t-165)+7.5;
  elseif t<500
    g=(0.05).*(t-460)+14;
  elseif t<550
    g=(0.03).*(t-500)+12.5;
  elseif t<560
    g=(0.5).*(t-550);
  elseif t<700
    g=(14.3E-3).*(t-560)+5;
  else
    g=0;
  end
end
```