Pulley Raising a Mass (no Friction and no Gravity)

$$\left(\frac{mr^2}{2}+Mr^2\right)\ddot{\theta}=T-\cancel{f}^{0}+\cancel{Mgr}^{0}$$

$$\left(\frac{mr^2}{2}+Mr^2\right)\ddot{\theta}=T$$

$$\ddot{\theta}=\frac{T}{\left(\frac{mr^2}{2}+Mr^2\right)}$$

State Space Model:

$$x=\begin{bmatrix}\theta\\ \dot{\theta}\end{bmatrix}$$

$$\frac{dx}{dt}=\begin{bmatrix}0 & 1\\ 0 & 0\end{bmatrix}x+\begin{bmatrix}0\\ \left(\frac{mr^2}{2}+Mr^2\right)^{-1}\end{bmatrix}T$$

where T is our input torque.

The corresponding transfer function is:

$$\Theta s^2 = \frac{T}{\left(\frac{mr^2}{2} + Mr^2\right)}$$

$$G(s) = \frac{\Theta}{T} = \frac{1}{\left(\frac{mr^2}{2} + Mr^2\right)s^2}$$

In Matlab we could represent this system as either a state space model or an equivalent transfer function model. To represent it as state space, we could initialize the function like this:

A= [0 1; 0 0];

B= [0; 1/((0.5*m*(r^2))+(M*(r^2))];

C= [1, 0]; %we aren't really interested in an output but we can set y=theta like this

sys=ss[A,B,C,0];

Alternatively we could represent the system through its state space representation like this:

G=tf([1],[((0.5*m*(r^2))+(M*(r^2)) 0 0]); %note that these numbers represent the coefficients of the numerator and denominator respectively

Furthermore, we can transfer between the two like this:

sys_tf=tf(sys)

g_ss=ss(G)

Another helpful thing to know is that we can represent the PID controller itself as a transfer function like this:

pid_tf=tf([kd kp ki],[0 1 0]);

this is useful if we want to convolve the PID controller with the transfer function of our system. If you remember MAE146b, in doing this we use unity feedback. A useful Matlab command in this case is feedback() which will allow us to get a transfer function of our controlled system by setting:

cloop=feedback(G*pid_tf,1) %where g*pid_tf represents the controlled feed-forward branch and the one represents the gain on the feedback branch

finally, we can check if our PID values truly control our system by running:

step(cloop); %this shows a graphical representation of our system's step response

other useful controls Matlab commands that you should learn on your own:

rlocus()

nyquist()

pid() and pidstd()

sisotool()

simulink

Other useful tutorials:

http://www.engin.umich.edu/group/ctm/PID/PID.html