For the last lab we will be building a speed controller for the three disk system using state space techniques. Since there is a lot of fast moving parts in this lab we will take a stepped approach, starting with a speed controller for a single disk, then working our way up to the three disk system.

Single Disk Speed Control

To begin, let's return to the single disk system, setup in our typical configuration of four weights each spaced 9 cm from the center of the disk, and a friction coefficient of $b = 0.005$.

E1. Build up a state space model of the single disk system where the input is the voltage going into the motor controller and the output is the disk's angular position. The state variables should be

$$x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} \theta_1 \\ \dot{\theta}_1 \end{bmatrix}$$

E2. Use the state space block in Simulink to implement your plant in simulation.

A natural first iteration to a speed controller is to simply use a ramp function as your reference signal and feedback the disk position. Lets try this.

E3. Compute the characteristic equation of the closed loop system with feedback gain K. Compute K such that the system has a damping ratio of 0.2.

E4. What is the expected steady state error to a ramp input? Does this really matter in a speed control system?

E5. Simulate the feedback system with a ramp input. Use a ramp slope of 4π. Implement the controller on the TDS and compare the results

What we are really interested in is the speed of the disk, we are using the ramp function to try and tell it how fast to go. Is this a good way of going about it? Think of a speed controller in a car that comes upon a hill, this can be modeled as a disturbance and will increase the position error. What happens at the top of the hill? Lets investigate. First we'll need to build an observer in order to get an estimate of the velocity.

E6. Build an observer of the single disk system with poles at $s_{1,2} = -15, -20$. Be sure to verify that the system is observable, then the MATLAB function *place* can be used to pick the observer gains.

E7. Implement the observer in simulation and on the physical system. Test both to ensure they are working correctly.

Now let's "simulate" the hill on the physical system.

E8. After any oscillations have settled out, pinch the top spring with your fingers, this is our hill, an extra torque to over come. What happens when you release your fingers (get to the top of the hill). Plot the velocity of the disk and explain what is going on.

Hopefully this little demonstration will exemplify why a controller based on the velocity might be better. So let's do that. Recall for state feedback

$$\begin{aligned} \dot{x} &= Ax + Bu \\ u &= -Kx. \end{aligned} \tag{1}$$

Therefore the closed loop system is

$$\dot{x} = [A - BK]x = A_c x$$

and the characteristic equation is

$$\chi(s) = \det(sI - A_c)$$

E9. Compute symbolically the characteristic equation of the state feedback system, with $K = [k_p\, k_v]$.

E10. Since we do not care so much about the position (that's what got us into trouble previously) set $k_p = 0$, and compute the k_v value such that there is a pole at the origin and one at $s = -5$

Now we would like to add a reference input such that the control is

$$u = -K(x - \alpha x_{ref})$$

where the constant, α, is a gain on the reference in order to drive the steady state error to zero. The question now is how do we find α? Since we do not care about the position of the disk let's look at the dynamic equation for the velocity of the disk x_2,

$$\dot{x}_2 = \frac{-b}{J_1} x_2 - \frac{K_m}{J_1} k_v (x_2 - \alpha x_{2_{ref}}) \tag{2}$$

At steady state we do not expect the velocity to be changing, we therefor set (2) equal to zero and solve for α.

E11. Implement the state feedback control design with reference input in your simulation and on the physical model using the observer state. Simulate the system first to ensure stability. Then try it out on the TDS.

E12. Repeat the hill experiment on the TDS, how do the results differ from the position feedback method.

Multi Disk Speed Control

As we add disks to the system we will need to make changes to the controller in order to compensate for the coupling between the disks. Lets begin by adding a disk to the system and recording the response.

E13. Add a second disk to the top position with weights at 9 cm. Update your system model and observer to reflect the changes, keep the first disk position as the output. Put the new observer poles at $(-40,\ -35\ -\ 30,\ -25)$. NOTE: by putting the second disk at the top position the spring constant will be halved.

E14. Using the controller you designed for the single disk use a pulsed signal for your velocity reference signal that changes between $8\pi\, rad/s$ and $0\, rad/s$ every 10 seconds. Record the results to be compared with future controllers. NOTE: You will have to use the demux block to pull off the disk 1 position and velocity from the estimated state vector, also use the mux block to build the reference signal with the pulsed velocity.

Now let's see if we can improve upon the design. Recall that we would like to specify a speed at which both disks will spin. We've already witnessed the negative effects of using the position feedback so we would like to avoid this. The problem is as we add disks the order of the system increases making the simple calculations we did on the single disk system to specify the feedback gain difficult.

A good idea might be to change the state variables that we are concerned with and design a controller using the resulting transformed system. So how do we do this? Well let's say that we specify how fast disk 1 spins then regulate the relative position and speed of the disk two to zero. We are therefore interested in

$$\begin{aligned} \phi &= \theta_2 - \theta_1 \\ \dot{\phi} &= \dot{\theta}_2 - \dot{\theta}_1, \end{aligned}$$

and would like our state vector to be $\bar{x} = [\theta_1,\ \dot{\theta}_1,\ \phi,\ \dot{\phi}]^T$. Now how do we write the state equations in terms of the new coordinates? Since the system is linear this turns out to be fairly easy. First note that the new system state, $\bar{x}$, can be related to the old state, x through the following transformation

$$\bar{x} = Tx \tag{3}$$

where

$$T = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ -1 & 0 & 1 & 0 \\ 0 & -1 & 0 & 1 \end{bmatrix}$$

Taking the derivative of (3) and plugging in (1) we find the transformed system is

$$\begin{aligned} \dot{\bar{x}} &= TAT^{-1}\bar{x} + TBu && (4) \\ y &= CT^{-1}\bar{x} && (5) \end{aligned}$$

and we label the new state matrices as follows,

$$\bar{A} = TAT^{-1}, \quad \bar{B} = TB, \quad \bar{C} = CT^{-1}.$$

We can now design a controller for the transformed system by using state feedback, $u = -\bar{K}\bar{x}$, and placing the poles of $(\bar{A} - \bar{B}\bar{K})$.

E15. Compute the state matrices $\bar{A}$, $\bar{B}$, and $\bar{C}$. Using the place command in MATLAB, find the feedback gain $\bar{K}$ such that the system poles are at $(0, -5, -10, -15)$, recall we want a pole at the origin to effectively ignore the position of disk 1.

With the controller in hand we need to update our estimator.

E16. Show that if the estimated state of the transformed dynamics is given by $\tilde{x} = T\hat{x}$, then

$$\dot{\tilde{x}} = TAT^{-1}\tilde{x} + TBu + TL(y - \hat{y}).$$

E17. Update your observer according to the above equation and implement the state feedback control. Use the simulation to correct your steady state offset.

E18. Run your new controller on the TDS with two disks.

For the last part of the lab use what you learned to design a feedback controller for the entire three disk system.

E19. design and implement a speed controller for the 3 disk system.