Christopher Lum
lum@uw.edu

Lecture 09c
Practical Implementation Issues with a Full State Feedback Controller

YouTube video entitled ‘Practical Implementation Issues with a Full State Feedback Controller’ covering this is located at https://youtu.be/9vCTokJ5RQ8.

Outline

-Introduction
-Plant Model
-Control Saturation Issues
-Inability to Measure Full State of System

Introduction

In our previous lecture, we investigated the concept of full state feedback control. We showed that if the system was controllable, then full state feedback control has the power to move the eigenvalues of the closed loop system to any arbitrary location. While this is mathematically feasible, as engineers, we should stop and ask what is the price we pay for this ability? As Voltaire and more recently Spider Man’s uncle Ben Parker once said “with great power comes great responsibility”. We would investigate if there are any practical implementation issues that may arise from our choices of closed loop pole locations.

Plant Model

Let us investigate applying full state feedback control to a DC motor. If you’d like to see a more detailed description of the system and derivation of the associated state space representation please check out this other video. (***NEED YOUTUBE URL***),

Example: DC Motor System
Recall that the state space representation of the DC motor system

$$\dot{\overline{x}}(t) = A\,\overline{x}(t) + B\,\overline{u}(t) \qquad \textbf{(Eq.1)}$$

$$\overline{y}(t) = C\,\overline{x}(t) + D\,\overline{u}(t)$$

where $A = \begin{pmatrix} 0 & 1 & 0 \\ 0 & -\frac{c}{J_m} & \frac{K_T}{J_m} \\ 0 & -\frac{K_V}{L_a} & -\frac{R_m+R}{L_a} \end{pmatrix}$ $\quad B = \begin{pmatrix} 0 & 0 \\ 0 & -\frac{1}{J_m} \\ \frac{1}{L_a} & 0 \end{pmatrix}$

$$C = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \qquad D = \begin{pmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{pmatrix}$$

$$\overline{x}(t) = \begin{pmatrix} \theta(t) \\ \dot{\theta}(t) \\ i(t) \end{pmatrix} \qquad \overline{u}(t) = \begin{pmatrix} V_a(t) \\ T_L(t) \end{pmatrix}$$

For control purposes, we only have control of the armature voltage, not the disturbance torque, so our model of the system is reduced to

$$\dot{\overline{x}}(t) = A\,\overline{x}(t) + B_1\,u_1(t) \qquad \textbf{(Eq.2)}$$

$$\overline{y}(t) = C\,\overline{x}(t) + D_1\,u_1(t)$$

where $A = \begin{pmatrix} 0 & 1 & 0 \\ 0 & -\frac{c}{J_m} & \frac{K_T}{J_m} \\ 0 & -\frac{K_V}{L_a} & -\frac{R_m+R}{L_a} \end{pmatrix}$ $\quad B_1 = B(:,1) = \begin{pmatrix} 0 \\ 0 \\ \frac{1}{L_a} \end{pmatrix}$

$$C = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \qquad D_1 = D(:,1) = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$$

$$\overline{x}(t) = \begin{pmatrix} \theta(t) \\ \dot{\theta}(t) \\ i(t) \end{pmatrix} \qquad u_1(t) = V_a(t)$$

Using realistic numbers, we have

$$A = \begin{pmatrix} 0 & 1 & 0 \\ 0 & -0.29 & 71.93 \\ 0 & -63.24 & -1020.35 \end{pmatrix} \qquad B_1 = \begin{pmatrix} 0 \\ 0 \\ 641.81 \end{pmatrix}$$

We can calculate the open loop eigenvalues/poles of the system from the *A* matrix. Note that for the remainder of the discussion, we consider eigenvalues and poles to be the same (in other words we assume there is no pole/zero cancellation).

open_loop_poles = eig(A)

we obtain the open loop poles

```
open_loop_poles =

  1.0e+003 *

         0
   -0.0048
   -1.0159
```

This yields

$\lambda_1 = 0$
$\lambda_2 = -4.77$
$\lambda_3 = -1015.9$

We now desire to move the closed loop poles of the system. We can first verify that the system is controllable. We can use the Matlab command ‘ctrb’ to compute the controllability matrix and the ‘rank’ command to compute the rank of the matrix

Pc = ctrb(A,B1)
rank(Pc)

```
Pc =

  1.0e+008 *

         0         0    0.0005
         0    0.0005   -0.4712
    0.0000   -0.0065    6.6527


ans =

     3
```

So we see the system is fully controllable, so we should be able to move the closed loop poles to any desired location.

We can first verify that placing the closed loop poles at the open loop poles requires no control.

K = place(A, B1, open_loop_poles)

```
K =

  1.0e-014 *

   -0.0177   -0.1672         0
```

As can be seen, $K \approx (\,0 \quad 0 \quad 0\,)$, which makes sense considering we did not attempt to move the poles. Let us now investigate what issues might arise if we try to move the poles.

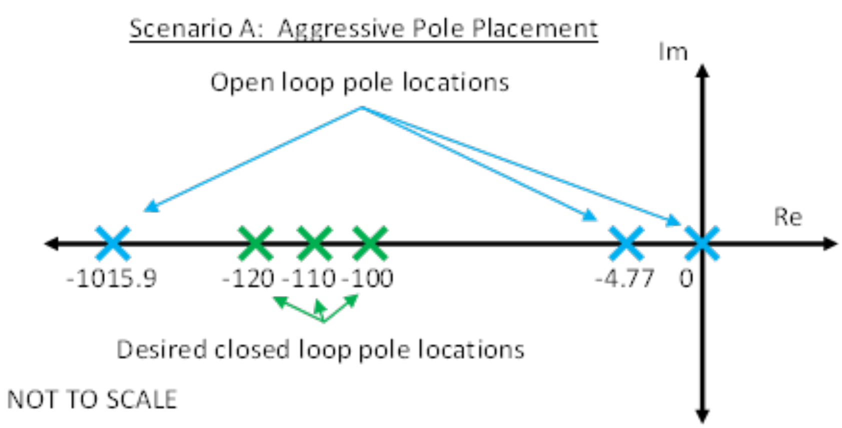
Control Saturation Issues

Let us consider the location of the open loop poles of the system.

$$\lambda_1 = 0$$
$$\lambda_2 = -4.77$$
$$\lambda_3 = -1015.9$$

Let us first attempt to move the poles to an arbitrary location

$$\lambda_{1,\text{cl}} = -100$$
$$\lambda_{2,\text{cl}} = -110$$
$$\lambda_{3,\text{cl}} = -120$$



At first glance, these appear to be desirable pole locations, they are all on the real axis and far in the left hand plane so we should have a fast response. Since the system is controllable, we can use the following code to calculate the required gain K to achieve these closed loop poles

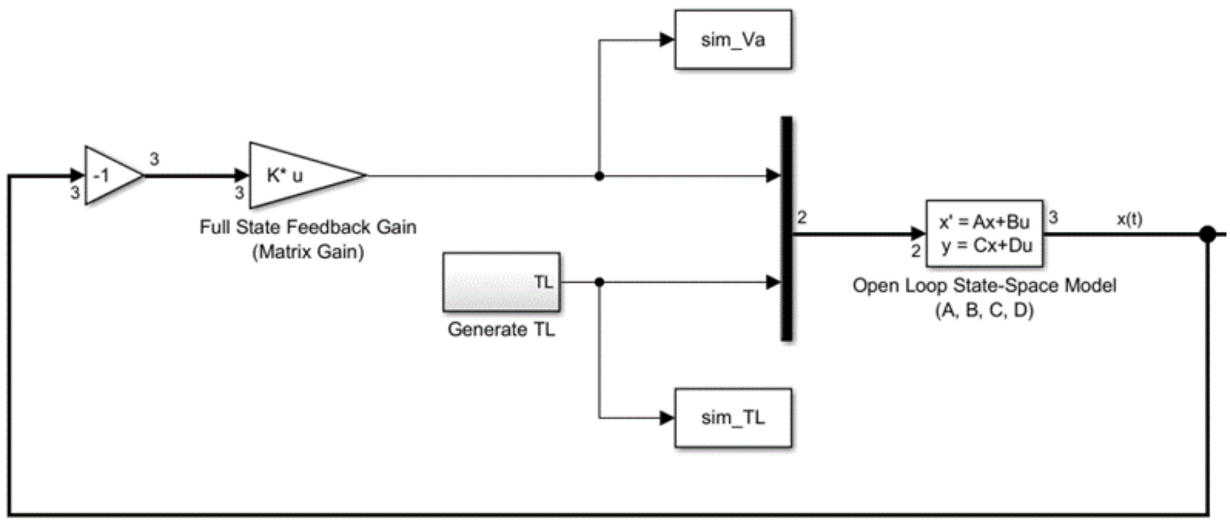
```
desired_closed_loop_poles = [-100; -110; -120];
K = place(A, B1, desired_closed_loop_poles);
```

This yields

```
K =

   28.5942    0.6836   -1.0761
```

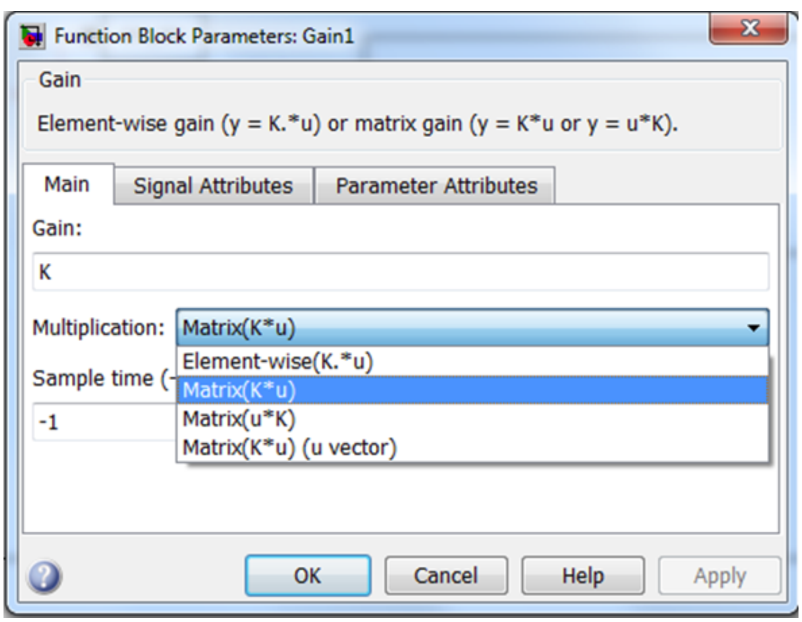
We can implement this in simulation with the following model.



Note that the control law is

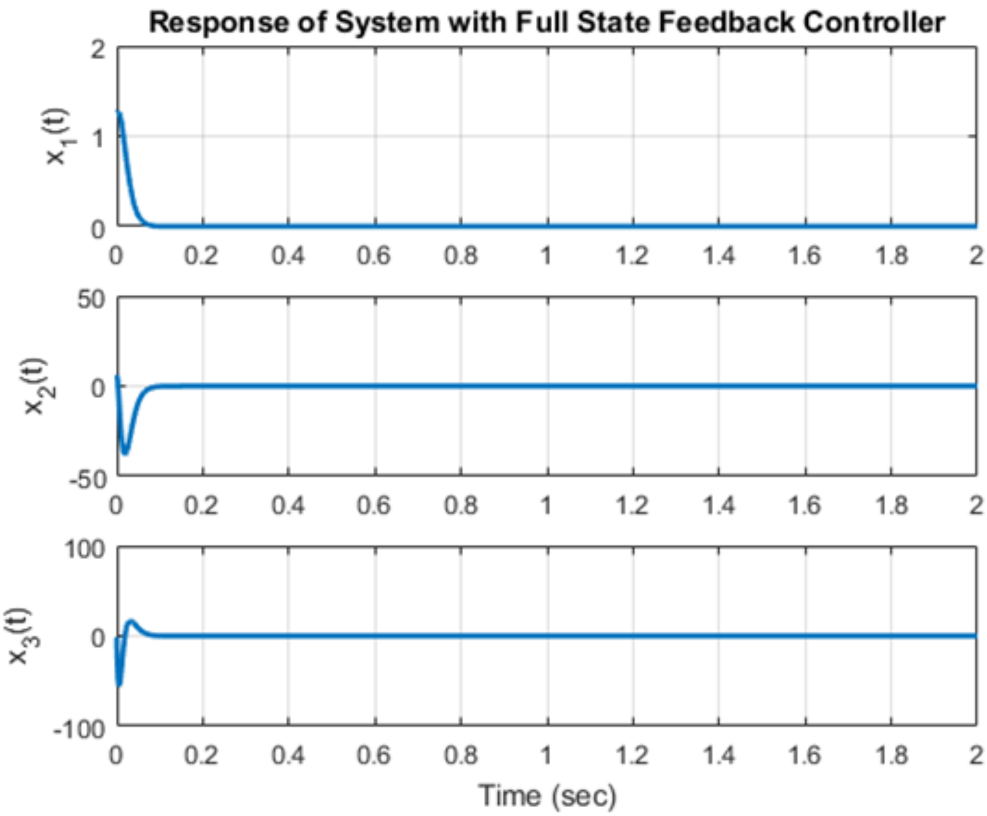
$$u(t) = -K\,\overline{x}(t)$$

This is a matrix operation. By default, when we use a gain block in Simulink, this attempts to perform element-wise multiplication. In this situation, we need to change it to matrix multiplication as shown below.

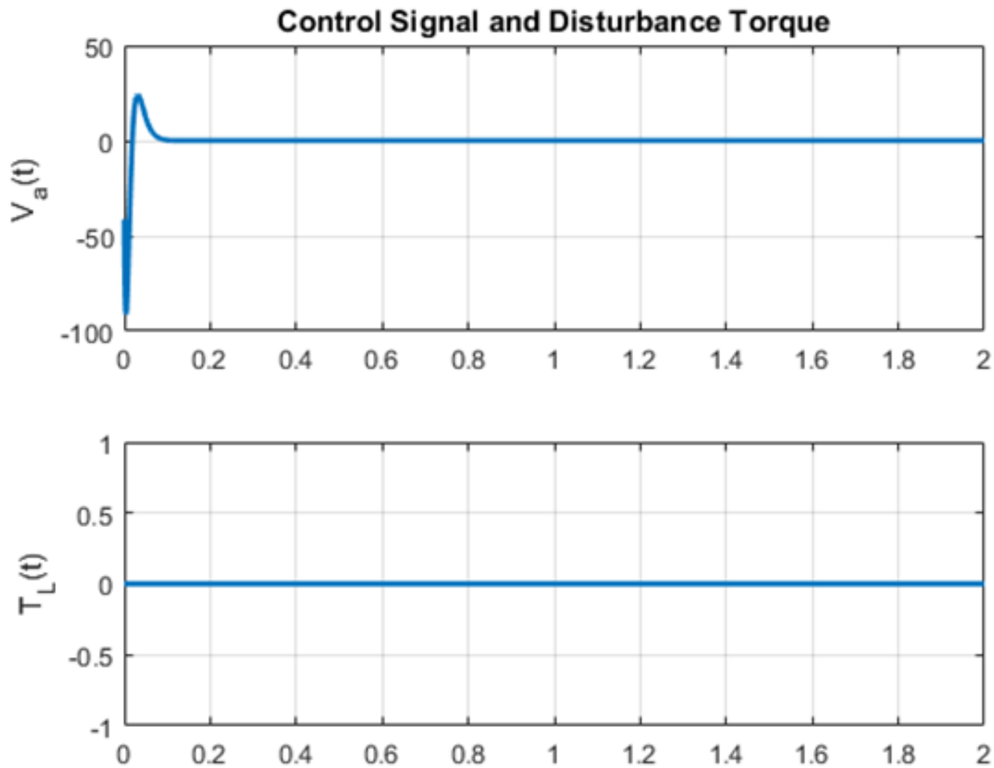


The response of the system in response to some non-zero initial conditions is shown below

$$\overline{x}(0) = \begin{pmatrix} \theta(0) \\ \dot{\theta}(0) \\ i(0) \end{pmatrix} = \begin{pmatrix} 72 \frac{\pi}{180} \text{ rad} \\ 2\,\pi \text{ rad}/s \\ -1 \text{ amp} \end{pmatrix}$$



At first glance, this appears to be a well designed controller because the system is regulated to zero ($\overline{x}(t) \to \overline{0}$) in a very short time. However, we can investigate the associated inputs to the system (the control signal and disturbance torque)

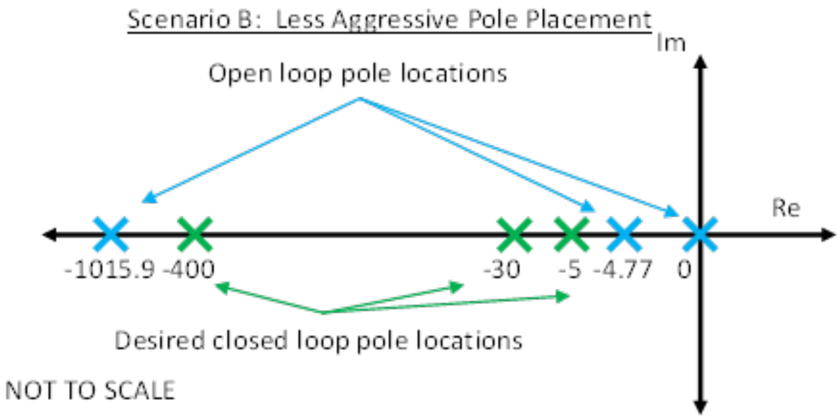


By looking at the control signal, we see that the cost of quickly regulating the system states to zero is a large control signal (voltages of almost -90 volts are required). Because our system will saturate at lower voltages, this type of controller is too aggressive.

The reason why the controller is so aggressive is because we are attempting to drastically change the location of the open loop poles (ie the closed loop poles are very different from the open loop poles). Physically, this means we are trying to effect a large change the system dynamics (talk about example of trying to get your dog to stop eating off the dinner table, quit sleeping on nice living room furniture, and cease stealing your kids toys). The price for this change is a large control signal.

We can repeat this procedure with desired closed loop poles which are closer to the original, open loop poles

$$\lambda_{1,cl} = -5$$
$$\lambda_{2,cl} = -30$$
$$\lambda_{3,cl} = -400$$

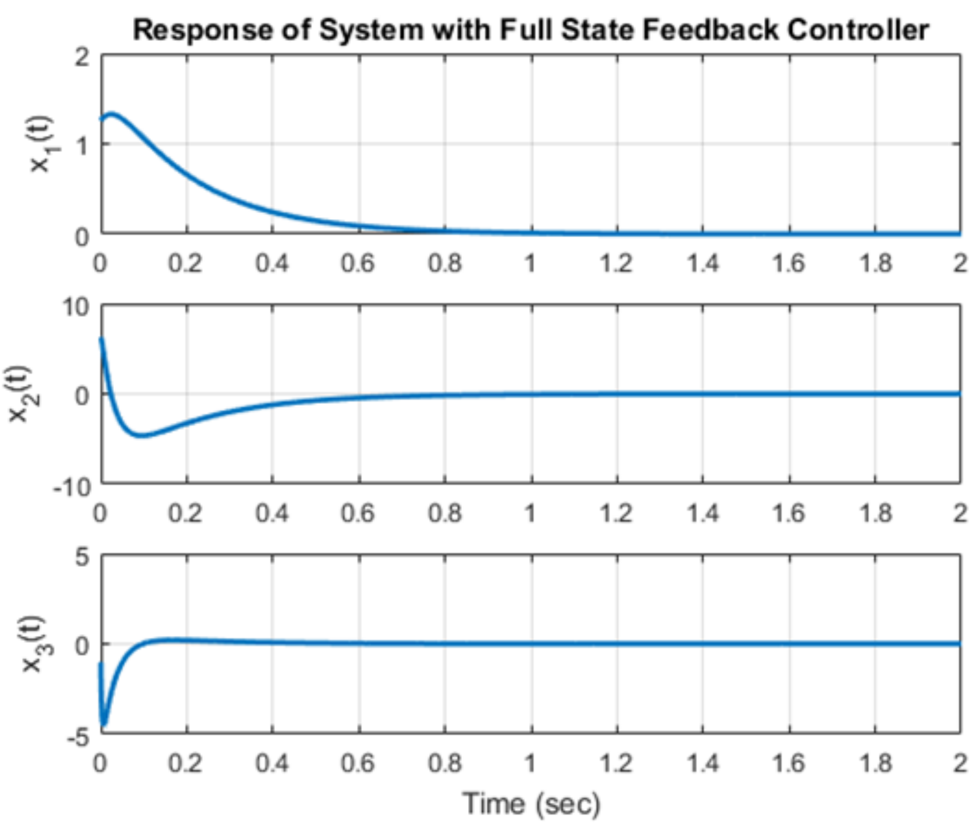


This yields a gain *K*

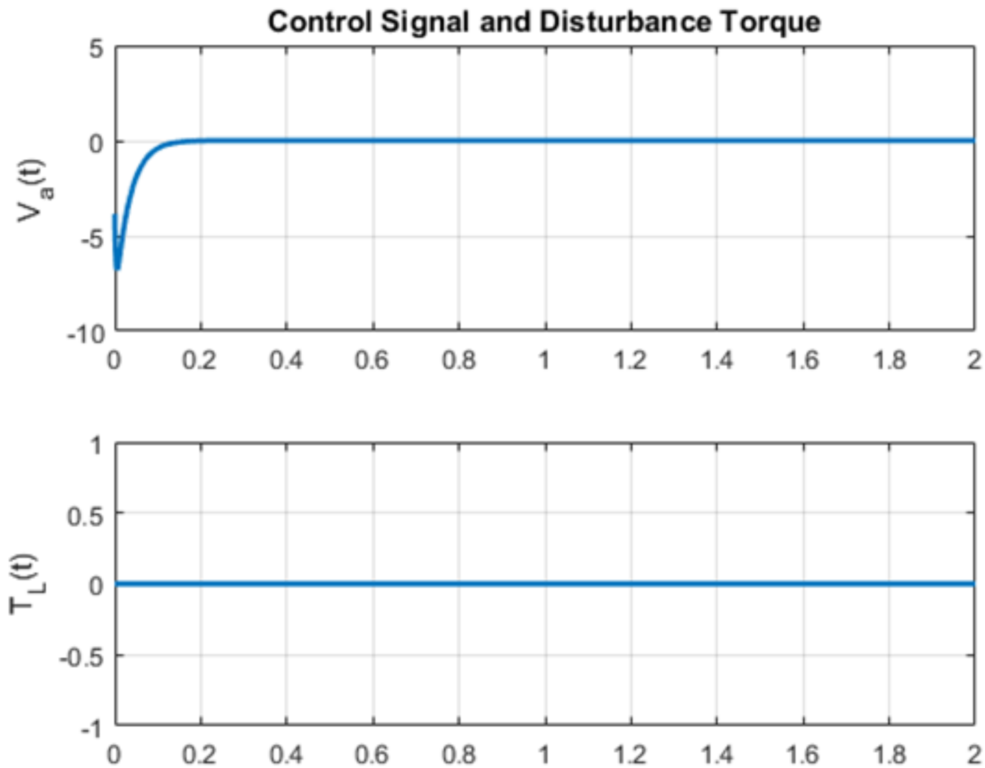
```
K =

    1.2997    0.2053   -0.9125
```

The system response to the same initial conditions is shown below.



The associated control signal is shown below



As can be seen, the system states are still regulated to zero $\left(\overline{x}(t) \rightarrow \overline{0}\right)$ although this happens much slower. The advantage of this slower response is that the required control signal to regulate the system states to zero is now lower and within reasonable values.

Inability to Measure Full State of System

The major deficiency of full state feedback is that it requires direct measurement of all system states. In other words, we must have $\overline{y}(t) = \overline{x}(t)$ or be able to compute the full system state from the measurement (ie $\overline{x}(t) = f(\overline{y}(t))$). This might not seem like a major obstacle with our simple DC motor system with only 3 states but consider a simple rigid body, 6DOF dynamic model of an aircraft. It can be shown that a state vector for this system is

$$\overline{x} = \begin{pmatrix} u & v & w & p & q & r & \phi & \theta & \psi & P_N & P_E & P_D \end{pmatrix}^T$$

In this case, you would need to purchase sensors to measure every single state. This can be an expensive proposition. If the system had even more states, then this measurement problem becomes even more complicated. This sets the stage for estimation ***(YouTube video URL here).***

Let us express the control law for our DC motor full state feedback system as

$$u(t) = -K\,\overline{x}(t)$$

$$= -\begin{pmatrix} k_1 & k_2 & k_3 \end{pmatrix} \begin{pmatrix} \theta(t) \\ \omega(t) \\ i(t) \end{pmatrix}$$

$$u(t) = -k_1\,\theta(t) - k_2\,\omega(t) - k_3\,i(t) \qquad \textbf{\textit{(Eq.1)}}$$

In the case of our DC motor, we only directly measure the position of the motor. We therefore need to somehow calculate the velocity, $\omega(t)$, and the current $i(t)$

$\theta(t)$ (directly measured)
$\omega(t)$ (can be approximated using a numerical derivative)
$i(t)$ (can calculate $i(t) = V_R(t)/R$)

As we can seen, there are several issues with this scenario. First, as we experienced previously, the numerical derivative may cause noise issues. The previous solution to this problem was to employ a pseudo-derivative

$$\tilde{\omega}(s) = \frac{a\,s}{s+a}\,\theta(s)$$

However, we see that $\tilde{\omega}(s) \neq \omega(s)$, and therefore, technically, this cannot be used for full state feedback. This problem is further exacerbated if we use additional filtering

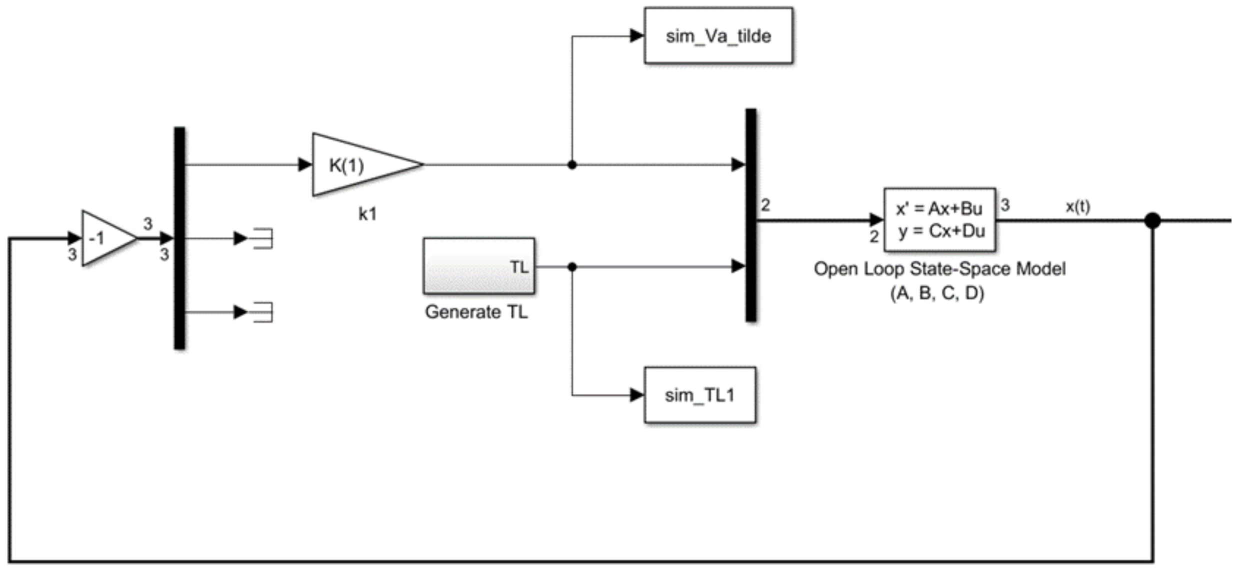
$$\tilde{\omega}(s) = \frac{a\,s}{s+a}\left(\frac{a}{s+a}\right)^n \theta(s)$$

Additionally, it is may be difficult to measure $i(t)$. For the purposes of this academic exercise, we can assume that we cannot measure $i(t)$. Therefore, we cannot implement Eq.1 because we cannot directly measure $\omega(t)$ or $i(t)$.

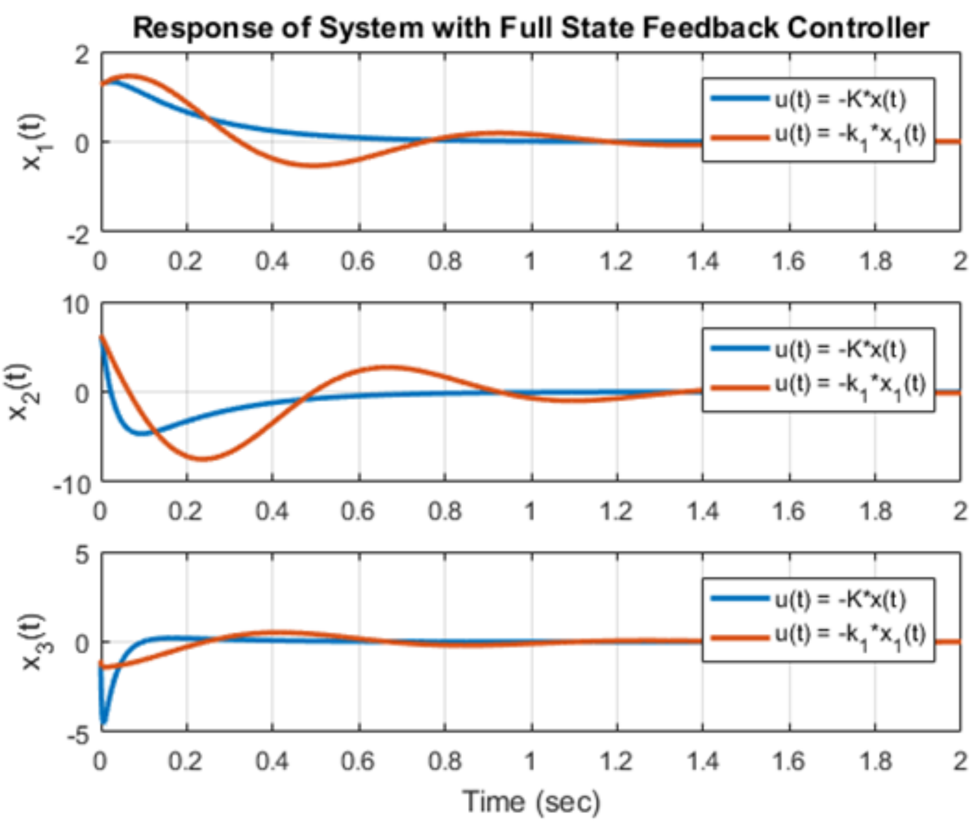
We are therefore forced to implement a controller that neglects the terms k_2 and k_3 and simply uses a control law of

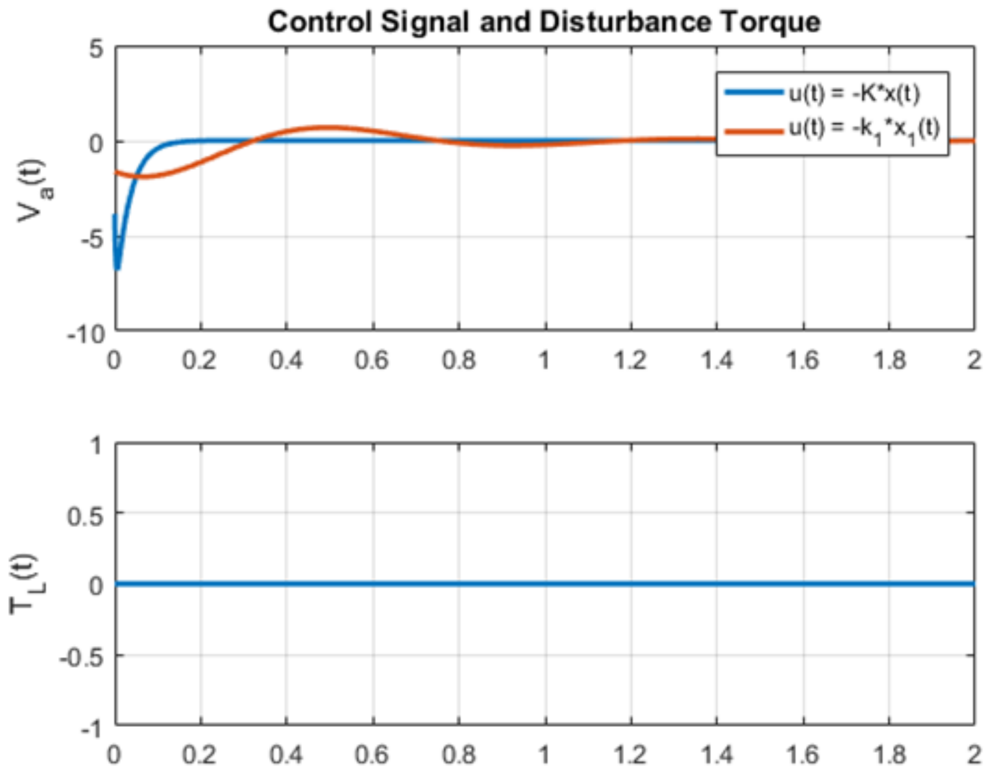
$$u(t) = -k_1\,\theta(t) \qquad \textbf{\textit{(Eq.2)}}$$

Eq.2 is easily implementable because we directly measure $\theta(t)$ using an encoder.



However applying this control to the previous scenarios shows why this is not ideal. For example, consider placing the poles as we examined in scenario B (the less aggressive pole placement)

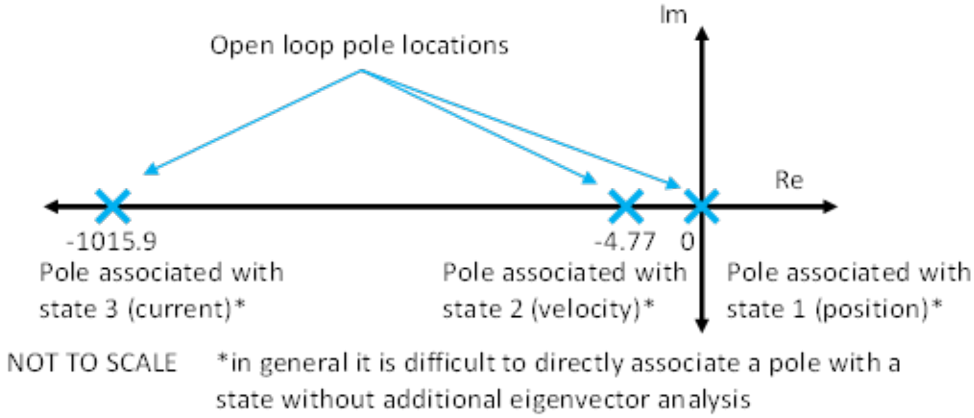




We see there are now oscillations in the response and control and we do not achieve the desired closed loop pole locations we originally requested.

To combat this problem, we note that if we were somehow able to chose desired closed loop pole locations such that the 'place' command returns values of $k_2 \approx k_3 \approx 0$, then the control law for full state feedback would reduce to Eq.2 naturally and would not involve any major compromises nor surprises. Therefore, we would like to choose closed loop pole locations which yield $k_2 \approx k_3 \approx 0$.

From physical intuition, we know that the pole at the origin is associated with the position state $\left(\theta(s) = \frac{1}{s}\,\omega(s)\right)$. We also know that the electrical dynamics of the system are very fast, so most likely the pole at $\lambda = -1015.9$ is associated with the current state. Finally, the velocity state is most likely associated with the pole at $\lambda = -4.77$.



In general, it is difficult to directly associate a pole with a state without additional eigenvector analysis. If we look at the eigenvectors of the A matrix, we have

$$[V,\ D] = \text{eig}(A)$$

$$D = \begin{pmatrix} 0 & 0 & 0 \\ 0 & -4.77 & 0 \\ 0 & 0 & -1015.9 \end{pmatrix} V = \begin{pmatrix} 1 & -0.20 & 0.0001 \\ 0 & 0.97 & 0.0706 \\ 0 & -0.06 & 0.9975 \end{pmatrix}$$

$\lambda_1 = 0$ $\quad \overline{v}_1 = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}$ $\quad$ (completely associated with $x_1(t) = \theta(t)$)

$\lambda_2 = -4.77$ $\quad \overline{v}_2 = \begin{pmatrix} -0.20 \\ 0.97 \\ -0.06 \end{pmatrix}$ $\quad$ (mostly associated with $x_2(t) = \omega(t)$ with some association with $x_1(t)$)

$\lambda_3 = -1015.9$ $\quad \overline{v}_3 = \begin{pmatrix} 0.0001 \\ 0.0706 \\ 0.9975 \end{pmatrix}$ $\quad$ (mostly associated with $x_3(t) = i(t)$)

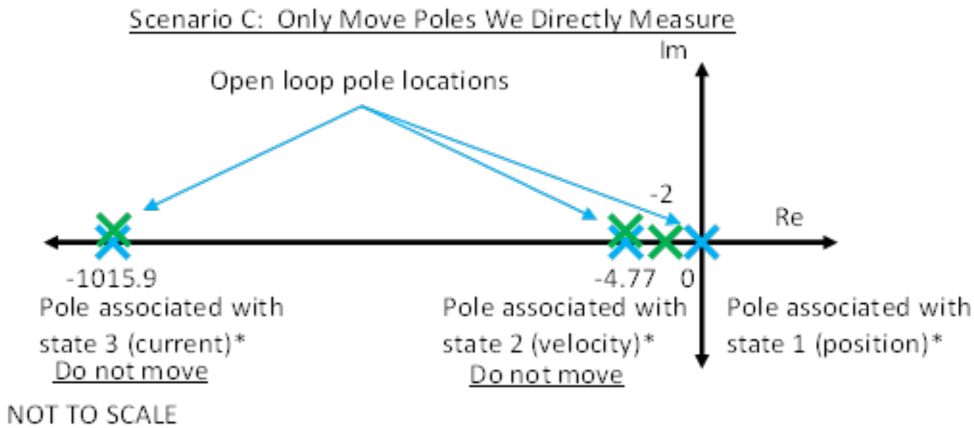
In this case, we see that each pole is mostly associated with a given state. Therefore, if we do not attempt to change poles associated with states we cannot measure, the associated gain will hopefully be small or negligible.

With this assumption, let us choose the following desired closed loop poles

$\lambda_{1,\text{cl}} = -2$ $\quad$ (move pole associated with $\theta(t)$)

$\lambda_{2,\text{cl}} = -4.77$ $\quad$ (do not move pole associated with $\omega(t)$)

$\lambda_{3,\text{cl}} = -1015.9$ $\quad$ (do not move pole associated with $i(t)$)



With these desired closed loop poles, the pole placement method yields

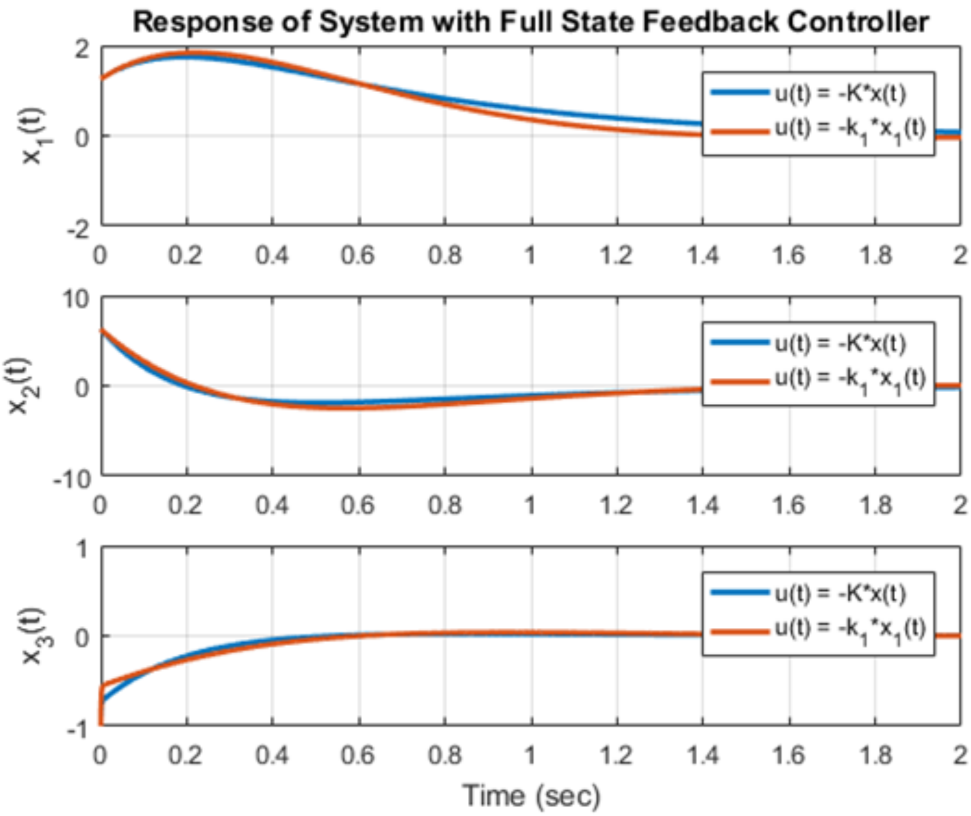
```
K =

    0.2099    0.0442    0.0031
```

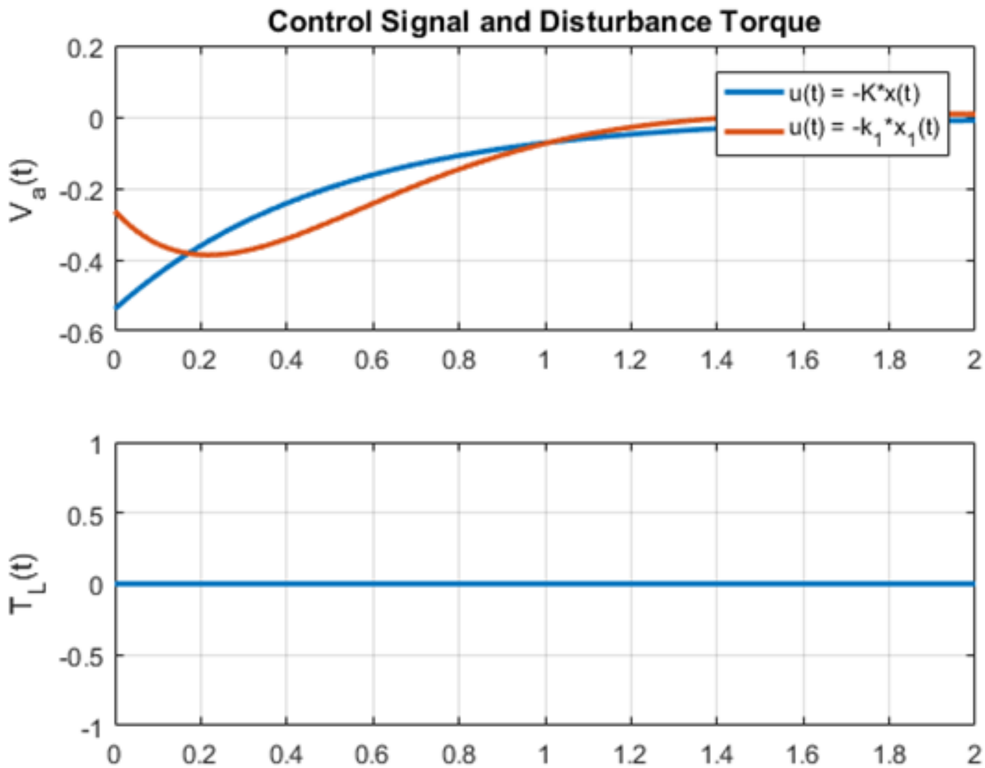
As can be seen, this yields the desired response that $k_2 \approx k_3 \approx 0$. Therefore, we can implement the approximate full state feedback control law of

$$u(t) = -k_1\, x_1(t) = -k_1\, \theta(t)$$

We can compare the response of the system under the full control law of $u(t) = -K\,\overline{x}(t)$ and the approximate control law of $u(t) = -k_1\, x_1(t)$. The response of the system is shown below



As can be seen, performance has degraded somewhat when using the approximate control law, but it still successfully regulates the states to zero. The control signal for the two system is shown below



Once again, we see that the control signal, $u(t) = V_a(t)$ is slightly different for the two systems, but the net effect of both control laws is $\overline{x}(t) \to \overline{0}$.

Note that in general, you cannot associate a pole with a single state as most systems are coupled. This means that a single mode affects several states. For example, consider the same rigid body, 6DOF dynamic model we discussed at the beginning of this section with state vector $\overline{x} = (u \quad v \quad w \quad p \quad q \quad r \quad \phi \quad \theta \quad \psi \quad P_N \quad P_E \quad P_D)^T$. A linearized model yields eigenvectors where some have a single non-zero entry (showing that the associated mode/eigenvalue is associated with just a single state) and other vectors which have non-zero entries in many locations (showing that the associated mode/eigenvalue is associated with several states). See the Matlab code for more information. This makes it difficult to determine an exact relationship between desired closed loop pole location and full state feedback controller gains. It would be great if there was a tool/method that allowed us to directly specify which states and controls are important in the context of designing a full state feedback controller. This is exactly what the linear quadratic regulator (LQR) problem is setup to do and we will investigate this in a future lecture (see lecture/video 'Introduction to Linear Quadratic Regulator (LQR) Control' located at https://youtu.be/wEevt2a4SKI).