April 03, 2024

6.3100 Lecture 16 Notes – Spring 2024

Connection between CT Classical Control and State-Space Control

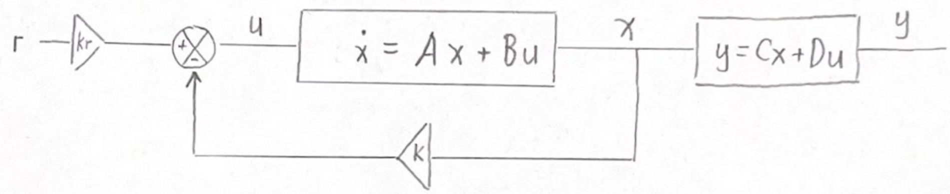
Dennis Freeman, , Elfar Adalsteinsson, and Kevin Chen

Outline:

1. Open loop and closed loop transfer functions
2. Example: inverted pendulum
3. Converting between state space representation and open-loop transfer functions
4. Relationship between eigenvalues and closed-loop poles

1. Open loop and closed loop transfer functions

We introduced the state space control formalism in the previous lecture. This control approach can be represented by the block diagram shown below.



Here r is the reference input, u is the controller input, x is the system state, and y is the system output. We need to design Kr and K. We want to make a distinction of terminology here:

- H_{open} refers to the open loop transfer function, which describes the behavior of a plant. It relates controller input u to system state x.
- H_{close} refers to the closed loop transfer function, which describes the behavior of the entire system. It relates the reference input r to the system output y.

To obtain H_{open} and H_{close}, we can perform Laplace transform and write the block diagram in the "s" domain. In the transform process, we have:

$$r(t) \rightarrow R(s)$$

$$u(t) \rightarrow U(s)$$

$$x(t) \rightarrow X(s);\ \dot{x}(t) \rightarrow sX(s)$$

$$y(t) \rightarrow Y(s)$$

The block diagram in the "s" domain becomes:

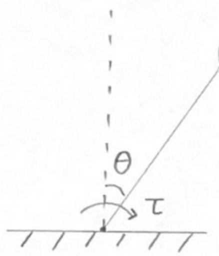
Based on the definition of H_{open} and H_{close}, we have the following relationships:

$$Y = H_{open}U$$

$$Y = H_{close}R$$

2. **Example: inverted pendulum**

We use the example of an inverted pendulum to illustrate the connection between state space representation and transfer functions. Consider the diagram below:



We want to stabilize this pendulum through applying a torque τ. The system output is the measured angle θ. First, using Newton's second law, we can write a differential equation that describes this system:

$$J\ddot{\theta} = mgl\sin\theta + \tau$$

where $J = ml^2$ is the system moment of inertia. Simplifying this equation and we will obtain:

$$\ddot{\theta} = \frac{g}{l}\sin\theta + \frac{1}{ml^2}\tau \approx \frac{g}{l}\theta + \frac{1}{ml^2}\tau$$

To write it in the state space form, we write two first order differential equations:

$$\frac{d}{dt}\theta = \dot{\theta}$$

$$\frac{d}{dt}\dot{\theta} = \frac{g}{l}\theta + \frac{1}{ml^2}\tau$$

We can write those equations in the matrix form and obtain the state-space representation:

$$\frac{d}{dt}\begin{pmatrix}\theta \\ \dot{\theta}\end{pmatrix} = \begin{pmatrix}0 & 1 \\ g/l & 0\end{pmatrix}\begin{pmatrix}\theta \\ \dot{\theta}\end{pmatrix} + \begin{pmatrix}0 \\ 1/ml^2\end{pmatrix}\tau$$

$$\theta = \begin{pmatrix}1 & 0\end{pmatrix}\begin{pmatrix}\theta \\ \dot{\theta}\end{pmatrix} + 0\tau$$

Let's put in some numbers. For this problem, we let $g = 9.8\ m/s^2$, $l = 0.1m$, and $m = 0.1\ kg$. Substituting in these numbers, we will obtain the A, B, C, and D matrices:

$$A = \begin{pmatrix}0 & 1 \\ 98 & 0\end{pmatrix}, B = \begin{pmatrix}0 \\ 1000\end{pmatrix}, C = \begin{pmatrix}1 & 0\end{pmatrix}, D = 0$$

3. Converting between state space representation and open-loop transfer functions

State space to open-loop transfer function

Having a state-space representation, how can we find the open-loop transfer function H_{open}? We can write that:

$\dot{x} = Ax + Bu$ $y = Cx + Du$ $\rightarrow$	$sIX = AX + BU$ $Y = CX + DU$ $\rightarrow$	$(sI - A)X = BU$ $Y = C(sI - A)^{-1}BU + DU$

In most problems, D = 0. We have:

$$H_{open} = C(sI - A)^{-1}B$$

Based on the prior example, we have:

$$\begin{aligned} H_{open} &= C(sI - A)^{-1}B \\ H_{open} &= \begin{pmatrix} 1 & 0 \end{pmatrix} \begin{pmatrix} s & -1 \\ -98 & s \end{pmatrix}^{-1} \begin{pmatrix} 0 \\ 1000 \end{pmatrix} \\ H_{open} &= \frac{1}{s^2 - 98} \begin{pmatrix} 1 & 0 \end{pmatrix} \begin{pmatrix} s & 1 \\ 98 & s \end{pmatrix} \begin{pmatrix} 0 \\ 1000 \end{pmatrix} \\ H_{open} &= \frac{1000}{s^2 - 98} \end{aligned}$$

We can also use computational software for performing this transformation. We can write:

$$[num, den] = ss2tf(A, B, C, D)$$

and we will obtain:

$$num = [0\ 0\ 1000] \text{ and } den = [1\ 0 - 98]$$

This means the transfer function is given by:

$$H_{open} = \frac{0s^2 + 0s + 1000}{1s^2 + 0s - 98}$$

Note that H_{open} is unique, as there is only one H_{open} given A, B, C, and D.

Transfer function to state space representation

We can also perform the transformation in the reverse direction. Given an open-loop transfer function, we can write out the state space representation.

$$Y = H_{open}U \rightarrow \Theta(s) = H_{open}\mathrm{T}(s)$$

where the time domain variables are transformed to the "s" domain: $\theta(t) \rightarrow \Theta(s)$ and $\tau(t) \rightarrow \mathrm{T}(s)$. From the previous question, we have:

$$\Theta(s) = \frac{1000}{s^2 - 98}\mathrm{T}(s)$$

We can factor the denominator and obtain:

$$\Theta(s) = \frac{1000}{s^2 - 98}\mathrm{T}(s) = \frac{1000}{s + \sqrt{98}}\frac{1}{s - \sqrt{98}}T(s)$$

Next, we can define two state variables:

$$X_1 = \Theta = \frac{1000}{s + \sqrt{98}}\frac{1}{s - \sqrt{98}}T(s)$$

$$X_2 = \frac{1}{s - \sqrt{98}}T(s)$$

This implies that: $X_1 = \frac{1000}{s+\sqrt{98}}X_2$. Then we can convert those two equations back into the time domain:

$(s - \sqrt{98})X_2 = T$ $\frac{d}{dt}x_2 - \sqrt{98}x_2 = \tau$ $\frac{d}{dt}x_2 = \sqrt{98}x_2 + \tau$	$X_1(s + \sqrt{98}) = 1000X_2$ $\frac{d}{dt}x_1 + \sqrt{98}x_1 = 1000x_2$ $\frac{d}{dt}x_1 = -\sqrt{98}x_1 + 1000x_2$

Finally, we can write these two equations in the matrix form:

$$\frac{d}{dt}\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} -\sqrt{98} & 1000 \\ 0 & \sqrt{98} \end{pmatrix}\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \begin{pmatrix} 0 \\ 1 \end{pmatrix}\tau$$

$$\theta = (1 \quad 0)\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + 0\tau$$

The A, B, C, and D matrices are given by:

$$A = \begin{pmatrix} -\sqrt{98} & 1000 \\ 0 & \sqrt{98} \end{pmatrix}, B = \begin{pmatrix} 0 \\ 1 \end{pmatrix}, C = (0 \quad 1000), D = 0$$

We can also use numerical tools to perform the conversion. In MATLAB, we can use the following command:

$$[A, B, C, D] = tf2ss(num, den)$$

MATLAB will return:

$$A = \begin{pmatrix} 0 & 98 \\ 1 & 0 \end{pmatrix}, B = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, C = (0 \quad 1000), D = 0$$

Note that these two state space representations are different! A quick takeaway is that state space representation is not unique. It depends on what “state” x we choose.

4. Relationship between eigenvalues and closed-loop poles

Having examined the relationship between H_{open} and state space representations, let's now study H_{close}. Specifically, we are interested in the stability of a control system. From previous lectures, we know a control system is stable if:

- All poles of H_{close} are negative
- All eigenvalues of the matrix $A - BK$ are negative

How are those two conditions connected? Let's try to convert a state space controller into a closed-loop transfer function. Supper D = 0, the state space controller is given by:

$$\dot{x} = Ax + Bu$$
$$y = Cx$$
$$u = K_r r - Kx$$

The Laplace transform of these equations are given by:

$$sX = AX + BU$$
$$Y = CX$$
$$U = K_r R - KX$$

We can solve for X and Y and obtain:

$$X = (sI - (A - BK))^{-1} BK_r R$$
$$Y = CX = C\big(sI - (A - BK)\big)^{-1} BK_r R = H_{close} R$$

This gives the expression of H_{close}:

$$H_{close} = C\big(sI - (A - BK)\big)^{-1} BK_r$$

Now let's try to relate the eigenvalues of A-BK to the closed loop poles.

$$eig(A - BK)\ are\ the\ roots\ of\ \det\big(sI - (A - BK)\big) = 0$$

Here, we use a linear algebra result:

$$\big(sI - (A - BK)\big)^{-1} = \frac{adj(sI - (A - BK))}{\det\,(sI - (A - BK))}$$

Note:

- adj(P) is the adjoint matrix of the matrix P, where each element is the cofactor of P_{ij}.
- Det(P) is the determinant (which is a number) of the matrix P

So we have the equation:

$$H_{close} = C\big(sI - (A - BK)\big)^{-1} BK_r = \frac{1}{\det\,(sI - (A - BK))} C\ adj\big(sI - (A - BK)\big) BK_r$$

This shows roots of $\det\big(sI - (A - BK)\big) = 0$ are the closed-loop poles. Eigenvalues are the poles!