6.3100/2 - Lecture 11/8/23

Goals:

1) Stabilize the system
2) Achieve good performance on some metric

$$\begin{tikzcd}
\text{Standard form for system} \arrow[dr] & & \\
& \boxed{\text{Optimizer}} \arrow[r] & \text{controller in a standard form} \\
\text{Parameters for metric} \arrow[ur] & &
\end{tikzcd}$$

Today: State space form

State space form:

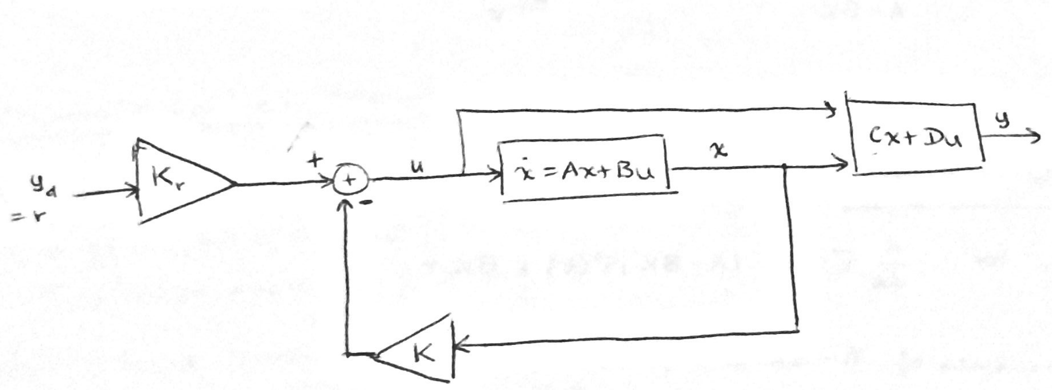
$$\frac{d}{dt}\vec{x}(t) = A\vec{x}(t) + Bu(t)$$ [time evolution of state]

$$y(t) = C\vec{x}(t) + Du(t)$$ [output]

$$u(t) = K_r \overset{\shortparallel}{y_d}(t) - K\vec{x}(t) = K_r r(t) - K\vec{x}(t)$$ [controller]

Notes

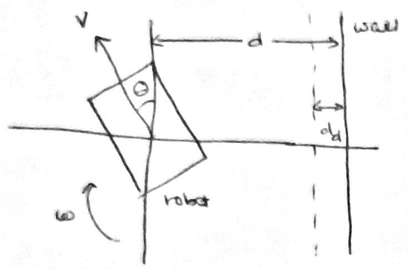
- Separating notions of state (the thing we stabilize) and output (the thing we optimize performance on). Done in both CT and DT (even though we didn't explicitly intro. this notation in DT)
- $\vec{x}(t)$ has n dims, $y(t)$ and $u(t)$ both scalars $\rightarrow$ single input, single output (SISO) system
- $\vec{x}(t)$ must contain sufficient info to describe behavior of system
- $\vec{x}(t)$ can "contain" $y(t)$ [Implicitly the case in many examples we have seen]



[illegible]

Note: Similarities and differences in structure from PID control

Wall-following example



$$\frac{d}{dt}\begin{bmatrix} d(t) \\ \theta(t) \end{bmatrix} = \begin{bmatrix} V\sin\theta(t) \\ \omega(t) \end{bmatrix} \approx \begin{bmatrix} V\theta(t) \\ \omega(t) \end{bmatrix}$$ small angle approx

V constant

$y(t) = d(t)$

we control $\omega(t)$

State space representation

$$\frac{d}{dt}\begin{bmatrix} d(t) \\ \theta(t) \end{bmatrix} = \begin{bmatrix} 0 & V \\ 0 & 0 \end{bmatrix}\begin{bmatrix} d(t) \\ \theta(t) \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix}\omega(t)$$

$$y(t) = \begin{bmatrix} 1 & 0 \end{bmatrix}\begin{bmatrix} d(t) \\ \theta(t) \end{bmatrix} + 0\,\omega(t)$$

$$\omega(t) = K_r d_d(t) - \begin{bmatrix} K_1 & K_2 \end{bmatrix}\begin{bmatrix} d(t) \\ \theta(t) \end{bmatrix}$$

Plugging in the controller:

$$\frac{d}{dt}\begin{bmatrix} d \\ \theta \end{bmatrix} = \begin{bmatrix} 0 & V \\ 0 & 0 \end{bmatrix}\begin{bmatrix} d \\ \theta \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix}\left(K_r d_d - \begin{bmatrix} K_1 & K_2 \end{bmatrix}\begin{bmatrix} d \\ \theta \end{bmatrix}\right)$$

$$= \left(\begin{bmatrix} 0 & V \\ 0 & 0 \end{bmatrix} - \begin{bmatrix} 0 \\ 1 \end{bmatrix}\begin{bmatrix} K_1 & K_2 \end{bmatrix}\right)\begin{bmatrix} d \\ \theta \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix}K_r d_d$$

$$= \underbrace{\begin{bmatrix} 0 & V \\ -K_1 & -K_2 \end{bmatrix}}_{A-BK}\begin{bmatrix} d \\ \theta \end{bmatrix} + \underbrace{\begin{bmatrix} 0 \\ 1 \end{bmatrix}}_{BK_r} K_r d_d$$

How to analyze stability?

In general, for $\frac{d}{dt}\vec{x}(t) = (A-BK)\vec{x}(t) + BK_r r$

Eigenvalues of $A-BK$ are poles

Real(eigvals(A−BK)) < 0

Stability of wall-follower

① Suppose: $K_2 = 0$ (use only d feedback)

$$\frac{d}{dt}\begin{bmatrix} d \\ \theta \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & V \\ -K_1 & 0 \end{bmatrix}}_{A-BK}\begin{bmatrix} d \\ \theta \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} K_r d_d$$

Find eigvals $(A-BK)$

$$\det\left(sI-(A-BK)\right) = 0$$

$$\Rightarrow \det\left(\begin{bmatrix} s & -V \\ K_1 & s \end{bmatrix}\right) = 0$$

$$\Rightarrow s^2 + K_1 V = 0$$

$\Rightarrow s = \pm j\sqrt{K_1 V}$ not stable !! [recall: similar conclusion in DT]

② Suppose $K_2 \neq 0$ (have d and θ feedback)

eigvals of $A-BK = \begin{bmatrix} 0 & V \\ -K_1 & -K_2 \end{bmatrix}$ are poles

recall in DT that we could stabilize system with d and θ feedback for the right choice of controller. Same conclusion holds in this case for CT.

how do we pick K_1 and K_2?

Pole placement (eigenvalue placement)

Pick K to make eigvals $(A-BK)$ have as negative a real part as possible

That is: $\min_{K_1,\dots,K_n} \left(\max_i \text{Real}\left(\text{eigvals}_i (A-BK)\right)\right)$

Make pole with least negative real part be as negative as possible

Pole placement for wall follower

$$\det\left(sI-(A-BK)\right) = \det\left(\begin{bmatrix} s & -V \\ K_1 & s+K_2 \end{bmatrix}\right) = 0$$

$$\Rightarrow s(s+K_2) - K_1(-V) = 0$$

$$\Rightarrow s^2 + K_2 s + K_1 V = 0$$

suppose want $\lambda_1 = -100$ and $\lambda_2 = -200$

$$(s-\lambda_1)(s-\lambda_2) = s^2 - (\lambda_1+\lambda_2)s + \lambda_1\lambda_2$$

$$(s+100)(s+200) = s^2 + 300s + 20000$$

$$\Rightarrow K_1 = \frac{20000}{V}, \quad K_2 = 300$$

Can you place λ_1 and λ_2 anywhere?

where do you actually want to place the poles? [consider: control input magnitude, robustness, performance on metric, etc...]

April 13, 2023

6.3100 Lecture 18 Notes – Spring 2023

Pole placements and propellor arm example

Dennis Freeman and Kevin Chen

Outline:

1. Review of state-space control
2. Pole placement: choosing K and Kr
3. Example: inverted pendulum
4. Example: propellor arm

1. Review of state-space control

Thus far we have introduced state space control without explaining how to design the controller. First, let's review the formulation. A state-space system is given by:

$$E\,\dot{x}(t) = Ax(t) + Bu(t)$$

$$y(t) = Cx(t) + Du(t)$$

where $u, y, and\ x$ are the input, output, and state variables. The core question is how do we design the input control signal u(t)?

From the first lecture of this semester, we learn that the simplest controller is a proportional controller:

$$u(t) = K_p(r(t) - y(t))$$

Following a similar form, we can use proportional feedback on the state:

$$u(t) = K_r r(t) - Kx(t)$$

Note that r(t) is a scalar and it represents the reference or desired signal, and x(t) is a vector. So K_r is a scalar, and K is a n x 1 matrix. The design question we need to answer is how to choose K and K_r.

We can rewrite the matrix system by making substitutions:

$$E\dot{x} = Ax + B(K_r r - Kx)$$

$$\dot{x} = E^{-1}Ax + E^{-1}B(K_r r - Kx)$$

$$\dot{x} = E^{-1}(A - BK)x + E^{-1}BK_r r$$

From this condition, we know that the closed-loop system is stable if

$$real\left(eig\left(E^{-1}(A - BK)\right)\right) < 0 \text{ for all eigenvalues}$$

There are many methods of choosing K and K_r. Today, we are going to introduce the pole placement method.

2. Pole placement: choosing K and Kr

The core idea of the pole placement method is to directly set the eigenvalues of A-BK by choosing the "feedback" matrix K. The closed-loop system is stable if the real parts of all eigenvalues of A-BK is negative. Even if the open loop system A may be unstable, we can try to stabilize a system by using the feedback matrix K. That is, try to design K such that the matrix A-BK is stable.

Here, we will go through an example. Consider the line-following example introduced in the third week of the semester, the A and B matrices are given by:

$$A = \begin{bmatrix} 0 & V \\ 0 & 0 \end{bmatrix}, B = \begin{bmatrix} 0 \\ \gamma \end{bmatrix}$$

In the closed-loop form, the A-BK matrix is given by:

$$A - BK = \begin{bmatrix} 0 & V \\ -\gamma K_1 & -\gamma K_2 \end{bmatrix}$$

For this problem, let's set $V = 1$ and $\gamma = 2$.

Let's aim to choose K such that we can set $\lambda_1 = -1, and\ \lambda_2 = -2$. We can write:

$$eigen(A - BK) \rightarrow 0 = (s - 0)\big(s - (-\gamma k_2)\big) - V(-\gamma k_1)$$

$$= s^2 + \gamma k_2 s + V\gamma k_1$$

$$= s^2 + 2k_2 s + 2k_1$$

We have chosen the eigenvalues, which give:

$$(s - \lambda_1)(s - \lambda_2) = s^2 + (-\lambda_1 - \lambda_2)s + \lambda_1\lambda_2$$

We can use this equation to solve for k_1 and k_2:

$$k_1 = 1; k_2 = 1.5$$

In practice, we can use the MATLAB function place(). The syntax is:

$$K = place(A, B, [\lambda_1, \lambda_2])$$

The method of pole placement directly sets the eigenvalues, which is intuitive on aspects of system convergence rate and stability. However, it is not intuitive on the control effort. How hard are we driving the actuators? We will introduce another control design method next class.

After we set the matrix K, we can choose the matrix K_r. Here, we can choose K_r to remove the steady-state error. At the steady-state condition, we have:

$$E\dot{x} = 0 = Ax + B(K_r r - Kx)$$

$$y = Cx = -C(A - BK)^{-1}BK_r r$$

To make sure the output y is equal to reference r, we need to have:

$$1 = -C(A - BK)^{-1}BK_r$$

So we need to set:

$$K_r = \frac{-1}{C(A - BK)^{-1}B}$$

3. Example: inverted pendulum

We introduced the inverted pendulum example last week. The state space equation is given by:

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

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

In lecture 16, we set the A, B, C, D, and E matrices to:

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

Suppose we want to maintain the pendulum at the inverted angle: $r = 0$. Then we simply have $K_r = 0$, and $u = -Kx$. Note that the system is intrinsically unstable. To stabilize it, let's set the eigenvalues of A-BK to -1 and -10. We can write:

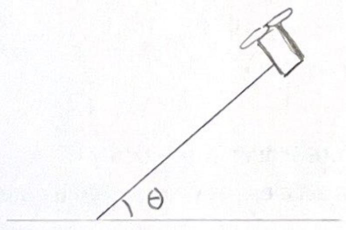
$$K = place(A, B, [-1 - 10])$$

MATLAB will return *K = [0.1080, 0.0110].*

While this is easy to implement, it is not intuitive on what are the optimal eigenvalues or pole locations. Again, we will introduce a more intuitive control approach in next lecture.

4. Example: propellor arm

In lab 2 and 3, we experimented with a propellor arm system using PD or PID control. Now we will introduce a state-space formulation and show the connection between this formulation and classical control. A sketch of the propellor arm is shown below.



This system is characterized by three differential equations:

$$\begin{aligned} \frac{d\theta_a}{dt} &= w_a \\ J_a\frac{dw_a}{dt} &= k_t l_a \frac{\Delta v_{emf}}{k_e} \\ J_m\frac{d}{dt}\Delta v_{emf} &= -\left(\frac{k_m k_e}{R_m + R_s} + k_f\right)\Delta v_{emf} + \frac{k_m k_e}{R_m + R_s}\Delta v_{pwm} \end{aligned}$$

In these equations, the control variable is Δv_{pwm}, the state variables are $[\theta_a, w_a, \Delta v_{emf}]$, and the output variable is θ_a. We can rewrite the equations in the matrix form:

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & J_a & 0 \\ 0 & 0 & J_m \end{bmatrix}\begin{bmatrix} \dot{\theta_a} \\ \dot{w_a} \\ \dot{\Delta v_{emf}} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & k_t l_a \frac{1}{k_e} \\ 0 & 0 & \frac{k_m k_e}{R_m + R_s} - k_f \end{bmatrix}\begin{bmatrix} \theta_a \\ w_a \\ \Delta v_{emf} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \frac{k_m k_e}{R_m + R_s} \end{bmatrix}\Delta v_{pwm}$$

$$\theta_a = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}\begin{bmatrix} \theta_a \\ w_a \\ \Delta v_{emf} \end{bmatrix} + 0 \times \Delta v_{pwm}$$

The core idea here is that θ_a *and* w_a describe the propellor arm dynamics, and Δv_{emf} describes the motor dynamics. Next, let's look at the connection of state space controller to a PD controller. In the case of PD control, we have:

$$u = K_p(\theta_d - \theta_a) + K_d(\dot{\theta_d} - \dot{\theta_a})$$

If we stabilize near the equilibrium condition (that is set $\dot{\theta}$ to 0), then we have:

$$u = K_p\theta_d - K_p\theta_a - K_d\dot{\theta_a}$$

We can compare this with state-space control:

$$u = K_r r - Kx = K_r\theta_d - k_1\theta_a - k_2\dot{\theta_a} - k_3\Delta v_{emf}$$

We can match coefficients to realize that the PD controller and the state-space controller are similar:

$$\begin{aligned} K_r &= K_p \\ k_1 &= K_p \\ k_2 &= K_d \\ k_3 &= 0 \end{aligned}$$

Note that a PD controller is the same as a state-space controller when the term k_3 is set to 0. So a state space controller is more general. We'll learn in next class how to design a good state space controller.