6.3100 Lecture 15 Notes – Spring 2024

Introduction to State-Space Control

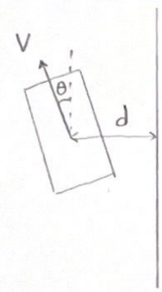
Dennis Freeman, Elfar Adalsteinsson, and Kevin Chen

Outline:

1. Formulate a control problem in the state space form
2. Solution of a state space control system
3. Stability of a state space controller

1. <u>Formulate a control problem in the state space form</u>

In the beginning of this semester, we introduced a line-following example formulated as a 2nd order discrete time control system. This example is illustrated below:



$$d[n] = d[n-1] + \Delta T\, V \sin\theta[n-1]$$
$$\theta[n] = \theta[n-1] + \Delta T\, \omega[n-1]$$
$$\omega[n] = \gamma u[n]$$

In lecture 5, we converted these equations into a 2nd order discrete time system and then implemented P and PD controllers. In this lecture, let's revisit the same problem in the continuous time domain. The system differential equations are given by:

$$\frac{d\theta}{dt} = \gamma u$$
$$\frac{dd}{dt} = V\sin\theta \approx V\theta$$

Using the classical control approach, we can rewrite these two equations:

$$\frac{d}{dt}\left(\frac{d}{dt}d\right) = V\frac{d}{dt}\theta = V\gamma u$$

If we use proportional control, then we set $u = K_p(d_d - d)$. The system equation becomes:

$$\ddot{d} = V\gamma K_p(d_d - d) => \ddot{d} + V\gamma K_p d = V\gamma K_p d_d$$

Note that this controller does not work very well because it will oscillate around the set point. To improve performance, we need to implement a PD controller.

The main message of today's lecture is to introduce a new control approach: state-space control. In contrast to solving a high order differential equation, we will solve a system of linear first order differential equations. This new approach is more systematic and it is widely used in many modern control systems.

Coming back to this control problem, let's ask some important questions.

First, what set of variables completely specify the physical system? In this problem, θ and d completely describe the system. If we know these two values, then we know the robot's distance to the line and its heading angle. θ and d are called the "states" of the system.

Second, what is my control signal? In this problem, u is the control input.

Third, what is the output of this problem? Here, the distance d is the output variable.

The state space representation has a generic form, which is given by:

$$E\dot{x} = A'x + B'u$$

$$y = Cx + Du$$

Here x is the state variables, y is the output, and u is the controller input. $A', B', C, D, and\ E$ are the matrices describing the relationships between the variables. For this line-following example, the state space representation is written as:

$$\dot{\theta} = gu, \quad \dot{d} = V\theta$$

$$\underbrace{\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}}_{E} \underbrace{\begin{pmatrix} \dot{d} \\ \dot{\theta} \end{pmatrix}}_{\dot{x}} = \underbrace{\begin{pmatrix} 0 & V \\ 0 & 0 \end{pmatrix}}_{A'} \underbrace{\begin{pmatrix} d \\ \theta \end{pmatrix}}_{x} + \underbrace{\begin{pmatrix} 0 \\ g \end{pmatrix}}_{B'} \underbrace{u}_{u}$$

$$\underbrace{d}_{y} = \underbrace{\begin{pmatrix} 1 & 0 \end{pmatrix}}_{C} \underbrace{\begin{pmatrix} d \\ \theta \end{pmatrix}}_{x} + \underbrace{0}_{D} \underbrace{u}_{u}$$

size of each term

E	2×2
x	2×1
A'	2×2
B'	2×1
u	1×1
y	1×1
C	1×2
D	1×1

It is very important to keep track of the sizes of each variable and matrix.

This example is called a single-input-single-output "SISO" system because the size of control "u" is 1 x 1 and the output "y" is 1 x 1. In more complex examples, a control system can have multiple inputs and multiple outputs. Those are called "MIMO" systems. In this course, we will mostly focus on SISO systems.

To simplify our analysis, we will make another transformation. We will assume the matrix E is invertible, and we will multiply the first equation by E^{-1}. The first equation becomes:

$$\dot{x} = E^{-1}A'x + E^{-1}B'u$$

Finally, we can "redefine" new A and B matrices, where $A = E^{-1}A'$, and $B = E^{-1}B'$. In the rest of this course, we will deal with the simplified problem:

$$\dot{x} = Ax + Bu$$

$$y = Cx + Du$$

After we identify the states x, control input u, output y, and matrices $A, B, C, and\ D,$ then we have fully prescribe the system. Now we need to design a good controller. It turns out we can always use a "proportional-like" controller:

$$u = K_r r - Kx$$

Here r is the reference output that I want my system to follow, and K_r and K are the variables that we need to design. Note that for a SISO system, K_r is a scalar value, and K is a row vector that has the same dimension (transpose) as x. For this example, K is a 2 x 1 vector.

Now let's plug in our controller design back into the system equation. We have:

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

$$y = Cx + D(K_r r - Kx) = (C - DK)x + DK_r r$$

The main design question is how to choose good K and K_r given a control problem.

2. Solution of a state space control problem

Here we want to solve the problem:

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

$$y = (C - DK)x + DK_r r$$

We make a simplifying assumption: $D = 0$. Setting $D = 0$ means the output is not directly related to the input, which is true for most control systems. In particular, it is true for all the control examples we consider in this class. With these simplifications, our system equation becomes:

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

$$y = Cx$$

Solving this matrix system requires some advanced linear algebra and differential equation results, so we will state the solution without giving a formal proof:

$$x(t) = \mathrm{e}^{(A-BK)t}\, x_0 + \int_0^t e^{(A-BK)(t-\tau)}\, BK_r r(\tau) d\tau$$

$$y = Cx$$

Here x_0 is the initial condition of the system. This is a complicated expression that involves a convolution integral. For simple desired reference function $r(t)$, we can analytically solve the integral. For complex expression of $r(t)$, we will mostly use computational tools.

3. Stability of a state space controller

Today, we are going to analyze system stability. Both the first and second terms are in the form of matrix exponential:

$$e^{(A-BK)t}$$

We have $x(t) \propto e^{(A-BK)t}x_0$. Note that $x(t)$ is a column vector, and $e^{(A-BK)t}$ is called a matrix exponential, and it has the matrix dimension. Before we analyze stability, let's discuss how to calculate a matrix exponential.

First, raising a matrix to the exponential is NOT equal to raising every matrix element to the exponential:

$$e^{\mathrm{P}} \neq \begin{pmatrix} e^{P_{11}} & e^{P_{12}} & \dots & e^{P_{1n}} \\ e^{P_{21}} & e^{P_{22}} & \dots & e^{P_{2n}} \\ \dots & \dots & \dots & \dots \\ e^{P_{n1}} & e^{P_{n2}} & \dots & e^{P_{nn}} \end{pmatrix}$$

Similar to scalar exponential functions, a matrix exponential is defined in terms of Taylor expansion. First, we write the definition:

$$e^P = I + P + \frac{1}{2!}P^2 + \frac{1}{3!}P^3 + \cdots$$

where P is a matrix. Theoretically, we now know how to calculate the matrix exponential as an infinite sum, but this is very tedious. Let's look for a simpler approach.

Again, we want to evaluate the matrix exponential $e^{(A-BK)t}$. Let $P = A - BK$. Now let's make another simplification. Let's assume the matrix P has no repeated eigenvalues (otherwise we need to invoke a more general theory called "Jordan Form"). Suppose P is a n x n matrix, we can write its eigenvalues and eigenvectors:

$$Pv_i = \lambda_i v_i$$

We can concatenate all eigenvectors into a matrix:

$$P(v_1|v_2|\dots|v_n) = (v_1|v_2|\dots|v_n)\begin{pmatrix} \lambda_1 & & & \\ & \lambda_2 & & \\ & & \dots & \\ & & & \lambda_n \end{pmatrix}$$

$$PV = V\Lambda$$

where

$$V = (v_1|v_2| \ldots |v_n) \text{ and } \Lambda = \begin{pmatrix} \lambda_1 & & & \\ & \lambda_2 & & \\ & & \ldots & \\ & & & \lambda_n \end{pmatrix}$$

We can write the matrix P as:

$$P = V\Lambda V^{-1}$$

Now we can expand the matrix exponential:

$$\begin{aligned}
e^{Pt} &= I + Pt + \frac{1}{2!}(Pt)^2 + \frac{1}{3!}(Pt)^3 + \cdots \\
&= I + V\Lambda V^{-1}t + \frac{1}{2!}(V\Lambda V^{-1}t)^2 + \frac{1}{3!}(V\Lambda V^{-1}t)^3 + \cdots \\
&= I + V\Lambda V^{-1}t + \frac{1}{2!}(V\Lambda V^{-1}V\Lambda V^{-1})t^2 + \frac{1}{3!}(V\Lambda V^{-1}V\Lambda V^{-1}V\Lambda V^{-1})t^3 + \cdots \\
&= I + V\Lambda V^{-1}t + \frac{1}{2!}(V\Lambda\Lambda V^{-1})t^2 + \frac{1}{3!}(V\Lambda\Lambda\Lambda V^{-1})t^3 + \cdots \\
&= VIV^{-1} + V\Lambda V^{-1}t + \frac{1}{2!}(V\Lambda^2 V^{-1})t^2 + \frac{1}{3!}(V\Lambda^3 V^{-1})t^3 + \cdots \\
&= V\left(I + \Lambda t + \frac{1}{2!}(\Lambda t)^2 + \frac{1}{3!}(\Lambda t)^3 + \cdots\right)V^{-1} \\
&= Ve^{\Lambda t}V^{-1}
\end{aligned}$$

This is a very convenient expression because we have:

$$e^{\Lambda t} = \begin{pmatrix} e^{\lambda_1 t} & & & \\ & e^{\lambda_2 t} & & \\ & & \ldots & \\ & & & e^{\lambda_n t} \end{pmatrix}$$

This is true only for a diagonal matrix.

If we want this matrix to be stable, then $e^{\lambda_1 t} \to 0$ for all λ_i. This implies $Re(\lambda_i) < 0$ for all i. Our control system is stable if $Re\left(eig\ (A - BK)\right) < 0$. So when we are designing the control matrix K, one criteria is that we need to choose K such that the above condition is met.