

# Lecture 2: Linear Systems

Erfan Nozari

November 22, 2021

In this note, we begin our actual study of linear systems & control, building on the preliminaries and linear algebra we have learned so far. I will start by some basic introduction to signals & systems (in general, not necessarily linear). Then I will introduce linearity, for functions and for systems. This will allow us to define the standard linear state space model, which we will study for the rest of this course.

## Contents

2.1	Signals & Systems	2
2.1.1	Example (Remote control car)	2
2.1.2	Example (Remote control car, revisited)	3
2.1.3	Example (Pendulum)	4
2.1.4	Exercise (Active Suspension System)	5
2.2	Linear Systems: Basic Definitions	6
2.2.1	Definition (Linear function)	6
2.2.2	Example (Affine functions)	6
2.2.3	Definition (Linear state-space system)	6
2.2.4	Exercise (Examples, revisted)	6
2.3	Solutions of LTI Systems	6
2.3.1	Theorem (Properties of matrix exponential)	9
2.3.2	MATLAB (Matrix exponential)	9
2.3.3	Example (Matrix exponential)	9
2.4	Superposition Property	10
2.5	Impulse Response	11
2.5.1	Example (Impulse response of MSD system)	12
2.5.2	Exercise (Impulse response of MSD system)	13
2.5.3	Exercise (Impulse response via diagonalization)	13

## 2.1 Signals & Systems

Before we can talk about controlling a system, we should know what a ‘system’ formally means! We have already been informally talking about ODEs as systems, so we here make it a bit more formal. A system is a mapping from one or more input ‘signals’ (signal = function of time) to one or more output signals:

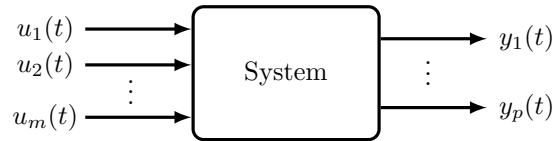


Figure 1: A general system

In this general form, this is called a Multi-Input Multi-Output (MIMO) system. If the number of inputs  $m$  equals the number of outputs  $p$  equals 1, the system is called Single-Input Single-Output (SISO).

**Example 2.1.1 (Remote control car)** A simple remote control car may have several inputs and outputs, for example:



□

A ‘model’ of a system is a mathematical formulation of the relationship between the inputs and outputs. Among all the different forms that a model can have, the most common form for physical systems is differential-algebraic equations. Among the different forms of differential-algebraic equations, we are interested in the ‘standard state space model’:

$$\dot{\mathbf{x}}(t) = \frac{d}{dt}\mathbf{x}(t) = f(\mathbf{x}(t), \mathbf{u}(t)), \quad \mathbf{x}(0) = \mathbf{x}_0 \quad (2.1a)$$

$$\mathbf{y}(t) = h(\mathbf{x}(t), \mathbf{u}(t)) \quad (2.1b)$$

In Eq. (2.1),

- $\mathbf{u}(t)$  and  $\mathbf{y}(t)$  are the vectors of input and output signals, respectively. If  $m = p = 1$ , then the system is SISO.
- We have a new vector  $\mathbf{x}(t)$  which is called the systems “state” and is internal to the system. Note that state is the only vector whose derivative appears in Eq. (2.1).
- The functions  $f$  and  $h$  map vectors to vectors. Such functions are called ‘vector fields’ and equations like Eq. (2.1) are called to be in ‘vector form’. For example, Eq. (2.1a) is equivalent to the expanded,

scalar form

$$\begin{aligned}\dot{x}_1(t) &= f_1(x_1(t), \dots, x_n(t), u_1(t), \dots, u_m(t)) \\ \dot{x}_2(t) &= f_2(x_1(t), \dots, x_n(t), u_1(t), \dots, u_m(t)) \\ &\vdots \\ \dot{x}_n(t) &= f_n(x_1(t), \dots, x_n(t), u_1(t), \dots, u_m(t))\end{aligned}$$

and similarly for Eq. (2.1b).

- Eq. (2.1a) is a first-order ODE.

**Example 2.1.2 (Remote control car, revisited)** Consider again the remote control car of Example 2.1.1. Based on simple Newtonian dynamics, we have

$$\begin{aligned}\dot{p}_x(t) &= v_\ell(t) \cos \theta(t) \\ \dot{p}_y(t) &= v_\ell(t) \sin \theta(t) \\ \dot{\theta}(t) &= v_a(t)\end{aligned}$$

Notice that this is already in the form of Eq. (2.1a) if we define

$$\begin{aligned}\mathbf{x}(t) &= \begin{bmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \end{bmatrix} = \begin{bmatrix} p_x(t) \\ p_y(t) \\ \theta(t) \end{bmatrix} \\ \mathbf{u}(t) &= \begin{bmatrix} u_1(t) \\ u_2(t) \end{bmatrix} = \begin{bmatrix} v_\ell(t) \\ v_a(t) \end{bmatrix} \\ f(\mathbf{x}, \mathbf{u}) &= \begin{bmatrix} f_1(\mathbf{x}, \mathbf{u}) \\ f_2(\mathbf{x}, \mathbf{u}) \\ f_3(\mathbf{x}, \mathbf{u}) \end{bmatrix} = \begin{bmatrix} u_1 \cos x_3 \\ u_1 \sin x_3 \\ u_2 \end{bmatrix}\end{aligned}$$

Defining the output  $\mathbf{y}(t)$  depends on what variables you can measure and what variables you are interested in. Consider three example scenarios:

- You can measure all three state variables  $p_x(t), p_y(t), \theta(t)$  and you are interested in all three of them. Then you would define

$$\mathbf{y}(t) = h(\mathbf{x}(t), \mathbf{u}(t)) = \mathbf{x}(t) \Rightarrow \begin{bmatrix} y_1(t) \\ y_2(t) \\ y_3(t) \end{bmatrix} = \begin{bmatrix} p_x(t) \\ p_y(t) \\ \theta(t) \end{bmatrix}$$

- You can measure all three state variables but you are only interested in the distance between the car and its starting location. Then you would define

$$y(t) = h(\mathbf{x}(t), \mathbf{u}(t)) = \sqrt{(x_1(t) - x_1(0))^2 + (x_2(t) - x_2(0))^2} = \|\mathbf{x}(t) - \mathbf{x}_0\|$$

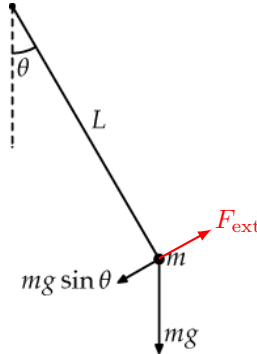
- You are interested in all three state variables, but can only measure the heading angle of the car. Then, you can define

$$y(t) = h(\mathbf{x}(t), \mathbf{u}(t)) = x_3(t)$$

or any function of  $x_3(t)$ , but the states  $x_1(t)$  and  $x_2(t)$  remain internal to the system and we cannot include them in the output.

□

**Example 2.1.3 (Pendulum)** Consider a simple pendulum with friction and a zero-mass rod:



Newton's second law in angular coordinates gives

$$\underbrace{mL^2}_{\substack{\text{moment} \\ \text{of} \\ \text{inertia}}} \underbrace{\ddot{\theta}(t)}_{\substack{\text{CCW} \\ \text{angular} \\ \text{accel.}}} = \overbrace{(-mg \sin \theta(t) - F_{\text{fric}}(\dot{\theta}(t)) + F_{\text{ext}}(t))L}^{\text{total CCW torque}}$$

$\underbrace{-mg \sin \theta(t)}_{\substack{\text{tangential} \\ \text{projection} \\ \text{of weight}}}$ 
 $\underbrace{-F_{\text{fric}}(\dot{\theta}(t))}_{\text{friction}}$ 
 $+ F_{\text{ext}}(t)$

The function  $F_{\text{fric}}(\cdot)$  gives friction as a function of velocity. A common form is  $F_{\text{fric}}(\dot{\theta}) = b\dot{\theta}$  for some constant  $b$ , but we keep it general.

The above equation does not look like the standard state-space form Eq. (2.1a) on the surface, but it can easily be put in that form. The major difference is that this equation is a second-order differential equation (involving  $\ddot{\theta}(t) = \frac{d^2}{dt^2}\theta(t)$ ) but Eq. (2.1a) is first order. The simple trick is to define the state as

$$\mathbf{x}(t) = \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} = \begin{bmatrix} \theta(t) \\ \dot{\theta}(t) \end{bmatrix}$$

Then we can rewrite Newton's second law as

$$\begin{aligned} \dot{x}_1(t) &= x_2(t) \\ mL^2 \dot{x}_2(t) &= -mgL \sin x_1(t) - F_{\text{fric}}(x_2(t))L + F_{\text{ext}}(t)L \end{aligned}$$

The first equation is trivial but is necessary to put the dynamics in the standard state-space form (why?). If we define the input as

$$u(t) = F_{\text{ext}}(t)$$

then we have

$$\begin{aligned} \dot{x}_1(t) &= x_2(t) \\ \dot{x}_2(t) &= -\frac{g}{L} \sin x_1(t) - \frac{1}{mL} F_{\text{fric}}(x_2(t)) + \frac{1}{mL} u(t) \end{aligned}$$

which is in the standard form.

Similar to Example 2.1.2, definition of output depends on what you can measure and are interested in. A simple choice is

$$y(t) = x_1(t) = [1 \quad 0] \mathbf{x}(t) = \theta(t).$$

If you are, for example, interested in the total energy of the pendulum (relative to its hanging position), and, have access to both  $\theta$  and  $\dot{\theta}$  (measured or numerically computed), then you can define

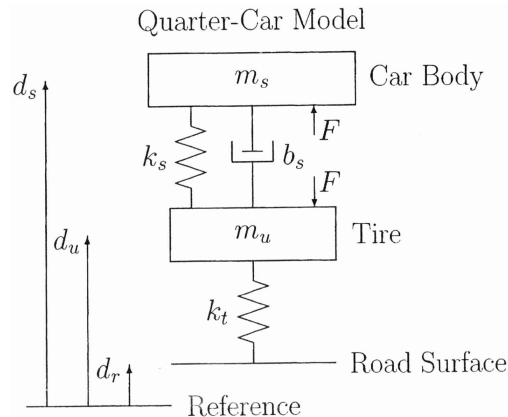
$$y(t) = E_{\text{kinetic}}(t) + E_{\text{potential}}(t) = \frac{1}{2}mL^2x_2(t)^2 + mgL(1 - \cos x_1(t))$$

and so on.  $\square$

The pattern that we saw in the above few examples for writing the dynamics of electromechanical systems in the standard state space form is general. For mechanical systems,

- the state equations are typically the Newton's second law, either in linear or angular coordinates. These are second-order ODEs that include second-order derivatives of (linear or angular) position.
- To put them in state space form, we take the position and its derivative (velocity) as state variables, and
- we use the descriptive equations of the elements (mass, spring, damper, etc) to simplify the equations.

**Exercise 2.1.4 (Active Suspension System)** The picture below shows a simplified point-mass model of a car active suspension system. The force  $F$  is an external force applied by the active suspension controller, the variables  $d_s, d_u, d_r$  are respectively the distance of the car body, tire, and ground from a reference level,  $m_s$  and  $m_u$  are the mass of the car and the tire,  $k_s$  and  $b_s$  is the constants of the (passive) suspension system's spring and shock absorber, and  $k_t$  is the spring constant of the tire. Assume  $d_r = 0$  and take the elevation of the car ( $d_s$ ) as the system's output.



Derive the dynamical equations governing the full system, and put it in the standard state space form.  $\square$

Notice that so far, we have not said anything about linearity, only about systems in general (linear or nonlinear). Our course is, however, about linear systems, which are a theoretically tiny but practically very common subset of all systems:

## 2.2 Linear Systems: Basic Definitions

Recall that a function is called linear if it is additive and homogeneous:

**Definition 2.2.1 (Linear function)** A function (vector field)

$$\mathbf{z} = f(\mathbf{x})$$

where  $\mathbf{x} \in \mathbb{R}^n$  and  $\mathbf{z} \in \mathbb{R}^m$ , is linear if it has the form

$$\mathbf{z} = f(\mathbf{x}) = \mathbf{A}\mathbf{x} \tag{2.2}$$

for some matrix  $\mathbf{A} \in \mathbb{R}^{m \times n}$ . □

Still not making sense of why we call such functions ‘linear’? Because, in the simplest case of scalar functions ( $m = n = 1$ ), Eq. (2.2) becomes

$$z = ax$$

for some scalar  $a$ , which is the equation of a line going through the origin and having the slope  $a$ .

**Example 2.2.2 (Affine functions)** Confusingly enough, functions of the form

$$z = ax + b$$

are not “linear”! (show why, using the definition). Even though they also represent a line, such functions are called “affine”. Similarly, in vector form,  $\mathbf{z} = \mathbf{A}\mathbf{x} + \mathbf{b}$  is affine, but not linear. □

Now that we know what a linear function is, we can easily define what a linear system is.

**Definition 2.2.3 (Linear state-space system)** A linear state-space system is any system of the form in Eq. (2.1) (in Lecture Note 1) in which the functions  $f$  and  $h$  are linear:

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t), \quad \mathbf{x}(0) = \mathbf{x}_0 \tag{2.3a}$$

$$\mathbf{y}(t) = \mathbf{C}\mathbf{x}(t) + \mathbf{D}\mathbf{u}(t) \tag{2.3b}$$

□

Note that the above definition is the definition of a linear state-space system. In general, a linear system does not always need to be in state-space form. However, in this course, whenever we say a ‘linear system’, we mean a linear state-space system.

**Exercise 2.2.4 (Examples, revisited)** Is the pendulum in Example 2.1.3 a linear system? Why or why not? What about the remote control car in Example 2.1.2? □

For the rest of this course, we will be studying linear systems of the form 2.3 and their properties. In addition to being linear, they are time-invariant (hence “LTI”, for linear time-invariant) and continuous-time. There are linear systems that are time-varying and/or discrete-time, but we won’t study them in this course.

## 2.3 Solutions of LTI Systems

Recall that an LTI system in continuous-time has the form

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t), \quad \mathbf{x}(0) = \mathbf{x}_0, \quad t \in \mathbb{R} \tag{2.4a}$$

$$\mathbf{y}(t) = \mathbf{C}\mathbf{x}(t) + \mathbf{D}\mathbf{u}(t) \tag{2.4b}$$

To do anything with this model, the first thing that we need to be able to do is find the solutions of these equations! Eq. (2.4a) is a differential equation, and solving it means finding the function  $\mathbf{x}(t)$  that satisfies it. Note that  $u(t)$  is always a given signal, and we only solve these equations for the unknown signal  $\mathbf{x}(t)$ . Once we obtain  $\mathbf{x}(t)$ , we can simply substitute it in  $\mathbf{y}(t) = h(\mathbf{x}(t), \mathbf{u}(t))$  to obtain  $\mathbf{y}(t)$ . So the main challenge in finding the solution to an LTI system is solving the differential equation for  $\mathbf{x}(t)$ .

Let's start by the simplest case possible, and move up to the general case in Eq. (2.4a):

1. **Scalar case without input.** Assume that  $n = 1$  and  $\mathbf{B} = 0$ , which simplifies Eq. (2.4a) to

$$\dot{x}(t) = ax(t), \quad x(0) = x_0 \quad (2.5)$$

You know from differential equations that the solution is  $x(t) = x_0 e^{at}$ . But do you remember why? You can see it immediately from the definition of the exponential function:

$$\begin{aligned} e^{at} &= 1 + at + \frac{1}{2}(at)^2 + \frac{1}{3!}(at)^3 + \dots \\ \frac{d}{dt}e^{at} &= 0 + a + \frac{1}{2}a^2 2t + \frac{1}{3!}a^3 3t^2 + \dots \\ &= a\left(1 + at + \frac{1}{2}(at)^2 + \dots\right) \\ &= ae^{at} \end{aligned}$$

Finally, because differentiation is a linear operation (which you now know what it means!), for any  $k$  it also holds that

$$\frac{d}{dt}ke^{at} = k\frac{d}{dt}e^{at} = kae^{at} = a(ke^{at})$$

and the value of  $k$  can be found by substituting  $x(t) = ke^{at}$  into the initial condition, giving

$$ke^0 = x_0 \Rightarrow k = x_0 \Rightarrow x(t) = x_0 e^{at}$$

This was perhaps a long derivation for the simple ODE in Eq. (2.5), but we need it to solve the more general cases next:

2. **Vector case without input.** Still let  $\mathbf{B} = 0$ , and let's solve to the vector ODE

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t), \quad \mathbf{x}(0) = \mathbf{x}_0 \quad (2.6)$$

Notice that the core of our solution method for the scalar case above was the series expansion of the exponential function. So we just generalize that to matrices!

$$e^{\mathbf{A}t} \stackrel{\text{def}}{=} \mathbf{I} + \mathbf{A}t + \frac{1}{2}(\mathbf{A}t)^2 + \frac{1}{3!}(\mathbf{A}t)^3 + \dots$$

this is called the matrix exponential. Notice that on the right hand side we have sums and products of square,  $n$  by  $n$  matrices, so the matrix exponential  $e^{\mathbf{A}t}$  is also a square  $n$  by  $n$  matrix. But it is not the same as element-wise exponentiation ( $(e^{\mathbf{A}t})_{ij}$  is generally not equal to  $e^{\mathbf{A}_{ij}t}$ ). We will see how to compute this matrix, but for now let's see how it helps us solve the ODE:

$$\begin{aligned} \frac{d}{dt}e^{\mathbf{A}t} &= \frac{d}{dt}\left(\mathbf{I} + \mathbf{A}t + \frac{1}{2}(\mathbf{A}t)^2 + \frac{1}{3!}(\mathbf{A}t)^3 + \dots\right) \\ &= \mathbf{0} + \mathbf{A} + \frac{1}{2}\mathbf{A}^2 2t + \frac{1}{3!}\mathbf{A}^3 3t^2 + \dots \\ &= \mathbf{A}\left(\mathbf{I} + \mathbf{A}t + \frac{1}{2}(\mathbf{A}t)^2 + \dots\right) \\ &= \mathbf{A}e^{\mathbf{A}t} \end{aligned}$$

This is great, right?!  $\mathbf{X}(t) = e^{\mathbf{A}t}$  satisfies  $\dot{\mathbf{X}}(t) = \mathbf{A}\mathbf{X}(t)$ , which is very similar to our ODE in Eq. (2.6). The difference is that  $\mathbf{X}(t) = e^{\mathbf{A}t}$  is a matrix, while our state  $\mathbf{x}(t)$  is a vector. Here we can use the linearity of differentiation again. Because differentiation is a linear operation, for any vector  $\mathbf{k} \in \mathbb{R}^n$ ,

$$\frac{d}{dt}e^{\mathbf{A}t}\mathbf{k} = \left(\frac{d}{dt}e^{\mathbf{A}t}\right)\mathbf{k} = \left(\mathbf{A}e^{\mathbf{A}t}\right)\mathbf{k} = \mathbf{A}\left(e^{\mathbf{A}t}\mathbf{k}\right)$$

so any vector  $\mathbf{x}(t) = e^{\mathbf{A}t}\mathbf{k}$  for any  $\mathbf{k}$  satisfies  $\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t)$ . The value of  $\mathbf{k}$  can then be found using the initial condition in Eq. (2.6). First, notice that

$$e^{\mathbf{A}t}\big|_{t=0} = e^{\mathbf{0}} = \mathbf{I} + \mathbf{0} + \frac{1}{2}\mathbf{0}^2 + \frac{1}{3!}\mathbf{0}^3 + \dots = \mathbf{I}$$

so substituting  $t = 0$  in  $\mathbf{x}(t) = e^{\mathbf{A}t}\mathbf{k}$  gives

$$\mathbf{x}(0) = \mathbf{I}\mathbf{k} = \mathbf{k}$$

so  $\mathbf{k} = \mathbf{x}_0$  and the solution to Eq. (2.6) is

$$\mathbf{x}(t) = e^{\mathbf{A}t}\mathbf{x}_0$$

3. **General case.** Now let's add the input and solve the general case in Eq. (2.4a). The trick is simple, multiply both sides by  $e^{-\mathbf{A}t}$  (from the left)! Here is how it works:

$$\begin{aligned}\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) &\Rightarrow e^{-\mathbf{A}t}\dot{\mathbf{x}}(t) = e^{-\mathbf{A}t}\mathbf{A}\mathbf{x}(t) + e^{-\mathbf{A}t}\mathbf{B}\mathbf{u}(t) \\ &\Rightarrow e^{-\mathbf{A}t}\dot{\mathbf{x}}(t) - e^{-\mathbf{A}t}\mathbf{A}\mathbf{x}(t) = e^{-\mathbf{A}t}\mathbf{B}\mathbf{u}(t)\end{aligned}$$

But notice that the left hand side is

$$e^{-\mathbf{A}t}\dot{\mathbf{x}}(t) - e^{-\mathbf{A}t}\mathbf{A}\mathbf{x}(t) = \frac{d}{dt}\left(e^{-\mathbf{A}t}\mathbf{x}(t)\right)$$

(just use the product rule for differentiation, which also holds for matrices, and the fact that  $e^{-\mathbf{A}t}\mathbf{A} = \mathbf{A}e^{-\mathbf{A}t}$ , which follows from the definition of matrix exponential.) So now you have

$$\frac{d}{dt}\left(e^{-\mathbf{A}t}\mathbf{x}(t)\right) = e^{-\mathbf{A}t}\mathbf{B}\mathbf{u}(t)$$

and we can take the integral of both sides to cancel the derivative

$$\begin{aligned}\int_0^t \frac{d}{d\tau}\left(e^{-\mathbf{A}\tau}\mathbf{x}(\tau)\right)d\tau &= \int_0^t e^{-\mathbf{A}\tau}\mathbf{B}\mathbf{u}(\tau)d\tau \\ e^{-\mathbf{A}t}\mathbf{x}(t) - e^{\mathbf{0}}\mathbf{x}(0) &= \int_0^t e^{-\mathbf{A}\tau}\mathbf{B}\mathbf{u}(\tau)d\tau \\ e^{-\mathbf{A}t}\mathbf{x}(t) &= \mathbf{x}_0 + \int_0^t e^{-\mathbf{A}\tau}\mathbf{B}\mathbf{u}(\tau)d\tau\end{aligned}$$

Now we can multiply both sides by the inverse of  $e^{-\mathbf{A}t}$  which, not surprisingly, is  $e^{\mathbf{A}t}$ :

$$\left(e^{\mathbf{A}t}\right)^{-1} = e^{-\mathbf{A}t}$$

So we get

$$\mathbf{x}(t) = e^{\mathbf{A}t}\mathbf{x}_0 + e^{\mathbf{A}t}\int_0^t e^{-\mathbf{A}\tau}\mathbf{B}\mathbf{u}(\tau)d\tau$$



which can also be written as

$$\boxed{\mathbf{x}(t) = e^{\mathbf{A}t}\mathbf{x}_0 + \int_0^t e^{\mathbf{A}(t-\tau)}\mathbf{B}\mathbf{u}(\tau)} \quad (2.7)$$

because you can take  $e^{\mathbf{A}t}$  inside the integral and use the fact that  $e^{\mathbf{A}t}e^{-\mathbf{A}\tau} = e^{\mathbf{A}(t-\tau)}$ . We can finally substitute the solution of  $\mathbf{x}$  into the output equation Eq. (2.4b) to get

$$\mathbf{y}(t) = \mathbf{C}e^{\mathbf{A}t}\mathbf{x}_0 + \mathbf{C} \int_0^t e^{\mathbf{A}(t-\tau)}\mathbf{B}\mathbf{u}(\tau) + \mathbf{D}\mathbf{u}(t) \quad (2.8)$$

You can see that the core of the solutions above, similar to the scalar case, is the matrix exponential. In deriving the solution, I mentioned some properties of it, but let's have them in one place more formally:

**Theorem 2.3.1 (Properties of matrix exponential)** For any square matrix  $\mathbf{A}$ , the matrix exponential is defined as

$$e^{\mathbf{A}} = \mathbf{I} + \mathbf{A} + \frac{1}{2}\mathbf{A}^2 + \frac{1}{3!}\mathbf{A}^3 + \dots$$

and has the properties that

- (i)  $e^{\mathbf{0}} = \mathbf{I}$
- (ii)  $e^{\mathbf{A}}$  is always invertible (nonsingular), and its inverse is  $(e^{\mathbf{A}t})^{-1} = e^{-\mathbf{A}t}$
- (iii)  $(e^{\mathbf{A}})^T = e^{\mathbf{A}^T}$
- (iv)  $e^{\mathbf{A}}$  commutes with any other “polynomial” function of  $\mathbf{A}$ . In particular,  $e^{\mathbf{A}}\mathbf{A} = \mathbf{A}e^{\mathbf{A}}$
- (v)  $e^{\mathbf{A}}e^{\mathbf{B}} = e^{\mathbf{A}+\mathbf{B}}$  is not always true. But it does hold if  $\mathbf{A}\mathbf{B} = \mathbf{B}\mathbf{A}$
- (vi) In particular,  $e^{\mathbf{A}t}e^{\mathbf{A}\tau} = e^{\mathbf{A}(t+\tau)}$

□

**MATLAB 2.3.2 (Matrix exponential)** In MATLAB, if you have a matrix  $\mathbf{A}$ ,

```
1 B = exp(A);
```

does not give you the matrix exponential! It gives you the element-wise exponential of  $\mathbf{A}$ . To get the matrix exponential, use

```
1 B = expm(A);
```

□

If the matrix exponential is still unclear, an example hopefully helps:

**Example 2.3.3 (Matrix exponential)** Consider an arbitrary matrix, for example

$$\mathbf{A} = \begin{bmatrix} 1 & 2 & 4 \\ 2 & 3 & 2 \\ 4 & 1 & 3 \end{bmatrix}$$

To compute  $\mathbf{B} = e^{\mathbf{A}}$ , we have to compute the infinite matrix series

$$\mathbf{B} = \mathbf{I} + \mathbf{A} + \frac{1}{2}\mathbf{A}^2 + \frac{1}{3!}\mathbf{A}^3 + \dots$$

But in reality, we can only compute this for a finite number of terms, giving

$$\mathbf{B} \simeq \mathbf{I} + \mathbf{A} + \frac{1}{2}\mathbf{A}^2 + \frac{1}{3!}\mathbf{A}^3 + \dots + \frac{1}{k!}\mathbf{A}^k, \quad \text{for sufficiently large } k$$

For the  $\mathbf{A}$  above, you can check via MATLAB that

$$\mathbf{I} + \mathbf{A} + \frac{1}{2}\mathbf{A}^2 + \frac{1}{3!}\mathbf{A}^3 + \dots + \frac{1}{10!}\mathbf{A}^{10} = \begin{bmatrix} 467.3772 & 342.0297 & 578.3641 \\ 447.7252 & 333.7776 & 553.4206 \\ 525.5164 & 382.4057 & 650.8638 \end{bmatrix}$$

Is this sufficiently close? We can compare it with a larger  $k$ , for example

$$\mathbf{I} + \mathbf{A} + \frac{1}{2}\mathbf{A}^2 + \frac{1}{3!}\mathbf{A}^3 + \dots + \frac{1}{20!}\mathbf{A}^{20} = \begin{bmatrix} 537.6017 & 393.5083 & 665.2758 \\ 515.2491 & 383.2764 & 636.9899 \\ 604.4054 & 440.2357 & 748.4988 \end{bmatrix}$$

which is quite different, so  $k = 10$  hadn't been enough. Comparing with  $k = 30$ , we see that

$$\mathbf{I} + \mathbf{A} + \frac{1}{2}\mathbf{A}^2 + \frac{1}{3!}\mathbf{A}^3 + \dots + \frac{1}{30!}\mathbf{A}^{30} = \begin{bmatrix} 537.6194 & 393.5213 & 665.2978 \\ 515.2662 & 383.2889 & 637.0110 \\ 604.4254 & 440.2503 & 748.5235 \end{bmatrix}$$

which is very close to the sum for  $k = 20$ . At the end of the day, when to stop depends on what accuracy you need. In this case,

$$e^{\mathbf{A}} = \begin{bmatrix} 537.6194 & 393.5213 & 665.2978 \\ 515.2662 & 383.2889 & 637.0110 \\ 604.4254 & 440.2503 & 748.5235 \end{bmatrix}$$

so the sum up to  $k = 30$  is accurate at least to 4 decimals. □

## 2.4 Superposition Property

An important property of linear systems is the so-called superposition property. Reading the previous discussions, you might have noticed that output of the system,

$$\mathbf{y}(t) = \underbrace{\mathbf{C}e^{\mathbf{A}t}\mathbf{x}_0}_{\text{Due to } \mathbf{x}_0} + \underbrace{\mathbf{C} \int_0^t e^{\mathbf{A}(t-\tau)} \mathbf{B}\mathbf{u}(\tau) + \mathbf{D}\mathbf{u}(t)}_{\text{Due to input } \mathbf{u}(t)}$$

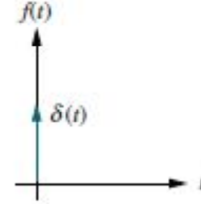
can be broken into a term only due to the initial condition  $\mathbf{x}_0$  and a term only due to the input  $\mathbf{u}(t)$ . In other words, there is no interaction (product, etc.) between the input and the initial condition. This is only the case because the system is linear, and is so important that it is sometimes used in practice as a way of checking the linearity of a real-world system! For us, it's just a property to be aware of.

## 2.5 Impulse Response

Recall (or learn) that the delta (impulse) function is the “function” (or function-like thing) that satisfies

$$\delta(t) = \frac{d}{dt}H(t) = \begin{cases} 0 & \text{if } t \neq 0 \\ \infty & \text{if } t = 0 \end{cases}$$

$$\int_{-\epsilon}^{+\epsilon} \delta(t)dt = 1 \quad \text{for all } \epsilon > 0$$



where  $H(t)$  is the Heaviside step function ( $H(t) = 1$  for  $t \geq 0$  and  $H(t) = 0$  for  $t < 0$ ). The delta function plays a very special role in linear systems theory, which is essentially because of its so-called “sifting property”:

$$\int_{-\infty}^{\infty} \delta(\tau)x(\tau)d\tau = x(0) \quad \text{for (almost) any function } x(t)$$

In other words,  $\delta(t)$  sifts the value of  $x(t)$  at  $t = 0$ , and the rest of  $x(t)$  is irrelevant, even though we are taking the integral from  $-\infty$  to  $\infty$ . This property of course follows directly from the definition of the delta function above:

$$\delta(t)x(t) = \delta(t)x(0) = 0 \quad \text{for all } t \neq 0$$

and  $\delta(t)x(t) = \delta(t)x(0)$  trivially for  $t = 0$ . Therefore

$$\int_{-\infty}^{\infty} \delta(\tau)x(\tau)d\tau = \int_{-\infty}^{\infty} \delta(\tau)x(0)d\tau = x(0) \int_{-\infty}^{\infty} \delta(\tau)d\tau = x(0) \cdot 1 = x(0)$$

Why is this property so important? Because it immensely simplifies most integrals involving  $\delta(t)$ , for example, Eq. (2.7)! Here is how:

First, let’s limit our discussion to systems with only 1 input signal ( $m = 1$ ). If you have a system with more than 1 input, we can simply use the superposition property to add up the responses due to different inputs. Since we have 1 input, Eq. (2.7) simplifies to

$$\mathbf{x}(t) = e^{\mathbf{A}t}\mathbf{x}_0 + \int_0^t e^{\mathbf{A}(t-\tau)}\mathbf{b}u(\tau)d\tau \quad (2.9)$$

where  $u(\tau)$  is now scalar and  $\mathbf{b}$  is a column vector. Now, set

$$u(t) = \delta(t) \quad \mathbf{x}(0) = \mathbf{0}$$

which means that the system is initially at rest ( $\mathbf{x}(0) = \mathbf{0}$ ) and we are exciting it with an impulse (like a shock) at time 0, and then letting the system free flow (because  $u(t) = \delta(t) = 0$  for  $t > 0$ ). And here is where the sifting property of the impulse function comes handy:

$$\begin{aligned} \mathbf{x}(t) &= \int_0^t e^{\mathbf{A}(t-\tau)}\mathbf{b}\delta(\tau)d\tau \\ &= e^{\mathbf{A}(t-\tau)}\mathbf{b} \Big|_{\tau=0} \\ &= e^{\mathbf{A}t}\mathbf{b} \end{aligned}$$

Now that we have the state, we can substitute it in the output equation to get the impulse response of this system:

$$\mathbf{y}(t) = \mathbf{C}e^{\mathbf{A}t}\mathbf{b} + \mathbf{d}\delta(t)$$

where  $\mathbf{d}$  is now a vector (instead of the matrix  $\mathbf{D}$ ). For historical reasons and the importance of the impulse response, it is often given a special notation and shown by  $\mathbf{h}(t)$ :

$$\boxed{\mathbf{h}(t) = \mathbf{C}e^{\mathbf{A}t}\mathbf{b} + \mathbf{d}\delta(t)} \quad (2.10)$$

But the importance of the impulse response is much more than a mere simplification of the integral in Eq. (2.9). Recall (or learn) the definition of the convolution integral: for two signals (functions)  $g(t)$  and  $h(t)$ , their convolution is the new signal

$$\begin{aligned} (g \star h)(t) &= \int_0^t g(t-\tau)h(\tau)d\tau \\ &= \int_0^t g(\tau)h(t-\tau)d\tau \end{aligned}$$

Notice that the convolution is symmetric, so  $g \star h = h \star g$ . Now, assume that our system is still starting from rest ( $\mathbf{x}(0) = \mathbf{0}$ ), but the (still scalar) input  $u(t)$  is arbitrary. The output of the system is, from Eq. (2.8)

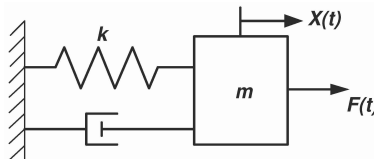
$$\begin{aligned} \mathbf{y}(t) &= \mathbf{C} \int_0^t e^{\mathbf{A}(t-\tau)}\mathbf{b}u(\tau) + \mathbf{d}u(t) \\ &= \int_0^t \mathbf{C}e^{\mathbf{A}(t-\tau)}\mathbf{b}u(\tau) + \mathbf{d}u(t) \\ &= \int_0^t \mathbf{C}e^{\mathbf{A}(t-\tau)}\mathbf{b}u(\tau) + \mathbf{d} \int_0^t \delta(\tau)u(t-\tau)d\tau \\ &= \int_0^t \mathbf{C}e^{\mathbf{A}(t-\tau)}\mathbf{b}u(\tau) + \int_0^t \mathbf{d}\delta(\tau)u(t-\tau)d\tau \\ &= \mathbf{C}e^{\mathbf{A}t}\mathbf{b} \star u(t) + \mathbf{d}\delta(t) \star u(t) \\ &= (\mathbf{C}e^{\mathbf{A}t}\mathbf{b} + \mathbf{d}\delta(t)) \star u(t) \end{aligned}$$

In other words, the response of an LTI system to any input  $u(t)$  is the convolution of that input with the system's impulse response:

$$\boxed{\mathbf{y}(t) = \mathbf{h}(t) \star u(t)} \quad (2.11)$$

In other words, the impulse response  $\mathbf{h}(t)$  contains all the information about the system ( $\mathbf{A}, \mathbf{b}, \mathbf{C}, \mathbf{d}$ )! Note that this is not the case for nonlinear systems.

**Example 2.5.1 (Impulse response of MSD system)** Consider the mass-spring-damper system



where the damper has a constant  $b$ . Newton's second law gives us

$$m\ddot{X} = -kX - b\dot{X} + F \quad (2.12)$$

which, using  $\mathbf{x} = \begin{bmatrix} X \\ \dot{X} \end{bmatrix}$  and  $u = F$  as we have seen before, and  $y = X$  as output, translates to the standard state space model

$$\begin{aligned} \dot{\mathbf{x}} &= \begin{bmatrix} 0 & 1 \\ -\frac{k}{m} & -\frac{b}{m} \end{bmatrix} \mathbf{x} + \begin{bmatrix} 0 \\ \frac{1}{m} \end{bmatrix} u \\ y &= [1 \ 0] \mathbf{x} + [0] u \end{aligned}$$

Having the  $\mathbf{A}$ ,  $\mathbf{b}$ ,  $\mathbf{C}$ ,  $\mathbf{d}$  matrices/vectors, we can readily find the impulse response from Eq. (2.10), and help from MATLAB:

```
1 syms b k m
2 A = [0 1; -k/m -b/m];
3 B = [0; 1/m];
4 C = [1 0];
5 h = simplify(C * expm(A) * B);
6 pretty(h)
```

The result is

$$h(t) = \frac{1}{\sqrt{\Delta}} \left( e^{-\frac{b-\sqrt{\Delta}}{2m}t} - e^{-\frac{b+\sqrt{\Delta}}{2m}t} \right), \quad \Delta = b^2 - 4km$$

Having the impulse response, we can simply convolve it with any new input and obtain the system's response!  $\square$

**Exercise 2.5.2 (Impulse response of MSD system)** In the above example, what happens to the impulse response if  $\Delta < 0$ ? Does it become complex valued, or does it remain real-valued? Why? Does your answer make intuitive sense?

Also, physically, how do you describe the difference between the  $b^2 - 4km > 0$  case vs. the  $b^2 - 4km < 0$  case? Does your description match the impulse response?  $\square$

**Exercise 2.5.3 (Impulse response via diagonalization)** Consider the same MSD system above and let  $b = 5, m = 0.5, k = 8$ . Use diagonalization to manually compute the system's impulse response.  $\square$