Supplimental Lecture

Review of last class

3 methods for representing dynamic systems:

1. Differential equations (partial + ordinary)
2. Transfer functions (limited to LTI)
3. State space ← what we will be focusing on in this class

Note: state space representations are not unique

State space for continuous linear time varying (C-LTV) systems

$$\dot{x}(t) = A(t)x(t) + B(t)u(t) \qquad x \in \mathbb{R}^n,\ u \in \mathbb{R}^m,\ y \in \mathbb{R}^p$$

$$y(t) = C(t)x(t) + D(t)u(t)$$

For continuous linear time invariant (C-LTI) systems

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

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

---

* Question @ intuition for Laplace transforms — jump back a bit to think @ what learned in studying SISO systems.

SISO system defined by a generic differential equation:

$$y^{(n)}(t) + \alpha_{n-1} y^{(n-1)}(t) + \dots + \alpha_1 \dot{y}(t) + \alpha_0 y(t) =$$

$$\beta_m u^{(m)}(t) + \dots + \beta_1 \dot{u}(t) + \beta_0 u(t)$$

<u>Main Insight</u>: Complex exponential functions are eigenfunctions for SISO LTI systems.

complex exponential functions: e^{st} where $s \in \mathbb{C}$ (s is a complex number)

$$s = \sigma + j\omega \qquad (\sigma, \omega \in \mathbb{R})$$

$$j^2 = -1$$

$$e^{st} = e^{(\sigma + j\omega)t} = e^{\sigma t} e^{j\omega t} = e^{\sigma t}\left[\cos\omega t + j\sin\omega t\right]$$

LaPlace Transform:

$$\mathcal{L}\{f(t)\} \triangleq \int_0^\infty f(t)e^{-st}\,dt = \hat{f}(s) \qquad s \in \mathbb{C}$$

(s = complex number)

Properties:

Linearity: $\mathcal{L}\{a_1 f_1(t) + a_2 f_2(t)\} = a_1 \hat{f}_1(s) + a_2 \hat{f}_2(s)$

Time Shift: $\mathcal{L}\{f(t-\tau)\} = e^{-s\tau}\hat{f}(s)$

differentiation: $\mathcal{L}\{\frac{d}{dt}f(t)\} = s\hat{f}(s) - f(0)$

$$\mathcal{L}\{\frac{d^2}{dt^2}f(t)\} = s^2\hat{f}(s) - s f(0) - \dot{f}(0)$$

fundamental transform:

$$\mathcal{L}\{e^{pt}\} = \frac{1}{s-p}$$

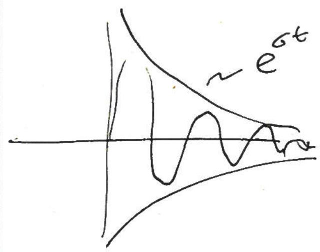
~~Inverse laplace~~ :

$$\cancel{\mathcal{L}^{-1}\{\hat{f}(s)\} = \frac{1}{2\pi j}\int}$$

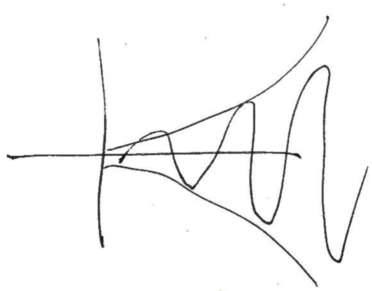
$e^{\sigma t}$ = exponential envelope

$[\cos\omega t + j\sin\omega t]$ = oscillations w/ in that envelope
@ freq $\omega = \mathrm{Im}\{s\}$

$\sigma < 0 \Rightarrow$ envelope decays to zero
$\Rightarrow$ amplitude decays to zero



$\sigma > 0 \Rightarrow$ envelope increases to ∞
$\Rightarrow$ Amplitude increases to ∞



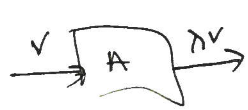
$\sigma = 0 \Rightarrow$ ~~envelope~~
$\Rightarrow$ Amplitude is constant in time.

~~eigen~~ Eigenfunction:

Eigenvector: Given an $n \times n$ matrix A, an eigenvector is an n dim vector v (not equal to zero) ~~that~~ that satisfies: $v \in \mathbb{R}^n$

$$Av = \lambda v \quad \text{for } \lambda \in \mathbb{C} \quad \text{(complex number)}$$

λ = eigenvalue
v = eigenvector



* If we sent in a vector v (multiply by A) what we get out is that same vector scaled by a complex number)

Eigenfunction: Is the equivalent w/ a dynamical system

$$e^{st} \longrightarrow \boxed{\text{LTI System}} \longrightarrow \lambda e^{st}$$

* we sent in an exponential function and what we get out is that same exponential function, but scaled by a complex number.

In this case the eigenvalue $\lambda(s) = \dfrac{\beta_m s^m + \beta_{m-1} s^{m-1} + \ldots + \beta_1 s + \beta_0}{s^n + \alpha_{n-1} s^{n-1} + \ldots + \alpha_1 s + \alpha_0}$

- Prove by substituting e^{st} into the original differential equation.

→ Another way of saying this is that solutions to a differential equation will be comprised of exponential functions and sin functions.

~~because when you differentiate something~~

→ A linear combination of solutions is also a valid solution so if we have multiple weighted inputs

$$u(t) = \sum_{k=1}^{N} U_k e^{s_k t}$$ = input is a weighted combination of exponential functions.

then

$$y(t) = \sum_{k=1}^{N} Y_k e^{s_k t}$$ = output is going to also be a weighted combination of those same exponential functions.

with the relationship $Y_k = \underbrace{\lambda(s_k)}_{\text{eigenvalue evaluated @ that frequency}} U_k$

→ No lets push this sum to being infinite

$$u(t) = \sum_{k=1}^{\infty} U_k e^{s_k t} \longrightarrow y(t) = \sum_{k=1}^{\infty} \lambda(s_k) U_k e^{s_k t}$$

* We know the e-valus as a function of s, and if we can find the weighted U_k values then we can immediately write down what the solution $y(t)$ will be.

→ ~~If we take the ex~~

– we can't do this for everything as a discrete sum, but we can for many systems if we take the extention of that as an infinite integral

$$u(t) = \frac{1}{2\pi j}\int_{\sigma_0 - j\omega}^{\sigma_0 + j\omega} U(s)\, e^{st}\, d\omega \qquad \sigma = \sigma_0 + j\omega$$

(line integral in the complex plane)

$$u(t) = \int U(s)\, e^{st}\, ds \quad \text{and} \quad y(t) = \int Y(s)\, e^{-st}\, ds$$

→ $y(t)$ is function of same complex exponentials w/ ~~same~~ different exponentials, but w/ relationship $Y(s) = \lambda(s) U(s)$

→ what we need to know is how to break down the input signal $u(t)$ into the eigenfunctions e^{st}. How is each eigenfunction weighted.

→ To do this we use LaPlace transform:

$$U(s) = \hat{u}(s) = \mathcal{L}\{u(t)\} = \int_0^\infty u(t)\, e^{-st}\, dt =$$

Amount of each complex exponential function e^{st} present in $u(t)$ for every possible value of s.

– ~~$U(s)$ is the complex amplitude of e^{st} present in the time signal $u(t)$~~

→ This generalizes: LaPlace breaks any function of time down to these component eigenfunctions e^{st} to see how much is present in the response in eigen space. then we can use the result to transform back.

Lets apply all this to our inverted pendulum problem:

$$m\ell^2\ddot{\theta} = mg\ell\theta - b\dot{\theta} + \tau$$

input (τ)

friction ($b\dot{\theta}$)

used small angle approximation ($mg\ell\theta$)

The Transfer function would be — (SISO example)

$$m\ell^2\ddot{\theta} - mg\ell\theta + k\ell\dot{\theta} = \tau$$

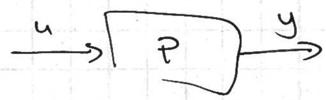
$$m\ell^2 s^2 \hat{\theta}(s) - mg\ell\,\hat{\theta}(s) + k\ell s\hat{\theta}(s) = \hat{\tau}(s)$$

(assume initial conditions are transient & decay quickly)

$$= m\ell^2\left(s^2 + \frac{k}{m\ell}s - \frac{g}{\ell}\right)\hat{\theta}(s) = \hat{\tau}(s)$$

$$\Rightarrow \hat{\theta}(s) = \left(\frac{1}{m\ell^2\left(s^2 + \frac{k}{m\ell}s - \frac{g}{\ell}\right)}\right)\hat{\tau}(s)$$

Block Diagrams —



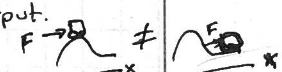
- we are familar w/ this representation using transfer functions

$$P : \hat{G}(s)$$

- In this class we focus on the state space representation

$$P: \dot{x} = Ax + Bu, \quad y = Cx + Du$$

Note: initial conditions of the state space will effect the output.



- Lets look @ composition rules for state space:

let $P_1 : \dot{x}_1 = A_1x_1 + B_1u_1, \quad y_1 = C_1x_1 + D_1u_1$ $\qquad x_1 \in \mathbb{R}^{n_1}, u_1 \in \mathbb{R}^{k_1}, y_1 \in \mathbb{R}^{m_1}$

$P_2 : \dot{x}_2 = A_2x_2 + B_2u_2, \quad y_2 = C_2x_2 + D_2u_2$ $\qquad x_2 \in \mathbb{R}^{n_2}, u_2 \in \mathbb{R}^{k_2}, y_2 \in \mathbb{R}^{m_2}$

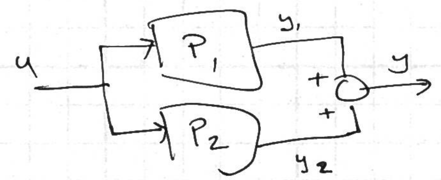
→ Note: systems can have different dimensions

Or in TF format:

$P_1 : \hat{G}_1(s)$

$P_2 : \hat{G}_2(s)$

Parallel Interconnection:



$u_1 = u_2$

$y = y_1 + y_2$

State Space:

$\dot{x}_1 = A_1 x_1 + B_1 u \,, \quad y_1 = C_1 x_1 + D_1 u$

$\dot{x}_2 = A_2 x_2 + B_2 u \,, \quad y_2 = C_2 x_2 + D_2 u$

$$y = y_1 + y_2 = C_1 x_1 + D_1 u + C_2 x_2 + D_2 u$$

$$= \begin{bmatrix} C_1 & C_2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} D_1 + D_2 \end{bmatrix} u$$

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} A_1 & \emptyset \\ \emptyset & A_2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} B_1 \\ B_2 \end{bmatrix} u$$

Transfer Function:

$$y = y_1 + y_2 = \hat{G}_1(s)\, u(s) + \hat{G}_2(s)\, u(s) = \left(\hat{G}_1(s) + \hat{G}_2(s)\right) u$$

Cascade Interconnection:

$$\xrightarrow{u} \boxed{P_1} \xrightarrow{z} \boxed{P_2} \longrightarrow y$$

State Space:

$$\dot{x}_1 = A_1 x_1 + B_1 u, \quad z = C_1 x_1 + D_1 u$$

$$\dot{x}_2 = A_2 x_2 + B_2 z, \quad y = C_2 x_2 + D_2 z$$

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} A_1 & 0 \\ ? & A_2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} B_1 \\ ? \end{bmatrix} u$$

$\longrightarrow$ plug in z to $\dot{x}_2$

$$\dot{x}_2 = A_2 x_2 + B_2 (C_1 x_1 + D_1 u)$$

$$= A_2 x_2 + \underline{B_2 C_1} x_1 + B_2 D_1 u$$

$$\Longrightarrow \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} A_1 & 0 \\ B_2 C_1 & A_2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} B_1 \\ B_2 D_1 \end{bmatrix} u$$

$$y = C_2 x_2 + D_2 (C_1 x_1 + D_1 u) = C_2 x_2 + D_2 C_1 x_1 + D_2 D_1 u$$

$$y = \begin{bmatrix} D_2 C_1 & C_2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + D_2 D_1 u$$

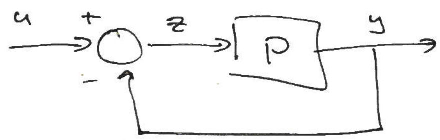
Transfer Function:

$$y = \hat{G}_2(s) z$$

$$z = \hat{G}_1(s) u$$

$$y = \hat{G}_2(s) \hat{G}_1(s) u$$

Feedback Interconnection:



State Space:

$$\dot{x}_1 = A_1 x_1 + B_1 z \qquad z = u - y$$

$$y_1 = C_1 x_1 + D_1 z$$

$$\dot{x}_1 = \cancel{A_1 x_1 + B(u-y)}\ A_1 x_1 + B(u-y)$$

$$y = C_1 x_1 + D_1 (u-y) \Rightarrow y + Dy = C_1 x_1 + D_1 u$$

$$(I+D) y = C_1 x_1 + D_1 u$$

$$y = (I+D_1)^{-1} C_1 x_1 + (I+D_1)^{-1} D_1 u$$

$$\dot{x}_1 = A_1 x_1 + B_1 (u-y) = A_1 x_1 + B_1 u - B_1 \left((I+D_1)^{-1} C_1 x_1 + (I+D_1)^{-1} D_1 u \right)$$

$$= \left(A_1 - B_1 (I+D_1)^{-1} C_1\right) x_1 + \left(B_1 - B_1 (I+D_1)^{-1} D_1\right) u$$

$(I + D_1) \longrightarrow$ needs to be well defined

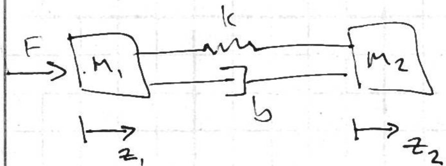
Transfer function:

$$\hat{y}(s) = \hat{G}_1(s)\left(\hat{u}(s) - \hat{y}(s)\right)$$

$$\left(I + \hat{G}_1(s)\right) \hat{y}(s) = \hat{G}_1(s)\, \hat{u}(s)$$

$$\hat{y}(s) = \left(I + \hat{G}_1(s)\right)^{-1} \hat{G}_1(s)\, \hat{u}(s)$$

Example: Spring-Mass-Damper



$$\left.\begin{aligned} m_1\ddot{z}_1 &= k(z_2 - z_1) + b(\dot{z}_2 - \dot{z}_1) + F \\ m_2\ddot{z}_2 &= k(z_1 - z_2) + b(\dot{z}_1 - \dot{z}_2) \end{aligned}\right\}$$ differential equation

- coupled 2nd order system

$x_1 = z_1$ $\qquad$ $y_1 = z_1$

$x_2 = \dot{z}_1$ $\qquad$ $y_2 = z_2$

$x_3 = z_2$ $\qquad$ $u = F$

$x_4 = \dot{z}_2$

$$\dot{x}_1 = \dot{z}_1 = x_2$$

$$\dot{x}_2 = \ddot{z}_1 = \frac{k}{m_1}(z_2 - z_1) + \frac{b}{m_1}(\dot{z}_2 - \dot{z}_1) + \frac{F}{m_1} = \frac{k}{m_1}(x_3 - x_1) + \frac{b}{m_1}(x_4 - x_2) + \frac{1}{m_1}u$$

$$\dot{x}_3 = \dot{z}_2 = x_4$$

$$\dot{x}_4 = \ddot{z}_2 = \frac{k}{m_2}(z_1 - z_2) + \frac{b}{m_2}(\dot{z}_1 - \dot{z}_2) = \frac{k}{m_2}(x_1 - x_3) + \frac{b}{m_2}(x_2 - x_4)$$

The state space model is:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -k/m_1 & -b/m_1 & k/m_1 & b/m_1 \\ 0 & 0 & 0 & 1 \\ k/m_2 & b/m_2 & -k/m_2 & -b/m_2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} 0 \\ 1/m_1 \\ 0 \\ 0 \end{bmatrix} u$$

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \end{bmatrix} u$$