Lecture 2 2025-08-28

Recap: guidance-navigation-controls vs. sense-plan-act

Today: propagating differential equations

$x \in \mathbb{R}^{n_x}$: state vector of dimension n_x
$u \in \mathbb{R}^{n_u}$: control vector of dimension n_u

Linear Systems

continuous & time-varying:

$$\dot{x}(t) = A(t)x(t) + B(t)u(t)$$

continuous & time-invariant:

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

discrete & time-varying:

$$x_{k+1} = A_k x_k + B_k u_k$$

discrete & time-invariant:

$$x_{k+1} = A x_k + B u_k$$

How to convert continuous LTI system to discrete?

$$\dot{x} = Ax + Bu \rightarrow \text{integrate over duration } \Delta t$$

last time: $\dot{x} \approx \dfrac{x_{k+1} - x_k}{\Delta t} = Ax_k + Bu_k$

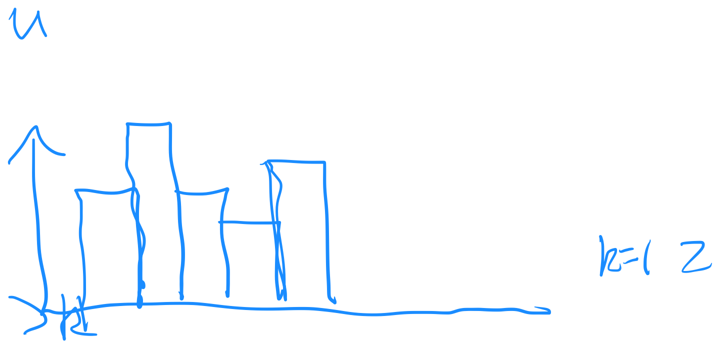
where we want Δt to be "small" relative to the time constant of the dynamics

$$\rightarrow \quad x_{k+1} - x_k = (\Delta t A) x_k + (\Delta t B) u_k$$

$$\rightarrow \quad x_{k+1} = (I + \Delta t A) x_k + (\Delta t B) u_k$$

where $I = \begin{pmatrix} 1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & 1 \end{pmatrix}$ is $n_x \times n_x$ identity

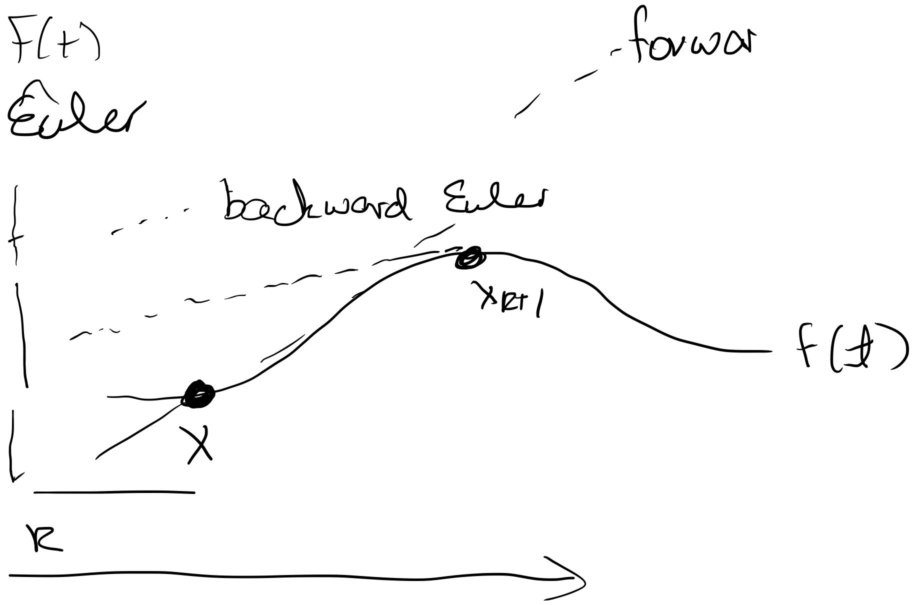
zero-order hold



Can we do better for LTI systems?

$$x_{k+1} = (I + \Delta t A) x_k + \Delta t B u_k$$

Forward Euler



alternatively, evaluate at $k+1$:

$$\dot{x} \approx \frac{x_{k+1} - x_k}{\Delta t} = A x_{k+1} + B u_{k+1}$$

$$\rightarrow (I - \Delta t A) x_{k+1} = x_k + \Delta t B u_{k+1}$$ Backward Euler's

Are there better integration schemes?

→ Trapezoidal integration

→ Hermite-Simpson

→ Runge-Kutta (e.g., RK4, RK8)

→ higher-order methods to model $x(t)$ and $u(t)$

e.g., pseudospectral methods

Analytical solutions via exponential matrix:

$\dot{x} = Ax + Bu$ $\xrightarrow{\text{solve scalar form}}$ $\dot{x} = ax + bu$

$x \in \mathbb{R}^{n_x}, u \in \mathbb{R}^{n_u}$ $\quad$ $x \in \mathbb{R}, u \in \mathbb{R}$

$\dot{x} = ax$ $\leftarrow$ how to solve this?

solve with characteristic equation:

$\dot{x} = ax \rightarrow s - a = 0 \rightarrow x(t) = x(t_0)e^{at}$

now add back in $bu(t)$:

$\dot{x}(t) = ax(t) + bu(t) \rightarrow \dot{x} - ax = bu$

multiply by integrating factor e^{-at}:

$$e^{-at}[\dot{x} - ax] = e^{-at}bu$$

$$= e^{-at}\dot{x} - a e^{-at} x$$

$$\frac{d}{dt}[ab] = \dot{a}b + a\dot{b} \rightarrow \frac{d}{dt}\left[e^{-at}x(t)\right]$$

→ integrate both sides from T_0 to T_f:

$$\int_{T_0}^{T_f} \frac{d}{dt}\left[e^{-at}x(t)\right] dt = \int_{T_0}^{T_f} e^{-at} bu(t)\, dt$$

$$\rightarrow e^{-at}x(t)\Big]_{t=T_0}^{t=T_f} = \int_{T_0}^{T_f} e^{-at} bu(t)\, dt$$

$$\rightarrow e^{-aT_f}x(T_f) - e^{-aT_0}x(T_0) = \int_{T_0}^{T_f} e^{-at} bu(t)\, dt$$

shuffle around terms & multiply by e^{aT_f}:

$$\rightarrow e^{-aT_f}x(T_f) = e^{-aT_0}x(T_0) + \int_{T_0}^{T_f} e^{-at} bu(t)\, dt$$

$$\rightarrow \quad x(T_f) = e^{-a(T_f - T_0)} x(T_0) + \int_{T_0}^{T_f} e^{-at} \, b u(t) \, dt$$

Can we apply this approach to state space eqn?

$$\dot{x}(t) = A x(t) + B u(t)$$

where $x \in \mathbb{R}^{n_x}$, $u \in \mathbb{R}^{n_u}$, $A \in \mathbb{R}^{n_x \times n_x}$, $B \in \mathbb{R}^{n_x \times n_u}$

recall our approach with scalar version was:

1. assume solution takes exponential form
2. multiply by integrating factor e^{-at}

Exponential matrix: given $A \in \mathbb{R}^{n \times n}$

$$e^{At} = \sum_{k=0}^{\infty} \frac{(At)^k}{k!}$$

$$= \frac{(At)^0}{0!} + \frac{(At)^1}{1!} + \frac{(At)^2}{2!} + \frac{(At)^3}{3!} + \dots$$

$$= I + At + \frac{A^2t^2}{2!} + \frac{A^3t^3}{3!} + \dots$$

$$\frac{d}{dt}\left[e^{At}\right] = \frac{d}{dt}\left[I + tA + \frac{t^2A^2}{2!} + \frac{t^3A^3}{3!} + \dots\right]$$

$$= A + tA^2 + \frac{t^2A^3}{2!} + \dots$$

$$= A\underbrace{\left[I + tA + \frac{t^2A^2}{2!} + \dots\right]}_{e^{tA}}$$

$$= Ae^{tA}$$

$$\rightarrow \frac{d}{dt}\left[e^{At}\right] = Ae^{At} = e^{At}A$$

Comparisons between e^{at} vs. e^{At}:

1. $e^{A(t_1+t_2)} = e^{At_1} e^{At_2}$

2. $e^{A0} = I$

3. $e^{A} e^{B} \neq e^{A+B}$ (only in certain cases)

Now let's apply e^{tA} to the LTI system:

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

$$\rightarrow \dot{x} - Ax = Bu$$ left multiply by e^{-At}

$$\rightarrow e^{-At}[\dot{x} - Ax] = e^{-At} Bu$$

$$= e^{-At}\dot{x} - e^{-At}Ax$$

$$= \frac{d}{dt}\left[e^{-At}x(t)\right]$$

Integrate both sides between T_0 to T_f

$$\rightarrow \int_{T_0}^{T_f} \frac{d}{dt}\left[e^{-At} x(t)\right] dt = \int_{T_0}^{T_f} e^{-At} Bu(t)\, dt$$

$$= \left. e^{-At} x(t) \right]_{t=T_0}^{t=T_f}$$

$$= e^{-AT_f} x(T_f) - e^{-AT_0} x(T_0)$$

$$\rightarrow e^{-AT_f} x(T_f) = e^{-AT_0} x(T_0) + \int_{T_0}^{T_f} e^{-At} Bu\, dt$$

$$\rightarrow x(T_f) = e^{A(T_f - T_0)} x(T_0) + \int_{T_0}^{T_f} e^{A(T_f - t)} Bu\, dt$$

Let's change the variables around to get this into a nicer form:

integrate over $d\tau$, let $T_0 = 0$ and $T_f = t$

$$\rightarrow \quad x(t) = \underbrace{e^{At}}_{\Phi(t,0)} x(0) + \int_0^t \underbrace{e^{A(t-\tau)}}_{\Phi(t,\tau)} B u(\tau)\, d\tau$$

$\Phi(t,0)$ & $\Phi(t,\tau)$ are called state transition matrices

---

example: a common LTI system is the double integrator:

$$\dot{x} = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} x + \begin{pmatrix} 0 \\ 1 \end{pmatrix} u$$

say we have a robot arm with n_q joints and $q \in \mathbb{R}^{n_q}$ is the vector of joint angles

& $\dot{q} \in \mathbb{R}^{n_q}$ is the vector of joint velocities

modeling the joints as a double integrator would look like:

$$\begin{pmatrix} \dot{q} \\ \ddot{q} \end{pmatrix} = \begin{pmatrix} 0^{n_q \times n_q} & I^{n_q \times n_q} \\ 0^{n_q \times n_q} & 0^{n_q \times n_q} \end{pmatrix} \begin{pmatrix} q \\ \dot{q} \end{pmatrix} + \begin{pmatrix} 0^{n_q \times n_u} \\ I^{n_q \times n_u} \end{pmatrix} u$$

where $u \in \mathbb{R}^{n_u}$ is the force applied to each joint,

i. e. $n_u = n_q \rightarrow u \in \mathbb{R}^{n_q}$

How can e^{tA} be computed if it's an infinite expansion?

Cayley-Hamilton Thm: every square matrix $A \in \mathbb{R}^{n \times n}$

satisfies its own characteristic equation where the characteristic equation is

$$p_A(\lambda) = \det|\lambda I - A| = 0$$

e. $A = \begin{pmatrix} 3 & 4 \\ 5 & 8 \end{pmatrix} \rightarrow \det\begin{bmatrix} \lambda-3 & -4 \\ -5 & \lambda-8 \end{bmatrix} =$

$$\rightarrow (\lambda-3)(\lambda-8)-(-4)(-5) = \lambda^2 - 11\lambda + 24 - 20$$
$$= \lambda^2 - 11\lambda + 4 = 0$$

CH says $\rightarrow A^2 - 11A + 4 = 0 \Rightarrow A^2 = 11A - 4$

$$\rightarrow A^3 = A^2 A = (11A-4)A = 11A^2 - 4A$$
$$= 121A - 44 - 4A = 117A - 44$$

$A^4 = A^3 A$ & this procedure continues.

→ takeaway: can compute e^{tA} only with A^k!

<u>Linearity</u>

A function $f: \mathbb{R}^n \to \mathbb{R}^m$ is linear if:

1. $f(x+y) = f(x) + f(y) \ \forall x, y \in \mathbb{R}^n$

2. $f(ax) = a f(x) \ \forall x \in \mathbb{R}^n, a \in \mathbb{R}$

e.g., $y = Ax$ where $A \in \mathbb{R}^{m \times n}$

pf: 1. $A(x_1 + x_2) = Ax_1 + Ax_2 = y_1 + y_2$ ✓

2. $A(ax_1) = aAx_1 = ay_1$ ✓

e.g., $y = Ax + b$ where $A \in \mathbb{R}^{m\times n}$ & $b \in \mathbb{R}^m$

change of variables $\rightarrow$ $\underbrace{\begin{bmatrix} y \\ 1 \end{bmatrix}}_{:= \bar{y}} = \underbrace{\begin{bmatrix} A & b \\ 0 & 1 \end{bmatrix}}_{:= \bar{A}} \underbrace{\begin{bmatrix} x \\ 1 \end{bmatrix}}_{:= \bar{x}}$

$\rightarrow \quad \bar{y} = \bar{A}\bar{x}$

pf:

1. $\bar{A}(\bar{x}_1 + \bar{x}_2) = \begin{pmatrix} A & b \\ 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 + x_2 \\ 1 + 1 \end{pmatrix}$

$= \begin{pmatrix} Ax_1 + b \\ +1 \end{pmatrix} + \begin{matrix} Ax_2 + b \\ 1 \end{matrix} = \begin{bmatrix} Ax_1 + b \\ 1 \end{bmatrix} + \begin{bmatrix} Ax_2 + b \end{bmatrix}$

$$= \begin{bmatrix} A & b \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ 1 \end{bmatrix} + \begin{bmatrix} A & b \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_2 \\ 1 \end{bmatrix}$$

$$= \bar{A}\,\bar{y}_1 + \bar{A}\,\bar{y}_2$$

2. $$\bar{A}(a\bar{x}_1) = \begin{bmatrix} A & b \\ 0 & 1 \end{bmatrix} \begin{bmatrix} a x_1 \\ a \end{bmatrix} = \begin{bmatrix} a A x_1 + ab \\ a \end{bmatrix}$$

$$= a \begin{bmatrix} A x_1 + b \\ 1 \end{bmatrix} = a\,\bar{y}_1$$

Quick comment on complexity:

If $A \in \mathbb{R}^{m \times n}$, what is the complexity of Ax?

pseudo-code:

for $i = 1, \dots, m$: # iterate over m rows

$y_i = 0$

for $j = 1, \dots, n$ # iterate over n columns

$y_i \mathrel{+}= a_{ij} \cdot x_j$

→ matrix-vector multiplication is complexity $\mathcal{O}(m \cdot n)$