LOST 1109

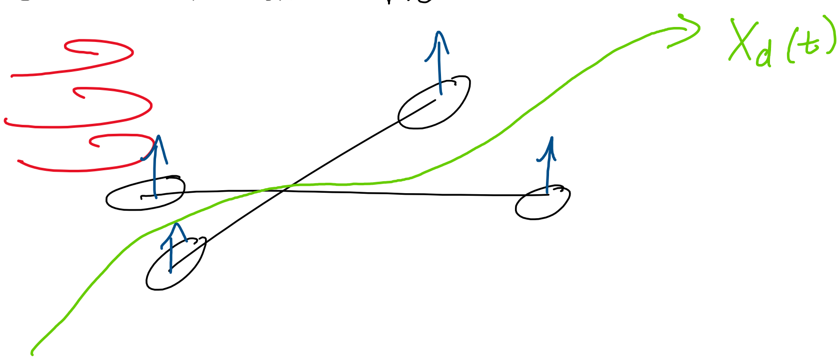
Wednesday, November 9, 2022 5:08 PM

Today's Agenda:

1. What is controls?
2. Review of ODEs
3. Feedback Control Design
4. Tuning Controllers

What is Control?

- Start w/ an example: Drone!



Pretty complex System!

- How can we get our quadrotor to PRECISELY follow $X_d(t)$

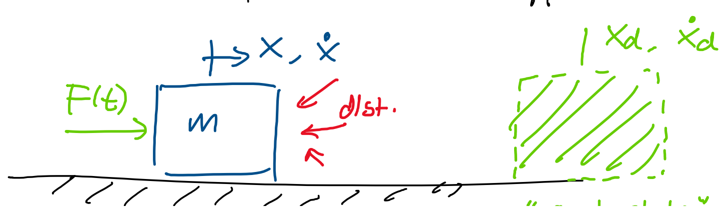
- Problem: UNKNOWN DISTURBANCES (wind!)
  - Can we still track $X_d(t)$??

Overall Q: Find an input s.t. we can TRACK a desired traj. $X_d(t)$ in the presence of DISTURBANCES!!

Pretty Hard Problem!

Where to begin?

- A mass m w/ a force F(t) applied!



"goal state"

Control Question: What is the force $F(t)$ that allows us to do this??

- Control "INPUT": force F on the block! We can change it to ANYTHING we want!

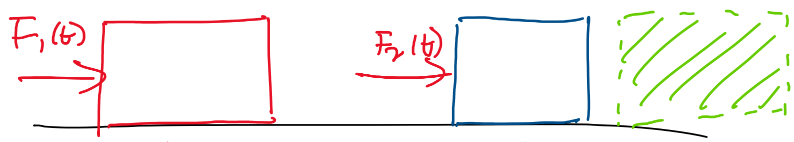
What do we NEED to get BEST possible $F(t)$??

1. System Dynamics: tells us How our block moves!

$$M\ddot{q} + C\dot{q} + N = T$$

$\Rightarrow$ tells us How block responds to force!

2. SENSOR INFORMATION (FEEDBACK)
   - read sensor info & use it to decide our input!



state 1
High force, far away from goal!

state 2
low force, CLOSE to goal!

"force input depends on ERROR"

Error: $e = x_d - x$

- How does error change in time?

$$\dot{e} = \dot{x}_d - \dot{x}$$

GEN:
$\Rightarrow$ Dynamics Model
$\Rightarrow$ SOME MEASURE OF ERROR!

☆ GEN IDEA IN CONTROLS: DRIVE OUR ERROR TO ZERO using a good choice of F. ☆

- Thermostat $\rightarrow$ uses PID!

What should F look like??

"external / applied" forces!

$M\ddot{q} + C\dot{q} + N = \tau$ Have COMPLETE control!

- How can we get a MATH. guarantee of behavior??
- is an ODE!
  - Apply theory of ODEs to prove system behavior!

Quick Review: Second Order ODEs:

- Why 2nd order?? $F = ma$
  $F = m\dot{v}$
  $\boxed{F = m\ddot{x}}$ } ALL dynamics eqns. are SECOND ORDER!

How can we solve a 2nd order ODE??
- linear, homogeneous 2nd order ODE!

$$a\ddot{x} + b\dot{x} + cx = 0$$

$X \Rightarrow$ Single Scalar Function!

Constants!

- What does it mean to solve the ODE?
  - Search for $X(t)$ that SATISFIES the constraints!
  - What we want: the func + its derivs (scaled up) to ALL equal zero!!

$$a\ddot{x} + b\dot{x} + cx = 0$$

Q: What is a func. that "looks similar" to its deriv??

$$X(t) = e^{\lambda t} \qquad \text{ex: } e^{2t},\ e^{-5t} \ldots$$

What should λ be if this is a soln??

$$\frac{d}{dt}(x(t)) = \frac{d}{dt}(e^{\lambda t}) = \lambda e^{\lambda t}$$

$$\frac{d^2}{dt^2}(x(t)) = \lambda^2 e^{\lambda t}$$

$$a\lambda^2 e^{\lambda t} + b\lambda e^{\lambda t} + ce^{\lambda t} = 0$$

$$(a\lambda^2 + b\lambda + c)\cancel{e^{\lambda t}} = 0$$

$$\boxed{a\lambda^2 + b\lambda + c = 0}$$

⇒ If we can SOLVE for λ, then we'll have a soln!

⇒ Two (potentially repeated) solutions, λ_1, λ_2. If $\lambda_1 \neq \lambda_2$

$$\boxed{X(t) = c_1 e^{\lambda_1 t} + c_2 e^{\lambda_2 t}}$$

constants! (from INIT. condition)

ex: $X(0) = 5$, $\dot{X}(0) = 10$

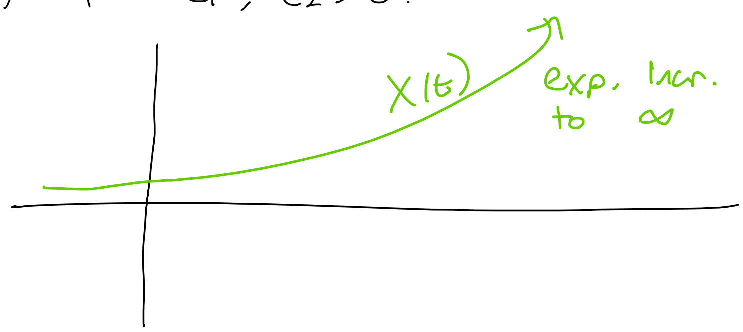
If λ_1, λ_2 COMPLEX, soln. will be ~ diff.

for $\lambda_1 = \lambda_2$

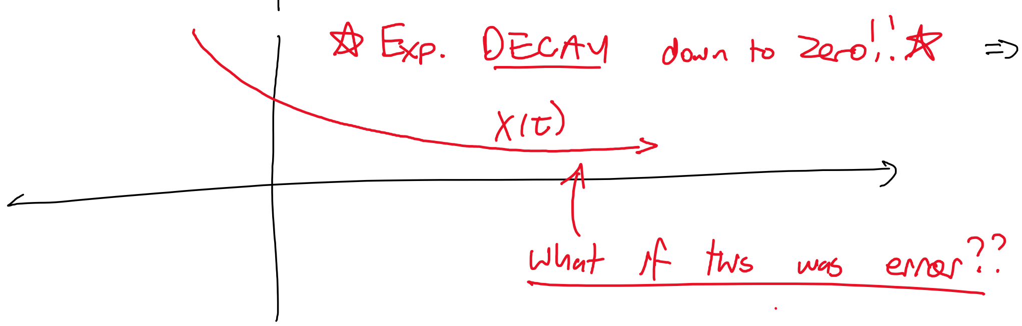
$c_1 e^{\lambda_1 t} + c_2 t e^{\lambda_1 t}$

IF you have REPEAT roots!

For case where λ_1, λ_2 are diff. & real, & $c_1, c_2 > 0$.



⇒ If $\lambda_1, \lambda_2 > 0$



⇒ If $\lambda_1, \lambda_2 < 0$

Pos. def: Property of a matrix!

- ⇒ All eigenvals. are > 0
- ⇒ It is symmetric!

– ALWAYS invertible!

$X^T M X > 0$

⇒ for all nonzero X

Here's an idea!

exp. decay is "ROBUST TO NOISE"

$$a\ddot{x} + b\dot{x} + cx = \delta$$

very small random amount!

- Soln. will STILL look exp!
  - will still asymp. DECAY!!

Idea: This convergence would be great for us to have in our controllers!!

⇒ If we can get our ERROR to follow one of these ODEs, we're pretty much reached our goal!

- Error can → 0 even w/ some noise!

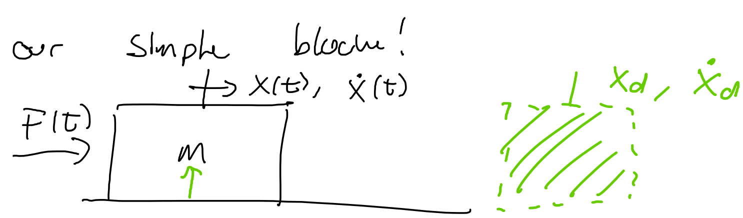
Q: How can we find an INPUT to our system to make error → 0 ??

GOAL EQN: make our error follow:

$$a\ddot{e} + b\dot{e} + ce = 0$$

} by adjusting a, b, c, error → 0.

let's look at our simple block!



Dynamics:

$$F = ma$$

$$\boxed{F(t) = m\ddot{x}(t)}$$

Scalar eqn. in x dir! Our GOVERNING eqn!

- let's see if we can find $f(t)$ s.t. we get

$$a\ddot{e} + b\dot{e} + ce = 0$$

Must-have!

First step: move all to one side!

$$\boxed{0 = F - m\ddot{x}}$$

} Pick ANYTHING for F!!

What do we need in F??

$F - m\ddot{x}$

⇒ All that's "missing" must incl. in F.

$$a\ddot{e} = m(\ddot{x}_d - \ddot{x})$$

$b\dot{e} \to k_d\dot{e}$

Const!

$ce \to k_p e$

↑const!

$$\boxed{F = m\ddot{x}_d + k_d\dot{e} + k_p e}$$

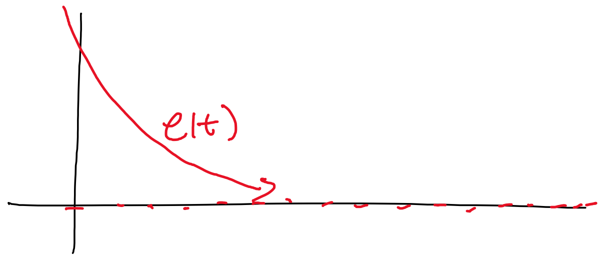
Plug in to our dynamics eqn. & MAKE SURE we get what we want!

$0 = F - m\ddot{x}$

$0 = m\ddot{x}_d + k_d\dot{e} + k_p e - m\ddot{x}$ $\qquad \ddot{e} = \ddot{x}_d - \ddot{x}$

$0 = m(\ddot{x}_d - \ddot{x}) + k_d\dot{e} + k_p e$

☆ $\boxed{0 = m\ddot{e} + k_d\dot{e} + k_p e}$ ☆ k_d & k_p: "TUNE" to get $\lambda_1, \lambda_2 < 0$



using F=ma, JUST SHOWED that this F can drive $e(t) \to 0$.

$0 = m\ddot{e} + k_d\dot{e} + k_p e$

$0 = m\lambda^2 + k_d\lambda + k_p$ Solve for λ.

let's examine the terms in our $F(t)$: (our controller)

$F = m\ddot{x}_d + k_p e + k_d\dot{e}$

"proportional term": k_p = Prop. GAIN

Deriv. term

☆ k_d = deriv gain ☆

$F = ma$

$F = m\ddot{x}$ ← ACTUAL accel!

- Tune it to get on e(t).

$F = m\ddot{x}_d + k_p e(t) + k_d\dot{e}(t)$

FEEDFORWARD! (doesn't use our ACTUAL state)

⇒ What we're using to get $\ddot{e}$!!

FEEDBACK TERM! Measures Current State, ADJUSTS F acc. to that!

"PD CONTROLLER"

- HOW DO WE GENERALIZE??

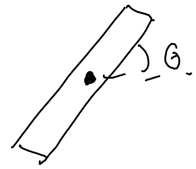
- PD control worked great before, can we ALWAYS use it??

Consider GENERAL dyn. eqn:

$\boxed{M\ddot{q} + C\dot{q} + N = T}$ $\qquad q = \begin{bmatrix} x \\ y \\ z \\ \vdots \end{bmatrix}$ → VECTOR!!

⇒ Q: How can we make $\boxed{q \rightarrow q_d ??}$

$$q_d = \begin{bmatrix} x_d \\ y_d \\ z_d \\ \theta_d \end{bmatrix}$$



- Have a CONTROL INPUT τ to our sys!
  - have COMPLETE CONTROL to set $\underline{\tau = \text{ANYTHING}}$!!

- Idea: let's use our SAME concept as before!

$$\left.\begin{aligned} e &= q_d - q \\ \dot{e} &= \dot{q}_d - \dot{q} \\ \ddot{e} &= \ddot{q}_d - \ddot{q} \end{aligned}\right\} \text{ALL VECTORS!} \quad \ddot{e} = \begin{bmatrix} \ddot{e}_1(t) \\ \ddot{e}_2(t) \\ \vdots \\ \ddot{e}_n(t) \end{bmatrix} \in \mathbb{R}^n$$

Want to find an τ:
1. CANCEL OUT UNWANTED TERMS
2. Produce an <u>error</u> diff. eqn:

$$\ddot{e} + K_d \dot{e} + K_p e = 0 \quad \text{\} MATRIX DIFF. EQN!}$$

Are MATRICES!

$$K_p = \begin{bmatrix} K_{p1} & K_{p2} & \cdots & \\ \vdots & \ddots & & \\ \vdots & & \ddots & \\ \vdots & & & K_{pn} \end{bmatrix}$$

⇒ we have COMPLETE CONTROL over all K_p gains!

$$\boxed{M\ddot{q} + C\dot{q} + N = \tau} \qquad F = m\ddot{x}$$

$$☆ \quad \ddot{e} + K_d\dot{e} + K_p e = 0 \quad ☆$$

let's bring $M\ddot{q}$ to the RHS:

Cancel! $C\dot{q} + N = \tau - M\ddot{q}$

- look at WHAT we <u>must "cancel", what we must</u> <u>ADD</u>!

$$\star \quad \boxed{\tau = \underbrace{C\dot{q} + N}_{} + \underbrace{M(\ddot{q}_d}_{} + \underbrace{K_d \dot{e} + K_p e}_{})} \quad \star$$

Cancel terms that DON'T exist in our error eqn.

gives me $\ddot{e}$

"PD TERMS"

let's plug it in!

$$\star \quad C\dot{q} + N = \tau - M\ddot{q} \quad \star$$

$$\cancel{C\dot{q}} + \cancel{N} = \cancel{C\dot{q}} + \cancel{N} + M(\ddot{q}_d + K_d \dot{e} + K_p e) - M\ddot{q}$$

$$0 = M(\ddot{q}_d - \ddot{q} + K_d \dot{e} + K_p e)$$

$$M^{-1} 0 = M^{-1} M (\ddot{e} + K_d \dot{e} + K_p e)$$

ALWAYS invertible! (Pos. def. matrices!)

$$\boxed{0 = \ddot{e} + K_d \dot{e} + K_p e}$$

perfect 2nd order error dynamics!!

⇒ How do vecs impact us??

$$K_d = \begin{bmatrix} k_d & & 0 \\ & k_d & \\ & & \ddots & \\ 0 & & & k_d \end{bmatrix} \qquad K_p = \begin{bmatrix} k_p & & 0 \\ & k_p & \\ & & \ddots & \\ 0 & & & k_p \end{bmatrix}$$

look @ an ARB. ROW!! Row "i"

$$\boxed{0 = \ddot{e}_i + k_d \dot{e}_i + k_p e_i}$$

SCALAR 2nd order ODE!

⇒ Can choose k_p, k_d s.t. $e_i \to 0$.

Car: "Bicycle model"

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} V\cos(\theta) & 0 \\ V\sin(\theta) & 0 \\ 0 & \cdots \end{bmatrix}$$

Depends NL. on ϕ.

$\sin(\varphi) \quad \cos(\varphi)$

$$\begin{bmatrix} \dot{e} \\ \ddot{e} \end{bmatrix} = \overbrace{\begin{bmatrix} 0 & I \\ -k_p & -k_D \end{bmatrix}}^{A} \begin{bmatrix} e \\ \dot{e} \end{bmatrix}$$

If the eigenval's have real parts < 0 then $e \to 0$.

$$e(t) = \underline{\exp(At)}\, e(0) \qquad \begin{bmatrix} e^{\lambda_1 t} & & \\ & e^{\lambda_2 t} & \\ & & \ddots \end{bmatrix}$$

(Not in Scope) "Jordan normal form"

Review: Our Controller:

$$\star \quad \underline{T = C\dot{q} + N + M(\ddot{q}_d + k_p e + k_d \dot{e})} \quad \star$$

PD term!!

Remember: when we plug T into our dynamics:

$$\underline{0 = \ddot{e} + k_d \dot{e} + k_p e}$$

tune k_p, k_d s.t. $e \to 0$.

$\Rightarrow$ If applied to a robot arm, called a "COMPUTED TORQUE" controller!

$$f_1 = \cos\theta + I\ddot{\theta} \leftarrow \ddot{e} + k_p\dot{e} + k_d e$$
$$f_2 = \sin\theta$$

$$f_1 = \cos\theta + I(\ddot{\theta}_d + k_p e + k_d \dot{e})$$

TUNING YOUR CONTROLLERS:

What do k_p, k_d actually do??

$$\ddot{e} + k_d \dot{e} + k_p e = 0$$

$$\lambda^2 + k_d \lambda + k_p = 0$$

$\Rightarrow \lambda$ COULD be complex!!

$$\star \quad \lambda_{1,2} = a \pm bi \quad \star$$

☆ $\boxed{e(t) = c_1 e^{at} \sin(bt) + c_2 e^{at} \cos(bt)}$ ☆

⇒ REAL PART of eigenval → gives us DECAY

⇒ COMPLEX PART → Gives us OSCILLATIONS!

Why need BOTH k_p & k_d??

- let's set $k_d = 0$, assume $k_p > 0$

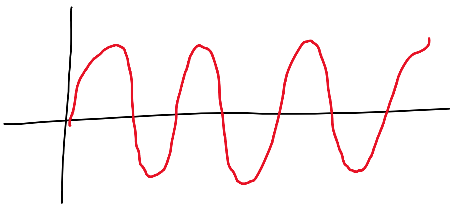
$$\ddot{e} + \cancel{k_d \dot{e}} + k_p e = 0$$

$$\lambda^2 + k_p = 0$$

$$\lambda^2 = -k_p$$

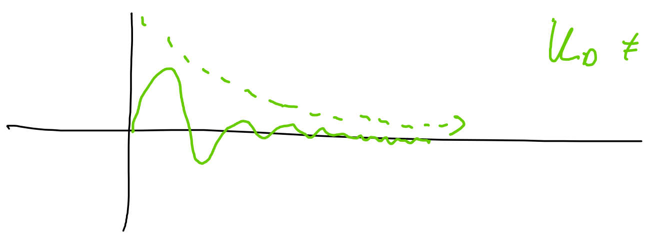
$$\lambda = \pm \sqrt{k_p}\, i$$

} NO REAL PART!!



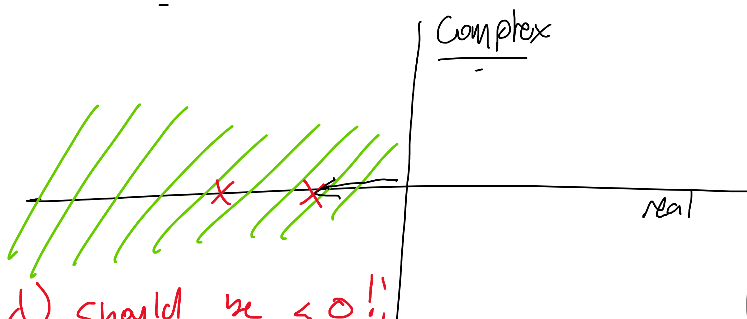
NO DECAY!!

- Introducing k_D: gives us a REAL PART that allows for exp. DECAY!



$k_D \neq 0$ } gives us "DAMPING"

How to Choose k_p & k_d??



$re(\lambda)$ should be < 0!!

Soln. to our char. eqn!:

$$\lambda^2 + k_d \lambda + k_p = 0$$

CHOOSE $\lambda_1 = -3$, $\lambda_2 = -4$

$$(\lambda + 3)(\lambda + 4) = 0$$

$$\lambda^2 + 7\lambda + 12 = 0$$

$$\boxed{k_d = 7 \qquad k_p = 12}$$

$$e(t) = c_1 e^{-3t} + c_2 e^{-4t}$$

settling time ≐ 2% of your goal state.