Intro to Robotics

Part(3)

(Joint Motion Control)

Joint Motion Control

Q: What is the goal of robot control ?

- find the time history torques to be sent to actuators in order to execute a task.
    - Regulation (constant reference)
    - Tracking (time-varying reference)

Challenges:

- External Disturbance
- Unmodeled dynamics & uncertain robot parameters
- Discretization

Open-loop Control

- The output of trajectory generator $\theta^d, \dot{\theta}^d, \ddot{\theta}^d$ is passed to the robot dynamic model:

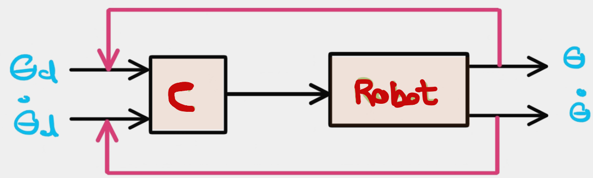
$$\tau = \underbrace{M(\theta^d)}_{\text{Inetria Matrix}} \underset{\text{acce}}{\ddot{\theta}^d} + \underbrace{C(\theta^d, \dot{\theta}^d)}_{\text{Coriolis \& Centerfuigal forces}} + \underbrace{G(\theta^d)}_{\text{Gravational Forces}}$$

Problems

- Disturbances
- Model inaccuracies

→ Tracking Errors

Feedback Control



Inputs:

- Position/velocity of the joints : $\theta(t)$, $\dot{\theta}(t)$ [Measurements]
- Reference trajectory: $\theta_d(t)$, $\dot{\theta}_d(t)$

Control can be :

- Decentralized control (mostly used) [Independent Joint Control]
- Centralized control (accounts for dynamic interaction between joints)

Controller Specifications

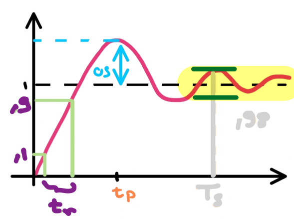
Frequency Domain

- closed loop Bw

[Highest frequency sinusoid it can track]

Time Domain

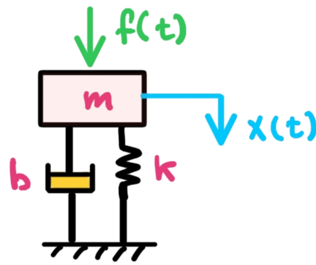
- rise time (T_r)
- settling time (T_s)
- Overshoot (os)%



Design spec. → Standard Form → Poles → Gains (k)

T_r, T_s, Os → ζ, ω_n → P_1, P_2 → MATLAB: place, acker

Review on Spring-Mass-Damper



$$m\ddot{x} = f(t) - kx - b\dot{x}$$

$$\boxed{m\ddot{x} + b\dot{x} + kx = f(t)}$$

(2nd order linear system)

I) Study of transient (natural dynamics) : f(t) = 0

$$\boxed{x(t) = C_1 e^{\lambda_1 t} + C_2 e^{\lambda_2 t}}$$

Where λ_1, λ_2 are the roots of the characteristics equation:

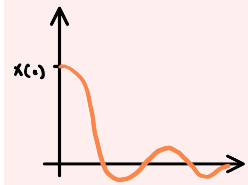
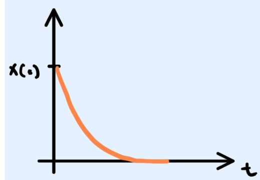
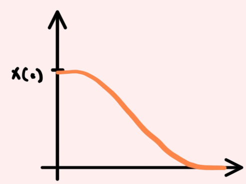
$$m\lambda^2 + b\lambda + k = 0$$

$$\boxed{\lambda_{1,2} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2m}}$$

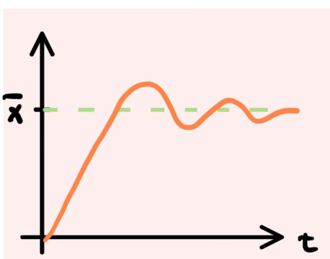
System is always stable if the poles have negative real part

- Let's take a look at different cases for system response.

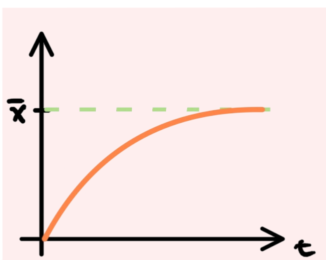
I) Study of transient (natural dynamics) : $f(t) = 0$

Case	When?	Solution	Response
Under Damped	$D - 4mk < 0$ $D < 2\sqrt{mk}$	Complex conjugate pair of roots	
Critically Damped	$D - 4mk = 0$ $D = 2\sqrt{mk}$	2 real roots $\lambda_1 = \lambda_2$	
Over Damped	$D - 4mk > 0$ $D > 2\sqrt{mk}$	2 real roots $\lambda_1 \neq \lambda_2$	

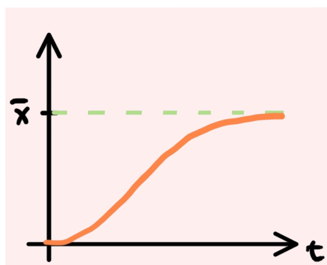
II) case when : $f(t) \neq 0 = cost$, $x(0) = 0$



Under Damped



Critically Damped



Over Damped

Notes on Under-damped 2nd Order Systems

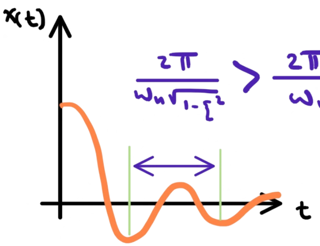
$$\ddot{x} + \frac{b}{m}\dot{x} + \frac{k}{m}x = 0 \Rightarrow \ddot{x} + 2\zeta\omega_n\dot{x} + \omega_n^2 x = 0$$

$\omega_n = \sqrt{\frac{k}{m}}$ [Natural Frequency]

$\frac{b}{m} = 2\zeta\omega_n \Rightarrow \zeta = \frac{b}{2\cdot\omega_n\cdot m}$ [Natural Damping Frequency]

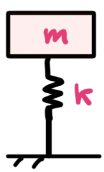
When $\zeta = 1$, $b = 2\sqrt{km}$ [Critically Damped]

Time Response for $\zeta < 1$ [Under-Damped]

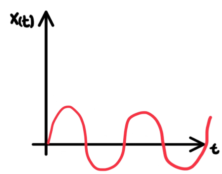


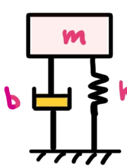
$$\omega_d = \omega_n\sqrt{1-\zeta^2}$$

$$\omega_d < \omega_n$$

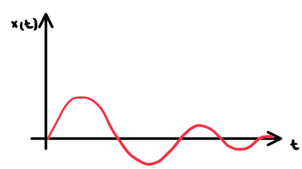


The system oscillates bouncing energy between **mass**(kinetic) and **spring**(potential)





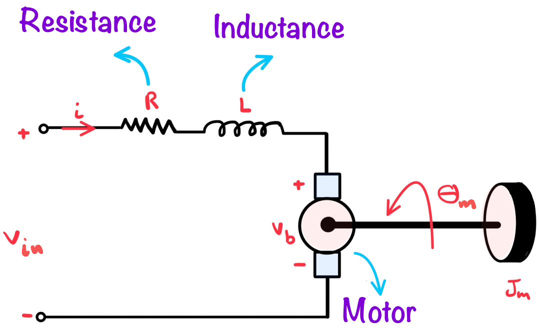
Unless there is damping element



Notes:

- From **dynamics** we know that robot is a 2nd order nonlinear system.
- we can apply linear control to the robot if:
  - Gear ratio (N) > 100
  - Robot operate at low speeds
- Under these assumptions we can
  - decouple the dynamics of each joint
  - Consider coupling as disturbance
- To see why this works, we will take a look at motor dynamics

Motor Modelling



Electrical Domain :

Lenz-Law: $V_b = K_b \dot{\theta}_m$ "back emf" (as motor spins it creates) some voltage

Loop-Equation: $V_{in} = i \cdot R + \frac{di}{dt} \cdot L + V_b$

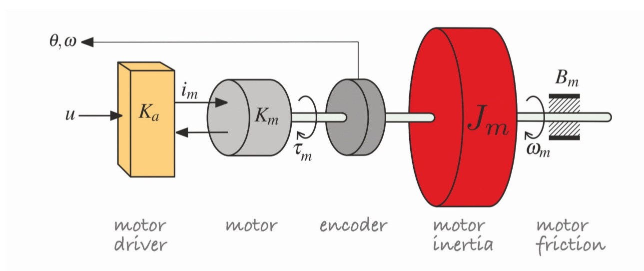
Lorenz-Law: $\tau_m = K_\tau \cdot i$

Power in Elec.domain: $P_m = i \cdot V_b = i\,(K_b \dot{\theta}_m)$

Mechanical Domain :

$$\tau_m = J_m \ddot{\theta}_m + b_m \dot{\theta}_m$$

$$P_m = \tau_m \dot{\theta}_m$$ "mechanical" Power



Motor Modelling

- from mechanical power and electrical power we notice that:

$$P_m = \underbrace{\tau_m \cdot \dot{\theta}_m = k_t \cdot i \cdot \dot{\theta}_m}_{\text{Mechanical}} = \underbrace{k_b \cdot \dot{\theta}_m \cdot i}_{\text{Electrical}}$$

$\therefore$ $\boxed{k_t = k_b}$

Coupling between Mechanical and Electrical Domains :

Motor torque: $\tau_m = k_t \cdot i = J_m \ddot{\theta}_m + b\,\dot{\theta}$

Input Voltage: $v_{in} = \underbrace{\cancel{L_a \cdot \frac{di}{dt}}}_{\simeq 0} + R \cdot i + k_b \cdot \dot{\theta}_m$

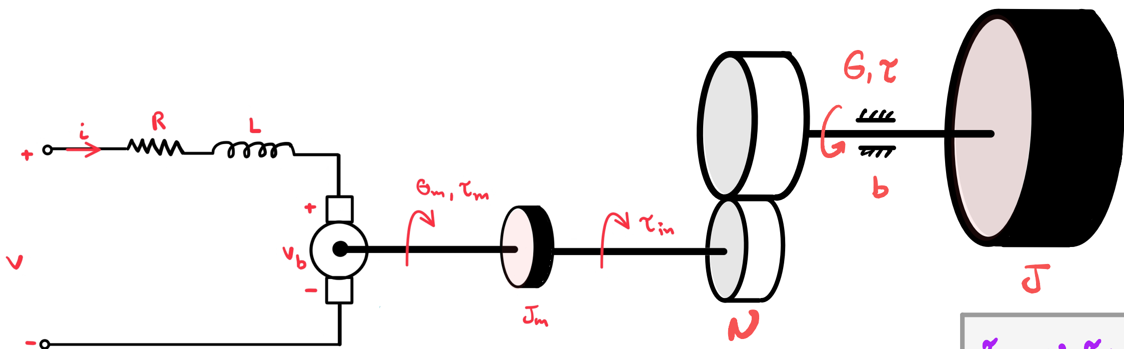
as L_a is so small

$$\boxed{v_{in} = \frac{R}{k_t} J_m \ddot{\theta}_m + \left(k_b + \frac{R \cdot b_m}{k_t}\right) \dot{\theta}_m}$$

Laplace Transform:

$$\boxed{T.F = \frac{\theta_m(s)}{V_{in}(s)} = \frac{k_t / R}{s\left[J_m \cdot s + \left(b_m + \frac{k_b \cdot k_t}{R}\right)\right]}}$$

Dc Motor Model with Gearbox:



$$\tau = N\,\tau_{in}$$
$$\theta = \frac{1}{N}\,\theta_m$$

- Torque at output of gearbox:

$$\tau = N \cdot \tau_{in} = N[\tau_m - b_m \dot{\theta}_m - J_m \ddot{\theta}_m] = b\,\dot{\theta} + J\,\ddot{\theta}$$

$$N\,\tau_m = (b + N^2 b_m) + \underbrace{(J + N^2 J_m)}_{\text{inertia at o/p}}\ddot{\theta}$$

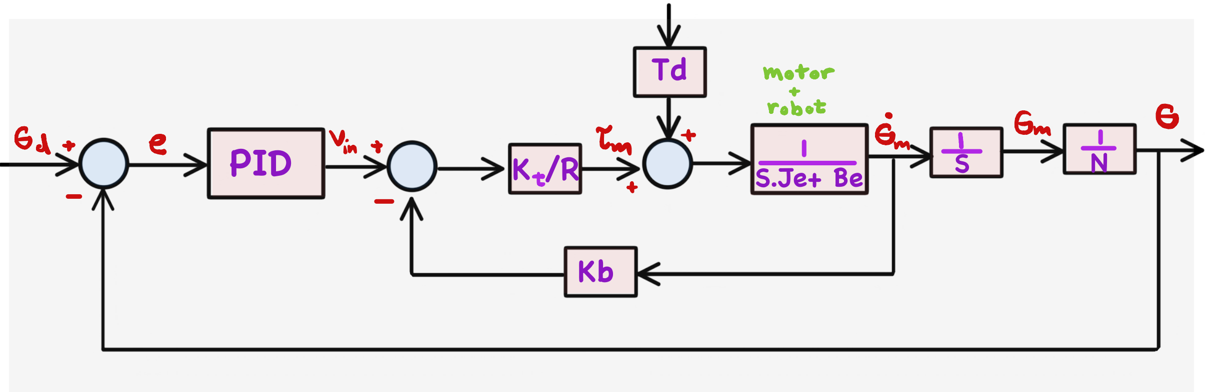
- In terms of motor angles:

$$\tau_m = \underbrace{\left(b_m + \frac{b}{N^2}\right)}_{J_{eff}}\dot{\theta}_m + \underbrace{\left(J_m + \frac{J}{N^2}\right)}_{B_{eff}}\ddot{\theta}_m$$

Notes:

- In multi-joint robot J is not fixed.
- When N is large, motor dynamics dominates. "That's why Linear Control" works
- Industrial robots N>100.
- Design critically damped for max J.

Block Diagram

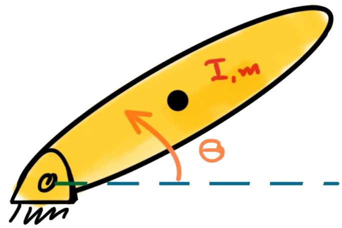


Td : disturbance torque due to

- Inertial coupling
- Centrifugal&Coriolis effect
- Gravitational effects

Regulation Problem

Approach(1): PD Control (linear)



Consider:

- 1-link moving in a plane ($\perp$ to gravity)

Dynamics: $\tau = I\ddot{\theta}$ $\quad$ [NO Gravity]

PD Idea: Create virtual spring/damper to derive the link towards a desired position θ_d

Let:

$$\boxed{\tau = k_p(\theta_d - \theta) + k_d(\dot{\theta}_d - \dot{\theta})}$$

(Torque Input)

Then: closed loop dynamics will be

$$\boxed{I\ddot{\theta} + k_d\dot{\theta} + k_p\theta = k_p\theta_d + k_d\dot{\theta}_d}$$

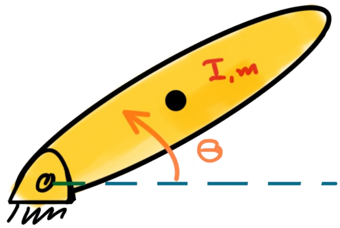
Q: What happens at steady-state?

$$\cancel{I\ddot{\theta}} + \cancel{k_d\dot{\theta}} + \cancel{k_p}\theta = \cancel{k_p}\theta_d + \cancel{k_d}\dot{\theta}_d$$

At steady-state: $\dot{\theta} = \ddot{\theta} = 0$ $\Rightarrow$ $\boxed{\theta = \theta_d}$

Regulation Problem

Approach(2): PD+Feedforward Control (linear)



Dynamics: $\tau = I\ddot{\theta}$

PD Idea: **use** our knowledge of the model **dynamics** to **generate** the required **torque** as feedforward control + the previous PD feedback control

Let:

$$\tau = \underbrace{I\ddot{\theta}_d}_{\text{From Dynamics}} + \underbrace{k_p(\theta_d - \theta) + k_d(\dot{\theta}_d - \dot{\theta})}_{\text{PD Control}}$$

Then: closed loop dynamics will be

$$I\underbrace{(\ddot{\theta}_d - \ddot{\theta})}_{\ddot{e}} + k_d\underbrace{(\dot{\theta}_d - \dot{\theta})}_{\dot{e}} + k_p\underbrace{(\theta_d - \theta)}_{e} = 0$$

$$\Downarrow$$

$$I\ddot{e} + k_d\dot{e} + k_p e = 0$$ "Error Dynamics"

Notes:

- Error dynamics converges to zero

- Transient response depends on (I,k_d,k_p)

Q: How to tune PD gains?🤔

- Knowing link inertia (I) from CAD, we can achieve tracking performance by proper selection for kp,kd

Example:

Let: $I = 2{,}5$

Performance Specifications:

- Settling Time ts = 2 s for reaching (98%)

2^{nd} order system: $1 - e^{-\zeta \omega_n t} = .98$

$$e^{-\zeta \omega_n t_s} = .02 \Rightarrow \zeta \omega_n t_s = \log \frac{1}{.02} = 4$$

$$\boxed{t_s = \frac{4}{\zeta \omega_n}}$$

- Critically damped system: $\zeta = 1$, $\omega_n = \frac{4}{t_s} = \frac{4}{2} = 2$ rad/s

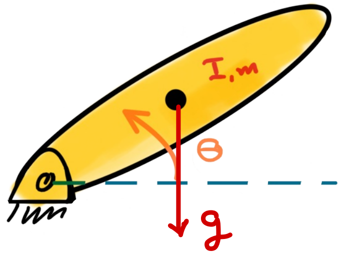
$$\omega_n = \sqrt{\frac{k_p}{I}} = 2 \Rightarrow \boxed{k_p = 4 \cdot I = 10}$$

$$k_d = 2\sqrt{k_p I} = 2\sqrt{10 * 2{,}5} = 10$$

$$\boxed{k_p = k_d = 10}$$

Regulation Problem

Approach(3): PID Control (linear)



Dynamics: $I\ddot{\theta} + f_{dist} + g = \tau$

PD Idea:

- Adding integral action can eliminate steady-state error against unknown constant disturbances.

Let: $$\tau = k_p \cdot e + k_i \int_0^t e \, d\tau + k_d \cdot \dot{e}$$

where, $e = \theta_d - \theta$, $\dot{e} = \dot{\theta}_d - \dot{\theta}$

Then: closed loop dynamics will be

$$I\ddot{\theta} + f_{dist} + g = k_p(\theta_d - \theta) + k_d(\dot{\theta}_d - \dot{\theta}) + k_i \int_0^t (\theta_d - \theta) d\tau$$

At steady-state: $\dot{\theta} = \ddot{\theta} = 0$

$$g + f_{dist} = k_p(\theta_d - \theta) + k_i \int_0^t (\theta_d - \theta) d\tau$$

d/dt

what happen when time evolves?

$$0 = k_p \dot{e} + k_i e$$

Error Converges to zero

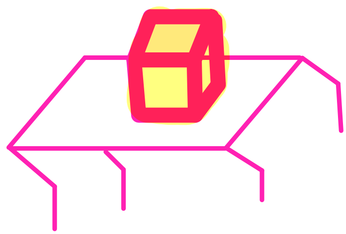
Notes:

- The I-Term grows **magnifying** the error till the disturbance is compensated.

- **Not good** with coulomb friction -> oscillations

Applications:

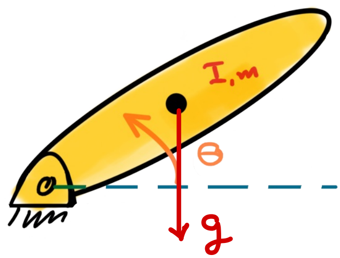
- Unknown payload
- Constant torque offset



Regulation Problem

Approach(4): PD+Gravity Compensation (non-linear)

Dynamics: $I\ddot{\theta} + g(\theta) = \tau$



Q: Why we didn't use PD control only?

- Let's take a look at steady-state error when we use PD only

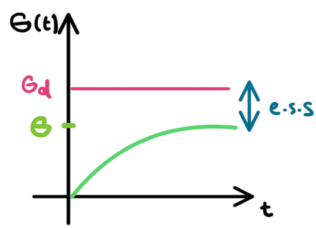
$$\underbrace{I\ddot{\theta} + g(\theta) = \underbrace{k_p(\underbrace{\theta_d - \theta}_{e}) + k_d(\underbrace{\dot{\theta}_d - \dot{\theta}}_{\dot{e}})}_{\tau\ /\ \text{Input}}}$$

- At steady-state: $\dot{\theta} = \ddot{\theta} = 0$

$$k_p(\theta_d - \theta) = g(\theta)$$

$$\boxed{\theta = \theta_d - \frac{g(\theta)}{k_p}}$$

Steady state error



At equilibrium: $\theta_d \neq \theta$

Q: What if we compensate for gravity ?🤔

Let: $$\tau = \underbrace{k_p(\theta_d - \theta) + k_d(\dot{\theta}_d - \dot{\theta})}_{\text{old PD Control}} + \underbrace{g(\theta)}_{\text{Compensate for Gravity}}$$

Then: closed loop dynamics will be

$$I\ddot{\theta} + \cancel{g(\theta)} = k_p(\theta_d - \theta) + k_d(\dot{\theta}_d - \dot{\theta}) + \cancel{g(\theta)}$$

Notes:

- stability ensured for all ($k_p, k_d > 0$).
- $g(\theta)$ is configuration dependant and we need to **recompute** it for **each step**.

Tracking Problem

Approach(1): Inverse Dynamics Control (non-linear)

Dynamics:

$$D(q)\ddot{q} + \underbrace{C(q,\dot{q})\dot{q} + G(q) + \text{friction}}_{=n(q,\dot{q})} = \tau$$

Idea:

- Use our knowledge of robot dynamics to compute the required torque to follow trajectory of $(q,\dot{q})$

Let:

$$\boxed{\tau = D(q_d)\ddot{q}_d + n(q_d,\dot{q}_d)}$$

- Assume $q(0) = q_d$, $\dot{q}(0) = \dot{q}_d$ "Exact Initialization"

Then: closed loop dynamics will be

$$\boxed{\ddot{q}_d = \ddot{q}}$$

Problems:

- **Initial state not matched** to the desired trajectory q_d.
- **External disturbance**.
- **Inaccurate model parameters**.
- **Unknown payload**.
- **Unmodeled dynamics**.

Tracking Problem

Approach(2): Feedback Linearization + (PD+Feedforward)

Dynamics: $\tau = D(q)\ddot{q} + n(q,\dot{q})$

Idea:

1. Use **feedback** for nonlinearity cancellation
2. Use **linear control** for stabilizing error to zero

Let:

Feedback Linearization

$$\tau = D(q)\left[\ddot{q}_q + K_P(q_d - q) + K_D(\dot{q}_d - \dot{q})\right] + n(q,\dot{q})$$

Feedforward

PD Linear Control

Then: closed loop dynamics will be

$$K_P = \begin{bmatrix} K_{P_1} & & 0 \\ & \ddots & \\ 0 & & K_{P_n} \end{bmatrix}$$

1. Feedback Linearization

$$K_D = \begin{bmatrix} K_{d_1} & & 0 \\ & \ddots & \\ 0 & & K_{d_n} \end{bmatrix}$$

$$D\cancel{(q)}\ddot{q} + n\cancel{(q,\dot{q})} = u = \underbrace{D\cancel{(q)}a + n\cancel{(q,\dot{q})}}_{\text{Nonlinear Control}}$$

$$\boxed{\ddot{q} = a}$$

$\boxed{K_P, K_D : \text{Positive Definite}}$

2. Linear control

$$a = \ddot{q} + K_D(\dot{q}_d - \dot{q}) + K_P(q_d - q) \Rightarrow \boxed{\ddot{e} + K_D\dot{e} + K_P e = 0}$$

Block Diagram