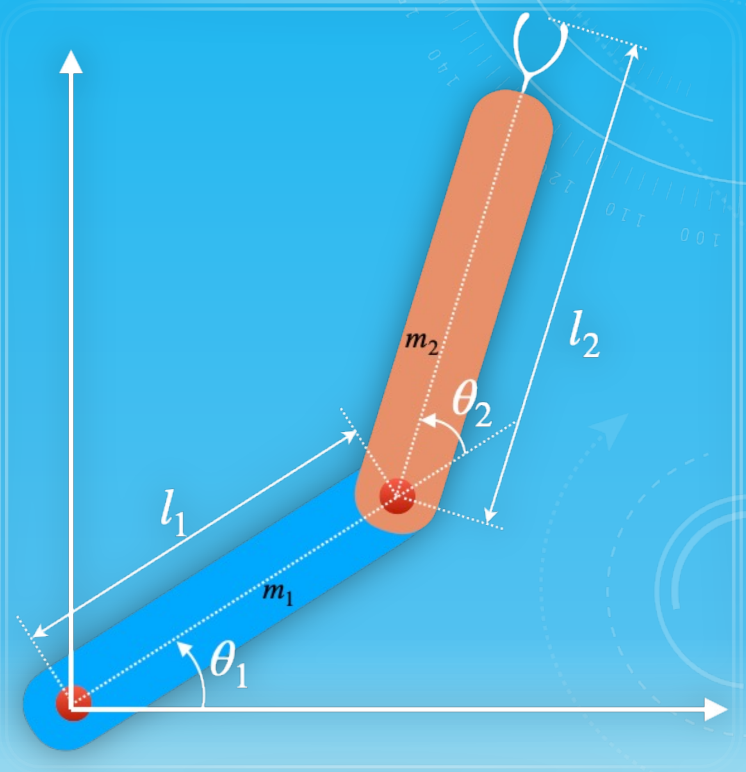
ROBOTICS (OPTIMAL CONTROL)

SUNEESH JACOB A.

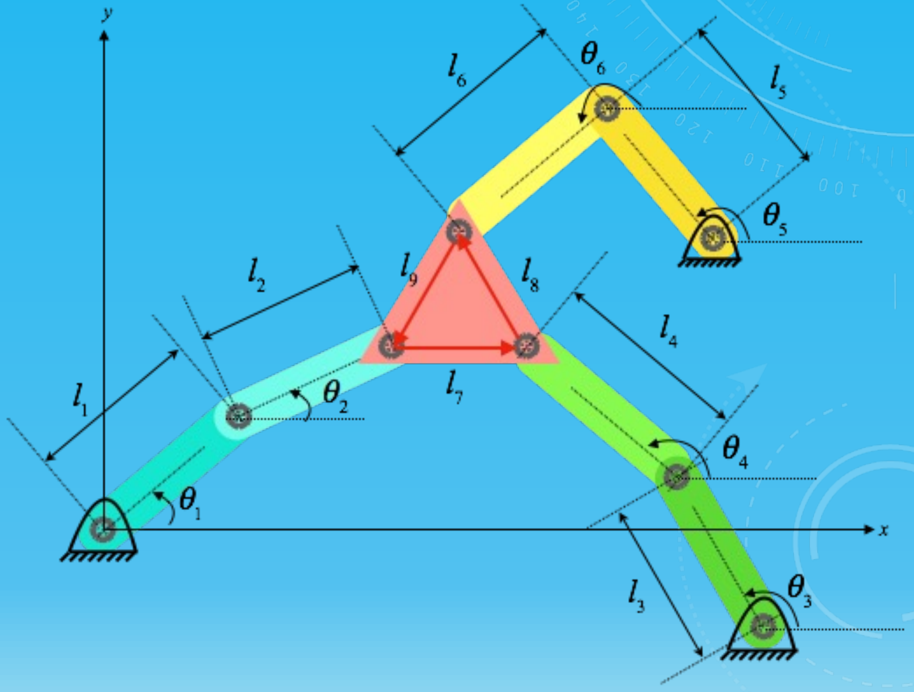
TWO-LINK ROBOT KINEMATICS

- $\vec{p} = \begin{Bmatrix} l_1 \cos(\theta_1) + l_2 \cos(\theta_1 + \theta_2) \\ l_1 \sin(\theta_1) + l_2 \sin(\theta_1 + \theta_2) \end{Bmatrix}$



EIGHT-LINK PARALLEL ROBOT KINEMATICS

- $\vec{p}_1 = \begin{Bmatrix} l_1\cos(\theta_1) + l_2\cos(\theta_2) \\ l_1\sin(\theta_1) + l_2\sin(\theta_2) \end{Bmatrix}$
- $\vec{p}_2 = \begin{Bmatrix} l_3\cos(\theta_3) + l_4\cos(\theta_4) \\ l_3\sin(\theta_3) + l_4\sin(\theta_4) \end{Bmatrix}$
- $\vec{p}_3 = \begin{Bmatrix} l_5\cos(\theta_5) + l_6\cos(\theta_6) \\ l_5\sin(\theta_5) + l_6\sin(\theta_6) \end{Bmatrix}$



- $\vec{o}_2 = \vec{o}_1 + \begin{Bmatrix} l_1\cos(\theta_1) + l_2\cos(\theta_2) + l_7\cos(\theta_7) - l_4\cos(\theta_4) - l_3\cos(\theta_3) \\ l_1\sin(\theta_1) + l_2\sin(\theta_2) + l_7\sin(\theta_7) - l_4\sin(\theta_4) - l_3\sin(\theta_3) \end{Bmatrix}$
- $\vec{o}_3 = \vec{o}_2 + \begin{Bmatrix} l_3\cos(\theta_3) + l_4\cos(\theta_4) + l_8\cos(\theta_8) - l_6\cos(\theta_6) - l_5\cos(\theta_5) \\ l_3\sin(\theta_3) + l_4\sin(\theta_4) + l_8\sin(\theta_8) - l_6\sin(\theta_6) - l_5\sin(\theta_5) \end{Bmatrix}$
- $\vec{o}_1 = \vec{o}_3 + \begin{Bmatrix} l_5\cos(\theta_5) + l_6\cos(\theta_6) + l_9\cos(\theta_9) - l_2\cos(\theta_2) - l_1\cos(\theta_1) \\ l_5\sin(\theta_5) + l_6\sin(\theta_6) + l_9\sin(\theta_9) - l_2\sin(\theta_2) - l_1\sin(\theta_1) \end{Bmatrix}$

WHAT IS MOTOR CONTROL

Control: Computing appropriate torque to bring the robot's actuating joint from one state to another state

Drive: Converting the small current/voltage signals to real-time (large) signals that can be fed to motor

WHY DO WE NEED MOTOR CONTROL

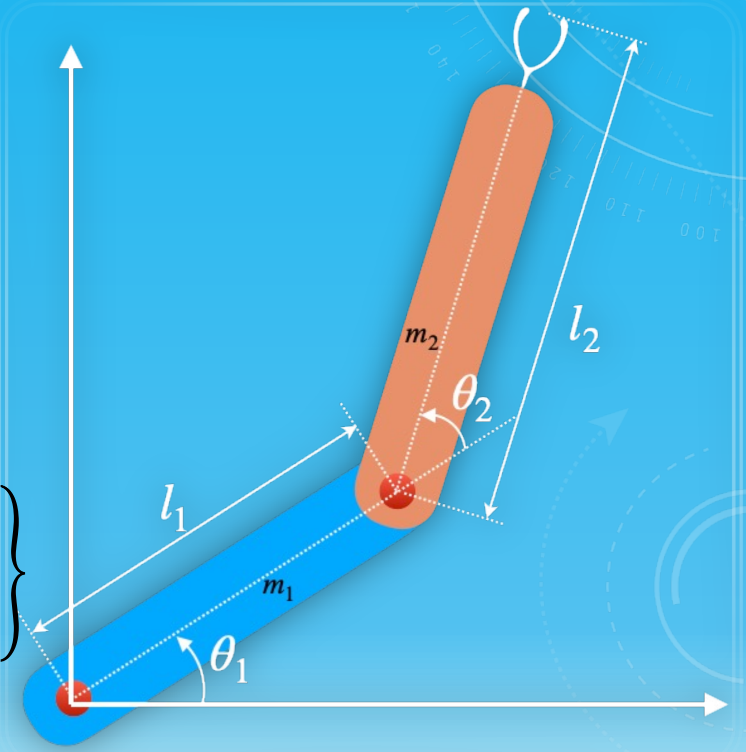
- Input → force
- Output → state (position and velocity)

DYNAMICS

- Dynamic equations

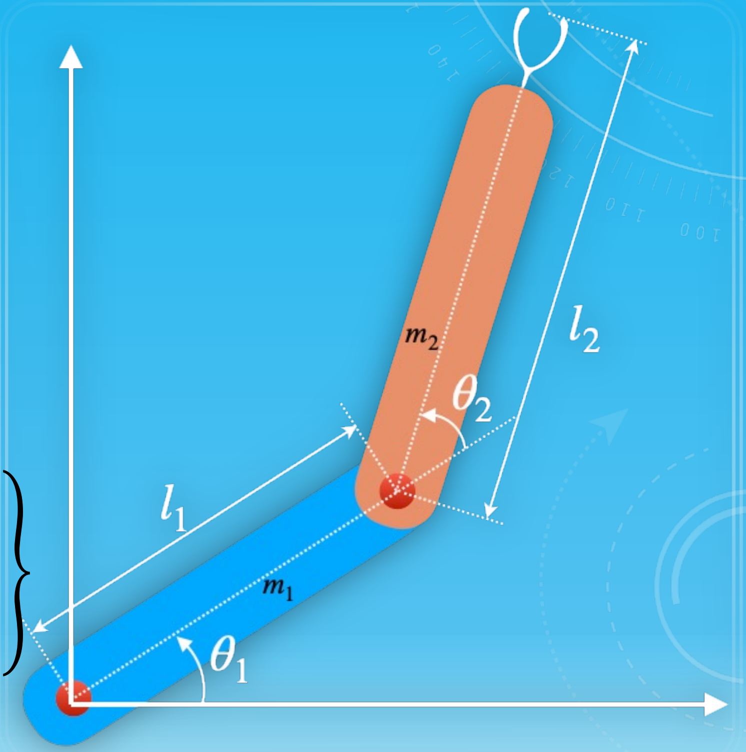
TWO-LINK ROBOT DYNAMICS

- $\vec{p}_1 = \begin{Bmatrix} \frac{l_1}{2}\cos(\theta_1) \\ \frac{l_1}{2}\sin(\theta_1) \end{Bmatrix}$
- $\vec{p}_2 = \begin{Bmatrix} l_1\cos(\theta_1) + \frac{l_2}{2}\cos(\theta_1+\theta_2) \\ l_1\sin(\theta_1) + \frac{l_2}{2}\sin(\theta_1+\theta_2) \end{Bmatrix}$
- $\vec{v}_1 = \begin{Bmatrix} -\frac{l_1}{2}\sin(\theta_1)\,\dot{\theta}_1 \\ \frac{l_1}{2}\cos(\theta_1)\,\dot{\theta}_1 \end{Bmatrix}$
- $\vec{v}_2 = \begin{Bmatrix} -l_1\sin(\theta_1)\,\dot{\theta}_1 - \frac{l_2}{2}\sin(\theta_1+\theta_2)\,(\dot{\theta}_1+\dot{\theta}_2) \\ -l_1\cos(\theta_1)\,\dot{\theta}_1 + \frac{l_2}{2}\cos(\theta_1+\theta_2)\,(\dot{\theta}_1+\dot{\theta}_2) \end{Bmatrix}$
- $\omega_1 = \dot{\theta}_1$
- $\omega_2 = \dot{\theta}_1 + \dot{\theta}_2$



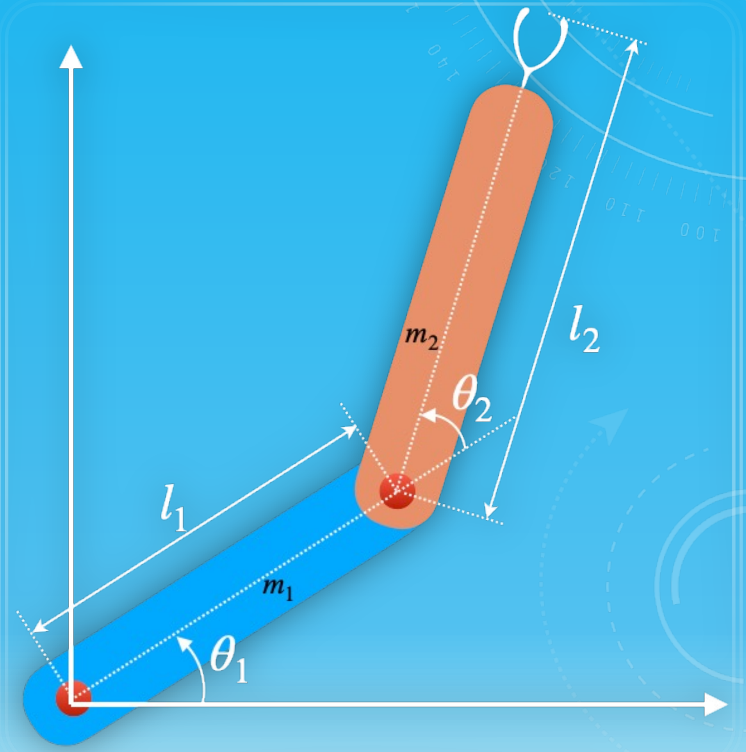
TWO-LINK ROBOT DYNAMICS

- $\vec{p}_1 = \begin{Bmatrix} \frac{l_1}{2}\cos(\theta_1) \\ \frac{l_1}{2}\sin(\theta_1) \\ 0 \end{Bmatrix}$
- $\vec{p}_2 = \begin{Bmatrix} l_1\cos(\theta_1) + \frac{l_2}{2}\cos(\theta_1+\theta_2) \\ l_1\sin(\theta_1) + \frac{l_2}{2}\sin(\theta_1+\theta_2) \\ 0 \end{Bmatrix}$
- $\vec{v}_1 = \begin{Bmatrix} -\frac{l_1}{2}\sin(\theta_1)\,\dot{\theta}_1 \\ \frac{l_1}{2}\cos(\theta_1)\,\dot{\theta}_1 \\ 0 \end{Bmatrix}$
- $\vec{v}_2 = \begin{Bmatrix} -l_1\sin(\theta_1)\,\dot{\theta}_1 - \frac{l_2}{2}\sin(\theta_1+\theta_2)\,(\dot{\theta}_1+\dot{\theta}_2) \\ -l_1\cos(\theta_1)\,\dot{\theta}_1 + \frac{l_2}{2}\cos(\theta_1+\theta_2)\,(\dot{\theta}_1+\dot{\theta}_2) \\ 0 \end{Bmatrix}$
- $\vec{\omega}_1 = \begin{Bmatrix} 0 \\ 0 \\ \dot{\theta}_1 \end{Bmatrix}$
- $\vec{\omega}_2 = \begin{Bmatrix} 0 \\ 0 \\ \dot{\theta}_1 + \dot{\theta}_2 \end{Bmatrix}$



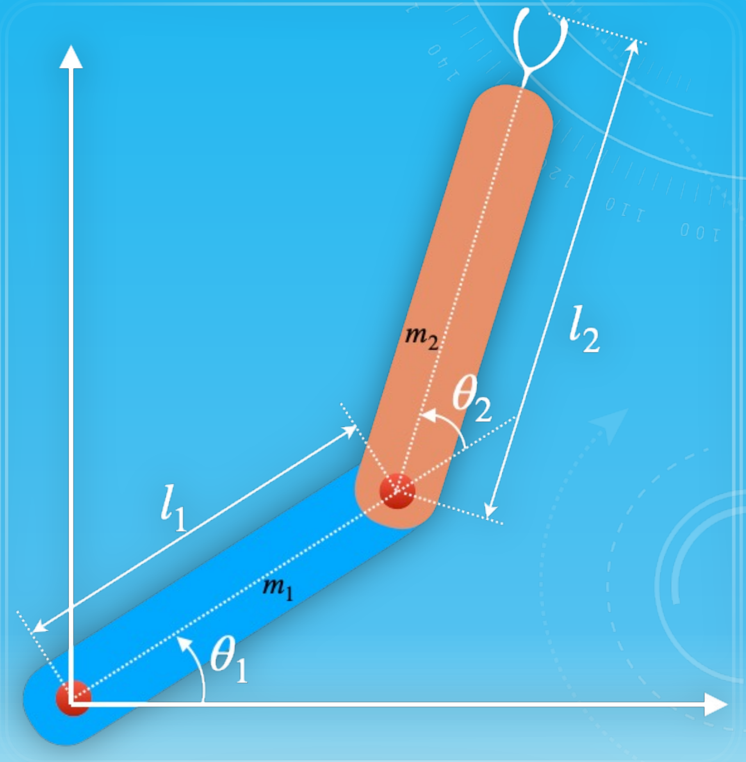
TWO-LINK ROBOT DYNAMICS

- $T = \frac{1}{2}m_1(\vec{v}_1 \cdot \vec{v}_1) + \frac{1}{2}m_2(\vec{v}_2 \cdot \vec{v}_2) + \frac{1}{2}I_1(\vec{\omega}_1 \cdot \vec{\omega}_1) + \frac{1}{2}I_2(\vec{\omega}_2 \cdot \vec{\omega}_2)$
- $V = 0$
- $L = T - V$
- $\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\theta}_1}\right) - \frac{\partial L}{\partial \theta_1} = \tau_1$
- $\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\theta}_2}\right) - \frac{\partial L}{\partial \theta_2} = \tau_2$



TWO-LINK ROBOT DYNAMICS

- $\tau_1 = \left(\frac{1}{3}m_1 l_1^2 + m_2\left(l_1^2 + \frac{1}{3}l_2^2 + l_1 l_2 \cos(\theta_2)\right)\right)\ddot{\theta}_1 + \left(\frac{1}{3}m_2 l_2^2 + \frac{1}{2}m_2 l_1 l_2 \cos(\theta_2)\right)\ddot{\theta}_2 - m_2 l_1 l_2 \sin(\theta_2)\left(\dot{\theta}_1\dot{\theta}_2 + \frac{1}{2}\dot{\theta}_2^2\right)$
- $\tau_2 = m_2\left(\frac{1}{3}l_2^2 + \frac{1}{2}l_1 l_2 \cos(\theta_2)\right)\ddot{\theta}_1 + \left(\frac{1}{3}m_2 l_2^2\right)\ddot{\theta}_2 + \left(\frac{1}{2}m_2 l_1 l_2 \sin(\theta_2)\right)\dot{\theta}_1^2$



TWO-LINK ROBOT DYNAMICS

- $\tau_1 = \left(\frac{1}{3}m_1 l_1^2 + m_2\left(l_1^2 + \frac{1}{3}l_2^2 + l_1 l_2\cos(\theta_2)\right)\right)\ddot{\theta}_1 + \left(\frac{1}{3}m_2 l_2^2 + \frac{1}{2}m_2 l_1 l_2\cos(\theta_2)\right)\ddot{\theta}_2 - m_2 l_1 l_2 \sin(\theta_2)\left(\dot{\theta}_1\dot{\theta}_2 + \frac{1}{2}\dot{\theta}_2^2\right)$
- $\tau_2 = m_2\left(\frac{1}{3}l_2^2 + \frac{1}{2}l_1 l_2\cos(\theta_2)\right)\ddot{\theta}_1 + \left(\frac{1}{3}m_2 l_2^2\right)\ddot{\theta}_2 + \left(\frac{1}{2}m_2 l_1 l_2\sin(\theta_2)\right)\dot{\theta}_1^2$

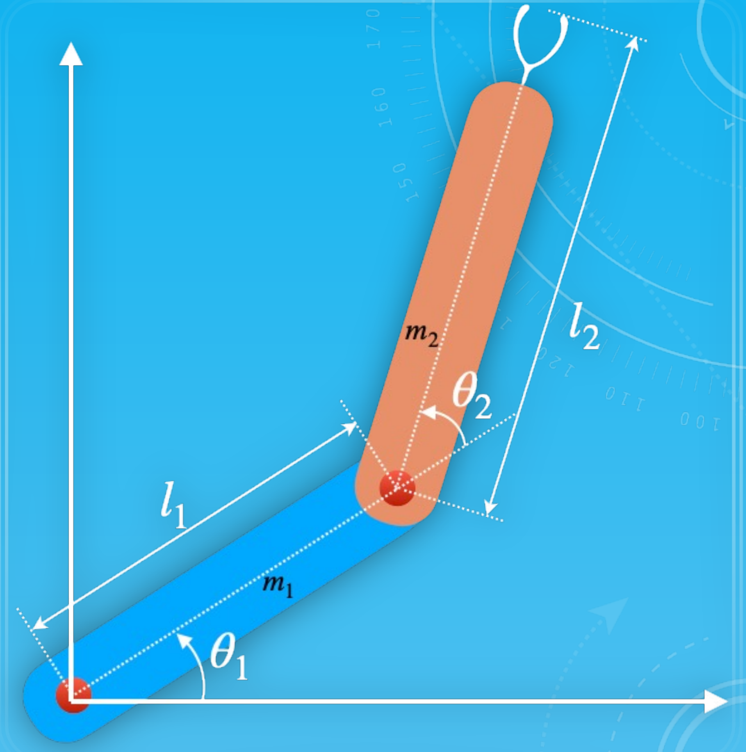
$$\begin{bmatrix} A & B \\ C & D \end{bmatrix}\begin{Bmatrix}\ddot{\theta}_1 \\ \ddot{\theta}_2\end{Bmatrix} + \begin{Bmatrix} E \\ F \end{Bmatrix} = \begin{Bmatrix}\tau_1 \\ \tau_2\end{Bmatrix}$$

$$\Rightarrow \begin{Bmatrix}\ddot{\theta}_1 \\ \ddot{\theta}_2\end{Bmatrix} = \begin{bmatrix} A & B \\ C & D \end{bmatrix}^{-1}\left(\begin{Bmatrix}\tau_1 \\ \tau_2\end{Bmatrix} - \begin{Bmatrix} E \\ F \end{Bmatrix}\right)$$

$$\ddot{\theta}_1 = \frac{D\tau_1 - B\tau_2 + BF - DE}{AD - BC}$$

$$\ddot{\theta}_2 = \frac{-C\tau_1 + A\tau_2 + EC - AF}{AD - BC}$$

$$\ddot{\theta}_1 = f_1(\theta_1, \dot{\theta}_1, \theta_2, \dot{\theta}_2, \tau_1, \tau_2)$$
$$\ddot{\theta}_2 = f_2(\theta_1, \dot{\theta}_1, \theta_2, \dot{\theta}_2, \tau_1, \tau_2)$$



$$A = \frac{1}{3}m_1 l_1^2 + m_2\left(l_1^2 + \frac{1}{3}l_2^2 + l_1 l_2\cos(\theta_2)\right)$$
$$B = C = \frac{1}{3}m_2 l_2^2 + \frac{1}{2}m_2 l_1 l_2\cos(\theta_2)$$
$$D = \frac{1}{3}m_2 l_2^2$$
$$E = -m_2 l_1 l_2\sin(\theta_2)\left(\dot{\theta}_1\dot{\theta}_2 + \frac{1}{2}\dot{\theta}_2^2\right)$$
$$F = \left(\frac{1}{2}m_2 l_1 l_2\sin(\theta_2)\right)\dot{\theta}_1^2$$

TWO-LINK ROBOT DYNAMICS

- $\tau_1 = \left(\frac{1}{3}m_1 l_1^2 + m_2\left(l_1^2 + \frac{1}{3}l_2^2 + l_1 l_2\cos(\theta_2)\right)\right)\ddot{\theta}_1 + \left(\frac{1}{3}m_2 l_2^2 + \frac{1}{2}m_2 l_1 l_2\cos(\theta_2)\right)\ddot{\theta}_2 - m_2 l_1 l_2 \sin(\theta_2)\left(\dot{\theta}_1\dot{\theta}_2 + \frac{1}{2}\dot{\theta}_2^2\right)$
- $\tau_2 = m_2\left(\frac{1}{3}l_2^2 + \frac{1}{2}l_1 l_2\cos(\theta_2)\right)\ddot{\theta}_1 + \left(\frac{1}{3}m_2 l_2^2\right)\ddot{\theta}_2 + \left(\frac{1}{2}m_2 l_1 l_2\sin(\theta_2)\right)\dot{\theta}_1^2$

$$\ddot{\theta}_1 = \frac{D\tau_1 - B\tau_2 + BF - DE}{AD - BC}$$

$$\ddot{\theta}_2 = \frac{-C\tau_1 + A\tau_2 + EC - AF}{AD - BC}$$

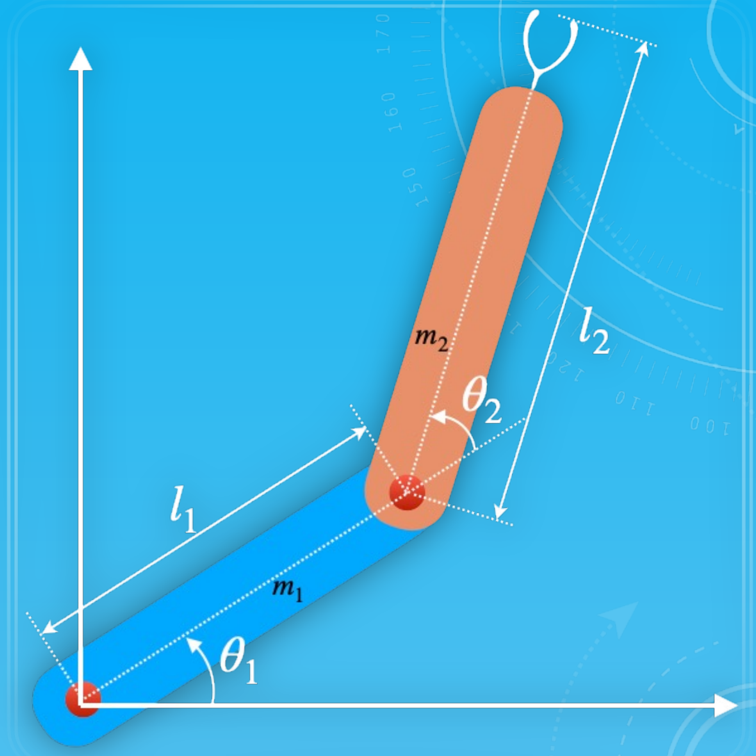
$$z_1 = \theta_1, z_2 = \theta_2, z_3 = \dot{\theta}_1, z_4 = \dot{\theta}_2$$

$$\dot{z}_1 = \dot{\theta}_1 = z_2$$

$$\dot{z}_2 = \dot{\theta}_2 = z_4$$

$$\dot{z}_3 = \ddot{\theta}_1 = \frac{D\tau_1 - B\tau_2 + BF - DE}{AD - BC}$$

$$\dot{z}_4 = \ddot{\theta}_2 = \frac{-C\tau_1 + A\tau_2 + EC - AF}{AD - BC}$$



$$A = \frac{1}{3}m_1 l_1^2 + m_2\left(l_1^2 + \frac{1}{3}l_2^2 + l_1 l_2\cos(\theta_2)\right)$$

$$B = C = \frac{1}{3}m_2 l_2^2 + \frac{1}{2}m_2 l_1 l_2\cos(\theta_2)$$

$$D = \frac{1}{3}m_2 l_2^2$$

$$E = -m_2 l_1 l_2\sin(\theta_2)\left(\dot{\theta}_1\dot{\theta}_2 + \frac{1}{2}\dot{\theta}_2^2\right)$$

$$F = \left(\frac{1}{2}m_2 l_1 l_2\sin(\theta_2)\right)\dot{\theta}_1^2$$

TWO-LINK ROBOT DYNAMICS

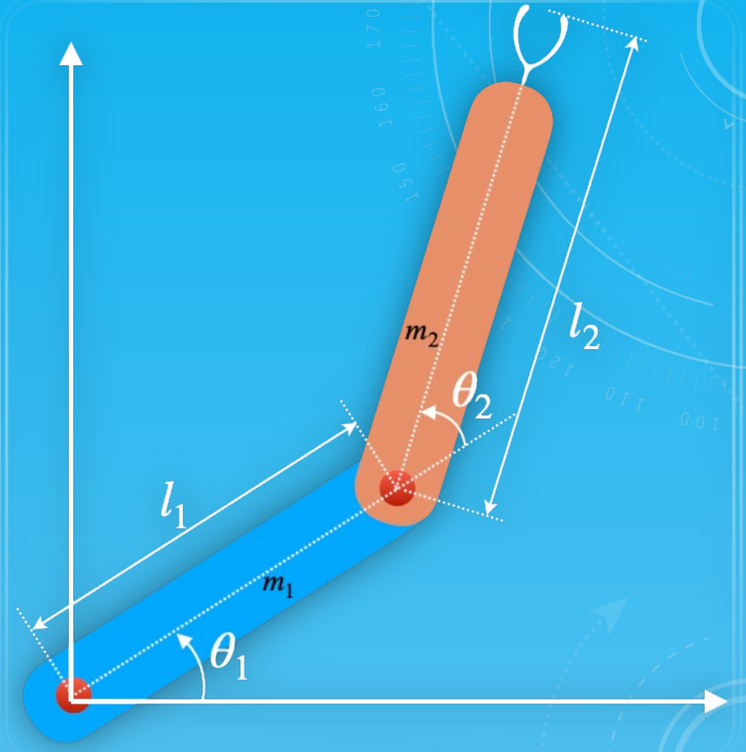
- $\tau_1 = \left(\frac{1}{3}m_1 l_1^2 + m_2\left(l_1^2 + \frac{1}{3}l_2^2 + l_1 l_2\cos(\theta_2)\right)\right)\ddot{\theta}_1 + \left(\frac{1}{3}m_2 l_2^2 + \frac{1}{2}m_2 l_1 l_2\cos(\theta_2)\right)\ddot{\theta}_2 - m_2 l_1 l_2 \sin(\theta_2)\left(\dot{\theta}_1\dot{\theta}_2 + \frac{1}{2}\dot{\theta}_2^2\right)$
- $\tau_2 = m_2\left(\frac{1}{3}l_2^2 + \frac{1}{2}l_1 l_2\cos(\theta_2)\right)\ddot{\theta}_1 + \left(\frac{1}{3}m_2 l_2^2\right)\ddot{\theta}_2 + \left(\frac{1}{2}m_2 l_1 l_2\sin(\theta_2)\right)\dot{\theta}_1^2$

$$\ddot{\theta}_1 = \frac{D\tau_1 - B\tau_2 + BF - DE}{AD - BC}$$

$$\ddot{\theta}_2 = \frac{-C\tau_1 + A\tau_2 + EC - AF}{AD - BC}$$

$$z_1 = \theta_1, z_2 = \theta_2, z_3 = \dot{\theta}_1, z_4 = \dot{\theta}_2$$

$$\begin{Bmatrix} \dot{z}_1 \\ \dot{z}_2 \\ \dot{z}_3 \\ \dot{z}_4 \end{Bmatrix} = \begin{Bmatrix} z_2 \\ z_4 \\ \dfrac{D\tau_1 - B\tau_2 + BF - DE}{AD - BC} \\ \dfrac{-C\tau_1 + A\tau_2 + EC - AF}{AD - BC} \end{Bmatrix} = f(t, \boldsymbol{z}, \boldsymbol{\tau})$$



$$A = \frac{1}{3}m_1 l_1^2 + m_2\left(l_1^2 + \frac{1}{3}l_2^2 + l_1 l_2\cos(\theta_2)\right)$$

$$B = C = \frac{1}{3}m_2 l_2^2 + \frac{1}{2}m_2 l_1 l_2\cos(\theta_2)$$

$$D = \frac{1}{3}m_2 l_2^2$$

$$E = -m_2 l_1 l_2 \sin(\theta_2)\left(\dot{\theta}_1\dot{\theta}_2 + \frac{1}{2}\dot{\theta}_2^2\right)$$

$$F = \left(\frac{1}{2}m_2 l_1 l_2\sin(\theta_2)\right)\dot{\theta}_1^2$$

SOLUTION

- Euler's method
  - $\frac{dz}{dt} = f(t, z)$
  - $\frac{z_{t+\Delta t} - z_t}{\Delta t} = f(t, z_t)$
  - $z_{t+\Delta t} = z_t + \Delta t f(t, z_t)$

SOLUTION

- Runge–Kutta method
  - $\frac{dz}{dt} = f(t, z)$
  - $z_{t+\Delta t} = z_t + \frac{\Delta t}{6}(k_1 + 2k_2 + 2k_3 + k_4)$
    - $k_1 = f(t, z_t)$
    - $k_2 = f\left(t + \frac{\Delta t}{2}, z_t + \Delta t \frac{k_1}{2}\right)$
    - $k_3 = f\left(t + \frac{\Delta t}{2}, z_t + \Delta t \frac{k_2}{2}\right)$
    - $k_4 = f(t + \Delta t, z_t + \Delta t k_3)$

CONTROL STRATEGIES

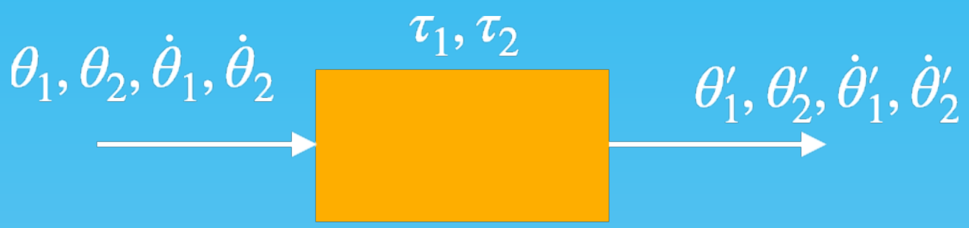
- Open-loop controllers
- Closed-loop controllers
- Fuzzy logic controllers
- ML-based controllers
- …

CLOSED-LOOP CONTROLLERS

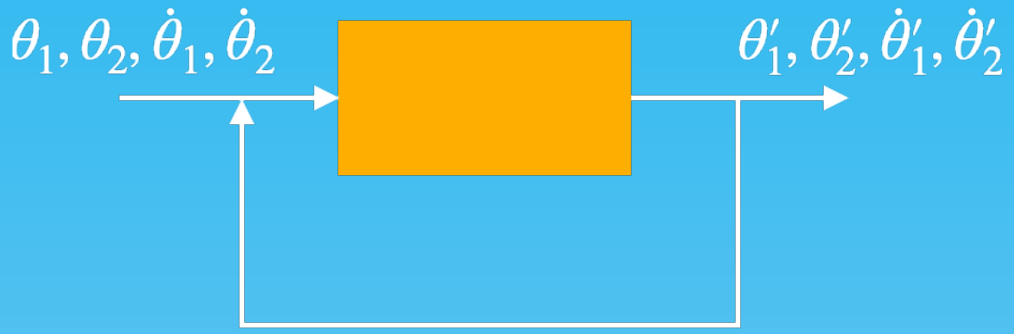
- PID controller

SOLUTION

- Open loop control



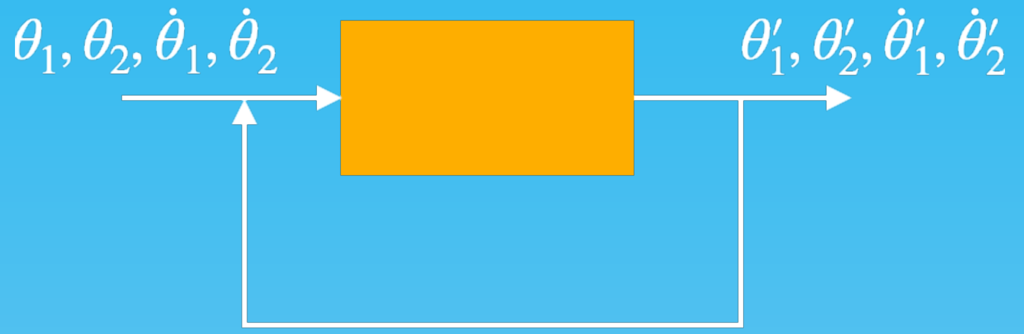
ERROR



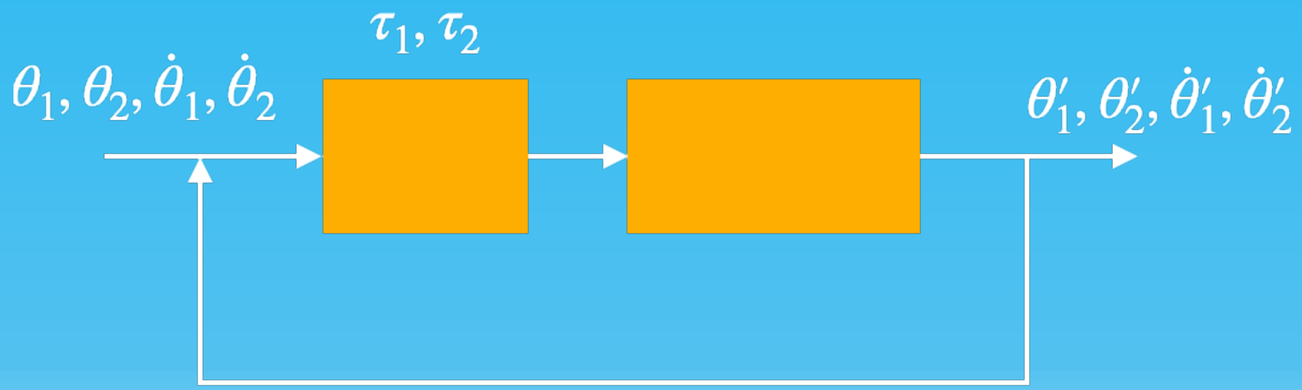
$\theta_1, \theta_2, \dot{\theta}_1, \dot{\theta}_2$
$\theta'_1, \theta'_2, \dot{\theta}'_1, \dot{\theta}'_2$

ERROR

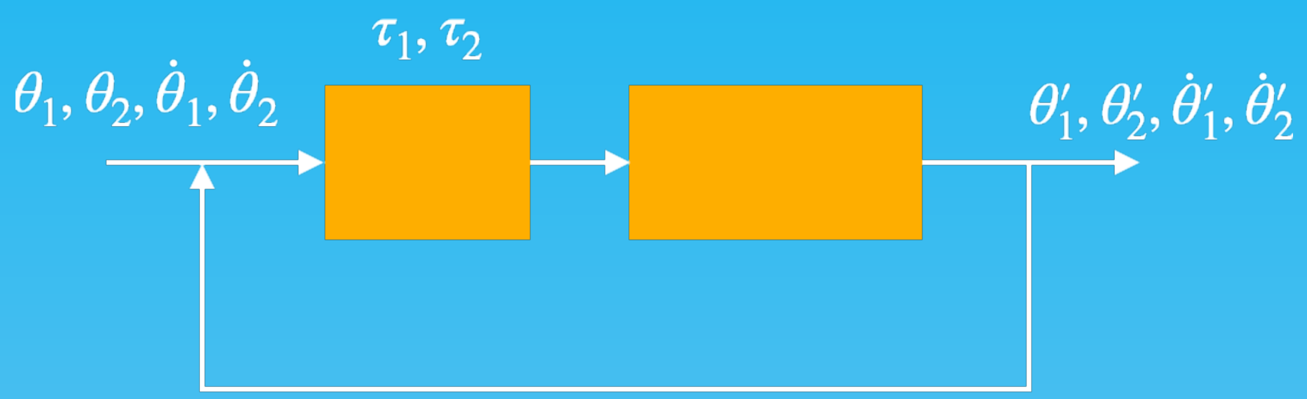
- $e_1 = \theta_1 - \theta'_1$
- $e_2 = \theta_2 - \theta'_2$
- $\dot{e}_1 = \dot{\theta}_1 - \dot{\theta}'_1$
- $\dot{e}_2 = \dot{\theta}_2 - \dot{\theta}'_2$



ERROR


τ_1, τ_2
$\theta_1, \theta_2, \dot{\theta}_1, \dot{\theta}_2$
$\theta'_1, \theta'_2, \dot{\theta}'_1, \dot{\theta}'_2$

ERROR



- $\tau_1 = F_1(\theta_1, \dot{\theta}_1, \theta_2, \dot{\theta}_2)$
- $\tau_2 = F_2(\theta_1, \dot{\theta}_1, \theta_2, \dot{\theta}_2)$

    - $\tau_1 = K_p e_1 + K_d \dot{e}_1 + K_i \int_0^t e_1 \, dt$

CONTROL STRATEGIES

- Open-loop controllers
- Closed-loop controllers
- Fuzzy logic controllers
- ML-based controllers
- …

OPEN-LOOP CONTROLLERS

STEPPER-MOTOR

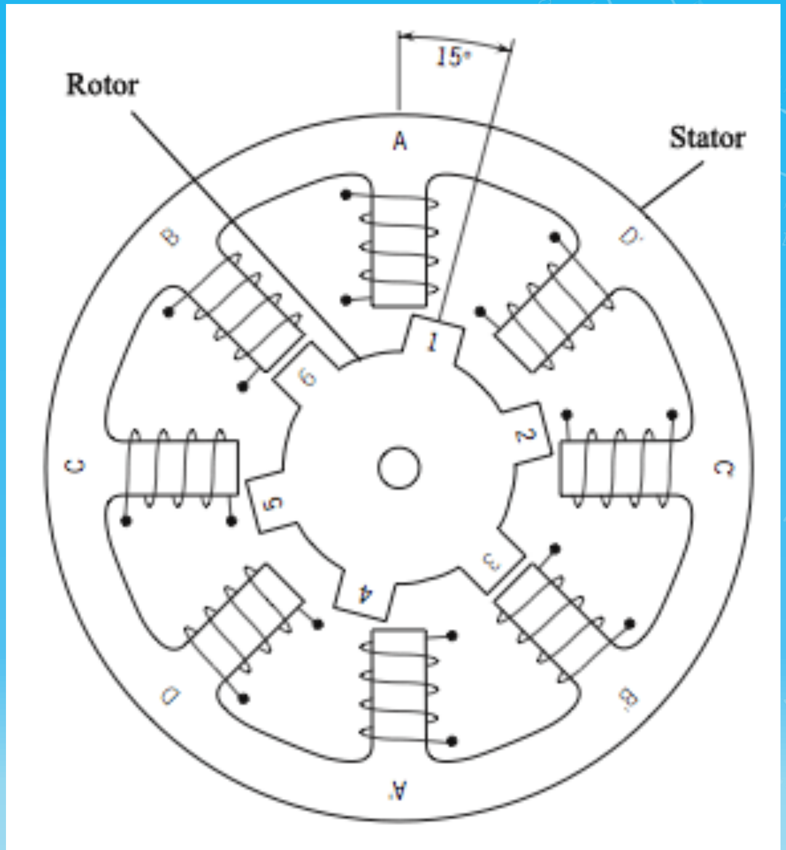


Image source: https://circuitdigest.com/sites/default/files/inlineimages/u/Stepper-Motor-Internal-Structure.png

FUZZY LOGIC CONTROL

- Rules-based

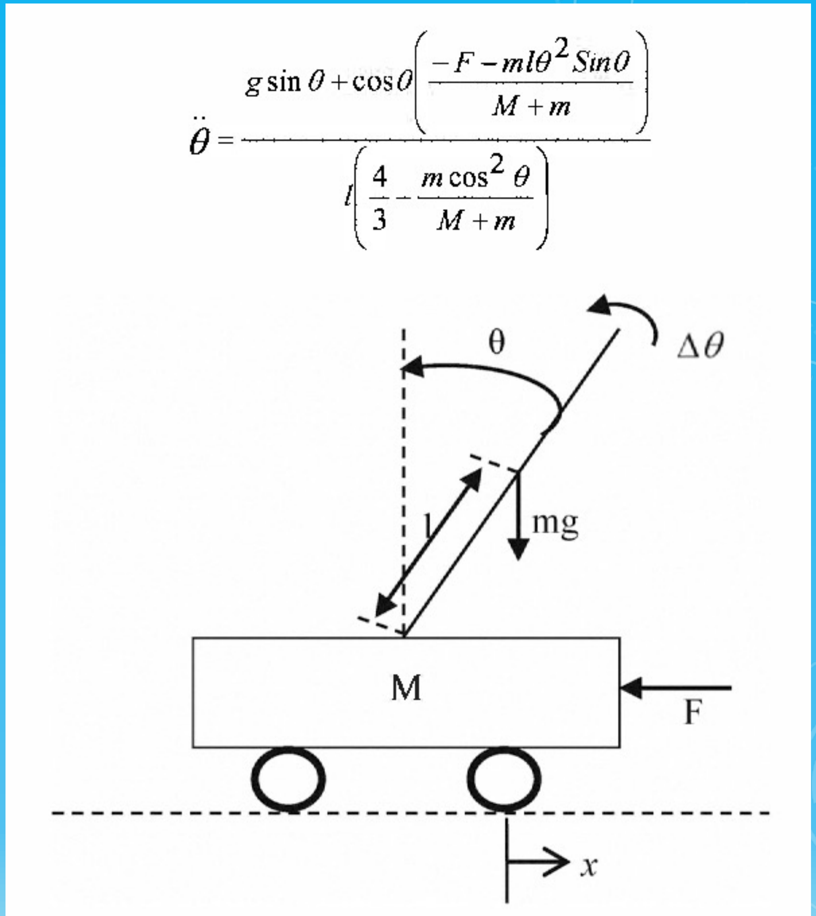


Figure 2. Cart-pole typed inverted pendulum system

Image source: https://www.semanticscholar.org/paper/Fuzzy-logic-controller-for-an-inverted-pendulum-Shill-Akhand/e6cf50180f74be11ffd7d9d520b36dd1650aae6c/figure/1

FUZZY LOGIC CONTROL

- Rules-based

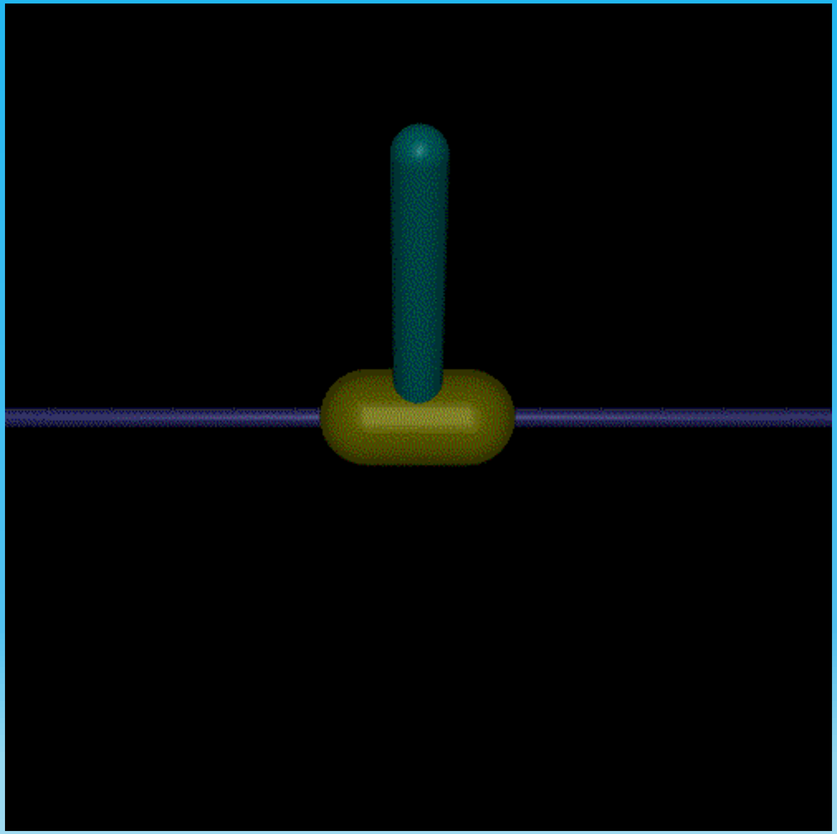


Image source: https://mgoulao.github.io/gym-docs/_images/inverted_pendulum.gif

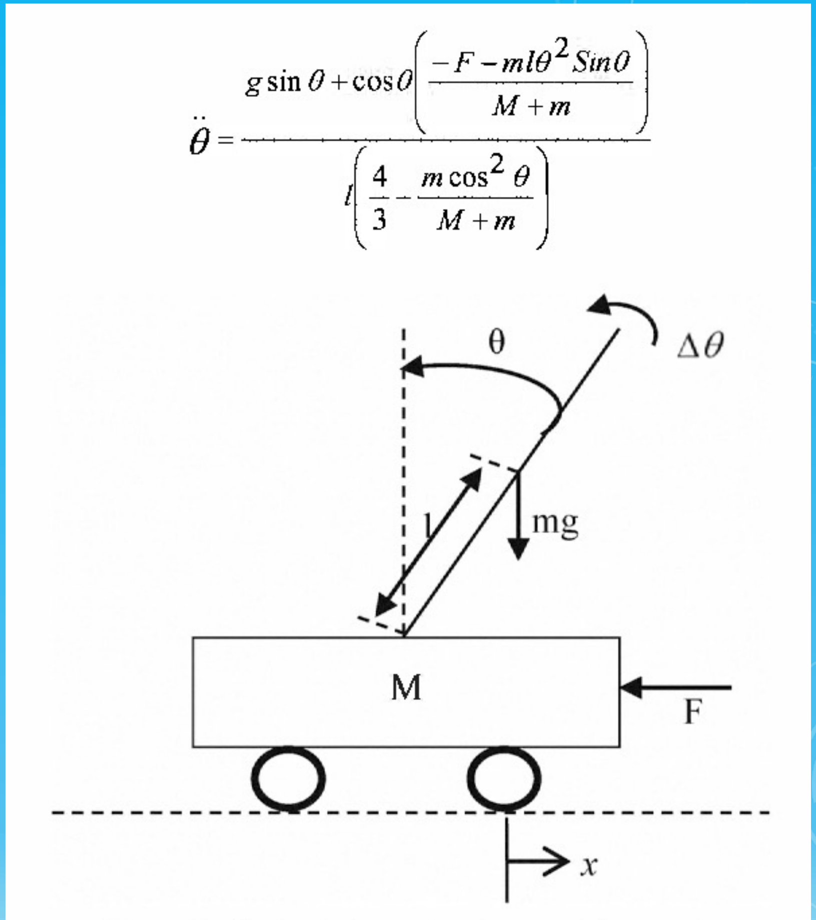


Figure 2. Cart-pole typed inverted pendulum system

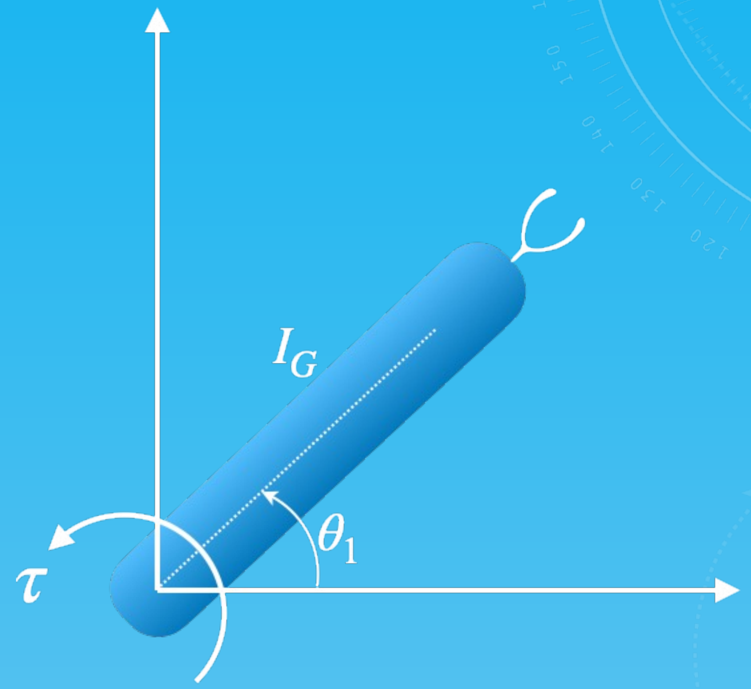
Image source: https://www.semanticscholar.org/paper/Fuzzy-logic-controller-for-an-inverted-pendulum-Shill-Akhand/e6cf50180f74be11ffd7d9d520b36dd1650aae6c/figure/1

MACHINE LEARNING

- Data-driven controlling

MACHINE LEARNING

- Data-driven controlling
  - Joint flexibility
  - Link flexibility
  - Base flexibility



MACHINE LEARNING

- Data-driven controlling
  - Joint flexibility
  - Link flexibility
  - Base flexibility

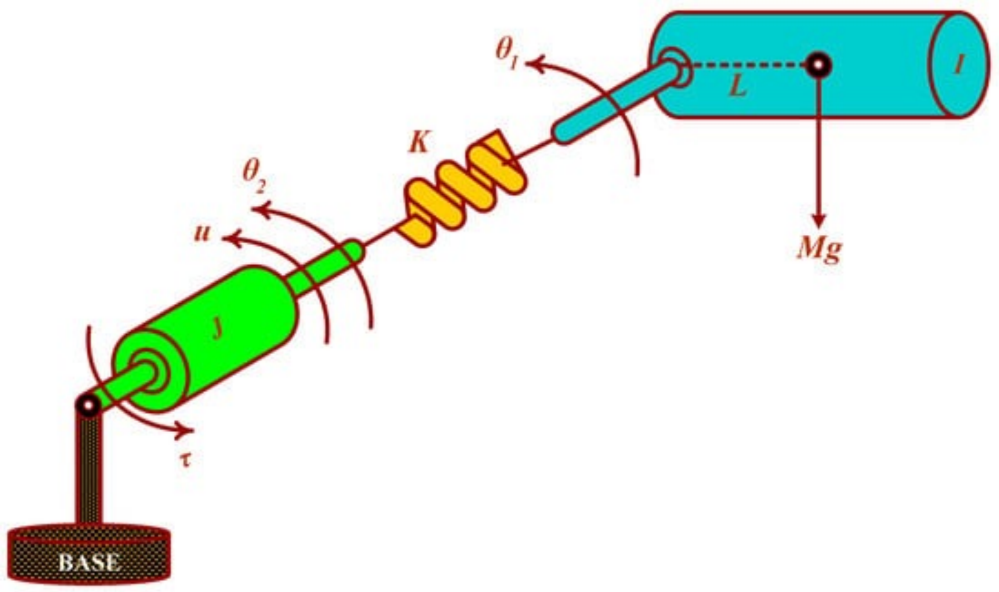


Image source: https://www.mdpi.com/sensors/sensors-21-03252/article_deploy/html/images/sensors-21-03252-g001-550.jpg

MACHINE LEARNING

- Data-driven controlling
    - Joint flexibility
    - Link flexibility
    - Base flexibility

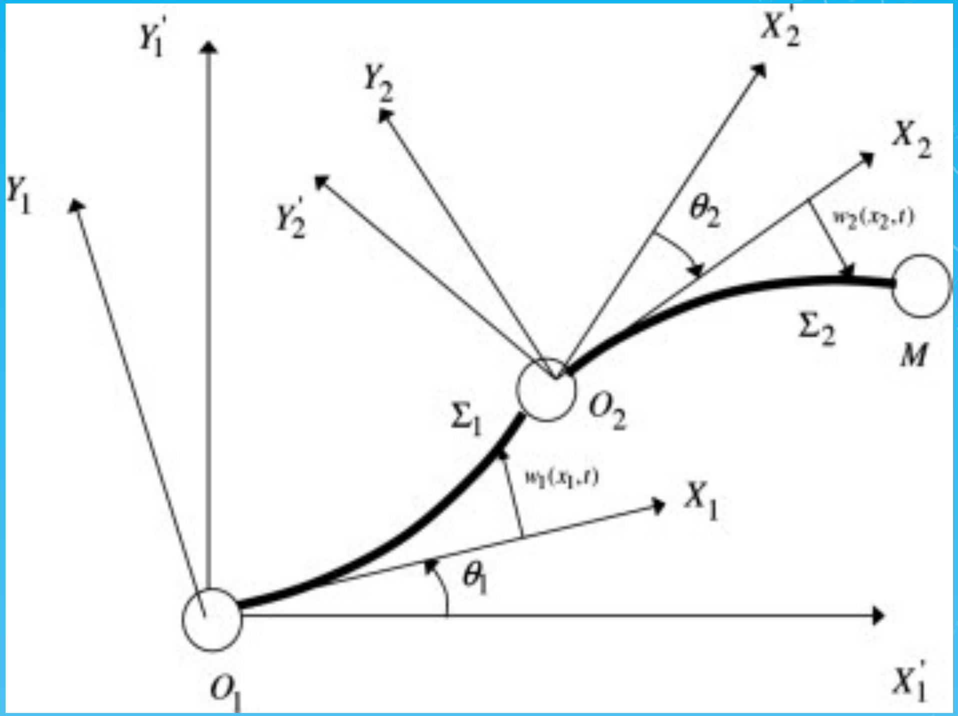


Image source: https://ars.els-cdn.com/content/image/1-s2.0-S0307904X09000183-gr2.jpg

MODEL PREDICTIVE CONTROL

Optimal Control

- Minimise

$$J = \boldsymbol{e}_1^2 + \boldsymbol{e}_2^2 + \cdots + \boldsymbol{e}_N^2 + \boldsymbol{\tau}_0^2 + \boldsymbol{\tau}_1^2 + \cdots + \boldsymbol{\tau}_{N-1}^2$$

- subject to

$$\boldsymbol{z}_{k+1} = \boldsymbol{z}_k + \Delta t\, \boldsymbol{f}(t, \boldsymbol{z}_k, \boldsymbol{\tau}_k) \quad \text{for} \quad k = 0,1,2,\dots,N-1$$

- where

$$\boldsymbol{e}_i = \boldsymbol{z}_i - \boldsymbol{z}_d$$

$$\boldsymbol{\tau}_i = \begin{Bmatrix} \tau_1 \\ \tau_2 \end{Bmatrix} \text{ at } i^{\text{th}} \text{ step}$$

MODEL PREDICTIVE CONTROL

Optimal Control

- Minimise

$$J = (\boldsymbol{z}_1 - \boldsymbol{z}_d)^2 + (\boldsymbol{z}_2 - \boldsymbol{z}_d)^2 + \cdots + (\boldsymbol{z}_N - \boldsymbol{z}_d)^2 + \boldsymbol{\tau}_0^2 + \boldsymbol{\tau}_1^2 + \cdots + \boldsymbol{\tau}_{N-1}^2$$

- subject to

$$\boldsymbol{z}_1 = \boldsymbol{z}_0 + \Delta t\, \boldsymbol{f}(t, \boldsymbol{z}_0, \boldsymbol{\tau}_0)$$

$$\boldsymbol{z}_2 = \boldsymbol{z}_1 + \Delta t\, \boldsymbol{f}(t, \boldsymbol{z}_1, \boldsymbol{\tau}_1)$$

$$\boldsymbol{z}_3 = \boldsymbol{z}_2 + \Delta t\, \boldsymbol{f}(t, \boldsymbol{z}_2, \boldsymbol{\tau}_2)$$

$$\vdots$$

$$\boldsymbol{z}_N = \boldsymbol{z}_{N-1} + \Delta t\, \boldsymbol{f}(t, \boldsymbol{z}_{N-1}, \boldsymbol{\tau}_{N-1})$$

MODEL PREDICTIVE CONTROL

Optimal Control

- Minimise

$$(\boldsymbol{z}_1 - \boldsymbol{z}_d)^T(\boldsymbol{z}_1 - \boldsymbol{z}_d) + (\boldsymbol{z}_2 - \boldsymbol{z}_d)^T(\boldsymbol{z}_2 - \boldsymbol{z}_d) + \cdots + (\boldsymbol{z}_N - \boldsymbol{z}_d)^T(\boldsymbol{z}_N - \boldsymbol{z}_d) +$$
$$\boldsymbol{\tau}_0^T\boldsymbol{\tau}_0 + \boldsymbol{\tau}_1^T\boldsymbol{\tau}_1 + \cdots + \boldsymbol{\tau}_{N-1}^T\boldsymbol{\tau}_{N-1}$$

- subject to

$$\boldsymbol{z}_1 = \boldsymbol{z}_0 + \Delta t\, \boldsymbol{f}(t, \boldsymbol{z}_0, \boldsymbol{\tau}_0)$$
$$\boldsymbol{z}_2 = \boldsymbol{z}_1 + \Delta t\, \boldsymbol{f}(t, \boldsymbol{z}_1, \boldsymbol{\tau}_1)$$
$$\boldsymbol{z}_3 = \boldsymbol{z}_2 + \Delta t\, \boldsymbol{f}(t, \boldsymbol{z}_2, \boldsymbol{\tau}_2)$$
$$\vdots$$
$$\boldsymbol{z}_N = \boldsymbol{z}_{N-1} + \Delta t\, \boldsymbol{f}(t, \boldsymbol{z}_{N-1}, \boldsymbol{\tau}_{N-1})$$

MODEL PREDICTIVE CONTROL

Optimal Control

- Minimise

$$(\boldsymbol{z}_1 - \boldsymbol{z}_d)^T \boldsymbol{Q}(\boldsymbol{z}_1 - \boldsymbol{z}_d) + (\boldsymbol{z}_2 - \boldsymbol{z}_d)^T \boldsymbol{Q}(\boldsymbol{z}_2 - \boldsymbol{z}_d) + \cdots + (\boldsymbol{z}_N - \boldsymbol{z}_d)^T \boldsymbol{Q}(\boldsymbol{z}_N - \boldsymbol{z}_d) +$$
$$\boldsymbol{\tau}_0^T \boldsymbol{R} \boldsymbol{\tau}_0 + \boldsymbol{\tau}_1^T \boldsymbol{R} \boldsymbol{\tau}_1 + \cdots + \boldsymbol{\tau}_{N-1}^T \boldsymbol{R} \boldsymbol{\tau}_{N-1}$$

- subject to

$$\boldsymbol{z}_1 = \boldsymbol{z}_0 + \Delta t\, \boldsymbol{f}(t, \boldsymbol{z}_0, \boldsymbol{\tau}_0)$$
$$\boldsymbol{z}_2 = \boldsymbol{z}_1 + \Delta t\, \boldsymbol{f}(t, \boldsymbol{z}_1, \boldsymbol{\tau}_1)$$
$$\boldsymbol{z}_3 = \boldsymbol{z}_2 + \Delta t\, \boldsymbol{f}(t, \boldsymbol{z}_2, \boldsymbol{\tau}_2)$$
$$\vdots$$
$$\boldsymbol{z}_N = \boldsymbol{z}_{N-1} + \Delta t\, \boldsymbol{f}(t, \boldsymbol{z}_{N-1}, \boldsymbol{\tau}_{N-1})$$

MODEL PREDICTIVE CONTROL

Optimal Control

- Minimise

$$J = \sum_{i=0}^{N-1} (\boldsymbol{z}_{i+1} - \boldsymbol{z}_d)^T \boldsymbol{Q} (\boldsymbol{z}_{i+1} - \boldsymbol{z}_d) + \boldsymbol{\tau}_i^T \boldsymbol{R} \boldsymbol{\tau}_i$$

- subject to

$$\boldsymbol{z}_{k+1} = \boldsymbol{z}_k + \Delta t\, \boldsymbol{f}(t, \boldsymbol{z}_k, \boldsymbol{\tau}_k) \quad \text{for} \quad k = 0,1,2,\ldots, N-1$$

Decision variables: $\boldsymbol{\tau}_i$

REINFORCEMENT-BASED CONTROL

DDPG

- Bellman's equation

$$Q(\boldsymbol{s}, \boldsymbol{a}) = R(\boldsymbol{s}, \boldsymbol{a}) + \gamma \max_{\boldsymbol{s}'} Q(\boldsymbol{s}', \boldsymbol{a}')$$

- Reward function

$$R(\boldsymbol{s}, \boldsymbol{a}) = -(\boldsymbol{s} - \boldsymbol{s}_d)^T(\boldsymbol{s} - \boldsymbol{s}_d) - \lambda_{\text{torque}} \boldsymbol{a}^T \boldsymbol{a}$$

REINFORCEMENT-BASED CONTROL

DDPG

- Bellman's equation

$$Q(\boldsymbol{s},\boldsymbol{a}) = R(\boldsymbol{s},\boldsymbol{a}) + \gamma \max_{\boldsymbol{s}'} Q(\boldsymbol{s}',\boldsymbol{a}')$$

- Reward function

$$R(\boldsymbol{s},\boldsymbol{a}) = -(\boldsymbol{s}-\boldsymbol{s}_d)^T(\boldsymbol{s}-\boldsymbol{s}_d) - \lambda_{\text{torque}}\boldsymbol{a}^T\boldsymbol{a}$$

$$\boldsymbol{s} = \begin{Bmatrix} \boldsymbol{z} \\ \boldsymbol{z}_d \end{Bmatrix}, \boldsymbol{a} = \boldsymbol{\tau}$$

REINFORCEMENT-BASED CONTROL

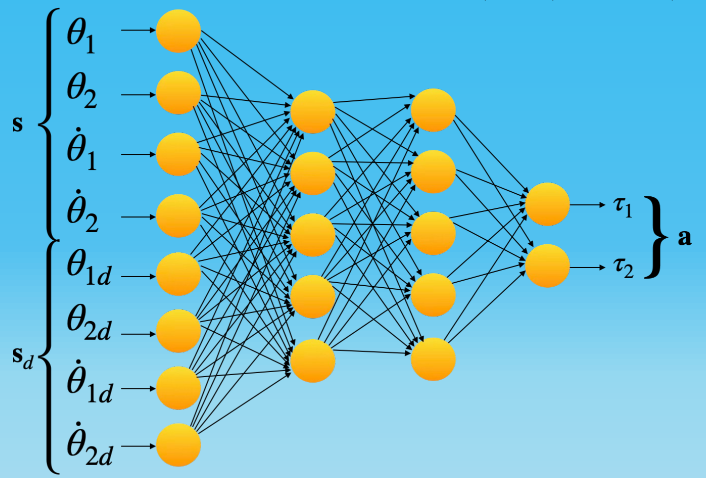
DDPG

- Bellman's equation

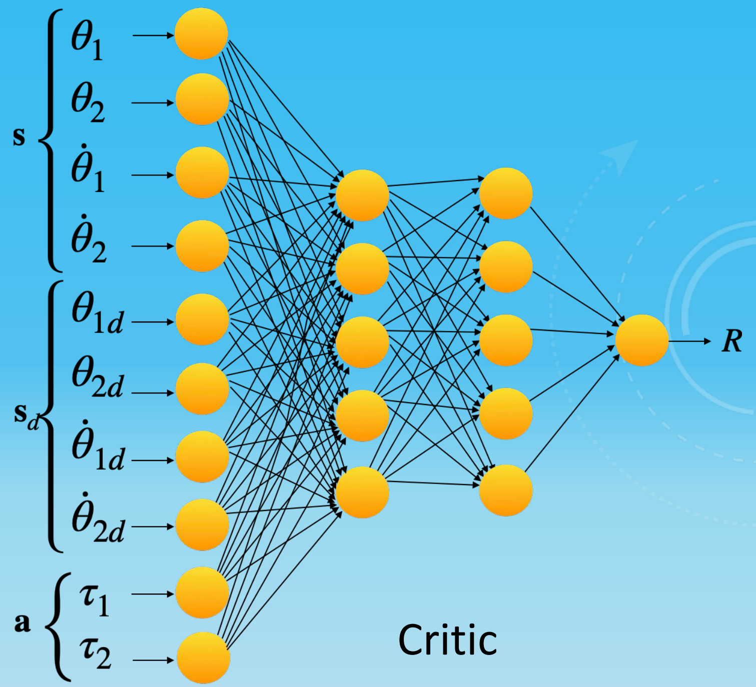
$$Q(\boldsymbol{s},\boldsymbol{a}) = R(\boldsymbol{s},\boldsymbol{a}) + \gamma \max_{\boldsymbol{s}'} Q(\boldsymbol{s}',\boldsymbol{a}')$$

- Reward function

$$R(\boldsymbol{s},\boldsymbol{a}) = -(\boldsymbol{s}-\boldsymbol{s}_d)^T(\boldsymbol{s}-\boldsymbol{s}_d) - \lambda_{\text{torque}}\boldsymbol{a}^T\boldsymbol{a}$$



Actor

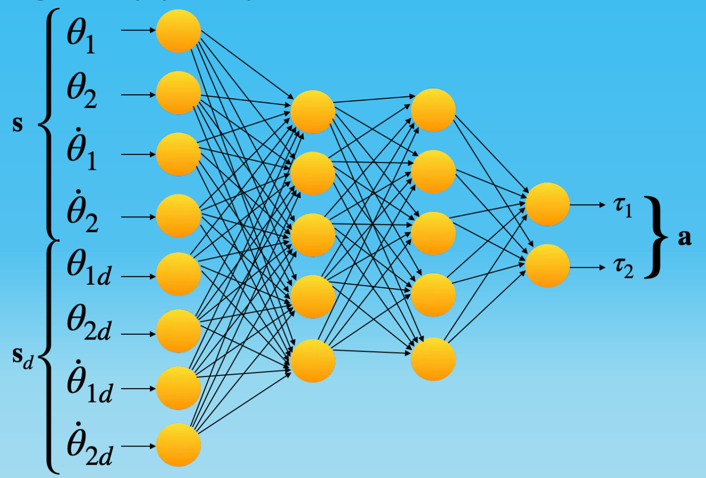


Critic

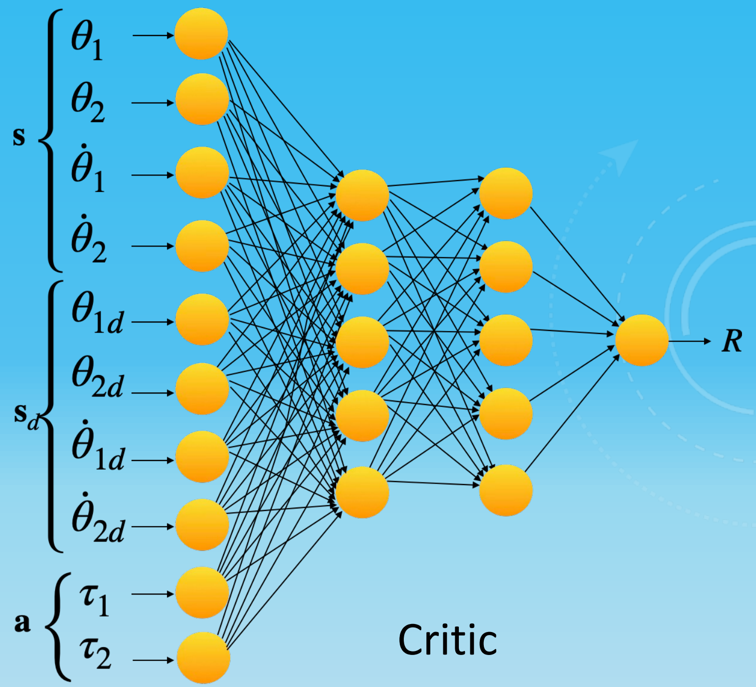
REINFORCEMENT-BASED CONTROL

DDPG

- Actor loss function: $-Q(\boldsymbol{s}, \boldsymbol{s_d}, \boldsymbol{a})$
- Critic loss function: $R(\boldsymbol{s}, \boldsymbol{a}) + \gamma \max_{\boldsymbol{s'}} Q(\boldsymbol{s'}, \boldsymbol{a'}) - Q(\boldsymbol{s}, \boldsymbol{a})$
- Replay buffer row: $[\boldsymbol{s} \quad \boldsymbol{a} \quad \boldsymbol{s}' \quad \boldsymbol{R} \quad \text{done}]$
- ϵ-greedy policy for actor network actions vs randomised actions



Actor

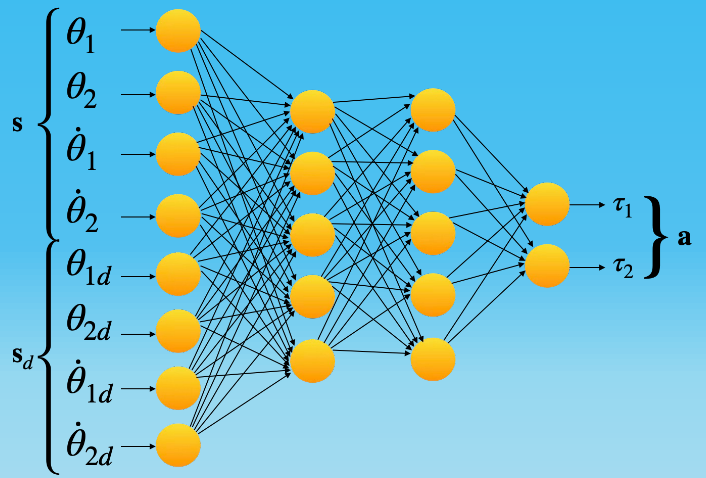


Critic

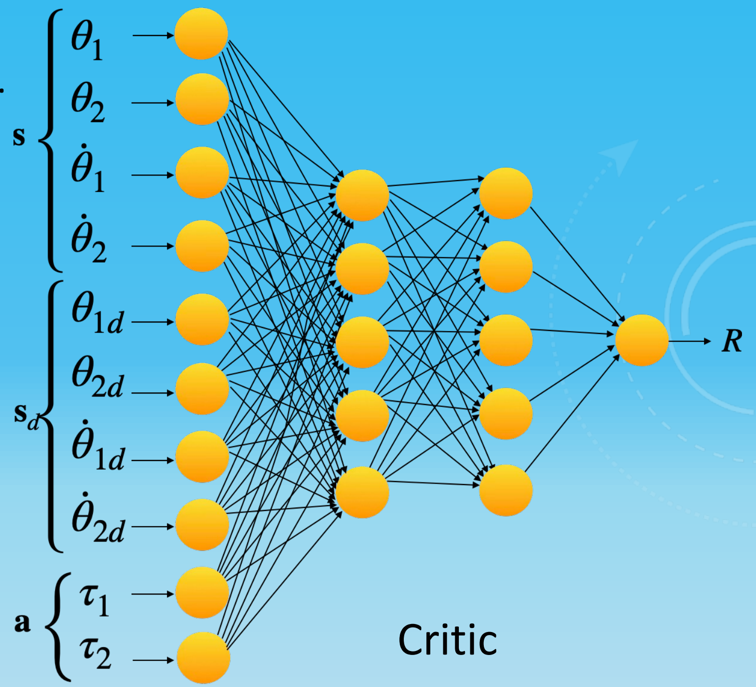
REINFORCEMENT-BASED CONTROL

DDPG

- Generation of actions
  - After training, use the actor network with the initial state and the desired state to generate actions for the next step.
  - Find the next state using differential equation and use this as the new input state.
  - Again find the actions using the first step, and repeat until the goal state is reached.



Actor



Critic

Queries?

Thank you!