Lecture 13 11/10/22 ME221

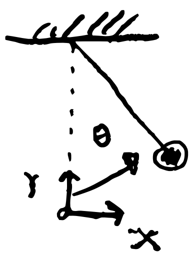
Last time:

- Lagrangian Mechanics

$$\mathcal{L} = T - V$$

$$\frac{d}{dt}\frac{\partial \mathcal{L}}{\partial \dot{q}} - \frac{\partial \mathcal{L}}{\partial q} = F_{ext} \quad *$$

Pendulum



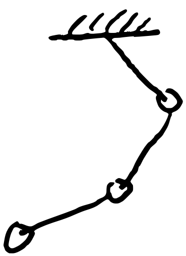
$$\underline{\tau = m l^2 \ddot{\theta} + m g l \sin\theta}$$

Today :

(1) learn how to simulate in Matlab

(2) Intro a few cocepts from Control Theory

A comment on multibody problems in Mechanics :



For multibody problem :

$$\mathcal{L} = \sum_i^n T_i - \sum_i^n V_i$$

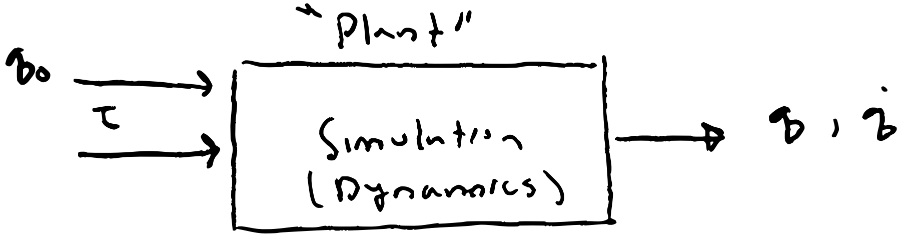
i: each link or body.

$$\frac{d}{dt}\begin{bmatrix} \frac{\partial \mathcal{L}}{\partial \dot{q}_1} \\ \vdots \\ \frac{\partial \mathcal{L}}{\partial \dot{q}_n} \end{bmatrix} - \begin{bmatrix} \frac{\partial \mathcal{L}}{\partial q} \\ \vdots \\ \frac{\partial \mathcal{L}}{\partial q} \end{bmatrix} = \begin{bmatrix} F_1 \\ \vdots \\ F_n \end{bmatrix}$$

system of Eq. with order n.

How to simulate the pendulum dynamics?

simulate → forward dynamics



Equations of motion:

$$\ddot\theta = \left(\frac{1}{ml^2}\right)\tau - g/l \sin\theta$$

all external torques, including control, damping, ext.
we use u

$$\underbrace{\dot x = f(x, u)}_{\text{system of first order Diff. Eqs.}}$$

this is the form we like!!

$$\ddot{\theta} = \left(\frac{1}{m\ell^2}\right)\tau - g/\ell \sin\theta$$

$\dot{x} = f(x,u)$

$$\underbrace{x = \begin{bmatrix} \theta \\ \dot{\theta} \end{bmatrix}}_{\text{state}} \Rightarrow \dot{x} = \frac{d}{dt}\begin{bmatrix} \theta \\ \dot{\theta} \end{bmatrix} = \underbrace{\begin{bmatrix} \dot{\theta} \\ \left(\frac{1}{m\ell^2}\right)\tau - g/\ell \sin\theta \end{bmatrix}}_{\text{dynamics}}$$

How to write this function in matlab?

```
% parameters
m = 1;
g = -9.81;
l = 1;
```

remember $x = \begin{bmatrix} x(1) \\ x(2) \end{bmatrix} = \begin{bmatrix} \text{pos} \\ \text{vel} \end{bmatrix} = \begin{bmatrix} \theta \\ \dot{\theta} \end{bmatrix}$

state

```
f =@(t, x, u) [ x(2); (1/(m*l^2))*u - g/l*sin(x(1)) ]
```

this type of function is called an "anonymous function"

* you can used the standard Function in matlab

Now we have of Equation of motion. What to do next?

Simulate

```
% Time Vector
dt = 1/1000;
T = 10;
t = 0:dt:(T-dt)

% Simulate
x0 = [3π/4 ; 0];
[t, x] = ode45(@(t,x) f(t,x,0), t, x0);
```

$$
\begin{bmatrix} t_1 \\ \vdots \\ t_n \end{bmatrix}, \quad
\begin{array}{cc} \text{Pos} & \text{vel} \end{array}
\begin{bmatrix} x_1(t_1), & x_2(t_1) \\ \vdots & \vdots \\ x_1(t_n), & x_2(t_n) \end{bmatrix}
$$

$\dot{x} = f(x,u)$

How to add damping?

→ add external forces (including control) through input variable u

```
u = @(x) -b·x(2);
```

```
(t,x) = ode45( @(t,x) f(t,x,u(x)), t, x0 )
```

How to control your robot (pendulum for now)?

What is control?

$x = q = \begin{Bmatrix} q_1 \\ \vdots \\ q_n \end{Bmatrix}$

choose $u = \tau = \begin{Bmatrix} \tau_1 \\ \vdots \\ \tau_n \end{Bmatrix}$ s.t. $\lim_{t \to \infty} x(t) = x_d(t)$

$x_d = q_d = \begin{Bmatrix} q_{d1} \\ \vdots \\ q_{dn} \end{Bmatrix}$

$\lim_{t \to \infty} \| \underbrace{x_d - x}_{\text{vectors!}} \| = 0 \Rightarrow \lim_{t \to \infty} \| e \| = 0$

$e = x_d - x$

$e_1 = x_{d_1} - x_1$

$e_2 = x_{d_2} - x_2$

$\vdots$

$\dot{x} = f(x, u)$

Simplest Control: Open loop.

$$\dot{x} = f(x, u)$$

$$x_d \longrightarrow \boxed{\text{Controller}} \xrightarrow{u} \underset{\text{"Plant"}}{\boxed{\text{Dynamics}}} \longrightarrow x$$

equivalent using our robot notation:

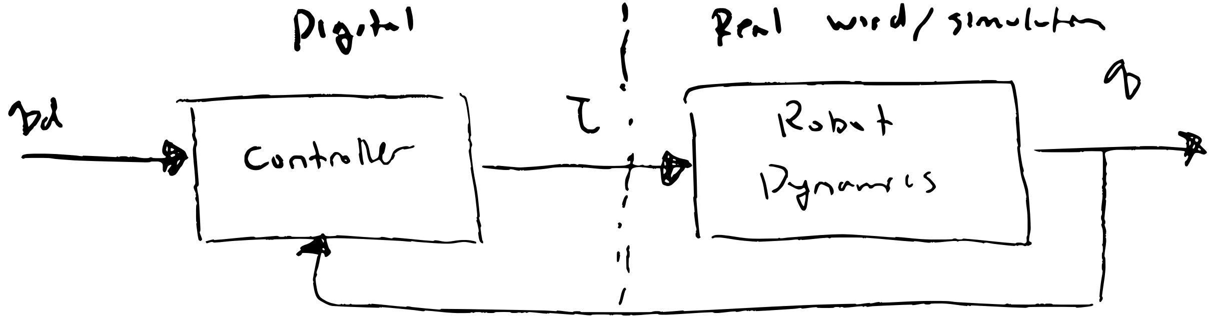
$$q_d \longrightarrow \boxed{\text{Controller}} \xrightarrow{\tau} \boxed{\text{Dynamics}} \longrightarrow q$$

Feedback Controller.

Why feedback?

$\longrightarrow$ unless you have a perfect model, feedback is necessary for performance

Main idea: use feedback to determine you input signal.



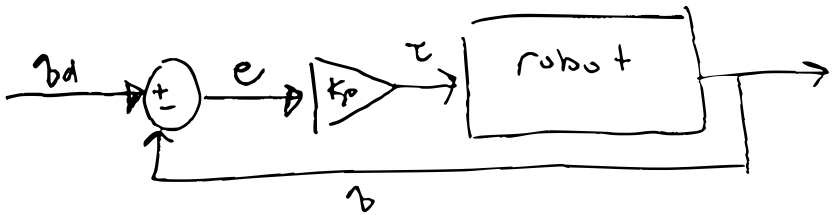
Proportional Control

main idea scale input with error

$u \sim e$

$\tau \sim e$

$e_i = q_{d,i} - q_i$



$$\tau = \begin{Bmatrix} \tau_1 \\ \vdots \\ \vdots \\ \tau_n \end{Bmatrix} = \begin{bmatrix} K_{p_1} & & 0 \\ & \ddots & \\ 0 & & K_{p_n} \end{bmatrix} \begin{bmatrix} q_{d_1} - q_1 \\ q_{d_2} - q_2 \\ \vdots \\ q_{d_n} - q_n \end{bmatrix}$$

$u = c(x)$

$K_p \cdot (x_d - x(1))$

user supplied.