Lecture 14 ME221 11/15/22

Last time:

① Dynamical System

$$\dot{x}(t) = \underbrace{f}_{\text{Equation of motion.}}(\underbrace{x(t)}_{\text{system state}}, \underbrace{u(t)}_{\text{system inputs}})$$

} State-space form

system state

$x = \begin{Bmatrix} x_1 \\ \vdots \\ x_n \end{Bmatrix}$

generalized coordinates

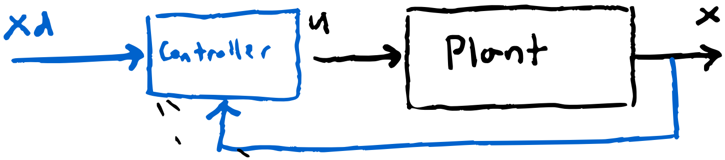
(for robot these are joint variables (q_i))

system inputs

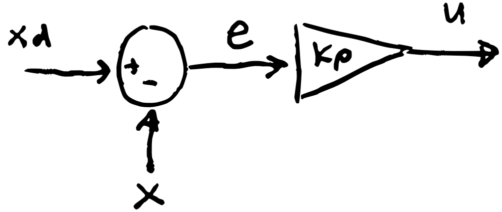
$u = \begin{Bmatrix} u_1 \\ \vdots \\ u_m \end{Bmatrix}$

(external forces)

2. Intro to Control



P- controller

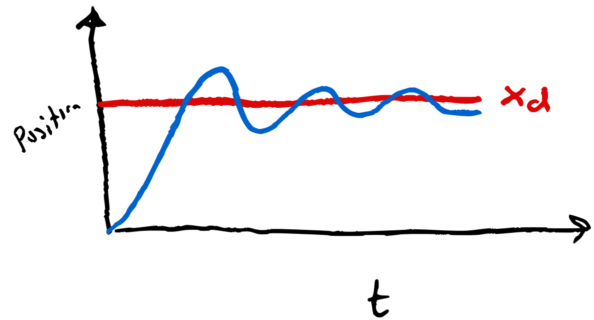


Today:

1. PD & PID controllers
2. Robot Eq. of motion

Limitations of P-Controller

* overshoot / oscillations
* steady state error



PD-Control

$$u_D = K_D \dot{e}$$

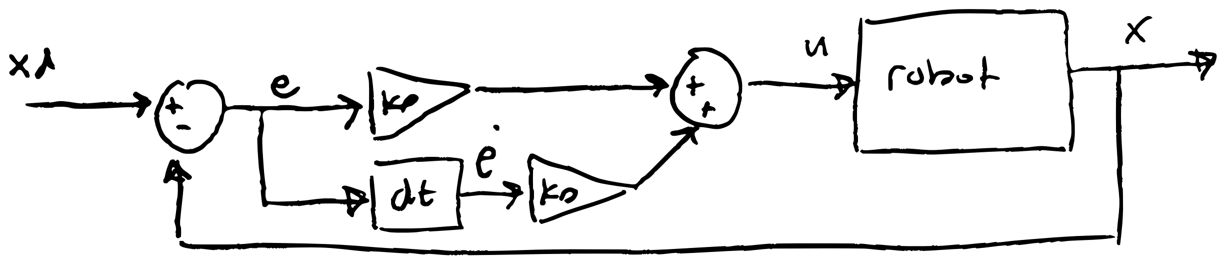
Derivative term basically "predicts" future error ; contributes control signal in proportion.

$$\dot{e} = \frac{d}{dt}(x_d - x)$$

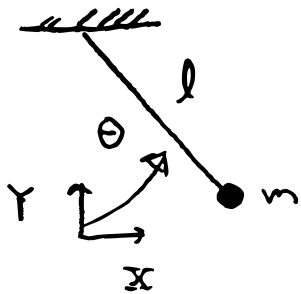
$$= -\dot{x}$$

* x_d doesn't have to be constant. but for our example x_d = constant.

PD- Control Block Diagram



let's implement on our pendulum:



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

$$\Downarrow$$

convert to our canonical form:

$$\dot{x} = \frac{d}{dt}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \frac{d}{dt}\begin{bmatrix} \theta \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} x_2 \\ -g/\ell \sin x_1 + \left(\frac{1}{m\ell^2}\right) u \end{bmatrix}$$

our state

$$x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} \theta \\ \dot{\theta} \end{bmatrix}$$

In matlab

```
% parameters
m = 1;
l = 1;
g = 9.81;
f =@(t, x, u) [ x(2); -g/l sin x(1) + (1/(m l^2)) u ];
```

u : for PD - controller .

$$u = \underbrace{-b\dot{\theta}}_{\text{joint damping}} + \underbrace{K_p(\theta_d - \theta)}_{\text{Proportional control}} + \underbrace{K_D(-\dot{\theta})}_{\text{Derivative control}}$$

In matlab :

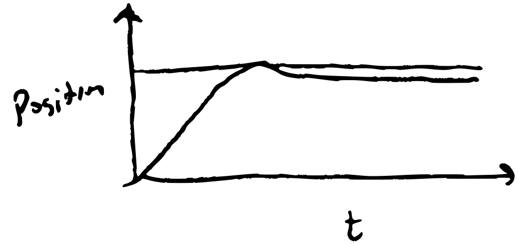
```
% parameters
b = 0.1
Kp = 100;
KD = 50;
xd = pi;

u = @(x) -b*x(2) + Kp(xd-x(1)) + KD(-x(2))

% simulation
[t,x] = ode45(@(t,x) f(t,x,u(x)), t, x0);
```

Major Limitation of PD-Control

* steady state error.

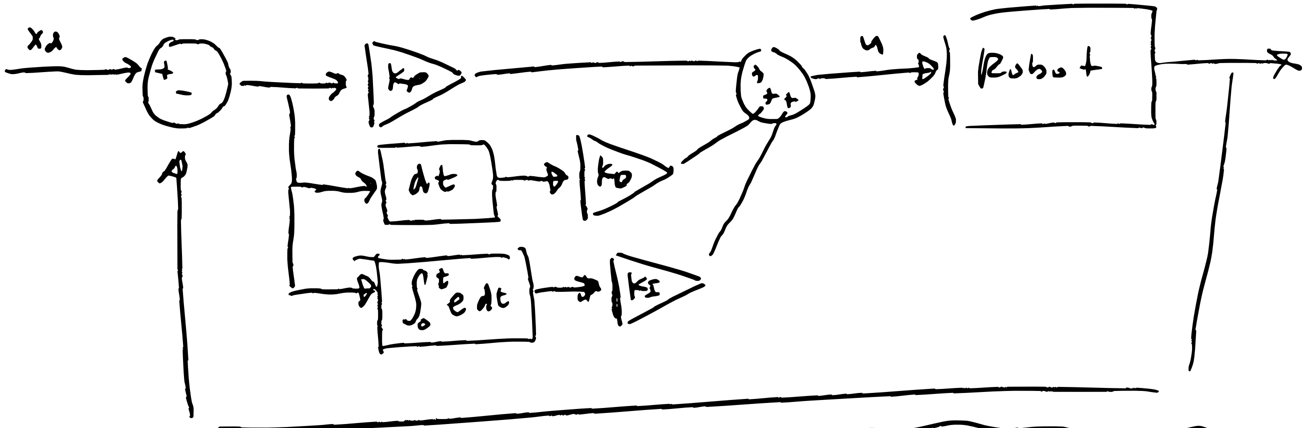


We can introduce an Integral term to fix this:

$$u_I(t) = K_I \int_0^t e(\tau)\, d\tau \, .$$

u_I accumulates the error so the control output is proportional to the total sum of error, which has the effect of eliminating steady state error!

PID - Block Diagram



$$u_I = K_I \cdot \underbrace{\left(\int_0^t e(\tau)d\tau \right)}_{\bar{e}} = K_I \cdot \bar{e}$$

we can add to our state-space!!

$$\bar{e} = \int_0^t e(\tau)d\tau \Rightarrow \dot{\bar{e}} = ? = e \Rightarrow \dot{x} = f(x, u=0)$$

$$\frac{d}{dt}\left[\begin{array}{c}\theta \\ \dot{\theta} \\ \hline \bar{e}\end{array}\right] = \left[\begin{array}{c}\dot{\theta} \\ -g/l \sin\theta + \left(\frac{1}{ml^2}\right) u \\ \hline \theta d - \theta\end{array}\right]$$

In our canonical form

$$\frac{d}{dt}\left[\begin{array}{c}x(1) \\ x(2) \\ \hline x(3)\end{array}\right] = \left[\begin{array}{c}x(2) \\ -g/l \sin(x(1)) + \left(\frac{1}{ml^2}\right) u \\ \hline xd - x(1)\end{array}\right]$$

We only need to <u>append</u> the new state to our old system of equations:

F =@(t, x, u) [f(t, x, u) ; xd - x(1)]

u =@(x) - b x(2) + Kp(xd - x(1)) + KD(- x(2)) + KI (x(3)) ;

Robot Eq. of Motion

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + g(q) = \tau - J^T(q) F_{ext}$$

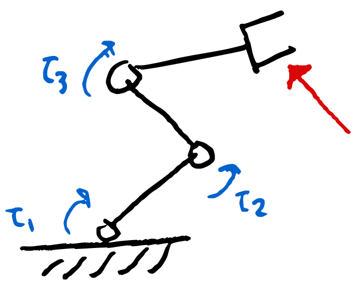
q: generalized coordinates (joint variables)

$M(q)$: Mass matrix

$C(q,\dot{q})$: Centrifugal + Coriolis forces

$g(q)$: gravity vector

$\tau = \begin{Bmatrix} \tau_1 \\ \vdots \\ \vdots \\ \tau_n \end{Bmatrix}$ joint torques



$F_{ext} = \begin{Bmatrix} F_x \\ F_y \\ F_z \\ \tau_x \\ \tau_y \\ \tau_z \end{Bmatrix}$

J: Jacobian

F_{ext}: external wrench.

Build me a dynamic model of your robot, this means find:

$$\boxed{M(q), \quad C(q, \dot{q}), \quad g(q)}$$