EECS 16 B   Module 2: Robotics and Control

Lecture 1.

Gireeja Ranade

ranade @ eecs.

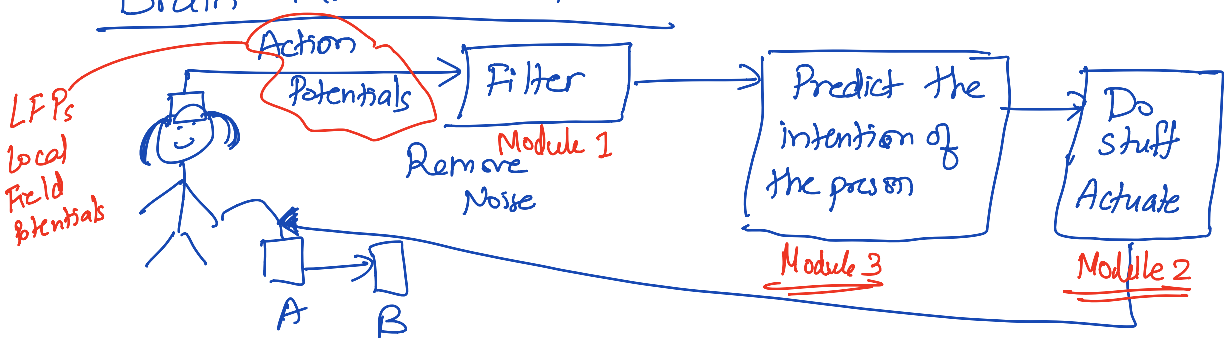
OH: Thursday after class.

Video: 1:35 TRN
2:22 Divert.

- Control
- Feed back.
- Estimation / Sys ID.
- Robustness to noise.

Today:
- Control introduction
- Continuous time and discrete time
  - ↳ Model conversion
- System identification

Brain Machine Interfaces



Real world

"Continuous time"

"Computers"

"Internal clock"

"Discrete time"

0 1 2 3

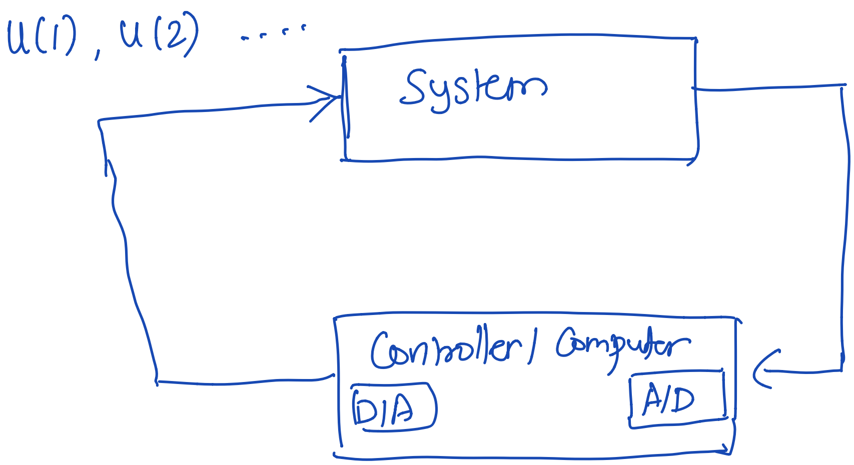
Objects in the physical world : governed by diff. eqn's

$F = ma$

physical state of system.

engineer's choice.

$$\frac{d\,\vec{x}(t)}{dt} = A \cdot \underbrace{\vec{x}(t)}_{\text{"state" of the system.}} + B \cdot \underbrace{\vec{u}(t)}_{\text{"control" input}}$$

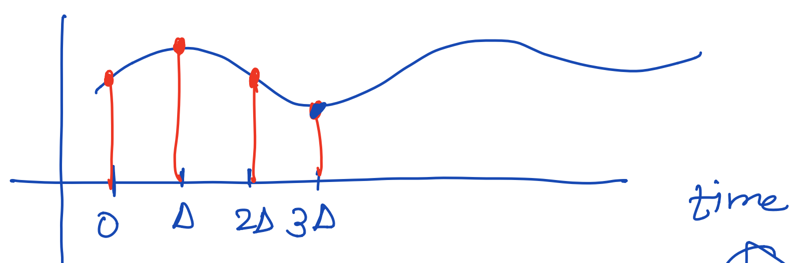


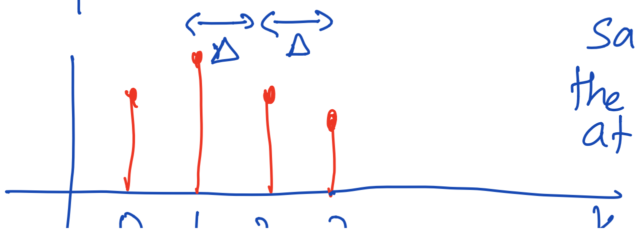
Discrete time

$$\vec{x}_d[k+1] = A_d\, \vec{x}_d[k] + B_d\, \vec{u}_d[k]$$

"Difference equation.

x_d, u_d etc: Discrete versions of my cont. time system.





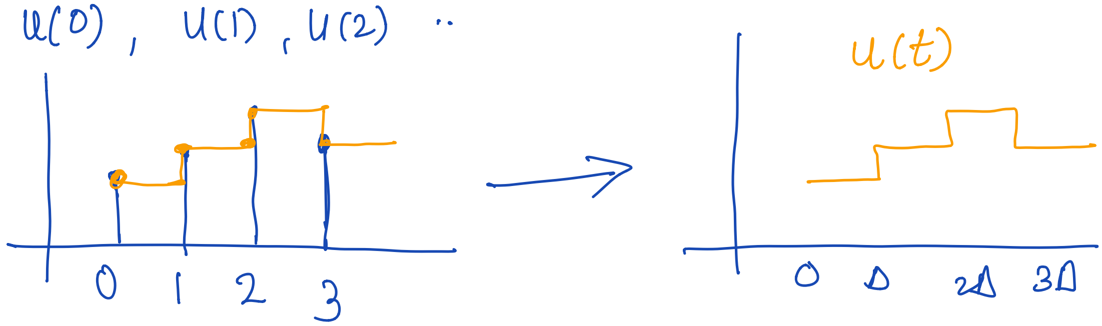
Sample i corresponds to the continuous time system at time $i\Delta$ ("samples")

$\cdot \quad 0 \quad 1 \quad 2 \quad 3 \quad \ldots \quad N \quad \cdot$

How is the input going to work?

Go from discrete input $\longleftrightarrow$ Continuous realworld

$u(0),\ u(1),\ u(2)\ \ldots$



"Piecewise constant"

$$\frac{d}{dt} x(t) = A x(t) + \underline{\underline{B u(t)}}$$

"Piecewise constant"

Going from continuous time to discrete time

Scalar system.

CT: $$\boxed{\frac{d}{dt} x(t) = \lambda \cdot x(t) + b \cdot u(t)} \quad (1)$$

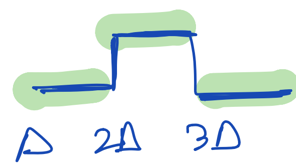
DT: $$x_d[k+1] = \lambda_d \cdot x_d[k] + b_d \cdot u_d[k]$$

Find λ_d, b_d in terms of λ, b

Solution: (1).

$$x(t) = e^{\lambda(t-s)} \cdot x(s) + \int_s^t e^{\lambda(t-\theta)} \cdot b \cdot u(\theta)\, d\theta$$

initial state



$u(t)$ → piecewise constant

open

$u(t) = u_d[i]$ when $t \in (i\Delta, (i+1)\Delta]$

$x(i\Delta) \longleftrightarrow x_d[i]$

$t \in (i\Delta, (i+1)\Delta]$

$$x(t) = e^{\lambda(t-i\Delta)} \cdot x(i\Delta) + \int_{i\Delta}^{t} e^{\lambda(t-\theta)} \cdot b \cdot u(\theta) \cdot d\theta$$

u does not depend on time

$$= e^{\lambda(t-i\Delta)} \cdot x(i\Delta) + b \cdot u_d[i] \cdot e^{\lambda t} \boxed{\int_{i\Delta}^{t} e^{-\lambda\theta}\, d\theta}$$

$$= e^{\lambda(t-i\Delta)} \cdot x(i\Delta) + b \cdot u_d[i] \cdot e^{\lambda t} \cdot \left(\frac{e^{-\lambda\theta}}{-\lambda} \right) \Bigg|_{i\Delta}^{t}$$

$$= \text{———} \,''\, \text{———} + b \cdot u_d[i] \, e^{\lambda t} \frac{\left(e^{-\lambda t} - e^{-\lambda i\Delta} \right)}{-\lambda}$$

$$= e^{\lambda(t-i\Delta)} \cdot x(i\Delta) + b \cdot u_d[i] \frac{\left(e^{\lambda(t-i\Delta)} - 1 \right)}{\lambda}$$

Now: $x(i\Delta) = x_d[i]$

Choose $t = (i+1)\Delta$.

$$\boxed{(i+1)\Delta - i\Delta = \Delta}$$

$$x((i+1)\Delta) = x_d[i+1]$$

$$\boxed{x_d[i+1] = e^{\lambda\Delta} \cdot x_d[i] + b \left(\frac{e^{\lambda\Delta} - 1}{\lambda} \right) u_d[i]}$$

$$\lambda_d = e^{\lambda\Delta} \quad , \quad b_d = \frac{b\cdot(e^{\lambda\Delta}-1)}{\lambda}$$

$\lambda = 0$

$$\frac{d}{dt}x(t) = b\cdot u(t)$$

On your own time

$$x(t) - x(i\Delta) = \int_{i\Delta}^{t} b\cdot u(\theta)\cdot d\theta .$$

$$x_d[i+1] = x_d[i] + b\cdot\Delta\cdot u_d[i]$$

Physical model

Another approach: "Data-centric approach."

Build a model from trying inputs and seeing what happens ???

System ID

$$x_d(k+1) = \lambda_d\cdot x_d(k) + b_d u_d(k) + w(k)$$

Goal: "Learn" λ_d, b_d from data

Unknowns: λ_d, b_d 2 unknowns

$$\begin{bmatrix} \lambda_d \\ b_d \end{bmatrix}$$ Vector of unknowns.

Start system at time $k=0$

$x_d(0) \longrightarrow$ observed.

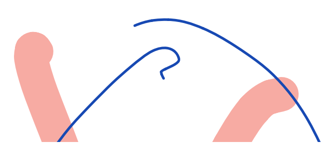
$$x_d[1] = \lambda_d \cdot x_d[0] + b_d \cdot u_d[0] + w[0]$$

$$x_d[2] = \lambda_d \cdot x_d[1] + b_d \cdot u_d[1] + w[1]$$

$$x_d[3] = \lambda_d \, x_d[2] + b_d \, u_d[2] + w[2]$$

$$\vdots$$

Least squares:



$$\underbrace{\begin{bmatrix} x_d[0] & u_d[0] \\ x_d[1] & u_d[1] \\ & \vdots \\ x_d[l] & u_d[l] \end{bmatrix}}_{\substack{D \\ l \times 2}} \underbrace{\begin{bmatrix} \lambda_d \\ b_d \end{bmatrix}}_{\substack{\vec{P} \\ 2 \times 1}} \approx \underbrace{\begin{bmatrix} x_d[1] \\ x_d[2] \\ \\ x_d[l+1] \end{bmatrix}}_{\substack{\vec{S} \\ l \times 1}} \cancel{- \begin{bmatrix} w[0] \\ w[1] \\ \\ w_d[l] \end{bmatrix}}$$

$$\hat{\vec{P}} = (D^T D)^{-1} D^T \vec{S} \quad !$$