Printout

Thursday, September 14, 2023 9:07 AM

What is control theory?

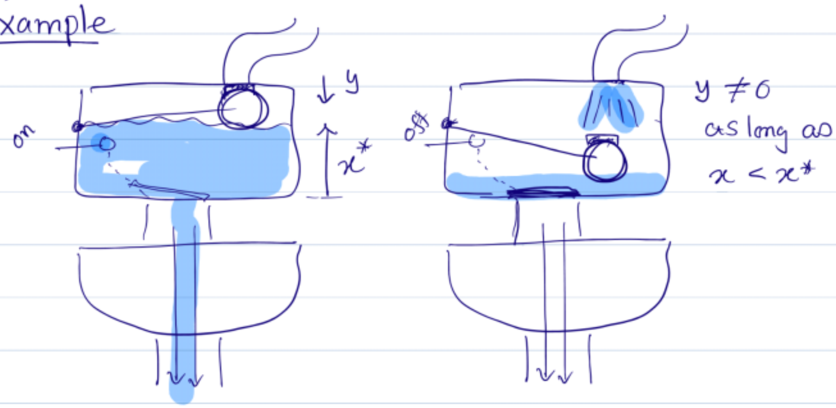
The art of making things do what you want them to do.

art : in this class, parameterized controllers or algorithms and ways of tuning those parameters

things : here : phenomena that can be represented with differential equations

what : in this class, follow some desired set-point, path, or trajectory

Example



Road map

- Models for control (differential equations)
  - → Solutions
- Control design
  - open loop
  - closed loop
  - PID
  - State feedback
  - MPC (mention)
- Requirements - Stability
  - Lyapunov theory Relation to invariance

Modeling , Differential equations

$$\frac{dx}{dt} = f(x(t), u)$$

Continuous time version

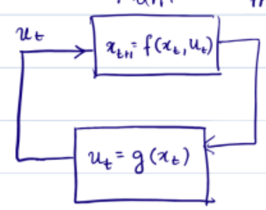
$\dot{x} = f(x, u)$

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

$x \in \mathbb{R}^n$

$f : \mathbb{R}^n \to \mathbb{R}^n$

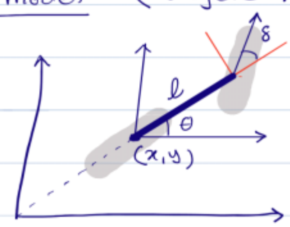
Recall vehicle model from Lecture 1



This could be written as

$$x_{t+1} = f(x_t, g(x_t))$$

Example Vehicle model (bicycle / Kinematic vehicle)



Ref. Brian Paden et al.

Survey of motion planning and control for Self-driving urban 2016

State variables

car pose $\begin{bmatrix} x \\ y \\ \theta \end{bmatrix}$

speed v (constant)

front wheel steering angle δ (control input)

$\dot{x} = v \cos\theta$

$\dot{y} = v \sin\theta$

$\dot{\theta} = \frac{v}{\ell} \tan\delta$

Solutions : Any function $x : \mathbb{R} \to \mathbb{R}^n$ s.t $\frac{dx(t)}{dt} = f(x(t))$

- when do they exist?
- when are they unique?

Example $\dot{x} = x^2$ with $x(0) = 1$

$x(t) = \frac{1}{}$ is a solution (check)

1-t

but as $t \to 1$ $x(t) \to \infty$ blows up

existence of solution is an issue

Example $\dot{x} = \sqrt{x}$ has two solutions

$x(t) = 0$

$x(t) = t^{2/4}$

uniqueness is a problem

Additional condition on f

Def $f : \mathbb{R}^n \to \mathbb{R}^n$ is Lipschitz continuous if $\exists L > 0$ such that for any pair $x, x' \in \mathbb{R}^n$

$$\boxed{\| f(x) - f(x') \| \leq L \| x - x' \|}$$

Example $6x + 4$; $|x|$ are Lipschitz

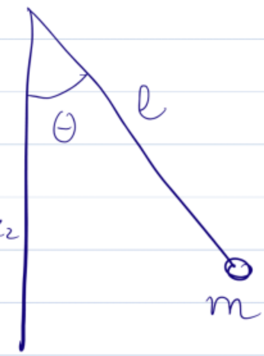
all differentiable functions with bounded derivatives are Lipschitz continuous

Non-Examples $\sqrt{x}$, x^2 are not Lipschitz

Thm if $f(x(t), u(t))$ is Lipschitz continuous in the first argument and $u(\cdot)$ is piece-wise continuous then

$\dot{x} = f(x(t), u(t))$ has unique Solutions

<u>Example</u> Pendulum



$x_1 = \theta \quad x_2 = \dot{\theta}$

$\dot{x}_1 = x_2 \quad \dot{x}_2 = \frac{g}{l} \sin(x_1) - \frac{k}{m} x_2$

g : $9.8\ m/s^2$ on earth

m : mass

$$\begin{bmatrix} \dot{x}_2 \\ x_1 \end{bmatrix} = \begin{bmatrix} -g/l \sin(x_1) - \frac{k}{m} x_2 \\ x_2 \end{bmatrix}$$

When does the pendulum not move?

$\dot{x} = f(x) = 0$ set RHS $= 0$

$x_2 = 0 \quad x_1 = 0, \pi$

<u>Def</u>

These are the <u>equilibria</u> of the system, i.e states x^* such that $f(x^*) = 0$

Recall model – reality gap

As in automata models the gap exists here

tradeoff

more detailed — Complex, intractable

more accurate — harder to analyze

In this class we will focus on Linear ODEs

$$\dot{x} = f(x,u) = A(t)x(t) + B(t)u(t)$$

Any linear function can be represented in this form. $f : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$

$A(t) \in \mathbb{R}^{n \times n}$ a matrix; the entries

$B(t) \in \mathbb{R}^{m \times n}$ can be functions of time t

Linear time varying system (LTV)

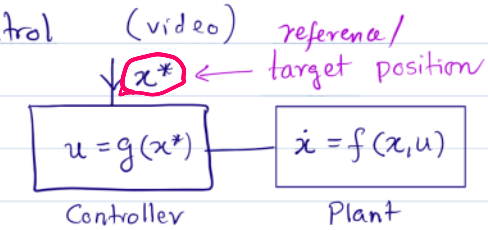
if $A(t)$ $B(t)$ are independent of time

then Linear time Invariant (LTI) System

Control Design

Simple strategy

Open loop control (video)



Does not use the state of the plant (no sensors)

Used in Dryer, Coffee machines, volume control in audio

Emergency stop in vehicle.

input $u(t)$ fixed as a function of the target x^*

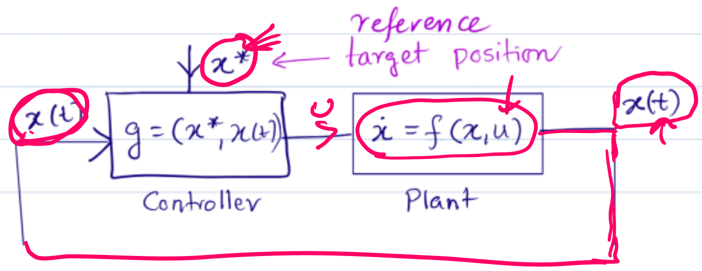
Does not respond to state / environment

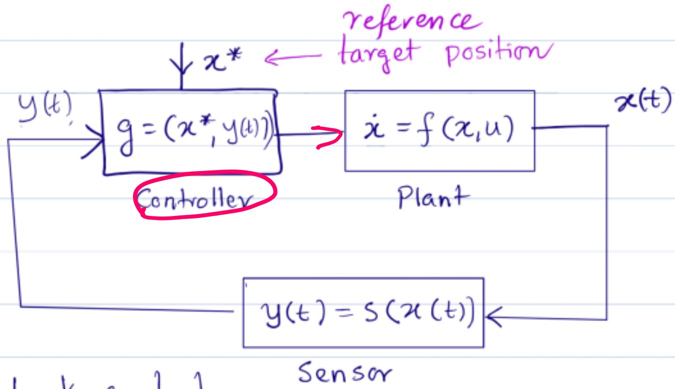
<u>Feedback</u> gives us better control

Closed-loop / Feedback control

$$u(t) = g(x^*, x(t))$$

$$\text{e.g. } u(t) = g(\underbrace{x^* - x(t)}_{\text{error}})$$





Error feedback control

Define the feedback $g(\,)$ as a scaled version of the error

$$u(t) = g(x^*, x(t)) = K \underbrace{(x^* - x(t))}_{\text{error}}$$

Proportional gain

Intuition: if car is really far to the left of the lane then we would move hard to the right; if it is slightly to the left, the we'd want to move to the right less aggresively.

This is also called proportional control (P-Control)

We can get more sophisticated

- g not only depends on error but also the rate of change of error (derivative)

• g also depends on the history of the error (integral)

$$u(t) = k_p e(t) + k_I \int_0^t e(\tau)\,d\tau + k_D \frac{de(t)}{dt}$$

$\boxed{P, I, D}$

P I D

$$= k_p \left[e(t) + \frac{1}{T_I} \int_0^t e(\tau)\,d\tau + T_D \frac{de(t)}{dt} \right]$$

Tuning the gains is an art

Example $\dot{y}(t) = u(t) + d(t)$ ← disturbance

↑ Control input

Using only proportional control

$\boxed{K_p e(t)} \to u(t)$

$$u(t) = -k_p e(t)$$
$$= -k_p (y(t) - y^*)$$
$$\dot{y}(t) = -k_p (y(t) - y^*) + d(t)$$
$$= -k_p y(t) + k_p y^* + d(t)$$

Suppose the steady disturbance $d(t) = d_{ss}$

$$\dot{y}(t) = -k_p y(t) + k_p y^* + d_{ss} \Rightarrow \boxed{K_p Y_{ss}}$$

What is the steady state output?

$$\dot{y} = 0 \Rightarrow -k_p y(t) + k_p y^* + d_{ss} = 0$$
$$y(t) = d_{ss}/k_p + y^* \quad [\text{not } y^*]$$

define $y_{ss} = d_{ss}/k_p + y^*$ $\Rightarrow$ $\boxed{K_p Y_{ss}} = d_{ss} + K_p Y^*$

We can make steady state error small by increasing k_p.

v

What is the transient behavior?

$$\dot{y}(t) = -k_p\, y(t) + K_p\, y_{ss}$$

What is the solution of this ODE

$\dot{x} = -ax$

$x(t) = e^{-at} x(0)$

$\dot{x} = -ax + b$

$$y(t) = y(0)\, e^{-t/T} + y_{ss}\,(1 - e^{-t/T})$$

where $T = 1/k_p$

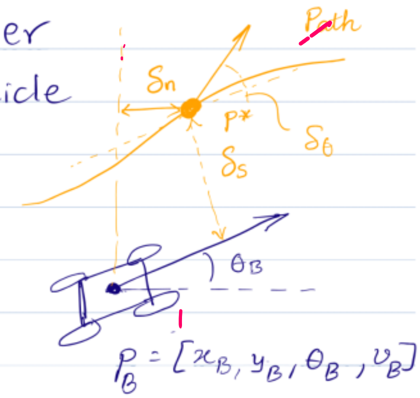
We can make the steady state errors small at the expense of slower convergence / longer transients

Path following controller

Recall bicycle model for vehicle

Consider the state of the vehicle $[x_B, y_B, \theta_B\, v_B] \in \mathbb{R}^4$

Consider a target position p^* on a path (Chosen by a higher level planner)



The error is now a vector

$$e(t) = [\delta_s(t), \delta_n(t), \delta_\theta(t), \delta_v(t)]$$

along track error and cross-track error

Distance ahead or behind the target p^* in the instantaneous direction of motion

$$\delta_s = \cos\theta_B(t).\,(x^*(t) - x_B(t)) + \sin\theta_B(t)\,(y^*(t) - y_B(t))$$

Cross track error : orthogonal to the intended direction of motion

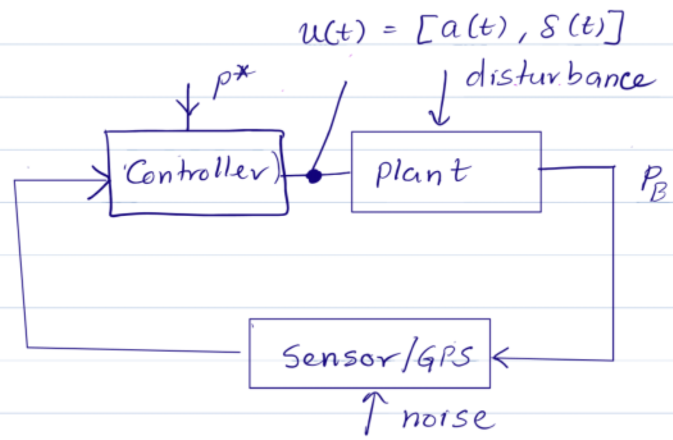
$$\delta_n = -\sin(\theta_B(t))\,(x^*(t) - x_B(t)) + \cos\theta_B(t)\,(y^*(t) - y_B(t))$$

Heading error $\delta_\theta = \theta^*(t) - \theta_B(t)$

Velocity error $\delta_v = v(t) - v_B(t)$

a: throttle

δ: steering



Now you can apply PID or state-feedback control after linearization

Pure-pursuit controller

$$u(t) = K \begin{bmatrix} \delta_s \\ \delta_n \\ \delta_\theta \\ \delta_v \end{bmatrix} \qquad K = \begin{bmatrix} k_s & 0 & 0 & k_v \\ 0 & k_n & k_\theta & 0 \end{bmatrix}$$

This performs PD-Control to correct against along-track error and cross-track error