ECE 484: Principles of Safe Autonomy (Fall 2025)
Lecture 8
Control

Professor: Huan Zhang
https://publish.illinois.edu/safe-autonomy/
https://huan-zhang.com
huanz@illinois.edu

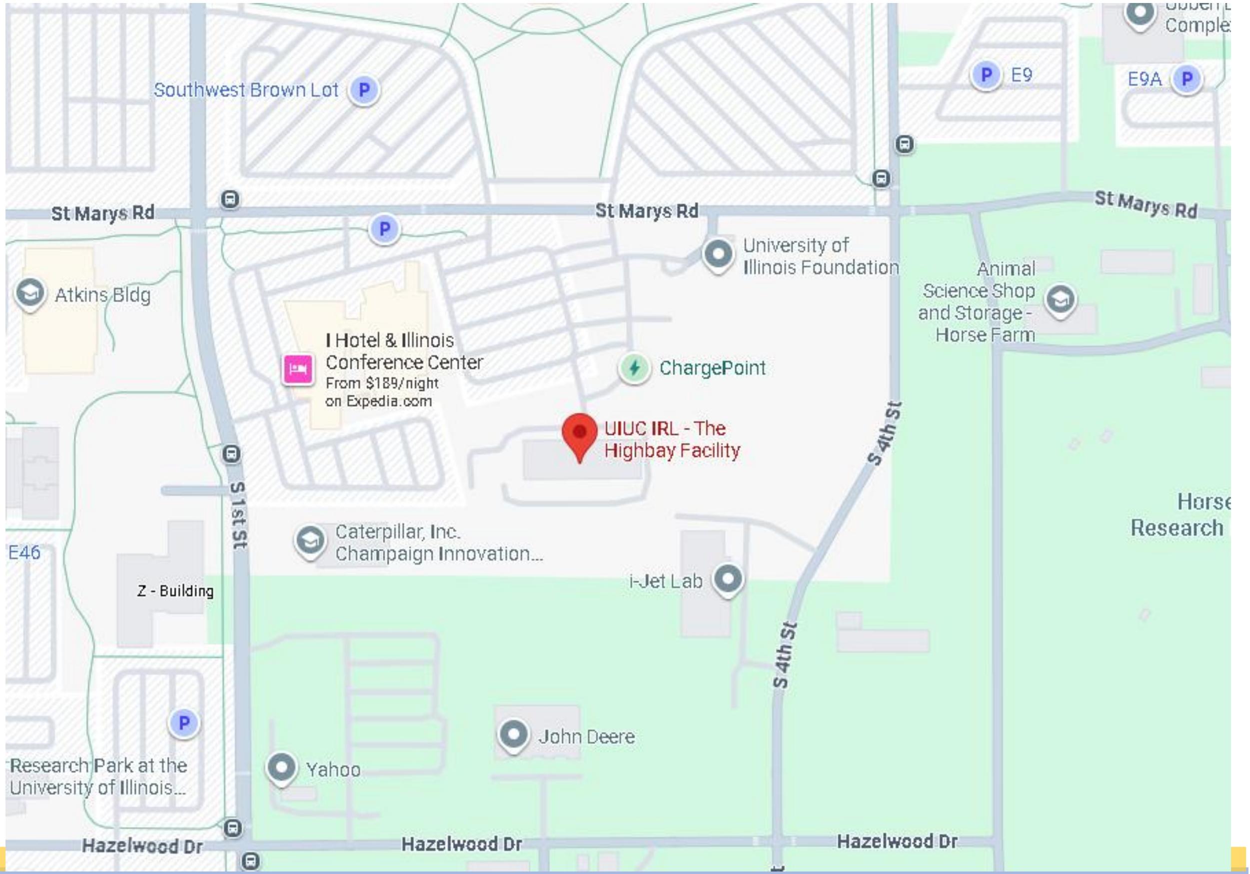
GEM field trip **9/23 11 am (upcoming Tuesday)**

Do not come to the ECEB classroom on **9/23 (next class)**

GEM car field trip at the Highbay facility:

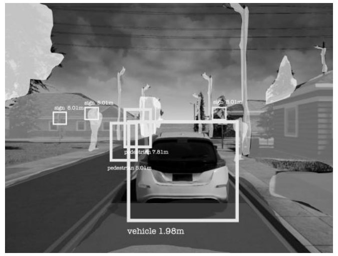
201 St Marys Rd, Champaign, IL 61820

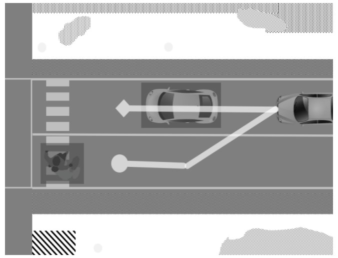
More information on **Campuswire**

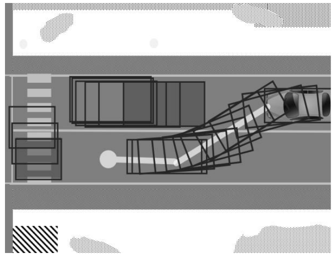


GEM platform

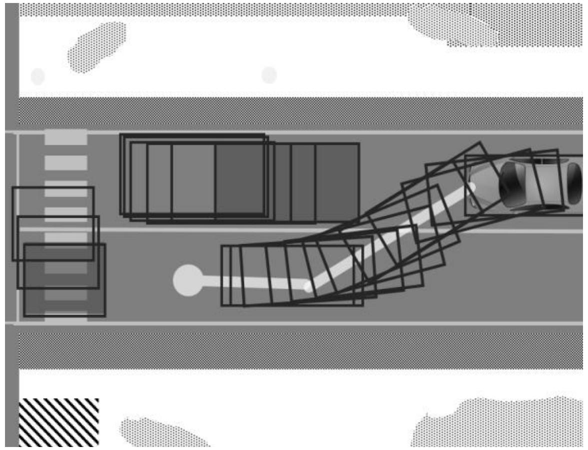
Autonomy pipeline







Sensing	Perception	Decisions and planning	Control
Physics-based models of camera, LIDAR, RADAR, GPS, etc.	Programs for object detection, lane tracking, scene understanding, etc.	Programs and multi-agent models of pedestrians, cars, etc.	Dynamical models of engine, powertrain, steering, tires, etc.

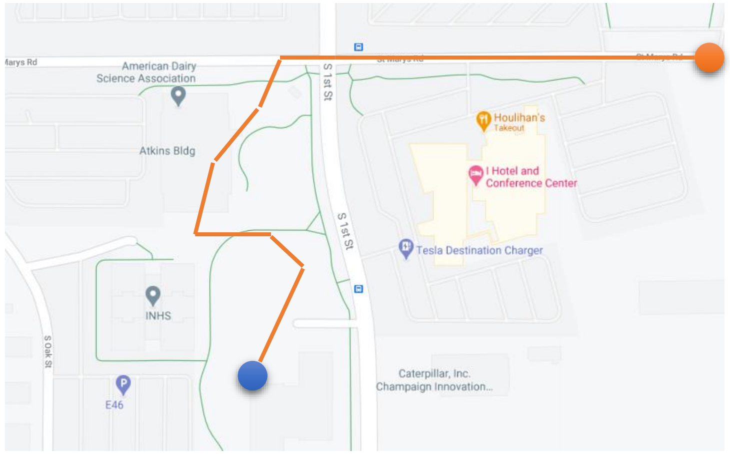


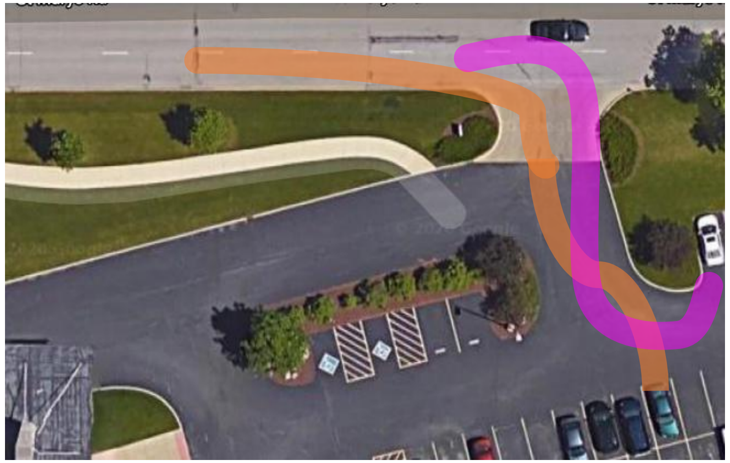
Control

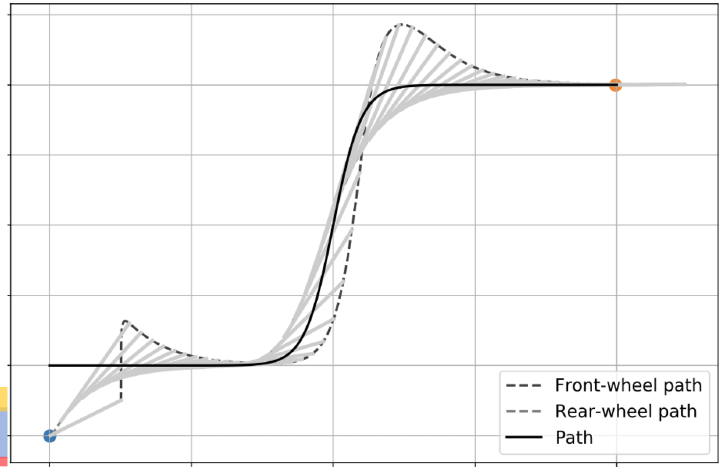
Dynamical models of engine, powertrain, steering, tires, etc.

Typical planning and control modules

- Global navigation and planner
  - Find paths from source to destination with static obstacles
  - Algorithms: Graph search, Dijkstra, Sampling-based planning
  - Time scale: Minutes
  - Output: reference center line, does not consider vehicle dynamics
- Local planner
  - Dynamically feasible trajectory generation
  - Dynamic planning w.r.t. obstacles
  - Time scales: 10 Hz
- Controller
  - Waypoint follower using steering, throttle
  - Algorithms: PID control, MPC, Lyapunov-based controller
  - Lateral/longitudinal control
  - Time scale: 100 Hz







What is control?

Control theory is the *art* of making *things* do *what you want* them to do

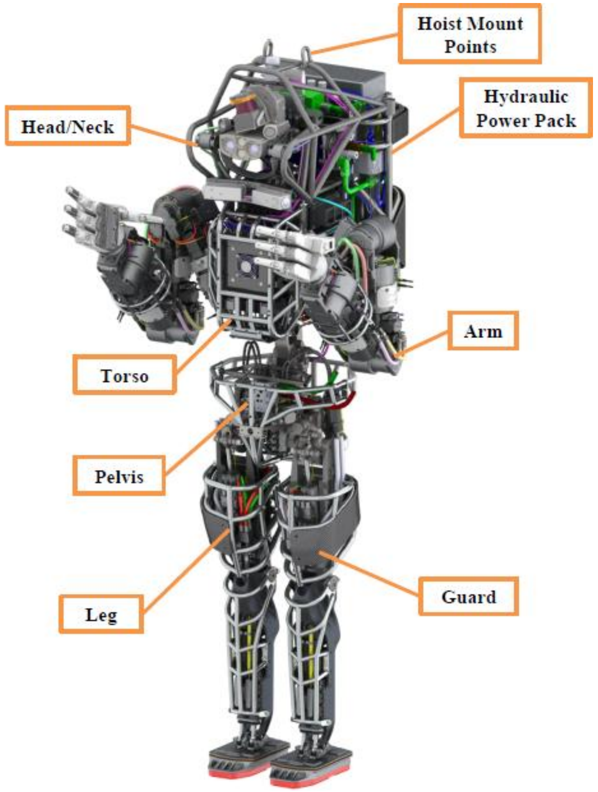
art: tuning or optimizing parameters

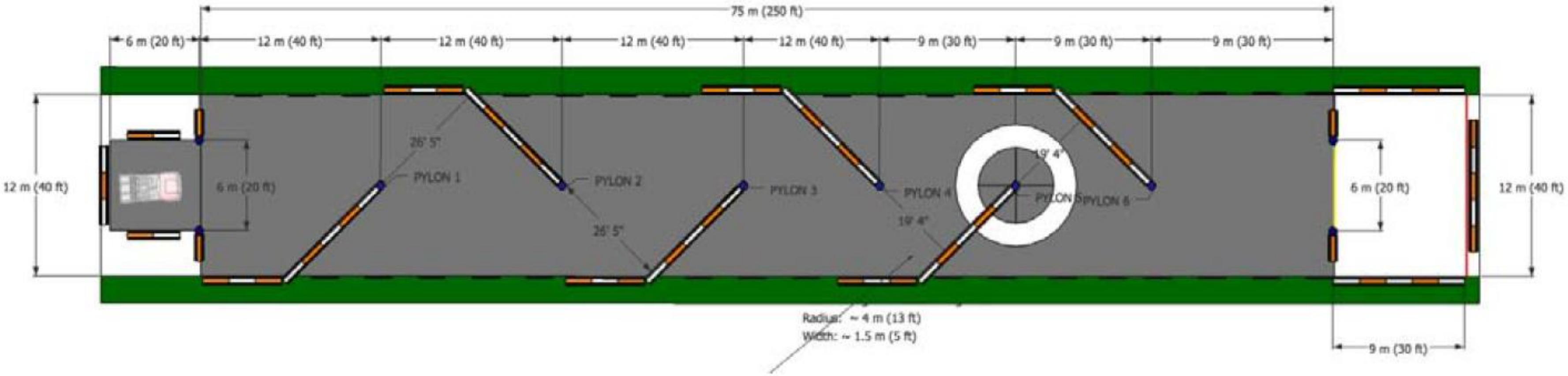
things: Differential equation models

what you want: tracking error or stability

Complex control tasks: DARPA Robotics Challenge

- 4 point task
  - Robot drives the vehicle through the course (1)
  - Robot gets out of the vehicle and travels dismounted out of the end zone (2)
  - Bonus point (1) if the robot completes all tasks without human interventions





https://www.youtube.com/watch?v=bFFMLUDuNCE https://www.youtube.com/watch?v=OesfwU1rsyg

Outline

- Modeling the control problem
  - Differential Equations; solutions and their properties
- Control design
  - PID
  - State feedback
  - MPC (brief)
- Requirements
  - Stability
  - Lyapunov theory and its relation to invariance

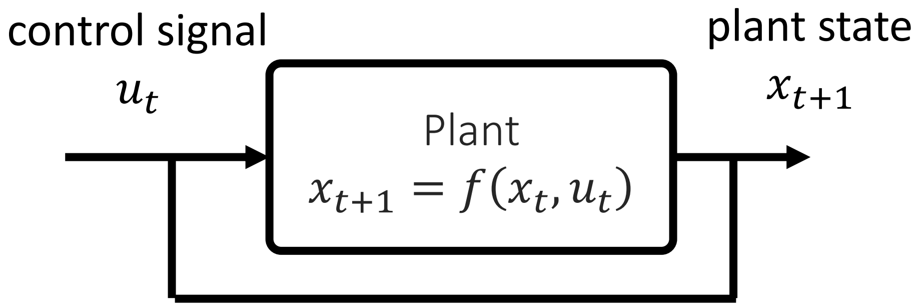
"Thing" being controlled: the Plant

The system we want to control is the **plant**

Plant state at time $t \in \mathbb{R}$ is denoted by a vector $x_t \in \mathbb{R}^n$

The input used to control the plant state is the **control signal** $u_t \in \mathbb{R}^m$

The **dynamics function** $f: \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$ models how the plant state changes in *discrete time* from a previous state x_t under the influence of a control input u_t, that is, $x_{t+1} = f(x_t, u_t)$



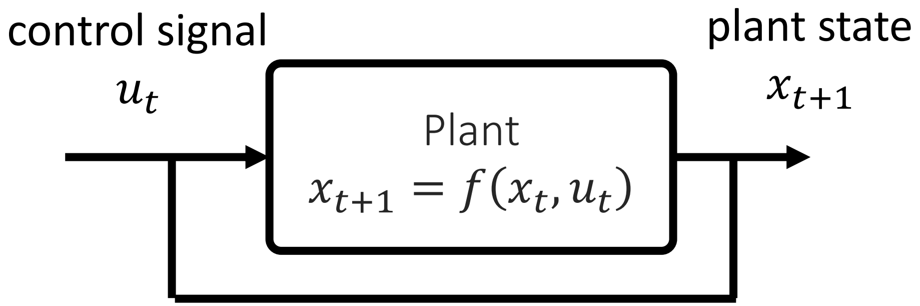
Discrete time mode and Automata

The **dynamics function** $f: \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}^n$ models how the plant state changes in discrete time from a previous state x_t under the influence of a control input u_t, that is, $x_{t+1} = f(x_t, u_t)$

This discrete time model defines a (nondeterministic) automaton

$A = \langle Q = \mathbb{R}^n, Q_0, D \subseteq Q \times Q \rangle$ with $D = \{(x, x') | \exists u \in \mathbb{R}^m \; x' = f(x, u)\}$

Executions of the automaton capture behaviors of the system in discrete time $x_0, x_1, x_2, \ldots$



Continuous time model of a plant

In *continuous time*, dynamics function defines how the the plant state changes continuously with time

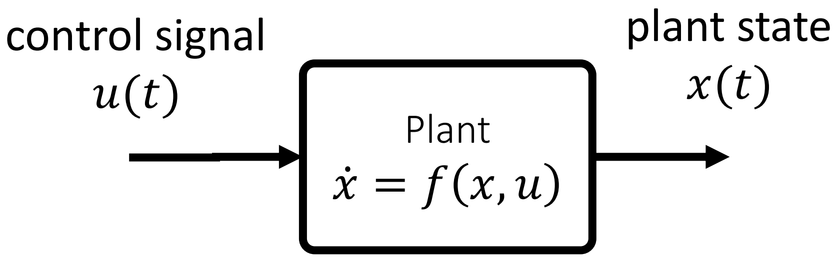
Instead of discrete values of x_t we are interested in how the state changes continuously with time and this is modeled as a function or a signal $x: [0, \infty) \rightarrow \mathbb{R}^n$

Similarly, $u: [0, \infty) \rightarrow \mathbb{R}^m$ models the input signal

Given $x(\cdot)$ and any t, $x(t)$ denotes the state of the system at time t

The Ordinary Differential Equation (ODE) relates the input and the state $\frac{dx(t)}{dt} = f(x(t), u(t))$

This is written in short as $\frac{dx}{dt} = f(x, u)$ or $\dot{x} = f(x, u)$



Example 1: Free swinging pendulum

$x \in \mathbb{R}^2$ x_1: angular position x_2: angular velocity

No input u; such models are called autonomous ODEs

$f: \mathbb{R}^2 \rightarrow \mathbb{R}^2$

$x_2 = \dot{x}_1$

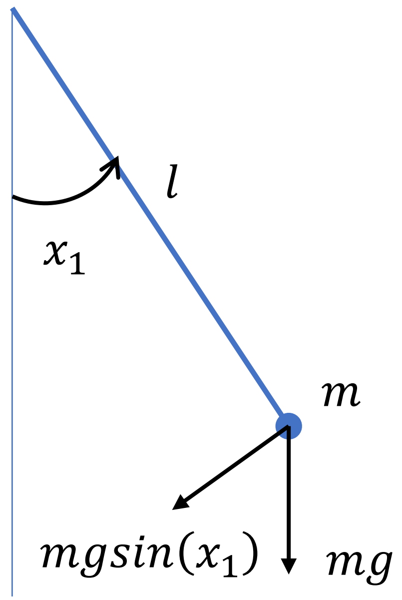
$$\dot{x}_2 = -\frac{g}{l}\sin(x_1) - \frac{k}{m}x_2$$

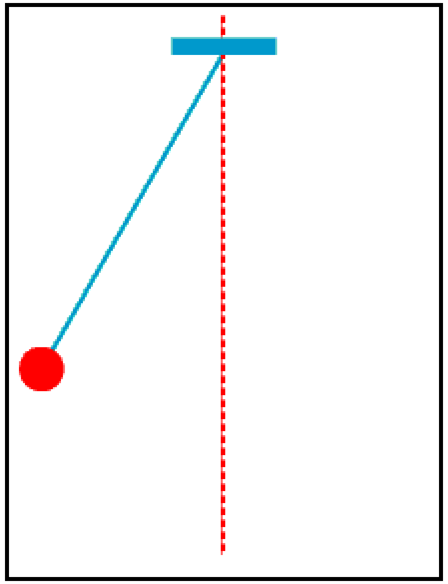
The dynamics equation can be written in vector form:

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

Model parameters

k: friction coefficient m: mass l: length



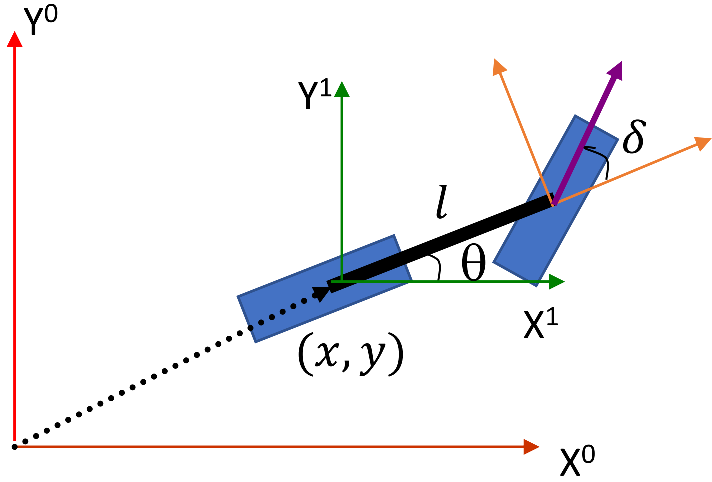


Example 2: Simple vehicle model: Dubin's car

Key assumptions

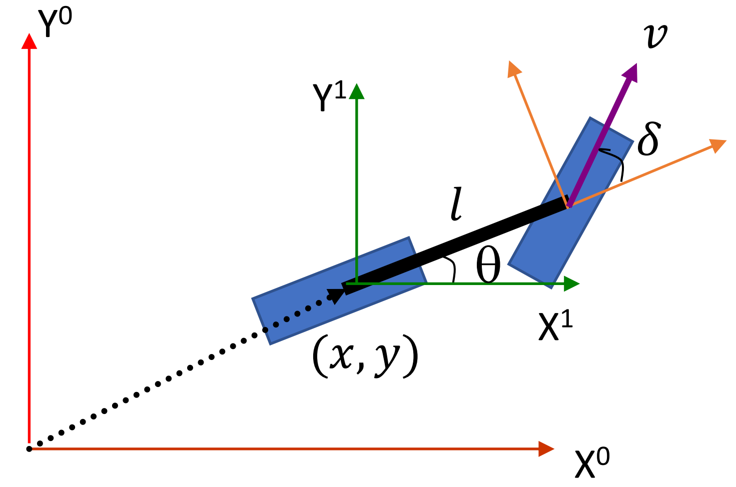
- Front and rear wheel are in vertical planes
- Front wheel moving at a constant speed v
- Steering input, front wheel steering angle δ
- No slip: wheels move only in the direction of the plane they reside in

Modeling one wheel is enough



Reference: Paden, Brian, Michal Cap, Sze Zheng Yong, Dmitry S. Yershov, and Emilio Frazzoli. 2016. A survey of motion planning and control techniques for self-driving urban vehicles. IEEE Transactions on Intelligent Vehicles 1 (1): 33–55.

Rear Wheel Model (Dubin's model)



Plant state: real wheel pose) = $\boldsymbol{x}: \mathbb{R}^3 = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix}$

Control input: front wheel steering angle $u: \mathbb{R}$ = δ

Model parameters: car length (l) speed (v)

$f: \mathbb{R}^4 \rightarrow \mathbb{R}^3$

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

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} v\ cos\theta \\ v\ sin\theta \\ \frac{v}{l}\ tan\delta \end{bmatrix}$$

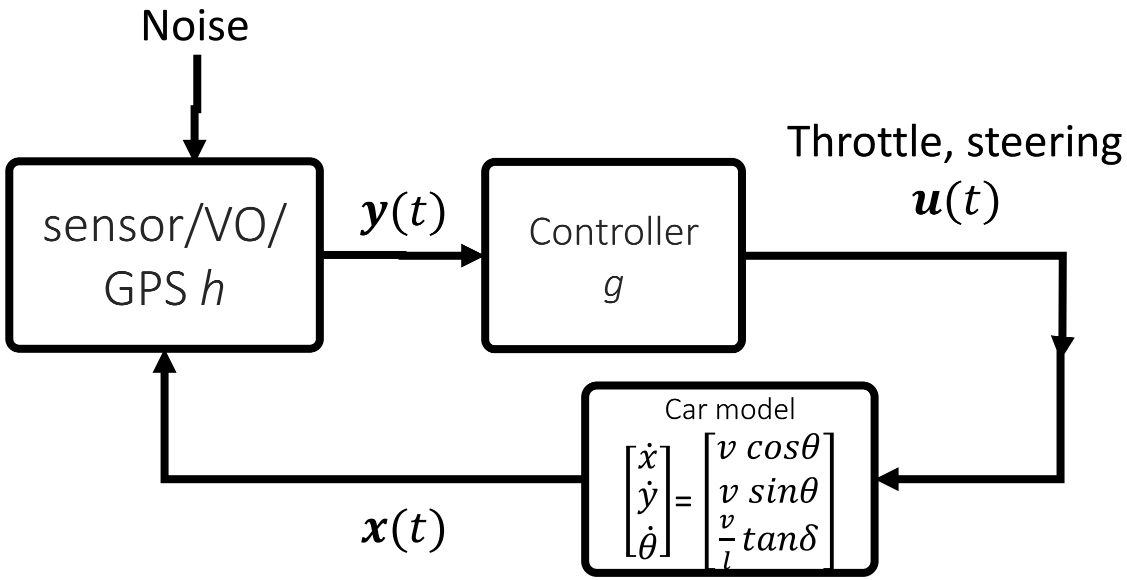
Control problem: Cruise control

The controller $g(\ \ )$ is the function (implemented in software) that computes the control signal $u(t)$ from the state $x(t)$

In practice, the controller will not have direct access to the state but instead will use sensors to get observations $y(t) = h(x(t))$ of the state

$$\dot{\boldsymbol{x}}(t) = f\left(\boldsymbol{x}(t), \boldsymbol{u}(t)\right)$$
$$\boldsymbol{y}(t) = h\left(\boldsymbol{x}(t)\right) + n(t)$$
$$\boldsymbol{u}(t) = g(\boldsymbol{y}(t))$$

Control design is the problem of figuring out g given certain requirements on $x(t)$



Behaviors may not be well-defined

Behaviors of physical processes are described in terms of ODEs

$$\dot{x}(t) \equiv \frac{dx(t)}{dt} = f\big(x(t), u(t)\big) \qquad - \ \text{Eq.}\,(1),$$

where time $t \in \mathbb{R}$; state $x(t) \in \mathbb{R}^n$; $input \;\; u(t) \in \mathbb{R}^m$; $f: \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}^n$

Initial value problem: Given ODE (1) and initial state $x_0 \in \mathbb{R}^n, t_0 \in \mathbb{R}$, and input u: $\mathbb{R} \rightarrow \mathbb{R}^m$, find a state trajectory or *solution* of (1)

Is a solution of (1) always defined?

Solution of an ODE

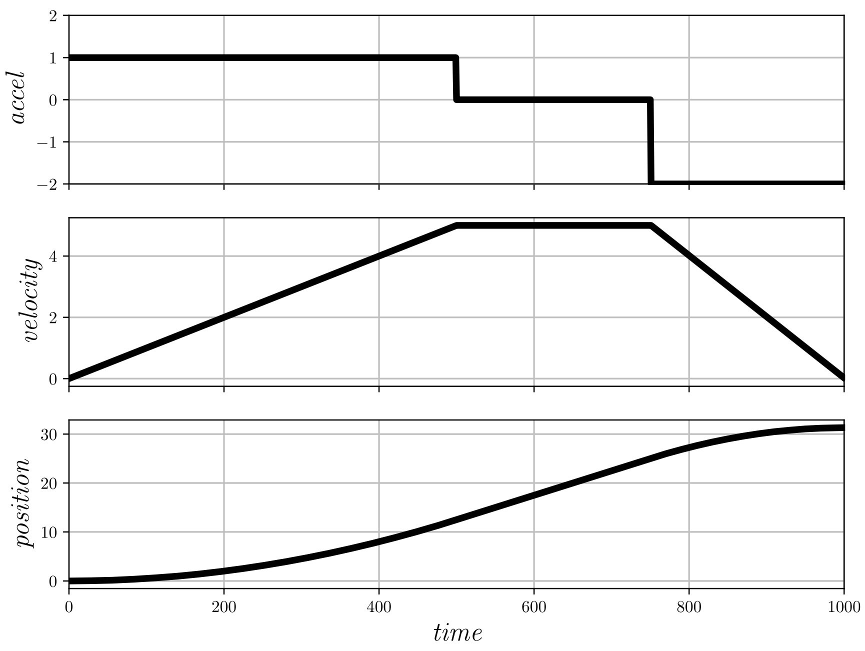
Definition 1. (First attempt) Given x_0 and u, $\xi: \mathbb{R} \to \mathbb{R}^n$ is solution or trajectory iff

(1) $\xi(t_0) = x_0$ and

(2) $\frac{d}{dt}\xi(t) = f(\xi(t), u(t)), \forall t \in \mathbb{R}$.

Mathematically OK, but too restrictive for autonomous systems.

Assumes that ξ is not only continuous, but also differentiable. This disallows $\mathrm{u}(t)$ to be discontinuous, which is often required for optimal control.



Consider the example:

$$\dot{x}(t) = v(t)$$
$$\dot{v}(t) = u(t)$$

and $u(t)$ is the acceleration input. The time points at which $u(t)$ is discontinuous, the solution ξ is not differentiable

Solutions may not exist even for autonomous ODEs

Example. $\dot{x}(t) = -sgn\big(x(t)\big); x_0 = c; t_0 = 0; c > 0$
Solution: $\xi(t) = c - t$ for $t \leq c$; check $\dot{\xi}(t) = -1 = -\,\mathrm{sgn}\big(\xi(t)\big)$
Problem: Solution undefined at $t = c$; f discontinuous in x

Example. $\dot{x}(t) = x^2; x_0 = c; t_0 = 0; c > 0$
Solution: $\xi(t) = \frac{c}{1-tc}$ works for $t < 1/c$; check $\dot{\xi}$
Problem: As $t \rightarrow \frac{1}{c}$ then $\xi(t) \rightarrow \infty$; f grows too fast

We need assumptions on smoothness of $f(.)$ to assure that solutions exist

Lipschitz continuity

A function $f: \mathbb{R}^n \to \mathbb{R}$ is **Globally Lipschitz continuous** if there exist $L > 0$ such that for any pair $x, x' \in \mathbb{R}^n$, $||f(x) - f(x')|| \leq L||x - x'||$

Examples: $6x + 4$; $|x|$ are Lipschitz continuous
All differentiable functions with bounded derivatives are Lipschitz continuous

Exercise: Are Lipschitz continuous functions closed under addition, multiplication?

Non-examples: $\sqrt{x}$; x^2 (locally Lipschitz but not globally Lipschitz)

Dynamical Systems

Describe behavior in terms of instantaneous laws

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

$t \in \mathbb{R}, x(t) \in \mathbb{R}^n, u(t) \in \mathbb{R}^m$

$f: \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}^n$dynamic function

Theorem. If $f(x(t), u(t))$ is Lipschitz continuous in the first argument and $u(t)$ is piece-wise continuous then (1) has unique solutions.

Control design

On-off control of a room heater with a thermostat

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

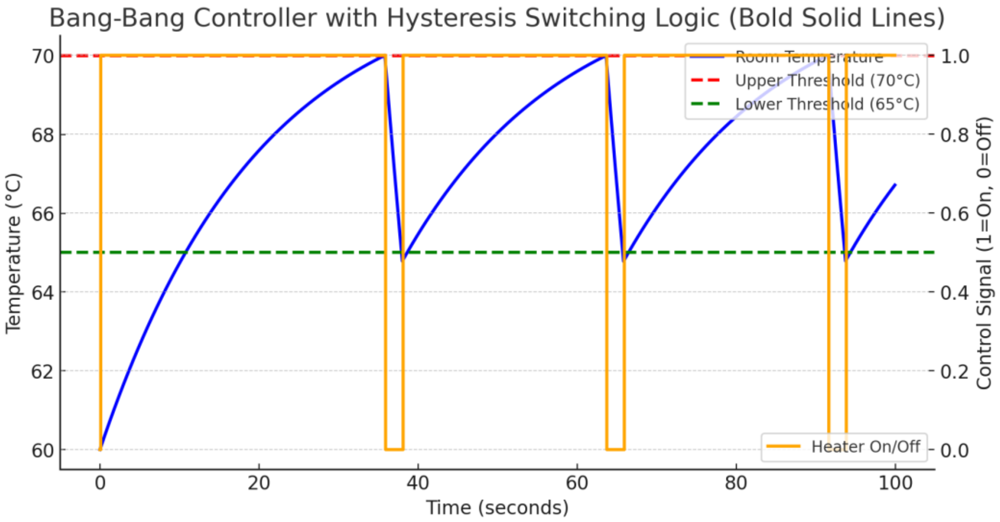
A simple thermostat controller

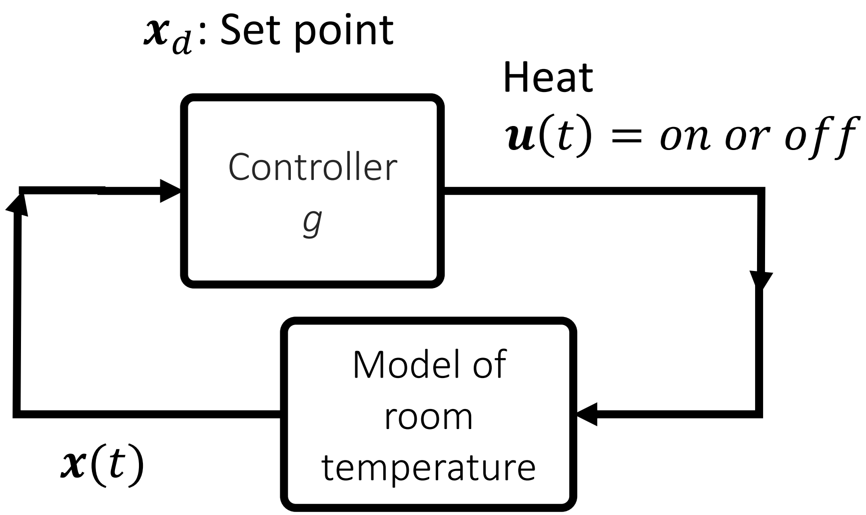
$g(x(t))$:

if $x(t) \geq x_d$ then $u(t)$ = off

else if $x(t) \leq x_d - \varepsilon$ then $u(t)$ = on

This is called bang-bang control



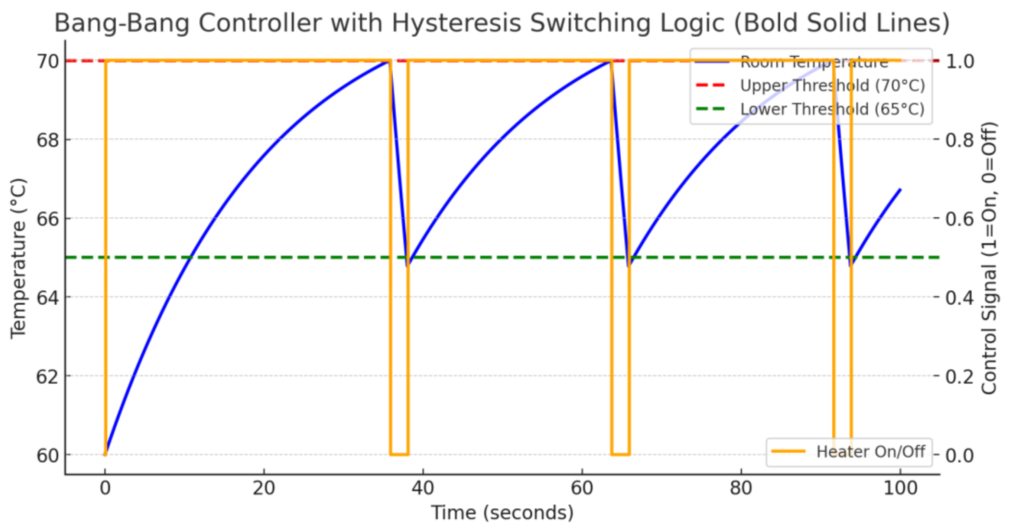


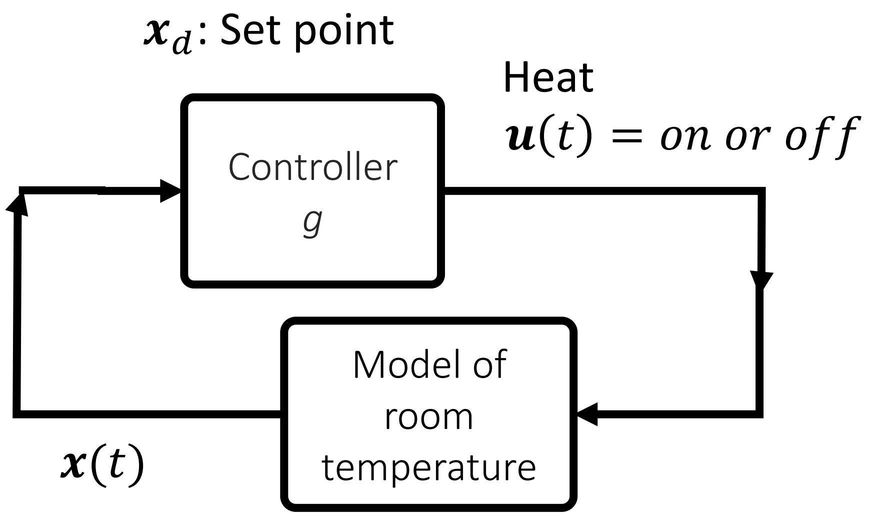
On-off control of a room heater with a thermostat

Bang-bang control is a feasible strategy when the controlled variable is observable

Disadvantages

- **Usually not energy efficient**
- Overshoots and undershoots because of inertia and delays
- Causes excess stress on the actuators
- Can cause the system to become unstable (to be defined later)





A Proportional controller

Plant $\dot{x}(t) = u(t) + d(t)$, where $d(t)$ is a small **disturbance signal**

The goal is the drive the plant state to a target steady state value, say $x_d = 70°$

Idea: Make the control input negatively proportional to the error: **Negative feedback**

Error: $e(t) = x(t) - x_d$

Proportional controller: $u(t) = -K_p e(t)$, the constant K_p is called **controller gain**

Using proportional (P) **negative feedback**

$u(t) = -K_P e(t) = -K_P(x(t) - x_d)$

$\dot{x}(t) = -K_P x(t) + K_P x_d + d(t)$

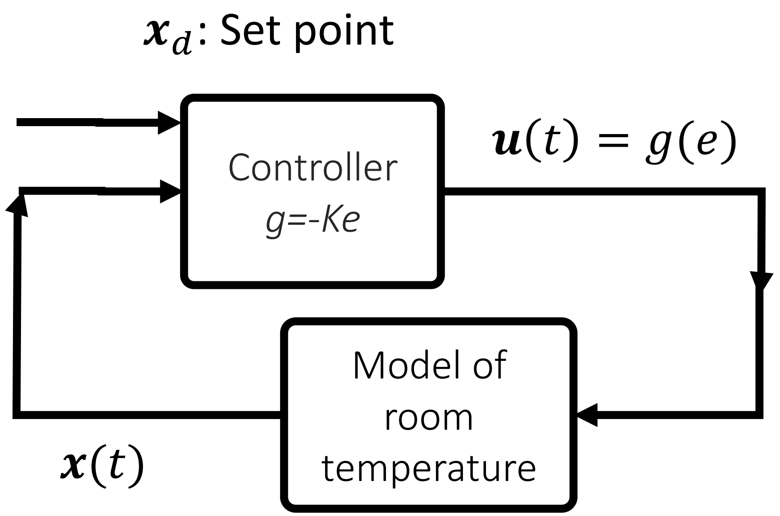
Consider a constant disturbance d_{ss}

$\dot{x}(t) = -K_P x(t) + K_P x_d + d_{ss}$

What is the steady state value? Trick: set RHS = 0

Set $-K_P x(t) + K_P x_d + d_{ss} = 0$

$x(t) = x_{ss} = \frac{d_{ss}}{K_P} + y_d$



Proportional controller example

With constant disturbance d_{ss} we rewrite the ODE

$$\dot{x}(t) = -K_P x(t) + K_P x_d + d_{ss} \text{ with } x_{ss} = \frac{d_{ss}}{K_P} + x_d$$

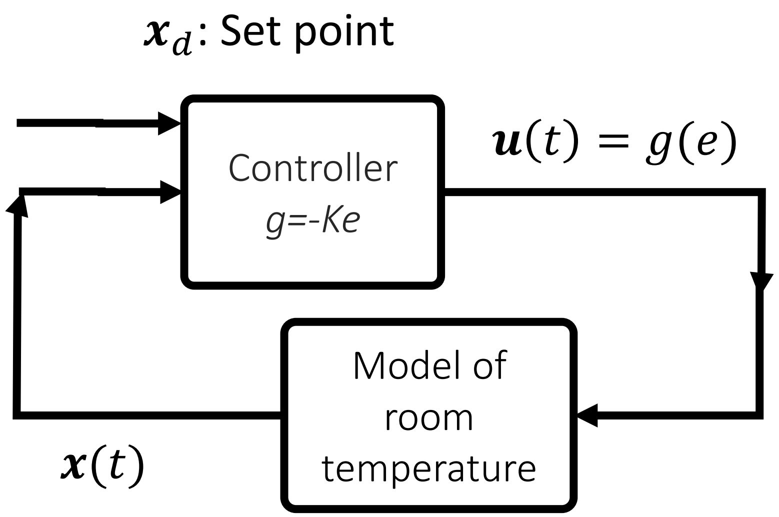
$$\dot{x}(t) = -K_P(x_{ss} - x(t))$$

The solution of this ODE

$$x(t) = x_{ss} + (x(0) - x_{ss})e^{-tK_p}$$

Transient behavior

$$x(t) = x(0)e^{-tK_p} + x_{ss}\left(1 - e^{-tK_p}\right)$$



General solution of first-order linear DE

$x(t) = x_{ss} + Ce^{-K_p t}$

Setting t=0

$x(0) = x_{ss} + C$

Proportional Controller

Transient behavior of the control system

$$x(t) = x(0)e^{-tK_p} + x_{ss}\left(1 - e^{-tK_p}\right); x_{ss} = \frac{d_{ss}}{K_P} + x_d$$

The proportional controller uses negative feedback to track the desired setpoint smoothly

Steady state error may not be 0

Larger proportional gain K_P more reactive the controller and faster the system converges to the target state K_P

Larger K_P implies smaller steady state tracking error

For systems with delays and inertia high proportional gain can cause oscillations or overshoots

There may be actuator limits that prevent $u(t) = -K_P e(t) = -K_P(x(t) - x_d)$ to be a feasible control input