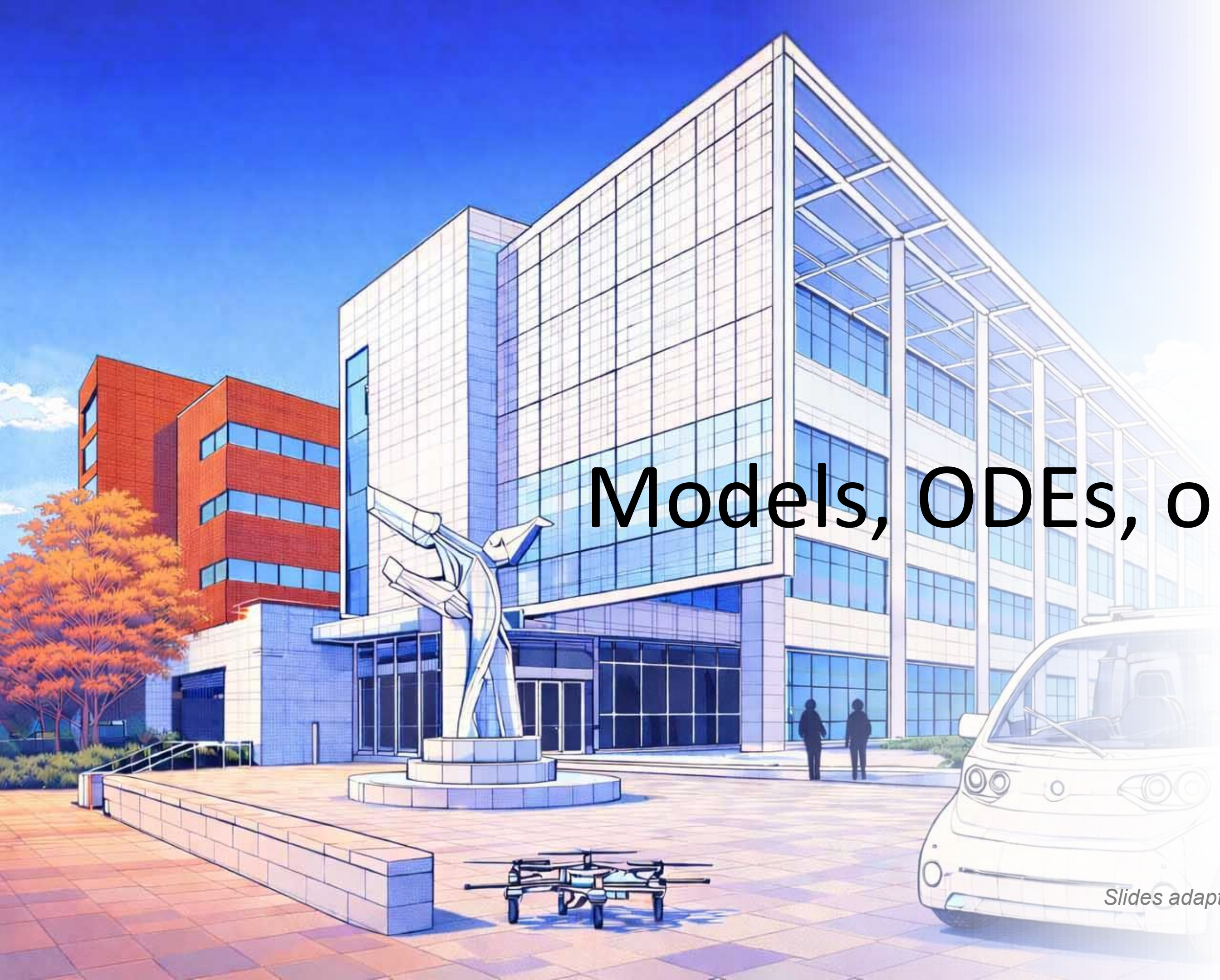


ECE 484
Lecture 8
Control (1): Models, ODEs, on-off control

Prof. Huan Zhang

Fall 2026

Slides adapted in part from materials by Prof. Sayan Mitra.

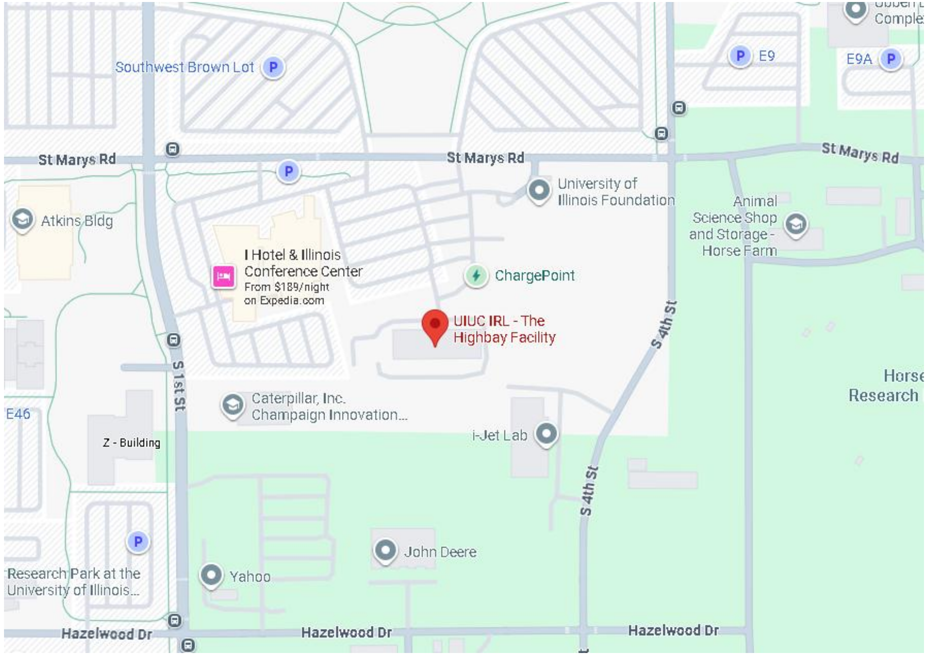
GEM field trip **9/22 11 am (upcoming Tuesday)**

Do not come to the ECEB classroom on **9/22 (next class)**

GEM car field trip at the Highbay facility:

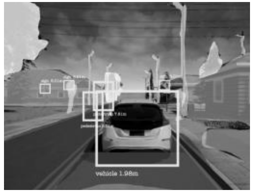
201 St Marys Rd, Champaign, IL 61820

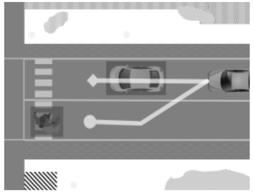
More information on **Campuswire**

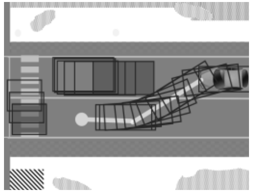


GEM platform

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.

Typical planning and control modules

Global navigation and planner

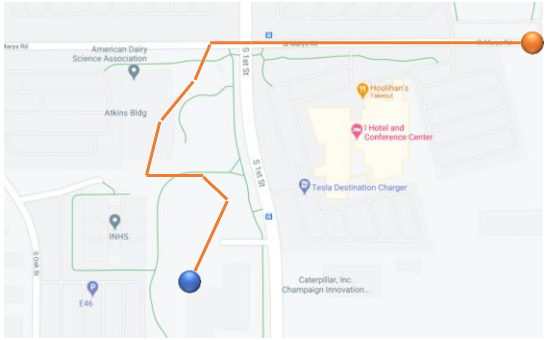
- Find paths from source to destination with static obstacles
- Algorithms: Graph search, Dijkstra, Sampling-based planning
- Time scale: Minutes
- Output: reference center line, does not consider vehicle dynamics

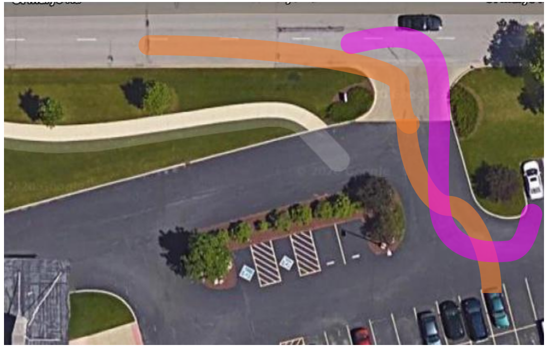
Local planner

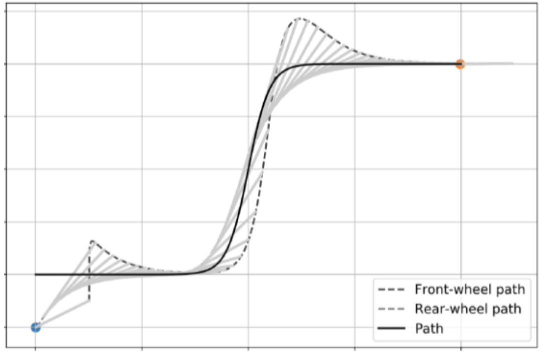
- Dynamically feasible trajectory generation
- Dynamic planning w.r.t. obstacles
- Time scales: 10 Hz

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

Why feedback? Cruise control on a hill

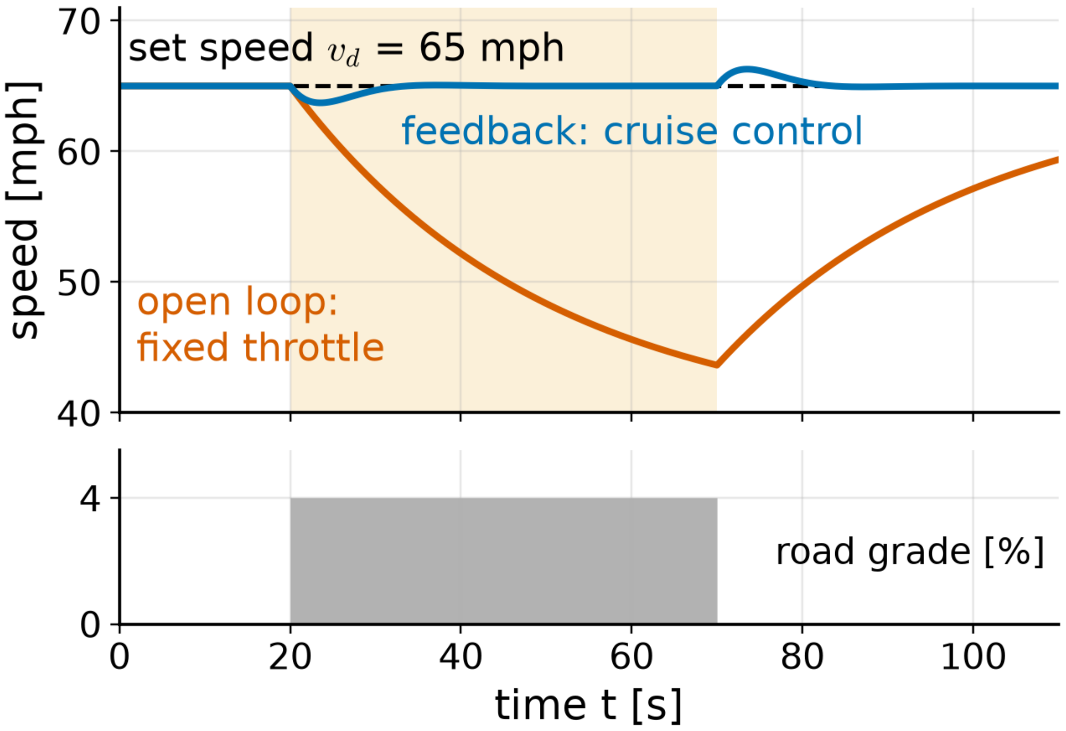
Open loop: pick the throttle that gives 65 mph on a flat road — and never look at the speedometer again

On a 4 % hill the speed sags towards 39 mph

Feedback: measure the speed, compare with the set speed, correct the throttle

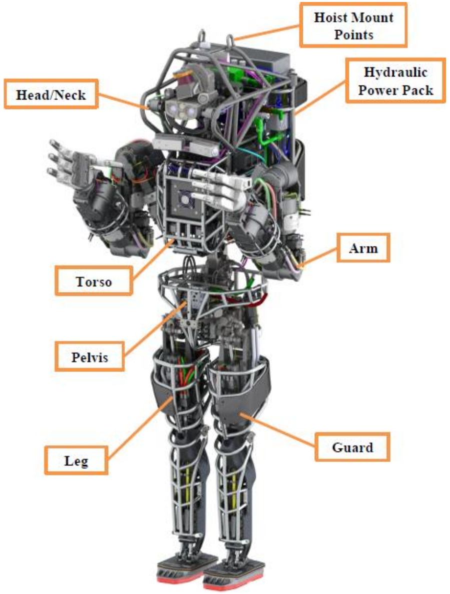
Same hill: a 1.3 mph dip, gone in about 14 s

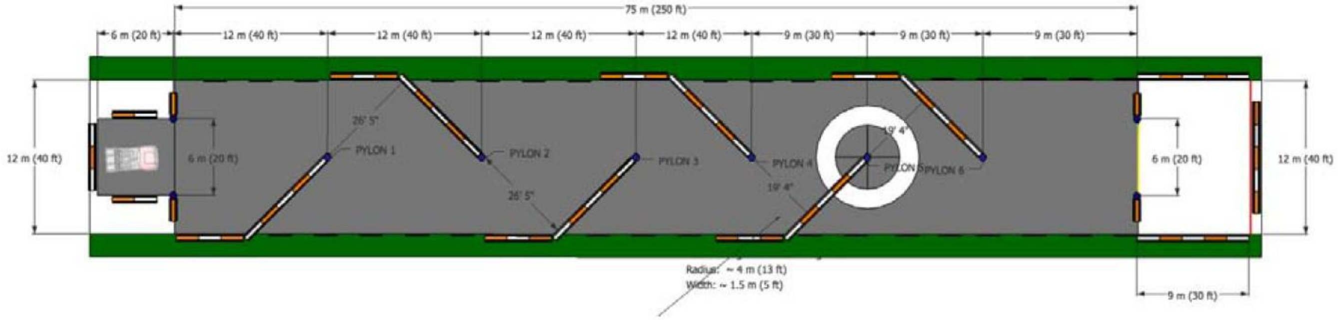
The controller never knew there was a hill. It only watched the error.



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

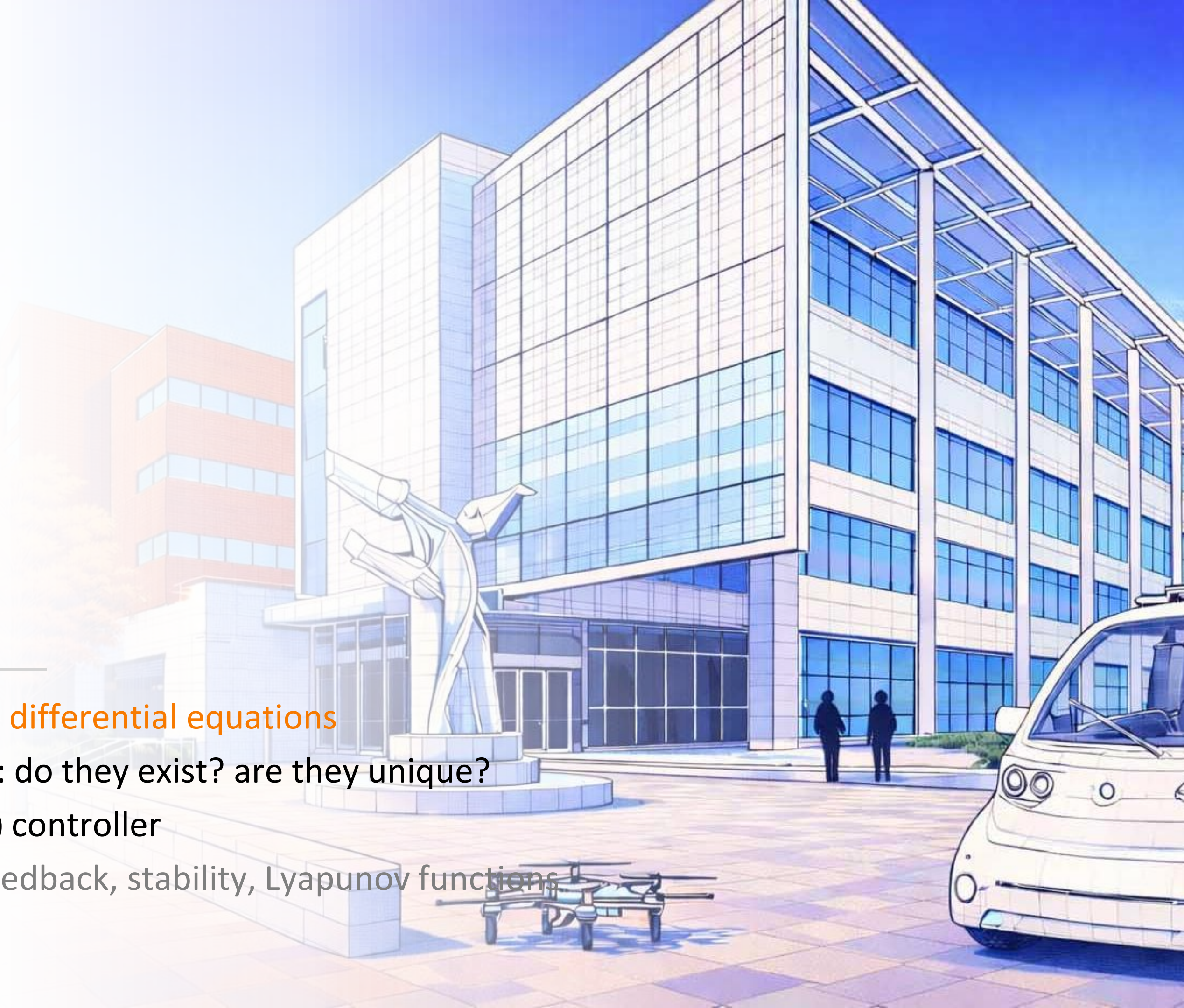
Outline

Modeling the plant: states, inputs, differential equations

Solutions of differential equations: do they exist? are they unique?

Control design: on-off (bang-bang) controller

Next lectures: PID control, state feedback, stability, Lyapunov functions



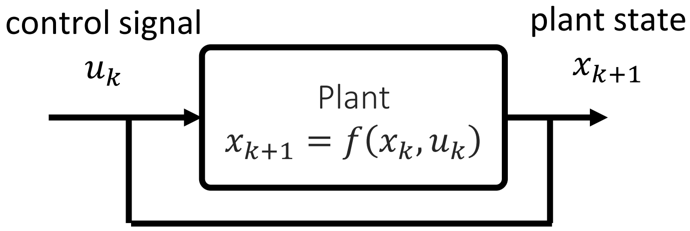
“Thing” being controlled: the Plant

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

Plant state at time step $k \in \mathbb{N}$ is denoted by a vector $x_k \in \mathbb{R}^n$

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

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_k under the influence of a control input u_k, that is, $x_{k+1} = f(x_k, u_k)$



Discrete-time model and automata

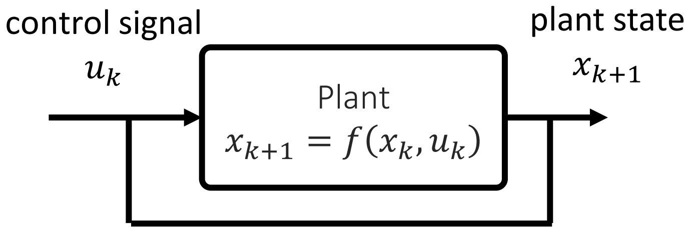
The **dynamics function** $f: \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$ models how the plant state changes in discrete time from a previous state x_k under the influence of a control input u_k, that is, $x_{k+1} = f(x_k, u_k)$

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

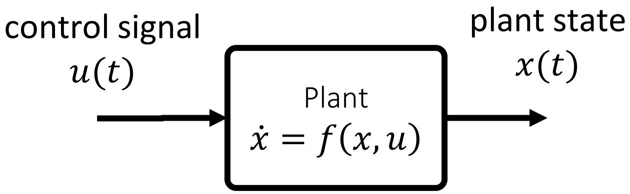
Instead of discrete values of x_k we are interested in how the state changes continuously with time and this is modeled as a function or a signal $x: [0, \infty) \to \mathbb{R}^n$

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

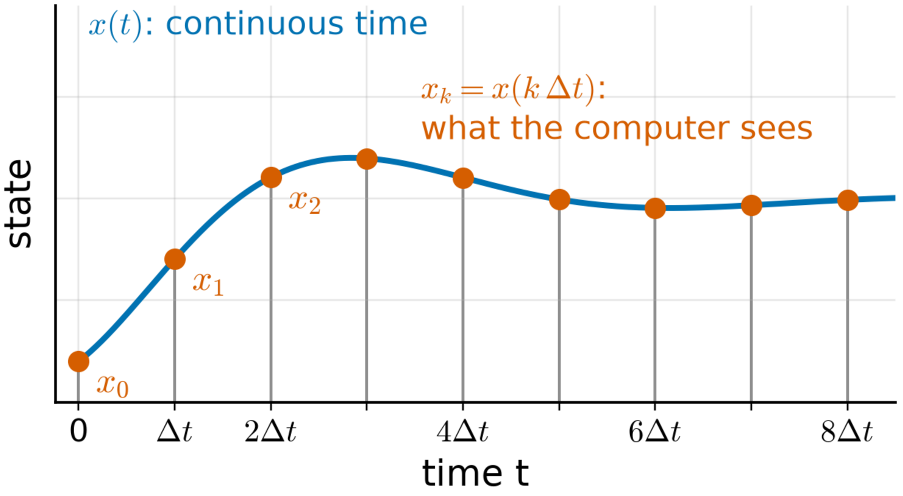
Given $x(\cdot)$ and any t, $x(t)$ denotes the state of the system at time t

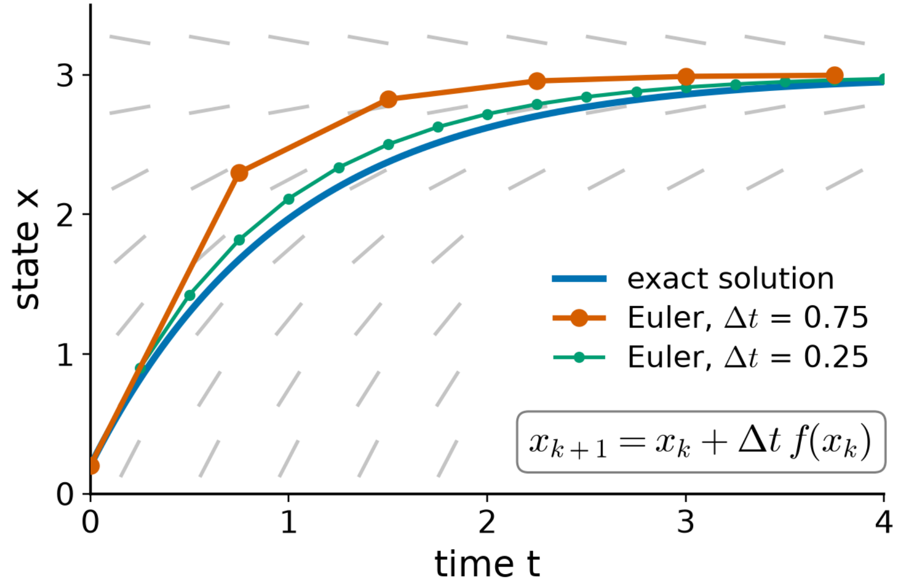
The Ordinary Differential Equation (ODE) relates the input and the state $\frac{dx(t)}{dt} = f(x(t), u(t))$

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



Two views of the same plant: samples and slopes





Discrete time: the computer samples the state, $x_k = x(k\,\Delta t)$, and holds the input u_k until the next sample

Continuous time: $\dot{x} = f(x, u)$ gives the slope at every point; a solution follows the slopes. A simulator takes small steps (Euler): $x_{k+1} = x_k + \Delta t\, f(x_k, u_k)$

Small Δt: the two views agree — we design in continuous time and implement in discrete time

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

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

$$x_2 = \dot{x}_1$$

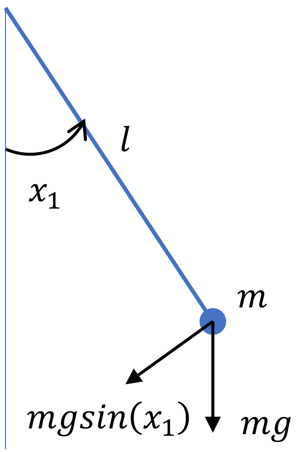
$$\dot{x}_2 = -\frac{g}{l}\sin(x_1) - \frac{k}{m}x_2$$

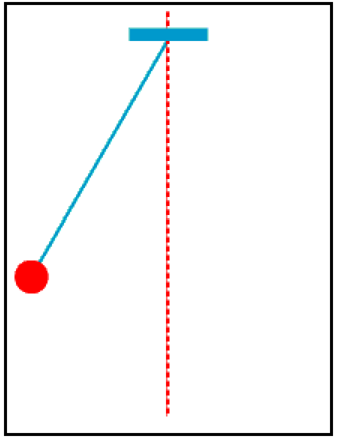
The dynamics equation can be written in vector form:

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

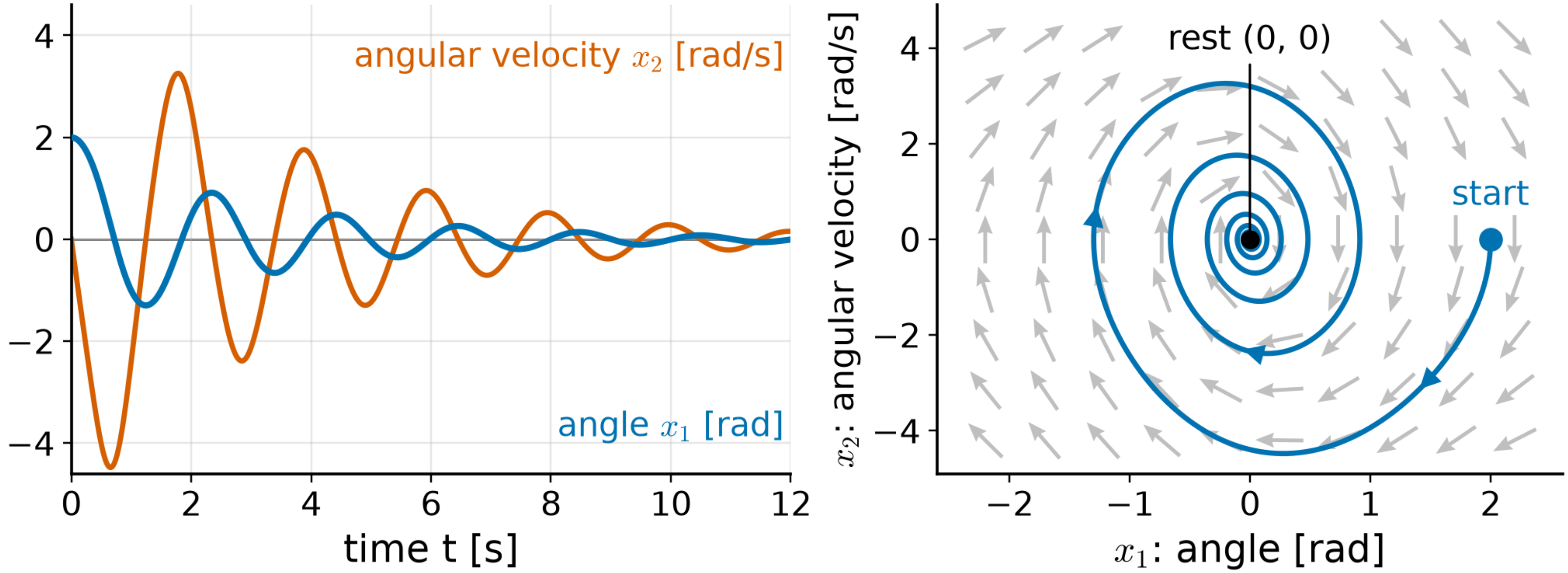
Model parameters

k: friction coefficient m: mass l: length





What the pendulum model predicts

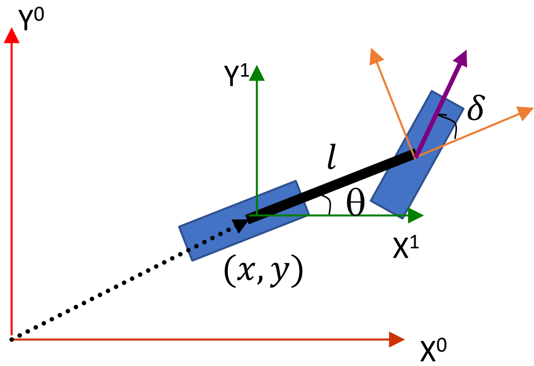


Left: the two states against time. Right: the same run in the state plane — the state is a point, $f(x)$ is the arrow it follows; friction makes it spiral into the rest position $(0,0)$

Example 2: a simple vehicle model (Dubins car)

Key assumptions

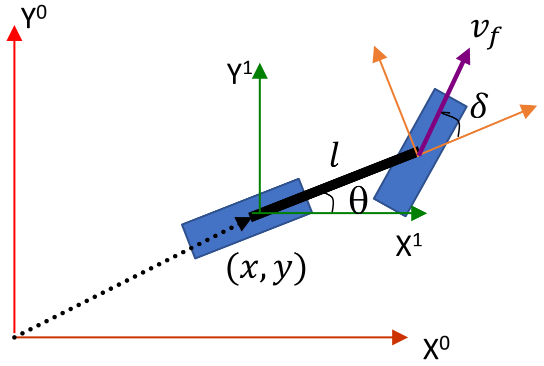
- Front and rear wheel are in vertical planes
- Rear wheel moves at a constant speed v
- Steering input, front wheel steering angle δ
- No slip: wheels move only in the direction of the plane they reside in



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 (kinematic bicycle / Dubins car)



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

Input = front-wheel steering angle: $u = \delta \in \mathbb{R}$

Parameters: wheelbase l, rear-wheel speed v

Front wheel (purple arrow): $v_f = v/\cos\delta$

$\dot{\boldsymbol{x}} = f(\boldsymbol{x}, u)$, with $f: \mathbb{R}^3 \times \mathbb{R} \rightarrow \mathbb{R}^3$

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

What the car model predicts: constant steering = a circle

Hold the steering angle δ constant:

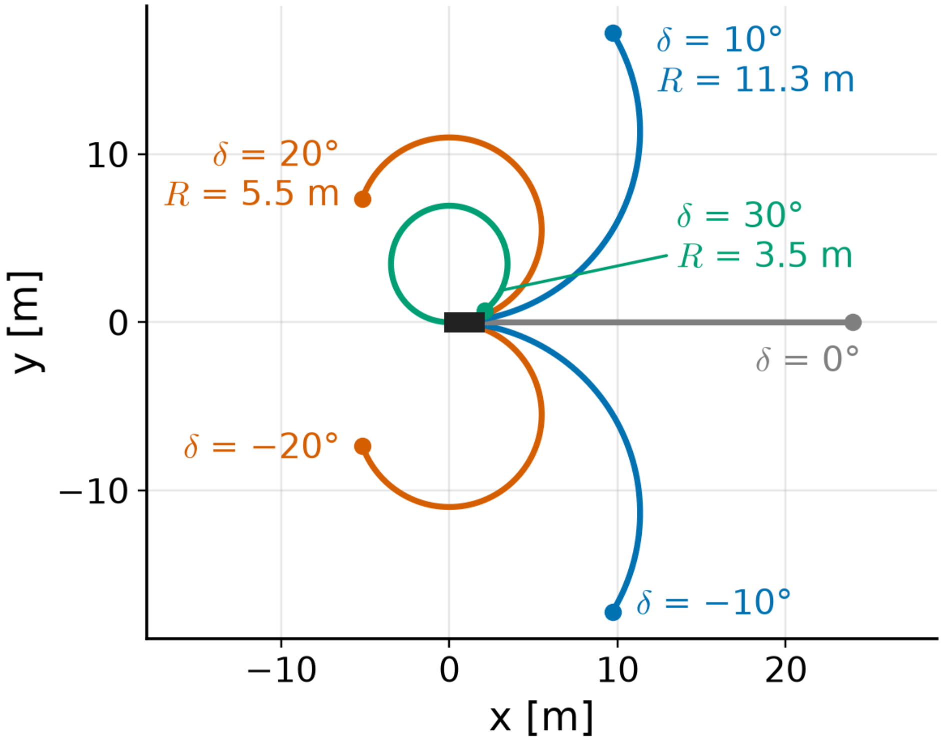
$\dot{\theta} = \frac{v}{l}\tan\delta$ is constant — the car turns at a constant rate while moving at constant speed: a circle

One lap: $2\pi R = v \cdot \frac{2\pi}{\dot{\theta}}$, so $R = \frac{v}{\dot{\theta}} = \frac{l}{\tan\delta}$

Check: $l = 2$ m, $\delta = 20°$ gives $R = 5.5$ m

Q: what does the model say for $\delta \to 90°$?

$R \to 0$: the car spins about its rear wheel. Real tyres slip long before — the no-slip assumption of slide 16 fails, so the model is only trusted for moderate δ and speed



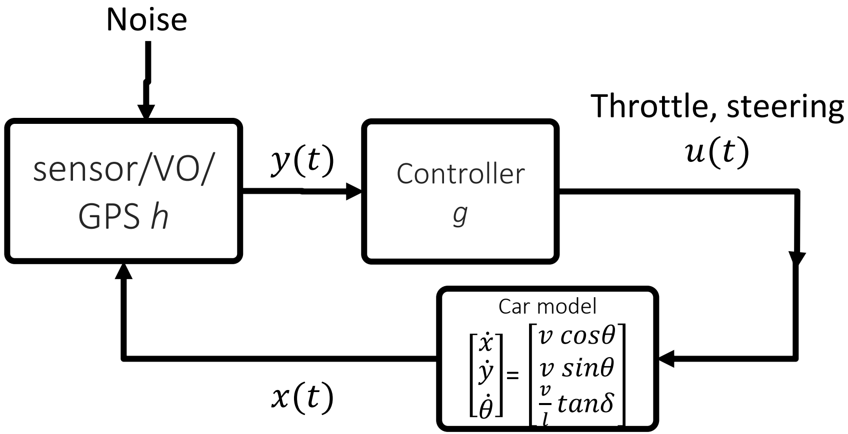
The control problem: closing the loop

The controller $g(\cdot)$ 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

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

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



Outline

Modeling the plant: states, inputs, differential equations

Solutions of differential equations: do they exist? are they unique?

Control design: on-off (bang-bang) controller

Next lectures: PID control, state feedback, stability, Lyapunov functions

Behaviors may not be well-defined

Behaviors of physical processes are described in terms of ODEs

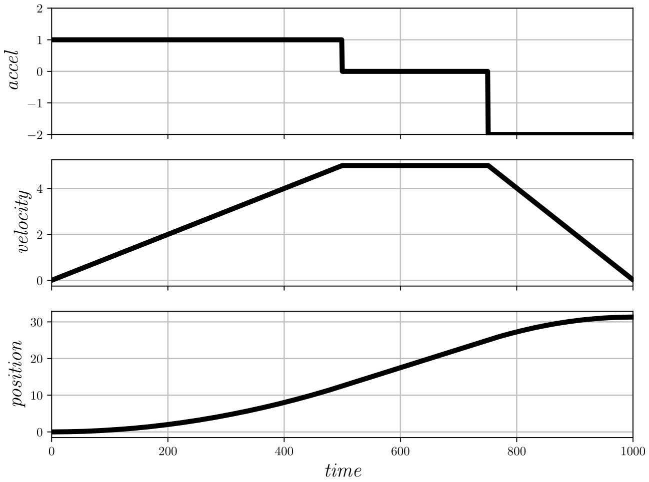
$$\dot{x}(t) \equiv \frac{dx(t)}{dt} = f\big(x(t), u(t)\big) \qquad - \text{ Eq. } (1),$$

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 $\mathrm{u}: \mathbb{R} \rightarrow \mathbb{R}^m$, 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, $\xi\colon \mathbb{R} \to \mathbb{R}^n$ is a 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

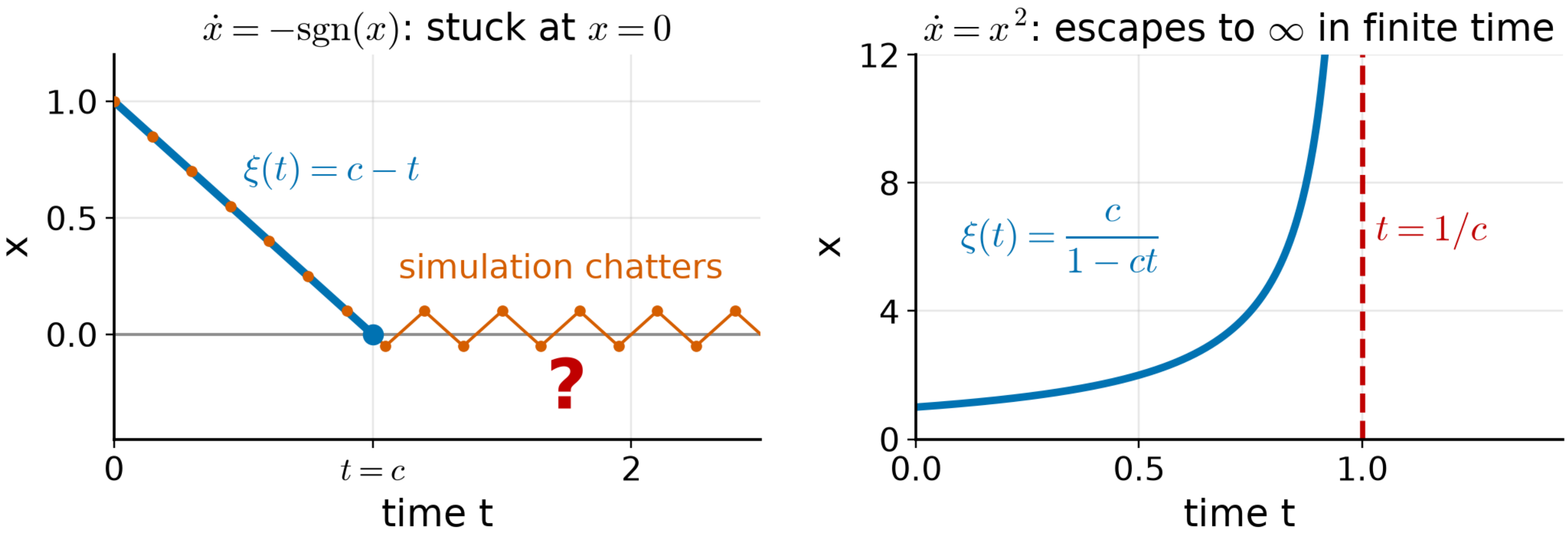
Solutions may not exist even for autonomous ODEs

Example. $\dot{x}(t) = -\mathrm{sgn}(x(t)); x_0 = c; t_0 = 0; c > 0$
Solution: $\xi(t) = c - t$ for $t \leq c$; check $\dot{\xi}(t) = -1 = -\mathrm{sgn}(\xi(t))$
Problem: ξ cannot be continued past $t = c$ as a differentiable function: at $x = 0$, f jumps from -1 to $+1$ (f is discontinuous in x)

Example. $\dot{x}(t) = x^2; x_0 = c; t_0 = 0; c > 0$
Solution: $\xi(t) = \frac{c}{1-ct}$ works for $t < 1/c$; check $\dot{\xi}$
Problem: as $t \to \frac{1}{c}$, $\xi(t) \to \infty$: f grows too fast (faster than linearly in x)

We need assumptions on the smoothness and growth of f to guarantee that solutions exist

The two failures in pictures



Left: the solution reaches 0 at $t = c$ and cannot continue; a simulator chatters around 0 instead. Right: the solution reaches ∞ at $t = 1/c$ — no state exists after that. A safety analysis needs neither to happen.

Solutions may not be unique either

$\dot{x} = \sqrt{x},\;\; x(0) = 0$

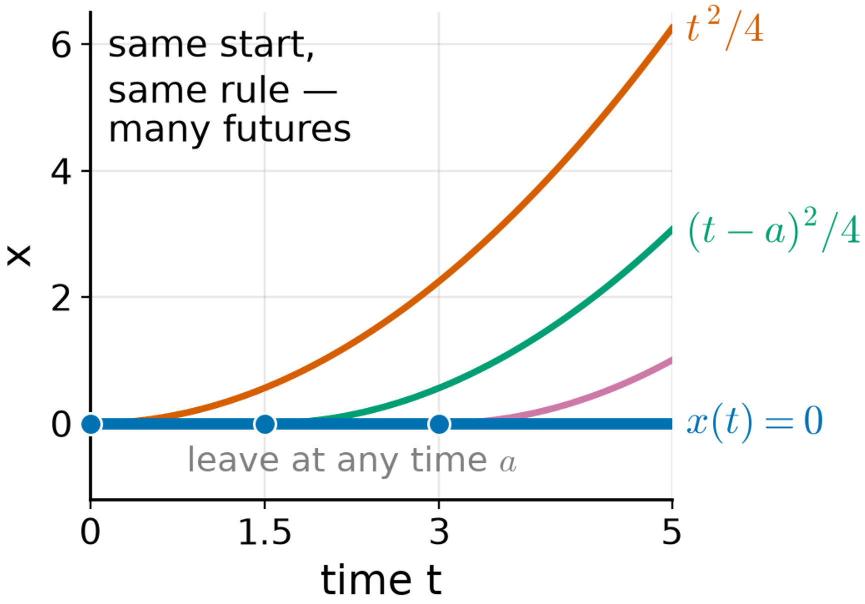
One solution: stay at rest, $x(t) = 0$

Another: $x(t) = \frac{t^2}{4}$ — check: $\dot{x} = \frac{t}{2} = \sqrt{\frac{t^2}{4}}$

Infinitely many: wait until any time a, then leave along $\frac{(t-a)^2}{4}$

The model does not determine the future — "the" trajectory does not exist

Cause: the slope of $\sqrt{x}$ is unbounded near 0



Lipschitz continuity

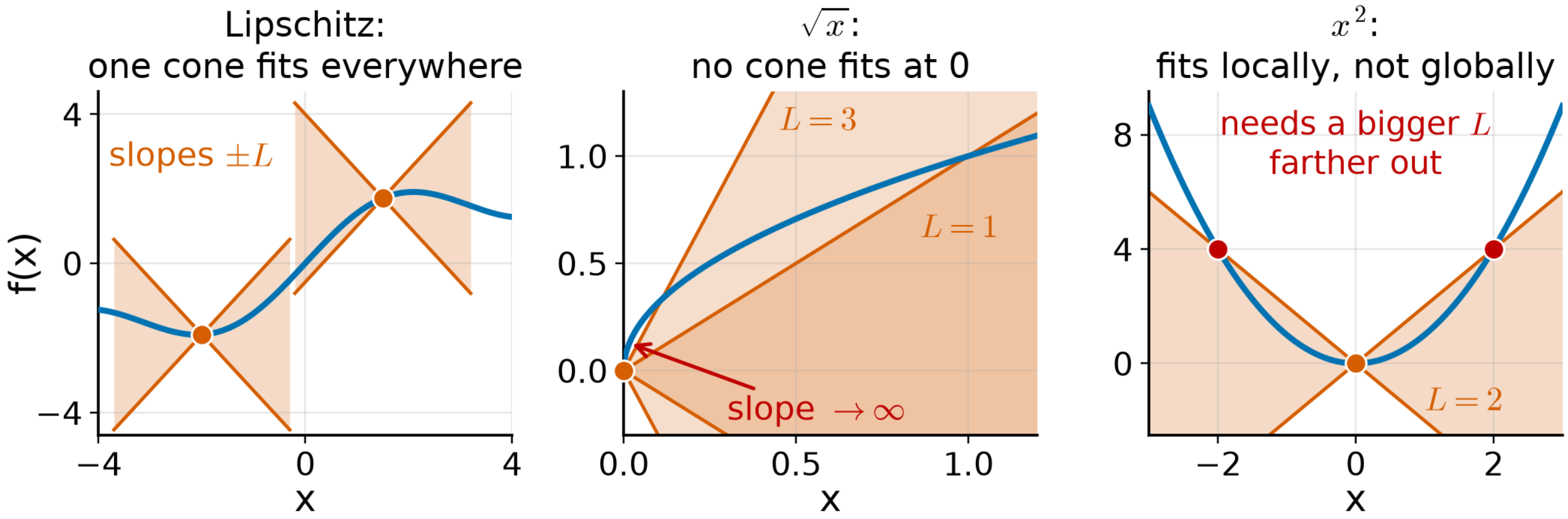
A function $f: \mathbb{R}^n \to \mathbb{R}^m$ is **Globally Lipschitz continuous** if there exists $L > 0$ such that for any pair $x, x' \in \mathbb{R}^n$, $\left||f(x) - f(x')|\right| \leq L\left||x - x'|\right|$

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: x^2 (locally but not globally Lipschitz); $\sqrt{x}$ on $[0, \infty)$ (not even locally Lipschitz at 0: its slope is unbounded near 0)

Lipschitz continuity in pictures



f is Lipschitz with constant L: from every point of its graph, the graph stays inside the cone of slopes $\pm L$, that is $|f(x) - f(x')| \leq L\,|x - x'|$

Existence and uniqueness of solutions

Describe behavior in terms of instantaneous laws

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

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

$f: \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$ dynamics function

Theorem. If $f(x, u)$ is globally Lipschitz in x (uniformly in u), continuous in u, and $u(\cdot)$ is piece-wise continuous, then for every x_0, t_0 the ODE (1) has exactly one solution, defined for all $t \geq t_0$ (Picard-Lindelöf). If f is only locally Lipschitz in x, the solution is still unique but may exist only on a finite interval (finite escape time, e.g. $\dot{x} = x^2$).

Outline

Modeling the plant: states, inputs, differential equations

Solutions of differential equations: do they exist? are they unique?

Control design: on-off (bang-bang) controller

Next lectures: PID control, state feedback, stability, Lyapunov functions

Running example: keep a room at 70 °F

Plant: the room. State $x(t)$ = temperature [°F]

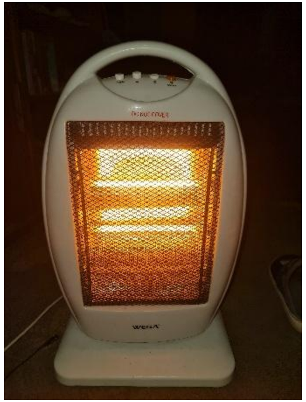
Input $u(t)$: heating rate of the heater [°F/min]

Disturbance $d(t)$: heat lost through walls and windows ($d < 0$)

A deliberately simple model: $\dot{x}(t) = u(t) + d(t)$

Goal: bring $x(t)$ to the set point $x_d = 70$ °F and keep it there

Three controllers for the same room: on-off, P, PI



On-off control of a room heater with a thermostat

Room: $\dot{x}(t) = u(t) + d(t)$

Heater on or off: $u(t) \in \{0, u_{on}\}$, chosen by

A simple thermostat controller

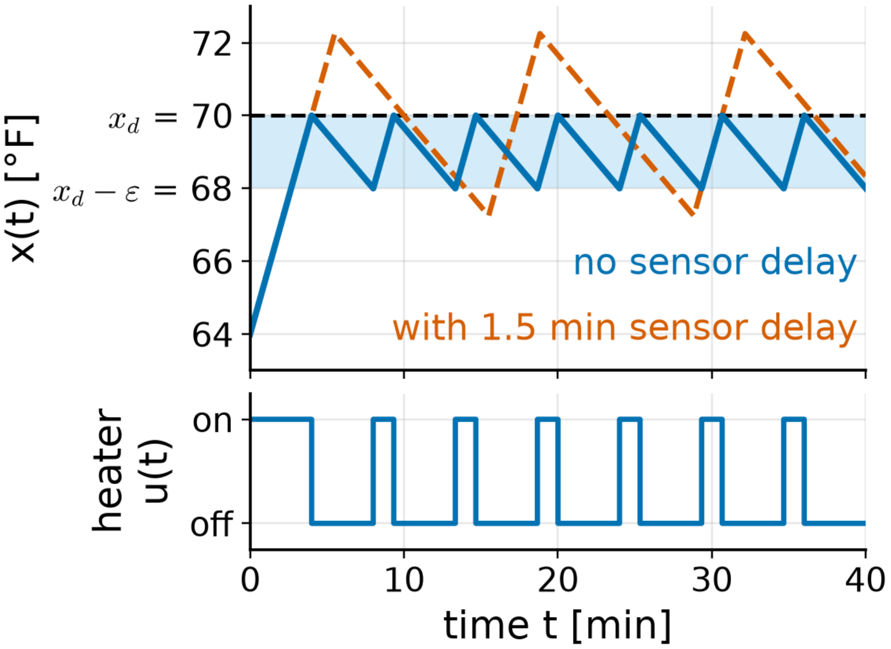
$g\big(x(t)\big)$:

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

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

otherwise keep the previous u (hysteresis: one bit of memory, so u = g(x, mode))

This is called bang-bang control



On-off control: simple, but …

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)