ECE484 Principles of Safe Autonomy
Control

Sayan Mitra

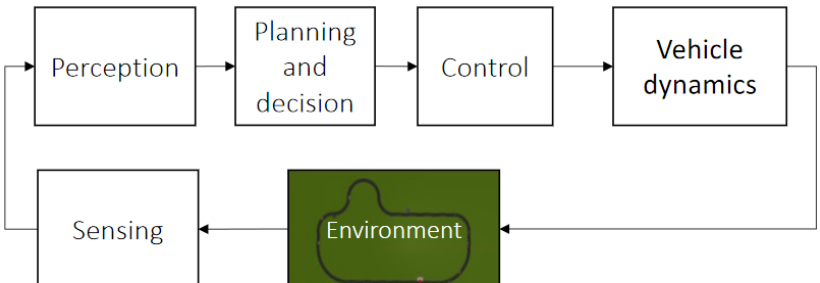
Spring 2026

Outline

- Lecture 8: Modeling the control problem

Control in the Autonomy Pipeline

- Control sits between perception, planning and physics (called the plant)
- Goal: make the system follow a reference trajectory or a set-point



Examples of Control Systems

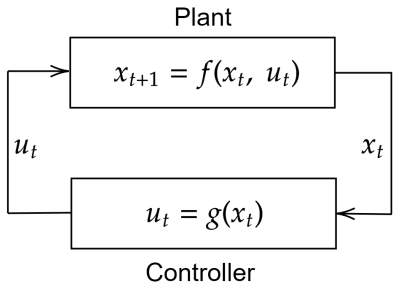
- Thermostat and heating systems
- Cruise control and adaptive cruise control
- Autopilot / drone attitude stabilization
- Robot arm position control

Control at different time scales

- **Global navigation** (minutes)
  Find paths from source to destination with static obstacles
  Algorithms: Graph search, Dijkstra, Sampling-based planning
  Output: reference center line, does not consider vehicle dynamics
- **Local planner** (2 seconds)
  Dynamically feasible trajectory generation avoiding obstacles
  Algorithms: Trajectory optimization, RRT*
  Output: reference trajectory, considers vehicle dynamics but not feedback
- **Controller** (10-50 Hz)
  Waypoint follower using steering, throttle
  Algorithms: PID control, MPC, Lyapunov-based controller

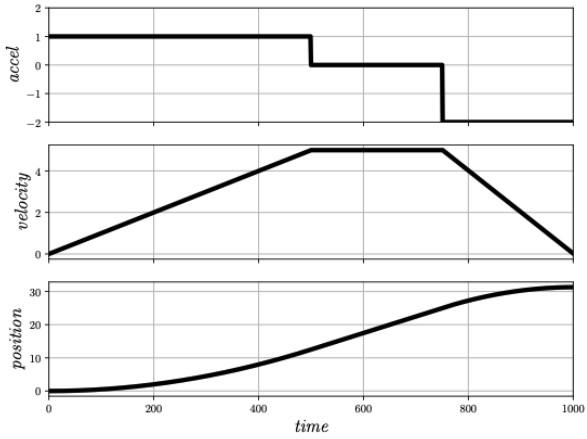
Modelling: Plant and Controller

- Plant: physical process being controlled
- Controller: algorithm/software generating inputs
- Discrete-time model:
  - $x_{t+1} = f(x_t, u_t)$
  - $u_t = g(x_t)$
  - Closed loop: $x_{t+1} = f(x_t, g(x_t)) = f'(x)$
  - Automaton $A = \langle Q, Q_0, \mathcal{D} := \{(x, f'(x))\}\rangle$
- Continuous-time model
  - $\dot{x}(t) = f(x(t), u(t))$
  - Closed loop: $\dot{x}(t) = f(x(t), g(x(t))) = f'(x(t))$



Autonomous ODEs and Solutions

- Autonomous ODE: $\dot{x} = f(x)$
- Solution is a differentiable function $x : \mathbb{R} \to \mathbb{R}^n$ *satisfying* the ODE, i.e., $\dot{x}(t) = f(x(t))$
- Discontinuous inputs can make $x(t)$ non-differentiable

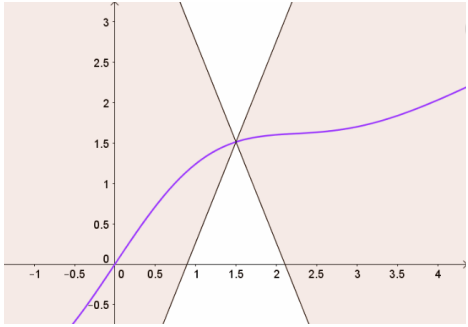


Existence and Uniqueness

- Not all ODEs have unique solutions
- Lipschitz continuity of f guarantees existence and uniqueness of solutions (ODE condition holds at all except the finitely many discontious points in the domain of $x(t)$)

Lipschitz condition

f is Lipschitz if $\exists L > 0$ such that $\forall x, x', \quad \|f(x) - f(x')\| \leq L\|x - x'\|$.

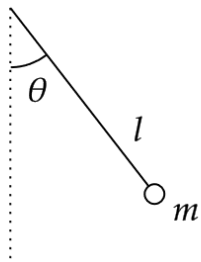


Example: Pendulum

- State: $x_1 = \theta$, $x_2 = \dot{\theta}$
- Parameters: g: gravity, l: length, k: friction, m: mass
- Dynamics:

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

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

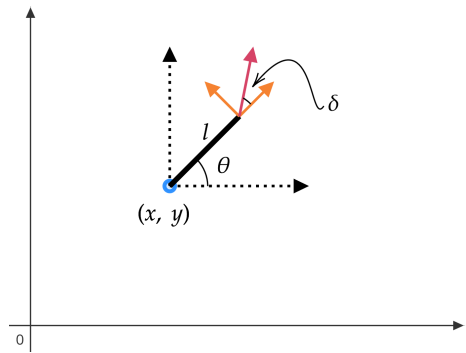


Example: Bicycle Model

- State: $p = [x, y, \theta]$; control: steering δ
- Parameters: ℓ: length
- Kinematics:

$$\dot{x} = v\cos\theta, \quad \dot{y} = v\sin\theta, \quad \dot{\theta} = \frac{v}{\ell}\tan\delta$$

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



Example: Cruise Control (Linear ODE)

- States: position x and speed v
- Input: longitudinal command $u(t)$ (throttle/brake)
- Simple linear model:

$$\dot{x} = v, \qquad \dot{v} = -av + bu, \qquad a > 0,\ b > 0$$

- a: natural speed decay due to drag/rolling resistance
- b: control effectiveness (how strongly u changes speed)
- Matrix form:

$$\dot{\mathbf{x}} = A\mathbf{x} + Bu, \quad \mathbf{x} = \begin{bmatrix} x \\ v \end{bmatrix}, \quad A = \begin{bmatrix} 0 & 1 \\ 0 & -a \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ b \end{bmatrix}$$

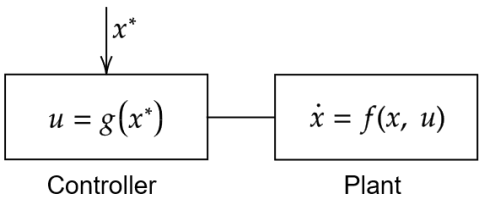
Linear Systems

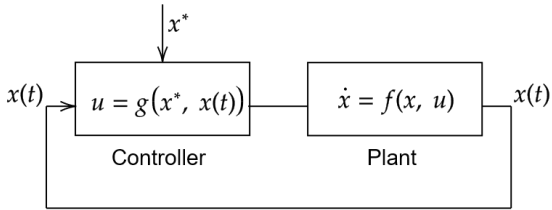
- The RHS is a linear function of the state and input variables
- Linear time-varying (LTV): $\dot{x} = A(t)x + B(t)u$
- Linear time-invariant (LTI): A, B constant
- $x(t) \in \mathbb{R}^n, \quad u(t) \in \mathbb{R}^m$
- $A(t) \in \mathbb{R}^{n \times n}, \quad B(t) \in \mathbb{R}^{n \times m}$
- LTI case: $A \in \mathbb{R}^{n \times n}, \quad B \in \mathbb{R}^{n \times m}$ are constant matrices

Outline: Control Design

- Bang-Bang
- PID control
- State space methods

Open vs Closed Loop





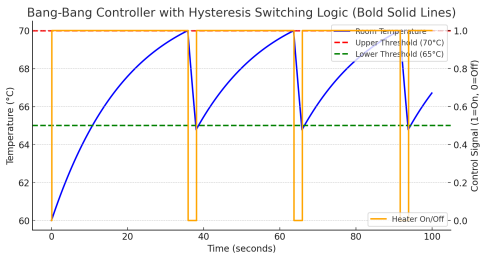
- Open-loop: input does not depend on state
- Closed-loop: feedback from plant output

On-off control of a room heater with a thermostat

- Bang-bang control switches between two modes: heater ON or OFF.
- Let $x(t)$ be room temperature, with thresholds $x_U > x_L$:

$$u(t) = \begin{cases} 0, & x(t) \geq x_U \quad \text{(heater OFF)}, \\ 1, & x(t) \leq x_L \quad \text{(heater ON)}, \\ u(t^-), & x_L < x(t) < x_U. \end{cases}$$

- Hysteresis (x_U, x_L) reduces rapid switching.



Bang-Bang Control: Disadvantages

- Usually not energy efficient (keeps switching between full ON and full OFF).
- Overshoot / undershoot common due to inertia and delays in the plant.
- Frequent switching increases actuator wear and can reduce hardware lifetime.
- Oscillations/chattering near thresholds; hysteresis reduces but does not eliminate.
- Not suitable when precise tracking or smooth control effort is required.

A Proportional Controller

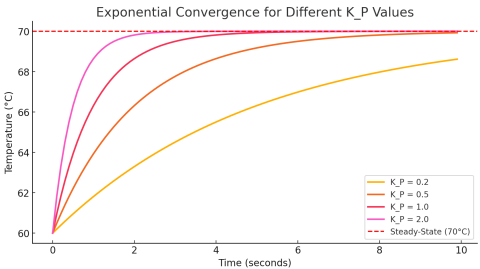
- Plant model (room temperature):

$$\dot{x}(t) = -a(x(t) - x_{\text{env}}) + u(t) + d(t), \qquad a > 0$$

- Setpoint: x_d, error: $e(t) = x(t) - x_d$
- Proportional controller; negative feedback:

$$u(t) = -K_P e(t), \qquad K_P > 0$$

- Larger gain K_P gives faster convergence but can increase overshoot.



From Scalar to Vector-State Systems

- For multi-state plants, use a linear model

$$\dot{x} = Ax + Bu, \qquad y = Cx, \qquad x \in \mathbb{R}^n.$$

- Around an equilibrium (x^*, u^*), define deviation $\tilde{x} = x - x^*$.
- Vector proportional state feedback:

$$u = u^* - K\tilde{x}.$$

$$\dot{\tilde{x}} = (A - BK)\tilde{x}$$

- Stability and transient response are determined by the eigenvalues of $(A - BK)$.
- The gain matrix K controls how strongly each state component is corrected.
- If only outputs are measured, use output-error feedback $u = -K_y(y - y^*)$.

PID Control

- Adds integral and derivative terms

$$u(t) = K_P e(t) + K_I \int_0^t e(\tau) d\tau + K_D \frac{de(t)}{dt}$$

- K_P: reduces error quickly
- K_I: reduces steady-state error
- K_D: damping / anticipatory action

Proportional Analysis: Steady-State Error

- Course-reader setup: $\dot{y}(t) = u(t) + d(t)$, with target y^*.
- Proportional feedback: $u(t) = -K_P(y(t) - y^*)$.
- Closed-loop model:

$$\dot{y}(t) = -K_P(y(t) - y^*) + d(t)$$

- If disturbance becomes constant, $d(t) = d_{\text{ss}}$:

$$0 = -K_P(y_{\text{ss}} - y^*) + d_{\text{ss}} \quad \Rightarrow \quad y_{\text{ss}} = y^* + \frac{d_{\text{ss}}}{K_P}$$

Proportional Analysis: Transient Behavior

- Rewrite dynamics using y_{ss}:

$$\dot{y}(t) = -K_P y(t) + K_P y_{\text{ss}}$$

- First-order solution:

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

- Larger K_P: faster convergence and smaller steady-state offset.
- Tradeoff: larger control effort; high gain can cause overshoot/instability with delays.

Cruise Control PID Example: Setup

- Longitudinal model: $\dot{v} = -av + bu + d(t)$ with $a, b > 0$
- Desired speed $v^*(t)$ and error $e(t) = v^*(t) - v(t)$
- Assume $v^*(t)$ is constant on the interval ($\dot{v}^* = 0$)
- PID law:

$$u(t) = K_P e(t) + K_I \int_0^t e(\tau) d\tau + K_D \dot{e}(t)$$

Cruise Control PID Example: Error ODE

- For $d(t) = 0$, substitute the PID law into error dynamics.
- Differentiating once gives:

$$(1 + bK_D)\,\ddot{e}(t) + (a + bK_P)\,\dot{e}(t) + bK_I\,e(t) = 0$$

- This is the second-order homogeneous ODE used for pole/eigenvalue analysis.

Cruise Control PID: State Space and Eigenvalues

- Define $x_1 = e,\ x_2 = \dot{e}$.

$$\dot{x}_1 = x_2, \qquad \dot{x}_2 = -\frac{a + bK_P}{1 + bK_D}x_2 - \frac{bK_I}{1 + bK_D}x_1$$

$$\dot{x} = Ax, \quad A = \begin{bmatrix} 0 & 1 \\ -\frac{bK_I}{1+bK_D} & -\frac{a+bK_P}{1+bK_D} \end{bmatrix}$$

$$\lambda^2 + \frac{a + bK_P}{1 + bK_D}\lambda + \frac{bK_I}{1 + bK_D} = 0$$

Cruise Control PID: Eigenvalue Formula

$$\lambda_{1,2} = -\frac{a + bK_P}{2(1 + bK_D)} \pm \frac{1}{2}\sqrt{\left(\frac{a + bK_P}{1 + bK_D}\right)^2 - 4\frac{bK_I}{1 + bK_D}}$$

- Gains move poles directly; damping/speed tradeoff is explicit in this expression.

Hurwitz Criterion (Second Order)

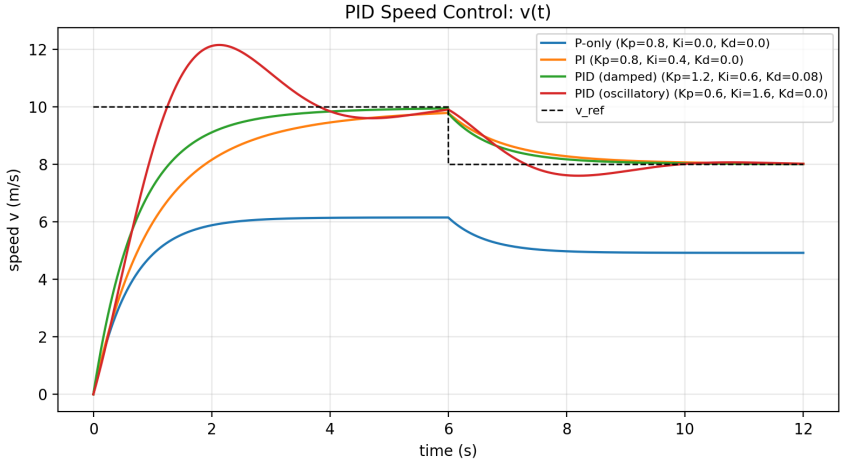
- For the cruise PID characteristic polynomial, asymptotic stability holds if:

$$1 + bK_D > 0, \qquad a + bK_P > 0, \qquad bK_I > 0.$$

- These are the course-reader gain conditions for negative-real-part eigenvalues.

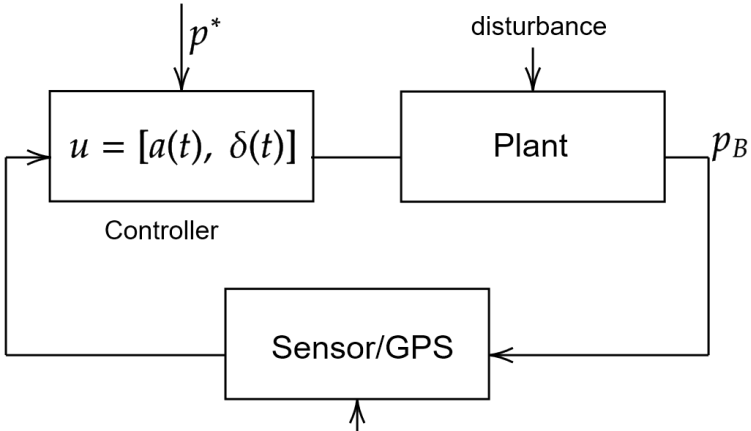
PID Speed Control Example

- Bicycle longitudinal model: $\dot{v} = -av + bu$
- PID on error $e = v^* - v$



Path Following Errors

- Along-track error δ_s
- Cross-track error δ_n
- Heading error δ_θ
- Speed error δ_v



Path Following Errors: Formulas

Let (x_B, y_B, θ_B) be the vehicle pose and (x^*, y^*, θ^*) the reference.

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

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

$$\delta_\theta = \theta^* - \theta_B, \quad \delta_v = v^* - v_B$$

These errors are computed in the body frame and drive the controller.

Pure Pursuit / PD Control

$$u(t) = K \begin{bmatrix} \delta_s \\ \delta_n \\ \delta_\theta \\ \delta_v \end{bmatrix}, \quad K = \begin{bmatrix} K_s & 0 & 0 & K_v \\ 0 & K_n & K_\theta & 0 \end{bmatrix}$$

- Decouples longitudinal and lateral control
- Gains tune responsiveness and stability

Outline: State Feedback

- Linear systems and solutions
- Eigenvalues and stability
- State feedback design

LTI Solutions

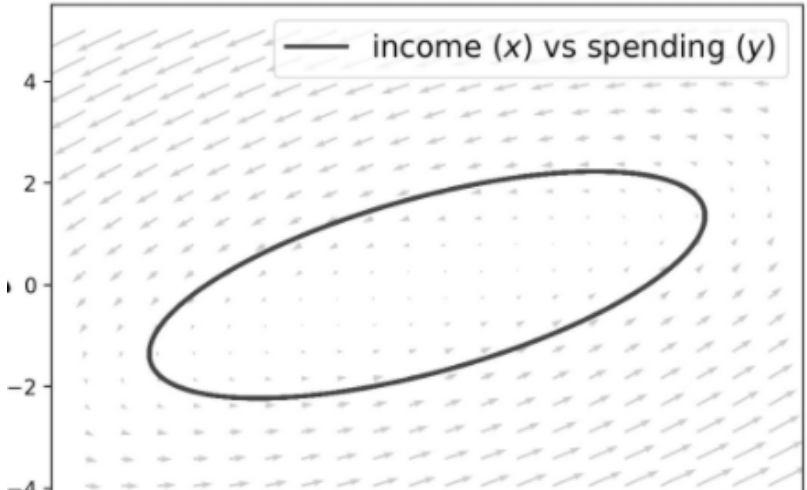
- LTI: $\dot{x} = Ax$
- Solution: $x(t) = e^{At}x(0)$

Stability

An LTI system is asymptotically stable iff all eigenvalues of A have strictly negative real parts.

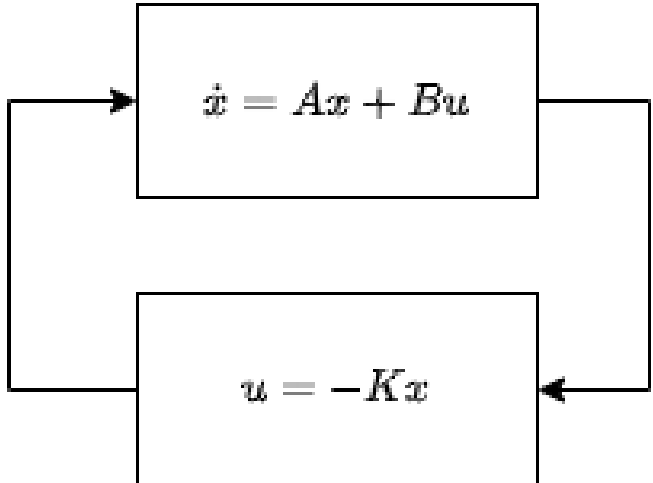
Vector Fields and Trajectories

- ODEs define a vector field $\dot{x} = f(x)$
- Trajectories follow field lines



State Feedback

- Plant: $\dot{x} = Ax + Bu$
- State feedback: $u = -Kx$
- Closed loop: $\dot{x} = (A - BK)x$



State Feedback: Closed-Loop Derivation

- Start with $\dot{x} = Ax + Bu$
- Choose $u = -Kx$
- Substitute:

$$\dot{x} = Ax + B(-Kx) = (A - BK)x$$

- Stability depends on eigenvalues of $A - BK$

Eigenvalue Placement

- Choose K so that $A - BK$ is Hurwitz
- Use characteristic polynomial to place eigenvalues

$$\det(\lambda I - (A - BK)) = 0$$

Linearized Path-Tracking Example

- Linearization gives $\dot{x} = Ax + Bu$
- Use $u = -Kx$ for stabilization

$$\begin{bmatrix} \dot{\delta}_s \\ \dot{\delta}_n \\ \dot{\delta}_\theta \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & v \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \delta_s \\ \delta_n \\ \delta_\theta \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \delta_v \\ \delta_k \end{bmatrix}$$