ECE484 Principles of Safe Autonomy
Control and Stability

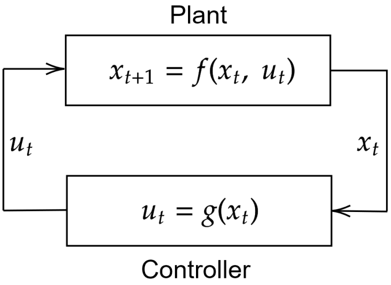
Sayan Mitra

Spring 2026

Outline

Plant and Controller

- Plant: physical process being controlled
- Controller: algorithm/software generating inputs
- Discrete-time closed-loop:
  - $x_{t+1} = f(x_t, u_t), \quad u_t = g(x_t)$
  - $x_{t+1} = f(x_t, g(x_t)) =: f'(x_t)$
- Continuous-time closed-loop:
  - $\dot{x} = f(x, u), \quad u = g(x)$
  - $\dot{x} = f(x, g(x)) =: f'(x)$



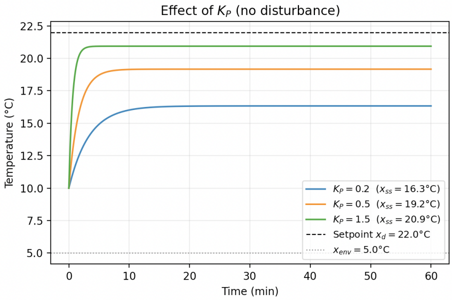
A Proportional Controller

- Plant model (room temperature):

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

- $x(t)$: room temperature, x_{env}: ambient temperature
- $u(t)$: heater input, $d(t)$: disturbance (e.g. open door)
- Goal: maintain $x(t)$ at setpoint x_d
- Error: $e(t) = x(t) - x_d$
- Proportional controller (negative feedback):

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



Proportional Control: Closed-Loop Analysis

- Substitute $u = -K_P(x - x_d)$, $d = 0$ into the plant:

$$\dot{x} = -(a + K_P)\, x + a\, x_{\mathrm{env}} + K_P x_d$$

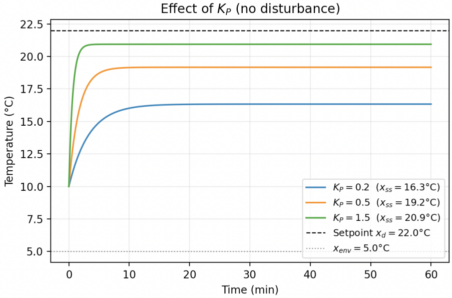
- Equilibrium x_{ss}: set $\dot{x} = 0$

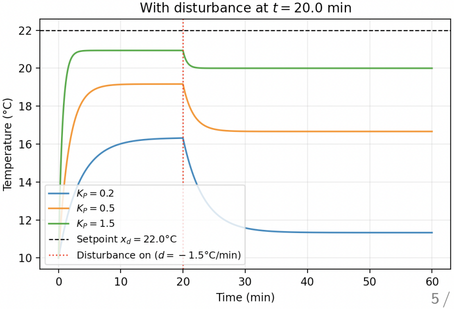
$$x_{\mathrm{ss}} = \frac{a\, x_{\mathrm{env}} + K_P x_d}{a + K_P}$$

- Deviation $\tilde{x} = x - x_{\mathrm{ss}}$ satisfies:

$$\dot{\tilde{x}} = \underbrace{-(a + K_P)}_{\lambda}\, \tilde{x}$$

- Solution: $x(t) = x_{\mathrm{ss}} + \big(x(0) - x_{\mathrm{ss}}\big) e^{\lambda t}$
- $\lambda < 0$: temperature converges to x_{ss}





Limitations of Proportional Control

- Steady-state offset: $x_{\mathrm{ss}} \neq x_d$ whenever $a > 0$ and $x_d \neq x_{\mathrm{env}}$
- Increasing K_P reduces the offset but amplifies noise and can cause overshoot
- No anticipation: P-control reacts only to the current error, not its trend
- Solution: add integral and derivative terms $\Rightarrow$ PID control

Outline

From Scalar to Vector: Linear Systems

- P-control gave us $\dot{\tilde{x}} = \lambda\,\tilde{x}$ with scalar rate $\lambda = -(a + K_P)$
- Solution: $\tilde{x}(t) = \tilde{x}(0)\,e^{\lambda t}$; decays iff $\lambda < 0$
- Real systems have *multiple* state variables:

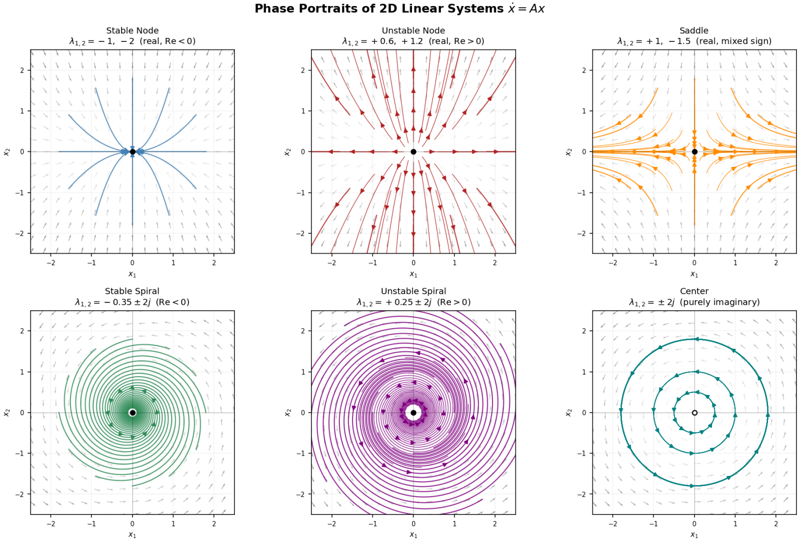
$$\dot{x} = A\,x, \qquad A \in \mathbb{R}^{n\times n}, \quad x \in \mathbb{R}^n$$

- Solution: $x(t) = e^{At}\,x_0$, where the matrix exponential is:

$$e^{At} = I + At + \frac{(At)^2}{2!} + \frac{(At)^3}{3!} + \cdots = \sum_{k=0}^{\infty} \frac{(At)^k}{k!}$$

- Generalizes $e^{\lambda t} = 1 + \lambda t + \frac{(\lambda t)^2}{2!} + \cdots$ to matrices
- The behavior of e^{At} is governed by the eigenvalues of A

Vector Fields and Phase Portraits of 2D Linear Systems



Row 1: real eigenvalues (stable node, unstable node, saddle). Row 2: complex eigenvalues (stable spiral, unstable spiral, center).

Eigenvalues and Stability

- Eigenvalues λ_i and eigenvectors v_i of A: $Av_i = \lambda_i v_i$
- Key observation: A acts as a *scalar* on each eigenvector, so

$$e^{At} v_i = \left(I + At + \frac{(At)^2}{2!} + \cdots \right) v_i = \left(1 + \lambda_i t + \frac{(\lambda_i t)^2}{2!} + \cdots \right) v_i = e^{\lambda_i t}\, v_i$$

- Write any initial condition as $x_0 = \sum_i c_i v_i$; linearity gives the modal decomposition:

$$x(t) = e^{At} x_0 = \sum_{i=1}^{n} c_i\, e^{\lambda_i t}\, v_i, \qquad c_i \text{ set by } x(0)$$

- Each mode $e^{\lambda_i t}$ behaves according to its eigenvalue:
  - $\mathrm{Re}(\lambda_i) < 0$: decays exponentially (stable mode)
  - $\mathrm{Re}(\lambda_i) > 0$: grows exponentially (unstable mode)
  - $\mathrm{Im}(\lambda_i) \neq 0$: oscillates (rotation in the mode direction)

Hurwitz Stability Criterion

$\dot{x} = Ax$ is asymptotically stable $\iff$ every eigenvalue of A has $\mathrm{Re}(\lambda_i) < 0$ (A is called Hurwitz).

Outline

PID Control Law

PID control law

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

- K_P (proportional): reacts to current error magnitude
- K_I (integral): accumulates past error; eliminates steady-state offset
- K_D (derivative): reacts to rate of change; damps overshoot

Cruise Control PID: Setup

- Longitudinal model: $\dot{v} = -av + bu, \quad a, b > 0$
- Desired speed v^* (constant), error $e(t) = v^* - v(t)$, so $\dot{e} = -\dot{v}$
- PID law:

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

- Substitute $v = v^* - e$ and the PID law into $\dot{e} = -\dot{v} = av - bu$:

$$\dot{e} = a(v^* - e) - b\left(K_P e + K_I \int_0^t e\, d\tau + K_D \dot{e}\right)$$

- Collect $\dot{e}$ terms: error ODE (integro-differential)

$$(1 + bK_D)\, \dot{e} \;=\; av^* - (a + bK_P)\, e - bK_I \int_0^t e(\tau)\, d\tau$$

Cruise Control PID: Differentiating the Error ODE

- Start from the integro-differential error ODE (v^* constant):

$$(1 + bK_D)\,\dot{e} = av^* - (a + bK_P)\,e - bK_I\int_0^t e(\tau)\,d\tau$$

- Differentiate once with respect to t:

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

- Rearrange into standard second-order homogeneous form:

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

- This is a linear ODE; we can write it as $\dot{x} = Ax$ and apply eigenvalue analysis

Cruise Control PID: State-Space Form

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

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & 1 \\ -\dfrac{bK_I}{1+bK_D} & -\dfrac{a+bK_P}{1+bK_D} \end{bmatrix}}_{A} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

- Characteristic polynomial of A:

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

- Error converges iff both eigenvalues satisfy $\mathrm{Re}(\lambda_{1,2}) < 0$

Cruise Control PID: Eigenvalue Analysis

- Eigenvalues of the error system:

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

- Hurwitz criterion: all eigenvalues have $\mathrm{Re} < 0$ iff

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

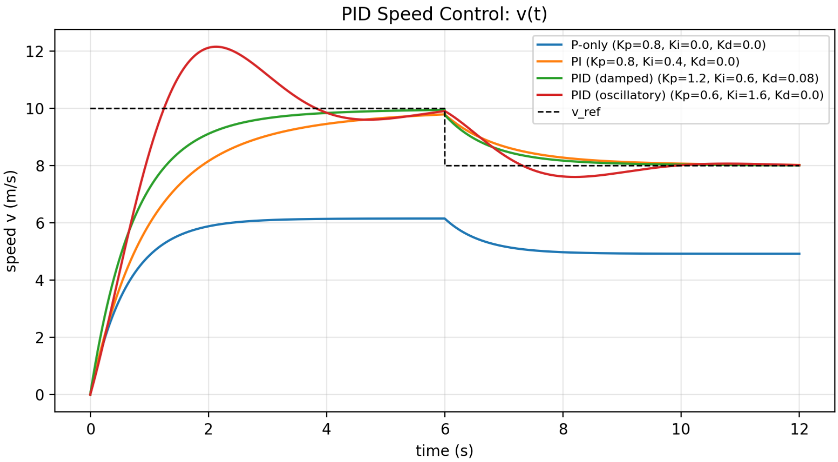
- The discriminant of the characteristic polynomial $\lambda^2 + p\lambda + q = 0$ is:

$$\Delta = p^2 - 4q = \left(\frac{a + bK_P}{1 + bK_D}\right)^2 - \frac{4bK_I}{1 + bK_D}$$

- Δ determines the shape of convergence (all gains satisfying Hurwitz):
  - $\Delta > 0$ (two real λ): overdamped — slow, monotone convergence
  - $\Delta = 0$ (repeated λ): critically damped — fastest without oscillation
  - $\Delta < 0$ (complex λ): underdamped — oscillatory but faster approach
- Gains K_P, K_I, K_D directly shape the eigenvalues $\lambda_{1,2}$

PID Speed Control Example

- Bicycle longitudinal model: $\dot{v} = -av + bu$
- PID on error $e = v^* - v$; different gain sets place $\lambda_{1,2}$ differently



Path Following: Error Definitions

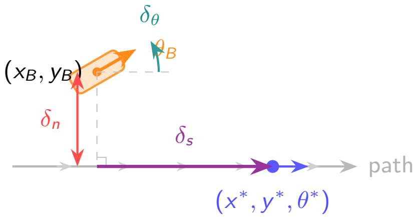
- Vehicle pose: (x_B, y_B, θ_B)
- Reference: (x^*, y^*, θ^*)
- Errors in the body frame:

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

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

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

- δ_s: along-track, δ_n: cross-track
- δ_θ: heading, δ_v: speed



PD Path-Following Control Law

- Stack errors into a vector and apply a gain matrix:

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

- Longitudinal channel: throttle from (δ_s, δ_v)
- Lateral channel: steering from $(\delta_n, \delta_\theta)$
- Decoupled design; each gain tunes a single channel
- Same structure as P-control but for a *vector* error in $\mathbb{R}^2$
- What if the state space is *not* $\mathbb{R}^n$?

PD Path-Following: Lateral Closed-Loop Analysis

- Linearize the lateral dynamics around the reference ($v \approx v^*$, small angles):

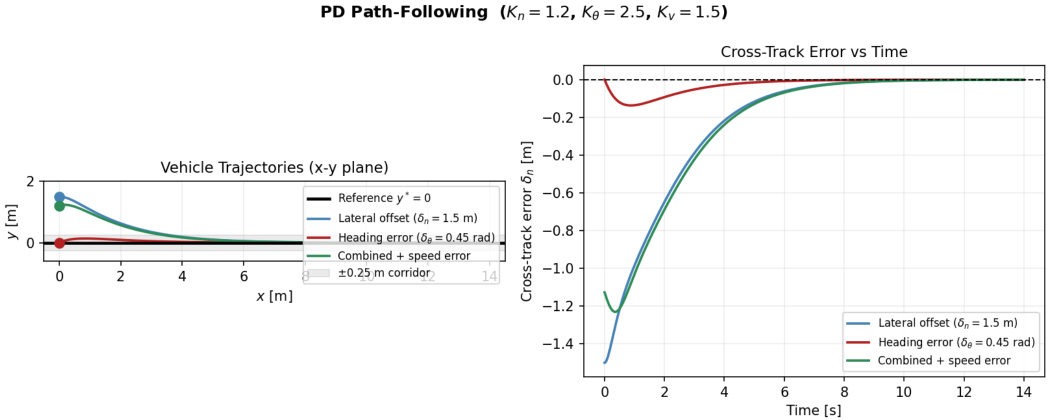
$$\dot{\delta}_n = v^* \, \delta_\theta, \qquad \dot{\delta}_\theta = v^*(K_n \, \delta_n + K_\theta \, \delta_\theta) \cdot (-1)$$

- Written as $\dot{z} = A_{\mathrm{lat}} \, z$, $\quad z = [\delta_n, \, \delta_\theta]^T$:

$$A_{\mathrm{lat}} = \begin{bmatrix} 0 & v^* \\ -K_n v^* & -K_\theta v^* \end{bmatrix}$$

- Characteristic polynomial: $\lambda^2 + K_\theta v^* \, \lambda + K_n v^{*2} = 0$
- Hurwitz iff $K_n > 0$ and $K_\theta > 0$: both errors converge to zero
- Discriminant $\Delta = (K_\theta v^*)^2 - 4 K_n v^{*2}$: choose K_θ, K_n to control damping

PD Path-Following: Simulation



Three initial conditions (lateral offset, heading error, combined + speed mismatch); all converge to $y^* = 0$. `Control/pd_path_following.py`

Outline

Attitude Control: Motivation

- Drone / spacecraft orientation lives on SO(3), not $\mathbb{R}^3$
- $\mathrm{SO}(3) = \{R \in \mathbb{R}^{3\times 3} \mid R^T R = I,\ \det R = +1\}$
- Cannot define error as $R - R_d$: matrix difference is not a rotation
- Cannot add rotations: $R_1 + R_2 \notin \mathrm{SO}(3)$ in general
- Need a geometric error that lives on the manifold

SO(3) Kinematics

The hat map $[\cdot]_\times$

- Encodes cross product as matrix multiply:

$$[\omega]_\times v \;=\; \omega \times v \quad \forall\, v$$

- $[\omega]_\times = \begin{bmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{bmatrix}$
- *Recall*: Stereo lecture used $[t]_\times$ to write the epipolar constraint $x'^\top [t]_\times R\, x = 0$
- Maps $\mathbb{R}^3 \to \mathfrak{so}(3)$: always skew-symmetric, $[\omega]_\times^T = -[\omega]_\times$

Attitude Kinematics: $\dot{R} = R\,[\omega]_\times$

- Take any vector p fixed in body frame
- World position: $q(t) = R(t)\,p$
- Euler's law (*world-frame* ω_W):

$$\dot{q} = \omega_W \times q$$

- Since $\omega_W = R\,\omega_B$ and $Ra \times Rb = R(a \times b)$:

$$\dot{q} = R\omega_B \times Rp = R(\omega_B \times p) = R\,[\omega_B]_\times\, p$$

- Chain rule: $\dot{q} = \dot{R}\,p$; true for *all* p, so:

$$\boxed{\dot{R} = R\,[\omega_B]_\times}$$

Defining the Rotation Error

- Given desired rotation $R_d(t)$, define the relative rotation:

$$R_e = R_d^T R \in \mathrm{SO}(3)$$

- $R_e = I$ iff $R = R_d$ (perfect tracking)
- Extract a vector error using the vee map $(\cdot)^\vee : \mathfrak{so}(3) \to \mathbb{R}^3$
- Skew-symmetric part of R_e carries the axis-angle information:

$$e_R = \frac{1}{2}\left(R_d^T R - R^T R_d\right)^\vee \in \mathbb{R}^3$$

- $e_R = 0$ iff $R = R_d$; $\quad \|e_R\| \approx \sin\theta$ for small misalignment θ
- This is the SO(3) analogue of $e = x - x_d$ in $\mathbb{R}^n$

Aside: Lie algebra $\mathfrak{so}(3)$

$\mathfrak{so}(3)$ is the **Lie algebra** of SO(3) — the set of all skew-symmetric 3×3 matrices. The vee map $(\cdot)^\vee$ is its inverse: it extracts the $\mathbb{R}^3$ vector from a skew-symmetric matrix. Recall the matrix exponential from earlier: $e^{[\hat{\omega}]_\times \theta} \in \mathrm{SO}(3)$ is a rotation by angle θ about axis $\hat{\omega}$ — the exponential map takes us from the Lie algebra back to the Lie group.

SO(3) PD Control Law

- Define the angular velocity error:

$$e_\omega = \omega - R^T R_d \, \omega_d \in \mathbb{R}^3$$

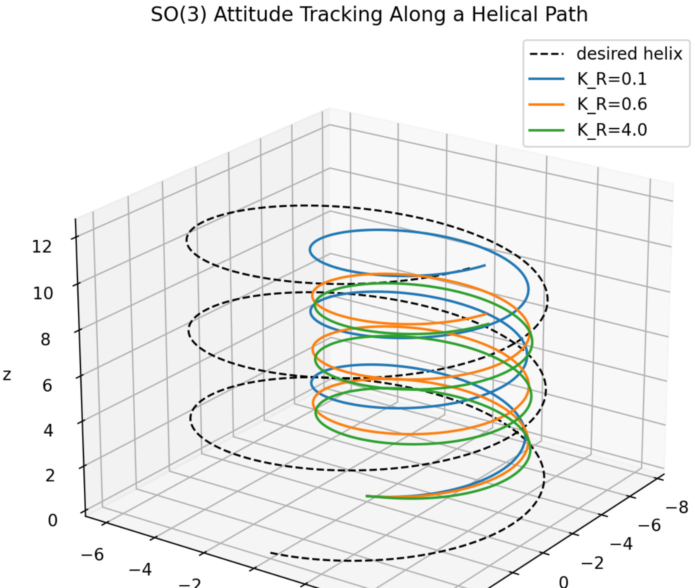
- Apply PD feedback on the geometric errors:

SO(3) PD control law

$$\omega = R^T R_d \, \omega_d - K_R \, e_R - K_\omega \, e_\omega, \qquad K_R, K_\omega > 0$$

- $K_R \, e_R$: proportional term, drives $R \to R_d$
- $K_\omega \, e_\omega$: derivative term, damps angular velocity error
- Identical structure to Euclidean PD; only the error definition differs

SO(3) Helical Path Tracking



Outline

General LTI Systems with Input

- For multi-state plants with control input, the natural model is:

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

- Around an equilibrium (x^*, u^*) satisfying $0 = Ax^* + Bu^*$, let $\tilde{x} = x - x^*$:

$$\dot{\tilde{x}} = A\tilde{x} + B\tilde{u}$$

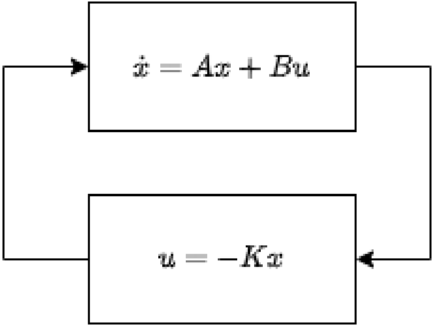
- We already know eigenvalues of A determine free $(u = 0)$ behavior
- The input u lets us reshape the closed-loop eigenvalues

State Feedback Design

- Plant: $\dot{x} = Ax + Bu$
- Choose state feedback: $u = -Kx, \quad K \in \mathbb{R}^{m \times n}$
- Closed-loop system:

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

- Goal: choose K so that $A - BK$ is Hurwitz



Eigenvalue Placement

- The closed-loop poles are roots of:

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

- Choose a desired polynomial $p^*(\lambda)$ with roots in the left half-plane
- Match coefficients of $p(\lambda)$ and $p^*(\lambda)$ to solve for K
- If (A, B) is controllable, any pole placement is achievable
- For 2×2 systems, two equations give two unknowns in K

Linearized Path-Tracking: State Feedback

- Linearizing the bicycle model around the reference gives $\dot{x} = Ax + Bu$:

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

- Apply $u = -Kx$ and place eigenvalues of $A - BK$ in left half-plane
- Lateral and longitudinal channels remain decoupled

Summary

- P-control: error $\rightarrow$ input; scalar eigenvalue $\lambda = -(a + K_P)$ governs convergence
- LTI systems: eigenvalues of A determine stability; A Hurwitz $\Rightarrow$ asymptotic stability
- PID: integral removes offset; derivative damps; gains place eigenvalues of error system
- PD on SO(3): geometric error e_R on rotation manifold; same law as Euclidean PD
- State feedback: place eigenvalues of $A - BK$ via gain matrix K
- Next lecture: Lyapunov stability for nonlinear systems and MPC for constrained control