Autonomous Robotics

Spring 2026

Abhishek Gupta, Siddhartha Srinivasa

TAs: Helen Wang, Sidharth Talia, Rohan Baijal, Christopher Tan

Recap

Computing control laws

We will only control steering angle;
fixed constant speed
As a result, no real control for along-track error
Some control laws will only minimize cross-track error, others will also minimize heading

$$u = K(e)$$

Bang-bang control

Simple control law - choose between hard left and hard right

$$u = \begin{cases} u_{max} & \text{if } e_{ct} < 0 \\ -u_{max} & \text{otherwise} \end{cases}$$

PID control

Select a control law that tries to drive error to zero (and keep it there)

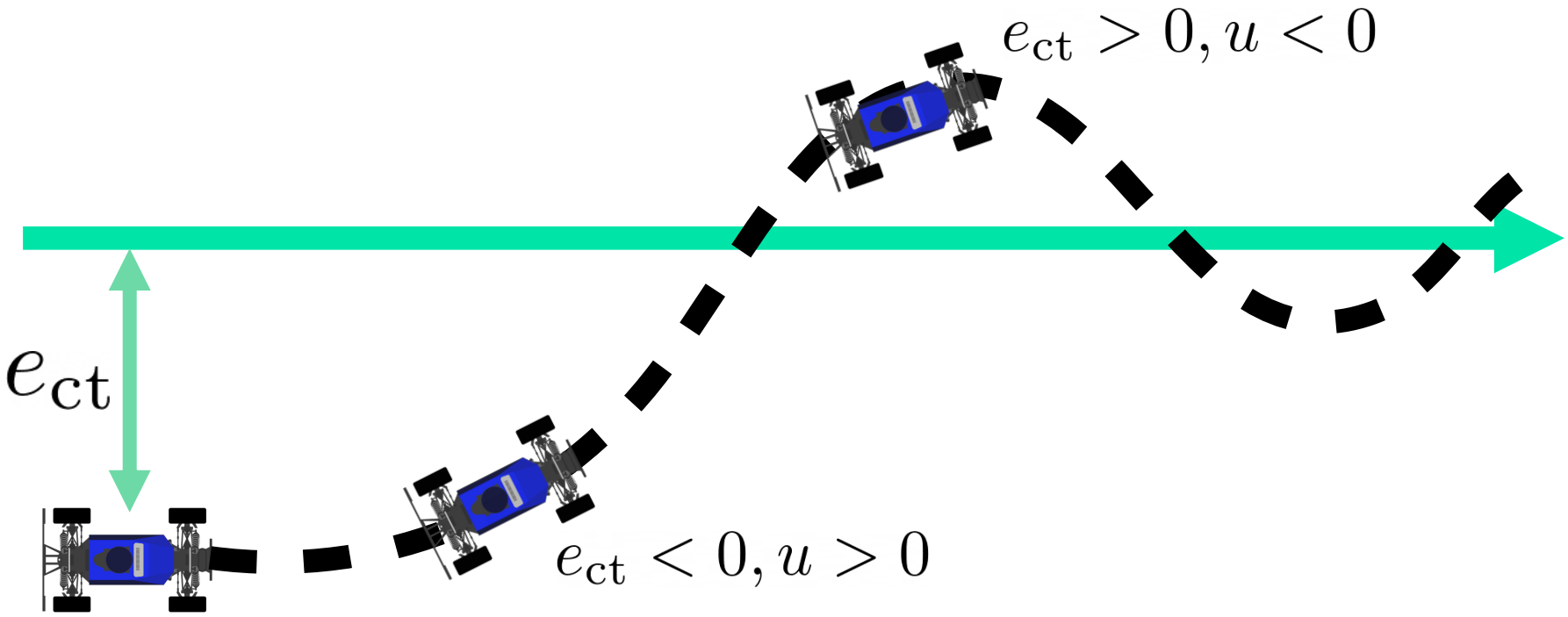
$$u = -\left(K_p e_{\mathrm{ct}} + K_i \int e_{\mathrm{ct}} dt + K_d \dot{e}_{\mathrm{ct}} \right)$$

PROPORTIONAL (PRESENT)

INTEGRAL (PAST)

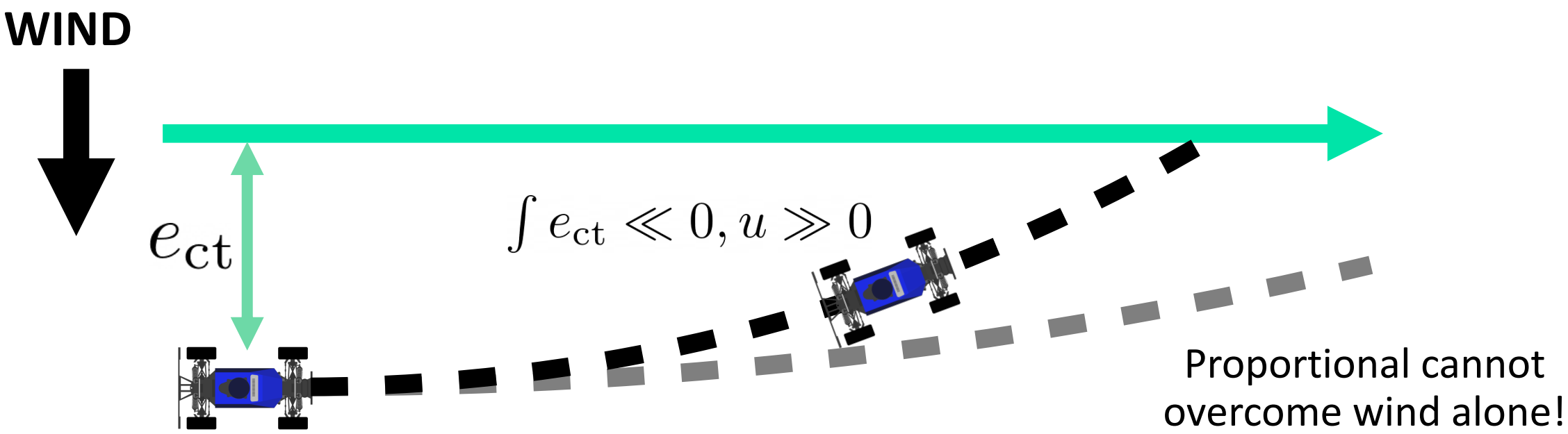
DERIVATIVE (FUTURE)

Proportional Control



$$u = -K_p e_{\mathrm{ct}}$$

Proportional Integral (PI) Control

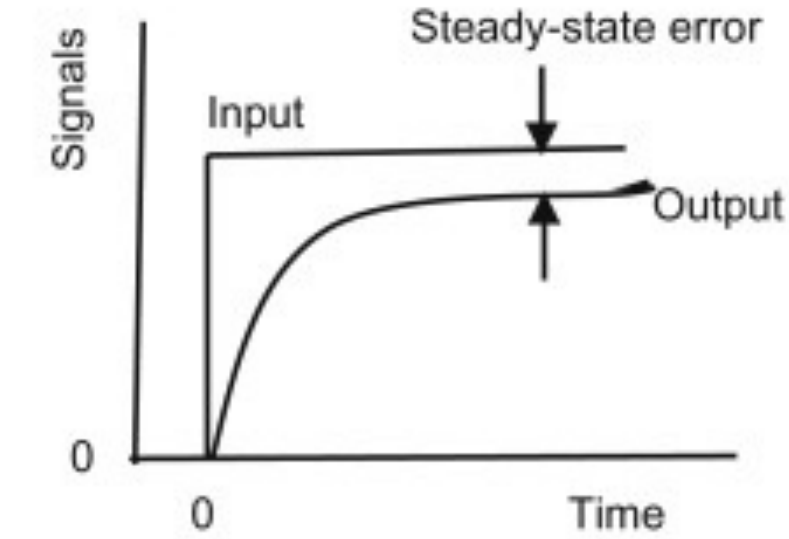


$$u = -\left(K_p e_{\mathrm{ct}} + K_i \int e_{\mathrm{ct}} dt\right)$$

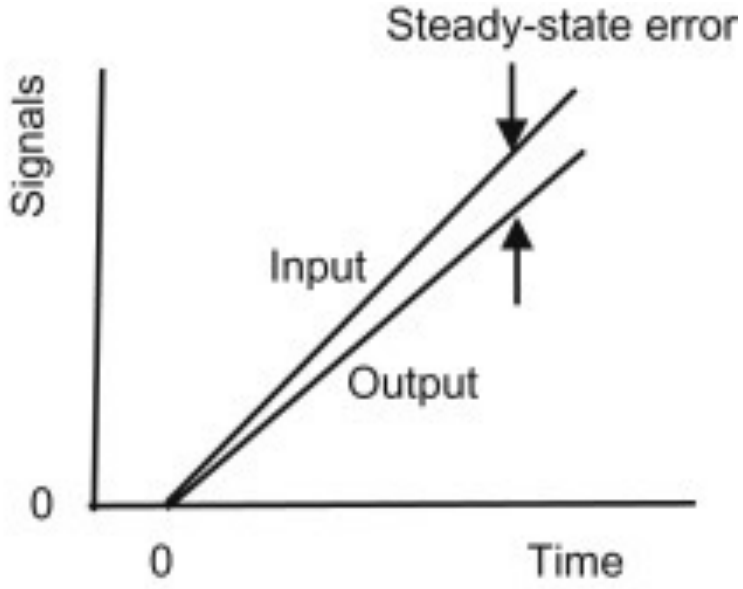
Proportional Integral (PI) Control

$$u = -\left(K_p e_{\mathrm{ct}} + K_i \int e_{\mathrm{ct}} dt\right)$$

Integral control gets rid of this term since the integral keeps growing



(a)



(b)

Lecture Outline

Recap

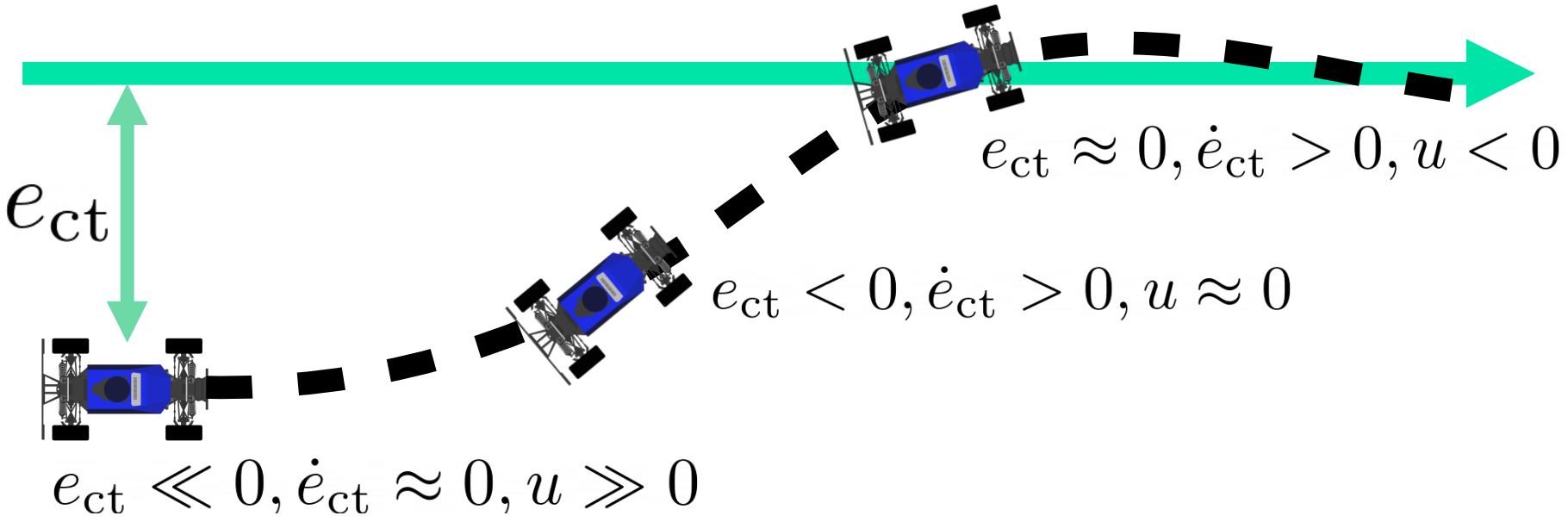
PID Control Wrap-up

Pure Pursuit

Optimal Control

Proportional Derivative (PD) Control

Apply the brakes when moving too fast! → converge to the steady state



$$u = -\left(K_p e_{\mathrm{ct}} + K_d \dot{e}_{\mathrm{ct}}\right)$$

How do you evaluate the derivative term?

Terrible way: Numerically differentiate error. Why is this a bad idea?

Smart way: Analytically compute the derivative of the cross track error

$$e_{ct} = -\sin(\theta_{ref})(x - x_{ref}) + \cos(\theta_{ref})(y - y_{ref})$$

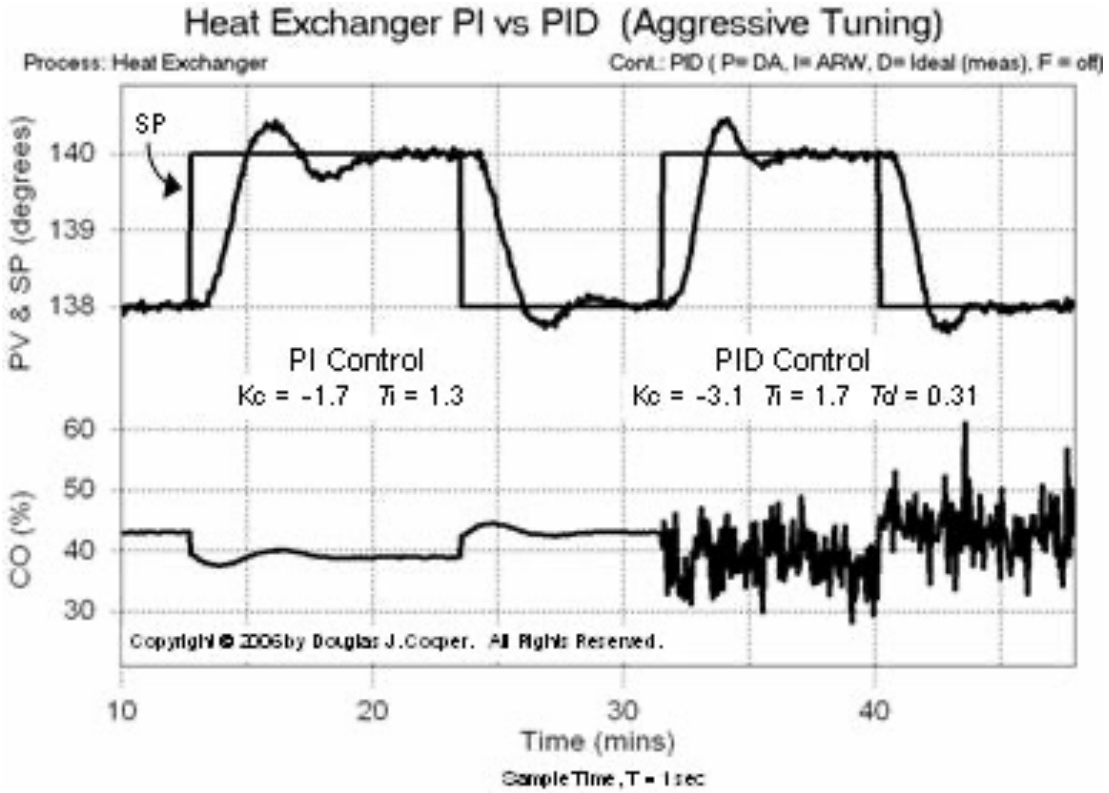
$$\begin{aligned}\dot{e}_{ct} &= -\sin(\theta_{ref})\dot{x} + \cos(\theta_{ref})\dot{y} \\ &= -\sin(\theta_{ref})V\cos(\theta) + \cos(\theta_{ref})V\sin(\theta) \\ &= V\sin(\theta - \theta_{ref}) = V\sin(\theta_e)\end{aligned}$$

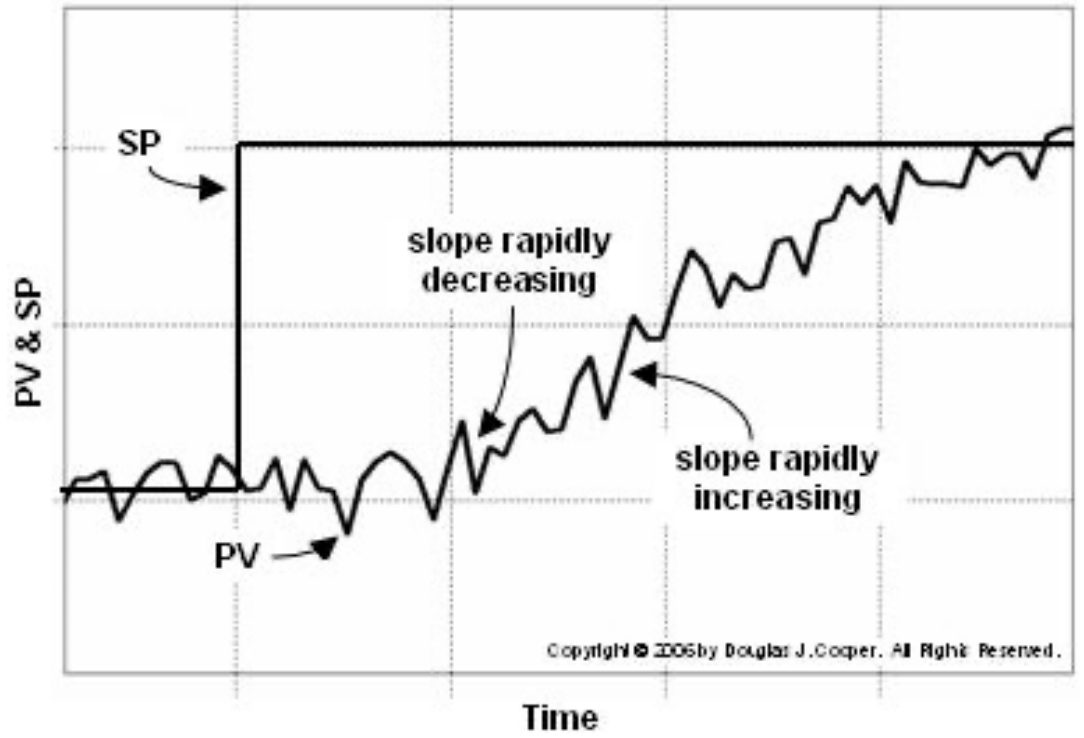
New control law! Penalize error in cross track and in heading

$$u = -\left(K_p e_{ct} + K_d V \sin\theta_e\right)$$

Challenges with using the derivative term

Noise can lead to wildly changing derivatives – leading to huge control variations





PID Intuition

$$u = -\left(K_p e_{\mathrm{ct}} + K_i \int e_{\mathrm{ct}} dt + K_d \dot{e}_{\mathrm{ct}} \right)$$

PROPORTIONAL (PRESENT)

INTEGRAL (PAST)

DERIVATIVE (FUTURE)

Proportional: minimize the current error!

Integral: if I'm accumulating error, try harder!

Derivative: if I'm going to overshoot, slow down!

Tuning PID controllers

$$u = -\left(K_p e_{\mathrm{ct}} + K_i \int e_{\mathrm{ct}} dt + K_d \dot{e}_{\mathrm{ct}} \right)$$

PROPORTIONAL (PRESENT)

INTEGRAL (PAST)

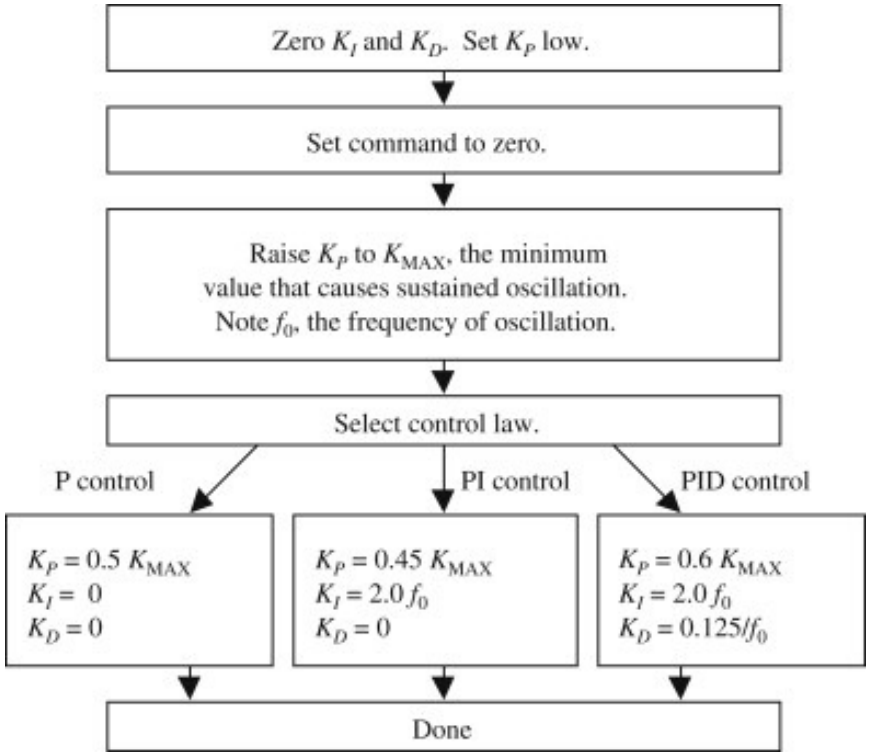
DERIVATIVE (FUTURE)

How do you set the K_p, K_i, K_d constants for a particular system?

Tuning PID controllers: Ziegler-Nichols

Heuristic/empirical method for computing K_p, K_i, K_d

$$u = -\left(K_p e_{\mathrm{ct}} + K_i \int e_{\mathrm{ct}} dt + K_d \dot{e}_{\mathrm{ct}} \right)$$



See how the system responds to proportional gain

Adjust integral and proportional accordingly

Lecture Outline

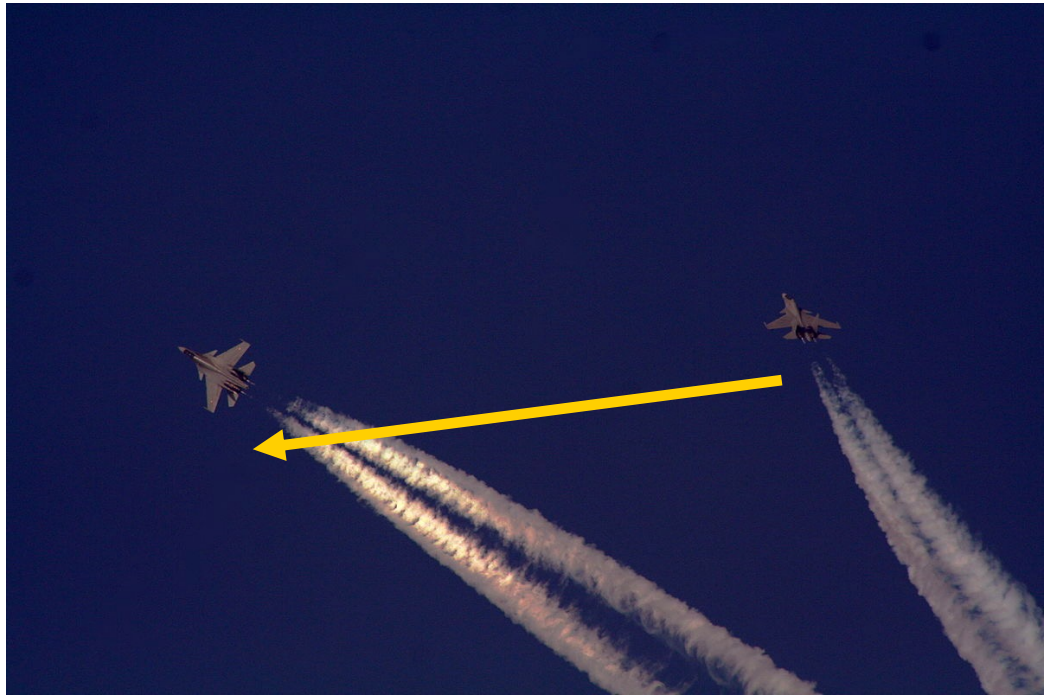
Recap

PID Control Wrap-up

Pure Pursuit

Optimal Control

Pure Pursuit Control



Aerial combat in which aircraft pursues another aircraft by pointing its nose directly towards it

Similar to carrot on a stick!

Rationale: Controller should leverage model!

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

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

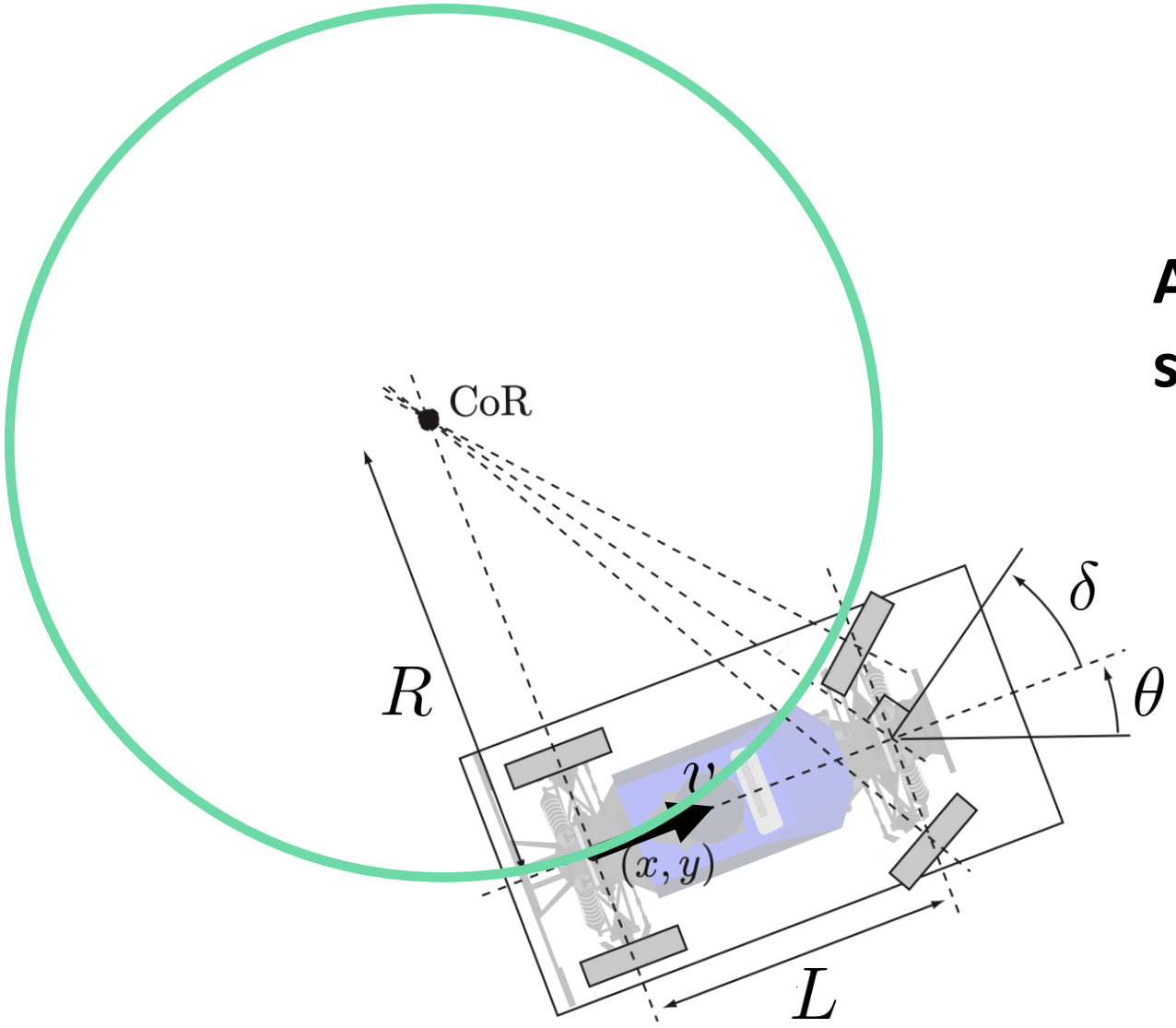
$$\dot{\theta} = \omega = \frac{v}{R} = \frac{v\tan\delta}{L}$$

PID control doesn't directly utilize the fact that we know the kinematic car model

Key Idea:

The car is always moving
in a circular arc

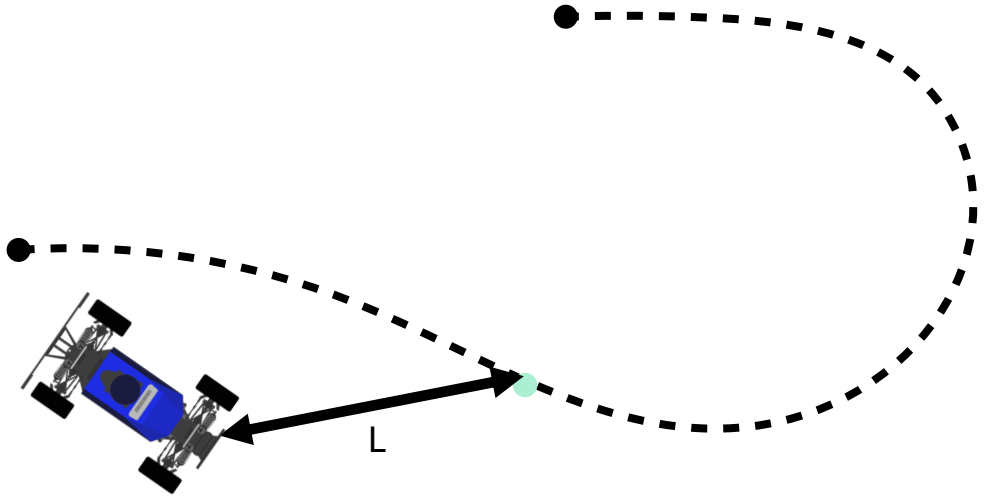
Pure Pursuit Controller



Assume the car is moving with fixed steering angle

COULTER, 1992

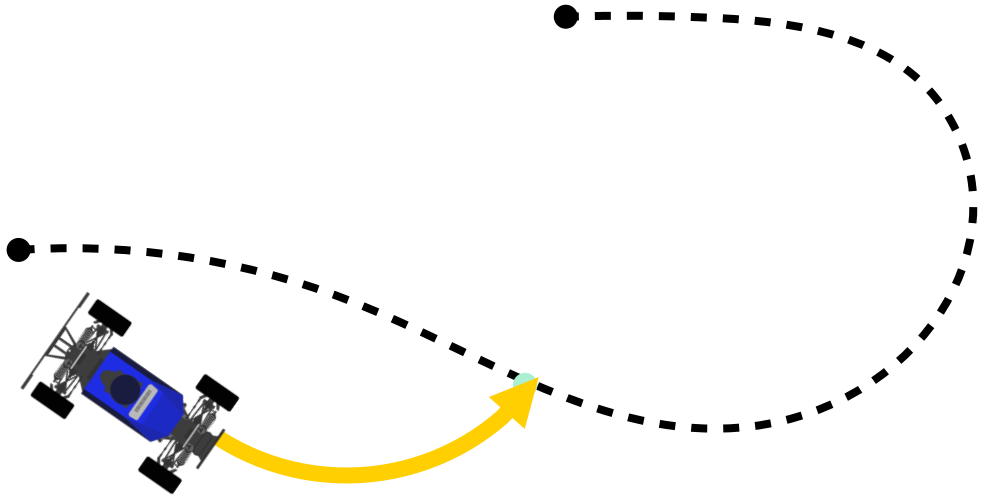
Consider a reference at a lookahead distance



$$\left\| \begin{bmatrix} x \\ y \end{bmatrix} - \begin{bmatrix} x_{ref} \\ y_{ref} \end{bmatrix} \right\| = L$$

Problem: Can we solve for a steering angle that guarantees that the car will pass through the reference?

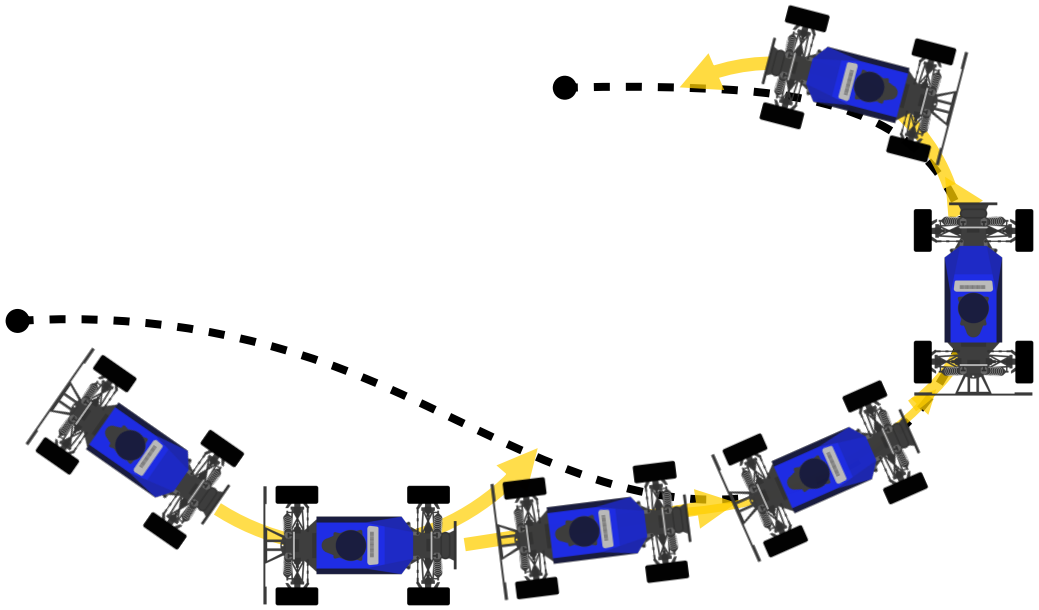
Solution: Compute a circular arc



We can always solve for a arc that
passes through a lookahead point

Note: As the car moves forward, the point keeps moving

Pure pursuit: Keep chasing looakahead



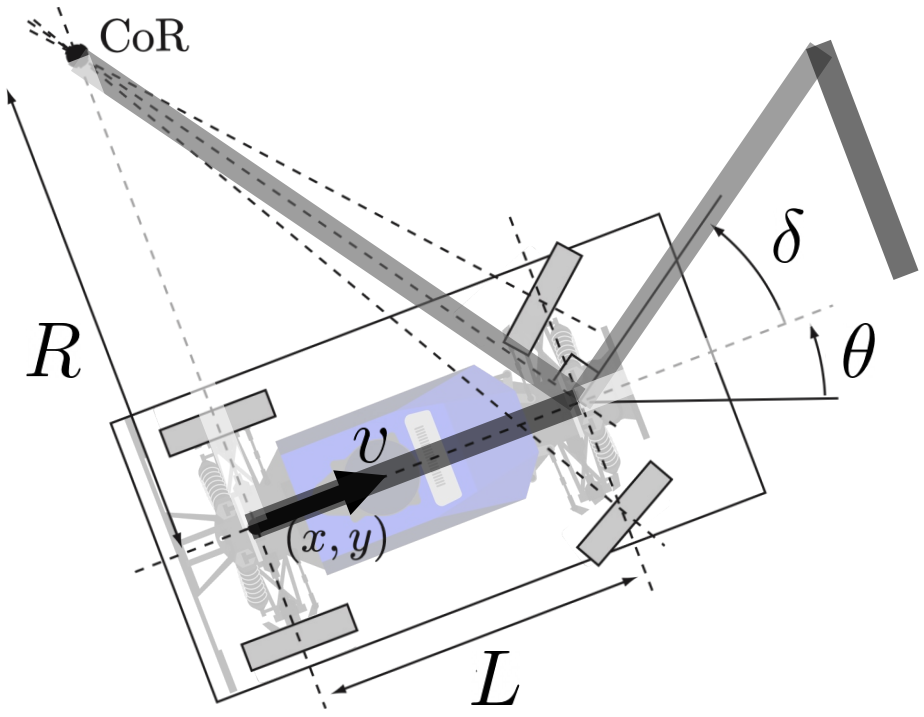
1. Find a lookahead and compute arc

2. Move along the arc

3. Go to step 1

Equations of Motion

RECALL



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

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

$$\dot{\theta} = \omega = \frac{v}{R} = \frac{v\tan\delta}{L}$$

$$\tan\delta = \frac{L}{R} \rightarrow R = \frac{L}{\tan\delta}$$

Kinematic Car Model

RECALL

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

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

X-COORDINATE

Y-COORDINATE

HEADING

$$\begin{bmatrix} v \\ \delta \end{bmatrix}$$

SPEED

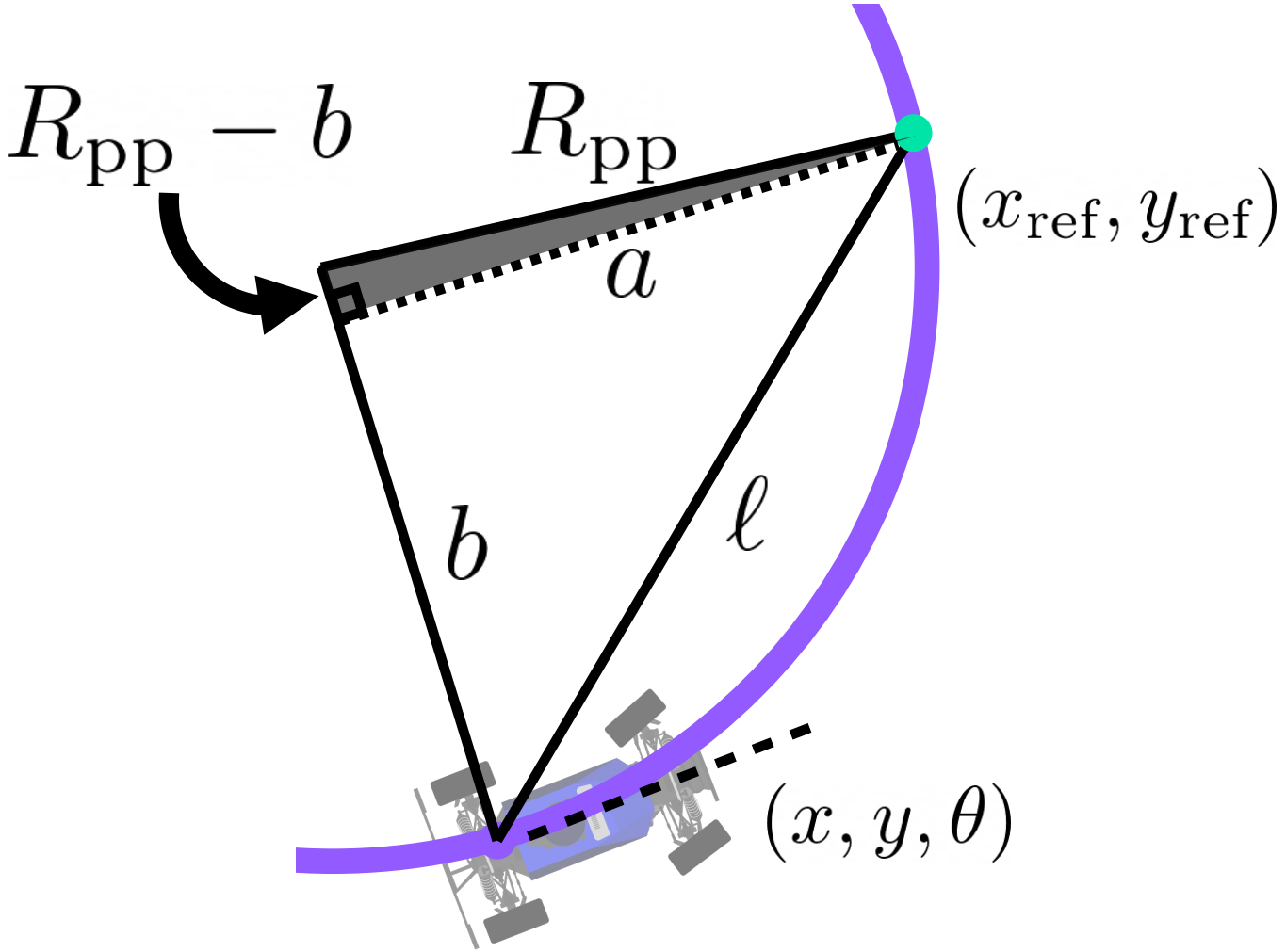
STEERING ANGLE

$$\begin{bmatrix} 1 \\ u \end{bmatrix}$$

Computing the Arc Radius

$$(R_{\text{pp}} - b)^2 + a^2 = R_{\text{pp}}^2$$

$$R_{\text{pp}} = \frac{a^2 + b^2}{2b}$$

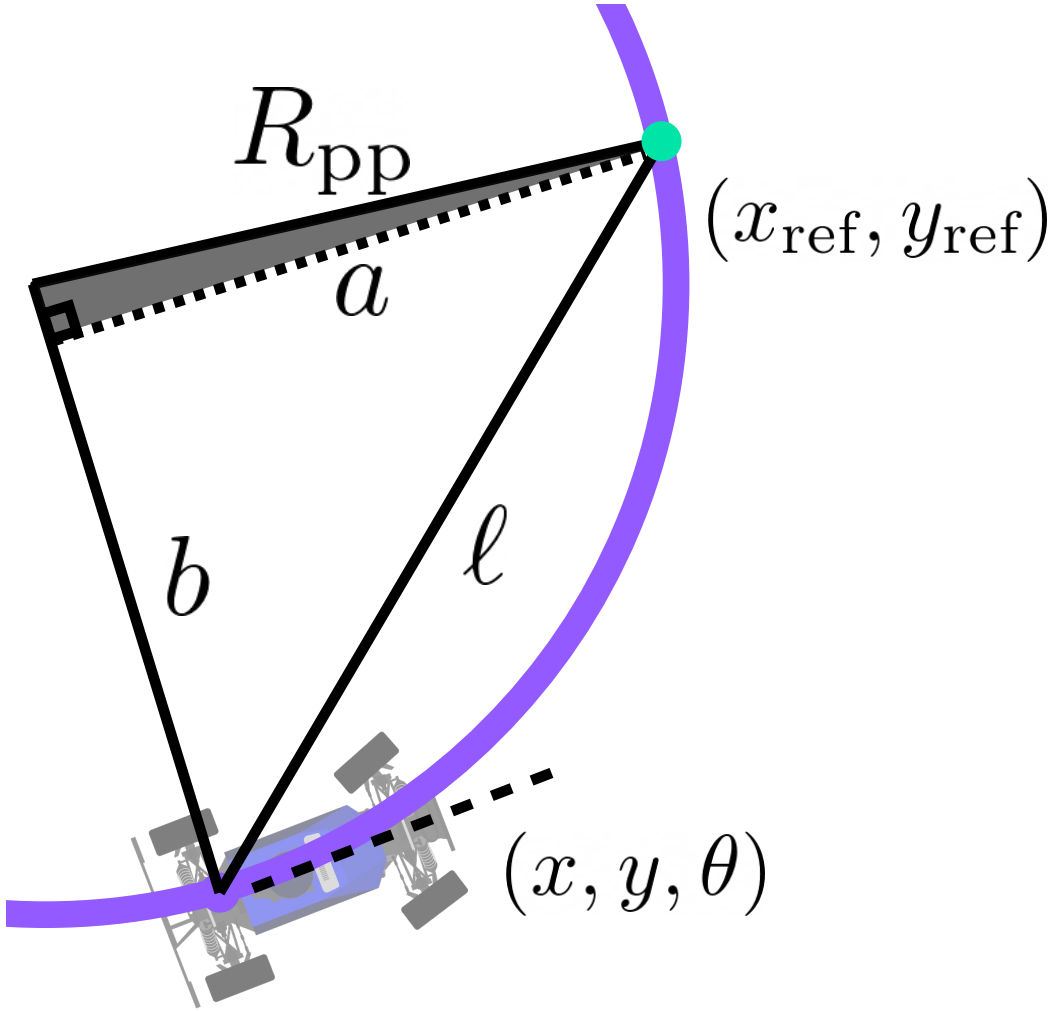


Computing the Arc Radius

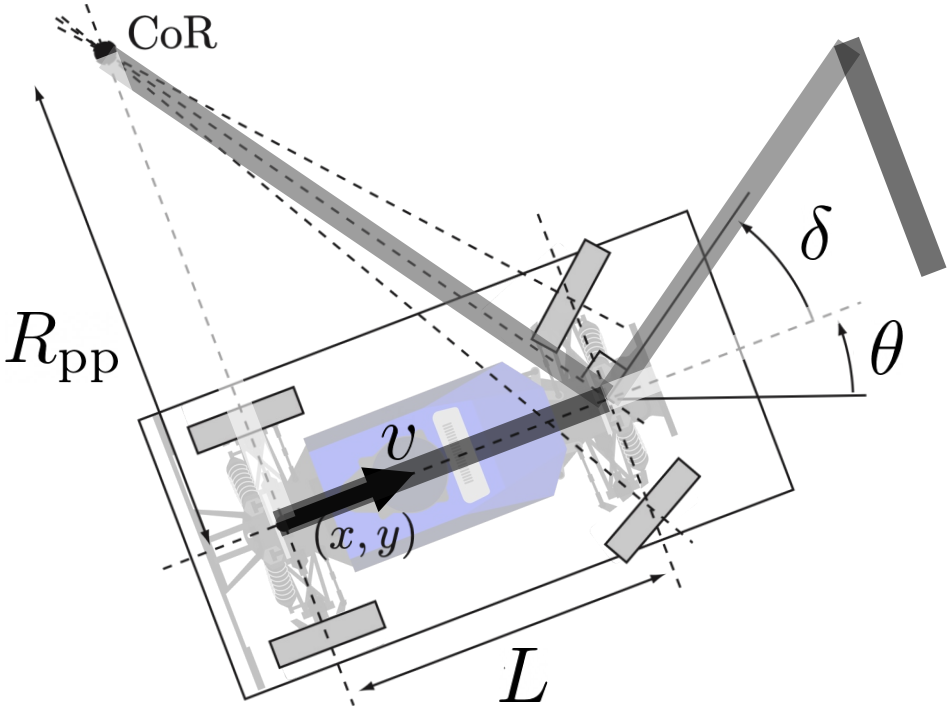
$$R_{\text{pp}} = \frac{a^2 + b^2}{2b}$$

$$\begin{bmatrix} a \\ b \end{bmatrix} = R(-\theta) \left(\begin{bmatrix} x_{\text{ref}} \\ y_{\text{ref}} \end{bmatrix} - \begin{bmatrix} x \\ y \end{bmatrix} \right)$$

Different than cross-track error (this is ref. position in robot frame; vice versa for cross-track error)



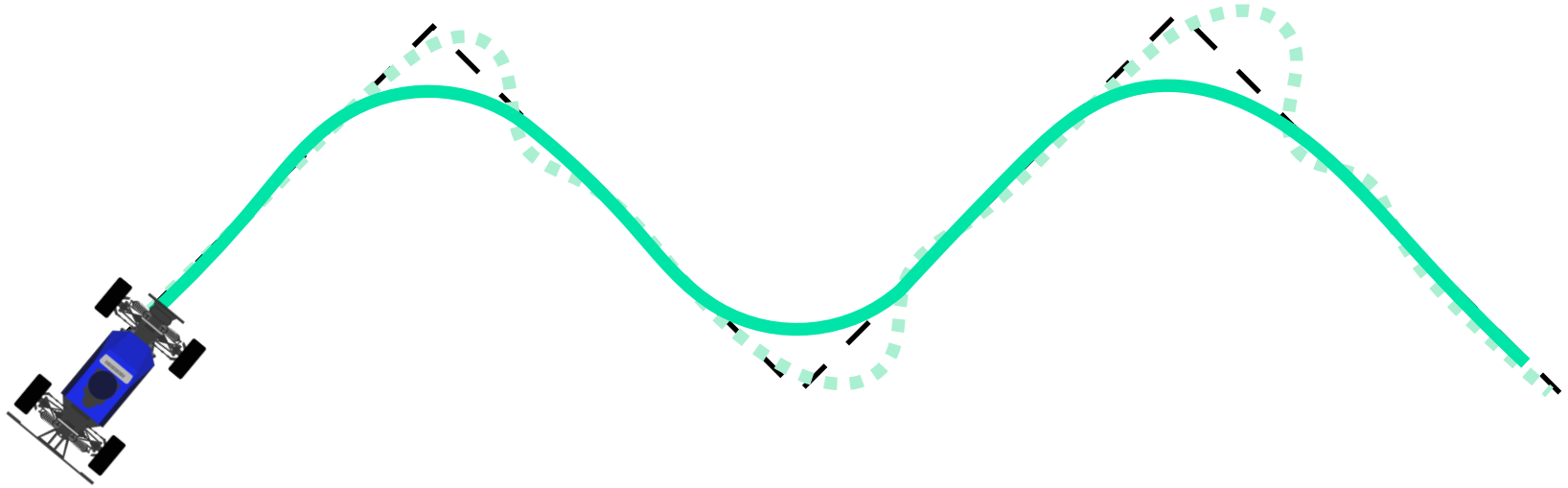
Computing the Steering Angle



$$R_{\text{pp}} = \frac{a^2 + b^2}{2b}$$

$$\tan\delta = \frac{L}{R_{\text{pp}}}$$

Question: How do I choose L?



Controller Design Decisions

1. Get a reference path/trajectory to track
2. Pick a reference state from the reference path/trajectory
3. Compute error to reference state
4. Compute control law to minimize error

Option 1:
Bang-bang control

Option 2:
PID control

Option 3:
Pure-pursuit control

Are we done?

Lecture Outline

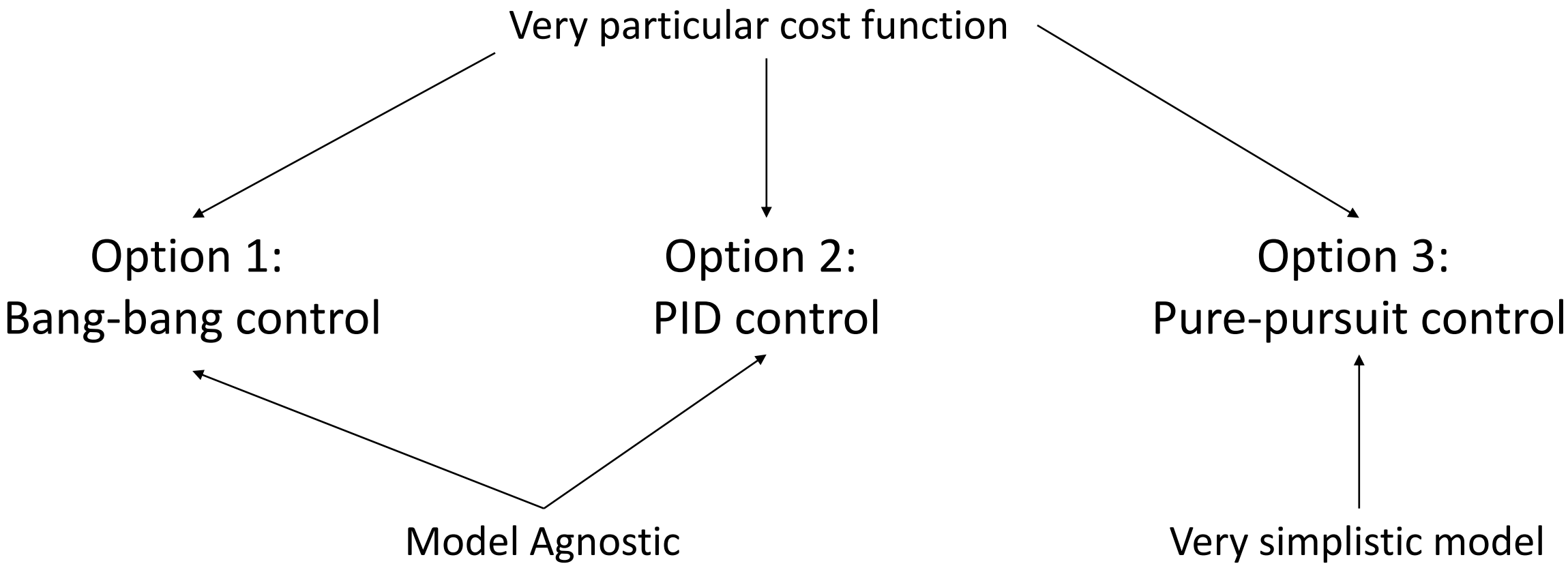
Recap

PID Control Wrap-up

Pure Pursuit

Optimal Control

Controller Design Decisions



Control as an Optimization Problem

For a sequence of H control actions

1. Use model to predict consequence of actions (i.e., H future states)
2. Evaluate the cost function

Compute optimal sequence of H control actions (minimizes cost)

Generalized Problem: Optimal Control

- Minimize sum of costs, subject to dynamics and other constraints

$$\min_{u_{1:T}} \sum_{t=1}^{T} c(x_t, u_t)$$

$$\text{s.t. } x_{t+1} = f(x_t, u_t)$$

Can be costs like smoothness, preferences, speed

Can be constraints like velocity/acceleration bounds

Linear Quadratic Regulator

Linear system (model)

$$x_{t+1} = Ax_t + Bu_t$$

Quadratic cost function to minimize

$$\sum_t x_t^\top Q x_t + u_t^\top R u_t$$

Linear System

Linear system (model)
Quadratic cost function to minimize

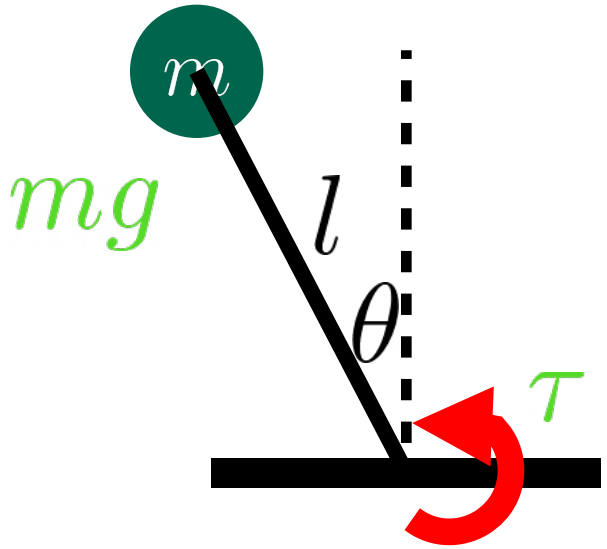
$$x_{t+1} = Ax_t + Bu_t$$
$$\sum_t x_t^\top Q x_t + u_t^\top R u_t$$

$$x_{t+1} = A x_t + B u_t$$

(N x 1) (N x N)(N x 1) (N x M)(M x 1)

STATE → NEXT STATE **CONTROL → NEXT STATE**

Example: Inverted Pendulum (Linear System)



$$mgl\sin\theta + \tau = ml^2\ddot{\theta}$$

$$\ddot{\theta} = \frac{g}{l}\sin\theta + \frac{\tau}{ml^2} \approx \frac{g}{l}\theta + \frac{\tau}{ml^2}$$

$$\begin{bmatrix}\theta_{t+1}\\ \dot{\theta}_{t+1}\end{bmatrix} = \begin{bmatrix}1 & \Delta t\\ \frac{g}{l}\Delta t & 1\end{bmatrix}\begin{bmatrix}\theta_t\\ \dot{\theta}_t\end{bmatrix} + \begin{bmatrix}0\\ \Delta t\end{bmatrix}\frac{\tau}{ml^2}$$

x_{t+1} A x_t B u_t

Quadratic Cost Function

Linear system (model)
Quadratic cost function to minimize

$$x_{t+1} = Ax_t + Bu_t$$
$$\sum_t x_t^\top Q x_t + u_t^\top R u_t$$

$$x_t^\top Q x_t$$

(1 x N)(N x N)(N x 1)

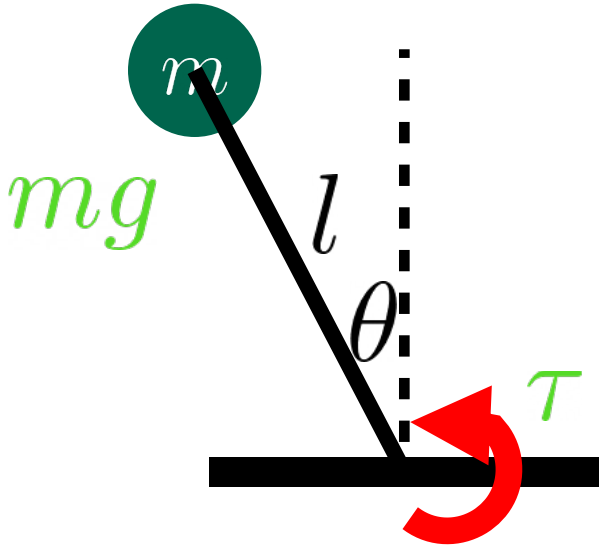
STATE COST

$$u_t^\top R u_t$$

(1 x M)(M x M)(M x 1)

CONTROL COST

Example: Inverted Pendulum (State Cost)



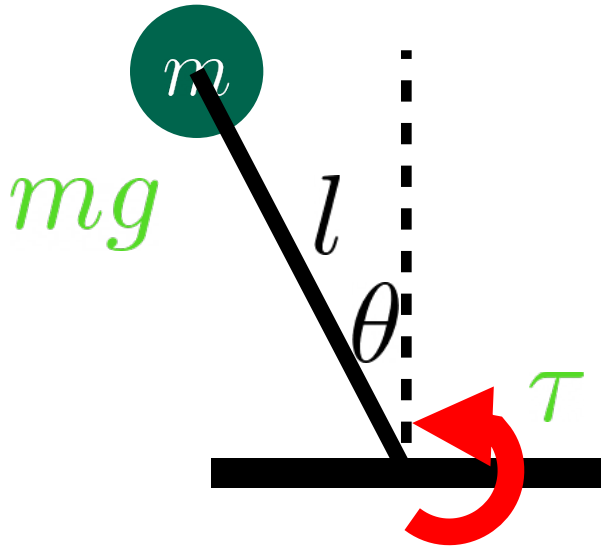
$$x_t^\top Q x_t \qquad \text{(QUADRATIC FORM)}$$

$$= \begin{bmatrix} \theta_t \\ \dot{\theta}_t \end{bmatrix}^\top \begin{bmatrix} Q_{\theta\theta} & Q_{\theta\dot{\theta}} \\ Q_{\dot{\theta}\theta} & Q_{\dot{\theta}\dot{\theta}} \end{bmatrix} \begin{bmatrix} \theta_t \\ \dot{\theta}_t \end{bmatrix}$$

$$= Q_{\theta\theta}\theta_t^2 + 2Q_{\theta\dot{\theta}}\theta_t\dot{\theta}_t + Q_{\dot{\theta}\dot{\theta}}\dot{\theta}_t^2$$

$$Q \succ 0 \leftrightarrow z^\top Q z > 0, \ \forall z \neq 0$$

Example: Inverted Pendulum (Control Cost)



$$u_t^\top R u_t \qquad \text{(QUADRATIC FORM)}$$

$$= \frac{\tau_t}{ml^2} \left[R_{\tau\tau} \right] \frac{\tau_t}{ml^2}$$

$$= R_{\tau\tau} \left(\frac{\tau_t}{ml^2} \right)^2$$

$$R \succ 0 \leftrightarrow z^\top R z > 0, \ \forall z \neq 0$$

Example: Inverted Pendulum