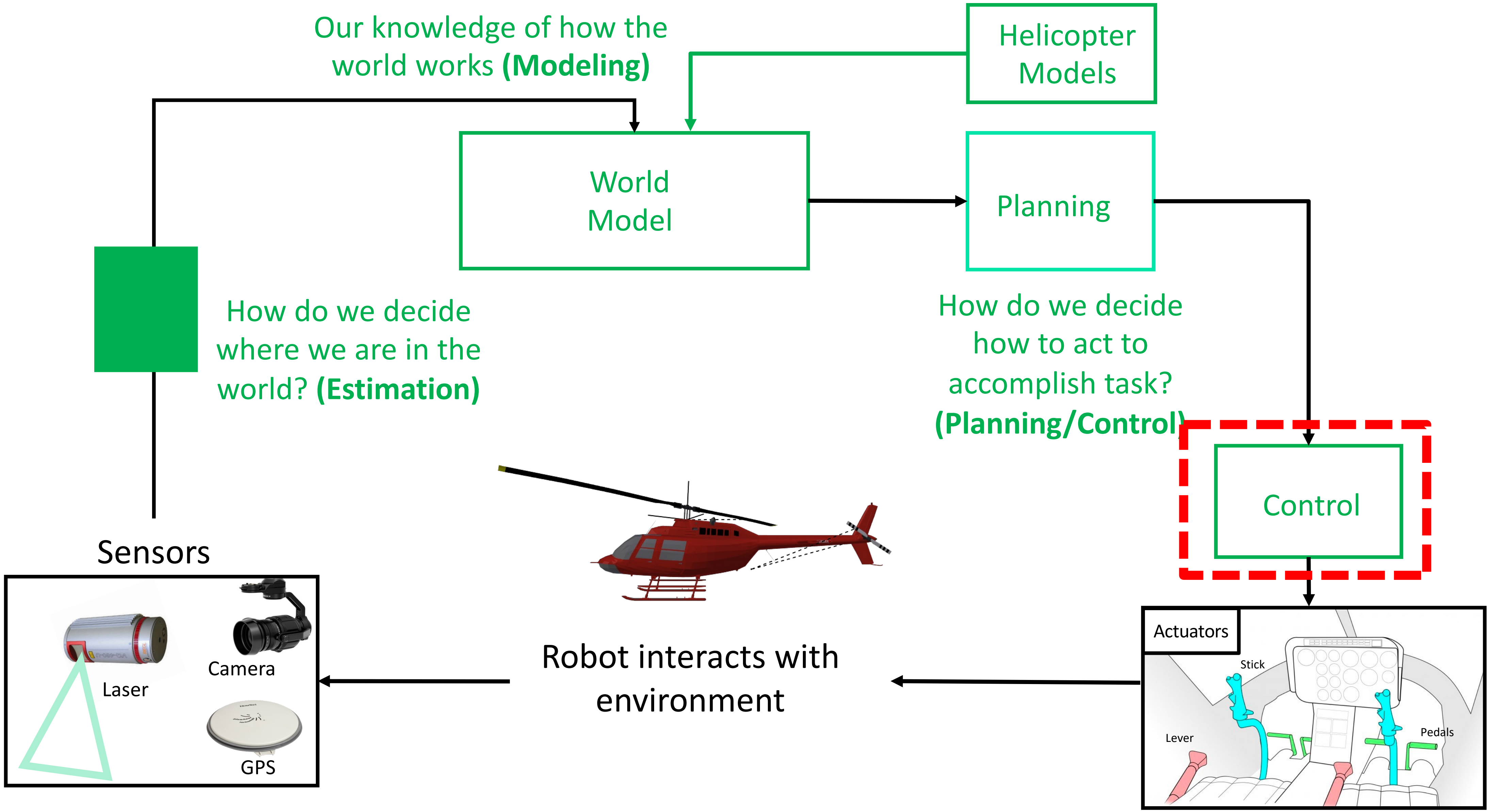
Autonomous Robotics
Spring 2026

Abhishek Gupta, Siddhartha Srinivasa

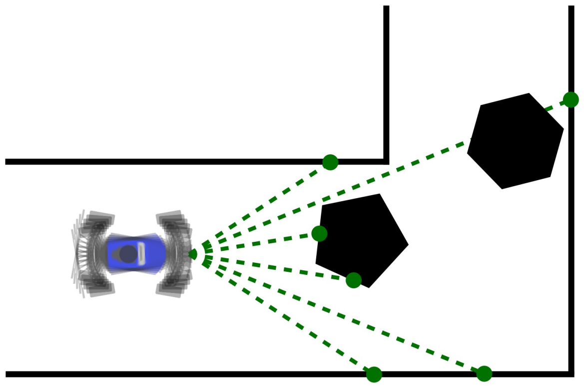
TAs: Helen Wang, Sidharth Talia, Rohan Baijal, Christopher Tan

Recap

Control in Sense-Plan-Act

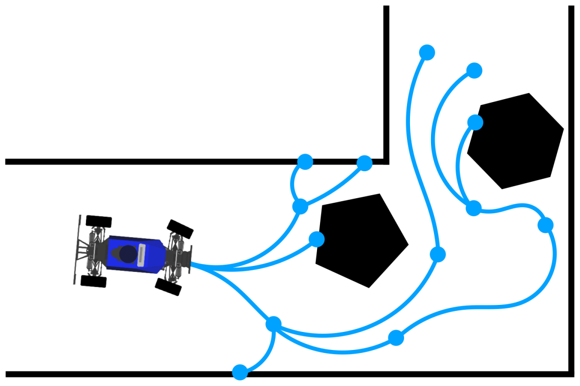


The Sense-Plan-Act Paradigm



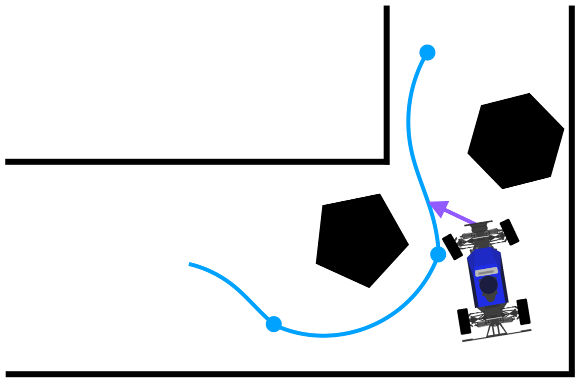
Estimate robot state

Solved over last 2.5 weeks



Plan sequence of motions

Assume to be solved for now

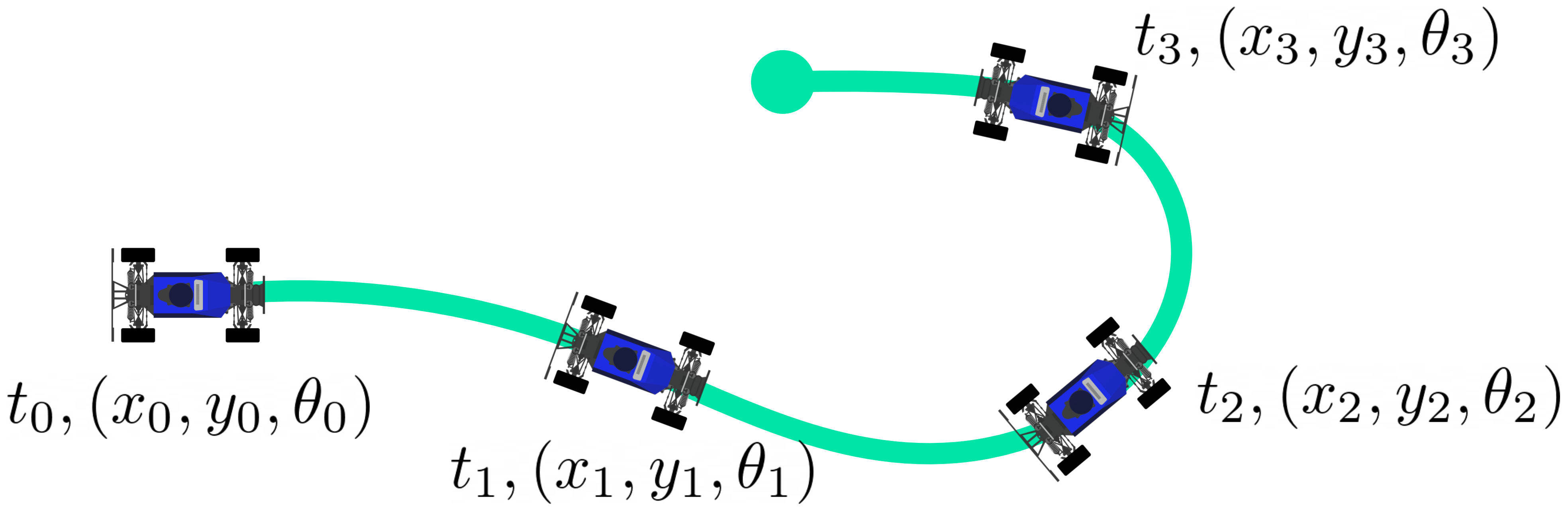


Control robot to follow plan

??

What is Control?

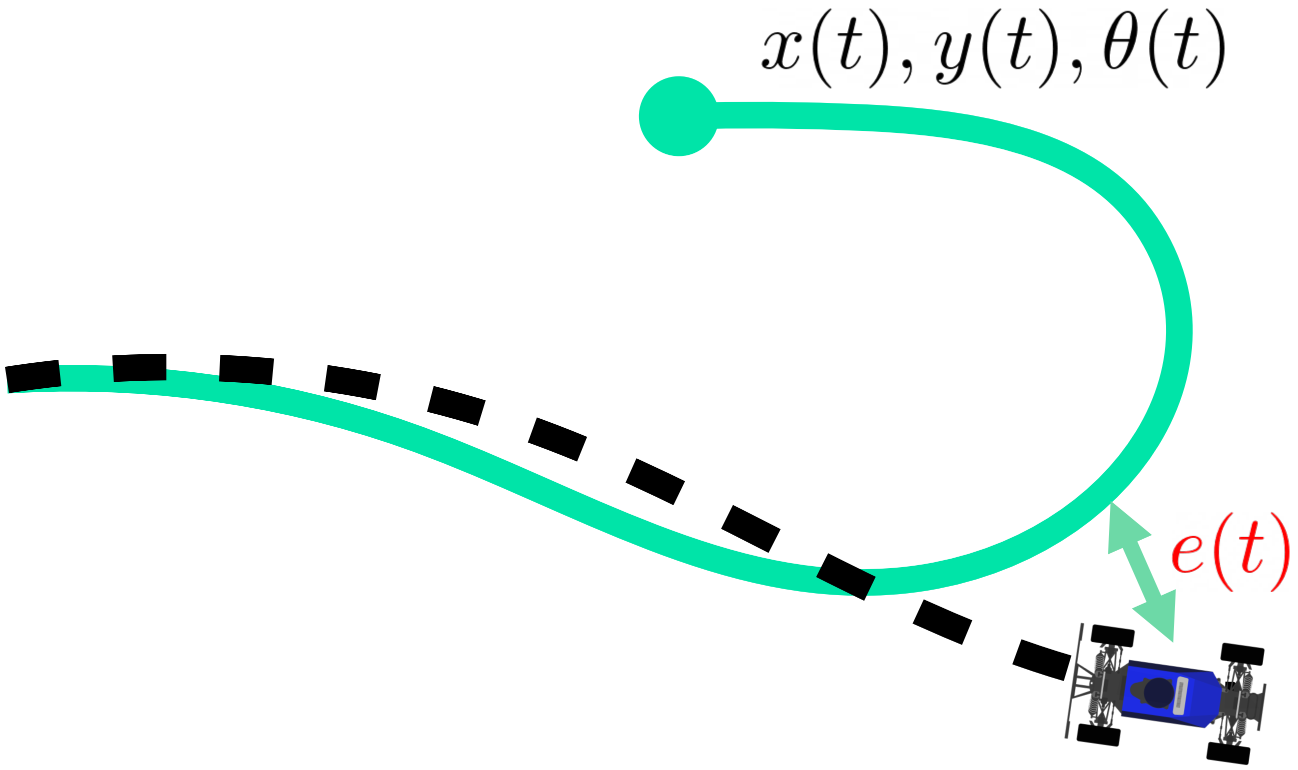
What is a Plan?



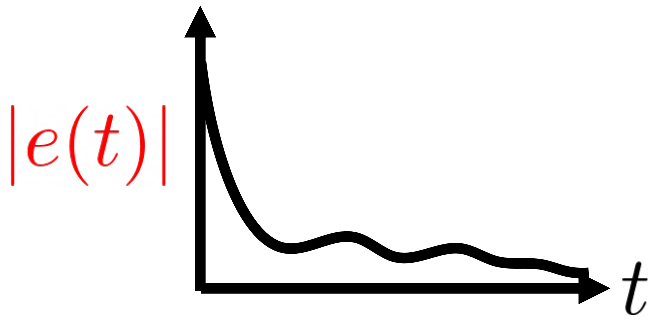
Can express this problem as **tracking a reference** trajectory

$$x(t), y(t), \theta(t)$$

Feedback Control

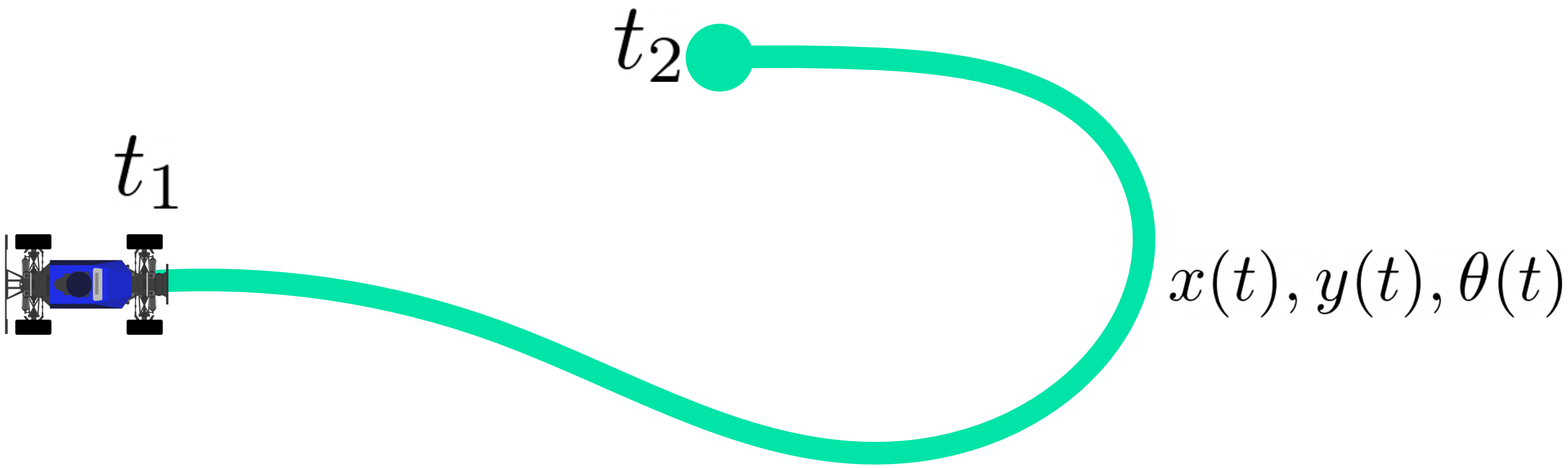


1. Measure error between reference and current state.
2. Take actions to minimize this error.



Ok let’s control racecars!

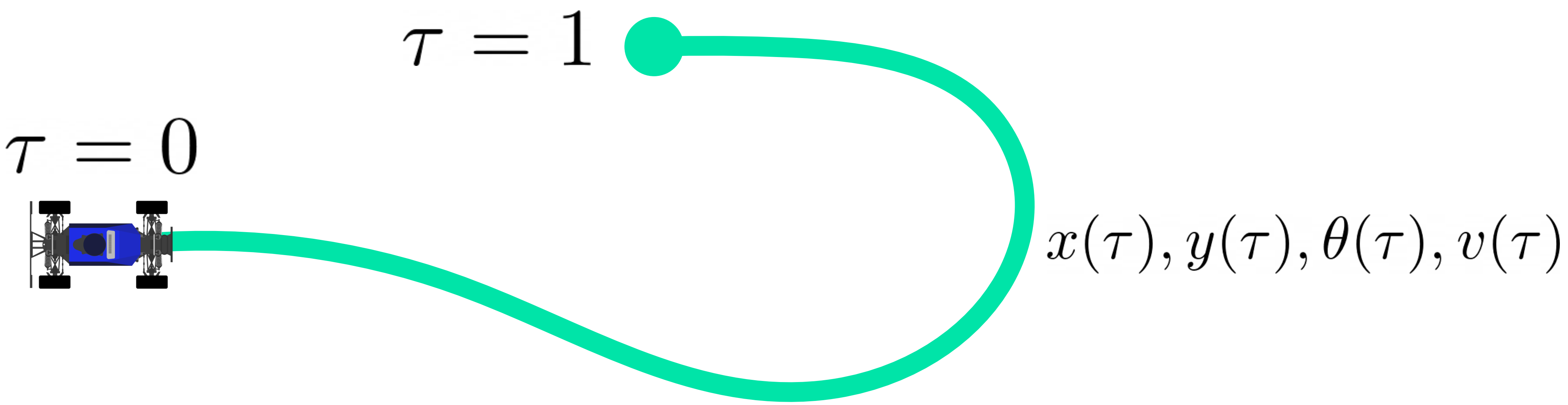
Reference Parameterizations



Option 1: **Time**-parameterized trajectory

Pro: Useful if we want the robot to respect time constraints
Con: Sometimes we only care about deviation from reference

Reference Parameterizations



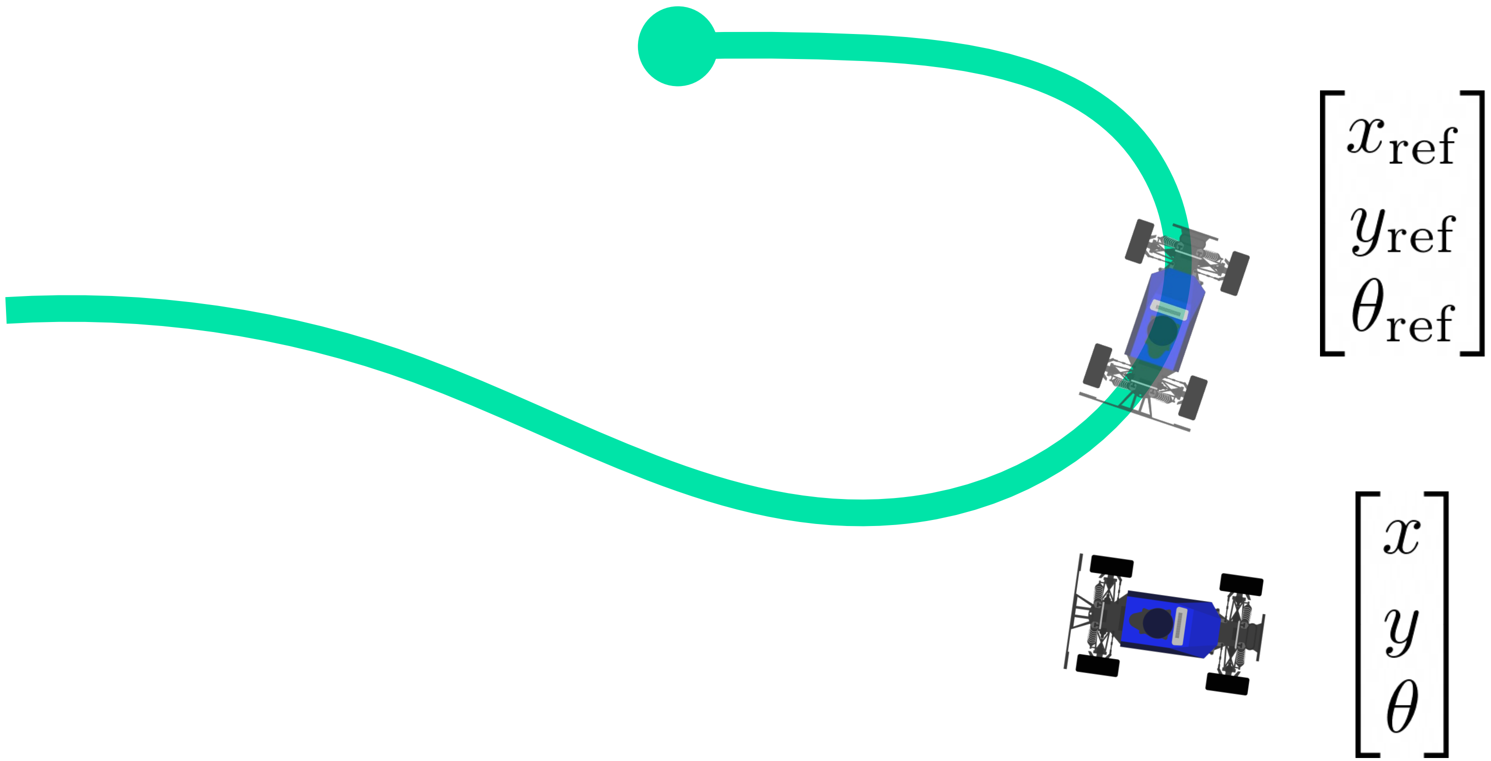
Option 2: **Index**-parameterized geometric path (untimed)

Pro: Useful for conveying shape for the robot to follow
Con: Can't control when robot will reach a point

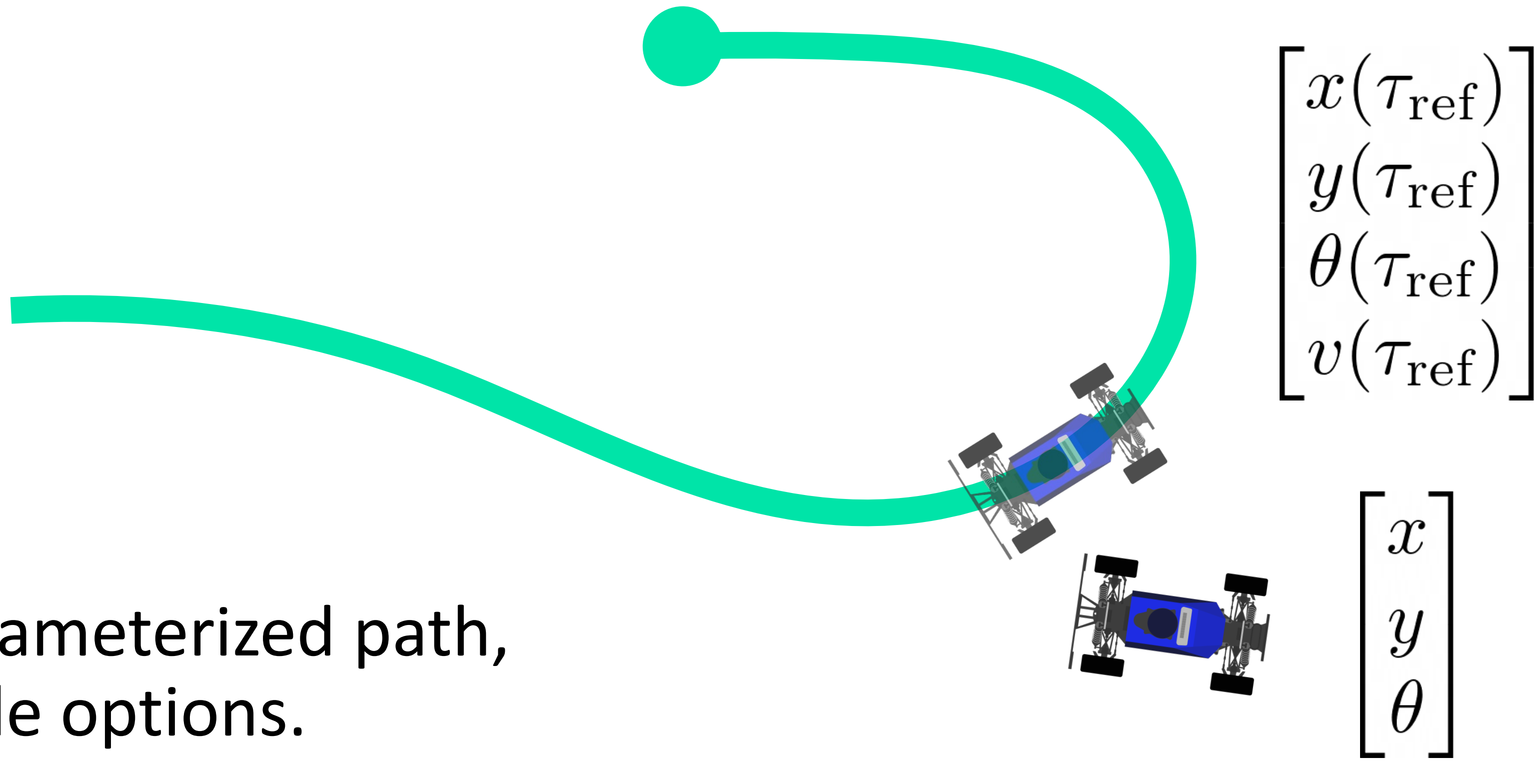
Controller Design Decisions

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

Step 2: Pick a reference (desired) state



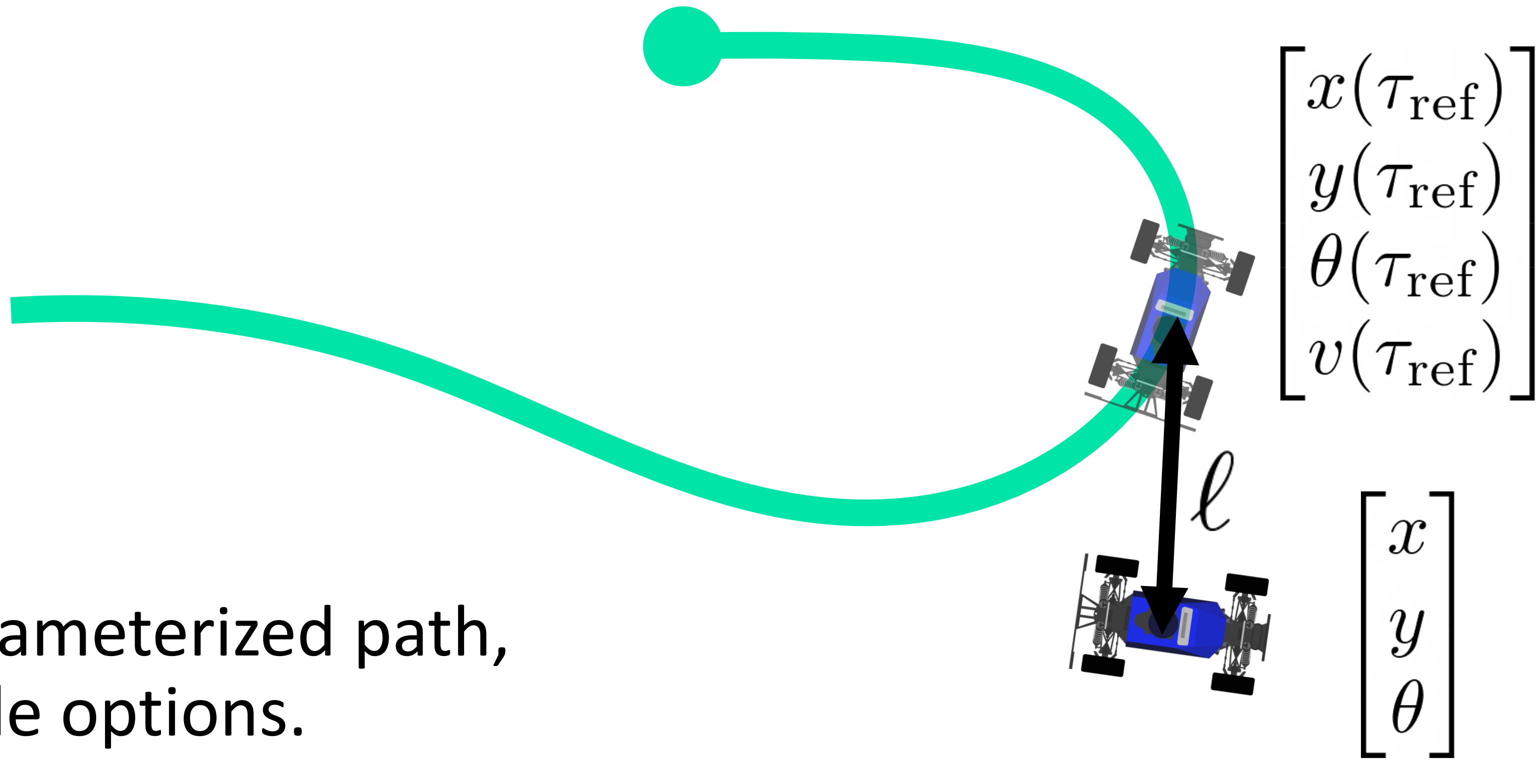
How do we choose a reference state?



For an **index**-parameterized path, there are multiple options.

Closest point $\quad \tau_{\text{ref}} = \arg\min_{\tau} \| \begin{bmatrix} x & y \end{bmatrix}^\top - \begin{bmatrix} x(\tau) & y(\tau) \end{bmatrix}^\top \|$

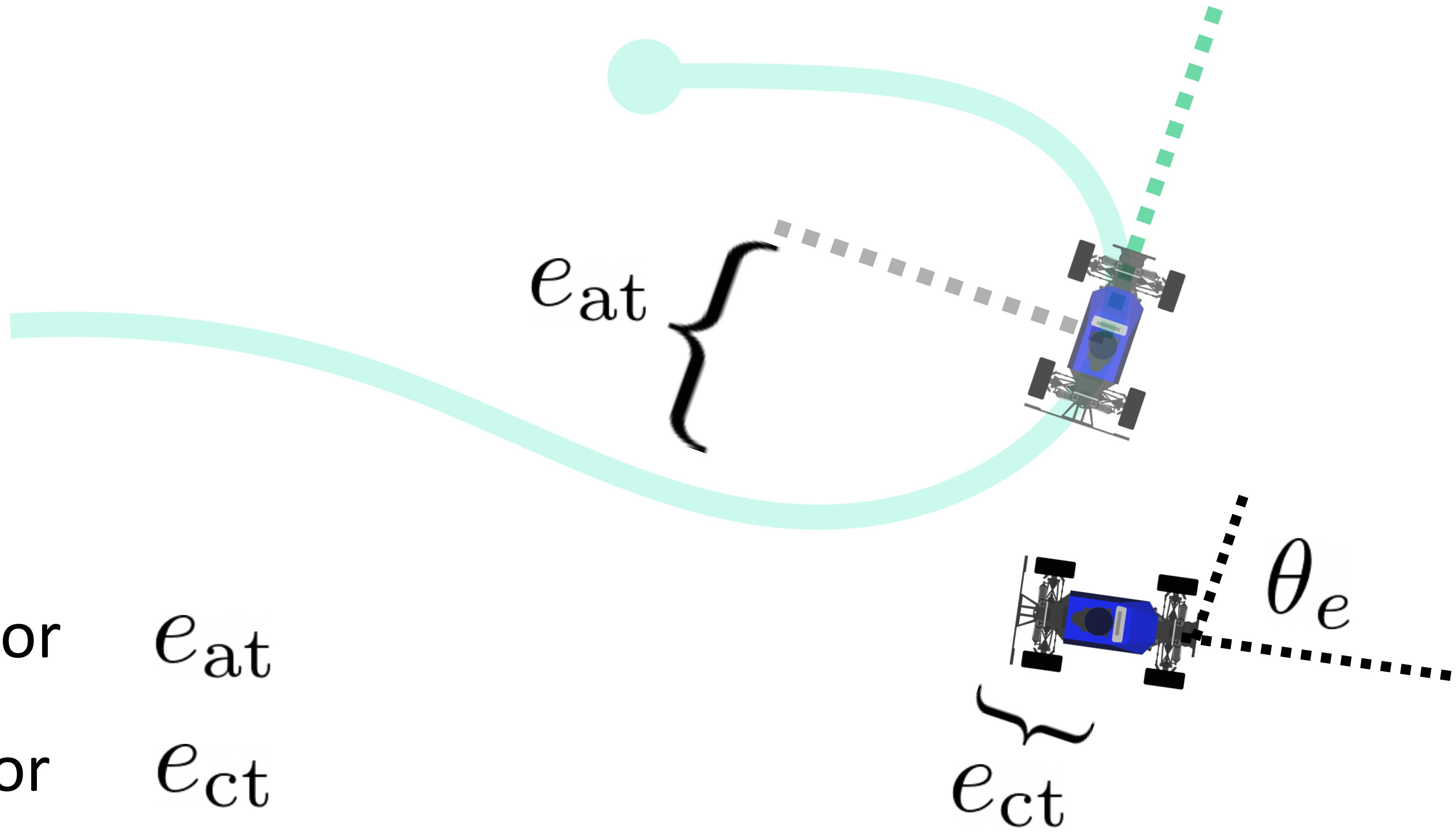
How do we choose a reference state?



For an **index**-parameterized path, there are multiple options.

Lookahead $$\tau_{\text{ref}} = \arg\min_{\tau} \left(\left\| \begin{bmatrix} x & y \end{bmatrix}^\top - \begin{bmatrix} x(\tau) & y(\tau) \end{bmatrix}^\top \right\| - \ell \right)^2$$

Step 3: Compute error to reference state

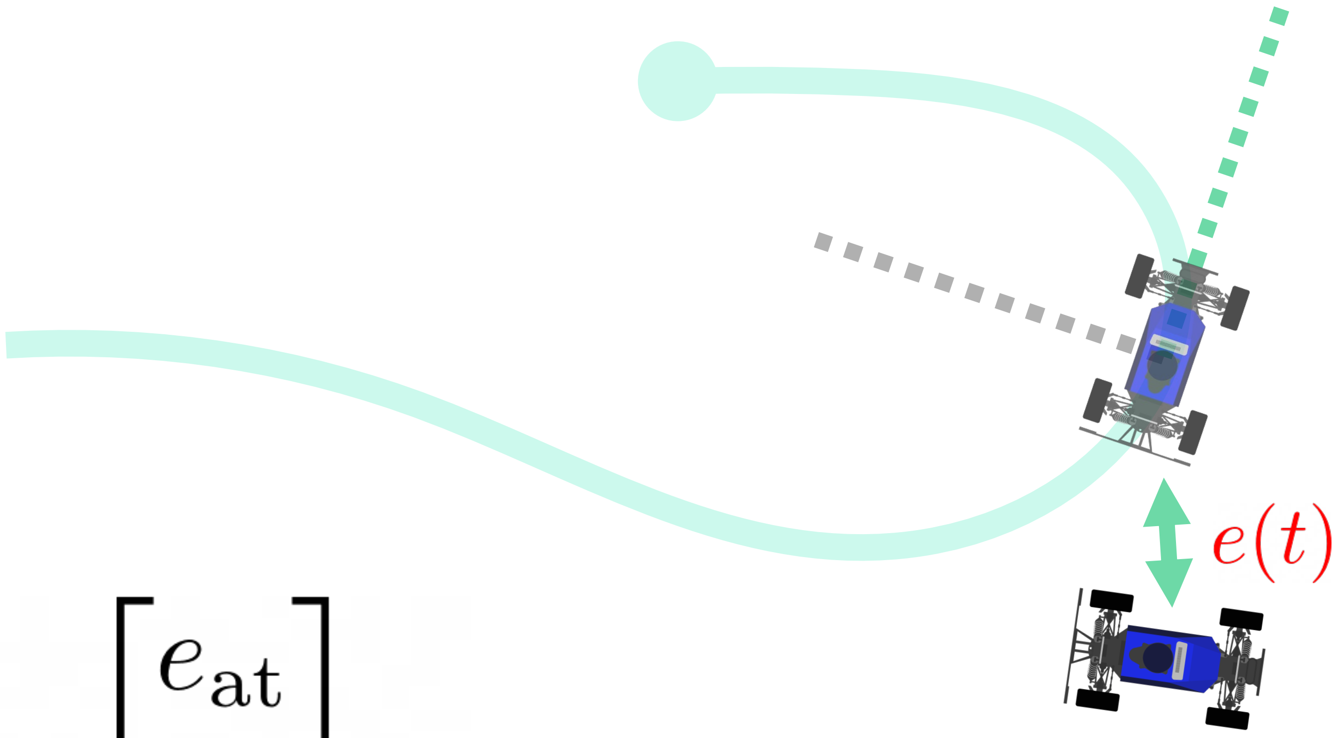


Along-track error e_{at}

Cross-track error e_{ct}

Heading error θ_e

Step 3: Compute error to reference state

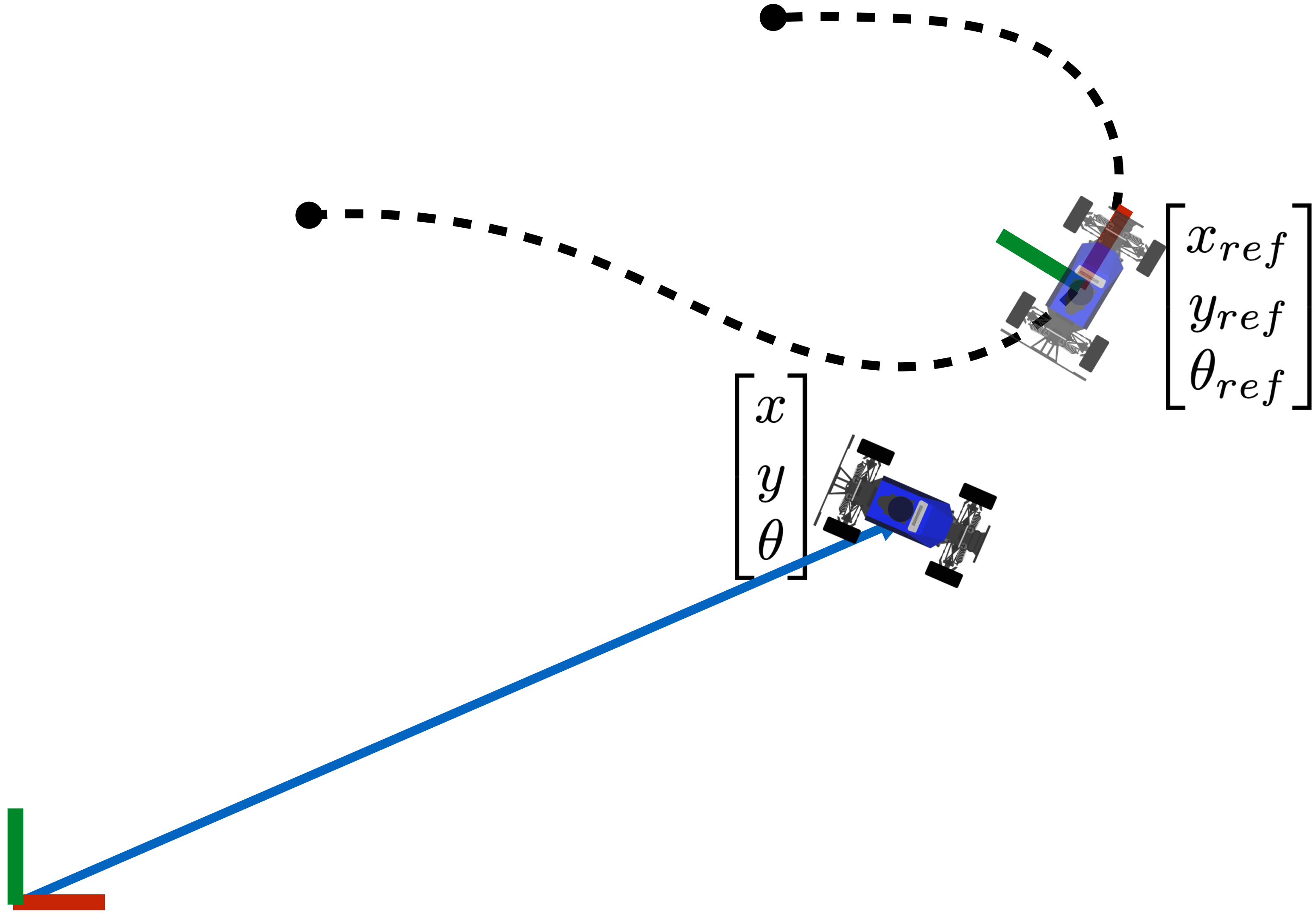


$$\begin{bmatrix} x \\ y \\ \theta \end{bmatrix}_G \rightarrow \begin{bmatrix} e_{\text{at}} \\ e_{\text{ct}} \\ \theta_e \end{bmatrix}_{\text{ref}}$$

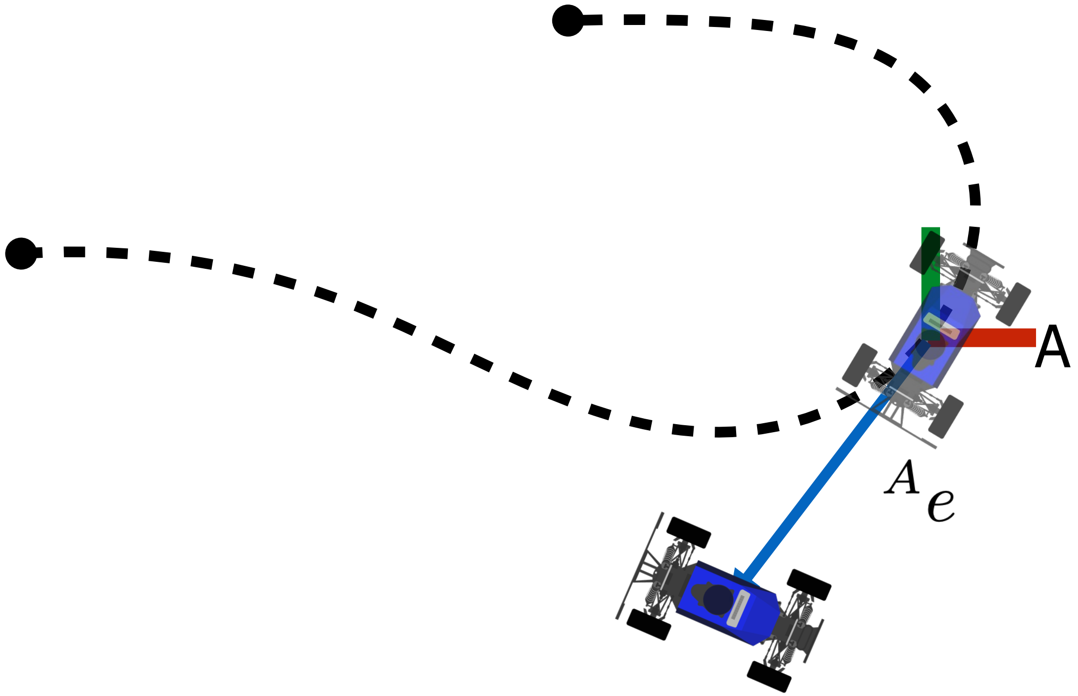
Aside: Rotation Matrices (Plane)

$$R = R_z(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$$

Step 3: Compute error to reference state



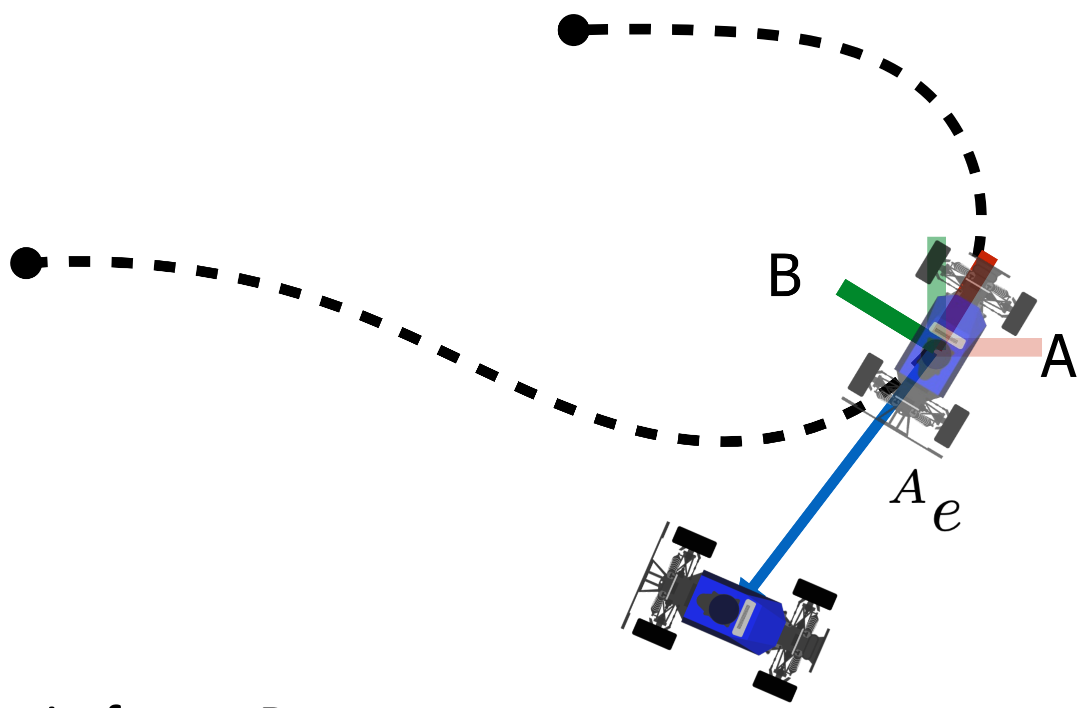
Step 3: Compute error to reference state



Position in frame A

$$^Ae = \begin{bmatrix} x \\ y \end{bmatrix} - \begin{bmatrix} x_{ref} \\ y_{ref} \end{bmatrix}$$

Step 3: Compute error to reference state

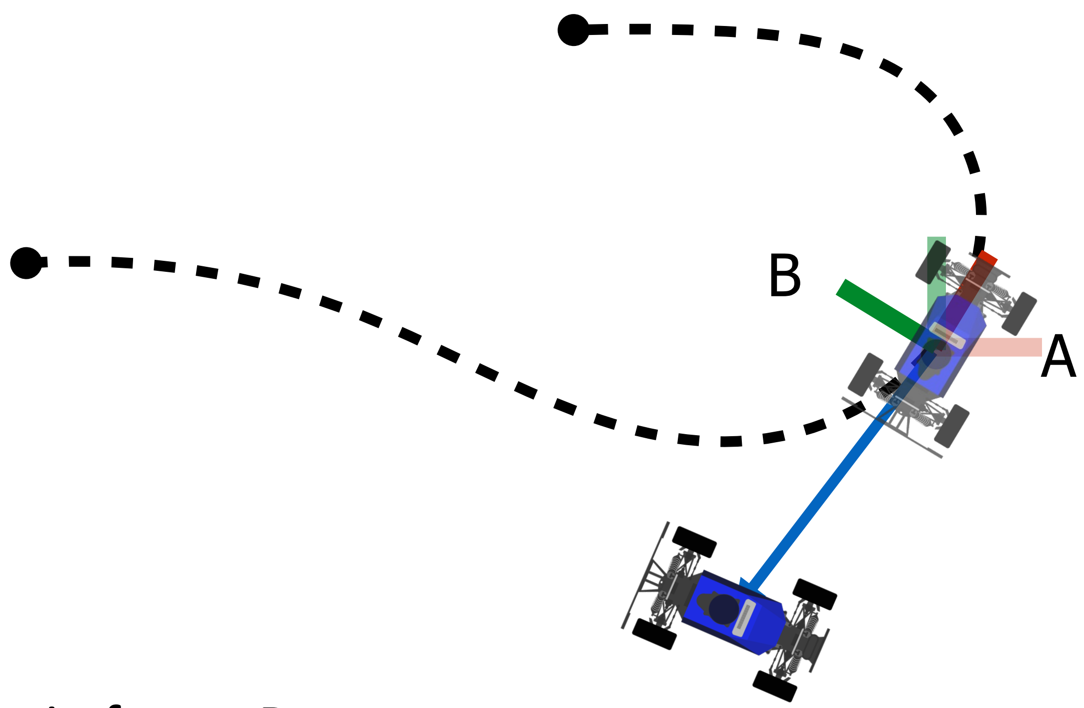


We want position in frame B

$$^{B}e = ^{B}_{A}R \;\; ^{A}e = R(-\theta_{ref}) \left(\begin{bmatrix} x \\ y \end{bmatrix} - \begin{bmatrix} x_{ref} \\ y_{ref} \end{bmatrix} \right)$$

(rotation of A w.r.t B) (rotation of A w.r.t B)

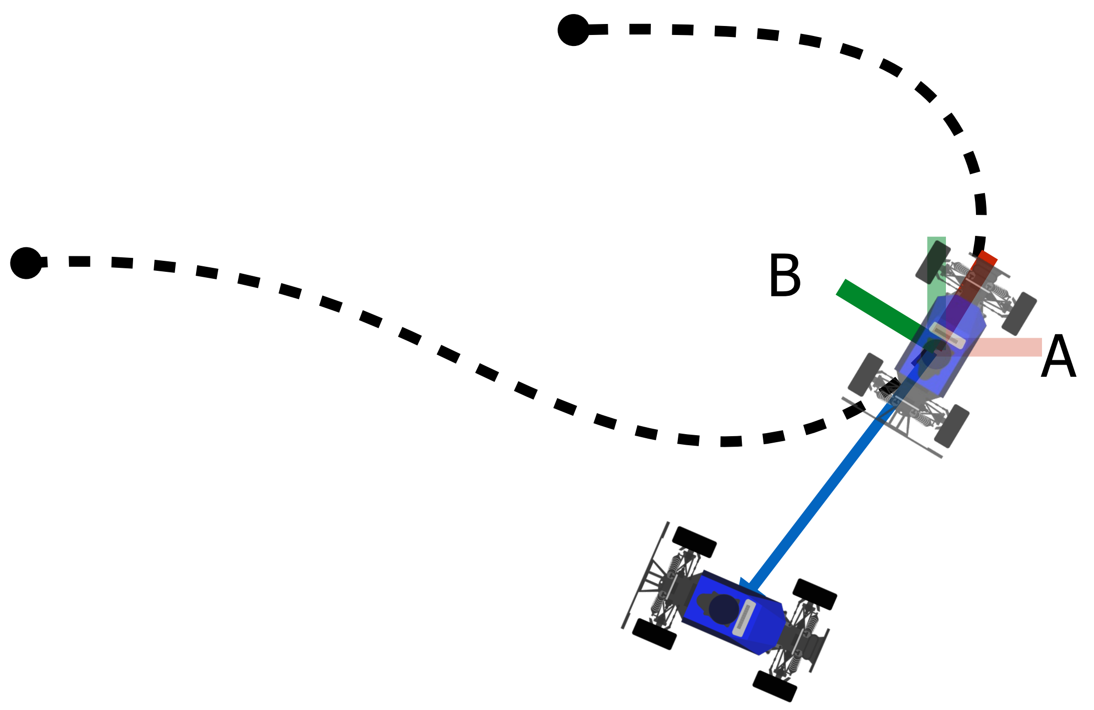
Step 3: Compute error to reference state



We want position in frame B

$$^{B}e = \begin{bmatrix} e_{at} \\ e_{ct} \end{bmatrix} = \begin{bmatrix} \cos(\theta_{ref}) & \sin(\theta_{ref}) \\ -\sin(\theta_{ref}) & \cos(\theta_{ref}) \end{bmatrix} \left(\begin{bmatrix} x \\ y \end{bmatrix} - \begin{bmatrix} x_{ref} \\ y_{ref} \end{bmatrix} \right)$$

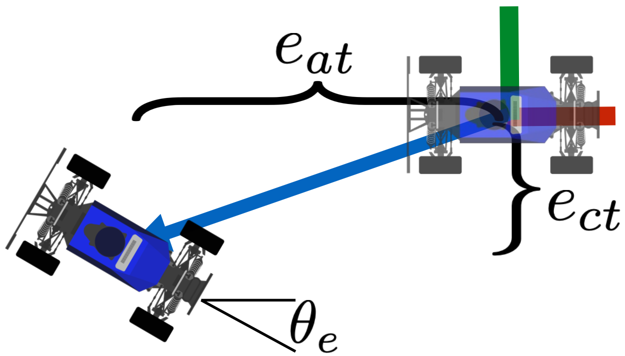
Step 3: Compute error to reference state



Heading error

$$\theta_e = \theta - \theta_{ref}$$

Step 3: Compute error to reference state



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

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

(Heading) $$\theta_e = \theta - \theta_{ref}$$

Step 4: Compute control law

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

$$u = K(e)$$

Different Control Laws

Proportional-integral-derivative (PID) control

Pure-pursuit control

Model-predictive control (MPC)

Linear-quadratic regulator (LQR)

And many many more!

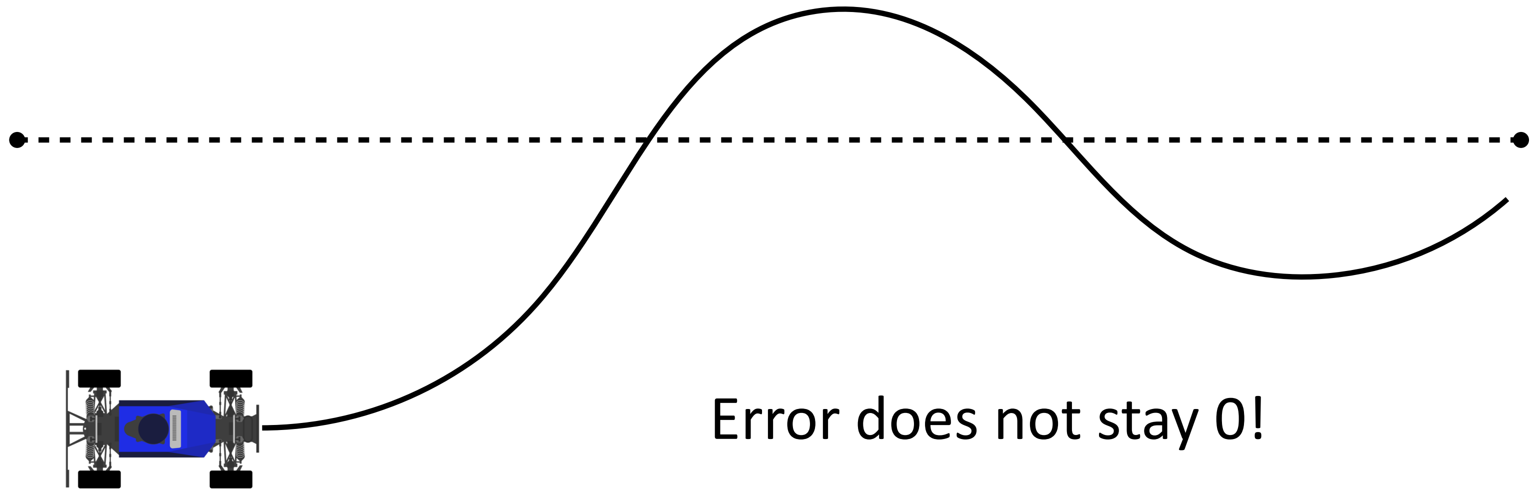
Bang-bang control

Simple control law - choose between hard left and hard right

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

Bang-bang control

What happens when we run this control?



Error does not stay 0!

Need to adapt the magnitude of control proportional to the error …

This clearly sucks! Come back on Monday to find out more