



機器導航與探索

Robotic Navigation and Exploration

Unit 3: Kinematic Model & Path Tracking Control

Min-Chun Hu anitahu@cs.nthu.edu.tw

CS, NTHU

直播連結 <https://www.youtube.com/@NTHURNE-I9v>

Outline

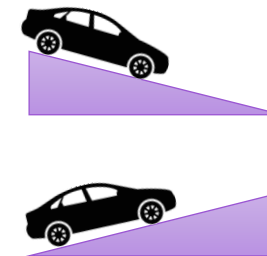
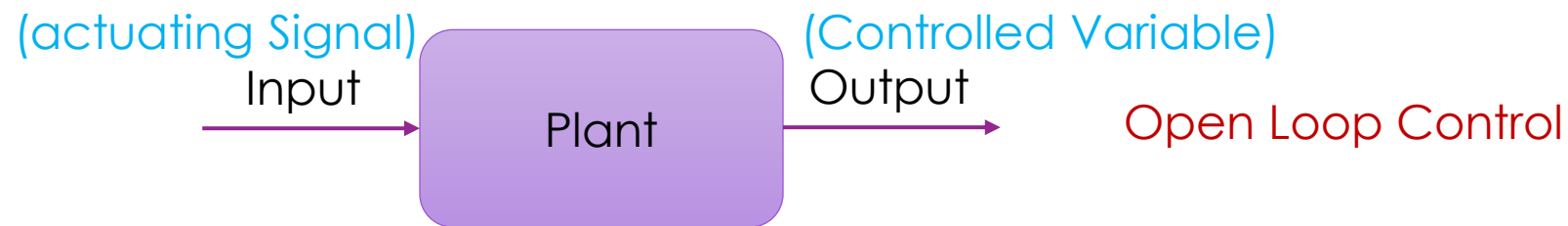
- Basics of Control System for Automobile
- PID (Proportional-Integral-Derivative) Control
- Kinematic Model
- Differential Drive
- Pure-Pursuit Control
- Bicycle Model
 - Pure Pursuit Control
 - Stanley Control (Path Coordinate and Control Stabilization)
 - Linear Quadratic Regulator (LQR)

Outline

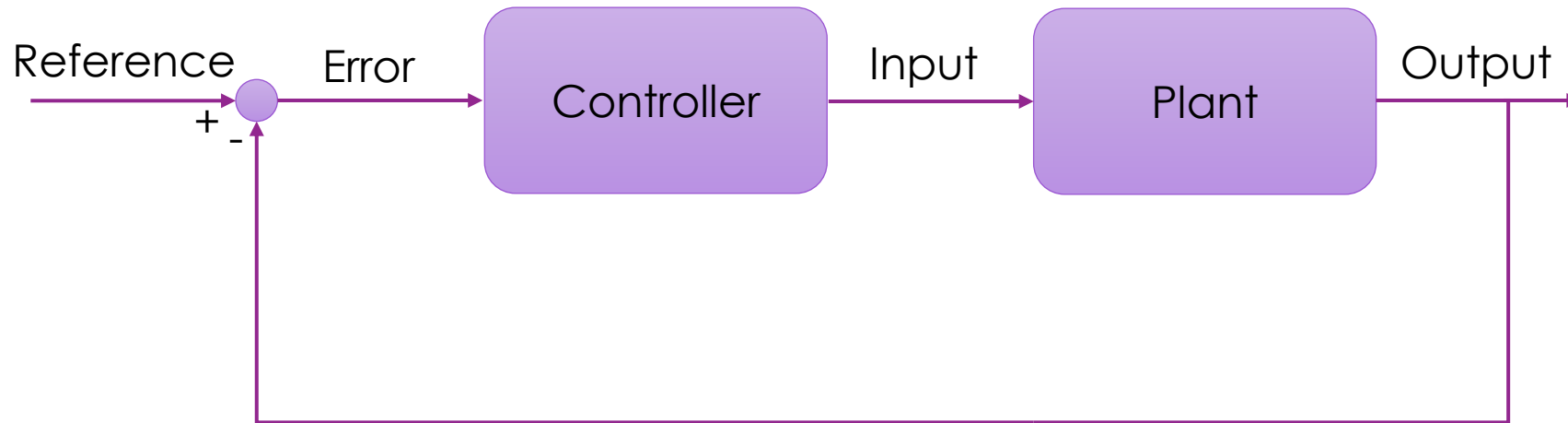
- Basics of Control System for Automobile
- PID (Proportional-Integral-Derivative) Control
- Kinematic Model
- Differential Drive
- Pure-Pursuit Control
- Bicycle Model
 - Pure Pursuit Control
 - Stanley Control (Path Coordinate and Control Stabilization)
 - Linear Quadratic Regulator (LQR)

Control Theory: Open Loop Control

- Control System: the mechanism that affects the future state of a system
- Control Theory: a strategy to change input to desired output

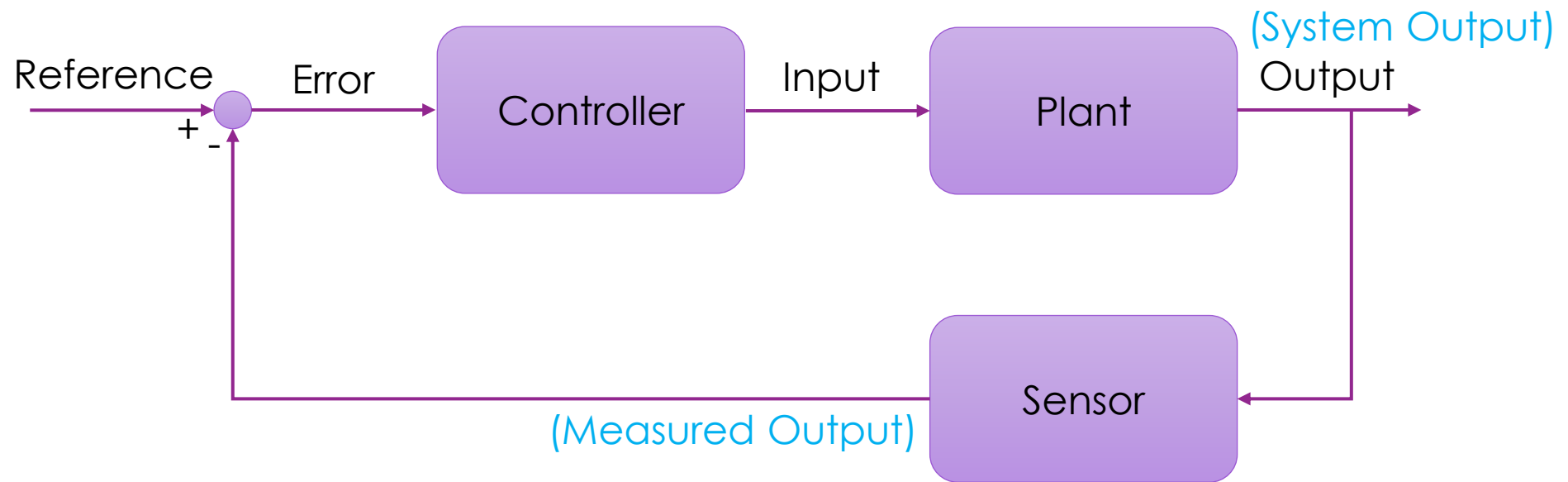


Control Theory: Close Loop Control



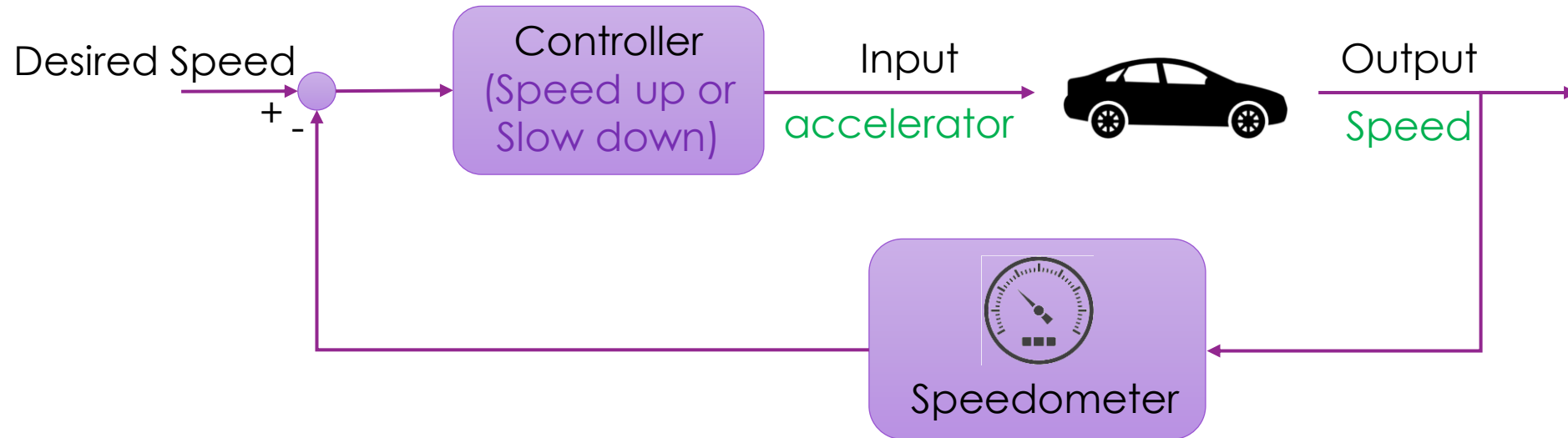
Close Loop Control
(Feedback Control)

Control Theory: Close Loop Control



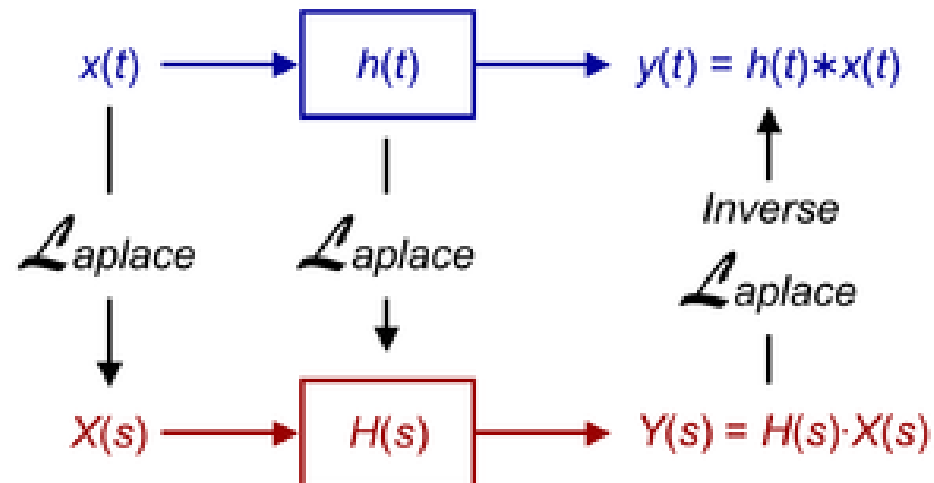
Close Loop Control
(Feedback Control)

Control Theory : Car Example



Linear Time Invariant System

Time domain



Frequency domain

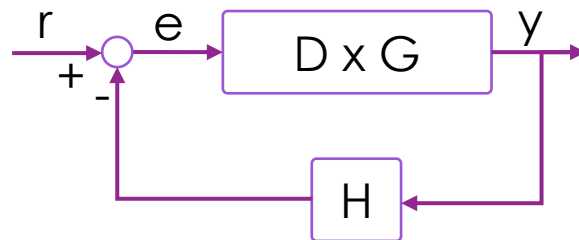
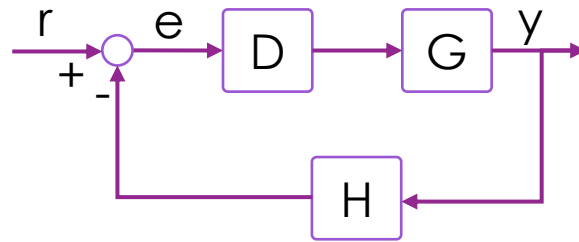
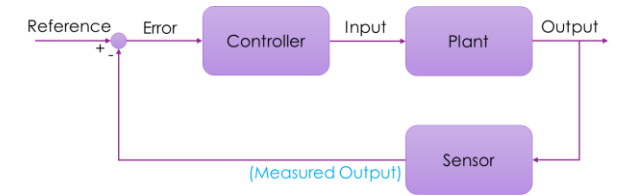
Laplace transform

$$\begin{aligned}\mathcal{L}\{f(t)\} &= \int_{0^-}^{\infty} e^{-st} f(t) dt \\ &= \left[\frac{f(t)e^{-st}}{-s} \right]_{0^-}^{\infty} - \int_{0^-}^{\infty} \frac{e^{-st}}{-s} f'(t) dt \quad (\text{by parts}) \\ &= \left[-\frac{f(0^+)}{s} \right] + \frac{1}{s} \mathcal{L}\{f'(t)\},\end{aligned}$$

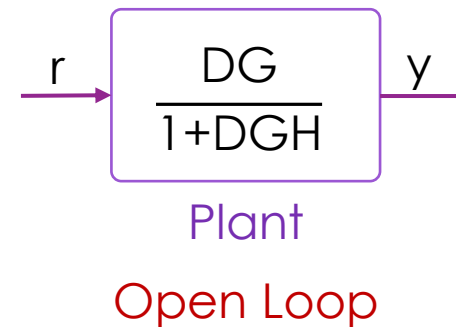
Basic Laplace Transform Pairs

Signal or Function	$f(t)$	$F(s)$
Impulse	$\delta(t)$	1
Step	$u(t) = 1, \quad t \geq 0$	$\frac{1}{s}$
Ramp	$r(t) = t, \quad t \geq 0$	$\frac{1}{s^2}$
Exponential	$e^{-\alpha t} \quad e^{-\alpha t} u(t)$	$\frac{1}{s + \alpha}$
Damped Ramp	$te^{-\alpha t}$	$\frac{1}{(s + \alpha)^2}$
Sine	$\sin(\beta t)$	$\frac{\beta}{s^2 + \beta^2}$
Cosine	$\cos(\beta t)$	$\frac{s}{s^2 + \beta^2}$
Damped Sine	$e^{-\alpha t} \sin(\beta t)$	$\frac{\beta}{(s + \alpha)^2 + \beta^2}$
Damped Cosine	$e^{-\alpha t} \cos(\beta t)$	$\frac{s + \alpha}{(s + \alpha)^2 + \beta^2}$
Simple Complex Pole	see next pg	see next pg

Linear Time Invariant System



Equivalent



$$e = r - yH$$

$$y = e \cdot D \cdot G \quad e = \frac{y}{DG}$$

$$r - yH = \frac{y}{DG}$$

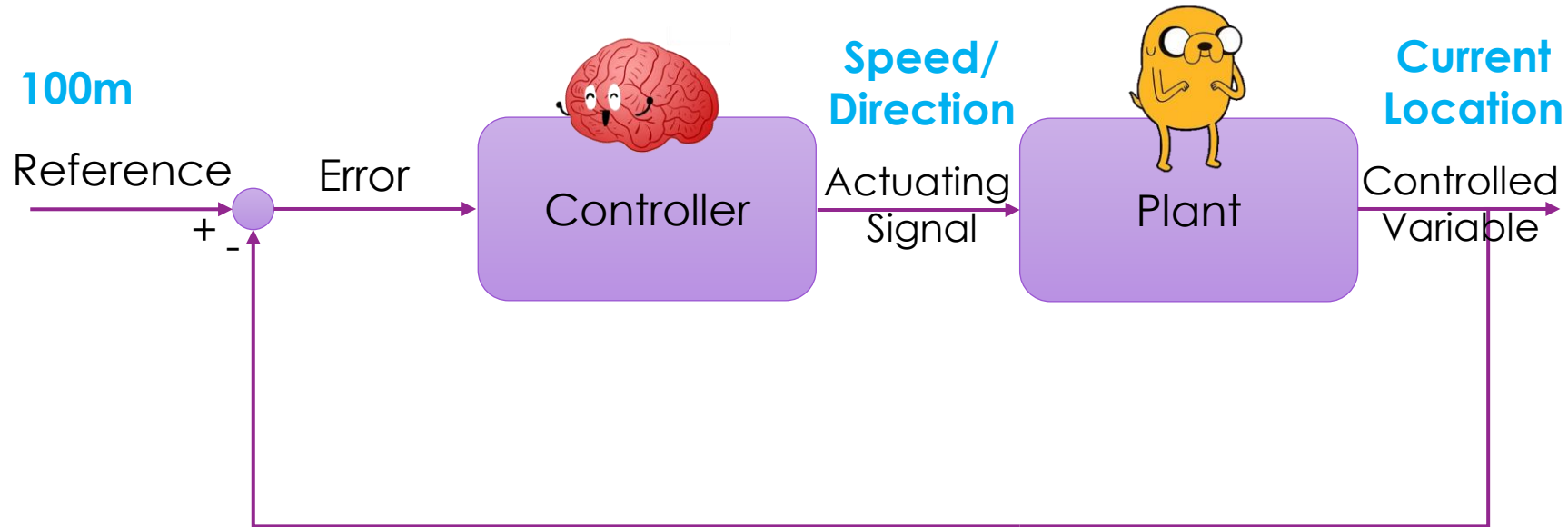
$$(DG)(r - yH) = y$$

$$DGr = y(1 + DGH) \quad \text{or} \quad y = \frac{DGr}{1 + DGH}$$

Outline

- Basics of Control System for Automobile
- PID (Proportional-Integral-Derivative) Control
- Kinematic Model
- Differential Drive
- Pure-Pursuit Control
- Bicycle Model
 - Pure Pursuit Control
 - Stanley Control (Path Coordinate and Control Stabilization)
 - Linear Quadratic Regulator (LQR)

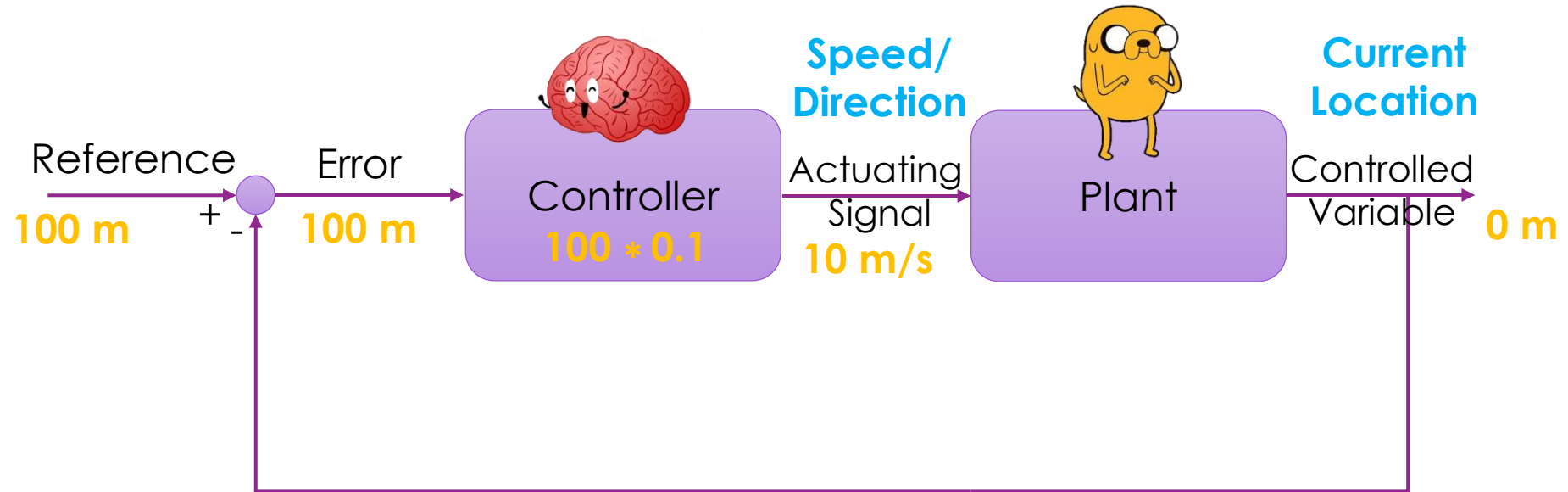
PID Control : Proportional Gain



0 m

100 m (desired location)

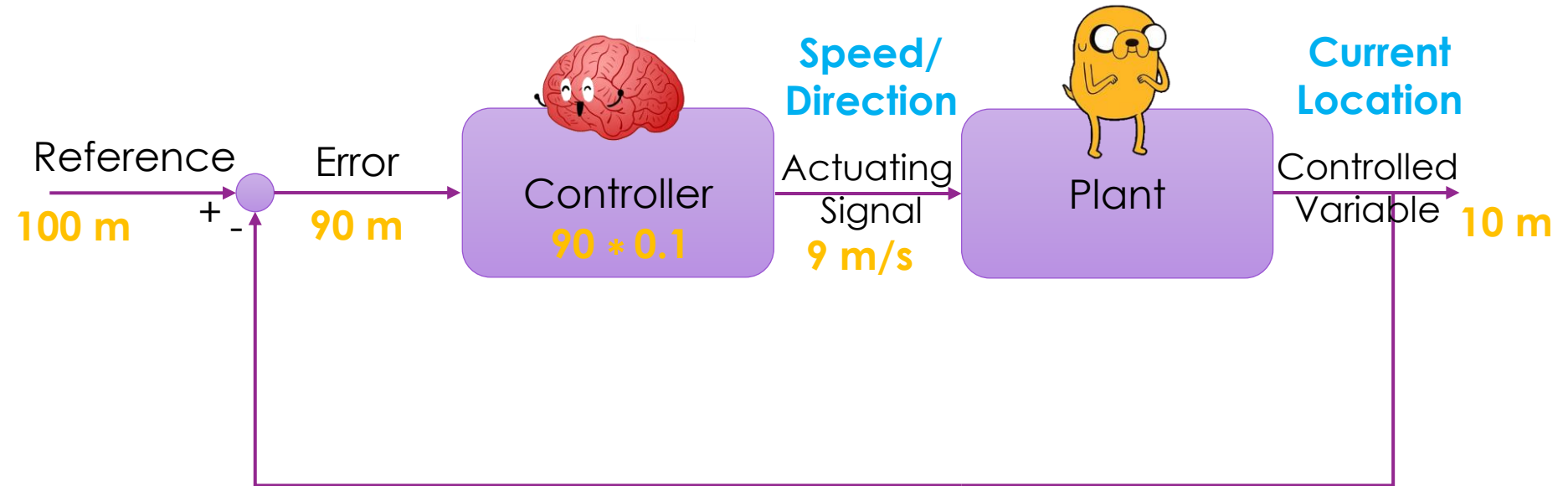
PID Control : Proportional Gain



0 m

100 m (desired location)

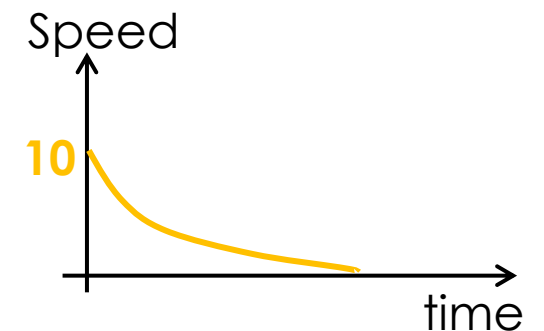
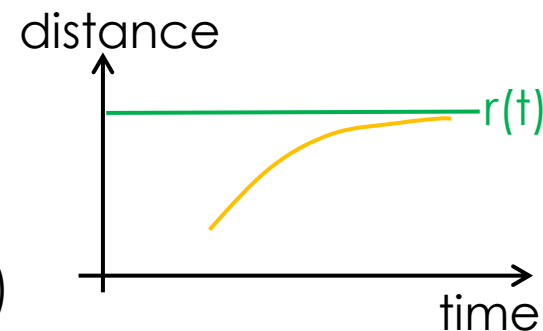
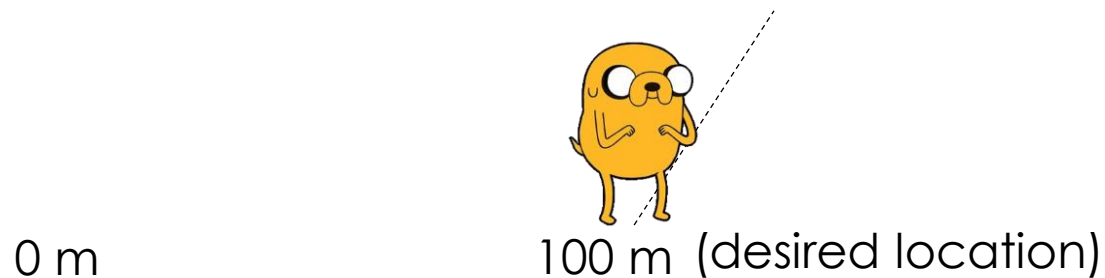
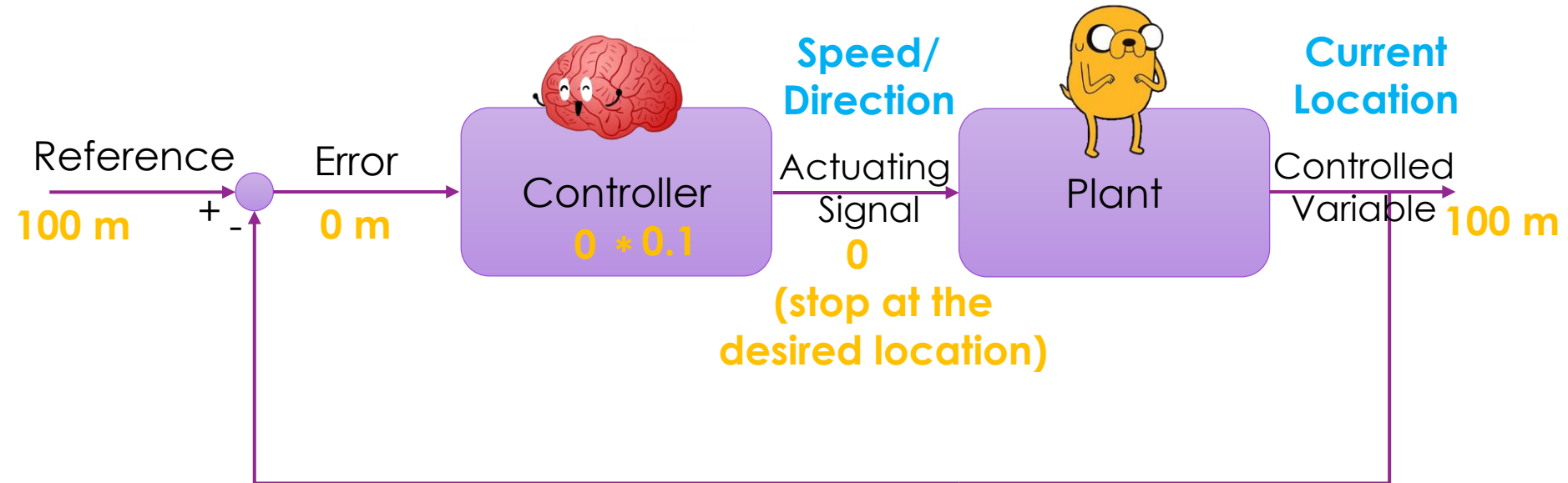
PID Control : Proportional Gain



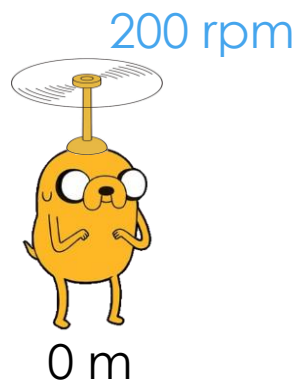
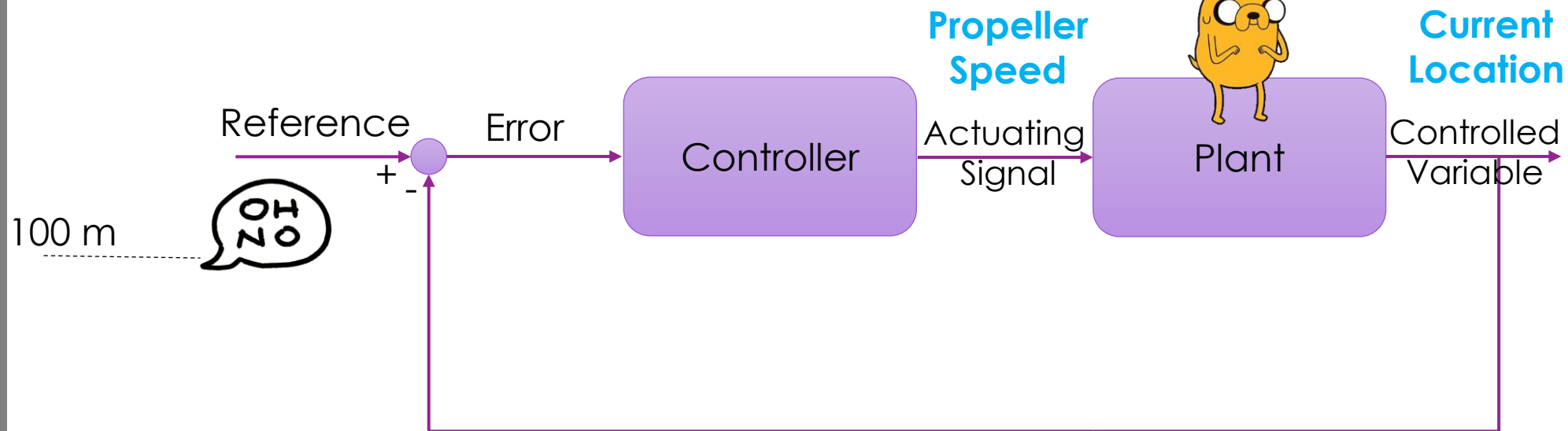
0 m

100 m (desired location)

PID Control : Proportional Gain



PID Control : Problem of Proportional Gain



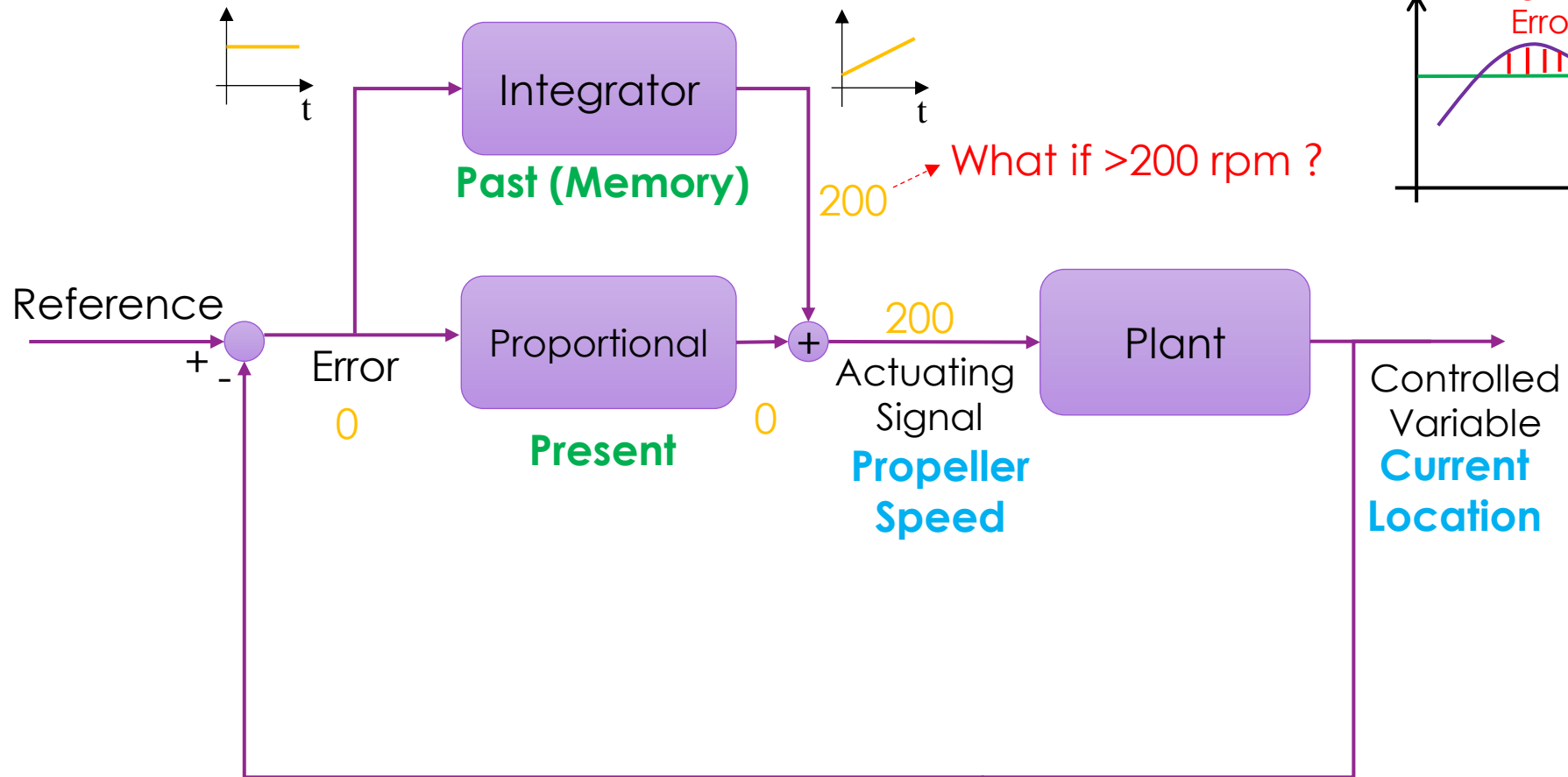
Steady state error

Error	*	Gain	=	Propeller Speed
100	*	2	=	200 rpm
40	*	5	=	200 rpm
20	*	10	=	200 rpm
2	*	100	=	200 rpm

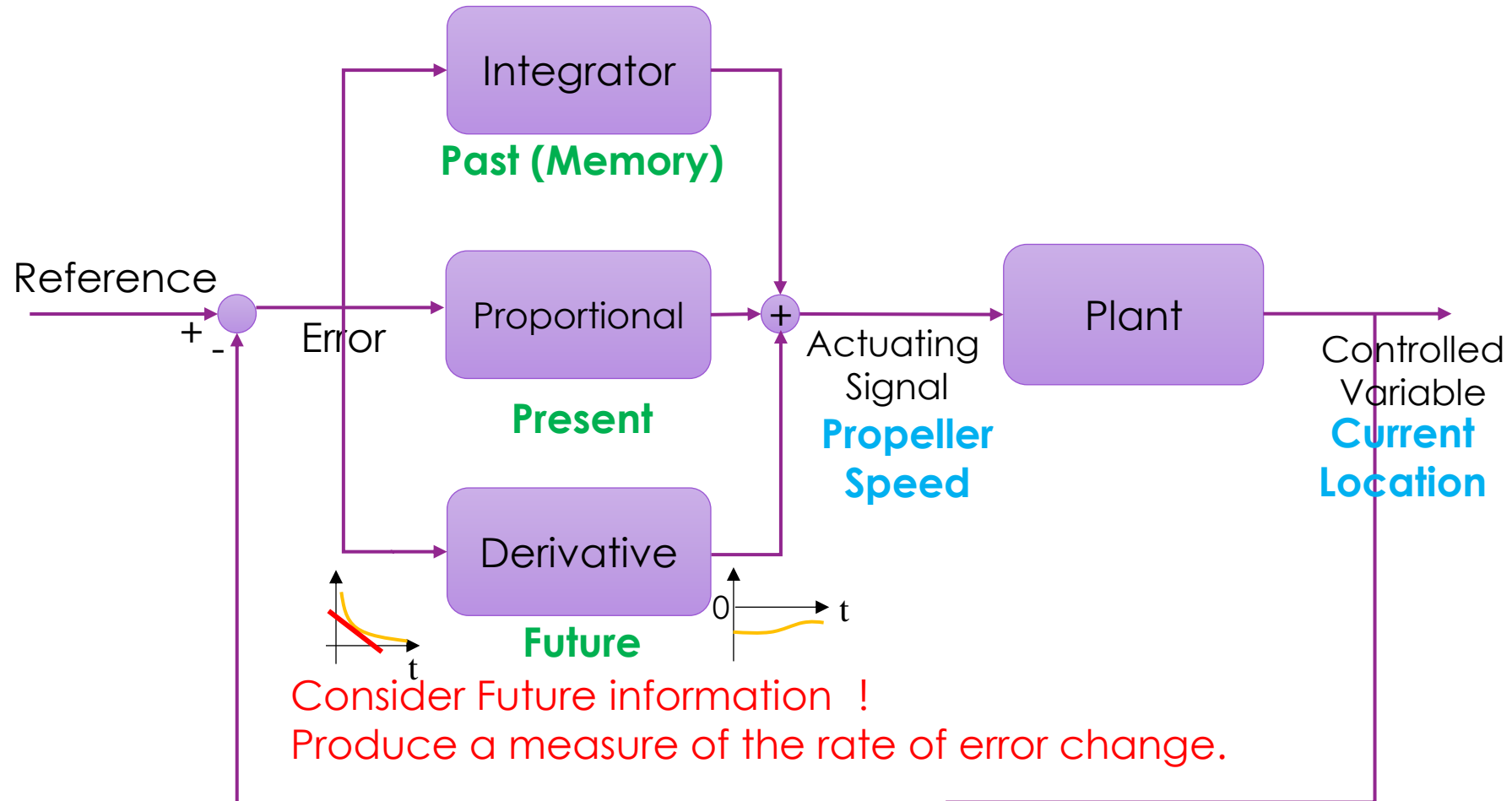
Idea: Consider past information !

PID Control : Integral Gain

Consider past information !
Sum up non-zero steady state error over time



PID Control : Differential Gain



PID Control

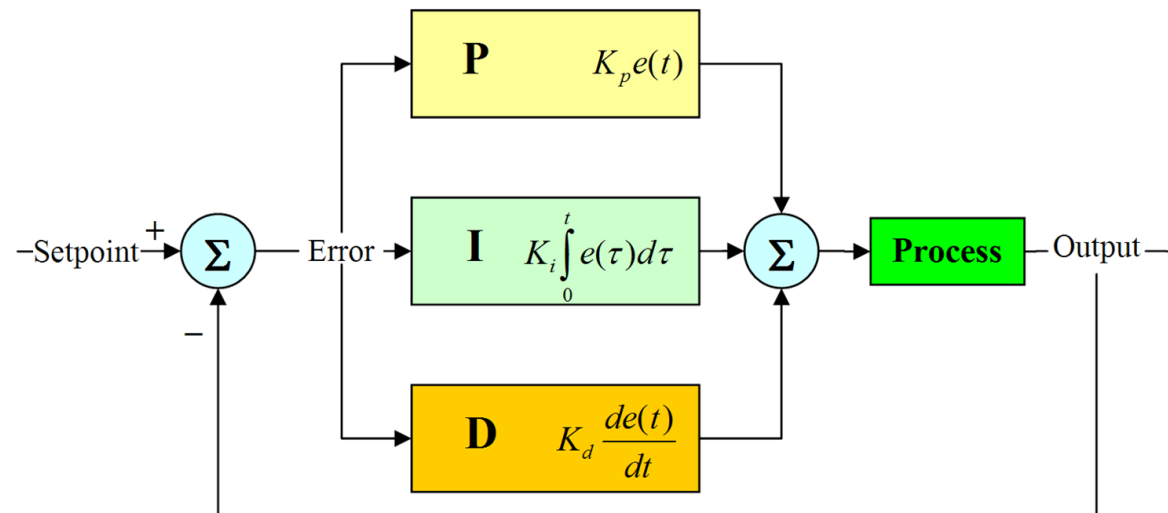
- **P**roportional / **I**ntegral / **D**erivative Control

Continuous Form :

$$Output = K_p e(t) + K_i \int_0^t e_t dt + K_d \frac{de(t)}{dt}$$

Discrete Form :

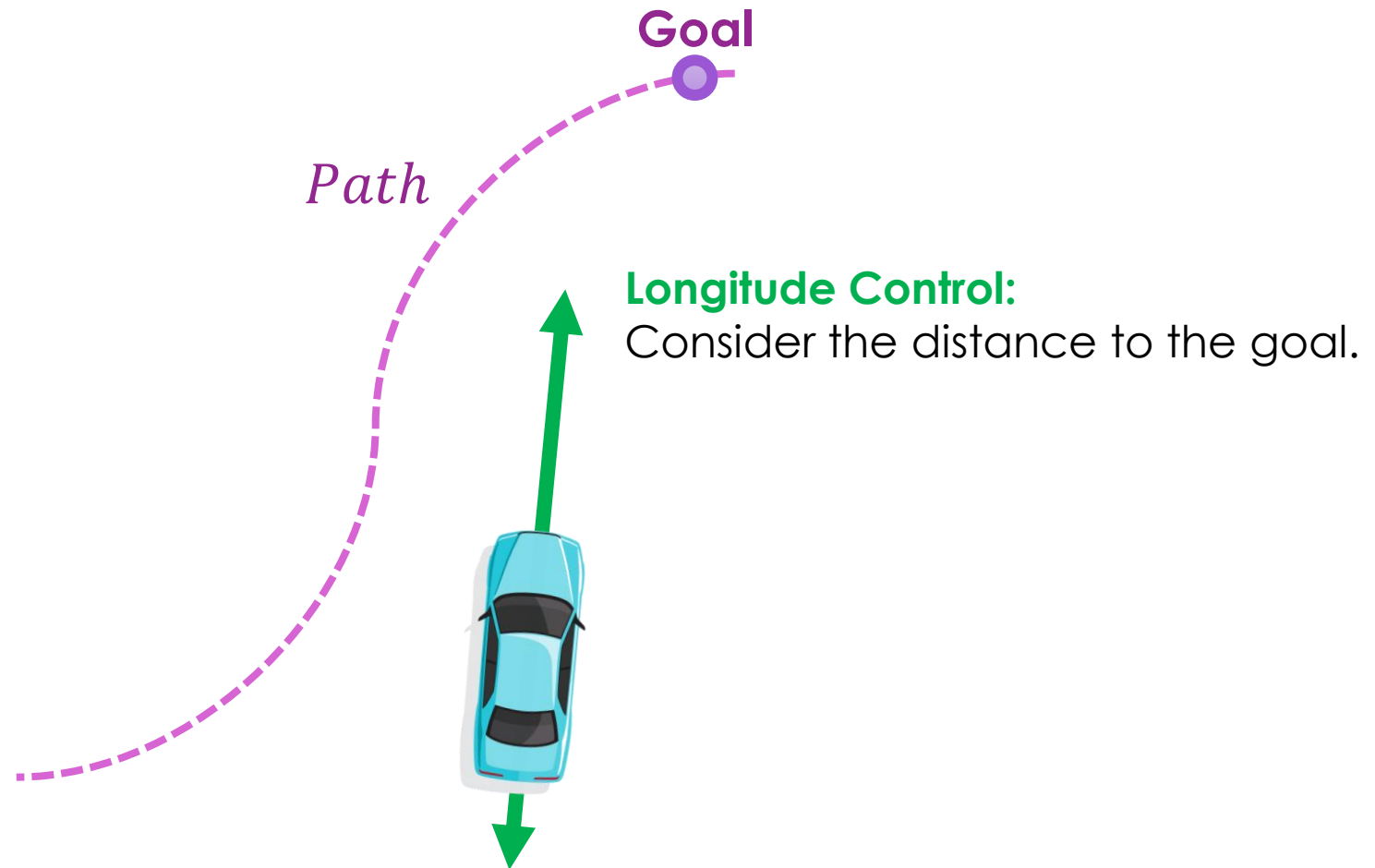
$$Output = K_p e(t) + K_i \sum_0^t e_t + K_d (e(t) - e(t-1))$$



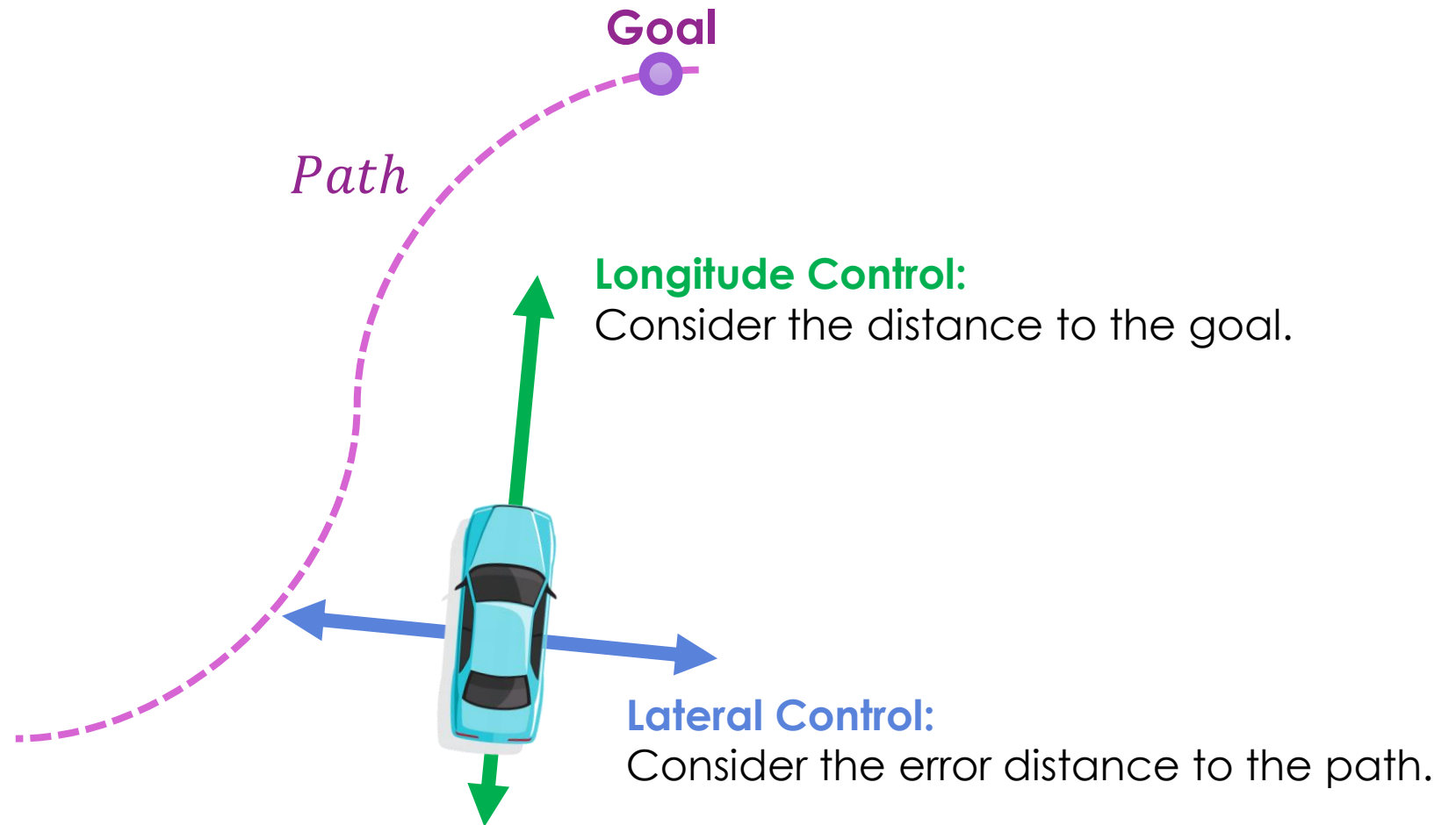
Outline

- Basics of Control System for Automobile
- PID (Proportional-Integral-Derivative) Control
- Kinematic Model
- Differential Drive
- Pure-Pursuit Control
- Bicycle Model
 - Pure Pursuit Control
 - Stanley Control (Path Coordinate and Control Stabilization)
 - Linear Quadratic Regulator (LQR)

Path Tracking Problem



Path Tracking Problem



Basic Kinematic Model

State:

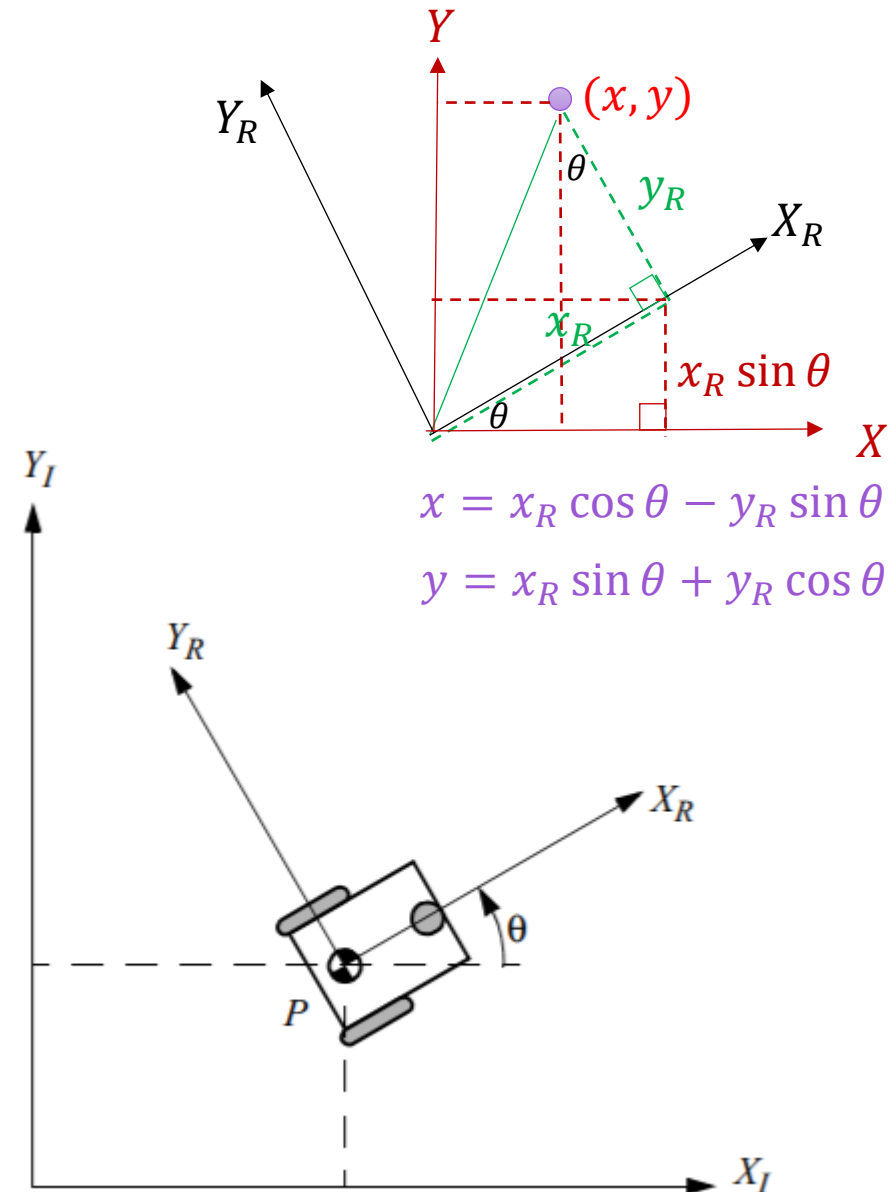
$$\xi_I = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix}$$

Rotation Matrix:

$$R(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Kinematic Model:

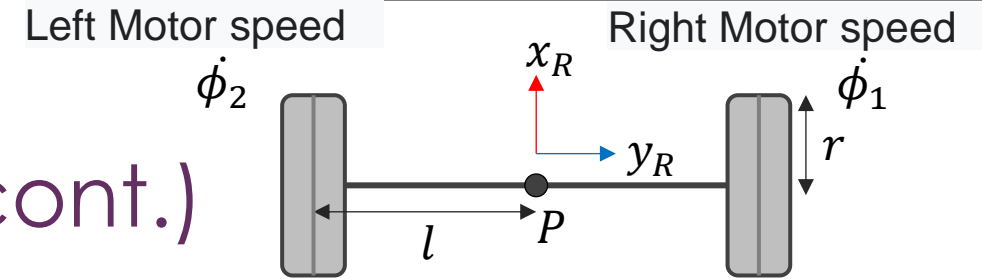
$$\begin{aligned} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} &= R(\theta) \begin{bmatrix} \dot{x}_R \\ \dot{y}_R \\ \dot{\theta} \end{bmatrix} \\ &= \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ 0 \\ \omega \end{bmatrix} \\ &= \begin{bmatrix} v \cos(\theta) \\ v \sin(\theta) \\ \omega \end{bmatrix} \end{aligned}$$



Outline

- Basics of Control System for Automobile
- PID (Proportional-Integral-Derivative) Control
- Kinematic Model
- Differential Drive
- Pure-Pursuit Control
- Bicycle Model
 - Pure Pursuit Control
 - Stanley Control (Path Coordinate and Control Stabilization)
 - Linear Quadratic Regulator (LQR)

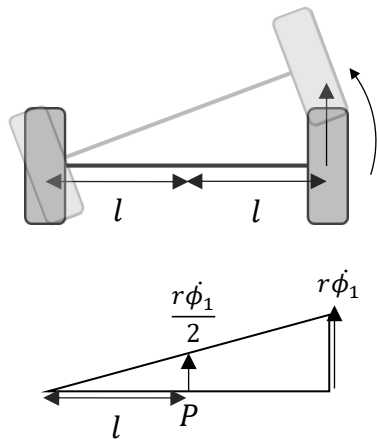
Differential Drive Vehicle (cont.)



Right Wheel:

$$\dot{x}_{R1} = \frac{r\dot{\phi}_1}{2}$$

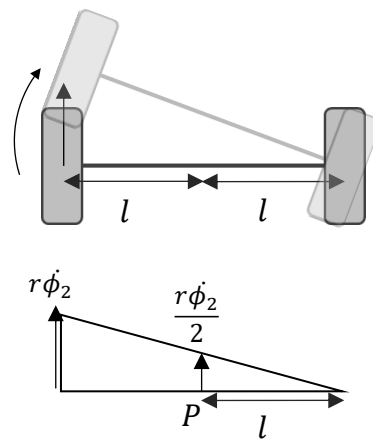
$$\omega_1 = \frac{r\dot{\phi}_1}{2l}$$



Left Wheel:

$$\dot{x}_{R2} = \frac{r\dot{\phi}_2}{2}$$

$$\omega_2 = \frac{-r\dot{\phi}_2}{2l}$$



Kinematic model for differential drive:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = R(\theta)^{-1} \begin{bmatrix} \dot{x}_R \\ \dot{y}_R \\ \dot{\theta} \end{bmatrix}$$

$$= \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{x}_R \\ \dot{y}_R \\ \dot{\theta} \end{bmatrix}$$

$$= \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \frac{r\dot{\phi}_1}{2} + \frac{r\dot{\phi}_2}{2} \\ 0 \\ \frac{r\dot{\phi}_1}{2l} - \frac{r\dot{\phi}_2}{2l} \end{bmatrix}$$

Differential Drive Vehicle

- Given target velocity v and angular velocity ω

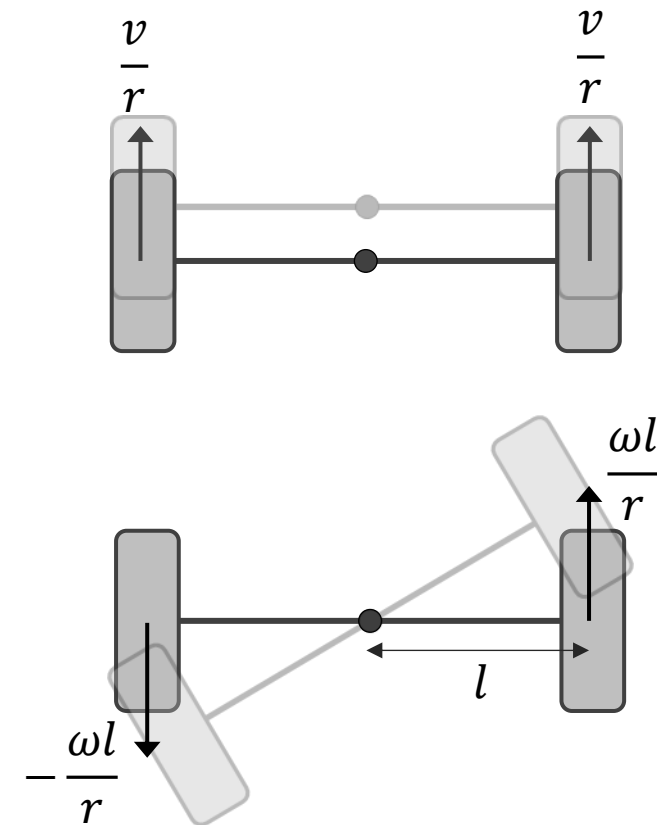
$$\begin{cases} v = \frac{r\dot{\phi}_1}{2} + \frac{r\dot{\phi}_2}{2} \\ \omega = \frac{r\dot{\phi}_1}{2l} - \frac{r\dot{\phi}_2}{2l} \end{cases}$$

$$\dot{\phi}_2 = \left(v - \frac{r\dot{\phi}_1}{2} \right) \frac{2}{r} = \frac{2v}{r} - \dot{\phi}_1$$

$$\omega = \frac{r\dot{\phi}_1}{2l} - \frac{r\left(\frac{2v}{r} - \dot{\phi}_1\right)}{2l} = \frac{r\dot{\phi}_1 - v}{l}$$

$$\dot{\phi}_1 = \frac{v}{r} + \frac{\omega l}{r}$$

$$\dot{\phi}_2 = \frac{v}{r} - \frac{\omega l}{r}$$



Outline

- Basics of Control System for Automobile
- PID (Proportional-Integral-Derivative) Control
- Kinematic Model
- Differential Drive
- Pure-Pursuit Control
- Bicycle Model
 - Pure Pursuit Control
 - Stanley Control (Path Coordinate and Control Stabilization)
 - Linear Quadratic Regulator (LQR)

Pure Pursuit Control

- Concept:
 - Modify the angular velocity to let the center achieve a point on path

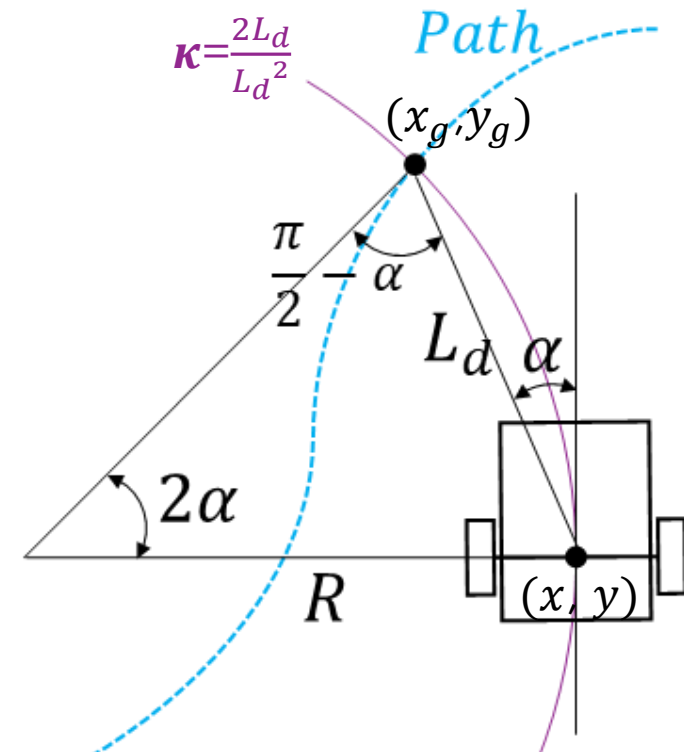
$$\alpha = \arctan\left(\frac{y - y_g}{x - x_g}\right) - \theta$$

$$\frac{L_d}{\sin(2\alpha)} = \frac{R}{\sin\left(\frac{\pi}{2} - \alpha\right)}$$

$$R = \frac{L_d \sin\left(\frac{\pi}{2} - \alpha\right)}{\sin(2\alpha)} = \frac{L_d \cos(\alpha)}{2 \sin(\alpha) \cos(\alpha)} = \frac{L_d}{2 \sin(\alpha)}$$

$$\omega = \frac{v}{R} = \frac{2v \sin(\alpha)}{L_d}$$

L_d usually set to $(kv + L_{fc})$, where k, L_{fc} are parameters.



Outline

- Basics of Control System for Automobile
- PID (Proportional-Integral-Derivative) Control
- Kinematic Model
- Differential Drive
- Pure-Pursuit Control
- Bicycle Model
 - Pure Pursuit Control
 - Stanley Control (Path Coordinate and Control Stabilization)
 - Linear Quadratic Regulator (LQR)

Kinematic Bicycle Model

- Speed and Steering Control

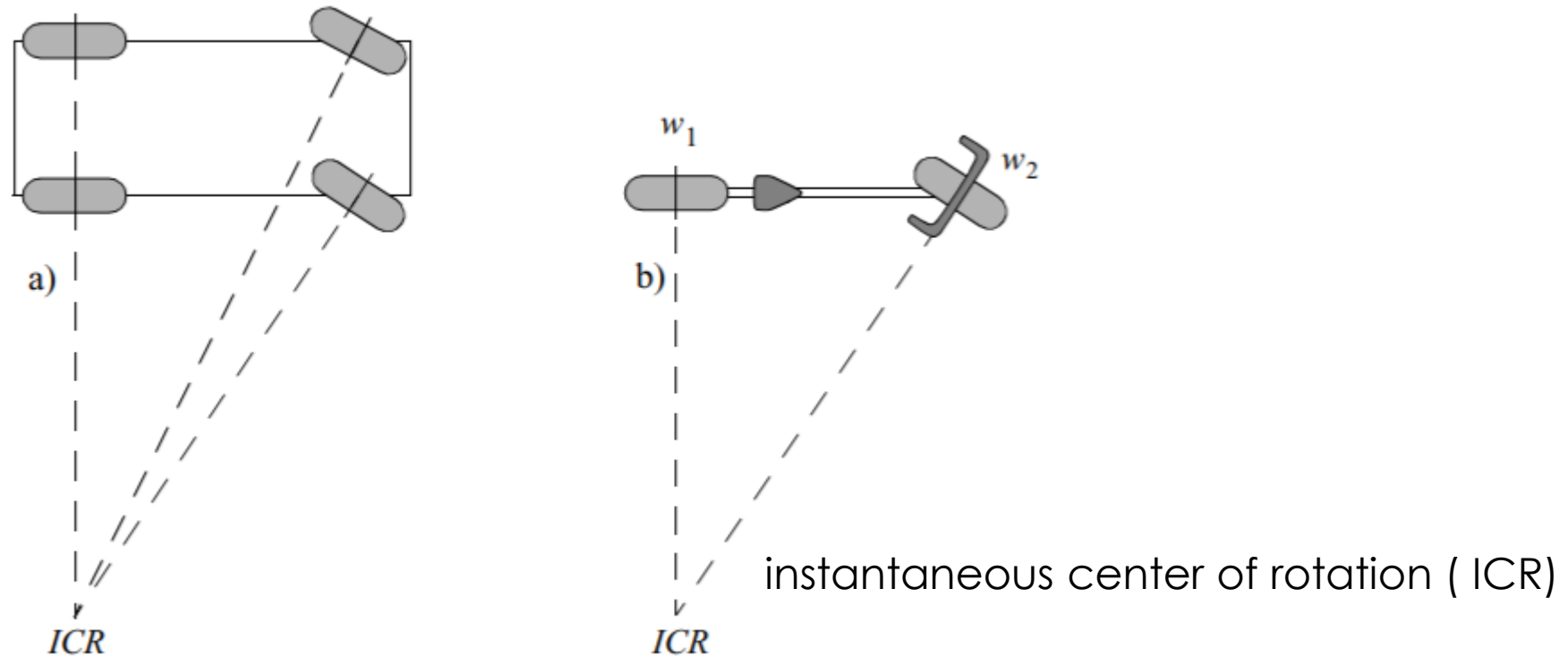


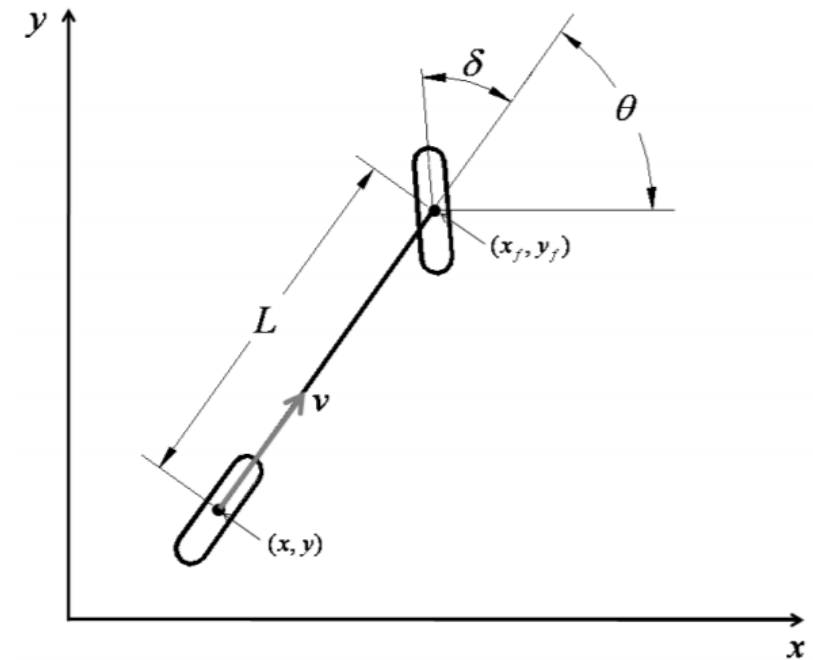
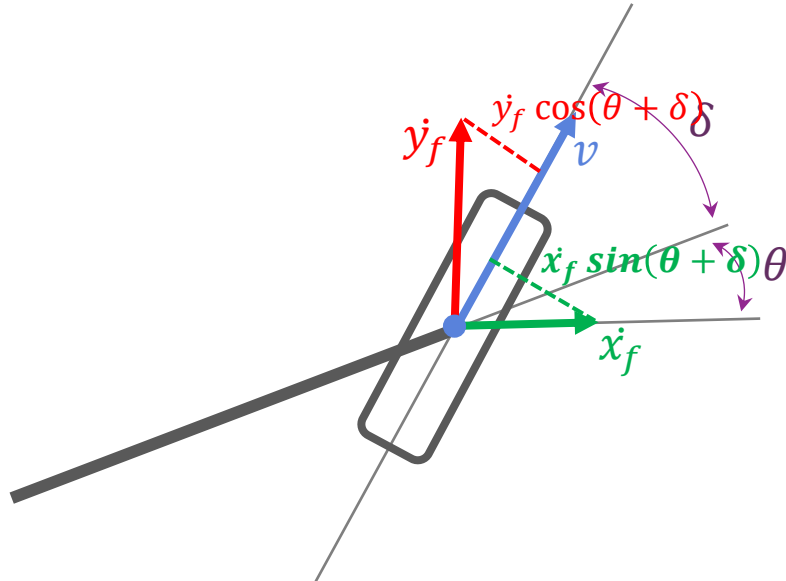
Figure 3.12
(a) Four-wheel with car-like Ackerman steering. (b) bicycle.

Kinematic Bicycle Model

- nonholonomic constraint equations

$$\dot{x}_f \sin(\theta + \delta) - \dot{y}_f \cos(\theta + \delta) = 0 \quad (1) \text{ Front Wheel}$$

$$\dot{x} \sin(\theta) - \dot{y} \cos(\theta) = 0 \quad (2) \text{ Rear Wheel}$$



Kinematic Bicycle Model

- nonholonomic constraint equations

$$\dot{x}_f \sin(\theta + \delta) - \dot{y}_f \cos(\theta + \delta) = 0 \quad (1) \text{ Front Wheel}$$

$$\dot{x} \sin(\theta) - \dot{y} \cos(\theta) = 0 \quad (2) \text{ Rear Wheel}$$

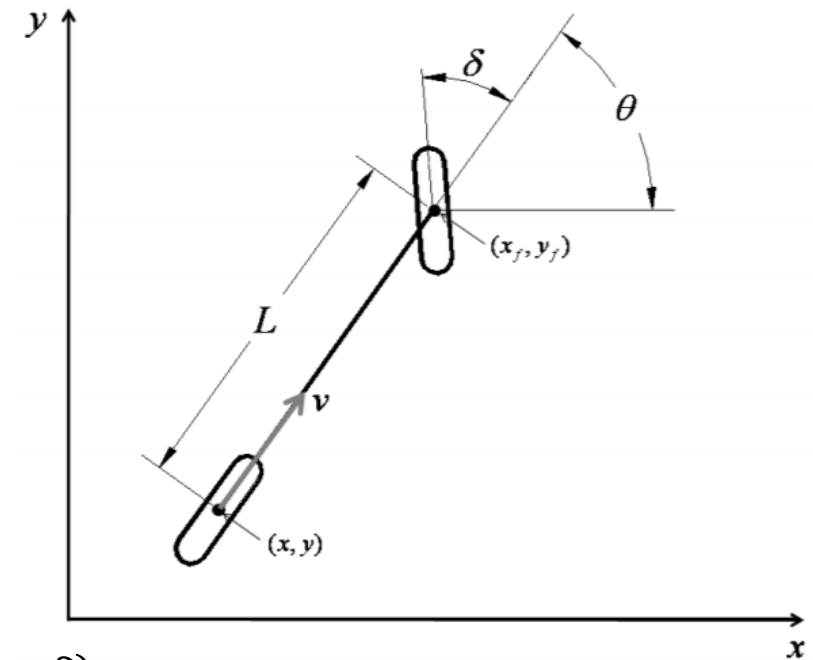
- Front Wheel Position

$$x_f = x + L \cos(\theta)$$

$$y_f = y + L \sin(\theta)$$

- Eliminating front wheel position from (1)

$$\begin{aligned} 0 &= (\dot{x} - \dot{\theta} L \sin(\theta)) \sin(\theta + \delta) - (\dot{y} + \dot{\theta} L \cos(\theta)) \cos(\theta + \delta) \\ &= \dot{x} \sin(\theta + \delta) - \dot{y} \cos(\theta + \delta) - \dot{\theta} L \sin(\theta) (\sin(\theta) \cos(\delta) + \cos(\theta) \sin(\delta)) \\ &\quad - \dot{\theta} L \cos(\theta) (\cos(\theta) \cos(\delta) - \sin(\theta) \sin(\delta)) \\ &= \dot{x} \sin(\theta + \delta) - \dot{y} \cos(\theta + \delta) - \dot{\theta} L \cos(\delta) \quad (3) \end{aligned}$$



Kinematic Bicycle Model

- nonholonomic constraint equations

$$\dot{x} \sin(\theta + \delta) - \dot{y} \cos(\theta + \delta) - \dot{\theta} L \cos(\delta) = 0 \quad (3)$$

$$\dot{x} \sin(\theta) - \dot{y} \cos(\theta) = 0 \quad (2)$$

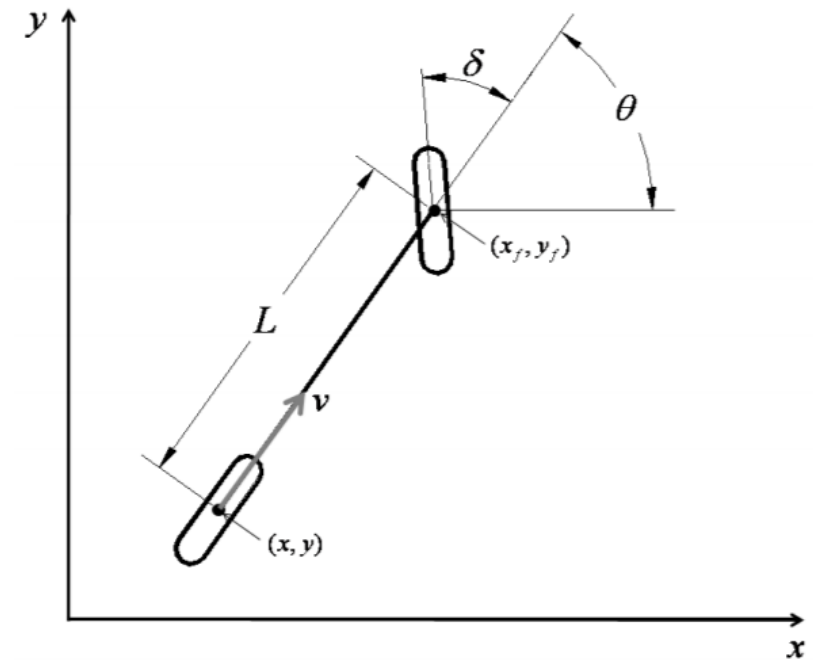
- Rear wheel satisfied the constrain (2) when

$$\dot{x} = v \cos(\theta) \quad (4)$$

$$\dot{y} = v \sin(\theta) \quad (5)$$

- Applying (4)(5) to (3)

$$\begin{aligned} \dot{\theta} &= \frac{\dot{x} \sin(\theta + \delta) - \dot{y} \cos(\theta + \delta)}{L \cos(\delta)} \\ &= \frac{v \cos(\theta) (\sin(\theta) \cos(\delta) + \cos(\theta) \sin(\delta)) - v \sin(\theta) (\cos(\theta) \cos(\delta) - \sin(\theta) \sin(\delta))}{L \cos(\delta)} \\ &= \frac{v (\cos^2(\theta) + \sin^2(\theta)) \sin(\delta)}{L \cos(\delta)} = \frac{v \tan(\delta)}{L} \end{aligned}$$



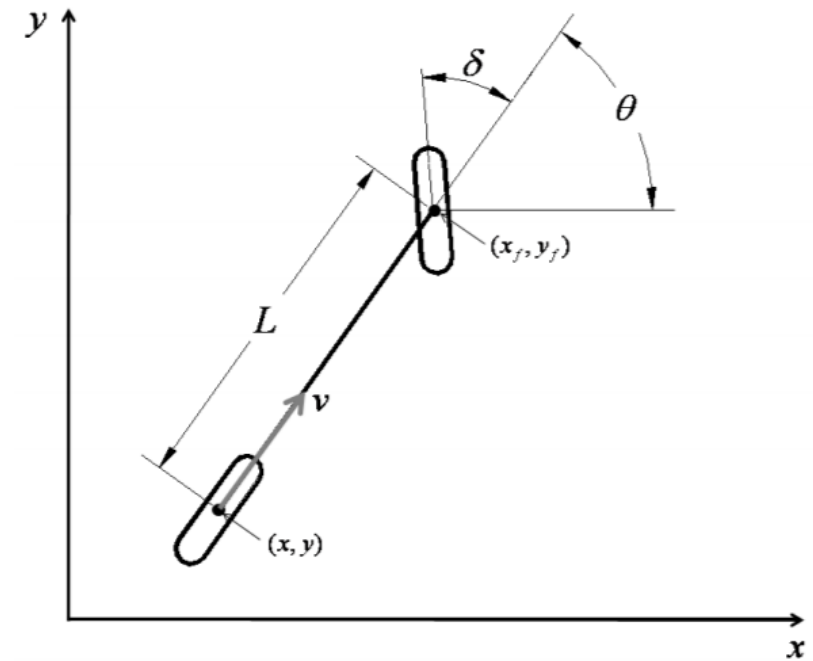
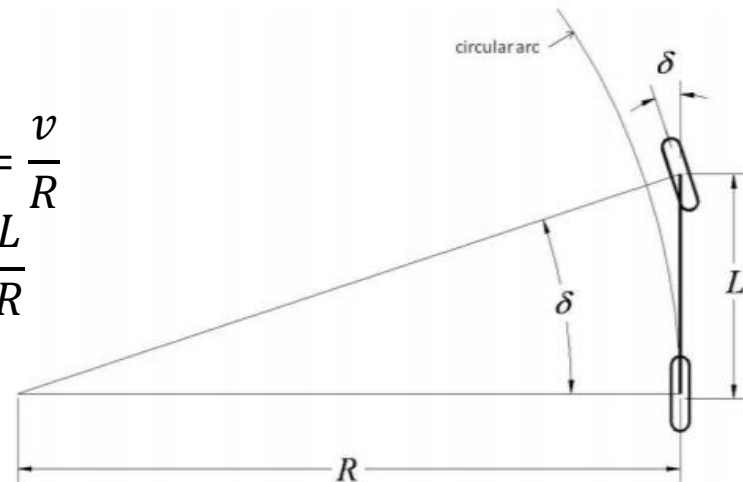
Kinematic Bicycle Model

- Kinematic Model

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

- Some Property

$$\begin{aligned} R\dot{\theta} &= v \\ \frac{v \tan(\delta)}{L} &= \frac{v}{R} \\ \tan(\delta) &= \frac{L}{R} \end{aligned}$$



Pure Pursuit Control for Bicycle Model

- Concept:
 - Control the steer to let the rear wheel achieve a point on the path.

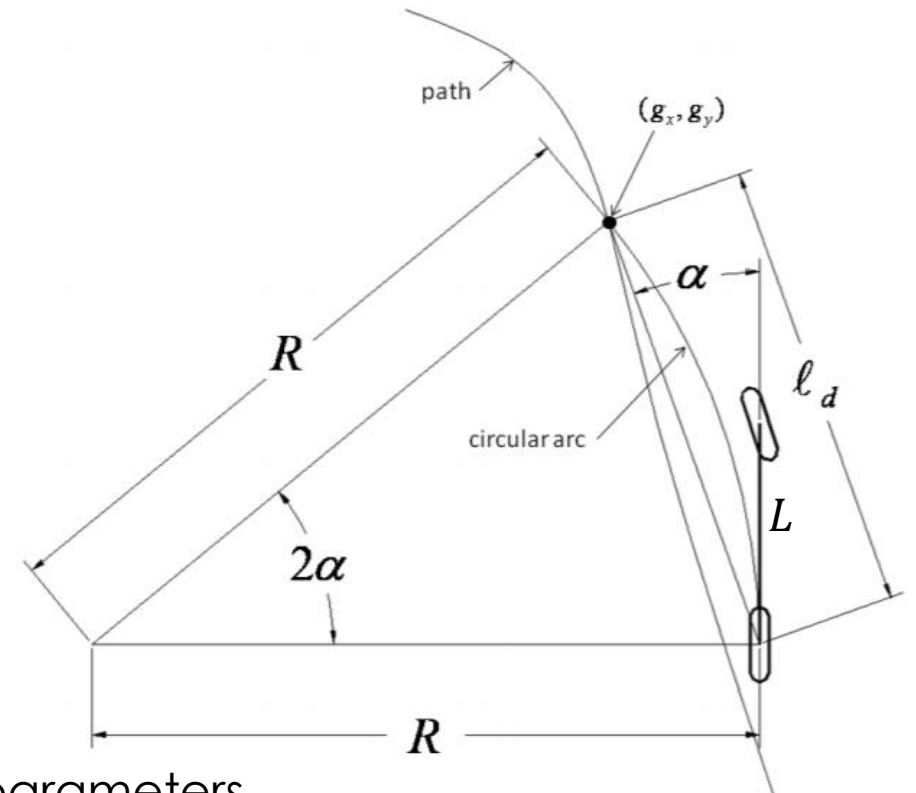
$$\alpha = \arctan\left(\frac{y - y_g}{x - x_g}\right) - \theta$$

$$R = \frac{L_d \sin\left(\frac{\pi}{2} - \alpha\right)}{\sin(2\alpha)} = \frac{L_d \cos(\alpha)}{2 \sin(\alpha) \cos(\alpha)} = \frac{L_d}{2 \sin(\alpha)}$$

$$\tan(\delta) = \frac{L}{R}$$

$$\delta = \arctan\left(\frac{L}{R}\right) = \arctan\left(\frac{2L \sin(\alpha)}{L_d}\right)$$

L_d usually set to $(kv + L_{fc})$, where k, L_{fc} are parameters.



Stanley Control

- Concept:
 - Exponential stability for front wheel feedback

- Differential of error distance

$$\dot{e} = v_f \sin(\delta - \theta_e)$$

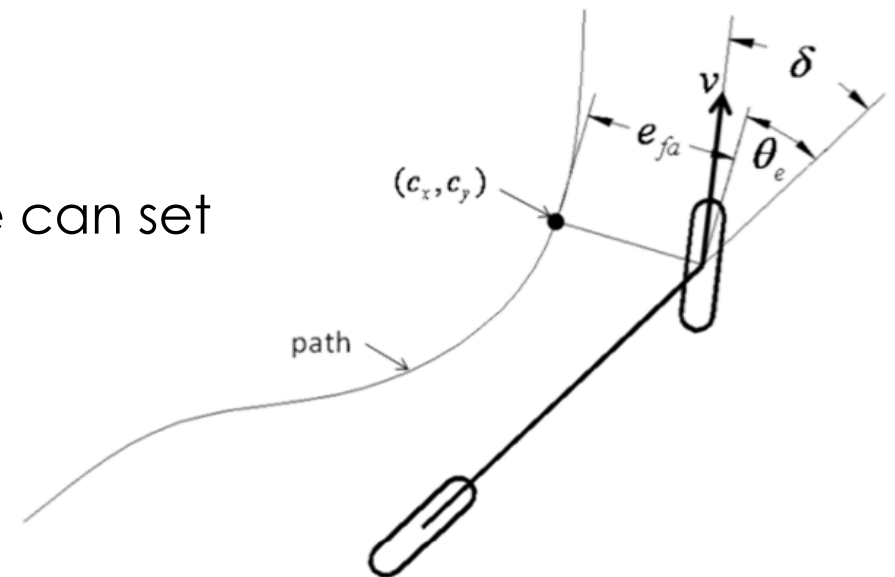
- To achieve exponential stability to path, we can set

$$\dot{e} = -ke, \text{ where } k > 0$$

$$-ke = v_f \sin(\delta - \theta_e)$$

$$\delta = \arcsin\left(-\frac{ke}{v_f}\right) + \theta_e$$

- It is not defined when $|-ke/v_f| > 1$. We can modify the control law to $\delta = \arctan\left(-\frac{ke}{v_f}\right) + \theta_e$, which satisfy the **local exponential stability (LES)**.



LQR Control

- If we use the motion model with more complex form (e.g. dynamic model), it is hard to directly analyze the error function.
- Linear **Q**uadratic **R**egulator (LQR) introduce the concept of **cost function**, and try to solve the optimization problem when the motion model is **linear form** and the cost function is **quadratic form**.
- The formulation of LQR problem:
 - Define state **x** and control **u**, the motion model is $\dot{x} = Ax + Bu$.
 - The cost function is setting to the quadratic form $c = \underbrace{x^T Q x}_{\text{State Error}} + \underbrace{u^T R u}_{\text{Minimum Control}}$
, in which **Q** is the state weighting matrix and **R** is the control weighting matrix.
 - The total objective function of an episode $J = \int_0^{Ter} [x(t)^T Q x(t) + u(t)^T R u(t)] dt + x^T(Ter) S x(Ter)$

LQR Control (cont.)

- The goal is to find the optimal control \mathbf{u}^* which minimize the total object function: $\min_u J = \min_u \int_0^T x(t)^T Q x(t) + u(t)^T R u(t) dt + x(T)^T S x(T)$
- To solve this problem, we first introduce the concept of optimal principle. If we have a optimal control sequence $[u_t^*, u_{t+1}^*, u_{t+2}^*, \dots, u_T^*]$, then the subsequence $[u_{t+1}^*, u_{t+2}^*, \dots, u_T^*]$ is also an optimal control sequence.
- Follow the concept, we can apply **dynamic programming** to recursively solve the optimal control from terminal state to current time.

LQR Control (cont.)

- However, we do not know the terminal time or even the terminal time is infinite in most time. In this case, we can solve the LQR using the recursive relation of value function.
- Introduce the value function $\mathbf{V}(\mathbf{x})$, which is the summing of the future cost. We can write down the recursive form of the discrete time value function:

$$V(x_t) = \min_u \{ x_t^T Q x_t + u_t R u_t + V(x_{t+1}) \}$$

- We can guess the value function to be quadratic form $V(x_t) = x_t^T P_t x_t$ (which P is symmetric positive-definite), and apply the linear motion model $Ax_t + Bu_t$ to value function:

$$\begin{aligned} V(x_t) &= \min_u \{ x_t^T Q x_t + u_t R u_t + x_{t+1}^T P_{t+1} x_{t+1} \} \\ &= \min_u \{ x_t^T Q x_t + u_t R u_t + (Ax_t + Bu_t)^T P_{t+1} (Ax_t + Bu_t) \} \\ &= \min_u \{ x_t^T (Q + A^T P_{t+1} A) x_t + 2x_t^T A^T P_{t+1} B u_t + u_t^T (R + B^T P_{t+1} B) u_t \} \end{aligned}$$

LQR Control (cont.)

- Solve the minimum equation

$$V(x_t) = x_t^T P_t x_t = \min_u \{x_t^T (Q + A^T P_{t+1} A) x_t + 2x_t^T A^T P B u + u_t^T (R + B^T P_{t+1} B) u_t\}$$

$$\frac{\partial}{\partial u} [x_t^T (Q + A^T P_{t+1} A) x_t + 2x_t^T A^T P B u_t^* + u_t^{*T} (R + B^T P_{t+1} B) u_t^*] = 0$$

$$2(x_t^T A^T P_{t+1} B)^T + 2(R + B^T P_{t+1} B) u_t^* = 0$$

$$u_t^* = -(R + B^T P_{t+1} B)^{-1} B^T P_{t+1} A x_t$$

- Apply u^* to the value function, and get the equation of P

$$x_t^T P_t x_t = x_t^T (Q + A^T P_{t+1} A - A^T P_{t+1} B (R + B^T P_{t+1} B)^{-1} B^T P_{t+1} A) x_t$$

$$P_t = Q + A^T P_{t+1} A - A^T P_{t+1} B (R + B^T P_{t+1} B)^{-1} B^T P_{t+1} A$$

Discrete Algebra Riccati Equation (DARE)

Remark: In continuous case, $\dot{P} = -PA - A^T P + PBR^{-1}P - Q$
is the **Continuous Algebra Riccati Equation (CARE)**

LQR Control (cont.)

- Given discrete Riccati algebra equation

$$P_t = Q + A^T P_{t+1} A - A^T P_{t+1} B (R + B^T P_{t+1} B)^{-1} B^T P_{t+1} A$$

- Suppose the value function is time-invariant, then

$$P = Q + A^T P A - A^T P B (R + B^T P B)^{-1} B^T P A$$

- In practice, we can first initialize $P^{(0)} = Q$, then iteratively apply the Riccati equation on until converge :

```
INITIALIZE:  $P \leftarrow Q$ 
REPEAT
   $P_{next} \leftarrow Q + A^T P A - A^T P B (R + B^T P B)^{-1} B^T P A$ 
   $\epsilon \leftarrow ||P_{next} - P||$ 
  IF  $\epsilon < threshold$  THEN
    return  $P_{next}$ 
  ENDIF
   $P \leftarrow P_{next}$ 
END
```

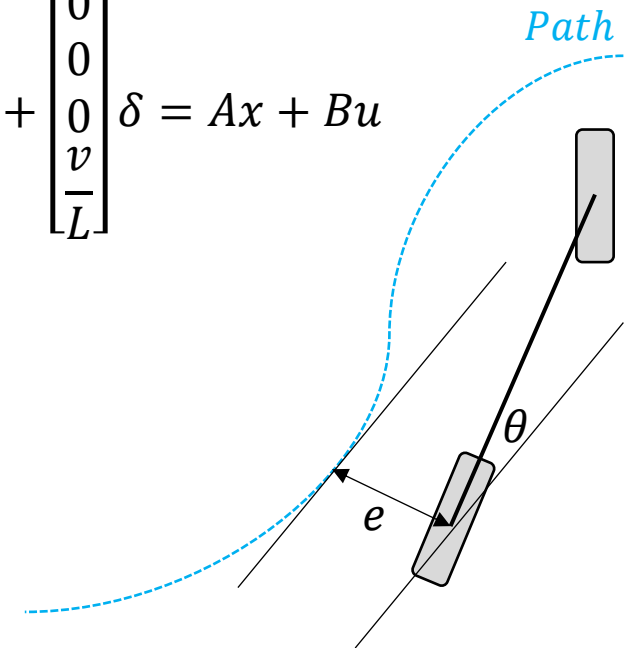

LQR Control for Kinematic Model

- Take an example to solve the LQR optimal control of the kinematic model.
- Define State: $x = [e, \dot{e}, \theta, \dot{\theta}]$, and set the matrix Q and R
- The linear approximate of kinematic motion model:

$$\frac{d}{dt} \begin{bmatrix} e \\ \dot{e} \\ \theta \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} 1 & dt & 0 & 0 \\ 0 & 0 & v & 0 \\ 0 & 0 & 1 & dt \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} e \\ \dot{e} \\ \theta \\ \dot{\theta} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ \frac{v \tan(\delta)}{L} \end{bmatrix} \approx \begin{bmatrix} 1 & dt & 0 & 0 \\ 0 & 0 & v & 0 \\ 0 & 0 & 1 & dt \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} e \\ \dot{e} \\ \theta \\ \dot{\theta} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ \frac{v}{L} \end{bmatrix} \delta = Ax + Bu$$

- Solve the DARE to get the P matrix
- $$P = Q + A^T P A - A^T P B (R + B^T P B)^{-1} B^T P A$$
- Finally, we can get the optimal control

$$u_t^* = -(R + B^T P_{t+1} B)^{-1} B^T P_{t+1} A x_t$$



LQR Control

- Following the steps:
 - Construct the matrix A, B, X of the linear approximation model.

$$Ax + Bu = \underbrace{\begin{bmatrix} 1 & dt & 0 & 0 \\ 0 & 0 & v & 0 \\ 0 & 0 & 1 & dt \\ 0 & 0 & 0 & 0 \end{bmatrix}}_{\mathbf{A}} \underbrace{\begin{bmatrix} e \\ \dot{e} \\ \theta \\ \dot{\theta} \end{bmatrix}}_{\mathbf{x}} + \underbrace{\begin{bmatrix} 0 \\ 0 \\ 0 \\ \frac{v}{L} \end{bmatrix}}_{\mathbf{B}} \underbrace{\delta}_{\mathbf{u}} \text{ (Bicycle Model)}$$

- Solve DARE and get the matrix P of the value function:

$$P = Q + A^T P A - A^T P B (R + B^T P B)^{-1} B^T P A$$

- Compute the optimal control:

$$u_t^* = -(R + B^T P_{t+1} B)^{-1} B^T P_{t+1} A x_t$$

Review of Control Algorithms

$$\delta = K_p e(t) + K_i \sum_0^t e_t + K_d (e(t) - e(t-1))$$

PID Control

Apply the
kinematic property.

Pure-Pursuit
Control

$$\delta = \arctan\left(\frac{2L \sin(\alpha)}{L_d}\right)$$

$$\delta = \arctan\left(-\frac{ke}{v_f}\right) + \theta_e$$

Stanley Control

Consider the
progressive stability.

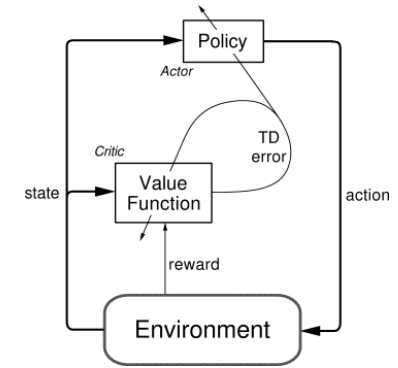
More complex
motion model.

LQR Control

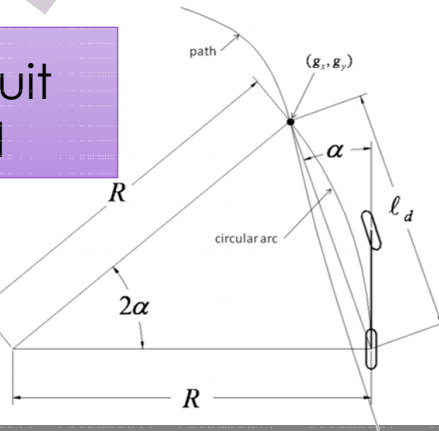
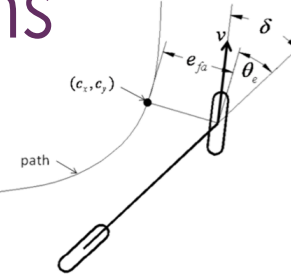
DARE:

$$P_t = Q + A^T P_{t+1} A - A^T P_{t+1} B (R + B^T P_{t+1} B)^{-1} B^T P_{t+1} A$$

**Model-free
Reinforcement Learning**



Don't need model.
Non-linear case.



Q&A