2

Control

★ Control Strategy

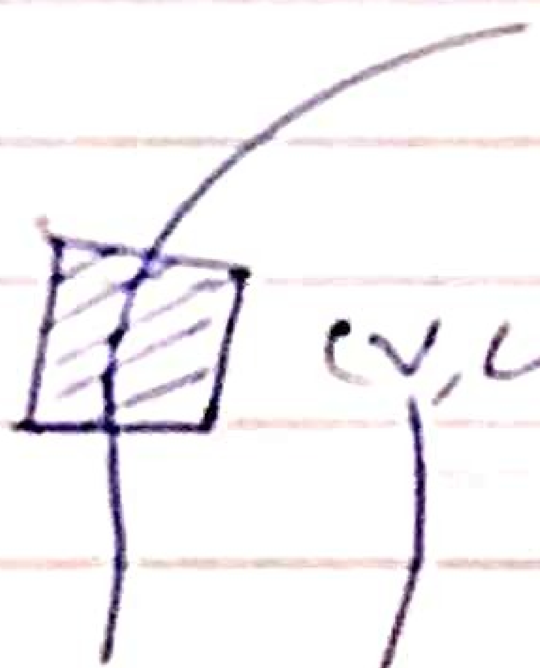
Perception {Traversability Semantics}
↓
Planning { Plan Trajectory}
↓
Control { Longitudinal Lateral}
↓
Actuation {Throttle/Braking Steering}

(Along the path)

(To Reduce the Cross track error)

Trajectory = Path + Velocity information

★ Understanding Motion

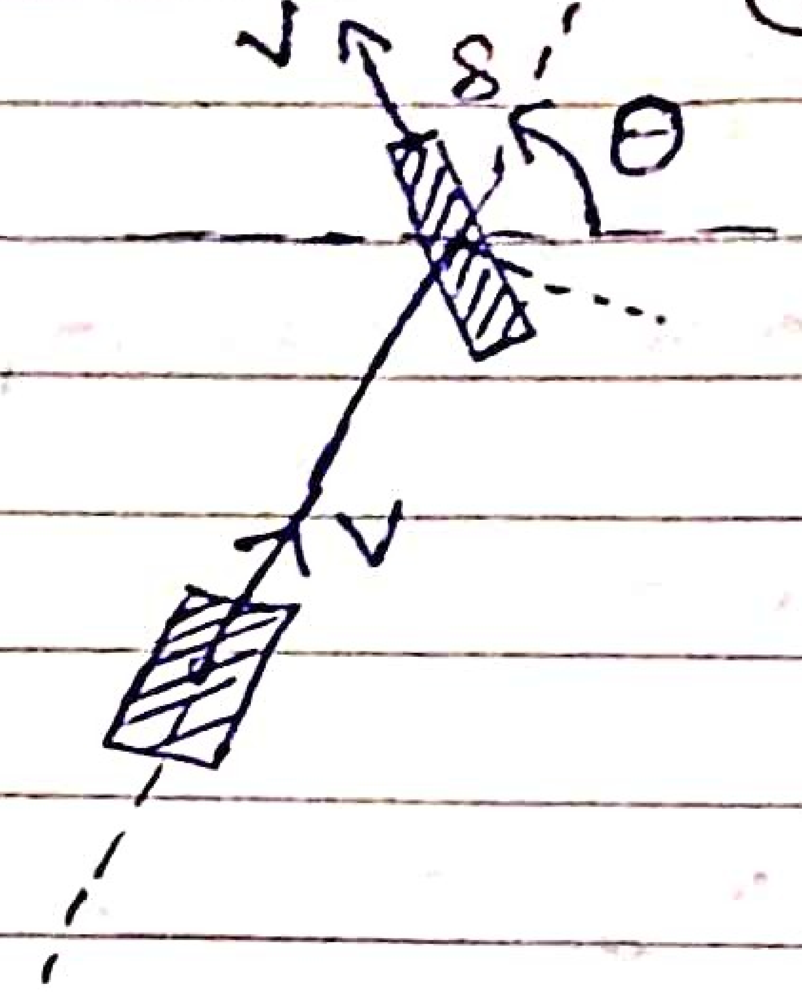


(v, ω)

Translational velocity

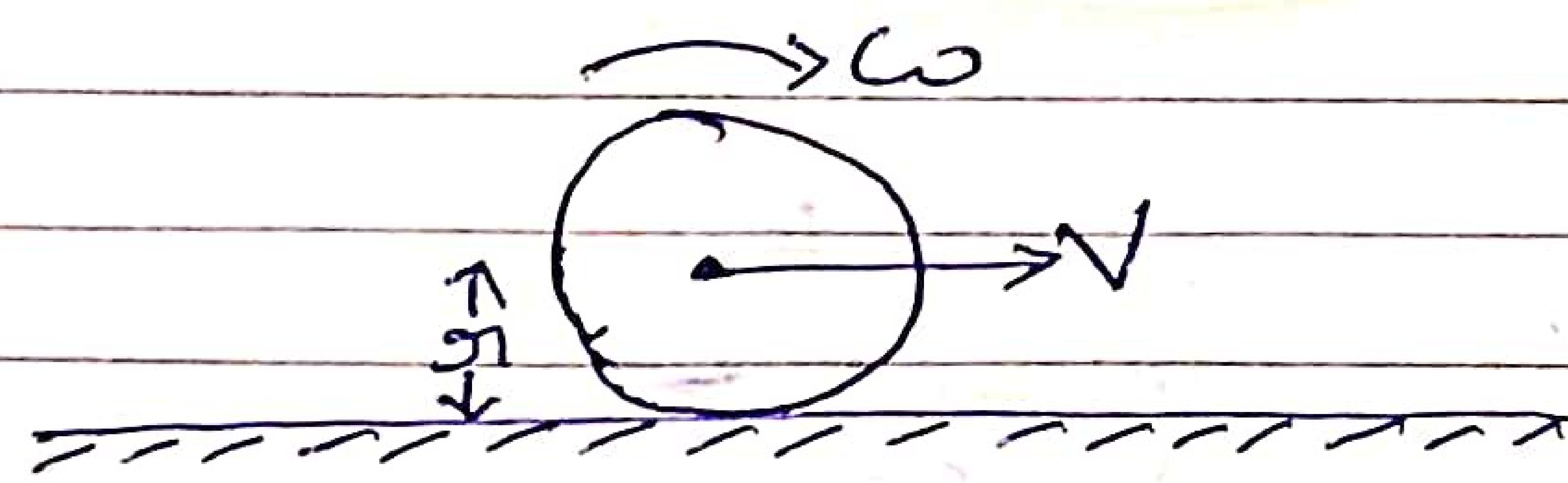
Angular velocity

★ Kinematic modeling (2D Bicycle Model)

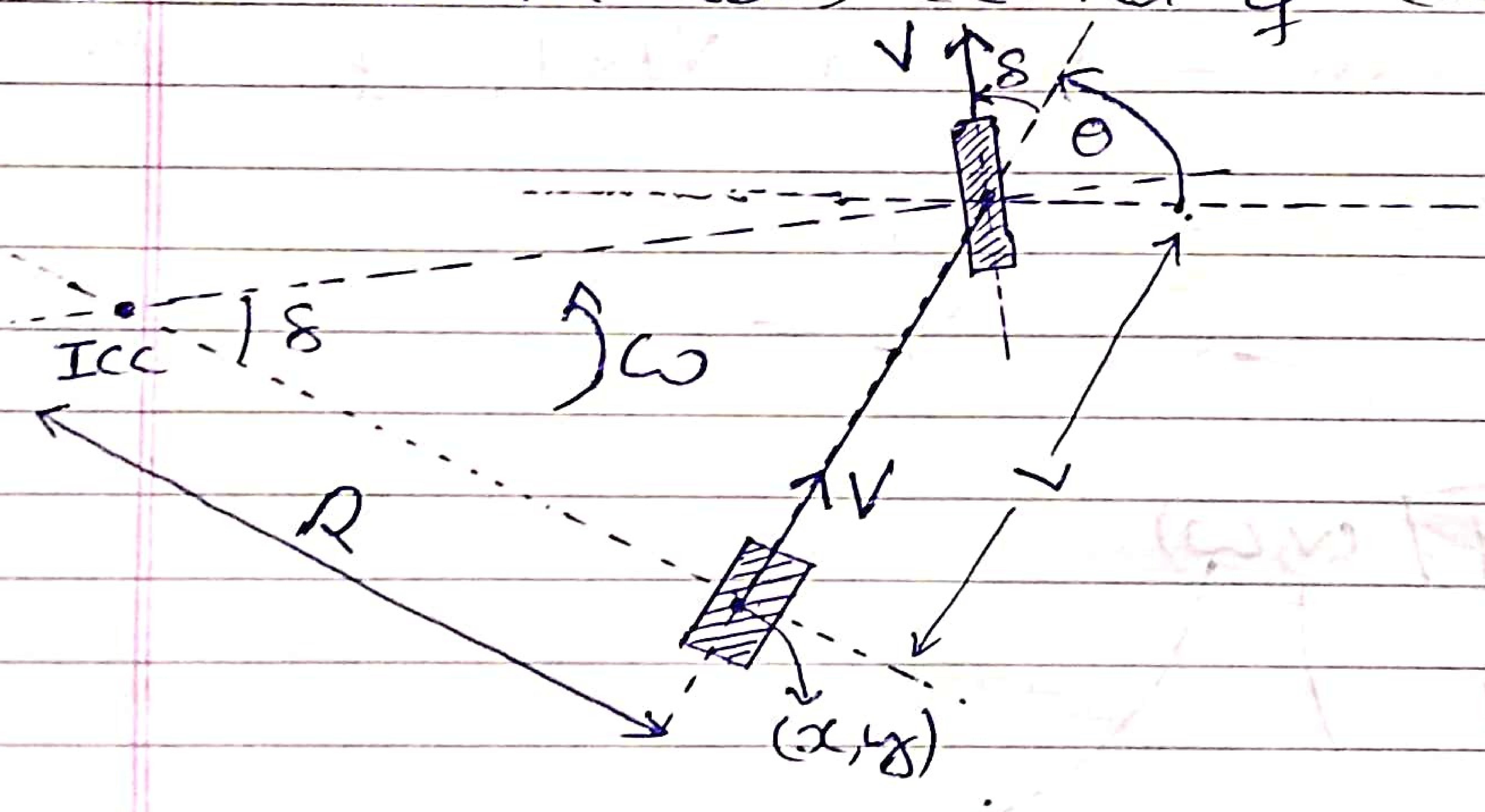


① Rolling Condition for wheels

$$V = r\omega$$



② Instantaneous center of curvature



$$\dot{x} = V\cos(\theta) \quad \text{——} ①$$

$$\dot{y} = V\sin(\theta) \quad \text{——} ②$$

$$\dot{\theta} = \omega = \frac{V}{R} = \frac{V}{L/\tan\delta} = \frac{V\tan\delta}{L} \quad \text{——} ③$$

State:

$[x, y, \theta, \delta]^T$

$\delta = \tan^{-1}\left(\frac{L}{R}\right)$

Control:

$[v, \dot{\delta}]^T$

Kinematics

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

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

$\dot{\theta} = \frac{v\tan(\delta)}{L}$

Constraints:

$v < v_{max}$

$\delta < |\delta_{max}|$

★ **Vehicle actuation**

★ **Steering Model**

$\delta_s = K_s \delta$

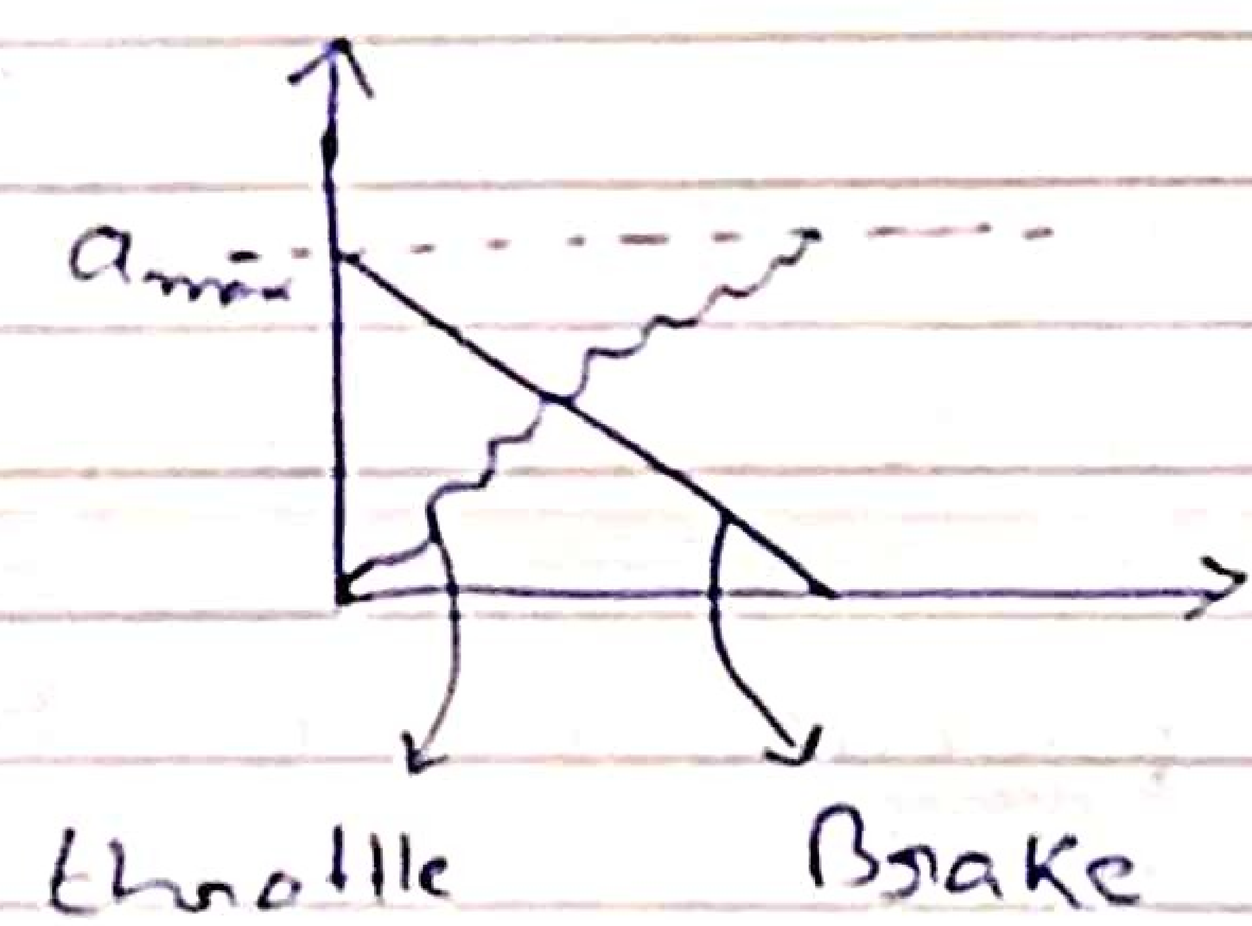
$\delta < |\delta_{max}|$

★ **Throttle / Brake**

$\delta_t = K_t a$

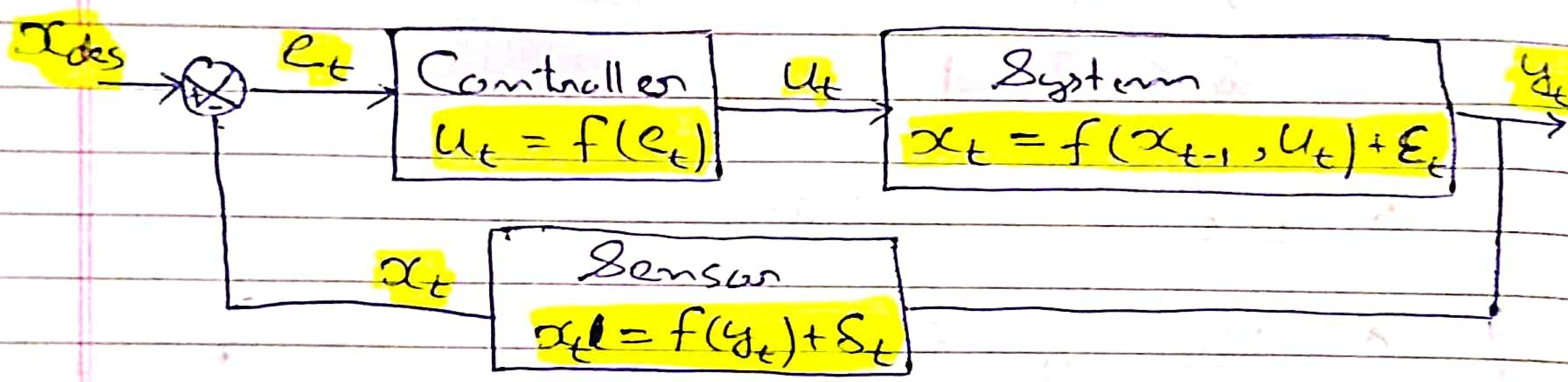
$\delta_b = -K_b a$

$a < a_{max}$



★ Feedback Control

⇓



★ PID Controller

⇒ Idea: $u_t = K_P(x_{des} - x_t) + K_D(\dot{x}_{des} - \dot{x}_t)$

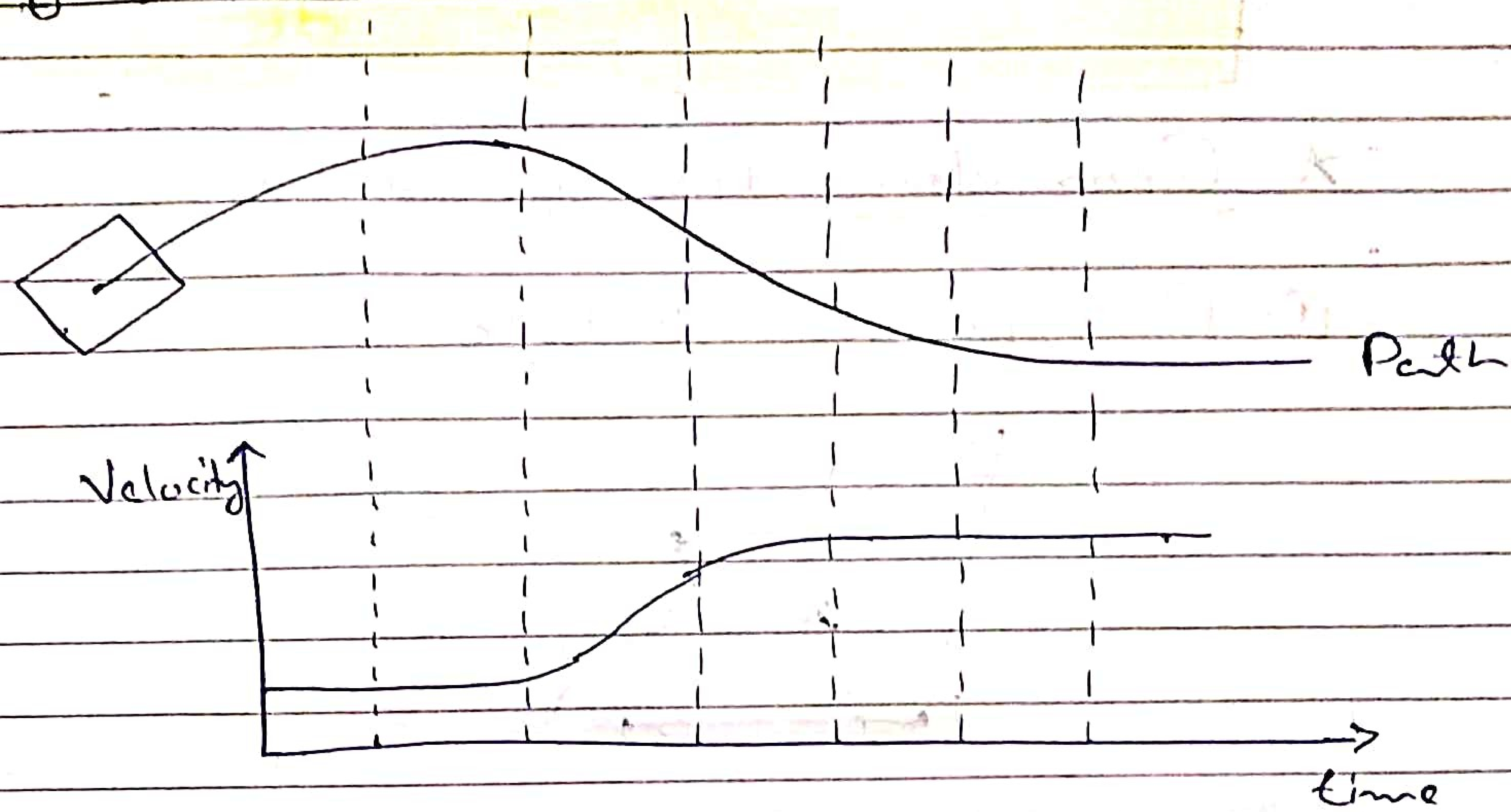
$$+ K_I \int_0^t (x_{des} - x_t)\,dt$$

★ How to follow a trajectory?

⇒ This problem can be brocken up into couple of parts:

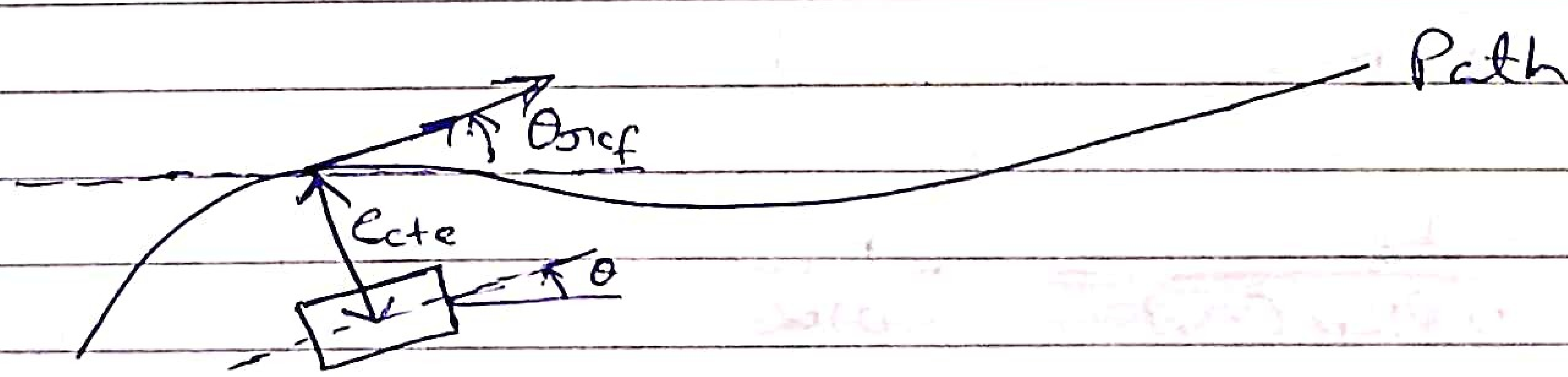
1. Longitudional Control
2. Laterd Control

★ Longitudinal Control



$$\ddot{x}_{des} = K_p(\dot{x}_{ref} - \dot{x}) + K_d \frac{d(\dot{x}_{ref} - \dot{x})}{dt} + K_I \int_0^t (\dot{x}_{ref} - \dot{x})\,dt$$

★ Lateral Control (To minimize cross-track error & heading / orientation error)



⇒ Two goals of lateral controller:

1. To minimize cross-track error (e_{cte})
2. To minimize heading / orientation error ($\theta_{ref} - \theta$)

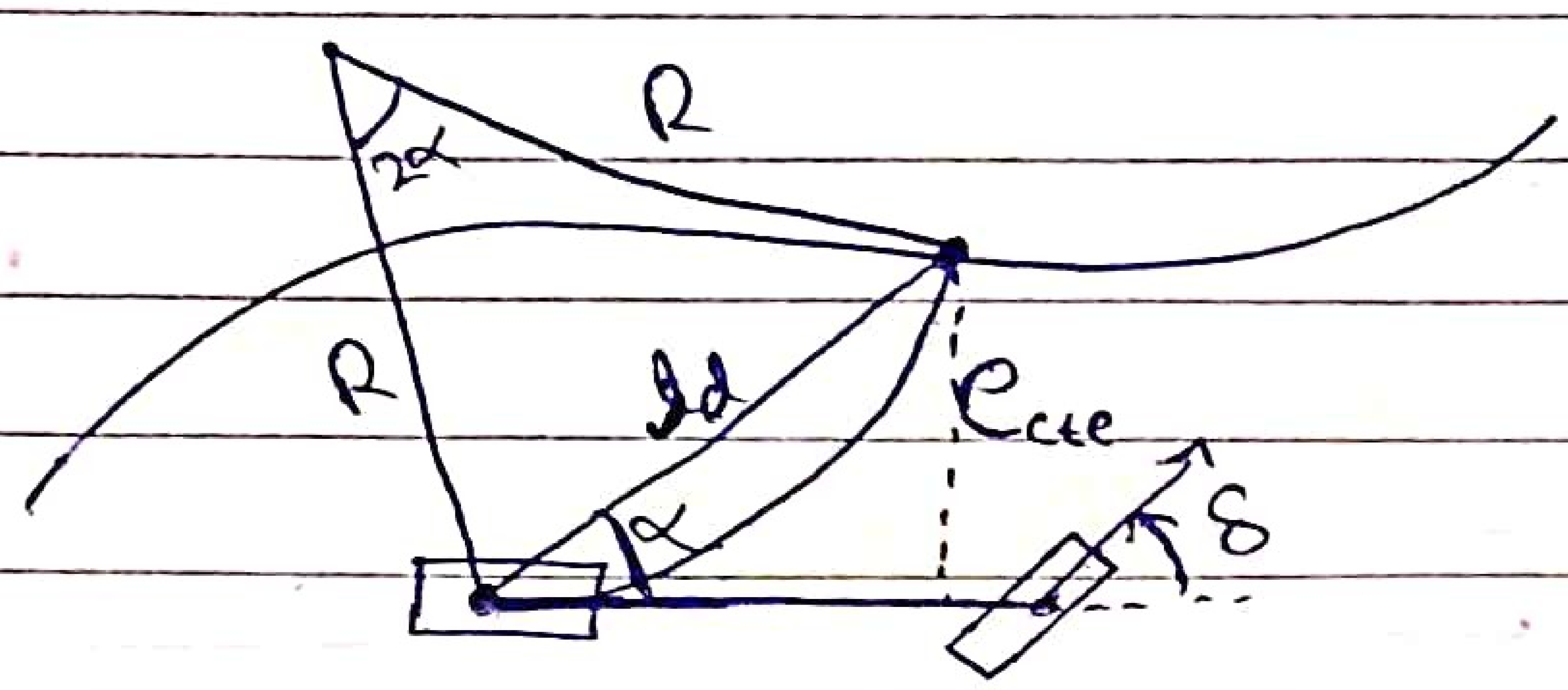
Date
Page

③ PID Control law

$$\dot{\delta}_{des} = -K_p e_{cte} - K_d \frac{de_{cte}}{dt} - K_I \int_0^t e_{cte}\, dt$$

★ Geometric steering Control

① Pure persuit controller



⇒ Here, instead of tracking closest point on the reference path, we track a point at a certain look-ahed distance.

l_d = look-ahed distance

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

$$\Rightarrow \frac{l_d}{2\sin\alpha\cos\alpha} = \frac{R}{\cos\alpha}$$

$$\Rightarrow \frac{l_d}{2\sin\alpha} = R$$

$$\boxed{R = \frac{l_d}{2\sin\alpha}}$$

$\tan\delta = \frac{L}{R}$ {From Kinematic modeling}

$$\boxed{\delta = \tan^{-1}\left(\frac{2L\sin\alpha}{l_d}\right)}$$

{Control law of pure pursuit Controller.}

$$\sin\alpha = \frac{e_{cte}}{l_d}$$

$$\Rightarrow \delta = \tan^{-1}\left(\frac{2L\,e_{cte}}{l_d^2}\right)$$

{So as $e_{cte} \to 0$ $\delta \to 0$
hence e_{cte} is controlled}

⇒ If we are going very fast and l_d is small then, this can lead to very fast manoeuvre of steering wheel.

⇒ One way to solve the above problem is by setting look ahead distance proportional to velocity

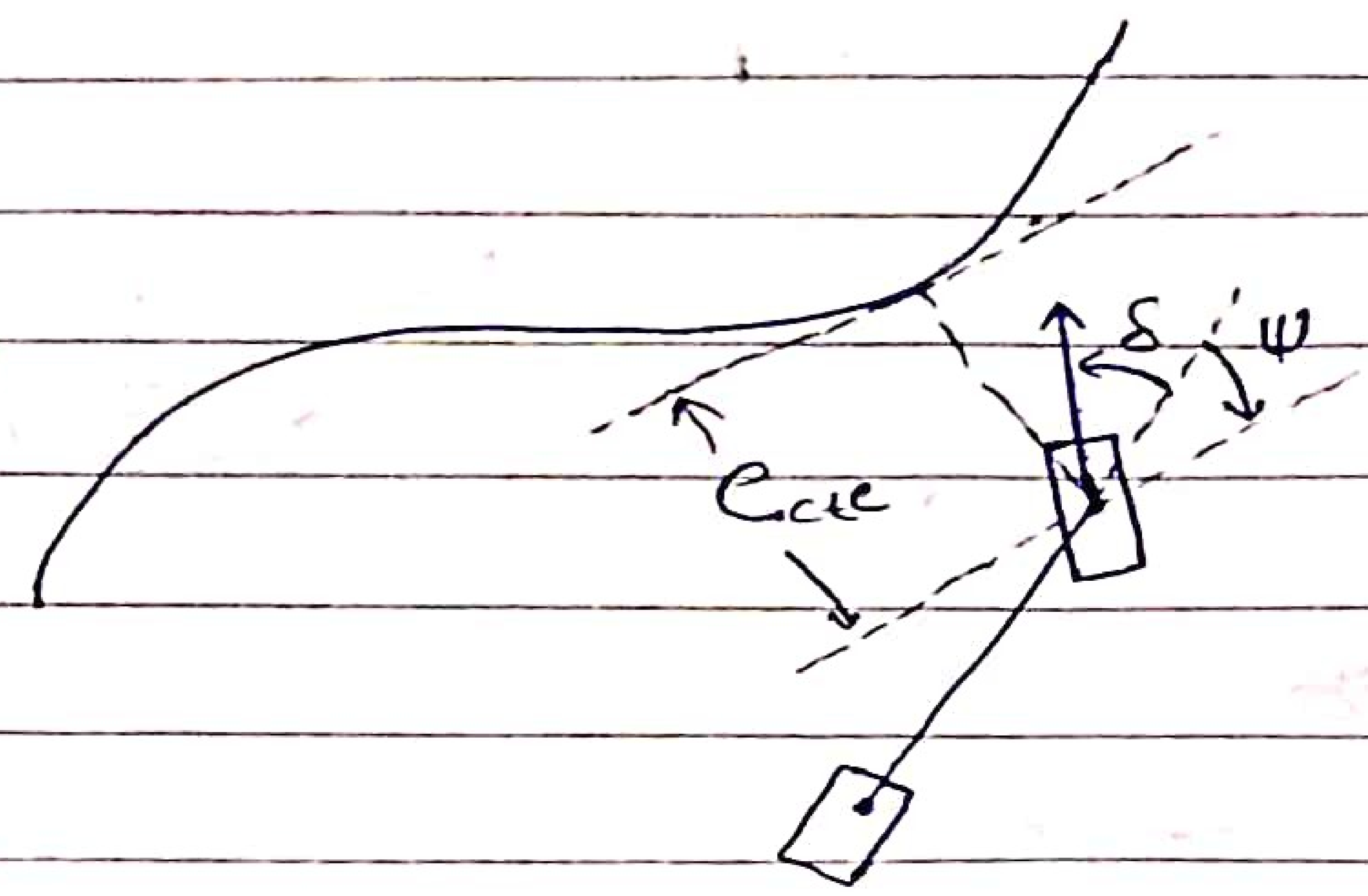
$$\boxed{l_d = K_{dd} V}$$

$$\boxed{\delta = \tan^{-1}\left(\frac{2L\sin\alpha}{K_{dd} V}\right)}$$

Date
Page

② Stanley Controller

⇒ Reduce both the error in heading and cross track error.



ⓐ Align Heading:

$$\delta = \psi$$

ⓑ Cross-track error:

$$\delta = \tan^{-1}\left(\frac{K e_{cte}}{V}\right) \quad \left\{\text{Proposed empirically}\right\}$$

ⓒ ~~Cross~~ Steering limits:

$$\delta \in [\delta_{min}, \delta_{max}]$$

⇒ Combined control law:

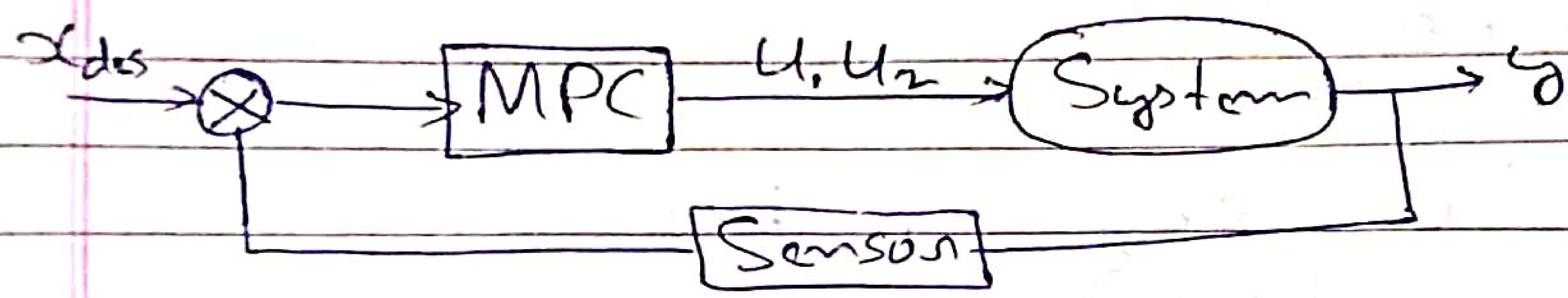
$$\boxed{\delta = \psi + \tan^{-1}\left(\frac{K e_{cte}}{V}\right)}$$

$$\delta \in [\delta_{min}, \delta_{max}]$$

★ Model Predictive Control

⇒ Uses a model of the system to make future predictions for the system.

⇒ Handle multiple Input/Output jointly.



⇒ Handles constraints naturally in optimization.

⇒ Discrete-time linear/non-linear System:

$$x_{t+1} = f(x_t, u_t)$$

⇒ Quadratic cost function

$$J = \sum \left(e_t^T Q e_t + u_t^T R u_t \right)$$

$$e_t = x_{des} - x_t$$

⇒ Goal: Find the control with the lowest cost.

⇒ No closed form solution, must be solved numerically.

↳ Feasible only for small planning horizon.

for K step

⇒ After performing the optimization, we apply control obtained for one step.

→ Then repeate the same process again.

★ **MPC design parameters**

① **Prediction Horizon**

→ How many step ahead do we look into.

② **Sample Time**

→ Frequency at which we need to send commands to car.

③ **Control Horizon**

→ Number of steps for which control will be calculated.

④ **Constraints**

max velocity, max acceleration
max steering angle etc...

⑤ **Weights**

i.e. Q & R