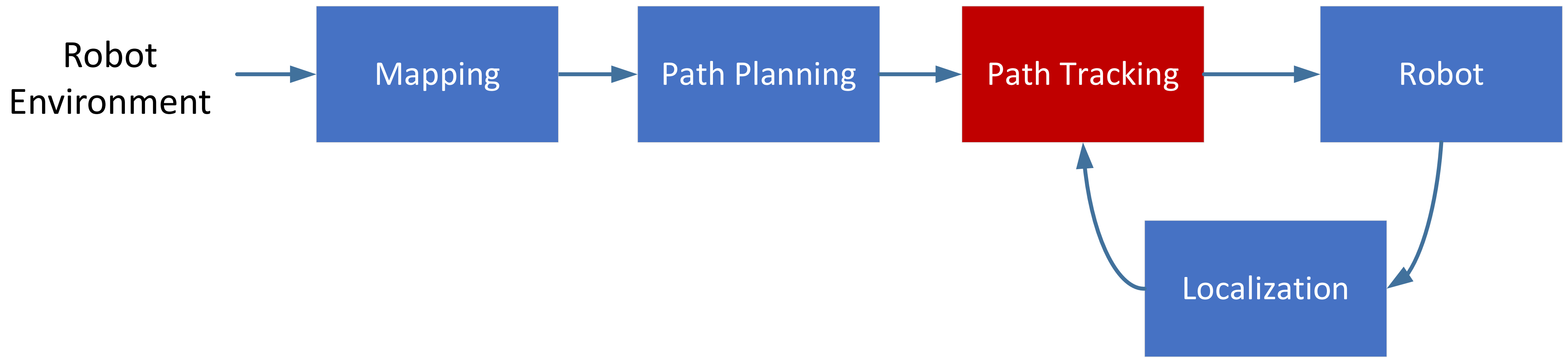
MOBILE ROBOT POSE CONTROL

Djoko Purwanto
Dept. of Electrical Engineering, ITS
djoko@its.ac.id

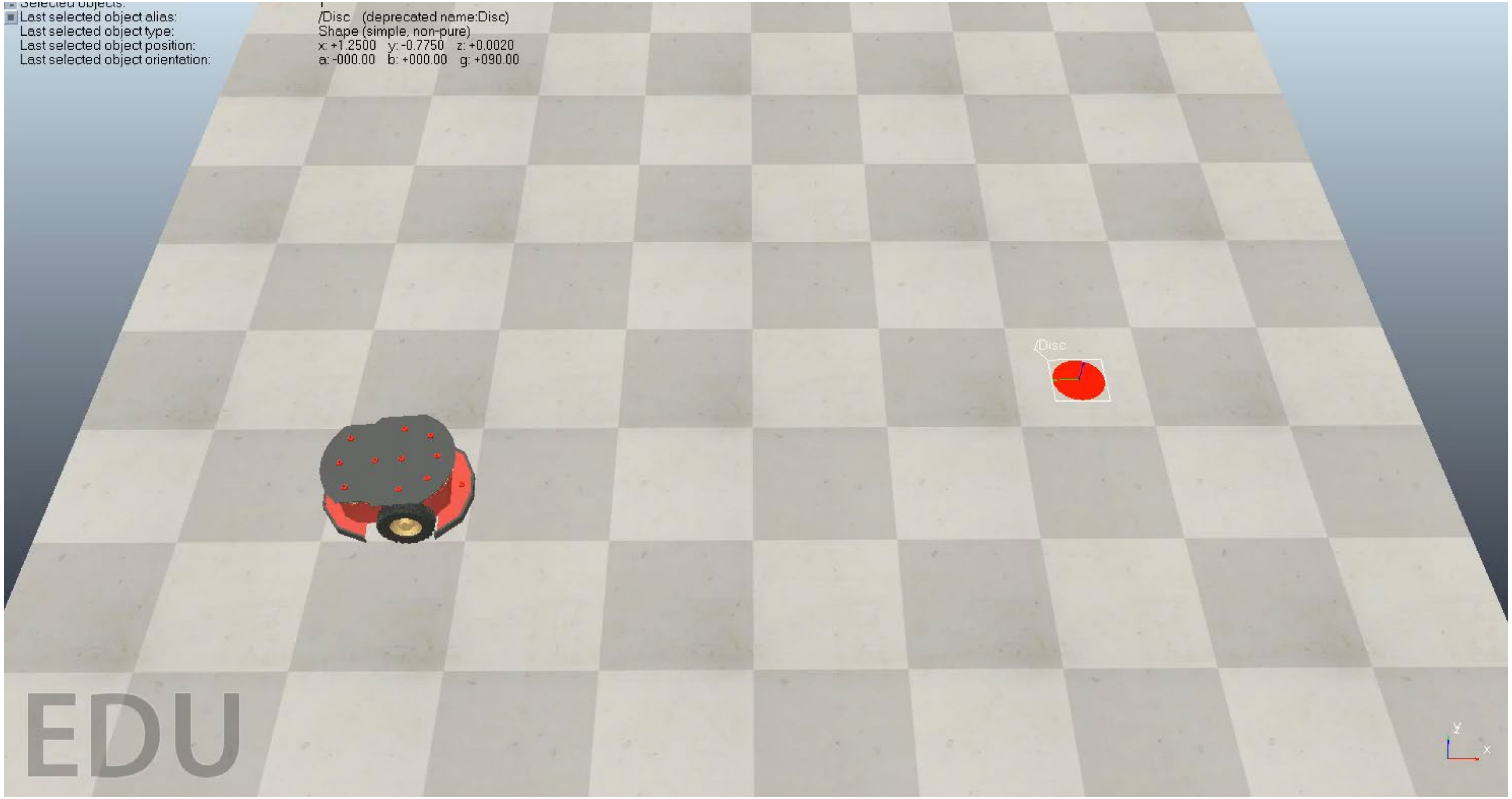
INTRODUCTION

Robot Navigation

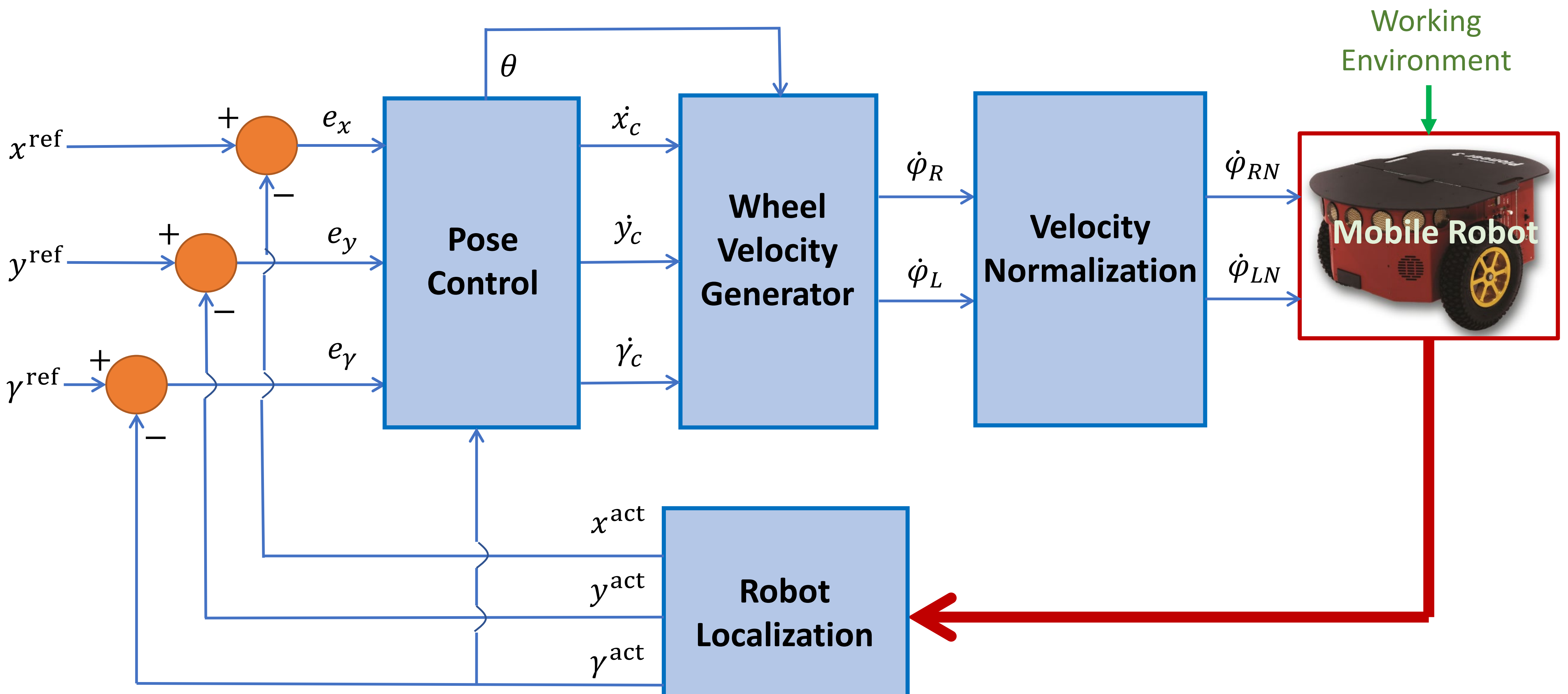


Pose Control is essential for accurate and reliable path tracking, ensuring the system can follow the desired trajectory with minimal errors and maximum stability.

POSE CONTROL



POSE CONTROL SYSTEM



Robot Localization

Get robot localization from Pioneer P3DX model in CoppeliaSim

Pose Control

$$\theta = \mathrm{tg}^{-1}\left(\frac{e_y}{e_x}\right)$$

$$\dot{x}_c = \begin{cases} 0 & |e_x| \le e_{\mathrm{tol}} \text{ and } |e_y| \le e_{\mathrm{tol}} \\ K_1 e_x & \text{otherwise} \end{cases}$$

$$\dot{y}_c = \begin{cases} 0 & |e_x| \le e_{\mathrm{tol}} \text{ and } |e_y| \le e_{\mathrm{tol}} \\ K_2 e_y & \text{otherwise} \end{cases}$$

$$\dot{\gamma}_c = \begin{cases} K_3 e_\gamma & |e_x| \le e_{\mathrm{tol}} \text{ and } |e_y| \le e_{\mathrm{tol}} \\ \theta - \gamma^{\mathrm{act}} & \text{otherwise} \end{cases}$$

Wheel Velocity Generator

$$\begin{bmatrix}\dot{\varphi}_R\\ \dot{\varphi}_L\end{bmatrix} = \begin{bmatrix}\frac{R}{2}\cos\theta & \frac{R}{2}\cos\theta\\ \frac{R}{2}\sin\theta & \frac{R}{2}\sin\theta\\ \frac{R}{2L} & -\frac{R}{2L}\end{bmatrix}^{+} \begin{bmatrix}\dot{x}_c\\ \dot{y}_c\\ \dot{\gamma}_c\end{bmatrix}$$

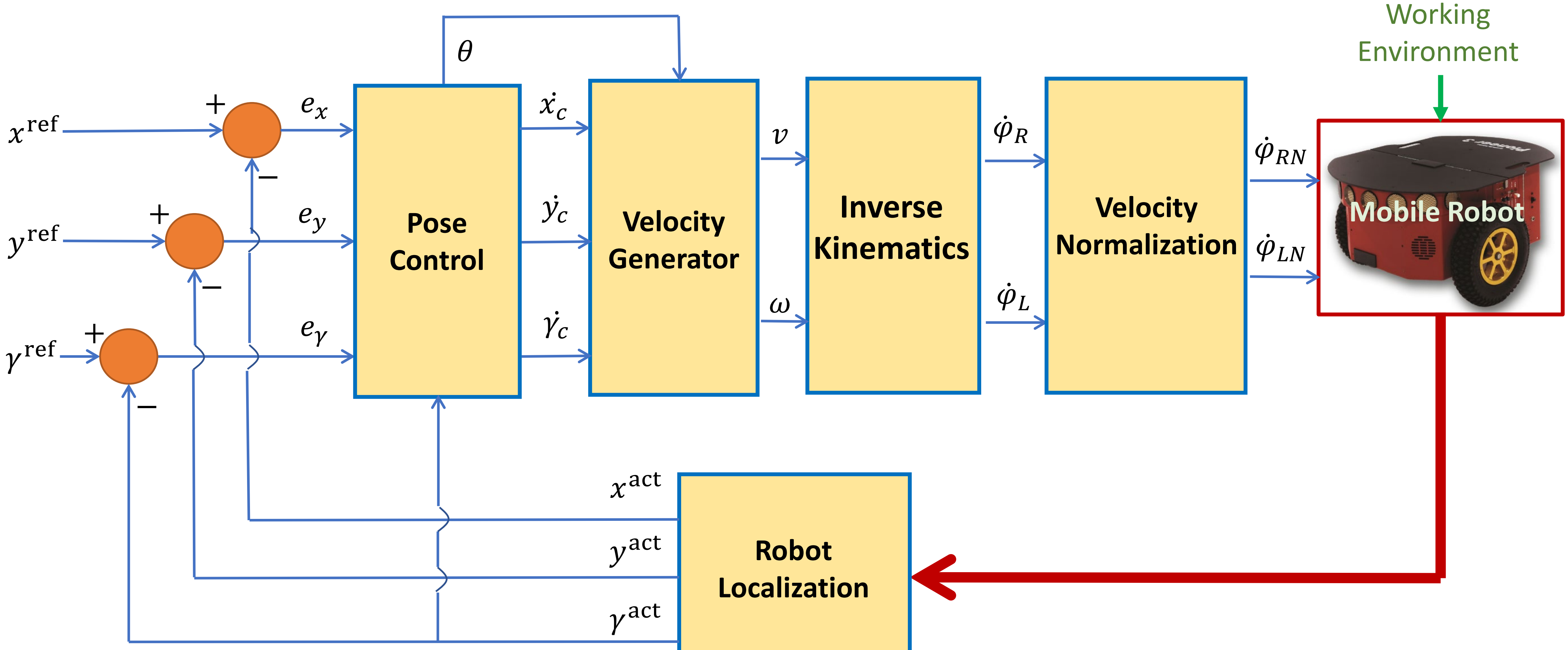
Velocity Normalization

$$\dot{\varphi}_{\max} = \max(\dot{\varphi}_R, \dot{\varphi}_L)$$

$$\dot{\varphi}_{RN} = \begin{cases}\frac{\dot{\varphi}_{\text{norm}}}{\dot{\varphi}_{\max}}\dot{\varphi}_R & \dot{\varphi}_{\max} > \dot{\varphi}_{\text{norm}}\\ \dot{\varphi}_R & \text{otherwise}\end{cases}$$

$$\dot{\varphi}_{LN} = \begin{cases}\frac{\dot{\varphi}_{\text{norm}}}{\dot{\varphi}_{\max}}\dot{\varphi}_L & \dot{\varphi}_{\max} > \dot{\varphi}_{\text{norm}}\\ \dot{\varphi}_L & \text{otherwise}\end{cases}$$

POSE CONTROL SYSTEM (2)



Robot Localization

Get robot localization from Pioneer P3DX model in CoppeliaSim

Pose Control

$$\theta = \mathrm{tg}^{-1}\left(\frac{e_y}{e_x}\right)$$

$$\dot{x}_c = \begin{cases} 0 & |e_x| \leq e_{\mathrm{tol}} \text{ and } |e_y| \leq e_{\mathrm{tol}} \\ K_1 e_x & \text{otherwise} \end{cases}$$

$$\dot{y}_c = \begin{cases} 0 & |e_x| \leq e_{\mathrm{tol}} \text{ and } |e_y| \leq e_{\mathrm{tol}} \\ K_2 e_y & \text{otherwise} \end{cases}$$

$$\dot{\gamma}_c = \begin{cases} K_3 e_\gamma & |e_x| \leq e_{\mathrm{tol}} \text{ and } |e_y| \leq e_{\mathrm{tol}} \\ \theta - \gamma^{\mathrm{act}} & \text{otherwise} \end{cases}$$

Velocity Generator

$$\begin{bmatrix} v \\ \omega \end{bmatrix} = \begin{bmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ 0 & 1 \end{bmatrix}^{+} \begin{bmatrix} \dot{x}_c \\ \dot{y}_c \\ \dot{\gamma}_c \end{bmatrix}$$

Inverse Kinematics

$$\begin{bmatrix} \dot{\varphi}_R \\ \dot{\varphi}_L \end{bmatrix} = \begin{bmatrix} \frac{R}{2} & \frac{R}{2} \\ \frac{R}{2L} & -\frac{R}{2L} \end{bmatrix}^{-1} \begin{bmatrix} v \\ \omega \end{bmatrix}$$

Velocity Normalization

$$\dot{\varphi}_{\max} = \max(\dot{\varphi}_R, \dot{\varphi}_L)$$

$$\dot{\varphi}_{RN} = \begin{cases} \frac{\dot{\varphi}_{\text{norm}}}{\dot{\varphi}_{\max}} \dot{\varphi}_R & \dot{\varphi}_{\max} > \dot{\varphi}_{\text{norm}} \\ \dot{\varphi}_R & \text{otherwise} \end{cases}$$

$$\dot{\varphi}_{LN} = \begin{cases} \frac{\dot{\varphi}_{\text{norm}}}{\dot{\varphi}_{\max}} \dot{\varphi}_L & \dot{\varphi}_{\max} > \dot{\varphi}_{\text{norm}} \\ \dot{\varphi}_L & \text{otherwise} \end{cases}$$

ASSIGNMENT 4

1. Open the CoppeliaSim, pick and place the Pioneer P3DX mobile robot and place a simple object as the destination pose of robot motion. (see the figure).
2. Create the computer programming to control the robot's motion from the initial to the destination position. Refer to the explanation on pages 4-6 to realize the system.

THANK YOU