Tampere University

Fundamental of Mobile Robot
AUT-710

Exercise 6

Widhi Atman

31.3.2023, **RN201,** 10:15 - Finish

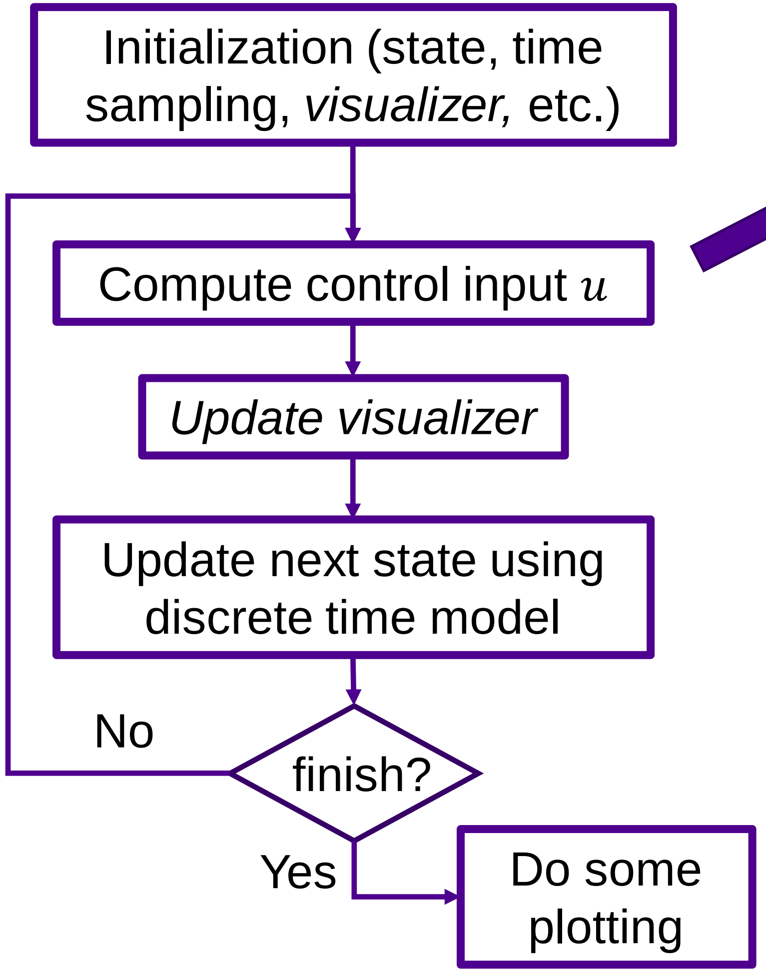
General Plan for Exercises

- Exercise 4: Implementation of model + Basic Control — **10 point**
- Exercise 5: Collision Avoidance with SI model — **20 point**
- Exercise 6: Control of Unicycle — **20 point**
  - Go to goal
  - Trajectory Tracking (Position and Posture cases)
  - Obstacle Avoidance towards Waypoints

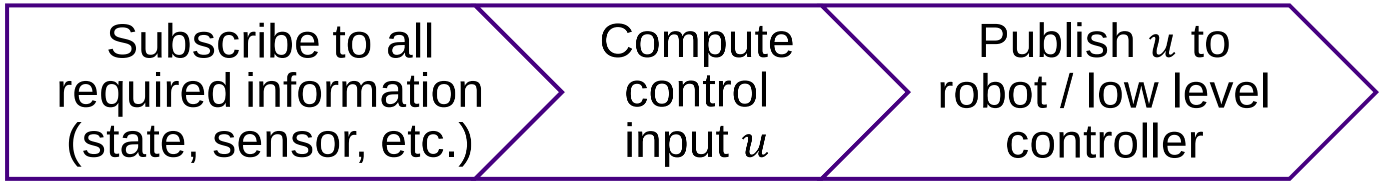
  Deadline: Monday 24.4.2022 at 23:59

- Mini Project **Deadline: Friday 5.5.2022 at 23:59** — **Bonus up to 20 point**

Flowchart of Simulator



If you are interested in implementing this to ROS

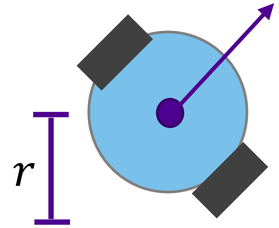


Specifications (for Exercise 6)

Time sampling $T = 10ms$

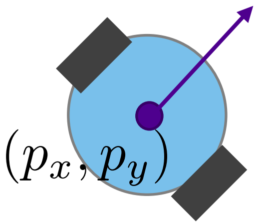
Robot's radius = $0.21\ m$
Robot's wheel radius = $0.1\ m$

Max wheel rotational speed (ω_l & ω_r) = 10 rad/s

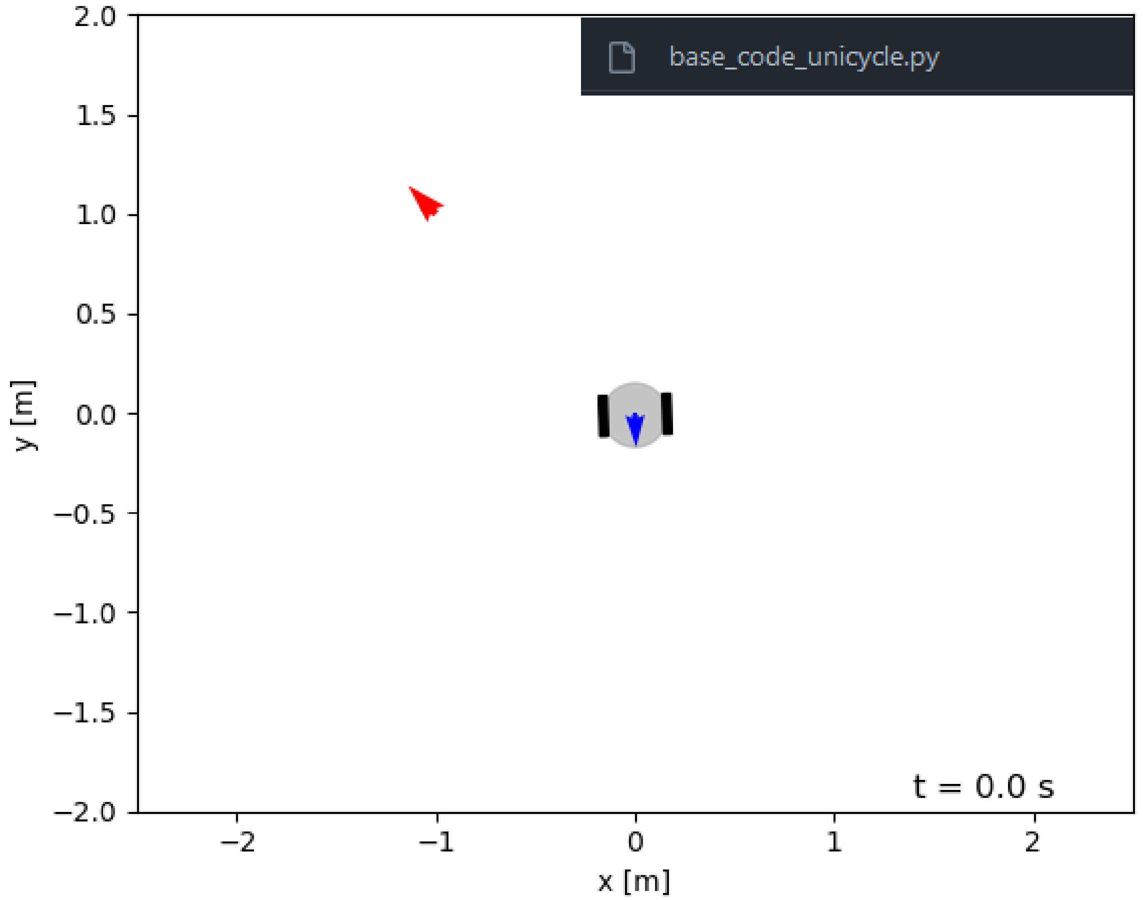


Tampere University

Exercise 6.1 – Scenario



$$x = \begin{bmatrix} p_x \\ p_y \\ \theta \end{bmatrix} \quad u = \begin{bmatrix} v \\ \omega \end{bmatrix}$$



Model: unicycle mobile robot

Initial Position: $x[0] = \begin{bmatrix} 0 & 0 & \frac{\pi}{2} \end{bmatrix}^T$

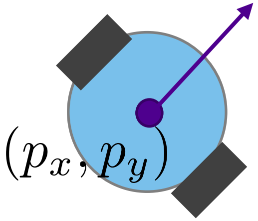
Goal: static at $x^d = [-1 \quad 1 \quad *]^T$.

* Can be any orientation at goal position

Control Objective:

- Reach the goal
- The robot moves within the allowed wheel rotational speed

Exercise 6.1 – Task **4 points**



$$x = \begin{bmatrix} p_x \\ p_y \\ \theta \end{bmatrix} \quad u = \begin{bmatrix} v \\ \omega \end{bmatrix}$$

By taking account of the robot’s size and limitation, design and implement the following **go-to-goal controller:**

a. $\begin{bmatrix} v \\ \omega \end{bmatrix} = \begin{bmatrix} \sqrt{\bar{u}_x^2 + \bar{u}_y^2} \\ k_\theta(\theta_d - \theta) \end{bmatrix}$ (lecture 11 slide 7)

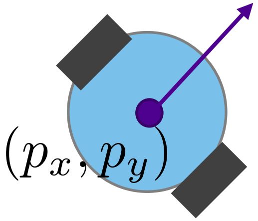
b. $\begin{bmatrix} v \\ \omega \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1/\ell \end{bmatrix} \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} \bar{u}_x \\ \bar{u}_y \end{bmatrix}$ (lecture 11 slide 9)

$\bar{u}_x$ and $\bar{u}_y$ is a designed control input for the single integrator model.

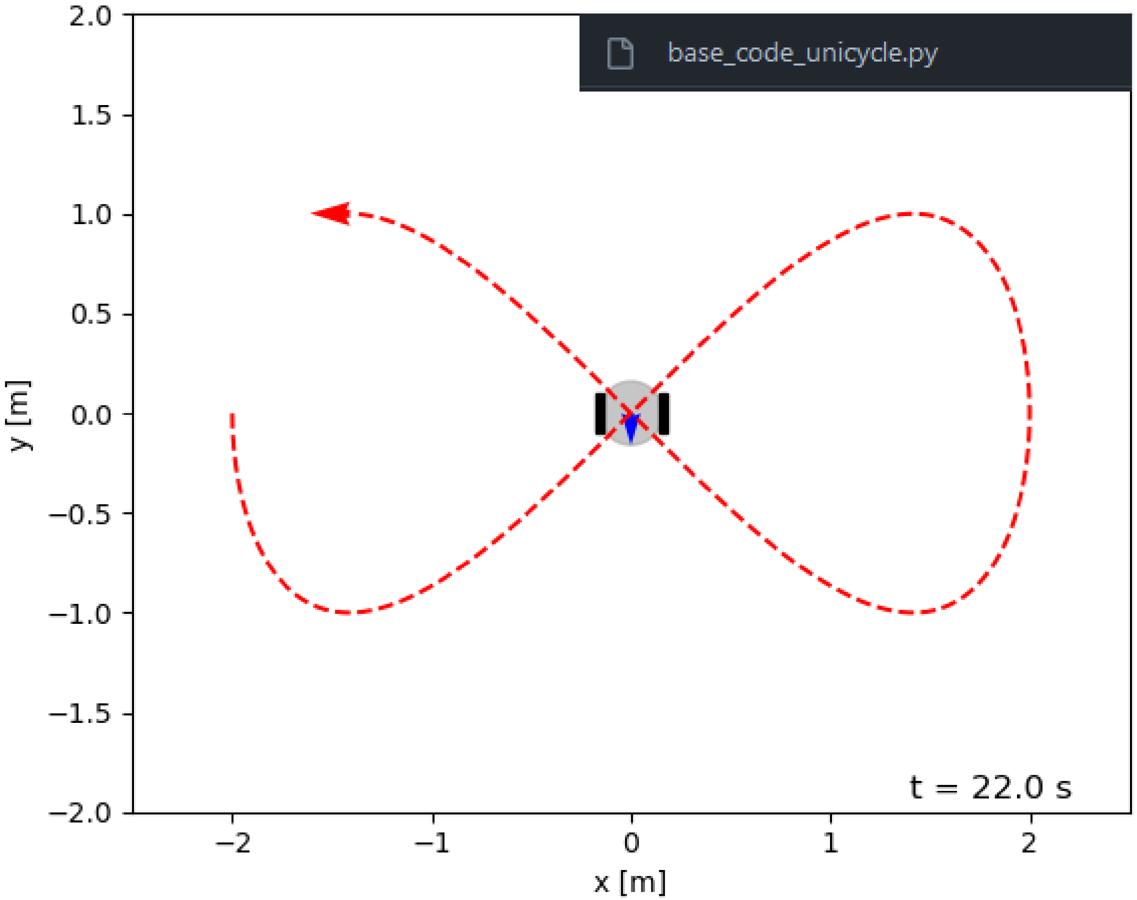
Describe your design approach as well as your observation on the resulting controller.

Show the result by plotting:
- *time series* of control input u, $\bar{u}_x$ and $\bar{u}_y$
- *time series* of wheel speed (ω_l & ω_r)
- *time series* of state trajectory x vs x^d
- XY trajectory of the robot

Exercise 6.2 – Scenario



$$x = \begin{bmatrix} p_x \\ p_y \\ \theta \end{bmatrix} \quad u = \begin{bmatrix} v \\ \omega \end{bmatrix}$$



Model: unicycle mobile robot

Initial Position: $x[0] = \begin{bmatrix} 0 & 0 & -\frac{\pi}{2} \end{bmatrix}^T$

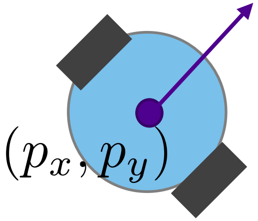
Goal: moving at

$$p_x^d[t] = -2\cos(0.25t)$$
$$p_y^d[t] = -\sin(0.5t)$$
$$\theta^d[t] = \mathrm{atan2}(\dot{p}_y^d, \dot{p}_x^d)$$

Control Objective:

- Track the trajectory of the goal
- The robot moves within the allowed wheel rotational speed

Exercise 6.2 – Task **10 points**



$$x = \begin{bmatrix} p_x \\ p_y \\ \theta \end{bmatrix} \quad u = \begin{bmatrix} v \\ \omega \end{bmatrix}$$

By taking account of the robot's size and limitation,
design and implement the following **trajectory tracking controller:**

a. For position case (lecture 11 slide 10-12)

b. For posture case (lecture 11 slide 13-20)

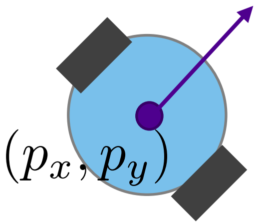
Describe your design approach as well as your observation on the resulting controller.

Show the result by plotting:

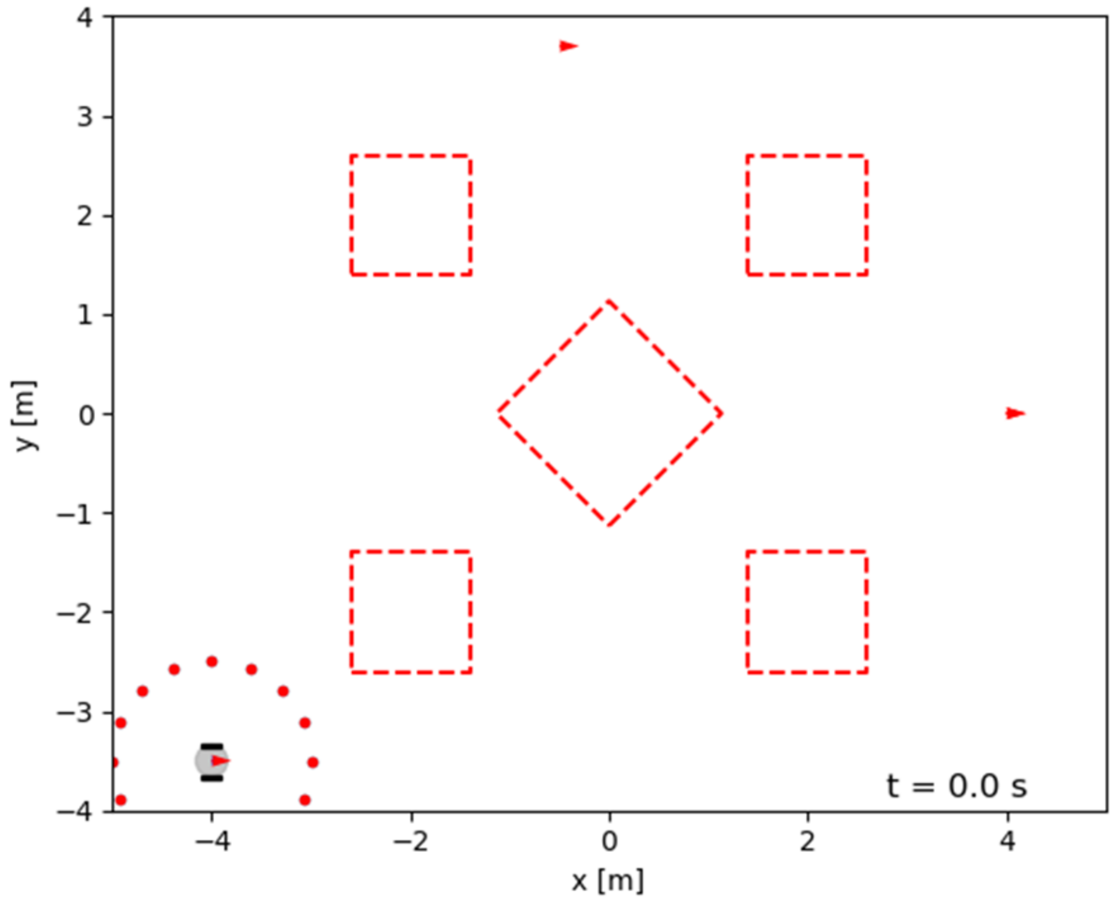
- *time series* of control input u
- *time series* of wheel speed (ω_l & ω_r)
- *time series* of state trajectory x vs x^d
- XY trajectory of the robot

Tampere University

Exercise 6.3 – Task



$x = \begin{bmatrix} p_x \\ p_y \\ \theta \end{bmatrix}$ $u = \begin{bmatrix} v \\ \omega \end{bmatrix}$



Model: unicycle mobile robot

Initial Position: $x[0] = [-4 \quad -3.5 \quad 0]^T$

Goal: 3 waypoints at

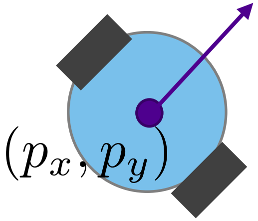
$$x_1^d = [4 \quad 0 \quad *]^T.$$
$$x_2^d = [-0.5 \quad 3.7 \quad *]^T.$$
$$x_3^d = [-4 \quad -3.5 \quad *]^T.$$

* Can be any orientation at each waypoint

Control Objective:

- Reach each waypoint $x_1^d \rightarrow x_2^d \rightarrow x_3^d$, switch to the next waypoint when the distance is less than 0.2m
- The robot moves within the allowed wheel rotational speed

Exercise 6.3 – Task **6 points**



$$x = \begin{bmatrix} p_x \\ p_y \\ \theta \end{bmatrix} \quad u = \begin{bmatrix} v \\ \omega \end{bmatrix}$$

By taking account of the robot’s size and limitation,
design and implement the suitable **controller**
using *the knowledge that you have from the course*.

Describe your design approach as well as your observation on the resulting controller.

Show the result by plotting:

- *time series* of control input u
- *time series* of wheel speed (ω_l & ω_r)
- *time series* of state trajectory x vs x^d
- XY trajectory of the robot
- any other plot that relevant to your controller design

Obstacles for 6.3

```
from ex6p3_obstacles import dict_obst_vertices
```

Import from the **exp6p3_obstacles.py** (put the file in the same folder)

Register the obstacles to be detected by the range sensor

```
    # Initiate the Obstacle Detection
    range_sensor = DetectObstacle( sensing_range, sensor_resolution)
    for key, value in dict_obst_vertices.items():
        range_sensor.register_obstacle_bounded( value )
```

Display the obstacles in the plot

```
    # Display the obstacle
    for key, value in dict_obst_vertices.items():
        sim_visualizer.ax.plot( value[:,0], value[:,1], '--r' )
```

Question?

- Consult them via
  - Exercise sessions on 31.3.2023, 14.4.2023, and 21.4.2023
  - Teams channel