Fundamental of Mobile Robot
AUT-710

Exercise 4

Widhi Atman

10.3.2023, **RN201,** 10:15 - Finish

Exercises Plan

- Exercise 4: Implementation of model + Basic Control
  - SI Go-to-goal → Proportional Control
  - SI Trajectory Tracking → Proportional + Feedforward Control
  - Unicycle Go-to-goal → Proportional Control for Orientation

10 point

Deadline: Monday 20.3.2022 at 23:59

- Exercise 5: Collision Avoidance with SI model
- Exercise 6: Control of Unicycle
- Mini Projects (optional)

20 point

20 point

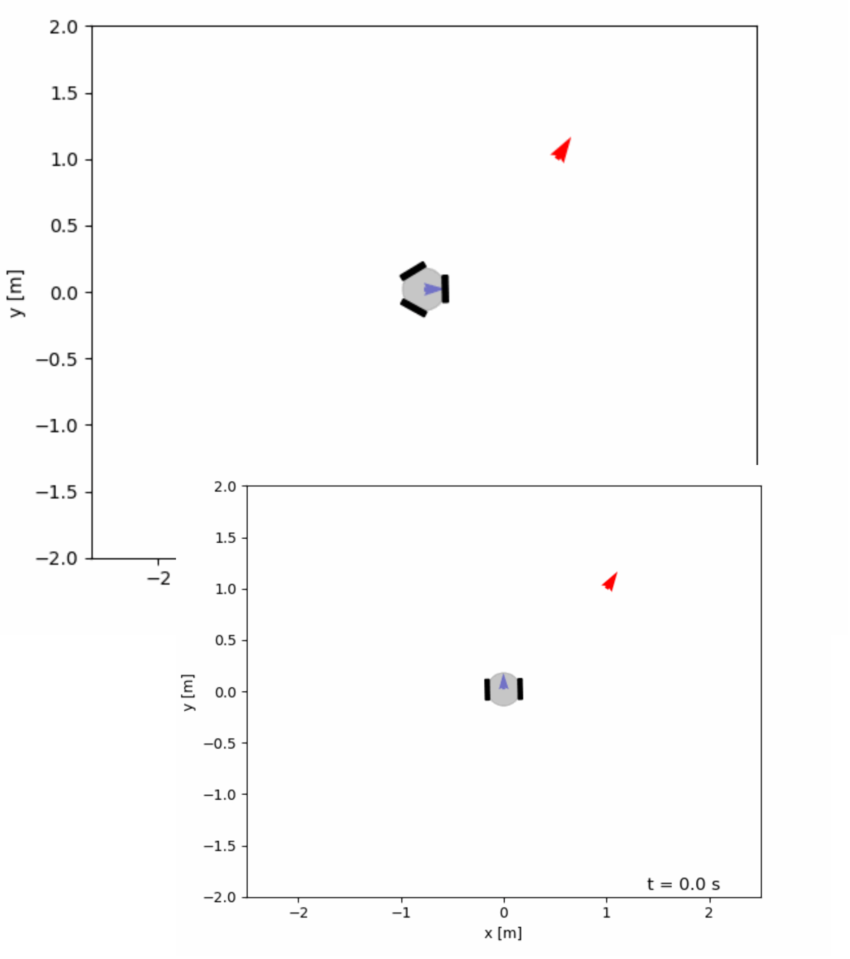
Bonus up to 20 point

Tools and Grouping

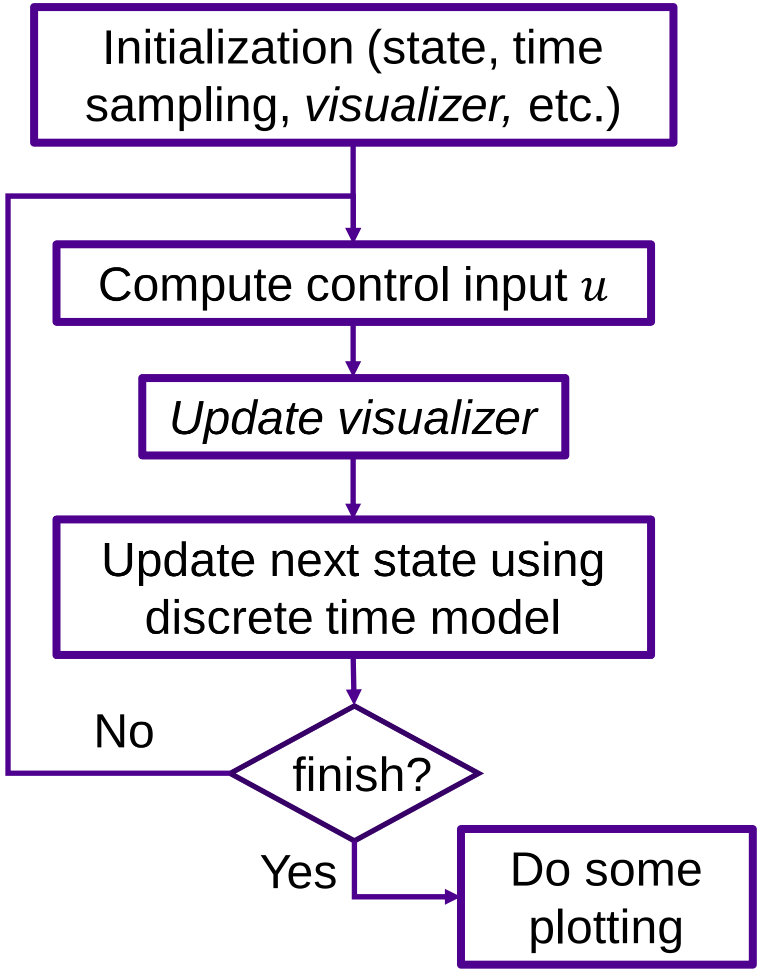
- Python: Matplotlib, Numpy, Cvxopt
  I provided base scripts for the exercise in Python
  https://github.com/TUNI-IINES/FunMoRo_control

  Not recommended, but you are free to use other language or software tools that you preferred (e.g., Matlab, C++), given that you prepare it yourself.

- Work in a group of 2 (same as before)



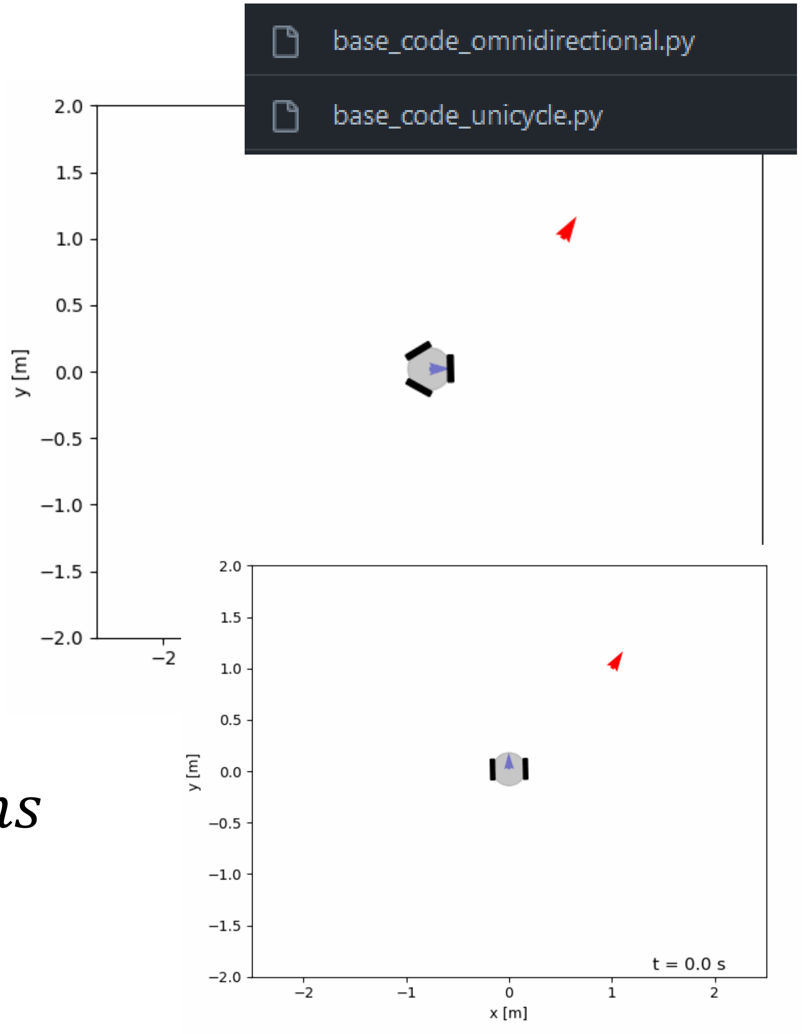
Flowchart of Simulator



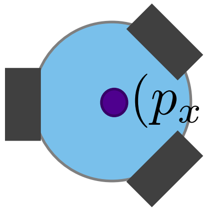
Parameter Setting
(for Exercise 4)

Time sampling $T = 10ms$

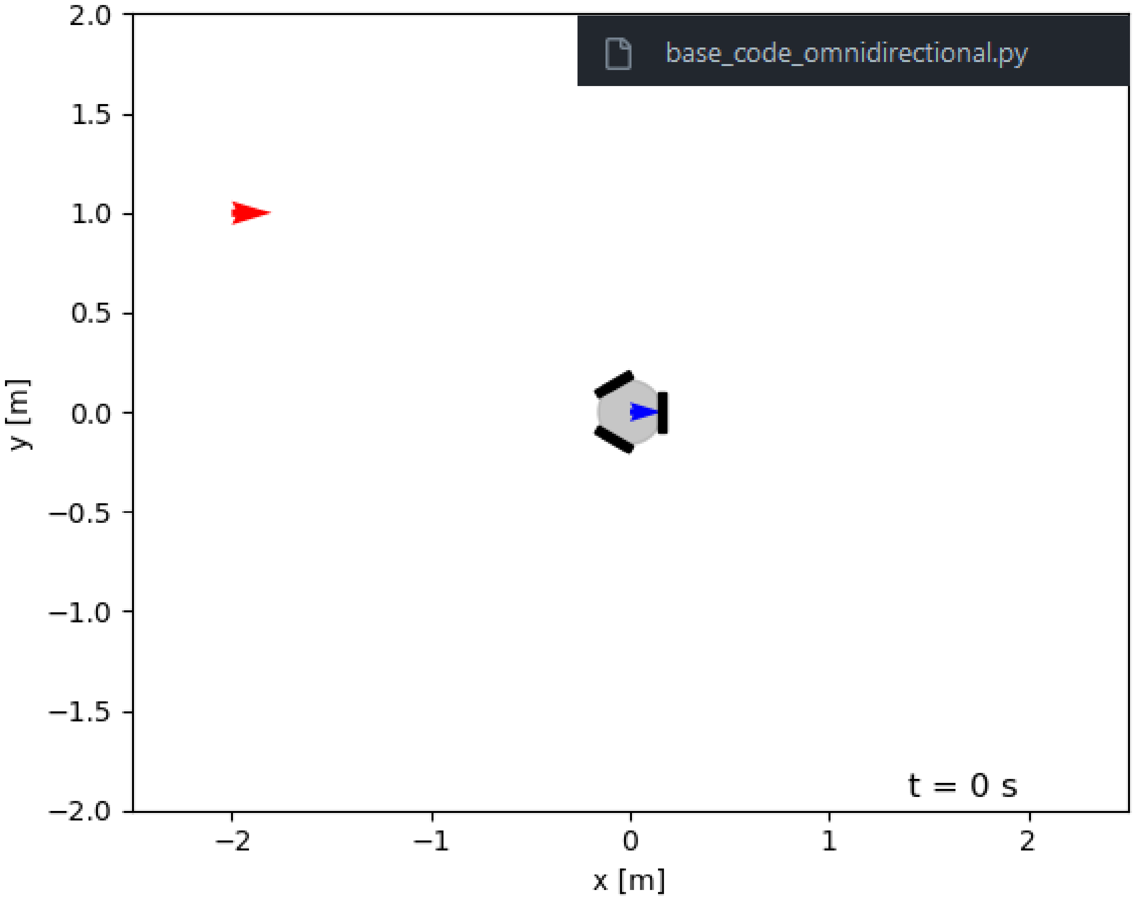
** the visualizer is optional*



Exercise 4.1 – Scenario



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

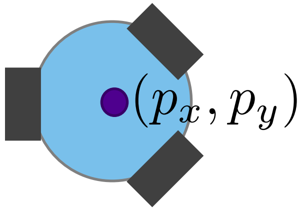


Model: omnidirectional mobile robot (**single-integrator model**)
Initial Position: $x[0] = [0 \quad 0 \quad 0]^T$
Goal: static at $x^d = [-2 \quad 1 \quad 0]^T$.

Control Objective:
- Reach the goal

Exercise 4.1

3 point



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

base_code_omnidirectional.py

With the objective to design control input u to reach the goal,

a. Implement **proportional control with static *k*** within 0~3.
Plot *time series* of v_x and p_x with 3 set of different k.

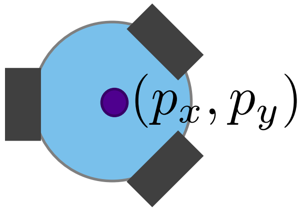
$$u = k(x^d - x), \ \ k > 0$$

b. Implement **proportional control with time-varying *k***
Plot *time series* of v_x and p_x with 3 pair of parameter v_0 and β .

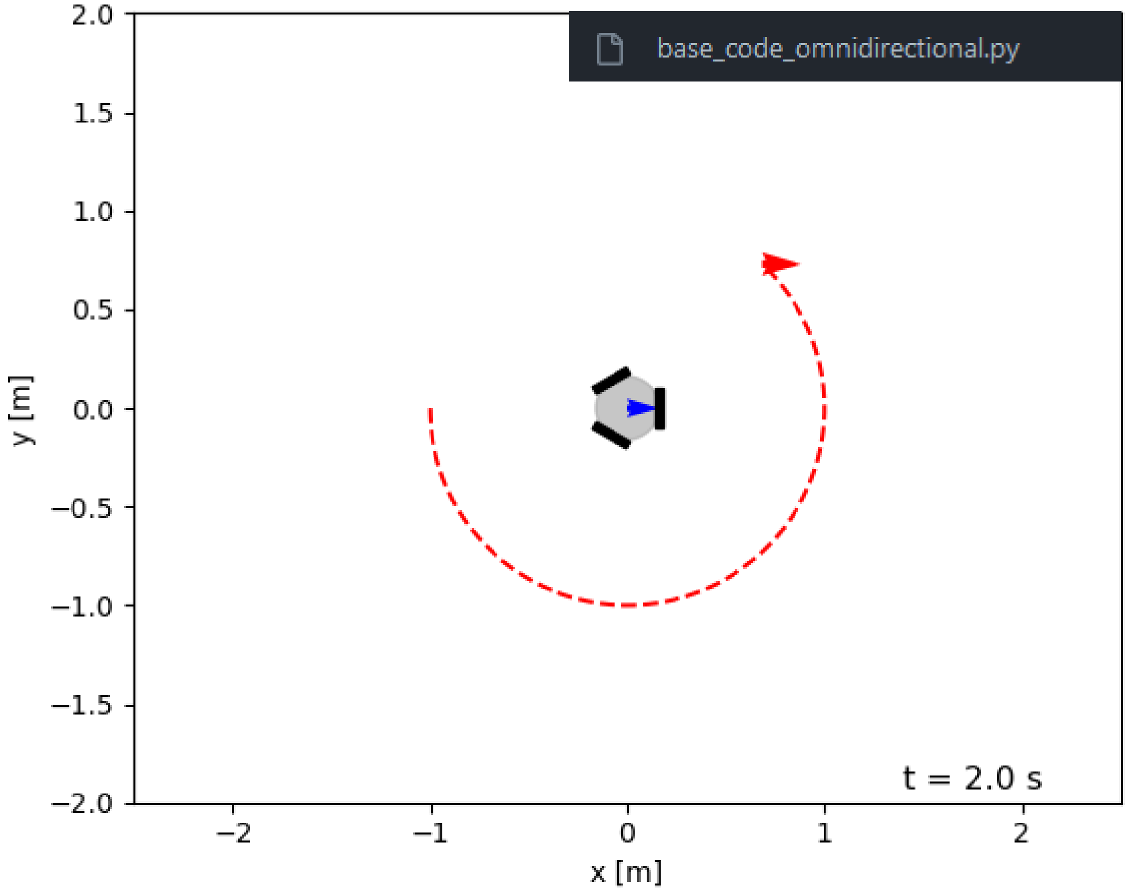
$$k = \frac{v_0 \left(1 - e^{-\beta \|\overline{e}\|}\right)}{\|\overline{e}\|}$$

c. Discuss how the variation of k, v_0 and β affects the control input and state trajectory.
What do you think is the appropriate value of k, or v_0 and β?

Exercise 4.2 – Scenario



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



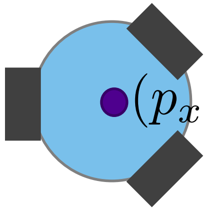
Model: omnidirectional mobile robot (**single-integrator model**)
Initial Position: $x[0] = [0 \quad 0 \quad 0]^T$
Goal: moving at

$$x^d[t] = [-\cos(2t) \quad -\sin(2t) \quad 0]^T$$

Control Objective:
- Track the moving goal

Exercise 4.2

2 point



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

base_code_omnidirectional.py

Track the moving goal
by designing **proportional control with feedforward term**

$$u = k(x^d(t) - x) + \dot{x}^d(t), \;\; k > 0$$

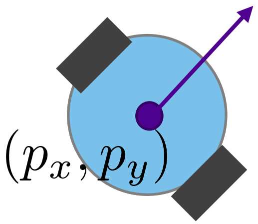
Describe your design process and
show the result by plotting:

- *time series* of control input u
- *time series* of error $(x^d - x)$,
- *time series* of state trajectory x vs x^d, and
- XY **trajectory** of the robot (or final snapshot of the simulator).

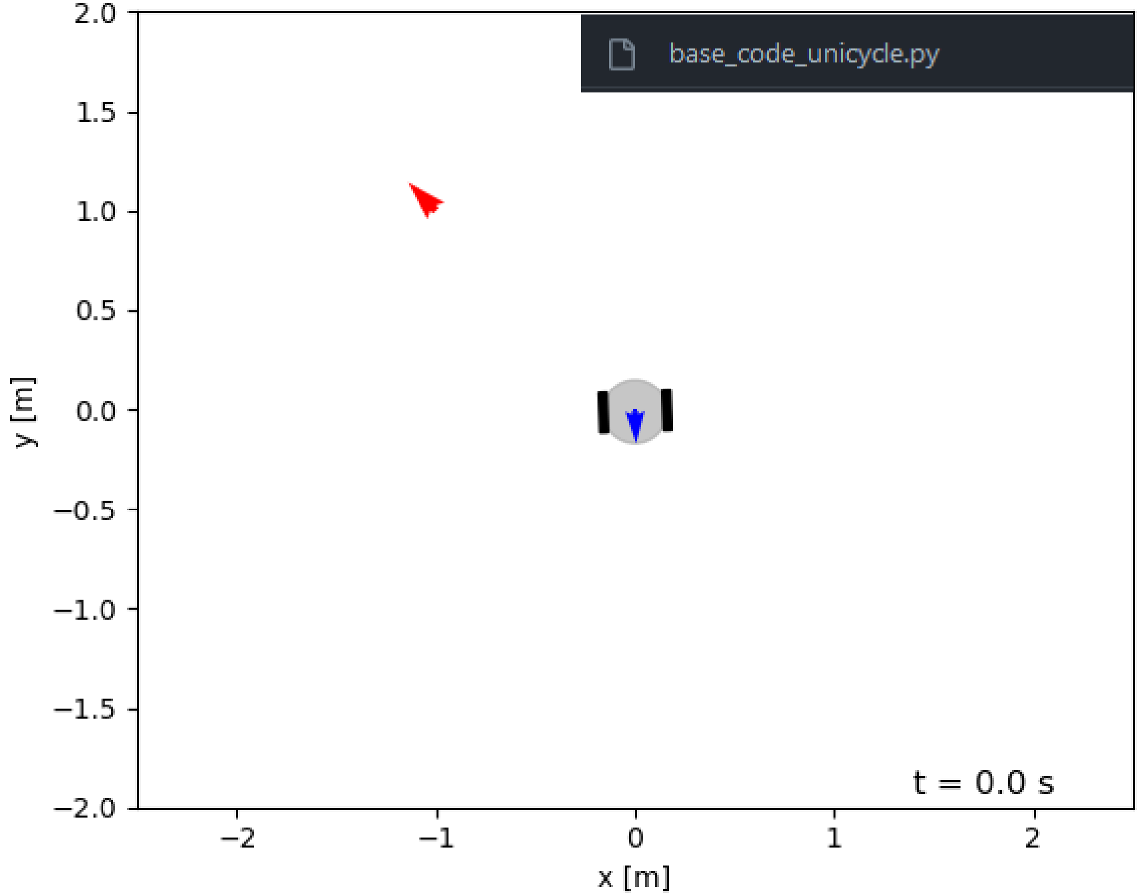
** Remember to modify the x^d in the simulator.*

Tampere University

Exercise 4.3 – 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: fixed position at $x^d = \begin{bmatrix} -1 & 1 & * \end{bmatrix}^T$

* can be any orientation at the goal position

Control Objective:

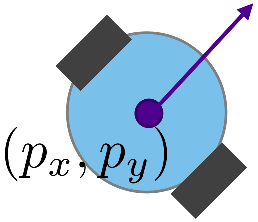
- Reach the goal (by designing ω)

with $v = \begin{cases} 0, & \text{if distance to goal} < 0.05m \\ 1, & \text{otherwise} \end{cases}$

Tampere University

Exercise 4.3

5 point



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

base_code_unicycle.py

a. Design a **proportional control for the orientation** to reach the goal position.
Describe your design approach and your observation.
Show the result by plotting:
- *time series* of control input u
- *time series* of error $(x^d - x)$,
- *time series* of state trajectory x vs x^d, and
- XY **trajectory** of the robot (or final snapshot of the simulator).

b. Find the minimum k in the proportional controller that ensure the robot can reach the goal.
Describe the problem with small gain k and analyze what affects the minimum k value.

Hint1: Compute desired angle θ^d towards goal position that constantly changes as the robot moves
Hint2: remember to ensure that $\bar{e}_\theta \in [-\pi, \pi]$

Question?

- Consult them via
  - Exercise sessions on 10.3.2023 and 17.3.2023
  - Teams channel