1 Hwk 2: The E-puck, Collective movements, Multi-Robot PSO

This homework requires the following equipment:

- Webots
- Matlab

Note that the homework is graded.
For any questions, please contact us at **dis-ta@groupes.epfl.ch**.

1.1 Information

In this assignment, you will find several exercises and questions.

- The notation $\mathbf{S_x}$ means that the question can be solved using only additional simulation.
- The notation $\mathbf{Q_x}$ means that the question can be answered theoretically, without any simulation; your answers should be submitted in your report. The length of answers should be approximately **two sentences** unless otherwise noted.
- The notation $\mathbf{I_x}$ means that the problem has to be solved by implementing some code and performing a simulation.
- The notation $\mathbf{B_x}$ means that the question is optional and provides bonus points. It should be answered if you have enough time at your disposal.

1.2 Outline

This homework is mainly dedicated to test your understanding of the 4th, 5th and 6th laboratory.

1.3 Grading

The report for this homework will be due on **Friday November 9 at 18:00**. **Late submissions will be penalized**, starting two hours after the deadline, we will take 5 points from your homework result per hour past since the deadline. Please format your solution as a **PDF file** with the name **[name]_hwk[#].pdf**, where **[name]** is your account user name and **[#]** is the number of the homework assignment, and upload it as the Report submission onto Moodle. If you have additional material (movies, code, and any other relevant files), upload a **[name]_hwk[#].zip** archive or a **[name]_hwk[#].tar.gz** archive instead with all files, including the report (hint: under linux, `tar cvzf [name]_hwk[#].tar.gz [directory or files]`). **Please use the same question notation in your answer file**!

Not respecting these guidelines can result in a penalty of up to 10 points on your final score!

The number of points for each graded exercise is given between parentheses. **The combined total number of points for the graded exercises is 100.**

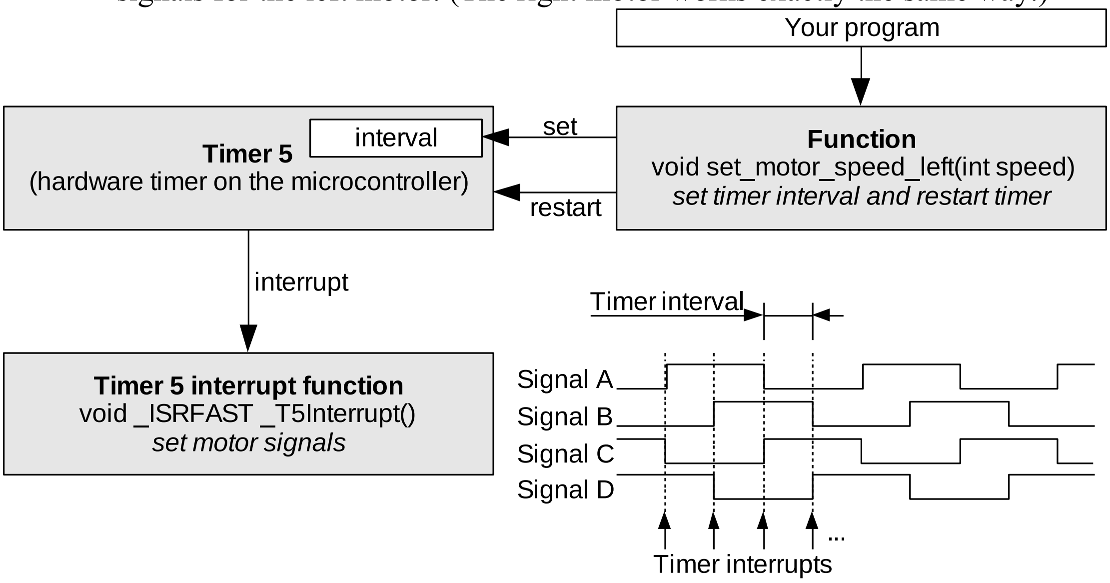
1.4 Before starting

To answer to the following questions, you will need to download **`hwk02.zip`** from Moodle. Uncompress `hwk02.zip`, containing the material needed to answer this homework. In addition, you will need to look at your solutions for the labs during this block.

2 Part 1: The e-puck (32)

Q_1(6): The basic obstacle avoidance algorithm used in lab 4 section 2.3 has an update rate of roughly 10 Hz and a mean speed of 3.5 cm/s. Assume you want to run this same algorithm on your computer in Switzerland to remotely control an e-puck in Singapore. The delay introduced by the network is 155 ms (one way). What mean speed would you have to choose such that the obstacle avoidance quality remains the same?

Q_2(5): The following sketch describes how the e-puck library generates the motor signals for the left motor. (The right motor works exactly the same way.)



Each time the motor signals change, the (stepper) motor moves one step. What happens if you call the set_motor_speed_left function extremely often?

Q_3(6): Assume that you want an e-puck to trace a circle, with the right wheel, clock-wise with a radius of 0.5 m using a dead-reckoning approach. Completing the tour should take 30 seconds. Assuming that there is no wheel slip and that the e-puck moves at a constant speed, what commands would you have to give the left and right wheel set-speed functions (constant speed)? The diameter of an e-puck wheel is 4.1 cm, the distance between the wheels is 5.3 cm and a wheel command of 1000 corresponds to 60 rpm (revolutions per minute) of the stepper motor.

I_1(15): However, as you noticed in the lab, wheel slip and other noise factors do greatly influence the result of dead-reckoning. Now, consider the task given in the previous question and assume that the wheel slip and other nondeterministic sources of noise are described by the error constants

$k_r = k_l = 0.03$. The e-puck starts at t_0 at $\vec{x}_0 = [x_0 \quad y_0 \quad \theta_0]^T = [0 \quad 0 \quad 0]^T$. The initial position uncertainty is:

$$P_0 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Update the position vector and position uncertainty every second, sampling interval Δt=1, until the circle is complete (in 30 seconds). Assume that the robot is moving at a constant speed. Use your left and right wheel speed result from Q_3 to calculate the distance travelled by the left and right wheel after each sampling, Δs_l and Δs_r respectively.

What is the position and orientation, $[x_{30}; y_{30}; \theta_{30}]$, at t_{30}? Use the formulas 5.2-5.5 in the book by Siegwart, Nourbakhsh *"Autonomous Mobile Robots"*. What is the covariance matrix P_{30} representing the position uncertainty at t_{30}? Use the error propagation law given in formula 5.9 to compute P_{30}, the Jacobian F for a robot like the e-puck is given in 5.11. Solve the problem with MATLAB and include the script in your homework submission.

3 Part 2: Collective Movements (38)

This section requires you to recall code that you wrote for Lab 5. It may be helpful to review your work during the completion of this section. For the implementation questions please use the code provided for the homework (part 2), not the code provided for Lab 5.

Q_4 (3): Recall the performance of your formation controller in **I_2** in lab 5. What did you observe? How does avoiding obstacles affect the performance of the controller?

Q_5 (3): What happened when you increased the speed of the leader in the relative formation? Why did this occur? Does this reflect what would occur on real robots, or is it just an aspect of the simulation environment?

Q_6 (4): Assume that you have an e-puck-like robot with the following assets (sensors and other):

- IR distance sensors
- Robot-to-robot radio communication
- Wheel encoders for robots speed
- Compass

Consider the Reynolds' rules for flocking:

a.) Separation: avoid collisions with nearby flockmates

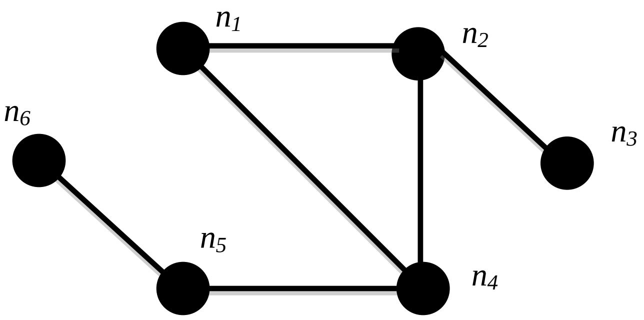
b.) Alignment: attempt to match velocity (speed and direction) with nearby flockmates

c.) Cohesion: attempt to stay close to nearby flockmates

Which of the robot's assets would you use to implement the Reynold's rules? For each of the three rules, describe (with one to three sentences) which assets you would use and how. Aside from the assets indicated above, you can also come up with additional ones.

I_2(10): In the lab you implemented two different methods for the followers to track the leader position: first, odometry and second, range and bearing. You will now implement a follower controller that fuses these strategies. Use the code you wrote for **I_6** and **I_8** in lab 5 and merge into a new follower controller that uses the fused information. Develop your code using the world file *flocking4.wbt*. To test your follower performance open *tester.wbt* and run it. In this world the leader moves randomly and the performance of your flocking algorithm is output into the console (1 is perfect and 0 is bad). Try to reach a performance of at least 0.7. State what performance you achieved and what your strategy was for implementing the algorithm. Include the follower code in your homework submission.

Q_7 (4): What is the Laplacian matrix of the graph shown below assuming a weight of 1 for each edge?



Q_8 (4): Assuming that each node of this graph has a state defined by its position in the 2D x,y-space (node n_1 has x_1, y_1; node n_2 has x_2, y_2; etc.), write the two differential matrix equations that will accomplish a rendezvous, with respect to the (x,y) position.

I_3 (10): Look at: http://en.wikipedia.org/wiki/Matrix_differential_equation and solve these 2 equations in Matlab (Hint: To complete the first two steps of solving the matrix ODE you can use Matlab's eig command to get the eigenvalues and eigenvectors of the Laplacian matrix). What happens when time tends to infinity? Can you derive the analytical values to which x_1, x_2, … and y_1, y_2,… tend? Include your Matlab script in your homework submission.

4 Part 3: Multi-Robot PSO (30)

In the first part of Lab 6, we tested shaping an obstacle avoidance behavior using multi-robot PSO. We considered 10 particles and 10 robots in a heterogeneous team learning approach. For the next implementation questions please use the code provided for the homework (part 3), load the world *pso_obs_1.wbt*, not the code provided for Lab 6.

I_4(15): Obstacle avoidance is learned on 10 e-pucks with an heterogeneous team. However, we want to be able to increase the number of particles in the population so that there are more particles than robots. The number of particles is changed with the variable SWARMSIZE in *pso.h*. Modify the supervisor code so as to be able to test any number of particles (at least any number that is a multiple of ten) with only 10 robots in a heterogeneous team learning approach. You only need to perform changes in *calc_fitness()*. Include the code for this function in your report.

Q_9 (5): Run the simulations for 10, 20 and 30 particles and compare the average performance you obtained. How does the number of particles affect the performance? Name at least one drawback that you observed in your simulations, of using many particles.

Q_{10} (5): Explain the differences between heterogeneous and homogenous team learning approaches.

Q_{11} (5): Explain the differences between private and public policies in solution sharing.

To complete this homework, create an archive **[name]_hwk[#].zip** or **[name]_hwk[#].tar.gz**. This archive should contain a PDF report with the name **[name]_hwk[#].pdf** containing your answers and all the files that came with the original archive and are now modified. You should then submit this archive on Moodle.