1 Lab 6: Multi-Level Modeling of Robotic Swarms

This laboratory requires the following equipment:

- C++ programming tools
- Webots
- Matlab

This exercise is a continuation of the work you did in Lab 5.

Information

In the following text you will find several exercises and questions. The type of question is indicated between parentheses:

- The notation **S** means that the question can be solved using only additional simulation.
- The notation **Q** means that the question can be answered theoretically, without any simulation.
- The notation **I** means that the problem has to be solved by implementing a piece of code and performing a simulation.
- The notation **B** means that the question is optional and should be answered if you have enough time at your disposal.

2 Modeling Robotic Swarms

Recall that in the previous lab, you developed a submicroscopic and microscopic model for a collision avoidance experiment. This week, you will develop a macroscopic model, followed by developing a multi-level model of a collision avoidance experiment.

2.1 Macroscopic Model

You have seen in the previous lab that it is possible to describe key properties of a swarm robotic system at a higher abstraction level. In this process, you might have noticed several aggregation processes: first, we calculated the *average* encountering rate from our simulation, as well as the *average* collision duration. In this part and at the highest modeling level, the metric we use to describe the robotic swarm is the *average* number of robots in each state.

We can now think of a mathematical formula describing the average number of robots in each state, given the probability that individual robots change their state.

1. **(Q):** Given the probability $p_R = p_r(N_0 - 1)$ that an individual robot encounters any other robot, and assume that you have $N_s(k)$ robots in search mode at time step k. How many robots will encounter a collision during the time step k?

2. **(Q):** What is the expression for the number of robots $N_a(k+1)$ in avoidance at time $k+1$? Formulate $N_a(k+1)$ in two different ways, as a difference equation and as function of N_0 and $N_s(k+1)$.

3. **(Q):** Assume a probability p_s for a robot to resume search, after being in collision. We might be interested in knowing how this probability translates into time dependent changes in the population of the robots in a certain state. Formulate now a difference equation of the form $N_s(k+1) = N_s(k) + \cdots$ that keeps track of the evolution of the population of the robots in the searching state.

4. **(Q):** Consider the difference equation for the search state $N_s(k+1) = N_s(k) + \cdots$ you derived earlier. Can you tell if the average number of robots in the search ever converge to a steady state? If so, calculate it. Will that always happen or only under certain conditions?

   Hint: Express the difference equation in the form $f(k+1) = af(k) + b$. This is called an affine dynamical system. Then, for the steady state, try to compute $N_s(k)$ at the limit $k \to \infty$.

3 A Multi-Level Model of a Collaborative Experiment

In this section we introduce a slightly more complex experiment, which uses behaviors developed earlier. The robots try to collaborate for the stick-pulling task.

The behavior of the robots can be described as follows: the robots wander in the arena until they detect a free stick. Pulling the stick upwards, a robot waits for a collaborator. If no other robot arrives to help during time T_{wait}, waiting is abandoned and the robot drops the stick back and continues searching. Otherwise, the two robots collaborate, and then resume searching. The number of sticks to pull in the arena is assumed to be constant, as the sticks are replaced in a random place after being handled. We assume that local communication among the robots is loss-less, i.e. that there are never two robots waiting at the same time holding on to a stick. The parameter T_{wait} is a crucial parameter for the collaboration rate, i.e. the ratio of successful collaborations over all time steps, which we want to define as the performance metric for this experiment.

5. **(Q):** What do you expect to happen if the parameter T_{wait} is too high? What happens if it is too low?

Let's start with a submicroscopic model in Webots. Unpack the archive containing the code as usual and open the webots world `sticks.wbt`. The sticks are simulated by the silver discs appearing in the arena.

Compile the supervisor and robot controllers. Open the file `sticks_config.h` in supervisor controller, where you can modify the T_{wait} parameter, maximum duration of the simulation and number of sticks.

After the simulation, the controller prints the number of sticks pulled, time taken and time spent by robots in wait and search states.

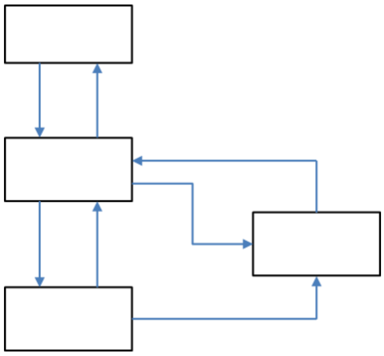
6. (I): Set the T_{wait} parameter to high (100s) and low (1s) values. What do you observe?

7. (I): Set the number of sticks to 2. Set T_{wait} to 2s and 30s. What do you observe for both values of the parameter?

8. (I): Set the number of sticks to 10. Now repeat the previous experiment by setting T_{wait} to 2s and 30s. Now what do you observe for both values of the parameter?

We will now try to build a probabilistic model based on the building blocks developed in the previous sections. For that, at the microscopic level, we describe the individual robot behavior as a PFSM, with transition probabilities and state durations we derive as described above (i.e. the probability to find a stick, and the time robot spend while avoiding each other).

9. (Q): Draw a PFSM using the framework given below. Label the states and transitions with meaningful names. One can simplify the PFSM by assuming that that the robots leave the stick and resume searching as soon as they find a collaborator. Draw a PFSM with two states that describes the simplified model.



A non-spatial microscopic model has been implemented in C++. Hereby, the program keeps track of the individual robots' and sticks' states and counts the number of successful collaborations, while state transitions are generated by random numbers reflecting the probabilities of the above model. In contrast to a submicroscopic model, specific intra-robot details (e.g., placement and cone of view of sensors, non-holonomicity of the vehicle), the trajectories and positions of robots and objects in the environment are ignored. Thus, it allows much faster simulation of different environmental or robotic parameters.

You find the microscopic model for the above experiment in the directory `microscopic/`. After compilation using `make`, you can run it by typing `./testr {robots} {twait}`, where the parameters allow you to specify the number of robots and desired waiting time.

Note: use mingw32-make on Windows.

10. (S): Run the program for different waiting times (also called gripping time) (between 0 and 10 seconds) and 4 robots (the maximum number of robots allowed is 10).

11. (Q): How does the waiting time relate to the collaboration rate (denoted *co*)?

In order to derive a macroscopic model for this experiment we can leverage the structure of the simplified PFSM developed for the microscopic model. We can also start with a PFSM that describes the swarm behavior. Such PFSM has the same structure as the one that models a robot's behavior at the microscopic level, however the states and the arrows linking them have a different interpretation. Here we can think about each state variable as a real number describing the average number of robots in that specific state (e.g. $N_s(k)$, and $N_w(k)$ as the average number of robots searching, and waiting alone, at time step k respectively). We will have $N_0=N_s(k)+N_w(k)$ the total number of robots, and M_0 the total number of sticks. The arrows linking the states will then represent the flow of the robots changing states. Assume that right after entrance of the second robot, both robots transition to a search state.

12. (Q): Draw a PFSM that describes the overall population dynamics, i.e. at the macroscopic level.

13. (Q): Assume p_{stick} is the probability of finding one stick out of the total number of sticks scattered in the arena, p_{collab} is the probability that a robot in search state finds another robot in waiting state, and that there are m sticks in the arena. In order to develop a model describing the dynamics of robots' states, first derive the expressions $\Delta_w(k)$ for the average number of robots that enter the wait state, $N_w(k)$, and then derive an expression $\Delta_{collab}(k)$ for the average number of robots that *leave* a stick after collaboration.

This allows us to formulate the following difference equation that describes the dynamics of the search state population:

$$N_s(k+1) = N_s(k) - \Delta_w(k) + \Delta_{collab}(k) + \Delta_w(k - T_{wait})\Gamma(k - T_{wait}; k)$$

14. (Q): What does Γ represent? *Hint: "Γ represents the fraction of robots that…"*

15. (Q): Formulate a similar equation for the waiting state.

16. (Q): In general, is the macroscopic model linear or non-linear? Why?

17. (Q): Show that the average time spent in a state can be calculated by taking the inverse of the (constant) probability to leave this state. You can do this by summing all possible waiting times multiplied by the probability that they occur.

Hint: $\sum_{k=1}^{\infty} kx^k = \frac{x}{(1-x)^2}, 0 \leq x \leq 1$

18. (Q): Assume that half of the robots are incapable of initiating a collaboration. i.e, they have a waiting time of zero ($T_{wait} = 0$). For the other half of the robots, would there be a change in the relation between collaboration rate and the waiting time? Why?

19. (I): Modify the code in the folder `microscopic/` to incorporate this behavior of half of the robots having $T_{wait} = 0$. Experiment with various values of waiting times and observe the relation with the collaboration rate.

Hint: Look for the Boolean variable `change_this_robot` *in test.c.*

4 References

[IJRR2004] **Modeling Swarm Robotic Systems: A Case Study in Collaborative Distributed Manipulation** / Martinoli, A.; Easton, K.; Agassounon, W. / *Int. Journal of Robotics Research*, Num. 4, Vol. 23 (2004), pages 415-436