Automatic Design of Controllers for Khepera IV Robots: A Comparison Between Finite State Machine and Neural Network based Architectures for Flocking

Ivan-Daniel Sievering

Professor : Alcherio Martinoli
Assistant(s) : Cyrill Baumann

With the dissemination of the robots, the need to make collaborate emerges. One of the collective behaviour that a group may exhibit is flocking. In 1987, Craig Reynolds proposed a controller to handle this exact problem: each member of the group is led by three different rules that are active simultaneously but with different weights. They are repulsion, attraction and alignment. In this report, two alternative controllers that use Reynold's rules are proposed and compared.

The first controller sequentially uses the rules through a Finite State Machine (FSM). Each state corresponds to one rule being on control. Transitions between them depend on some measures of the world. The second controller uses the rules in an additive way: an Artificial Neural Network (ANN) converts the worlds measures to a set of activation levels. The robot's displacement then is computed by summing the vector generated by each law multiplied by its corresponding activation level.

The two controllers are implemented on eight Khepera IV robots in the Webots simulator. They are free to displace inside of a 30x30m closed arena without obstacles. The controllers do not receive the value of the robots' sensors directly; instead, they receive enhanced interpretation of the world: the distance to the centre of flock, the distance to the nearest robot, the distance to the nearest obstacle and the speed difference between the robot and the flock.

For each controller, a manual solution is proposed. Additionally, Mixed Discrete Particle Swarm Optimisation (MDPSO) is used to find machine-learned (ML) solutions. In order to evaluate the quality of the flocking behaviour, the alignment, the cohesion and the separation are measured at each timestep and multiplied. This value is called instantaneous fitness. The sum of all the instantaneous fitness of a simulation gives the global fitness, which is the reward given to MDPSO. A velocity term should have also been included but has been unfortunately forgotten.

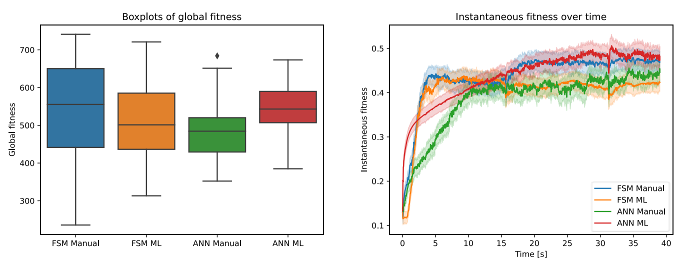

Figure 1: Performance of the controllers

The quantitative comparison between the different controllers is presented in figure Figure 1. Notice the differences between sequential and additive controllers: ANN controllers have less variance on the global performance than FSM but reach their instantaneous fitness plateau later. MDPSO managed to find a better solution than the manual solution for the ANN but not for the FSM.

The qualitative inspection of the solutions is done by running simulations on Webots (see videos). Doing so, the ML FSM solution has to be discarded because it has not implemented obstacle avoidance. The two manual solutions for their part provide good implementations. The ANN ML solution yields satisfactory results: the robots regroup (slowly) and then move together while keeping a safe distance. Whenever one of them is isolated, it is directly attracted toward the centre of the flock.

Thus, the ANN seems to be better suited than the FSM for flocking behaviour's automatic design. However, FSM is more comfortably designing by a human, can generate faster displacement and regroups the robots faster.

In future works, the ANN could be optimised with Vanilla Policy Gradient instead of MDPSO. More, obstacles may be added to the world to learn avoidance correctly in order to implement the controller to real robots later.