Towards a Generalized Distributed Robotics Simulator: Simulating Emergent Flocking on a Generalized Map

Aditya Kedia

Introduction

The field of distributed robotics is concerned with the study of the dynamics and control of systems of agents working together to reason, plan, solve problems, think abstractly, comprehend ideas and language, and learn. Engineers working in the field are concerned ultimately with designing systems of software agents, robots, sensors or even humans that can work with the same level of efficiency as human teams. A wonderful overview of the field is provided by Lynne E. Parker [1]. She suggests a framework for categorization of types of interactions in systems of distributed intelligence (Fig 1).

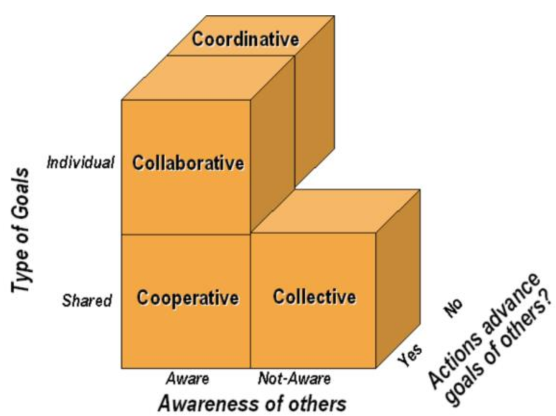

Fig. 1: **Categorization of types of interactions in systems of distributed intelligence [Parker, 2008]**

Inspiration for the field has often been drawn from biology, such as flocks of birds or swarms of bees and even colonies of ants and other insects. Algorithms and strategies in distributed robotics systems are often tested first, and sometimes only, in simulation. The aim of this project is to lower the barrier to entry for researchers in this field by providing a generalized GUI simulator. Such a simulator will allow a user to describe a world, the number of intelligent agents in this world and the parameters, characteristics and behaviors assigned to these agents. The simulator will then provide an animated graphical rendering of how the scenario evolves as time passes, and will allow a user to track relevant metrics related to either the distributed system or individual agents. Such a simulator will require minimal to no programming on the part of its users.

Problem Statement

As a first step towards developing such a generalized simulator, this project aims to simulate flocking/swarming as an emergent behavior in agents. In Parker's classification, swarming/flocking is a 'collective' problem. Several models of flocking have been proposed over the years [2] [3] [4]. However, in the interest of generality, this project explores Matarić's model [5] of flocking as an emergent behavior. According to Matarić, complex behaviors such as flocking, in a system of distributed agents can emerge as a property of the system due to the combination of simple behavior primitives exhibited by each agent. Some of the behavior primitives described by her include:

- **safe wandering-** minimizes collisions between agents and environment

- **following-** minimizes interference by structuring movement of any two agents
- **aggregation-** gathers the agents
- **dispersion-** minimizes the collisions between agents
- **homing-** enables the agent to proceed to a particular location

In this framework, flocking can be understood as a combination of aggregation, dispersion and homing. The weights associated with each behavior can then be tuned in order to get the desired behavior from the collection

The presented simulator simulates holonomic circular robots of a given radius with a 360-degree field of view sensors of a given range that form a flock and move together through a marked course to a goal location with minimal collisions between each other. The course simulated here is presented in figure 2. In future versions of the simulator, the course will be defined by the user as a set of points specified in the counterclockwise order. An example scenario is shown in figure 2.

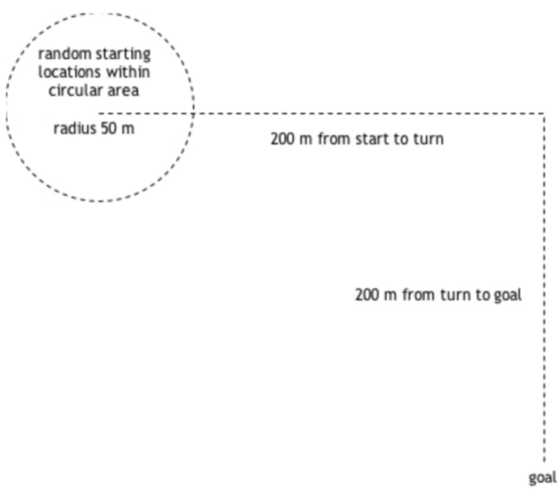

Fig. 2: **A Possible scenario for swarming simulation. The dotted line is the path to be followed**

While the homing and aggregation behaviors specified above are straightforward to understand and implement, dispersion or collision avoidance is its own area of research with a rich and growing body of literature. The current version of the simulator presents a simplifies collision avoacne apprach where an agent is repulsed by other agetns at a certain distance. The other approach to be provided as a choice in future versions of the simulator will be to use the reciprocal n-body collision avoidance proposed by Berg et al [6] where, under the assumption that the robots are all using the same approach for picking velocities that allow collision avoidance, each agent can treat the other as a velocity obstacle, and the problem of picking a collision-free velocity can be reduced to solving a low-dimensional linear program.

Note that the approaches and models discussed here all assume that the agents are independent, and make their own decisions regarding choosing their velocities at every time step. Thus, at any time T, the position new position U of a particular robot is given by

$$U = U' + dt * V$$

Here *dt* is the size of the time step, and V is the velocity of the robot derived as follows:

$$V = W_A * V_A + W_D * V_D + W_H * V_H$$

Where Vs and Ws are velocities due to a particular behavior, and weights assigned to those behaviors respectively. The subscript A refers to aggregation, D to dispersion and H to homing.

Computational Approach and Description

The simulator is implemented as a Windows Form app in C#. Each agent is implemented as an object of the agent class, which contains the properties of the agent, as described above, and the behavior primitives as functions. At every time step, the program calculates the distance of each agent from the others, and report to it the position of its peers that are within its prespecified sensor range. The form allows the user to choose the primitives that should be active during the simulation, and choose the weights for each of these behaviors. The program allows the user to make several other choices as can be seen in a snapshot of the app (figure 3).

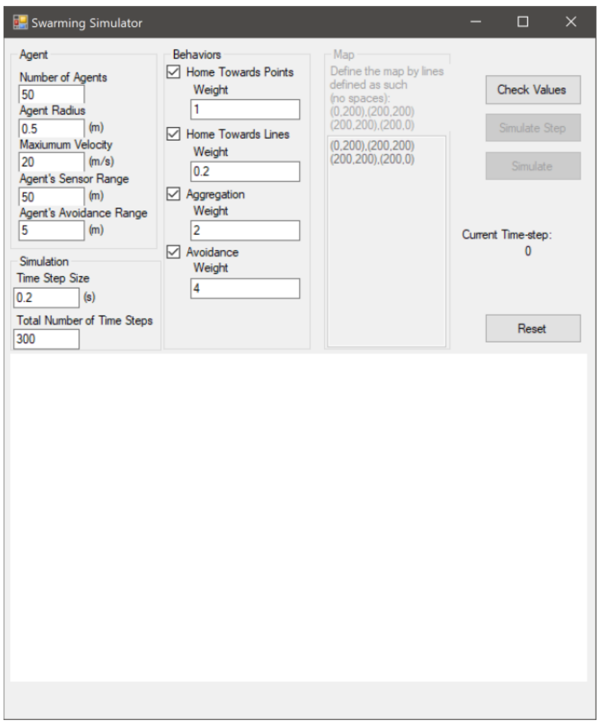

Fig. 3: **Initial state of the form with choices that the user must make**

During the entire simulation, the program tracks the centroid of the system of agents. The distance of the centroid to the lines on the map and the average distance of the agents from the centroid, at each time step, are recorded as the metric of interest and saved to a CSV file at the end of the simulation. This allows one to study the behavior of the flock as the number of agents changes, or as the weights of the different behaviors are varied.

A case study of the sample simulation is presented bleow.

Case Study

The simulator starts with a sample simulation of 50 agetns of radius 0.2 m with a sensor range of 50 m and a maximum velocity of 20 m/s. They execute homing, aggreagrion and avoidance behaviors hence resulting in swarmin/flocking. The map followed by the agents is shown in figure 2. The simulation is run for 300 time steps of 0.2 seconds each.

Figues 5 shows the distance of the centroid of the system of agents from the path of interest (the line that the robots follow to their target). Figure 4 shows the average distance of each agent from the centroid throughout the course.

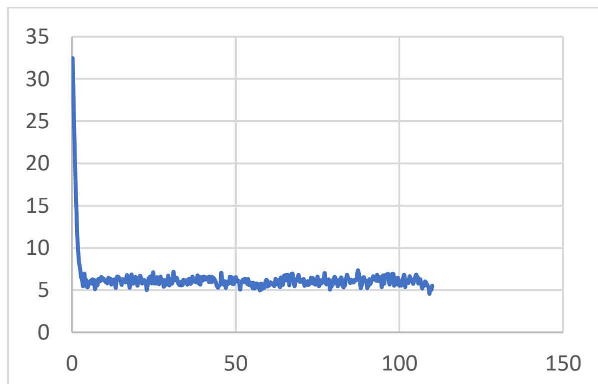

Fig. 4: **Average distance of each agent from the centroid vs Time**

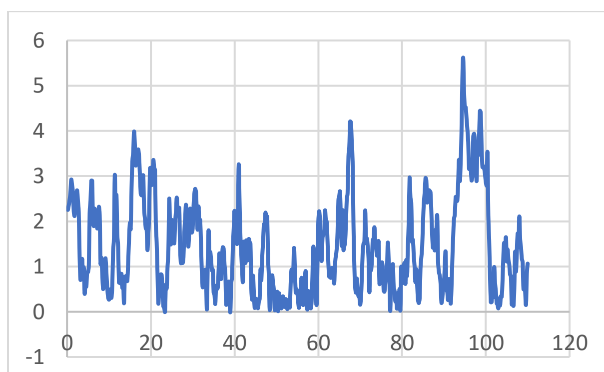

Fig. 5: **Average distnace of centroid from the line of interest vs Time**

The behavior can then be studied by chaning parametersand making several plots. The following plot for isntaance shows the distance of the centroid from the line for 25, 50, 75, and 100 agents.

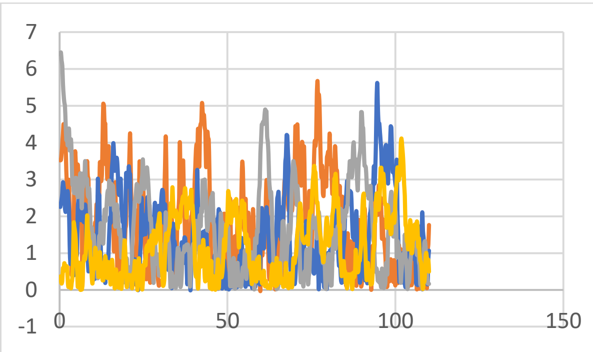

Fig. 6: **Distnace of centroid from the line of interest vs Time (Orange – 25, Blue – 50, Grey – 75, Yellow – 100)**

We can notice here how increasing the number of agents reduces the distance of the centroid to the path being followed. An ideal swarm would have a vanishingly small distance from its path.

Conclusion and Future Work

This report presents a generalized visual simulator to simulate the flocking of a collection of intelligent agents. It allows the user to choose several primitive behaviors, and the weights associated with them. The simulator then tracks several metrics of interest, which will allow the user to study the behavior of the system and the agents. Some of the improvemtns in future versions of the simulator are descried below:

- The simulator will allow the user to define a generalized map.
- the visualizer and the simulator will be generalized further by adding several other primitive behaviors, and even some primitive tasks as described by Balch et al [7]. They describe how several complex behaviors may be specified as combinations of basic tasks such as foraging, grazing and consuming.
- The nature and properties of the agents will also be generalized in the future version of this simulator such that the user can define the heterogenous agents and assign shapes and sizes to all.
- The agents will be prallelized so as to allow faster simulation.

References

[1] L. E. Parker, "Distributed Intelligence:Overview of the Field and its Application in Multi-Robot Systems," *Journal of Physical Agents, special issue on multi-robot systems, vol. 2, no. 2,* pp. 5-14, 2008.

[2] C. R. Kube, "Collective Robotics: From Social Insects to Robots," *Adaptive Behavior,* vol. 2, no. 2, 1993.

[3] C. W. Reynolds, "Flocks, herds and schools: A distributed behavioral model," in *Proceedings of the 14th annual conference on Computer graphics and interactive techniques*, New York, 1987.

[4] J. McLurkin, "Distributed Algorithms for Dispersion in Indoor Environments Using a Swarm of Autonomous Mobile Robots," in *Distributed Autonomous Robotic Systems 6*, Tokyo, Springer, 2007.

[5] M. J. Mataric, "Designging Emergent Behaviors: From Local Interactions to Collective Intelligence," *Conference on Simulation of Adaptive Behavior,* 1993.

[6] J. v. d. B. J. G. L. Manocha, "Reciprocal n-Body Collision Avoidance," in *Robotics*

Research. Springer Tracts in Advanced Robotics, Berlin, Heidelberg, Springer, 2011.

[7] T. Balch, "Communication in Reactive Multiagent Robotic Systems," *Autonomous Robots,* vol. 1, no. 1, pp. 27-52, 1994.