Multi-agent coordination combining flocking heuristics with centroid estimation and tracking

Christina Piberger

Universidad Politécnica de Madrid, Spain, c.piberger@alumnos.upm.es

Abstract. Recently, self-organized distributed computing has developed great appeal in the field of controlling multi-agent systems. Simple interaction rules lead to the emergence of collective behavior and give access to a huge spectrum of tasks such a *swarm* can perform. This work presents a new approach to decentralized multi-agent coordination by using Reynold's flocking heuristics and enhancing them with a velocity-based guidance rule derived from centroid tracking. The proposed algorithm was implement and simulated with the MASON framework.

Keywords: self-organizing systems, decentralized control, swarm, centroid estimation, MASON

1 Introduction

The coordination task in a multi-agent system which is restricted to decentralized control, describes a scenario in which communication is limited to individual perception (sensor measurements) of agents [1]. Biological systems, where simple agent-agent-interaction, leads to emergent and self-organized swarming behavior inspired interesting methods to approach that task. In [2] emergence is explained as a result of the interplay between collective *global behavior* and simple *local rules*. Main advantages of such a decentralized controlled system are robustness and flexibility, which address the fact that in such a *swarm*, each single entity is replaceable, not essential for the general functionality and therefor losses of agents are handled 'gracefully'. Among other reasons, this makes multi-agent systems highly attractive for many real-world applications. Underlined by the well-known conceptual construct by Aristotle: "The whole is greater than the sum of its parts."

This work proposes a new approach on multi-agent coordination, by combining Reynold's flocking heuristics [3] with a velocity-based guidance rule. More specifically, velocity changes are exploited as an implicit measure of an agent's excitement in order to navigate the swarm in a certain direction.

Reynold's formulated three simple rules for flocking, namely, separation, cohesion and alignment (the latest is also referred to as *velocity matching*). These rules for steering behavior allow agents to stick together, avoid collisions and perform an overall genuine movement.

In the following two sections the setting of the problem will be discussed, a new algorithm will be proposed, and details on how the simulation using the

MASON framework [4] was carried out will be given. The last section summarizes the work and will give some further perspective about future ideas.

2 Problem description

Some issues regarding the implementation of Reynold's rules are exhaustively discussed in [5]. *Fragmentation* (a division of the swarm) and *collapse* (all agents drift towards the center) are mentioned as classical pitfalls of *free-flocking*, which describes flocking with just the three rules of Reynold. It is demonstrated that the first two rules (avoidance and cohesion) can be modeled together as one potential field and hence a new flocking protocol is introduced—which mainly inspired this work.

The third rule (alignment) is essential for a collective group movement. However, instead of just adapting to its surrounding, each agent should have the possibility to actively change the direction of this movement. This can be realized by using a 'estimation and tracking' of the group's centroid, for it encodes information about group direction as well as velocity.

The work of [6] provides further insights to asynchronous decentralized centroid estimation and states that the calculation can be performed safely without any common reference frame.

In this work both ideas are combined into one overall algorithm which is introduced in the following section and was successfully verified by simulating a 'collection task' that will described afterwards.

3 The algorithm

One key aspect of the proposed algorithm is that velocity changes are used as an indirect measure of *excitement*. Unlike standard flocking—where the agents are adjusting their speed in order to achieve a smooth movement—the change of velocity is deliberately used to promote changes in the flock. This follows the intuitive idea of accelerating when seeing something interesting.

Agents do not communicate their excitement explicitly, but nearby agents will react to it as a consequence of recalculating the centroid. This results from the idea that a faster agent has more impact on the shift of the centroid than a slower (unexcited) agent and therefor drags the centroid with it. Surrounding agents will then adapt to that movement by setting their 'preference direction' accordingly.

The 'preference direction' can be also influenced by other aspects or measurements. An example is provided in the later simulation of a 'collection task', where individual agents will respond to objects in-sight of their vision sensor by setting the 'preference direction'.

The basic procedure can be seen in Algorithm 1 (note that the repulsive force is computed pairwise between an agent and its neighbors, while the attractive force only depends on the group centroid).

Algorithm 1

for all agents **do**
  repeat
    neighbors ← sensor measurement
    $F_{rep} = c_1 * \left(\frac{1}{r} - \frac{1}{r_0}\right)^2$ where r = distance to neighbor, and r_0 = sensor range
    $F_{attr} = c_2 *$ distance to centroid
    $F = F_{attr} + F_{rep}$
    // update 'excitement' and 'preference' according to vision sensor
    // update 'preference' to the centroid movement (mean-shift vector)
    // update position according to state (see Fig. 2)
  until agent leaves 'exploration' state

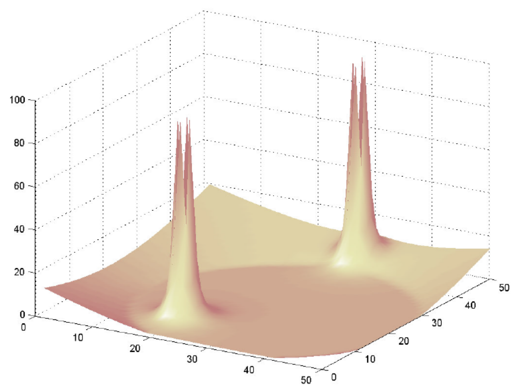

Fig. 1. Visualization of the used potential field function. It shows the presents two agents and the 'comfort zone' surrounding the centroid

3.1 Flocking heuristics as potential field

Article [7] provides an overview over possible approaches to properly model control schemes in a multi-agent system and the one that is used in this work is a *potential field-based* approach. This is also suggested in [5] where the separation and cohesion rule of flocking are formulated together in one smooth potential field.

An essential difference of the proposed Algorithm 1 is that only the repulsive force is calculated pairwise, while the attraction force is pointing to the group centroid. Moreover, the attraction (or cohesion) force is proportional to an agent's distance to the centroid, and in addition it is suppressed by a certain threshold. That generates a region without influence which is referred to as 'comfort zone'.

The two forces are then combined to form the overall potential field, which will be re-calculated by each agent in every time step, individually. An illustration of how an overall potential field may looks like is given in Fig. 1.

Note that such a gradient-based method typically suffers from problems with *local minima*, which will not be discussed further in the scope of this work. More sophisticated approaches to find appropriate potential field functions especially regarding dynamic motion of obstacles can be found in [8].

3.2 Centroid tracking

It has been shown that the centroid is used to compute the attractive force which represents Reynolds cohesion rule. Now the centroid is used for an additional purpose: The estimation of the overall group movement.

This extension is necessary, because the algorithm, as discussed so far, does not provide any interaction rule that compensates for the neglected alignment (or velocity matching) rule.

The idea was originally inspired by mean-shift tracking [9], which is already very popular in the computer vision community, however, since it is designed for arbitrary feature spaces, it could be applied to this setting without any problem.

The calculation of the so called *mean-shift vector* allows the agents to both, support the group movement, as well as change the direction. Especially in combination with increased velocity, this enables the agent to have a strong influence on the group behavior.

4 Simulation

To simulate the algorithm the 'Multiagent Simulation Toolkit' (MASON) [4] has been used. It internally works with representation of neighborhoods in so called *proximity nets* which are also described in [1] and used in the algorithm of [5].

The simulation supports multiple groups of swarms called 'alliances'. Even though the only way of communication between agents is to perceive each others appearance, it is assumed that an agent can distinguish between group members and also dead agents.

Furthermore, the simulation allows the robots to perform a 'collection task', where objects (in form of collectible items and obstacles) influence an agent's preference direction additionally to the mean-shift tracking.

4.1 Agent model

In the described setting of self-organized multi-agent coordination based on simple interaction rules, it can be stated that—according to Russel and Norvig's proposed types of agents [10]—a *simple reflex agent* model is used. Meaning, the agent has no information about the state of the environment or other agents, neither can it use searching-strategies to archive a goal or evaluate an utility-functions—as typically used in decision theory.

For describing the agents dynamics a *self-propelled particle* model is used, where all agents move with an equal translational velocity [7], however, as already

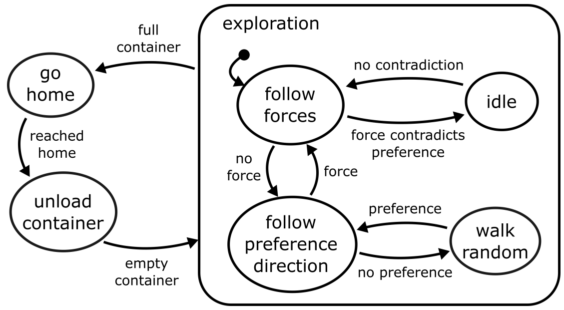

Fig. 2. State chart representing the FSM that each agents uses.

mentioned, the change of velocity of single agents is used to control the swarm behavior—therefor the agents are all homogeneous.

Each agent is equipped with a range sensor and a vision sensor, which are used to detect other agents and inspect the ground, respectively. This way the limited perception / interaction range is modeled.

Furthermore, the agents incorporate an internal 'finite state machine' (FSM) which can be seen in Fig. 2. When properly designed FSMs allow a straight forward implementation and are easy to interpret, as shown in the work of [11].

The concept of *idling* (or *waiting*) in combination with a FSM is also used in [12]. In other strongly related domains of multi-agent systems, like 'dynamic task allocation', it was shown that idling instead of making a too early decision can be beneficial for the overall system performance [13]. The idea to adapt thresholds based on idle-time, could also be interesting for the agent model used here.

5 Discussion and future work

This work proposed a new algorithm for leaderless, decentralized control with asynchronous communication using flocking heuristics and centroid tracking. Its practicability has been proven by simulation, and it can be stated that the concept of exploiting velocity changes as control mechanism is feasible.

Moreover, the centroid tracking idea might have an even higher potential. It encodes the average speed of the swarm, so for example, could a low group-velocity encourage agents to move even further away of the centroid.

Possible extensions to this work are using formation or mapping techniques to keep track of explored areas. However, such extensions would require an advanced agent model. Also the use of a leader or the implementation of a hierarchy to improve such a system should be investigated further. Unfortunately, sensor noises or communication errors are not taken into account in the simulation. So it is not a properly simulation of a real-world scenario yet.

References

1. Sonia Martinez, Jorge Cortes, and Francesco Bullo. Motion coordination with distributed information. *Control Systems, IEEE*, 27(4):75–88, 2007.
2. Tom De Wolf and Tom Holvoet. Emergence versus self-organisation: Different concepts but promising when combined. In *Engineering self-organising systems*, pages 1–15. Springer, 2005.
3. Craig W Reynolds. Flocks, herds and schools: A distributed behavioral model. In *ACM Siggraph Computer Graphics*, volume 21, pages 25–34. ACM, 1987.
4. Sean Luke, Gabriel Catalin Balan, and Liviu Panait. Mason: A java multi-agent simulation library. In *Agent 2003 Conference on Challenges in Social Simulation*, 2003.
5. Reza Olfati-Saber. Flocking for multi-agent dynamic systems: Algorithms and theory. *Automatic Control, IEEE Transactions on*, 51(3):401–420, 2006.
6. Mauro Franceschelli and Andrea Gasparri. Decentralized centroid estimation for multi-agent systems in absence of any common reference frame. In *American Control Conference, 2009. ACC'09.*, pages 512–517. IEEE, 2009.
7. Veysel Gazi and Baris Fidan. Coordination and control of multi-agent dynamic systems: Models and approaches. *Lecture notes in computer science*, 4433:71–102, 2007.
8. Shuzhi S. Ge and Yun J Cui. Dynamic motion planning for mobile robots using potential field method. *Autonomous Robots*, 13(3):207–222, 2002.
9. Dorin Comaniciu, Visvanathan Ramesh, and Peter Meer. Real-time tracking of non-rigid objects using mean shift. In *Computer Vision and Pattern Recognition, 2000. Proceedings. IEEE Conference on*, volume 2, pages 142–149. IEEE, 2000.
10. Stuart Russell and Peter Norvig. Artificial intelligence: a modern approach. 1995.
11. C. L. Garzon, H. R. Chamorro, M. M. Diaz, E. Sequeira, and L. Leottau. Swarm ant algorithm incorporation for navigation of resource collecting robots. In *Biomedical Robotics and Biomechatronics (2014 5th IEEE RAS & EMBS International Conference on*, pages 987–992. IEEE, 2014.
12. Onur Soysal et al. Probabilistic aggregation strategies in swarm robotic systems. In *Swarm Intelligence Symposium, 2005. SIS 2005. Proceedings 2005 IEEE*, pages 325–332. IEEE, 2005.
13. Vincent A Cicirello and Stephen F Smith. Wasp-like agents for distributed factory coordination. *Autonomous Agents and Multi-agent systems*, 8(3):237–266, 2004.