We are IntechOpen,
the world's leading publisher of
Open Access books
Built by scientists, for scientists

6,900
Open access books available

185,000
International authors and editors

200M
Downloads

154
Countries delivered to

Our authors are among the

TOP 1%
most cited scientists

12.2%
Contributors from top 500 universities

BOOK CITATION INDEX
CLARIVATE ANALYTICS
INDEXED

WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com

Multi-Flock Flocking for Multi-Agent Dynamic Systems

Andrew McKenzie, Qingquan Sun and Fei Hu
Department of Electrical and Computer Engineering
University of Alabama
USA

1. Introduction

There has been much research done on single-flock flocking Shaw (1975), Partridge (1984), Partridge (1982), Okubo (1986), Reynolds (1987), Vicsek et al. (1995), Toner & Tu (1998), Shimoyama et al. (1996), Mogilner & Edelstein-Keshet (1999), Helbing et al. (2000), Vicsek (2001), Parrish et al. (2002), Olfati-Saber (2006), but none done on multi-flock flocking Gazi & Fidan (2007). One might ask, "Why would we need multiple flock flocking?" Consider the following scenario: there are two groups (squads/flocks) of Unmanned Vehicles (UV), both being in between the other group and the other groups' objective/goal. If both groups had the same capabilities then all we would need to do is to swap the groups goals. Unfortunately the groups have different sensing capabilities. One group of UV's is equipped with infrared cameras and the other with high-resolution cameras. Since each group is in the way of the other, it would be great if they could move out of each other's way. This in turn would decrease the amount of time for both groups to meet their goals.

We propose a new flocking algorithm that allows flocks to maneuver around other flocks (if needed) decreasing the amount of time each flock takes to reach their respective goals. We will do this by adding an additional agent, τ, to Olfati-Saber's Olfati-Saber (2006) existing α, β and γ-agents. The resulting algorithm will be compared to Olfati-Saber's flocking algorithm. Both algorithms will be simulated in multiple scenarios using Matlab. The scenarios will consist of both flock's being in-between the other flock and the other flocks goal, using different size flocks and only 1 group for a baseline. Section 2 presents related works. Section 3 includes our approach and multi-flock flocking algorithm. Section 4 contains our simulation setup. The simulation results are in 5; followed by the analysis in Section 6. Conclusions and future directions are in Section 7.

2. Related works

2.1 Flocking

Flocking is a kind of group behavior that includes a common objective and local interactions over a large number of group members. We find the emergence of flocking from many animals such as birds, fish, penguins, bees, and crowds, as well as swarming, and schooling Reynolds (1987), Partridge (1982), Toner & Tu (1998), Shimoyama et al. (1996).

Currently the flocking technique is mainly used in massive sensing using mobile sensor networks, self-assembly of connected mobile networks, and performing military missions such as reconnaissance, and surveillance. The self-organized feature of flocks can provide a heuristic conception in the design of mobile sensor networks and robotics systems.

The development of flocking techniques has had three phases. The first phase was primarily from a theoretical perspective. The typical researchers include: Viscek et al. Vicsek et al. (1995), whose work was mainly focused on alignment in self-driven particle systems; Toner and Tu Toner & Tu (1998) proposed a new scheme called continuum mechanics; and Levine et al. Levine et al. (2000), who developed a novel algorithm called rotating swarms to simulate ant mills with the all-to-all interactions. Additionally, several other continuum models of swarms were proposed which include works by Mogilner and Eldstein-Keshet Mogilner & Edelstein-Keshet (1996), Mogilner & Edelstein-Keshet (1999), and Topaz and Bertozzi Topaz & Bertozzi (2004). Helbing et al. Helbing et al. (2000) designed an empirical particle-based flocking model to study the escape panic phenomenon.

The second phase focused on the consensus problem and network topology. The contributions were mainly made by Olfati-Saber and Murray Saber & Murray (2003) Olfati-Saber & Murray (2004), Jadbabaie et al. Jadbabaie et al. (2003), Moreau Moreau (2005), and Ren and Beard Ren & Beard (2005). Although in the alignment problem, there is no constraint on the consensus value, when used for networked dynamic systems, the objective is distributed computation of a function via agreement Saber & Murray (2003), Olfati-Saber et al. (2007). Olfati-Saber and Murray Saber & Murray (2003) Olfati-Saber & Murray (2004) created a graph-induced potential function based structural formation control. Another work that belongs to this phase is on formation control and graph Laplacian by Fax and Murray Fax & Murray (2004).

Nowadays, the stability analysis of particles or agents with all-to-all interactions draws more attention. With respect to this issue, Tanner et al. Tanner et al. (2007) proposed a centralized algorithm for a particle system which leads to irregular collapse. They also proposed a distributed algorithm that leads to irregular fragmentation. Since collapse and fragmentation are two usual pitfalls of flocking, stability analysis on collapse and fragmentation is the evaluation method used for modern flocking algorithms.

2.2 Distributed intelligence in multi-robot systems

In Parker (2008), Parker gives an overview of the distributed intelligence field and its use in multi-robot systems. She first defines distributed intelligence and then defines the domain space of distributed intelligence. She defined four types of interactions in distributed intelligence: collective; cooperative; collaborative; and coordinative.

Collective interaction is defined as when entities are not aware of other entities on their team but share the same goals. The entities also help each other even though they are not planned to do so. An example of collective interaction is swarm robotics. Cooperative interaction is defined by entities being aware of the others and share goals. They also benefit each other with their actions. An example of cooperative interaction would be a team of robots who share map information and are trying to explore an unknown area. Collaborative interaction is defined by the entities having individual goals, and being aware of other entities on the team. The entities actions help the others achieve their own goals. Coordinative interaction occurs when entities have individual goals and can communicate with other entities. They do

not have a common goal and their actions are not helpful to the other entities. Essentially all of the entities are selfish and only care about their own goal.

She also defined three paradigms for distributed intelligence, which are: bio inspired, emergent swarms paradigm; organizational and social paradigms; and knowledge-based, ontological, and semantic paradigms. The main message of her paper can only be come from her words when she wrote "The main message of these discussions is that the choice of paradigm is not always obvious, and is dependent upon the requirements of the application to be addressed. We also note that complex systems of multiple robots can make use of several different paradigms simultaneously"Parker (2008).

3. Approach

In this section we will present two distributed algorithms for multi-flock flocking algorithms. The first algorithm is Olfati-Saber's from Olfati-Saber (2006) and the second algorithm is our algorithm, which extends Olfati-Saber's but adds an additional agent that adds coordinative interaction.

3.1 Olfait-Saber's flocking algorithm

Olfati-Saber's algorithm includes 3 agents: *alpha*, *beta* and *gamma*. All entities, such as a single bird in a flock of birds, are physical agent's with dynamics $\ddot{q}_i = u_i$ called an *alpha*-agent. The other two agents, *beta* and *gamma*, are virtual agents which model the effects of obstacles and the groups collective objective. The flocking algorithm has the capability to perform multiple obstacle avoidance. The equation consists of three terms:

$$u_i = u_i^\alpha + u_i^\beta + u_i^\gamma \tag{1}$$

where u_i^α denotes the (α, α) interaction terms, u_i^β denotes the (α, β) interaction terms, and u_i^γ is the groups distributed navigational feedback. The (α, α) interaction is used to keep the agents in a lattice /flock form. The (α, β) interaction term is used for obstacle avoidance, where a virtual β-agent is on the closest point of the obstacle from the α-agent. The γ-agent is a virtual leader, used to lead the flock to the desired location. Each term in equation 1 is explained as:

$$u_i^\alpha = c_1^\alpha \sum_{j \in N_i^\alpha} \phi_\alpha(\left\| q_j - q_i \right\|_\sigma) \mathbf{n}_{i,j}$$

$$+ c_2^\alpha \sum_{j \in N_i^\alpha} a_{ij}(q)(p_j - p_i)$$

$$u_i^\beta = c_1^\beta \sum_{k \in N_i^\beta} \phi_\beta(\left\| \hat{q}_{i,k} - q_i \right\|_\sigma) \hat{\mathbf{n}}_{i,k} \tag{2}$$

$$+ c_2^\beta \sum_{j \in N_i^\beta} b_{i,k}(q)(\hat{p}_{i,k} - p_i)$$

$$u_i^\gamma = -c_1^\gamma \sigma_1(q_i - q_r) - c_2^\gamma(p_i - p_r)$$

where $\sigma_1(z) = \frac{z}{\sqrt{1+\|z\|^2}}$ and c_η^v are positive constants for all $\eta = 1, 2$ and $v = \alpha, \beta, \gamma$. Each α-agent's state is denoted by (q_i, p_i), where q_i is the position and p_i is the velocity of the agent. The pair (q_r, p_r) is the state of a static or dynamic γ-agent. The vectors $\mathbf{n}_{i,j}$ and $\hat{\mathbf{n}}_{i,k}$ are given by $n_{i,j} = \frac{q_j - q_i}{\sqrt{1+\varepsilon\|q_j-q_i\|^2}})$, $\hat{n}_{i,k} = \frac{\hat{q}_{i,k} - q_i}{\sqrt{1+\varepsilon\|\hat{q}_{i,k}-q_i\|^2}}$.

3.2 Our flocking algorithm with coordinative interaction

Our equation has four agents: α, β, γ and τ. Where the α, β and γ agents are the same as in Olfati-Saber's Algorithm. The τ-agent is a virtual agent which is used to add Coordinative Interaction between the α agents. The equation consists of four terms:

$$u_i = u_i^\alpha + u_i^\beta + u_i^\gamma + u_\tau \tag{3}$$

where u_i^α, u_i^β and u_i^γ are the same as in equation 1. The coordinative interaction is added using the τ-agent. Each term in equation 3 is explained as:

$$
\begin{aligned}
u_i^\alpha &= c_1^\alpha \sum_{j \in N_i^\alpha} \phi_\alpha(\left\|q_j - q_i\right\|_\sigma)\mathbf{n}_{i,j} \\
&\quad + c_2^\alpha \sum_{j \in N_i^\alpha} a_{ij}(q)(p_j - p_i) \\
u_i^\beta &= c_1^\beta \sum_{k \in N_i^\beta} \phi_\beta(\left\|\hat{q}_{i,k} - q_i\right\|_\sigma)\hat{\mathbf{n}}_{i,k} \\
&\quad + c_2^\beta \sum_{j \in N_i^\beta} b_{i,k}(q)(\hat{p}_{i,k} - p_i) \\
u_i^\gamma &= -c_1^\gamma \sigma_1(q_i - q_r) - c_2^\gamma(p_i - p_r) \\
u_i^\tau &= -c_1^\tau \sigma_2(q_i - q_r) \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \\
&\quad - c_2^\tau \sigma_2(p_i - p_r) \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}
\end{aligned}
\tag{4}
$$

As in equation 2, each α-agent's state is denoted by (q_i, p_i), where q_i is the position (with an x and y component) and p_i is the velocity of the agent. The pair (q_r, p_r) is the state of a static/dynamic γ-agent. The other components are defined as $\sigma_1(z) = \frac{z}{\sqrt{1+\|z\|^2}}$; $\sigma_2(z)$ is 1 if $\frac{\pi}{2} \geq \theta_{i-j} \geq \frac{3\pi}{2}$ else it is 0; $z \leq d_c$; $\theta_i = \tan(p_{i_y}/p_{i_x})$, where p_{i_y} and p_{i_x} are x and y components of the agents velocity respectively; $\theta_{i-j} = \theta_i - \theta_j$; and d_c is the cooperation distance for all j α-agents in both flocks. Essentially the τ agent is applied if two agents' (i and j) trajectories are going to intersect and their distance is less then d_c. c_η^v are positive constants for all $\eta = 1, 2$ and $v = \alpha, \beta, \gamma, \tau$. The vectors $\mathbf{n}_{i,j}$ and $\hat{\mathbf{n}}_{i,k}$ are given by $n_{i,j} = \frac{q_j - q_i}{\sqrt{1+\varepsilon\|q_j-q_i\|^2}})$, $\hat{n}_{i,k} = \frac{\hat{q}_{i,k} - q_i}{\sqrt{1+\varepsilon\|\hat{q}_{i,k}-q_i\|^2}}$.

4. Simulation setup

Both flocking algorithms, Olfati-Saber's algorithm (equation 1) and our algorithm (equation 3), were implemented using Matlab. Matlab version 7.6.0 (R2008a) was used to run the simulations. The following parameters were used for both algorithms: $d = 7$ meters, $r = 1.2d$, the time step size is 0.01 seconds. Both algorithms used the same values for all constants: $(c_1^\alpha, c_2^\alpha, c_1^\beta, c_2^\beta, c_1^\gamma \, and \, c_2^\gamma)$.

Simulations were run using 1 and 2 groups of agents, consisting of 5, 10, 15, 20, 25, 30, 35, 40, 45 and 50 agents in each group. The groups were initially positioned into lattices with lines of 10 agents. Each group started a distance of 150 meters from their goal. If 2 groups were used, the groups were put directly in-between the other group and its respective goal. The groups were started 100 meters apart from each other. The agents were limited to a maximum speed of 7 m/s. Agents were considered to have reached their groups goal if they got within 14 meters ($2d$) of the goal. Each simulation was run until all agents reached their goal or until 480 seconds, in simulation time, had passed. Each simulation's time to finish was recorded. The time to finish recorded the time from the start of the simulation till all agents had reached their goal. One image was saved every second of all simulations.

5. Simulation results

The simulations were broken into two groups for analysis. Group A contains the results from the simulations where the agents start in a symmetric lattice (i.e. groups that had a multiple of 10 agents in them as in Figure 1 a). The results for group A can be seen in Table 1 as well as in Figure 2.

Agents per group	1 Group with and without Tao agent	2 Groups with Tao agent	2 Groups without Tao agent
10	139.11	134.37	did not finish
20	143.6	167.15	190.19
30	148.01	170.93	212.19
40	152.69	183.39	218.18
50	159.23	186.28	227.9

Table 1. Simulation results showing the total time to finish for symmetric lattices, group A.

Group B contains the results from the simulations when the agents started in a non-symmetric lattice (i.e. groups that did not have a multiple of 10 agents in them, as in Figure 3). The results for group B can be seen in Table 2 and in Figure 4. The combined results of both groups can be seen in Figure 5.

Selected images from group 1 at key points in the experiments can be seen in Figures: 6, 7, 1, 8 and 9. Selected images from group 2 can be seen in Figure 3. Figure 6 shows images for 2 groups with 10 agents per group using the τ agent flocking algorithm. Images from the experiments with 2 groups with 50 agents per group using the τ agent are shown in figure 1, and without using the τ agent in figures 8 and 9.

(a) With τ-agent, n=50, t=2s

(b) With τ-agent, n=50, t=13s

(c) With τ-agent, n=50, t=15s

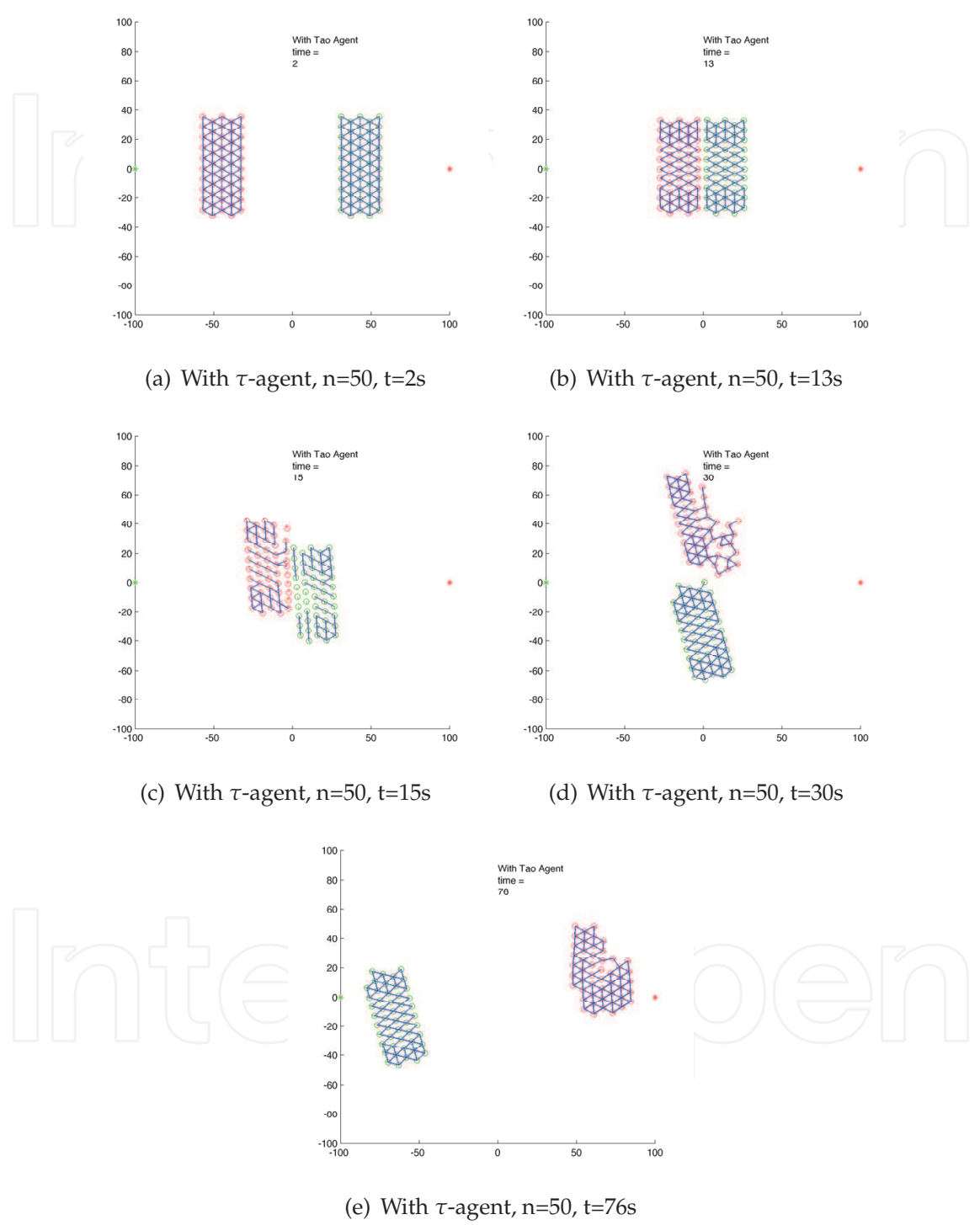(d) With τ-agent, n=50, t=30s

(e) With τ-agent, n=50, t=76s

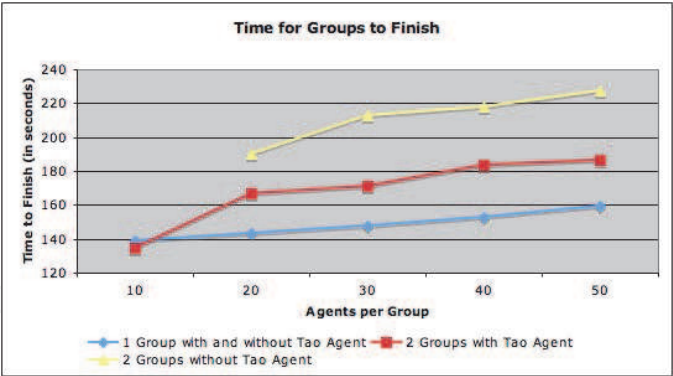Fig. 1. With τ-agent, 2 Groups, 50 agents per group

Fig. 2. Symmetric lattice finish times, Group 1
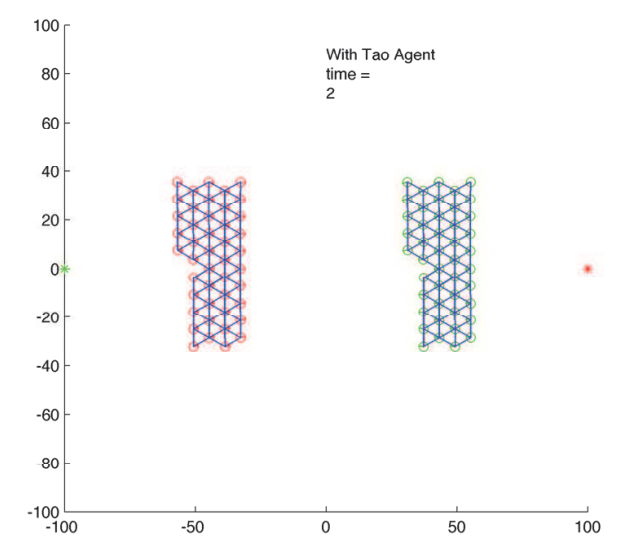
Fig. 3. Non-symmetric lattice, with τ, 2 groups and 45 agents per group

Agents per group	1 Group with and without Tao agent	2 Groups with Tao agent	2 Groups without Tao agent
5	113.23	112.61	did not finish
15	127.92	148.04	162.44
25	139.77	167	187.46
35	145.17	177.98	201.2
45	153.66	186.41	222.04

Table 2. Simulation results for Non-symmetric Lattice, group B.

Images comparing experiments with and without the τ agent, showing side by side images, can be seen in figures 10 and 11 for 10 agents per group and figures 12 and 13 for 50 agents per group.
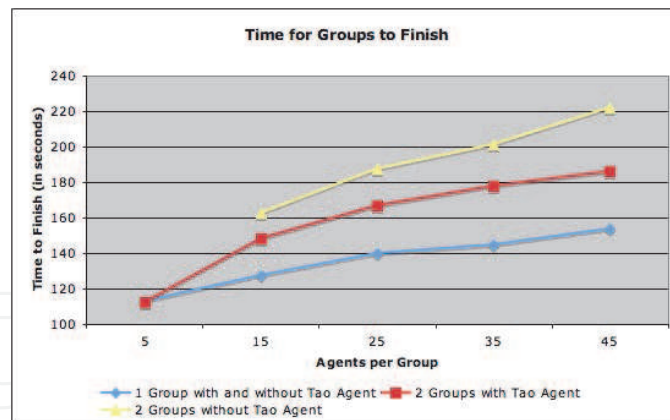
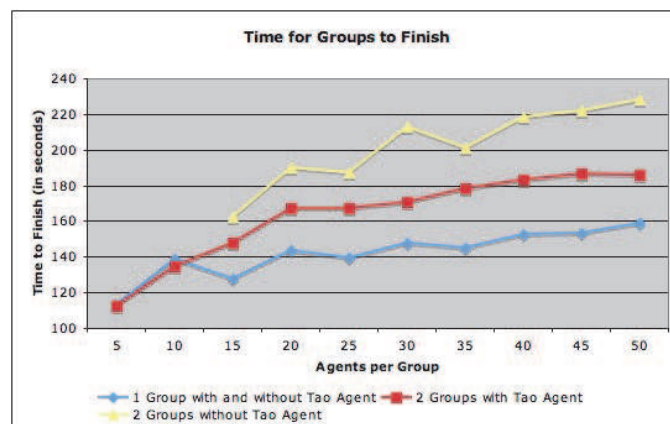Fig. 4. Non-symmetric lattice finish times, group B

Fig. 5. Non-symmetric and symmetric lattice finish times

6. Analysis

One can deduce from figure 5, that our additional τ agent, eq 3, improves the Olfati-Saber algorithm, eq 1. This is because without the τ agent, when the two flocks come close to each other a huge traffic jam occurs because the two flocks want to go the same direction. This can be seen in figures 7, 8 and 9. What is interesting is that a deadlock occurred in the experiments with 5 and 10 agents per group. Although, the deadlock was unexpected, it is easily explainable. The reason the deadlock occurs is because the flocks do not have the additional velocity, essentially a push, from the additional rows of agents behind them as in all of the other experiments.

Another interesting result is that all of the 1 group experiments, group B finished faster then their counterpart in group A, n=15 vs. n=10, n=25 vs. n=20, n=35 vs. n=30, and n=45 vs. n=40). This also occurred in the without τ, equation 1, experiments except for n=15 vs. n=10, because the agents did not finish in n=5 and n=10 This result is due to the fact that the groups are not symmetric but rather rotated versions of the other which can be seen in figure 3. Because of the non symmetric lattices the groups actions are different from each other when the traffic jam occurs, which in turn allows the groups to get around each other faster.
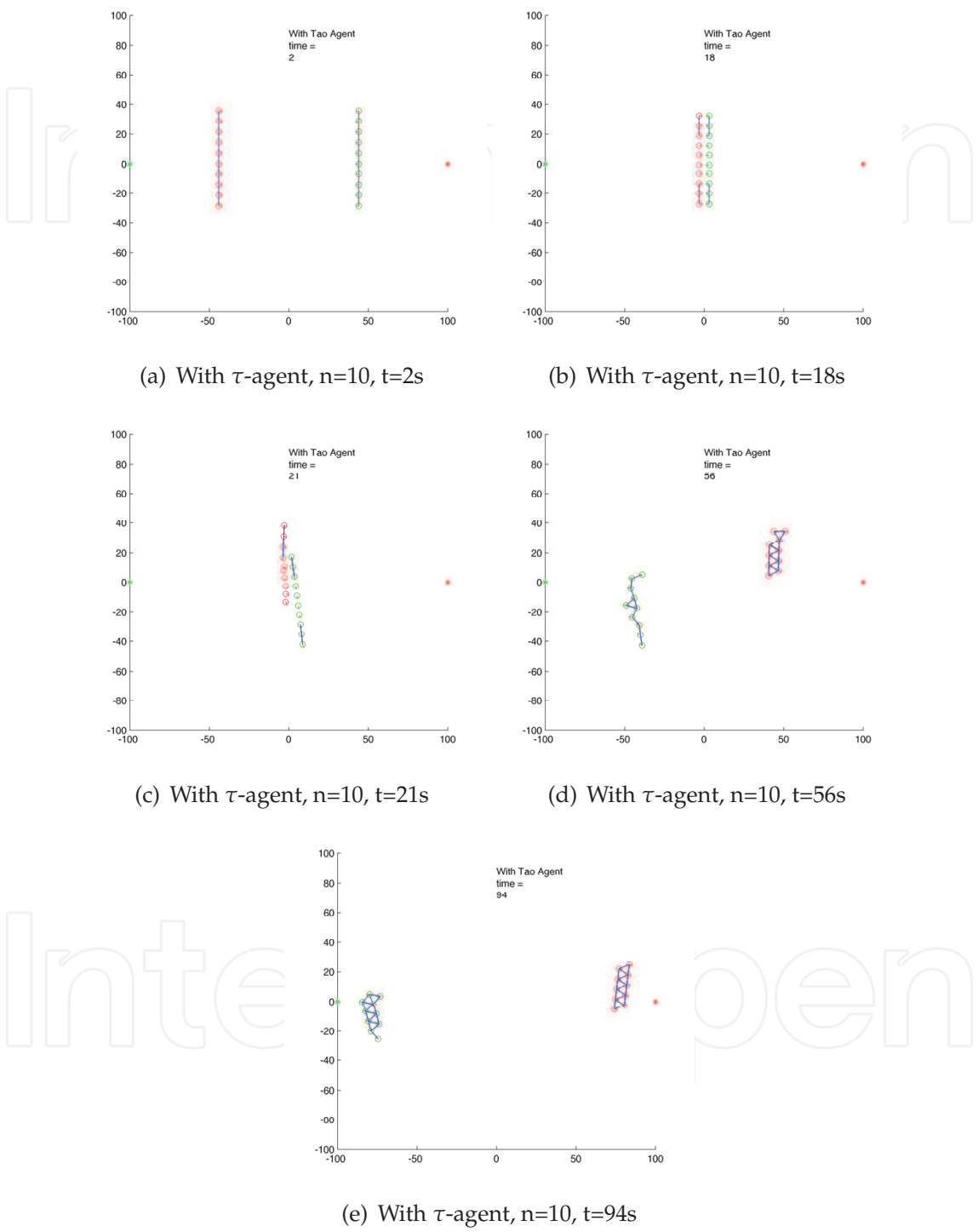
(a) With τ-agent, n=10, t=2s

(b) With τ-agent, n=10, t=18s

(c) With τ-agent, n=10, t=21s

(d) With τ-agent, n=10, t=56s

(e) With τ-agent, n=10, t=94s

Fig. 6. With Tao Agent, 2 Groups, 10 agents per group

(a) Without τ-agent, n=10, t=2s

(b) Without τ-agent, n=10, t=18s

(c) Without τ-agent, n=10, t=21s

(d) Without τ-agent, n=10, t=40s

(e) Without τ-agent, n=10, t=480s

Fig. 7. Without Tao Agent, 2 Groups, 10 agents per group

(a) Without τ-agent, n=50, t=2s

(b) Without τ-agent, n=50, t=14s

(c) Without τ-agent, n=50, t=16s

(d) Without τ-agent, n=50, t=86s

(e) Without τ-agent, n=50, t=101s

(f) Without τ-agent, n=50, t=122s

Fig. 8. Without τ-agent, 2 Groups, 50 agents per group

(a) Without τ-agent, n=50, t=138s   (b) Without τ-agent, n=50, t=146s

(c) Without τ-agent, n=50, t=159s   (d) Without τ-agent, n=50, t=189s

Fig. 9. Without τ-agent, 2 Groups, 50 agents per group

(a) Without τ-agent, n=10, t=2s

(b) With τ-agent, n=10, t=2s

(c) Without τ-agent, n=10, t=18s

(d) With τ-agent, n=10, t=18s

(e) Without τ-agent, n=10, t=21s

(f) With τ-agent, n=10, t=21s

Fig. 10. With and Without τ-agent, 2 Groups, 10 agents per group

(a) Without τ-agent, n=10, t=40s

(b) With τ-agent, n=10, t=56s

(c) Without τ-agent, n=10, t=480s

(d) With τ-agent, n=10, t=94s

Fig. 11. With and Without τ-agent, 2 Groups, 10 agents per group

(a) Without τ-agent, n=50, t=2s

(b) With τ-agent, n=50, t=2s

(c) Without τ-agent, n=50, t=14s

(d) With τ-agent, n=50, t=13s

(e) Without τ-agent, n=50, t=16s

(f) With τ-agent, n=50, t=15s

Fig. 12. With and Without τ-agent, 2 Groups, 50 agents per group

(a) Without τ-agent, n=50, t=86s

(b) With τ-agent, n=50, t=30s

(c) Without τ-agent, n=50, t=101s

(d) With τ-agent, n=50, t=76s

Fig. 13. With and Without τ-agent, 2 Groups, 50 agents per group

7. Conclusion and future work

In this paper we presented an improved multi flock flocking algorithm (equation 3), that added coordinative interaction to Olfati-Saber's flocking algorithm Olfati-Saber (2006) (equation 1). Our algorithm performed better than Olfati-Saber's algorithm in simulations where two groups were in between the other group and the other groups' goal. It performed exactly as Olfati-Saber's algorithm when there was only one flock. This was because our algorithm was based on Olfati-Saber's, and the only difference was when two flocks were close enough to the other flock and the flocks trajectories were set to intersect the other.

The next stage of this work would be to improve the algorithm by modifying our τ agent to change the agent's trajectory based on the trajectory of both agents instead of just rotating τ agent 90 degrees. Another improvement would be to weight the τ agent based on the number of agents in the other flocks. These improvements should allow for a better interaction between multiple flocks instead.

8. References

Fax, J. & Murray, R. (2004). Information flow and cooperative control of vehicle formations, *Automatic Control, IEEE Transactions on* 49(9): 1465–1476.

Gazi, V. & Fidan, B. (2007). Coordination and control of multi-agent dynamic systems: models and approaches, *Proceedings of the 2nd international conference on Swarm robotics*, SAB'06, Springer-Verlag, Berlin, Heidelberg, pp. 71–102.
URL: *http://portal.acm.org/citation.cfm?id=1763837.1763843*

Helbing, D., Farkas, I. & Vicsek, T. (2000). Simulating dynamical features of escape panic, *Nature* 407(6803): 487–490.
URL: *http://dx.doi.org/10.1038/35035023*

Jadbabaie, A., Lin, J. & Morse, A. (2003). Coordination of groups of mobile autonomous agents using nearest neighbor rules, *Automatic Control, IEEE Transactions on* 48(6): 988–1001.

Levine, H., Rappel, W.-J. & Cohen, I. (2000). Self-organization in systems of self-propelled particles, *Phys. Rev. E* 63(1): 017101.

Mogilner, A. & Edelstein-Keshet, L. (1996). Spatio-temporal order in populations of self-aligning objects: formation of oriented patches, *Physica D* 89: 346–367.

Mogilner, A. & Edelstein-Keshet, L. (1999). A non-local model for a swarm, *Journal of Mathematical Biology* 38: 534–570. 10.1007/s002850050158.
URL: *http://dx.doi.org/10.1007/s002850050158*

Moreau, L. (2005). Stability of multiagent systems with time-dependent communication links, *Automatic Control, IEEE Transactions on* 50(2): 169–182.

Okubo, A. (1986). Dynamical aspects of animal grouping: swarms, schools, flocks. and herds, *Advances in Biophysics* 22: 1–94.

Olfati-Saber, R. (2006). Flocking for multi-agent dynamic systems: algorithms and theory, *Automatic Control, IEEE Transactions on* 51(3): 401 – 420.

Olfati-Saber, R., Fax, J. & Murray, R. (2007). Consensus and cooperation in networked multi-agent systems, *Proceedings of the IEEE* 95(1): 215–233.

Olfati-Saber, R. & Murray, R. (2004). Consensus problems in networks of agents with switching topology and time-delays, *Automatic Control, IEEE Transactions on* 49(9): 1520–1533.

Parker, L. (2008). Distributed intelligence: Overview of the field and its application in multi-robot systems, *Journal of Physical Agents* 2(1).
URL: *http://www.jopha.net/index.php/jopha/article/view/18/14*

Parrish, J. K., Viscido, S. V. & GrÃijnbaum, D. (2002). Self-organized fish schools: An examination of emergent properties, *Biol. Bull* 202: 296–305.

Partridge, B. L. (1982). The structure and function of fish schools, *Scientific American* 246(6): 114–123.

Partridge, B. L. (1984). The chorus-line hypothesis of maneuver in avian flocks, *Nature* 309: 344 – 345.

Ren, W. & Beard, R. (2005). Consensus seeking in multiagent systems under dynamically changing interaction topologies, *Automatic Control, IEEE Transactions on* 50(5): 655 – 661.

Reynolds, C. W. (1987). Flocks, herds and schools: A distributed behavioral model, *Proceedings of the 14th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '87, ACM, New York, NY, USA, pp. 25–34.
URL: *http://doi.acm.org/10.1145/37401.37406*

Saber, R. & Murray, R. (2003). Consensus protocols for networks of dynamic agents, *American Control Conference, 2003. Proceedings of the 2003*, Vol. 2, pp. 951–956.

Shaw, E. (1975). Fish in schools, *Natural History* 84(8): 40–45.

Shimoyama, N., Sugawara, K., Mizuguchi, T., Hayakawa, Y. & Sano, M. (1996). Collective motion in a system of motile elements, *Phys. Rev. Lett.* 76(20): 3870–3873.

Tanner, H. G., Jadbabaie, A. & Pappas, G. J. (2007). Flocking in fixed and switching networks, *Automatic Control, IEEE Transactions on* 52(5): 863–868.

Toner, J. & Tu, Y. (1998). Flocks, herds, and schools: A quantitative theory of flocking, *Phys. Rev. E* 58(4): 4828–4858.

Topaz, C. & Bertozzi, A. L. (2004). Swarming patterns in a two-dimensional kinematic model for biological groups, *SIAM Journal on Applied Mathematics* 65(1): 152–174.

Vicsek, T. (2001). A question of scale, *NATURE* 411: 421.
URL: *doi:10.1038/35078161*

Vicsek, T., Czirók, A., Ben-Jacob, E., Cohen, I. & Shochet, O. (1995). Novel type of phase transition in a system of self-driven particles, *Phys. Rev. Lett.* 75(6): 1226–1229.

Mobile Robots - Control Architectures, Bio-Interfacing, Navigation, Multi Robot Motion Planning and Operator Training
Edited by Dr. Janusz Będkowski

The objective of this book is to cover advances of mobile robotics and related technologies applied for multi robot systems' design and development. Design of control system is a complex issue, requiring the application of information technologies to link the robots into a single network. Human robot interface becomes a demanding task, especially when we try to use sophisticated methods for brain signal processing. Generated electrophysiological signals can be used to command different devices, such as cars, wheelchair or even video games. A number of developments in navigation and path planning, including parallel programming, can be observed. Cooperative path planning, formation control of multi robotic agents, communication and distance measurement between agents are shown. Training of the mobile robot operators is very difficult task also because of several factors related to different task execution. The presented improvement is related to environment model generation based on autonomous mobile robot observations.

How to reference
In order to correctly reference this scholarly work, feel free to copy and paste the following:

Andrew McKenzie, Qingquan Sun and Fei Hu (2011). Multi-Flock Flocking for Multi-Agent Dynamic Systems, Mobile Robots - Control Architectures, Bio-Interfacing, Navigation, Multi Robot Motion Planning and Operator Training, Dr. Janusz Będkowski (Ed.), ISBN: 978-953-307-842-7, InTech, Available from: http://www.intechopen.com/books/mobile-robots-control-architectures-bio-interfacing-navigation-multi-robot-motion-planning-and-operator-training/multi-flock-flocking-for-multi-agent-dynamic-systems

INTECH
open science | open minds