

Self-organized Collective Motion with a Simulated Real Robot Swarm^{*}

Mohsen Raoufi¹, Ali Emre Turgut², and Farshad Arvin³

¹ Department of Aerospace Engineering, Sharif University of Technology, Tehran, Iran, email: mohsen.raoufi@ae.sharif.edu

² Mechanical Engineering Department, Middle East Technical University, 06800 Ankara, Turkey, email: aturgut@metu.edu.tr

³ School of Electrical and Electronic Engineering, The University of Manchester, M13 9PL, Manchester, UK, email: farshad.arvin@manchester.ac.uk

Abstract. Collective motion is one of the most fascinating phenomena observed in the nature. In the last decade, it aroused so much attention in physics, control and robotics fields. In particular, many studies have been done in swarm robotics related to collective motion, also called flocking. In most of these studies, robots use orientation and proximity of their neighbors to achieve collective motion. In such an approach, one of the biggest problems is to measure orientation information using on-board sensors. In most of the studies, this information is either simulated or implemented using communication. In this paper, to the best of our knowledge, we implemented a fully autonomous coordinated motion without alignment using very simple Mona robots. We used an approach based on Active Elastic Sheet (AES) method. We modified the method and added the capability to enable the swarm to move toward a desired direction and rotate about an arbitrary point. The parameters of the modified method are optimized using TCACS optimization algorithm. We tested our approach in different settings using Matlab and Webots.

Keywords: Swarm Robotics · Collective Motion · Coordinated Motion · Flocking · Self-organized.

1 Introduction

Collective motion (CM) is an eye-catching demonstration of a more broad phenomenon called collective behaviour [22]. CM is observed in diverse domains such as: physics [11], chemistry [12], from micro-size creatures such as bacteria colonies [20], cells [1], macro-molecules [18] to handful macro-size examples of fish [24], birds [4,5], even humans [15] and to non-living systems [21]. The principle feature of CM is that each individual agent behaves based on the interaction between its neighbors; resulting in polarized motion of the group. This interaction might be either simple (attraction/repulsion) or complex (combinations of simple interactions) [23].

^{*} This work was supported by EPSRC Impact Acceleration Account (EP/R511626/1).

In most of the works on CM, agents use both orientation and proximity information of their neighbors [8,6]. In contrast to these works, some works introduced methods that do not rely on this information. An individual-based model is defined in [17], where the interaction between neighbor agents is defined only by escape-pursuit behavior. In [9], inelastic collisions between isotropic agents, and in [13] a pairwise repulsive force between deform-able, self-propelled particles are introduced as interaction mechanism regardless of the orientation of neighbors. These methods leave an indirect effect on individual agents, and as a result, enable them to establish aligning interaction in an effective, implicit manner. By so doing, not only the swarm achieves CM, but also the reduced dynamics reaches consensus in the heading direction of agents. An example of this implicit aligning is [7], in which, Ferrante et al. identified a new elasticity-based approach for achieving collective motion and illustrated the behavior by introducing an Active Elastic Sheet (AES) model. In their proposed model, the motion of an individual is determined by only attraction-repulsion forces. This feature is very useful in practical cases, such as a swarm of robots in which a specific robot have not any information of its neighbors' orientation [3].

In this paper, we improve the AES model presented in [7], and optimize some of the parameters of the proposed algorithm using a heuristic optimization algorithm. To illustrate the improved performance of the model, various simulations with different assumptions are performed. In addition, some simulations are conducted in Webots [14] using the model of Mona robot [2]. Finally, We implement the AES model on a swarm of simulated real Mona robots for the first time. To the best of our knowledge, this is the first implementation of fully autonomous CM using such a simple robotic platform without alignment information.

This paper is organized as follows: in the Sect. 2, we introduce the concept of AES mechanism to achieve CM, and the modifications are addressed there. Then, the optimization algorithm is discussed and the objective function is defined in Sect. 3. We mention the settings of simulations in the Sect. 4. In Sect. 5, the results are shown and we discuss them. Sect. 6 closes the paper with a conclusion about the current work.

2 Collective Motion Control

In this section, the dynamic model of AES along with some modifications is presented. The model is derived based on a simple two-dimensional active elastic sheet mechanism [7]. After defining the basic mechanism, the numerical dynamics of particles and modifications will be elaborated upon.

2.1 Active Elastic Sheet Model and Numerical Dynamics

Each agent i out of N agents, is affected by the attraction-repulsion force of its neighbors. This force will leave an effect on both the linear velocity $\dot{\mathbf{x}}_i$ and the rotational velocity $\dot{\theta}_i$ of the agent. This effect is formulated in continuous-time

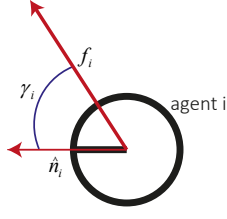


Fig. 1: The angle between force and heading vectors

form as:

$$\dot{\mathbf{x}}_i(t) = \left\{ v_0(t) + \alpha \left[\left(\mathbf{f}_i(t) + D_r \hat{\boldsymbol{\xi}}_r(t) \right) \cdot \hat{\mathbf{n}}_i(t) \right] \right\} \hat{\mathbf{n}}_i(t) \quad (1)$$

$$\dot{\theta}_i(t) = \beta \left\{ \left[\mathbf{f}_i(t) + D_r \hat{\boldsymbol{\xi}}_r(t) \right] \cdot \hat{\mathbf{n}}_i^\perp(t) \right\} + D_\theta \boldsymbol{\xi}_\theta(t) \quad (2)$$

where, v_0 is the forward biasing speed, and parameters α and β are the inverse transitional and rotational coefficients, respectively. The unit vector $\hat{\mathbf{n}}_i(t)$ is the heading vector which is parallel to the heading direction of agent i , and $\hat{\mathbf{n}}_i^\perp(t)$ is perpendicular to it.

$$\hat{\mathbf{n}}_i(t) = \begin{bmatrix} \cos(\theta_i(t)) \\ \sin(\theta_i(t)) \end{bmatrix} \quad (3)$$

The term $\mathbf{f}_i \cdot \hat{\mathbf{n}}_i^\perp$ of Eq. (2) indicates that the angle between force and heading vectors (γ_i) defines the rotation speed of the agent (Fig. 1). So, the rotation speed will be zero if two vectors \mathbf{f}_i and $\hat{\mathbf{n}}_i$ are parallel (i.e. \mathbf{f}_i and $\hat{\mathbf{n}}_i^\perp$ are perpendicular.)

Agents are subject to measurement and actuation noises $D_r \hat{\boldsymbol{\xi}}_r(t)$ and $D_\theta \boldsymbol{\xi}_\theta(t)$; in which, $\hat{\boldsymbol{\xi}}_r(t)$ is a randomly oriented unit vector and $\boldsymbol{\xi}_\theta(t)$ is a random variable with standard normal distribution. D_r and D_θ are noise strength coefficients. Inducing total force vector $\mathbf{f}_i(t)$ over agent i at time t can be calculated by the summation of linear spring-like forces, which link agent i to its interacting neighbor j , member of set S_i :

$$\mathbf{f}_i(t) = \sum_{j \in S_i} -\frac{k}{l_{ij}} (\|\mathbf{r}_{ij}(t)\| - l_{ij}) \frac{\mathbf{r}_{ij}(t)}{\|\mathbf{r}_{ij}(t)\|} \quad (4)$$

$$\mathbf{r}_{ij}(t) = \mathbf{x}_j(t) - \mathbf{x}_i(t) \quad (5)$$

l_{ij} is equilibrium distance (or alternatively called natural length) of the spring that links agent i and j . In order to simulate the model, we need to numerically integrate the dynamic equations of motion. Integrating Eq. (1) and (2) using Euler method at time step t , results in the governing discrete-time kinematic equations of each particle which is expressed by the following equations:

$$\mathbf{x}_i^{t+1} = \mathbf{x}_i^t + \left\{ v_0 \hat{\mathbf{n}}_i^t + \alpha \left[\left(\mathbf{f}_i^t + \frac{D_r}{\sqrt{\Delta t}} \hat{\boldsymbol{\xi}}_r^t \right) \cdot \hat{\mathbf{n}}_i^t \right] \hat{\mathbf{n}}_i^t \right\} \Delta t \quad (6)$$

$$\theta_i^{t+1} = \theta_i^t + \left\{ \beta \left[\left(\mathbf{f}_i^t + \frac{D_r}{\sqrt{\Delta t}} \hat{\boldsymbol{\xi}}_r^t \right) \cdot \hat{\mathbf{n}}_i^{\perp, t} \right] + \frac{D_\theta}{\sqrt{\Delta t}} \xi_\theta^t \right\} \Delta t \quad (7)$$

here, \mathbf{x}_i^t and θ_i^t are the position and orientation of agent i at step time t , respectively, and Δt is the numerical time-step interval of integration. It is noteworthy that the velocity vector in Eq. (1) points parallel to the heading of the agent which means it is assumed that each agent can only turn, and move forward/backward. Modeling agents with 2 degrees of freedom provides many advantages to robotic applications in which the robots are not able to move omni-directionally, e.g. Mona [2] robot.

2.2 Modifications to AES

In this section, we present some modifications to AES model, which result in collective linear and rotational motion of swarm. Firstly, we add a stimulating force to compel the swarm to move toward a specific direction. This modification leads to faster convergence of collective motion. Furthermore, the desired direction of swarm movement is also achievable by such modification. To model this auxiliary inducing force, at a certain time, we simply added the auxiliary force $\mathbf{f}_{aux, l}$ to the induced force of Eq. (4). This force is parallel to the desired velocity unit vector $\hat{\mathbf{v}}_d$ and its magnitude is defined by a weighting parameter w_l as follows:

$$\mathbf{f}_{aux, l} = w_l \hat{\mathbf{v}}_d \quad (8)$$

The second modification enables the swarm to rotate collectively by adding another auxiliary force. This force is also related to another desired linear velocity which itself is proportional to the rate of rotation. According to the kinematics of 2D rotation, the linear speed of each agent is proportional to its distance with respect to the center of rotation \mathbf{x}_c . Thus, pointed in the following equation, the linear velocity of rotation \mathbf{v}_r is related to \mathbf{r}_{ic} , the position vector of agent i wrt the center of rotation, and ω is the rate of rotation. $\mathbf{r}_{ic, x}$ and $\mathbf{r}_{ic, y}$ are the x- and y-component of \mathbf{r}_{ic} .

$$\mathbf{v}_r = \omega \begin{bmatrix} \mathbf{r}_{ic, y} \\ -\mathbf{r}_{ic, x} \end{bmatrix} \quad (9)$$

$$\mathbf{r}_{ic} = \mathbf{x}_i - \mathbf{x}_c \quad (10)$$

Rotation center \mathbf{x}_c is an arbitrary point, yet it is common to consider it on the center of swarm. Hence, same as the previous approach, we calculated the modified force for rotation by the following equation:

$$\mathbf{f}_{aux, r} = w_r \mathbf{v}_r \quad (11)$$

where, w_r is the weighting parameter of induced force for swarm rotation. It is worth mentioning that $\mathbf{f}_{aux, r}$ should be calculated for each agent, but $\mathbf{f}_{aux, l}$ is a universal force and is same for all agents. The linearity feature of the aforementioned phenomena enables us to add both terms together and form the total

auxiliary force, which still satisfies both motions. So, the final modified inducing force is:

$$\mathbf{f} = \mathbf{f}_{neighbors} + \mathbf{f}_{aux, l} + \mathbf{f}_{aux, r} \quad (12)$$

$\mathbf{f}_{neighbors}$ is exactly the same as Eq. (4), but we added the subscript to show that this is the force which is defined by the influence of interacting neighbors.

2.3 Degree of Alignment

Since the symmetry is a principal feature for achieving flocking behavior, a criteria is introduced to measure it. This metric is defined by a specific system variable, called the degree of alignment. One of the most common, and widely used definition is represented in [23]. In this case the variable ψ is the average normalized velocity vectors:

$$\psi = \frac{1}{Nv_0} \left\| \sum_{i=1}^N \mathbf{v}_i \right\| \quad (13)$$

where N is the total number of the agents, v_0 is the average absolute velocity, and \mathbf{v}_i is the velocity vector of the agent i . However, in methods with 2-DoF agents, like AES, as previously mentioned, the velocity vector is parallel to the heading direction. So, the degree of alignment of the system is defined as [7]:

$$\psi = \frac{1}{N} \left\| \sum_{i=1}^N \hat{\mathbf{n}}_i \right\| \quad (14)$$

Because our purpose is to simulate the AES method on robots, we added another criterion which is defined from a control viewpoint. In this respect, the alignment of each particle ψ_i is defined by its desired direction of movement, i.e. an agent is called aligned when its orientation is parallel to its desired direction of motion. Referred to Fig. 1, the alignment is defined by the angle γ_i as follows:

$$\psi_i = |\cos(\gamma_i)| = \left| \frac{\mathbf{f}_i \cdot \hat{\mathbf{n}}_i}{\|\mathbf{f}_i\|} \right| \quad (15)$$

In this definition, ψ_i is between zero (agent's orientation is perpendicular to the force, $\gamma_i = \pi/2$) and one (agent is directed toward the force, $\gamma_i = 0$ or $\gamma_i = \pi$). Note that, as the norm of the unit vector $\hat{\mathbf{n}}_i$ is equal to one, it is eliminated from the denominator. Therefore, the degree of system alignment is calculated by the following equation:

$$\psi = \frac{1}{N} \left\| \sum_{i=1}^N \psi_i \right\| \quad (16)$$

Mathematically, the value of this variable is between zero (asymmetric, disordered, random orientation) and one (fully symmetric orientation).

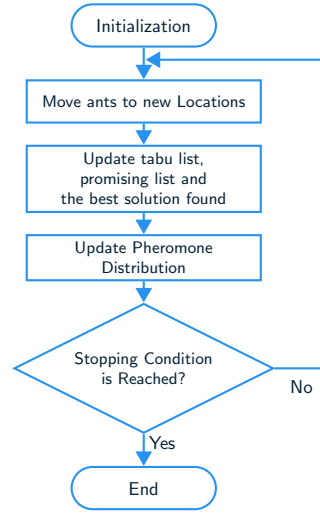


Fig. 2: General Flowchart of TCACS

3 Optimization

In this section, the parameter optimization of the algorithm will be introduced. For so doing, we utilized a heuristic optimization algorithm called Tabu Continuous Ant Colony System (TCACS) [10]. TCACS combines two algorithms: Continuous Ant Colony System (CACS) [16], and Tabu Search (TS) [19]. We will mention a brief description of the algorithm in the next subsection. Our aim is to find the optimal decision parameters for AES model with a specific setup configuration. The decision parameters to be tuned are α , β , and k .

3.1 TCACS Optimization Algorithm

The structure of the algorithm is very similar to CACS; however, the advantages of tabu concept are also added. Borrowing the concept of tabu balls (meta-spherical shapes in the decision-space), prevents the ants choosing their destination within the tabu regions. The flowchart of TCACS is illustrated in Fig. 2.

In this paper, we considered the parameters of TCACS as $N_{ants} = 15$, $\gamma = 0.5$, $m = 2$ (which are number of ants, weighting factor, and PCA factor, respectively) and used Roulette as the weighting strategy.

3.2 Objective Function

One of the most important steps in optimization is the definition of objective function, which can leave a considerable effect on the performance of the algorithm to solve the problem. Various objective functions can stand as candidates,

depending on what the aim of the problem is and which conditions needs to be satisfied. In case of collective motion using the modified AES method, we have two main objectives:

- To minimize the total induced force, which, generally speaking, means that agents are at their ideal position according to the elastic force of the sheet. Satisfying this objective will result in maintaining the shape of swarm same as its natural shape.
- To maximize the rate of convergence, which will be reached when the orientation of agents are about the same as the desired ones. To fulfill this objective, the optimization algorithm will try to maximize the degree of system alignment.

Although the defined problem is multi-objective, we combine both previous functions, multiplied by their weights, into a single objective function. This will help us to simplify and solve the parameter tuning problem in a more straight-forward manner.

It is notable that the system is subject to noise; consequently, the objective function gets stochastic, and it is necessary to reduce the effect of noise on it, otherwise, the performance of optimization will be reduced. Therefore, we calculate the average of multiple Monte Carlo simulations as the output of the objective function. With this in mind, the objective function is as follows:

$$J = \frac{1}{N_{MC}} \sum_{m=1}^{N_{MC}} \left\{ \sum_{t=0}^{t_f} \left[w_1 \left(\sum_{i=1}^N \|\mathbf{f}_i^t\| \right) + w_2 \psi^t \right] \right\} \quad (17)$$

In which, the first term of the RHS is calculated by Eq. (12), and the second term by Eq. (16). Additionally, N_{MC} is the number of Monte Carlo simulations which is set to 10, and t_f is the final time of simulation. Considering the fact that the first objective is a minimization problem, and the second objective is a maximization problem, we determined w_1 (the weighting parameter of the first function) positive and w_2 negative. As the result, the optimization can be alternatively called minimization, and the objective function is a *cost* function. Determining the value of weighting parameters depends mainly on the purpose of the optimization, however, a common initial setting is normalizing the order of each term; so, we considered them as $w_1 = 1/N$, and $w_2 = -1$.

4 Experimental Setup

To model Mona robot [2] in Webots, we assembled different parts including upper and lower boards, motors, and wheels which is shown in Fig. 3-b. A population of 100 Monas, where presented in the simulations and each one has its own specific ID in order for the *supervisor* to recognize them in the simulation world-model, and calculate the adequate displacement of linear and rotational motion. Besides the simulations in Webots, we also conducted some simulations in Matlab to check the modifications and to run optimization programs. In this

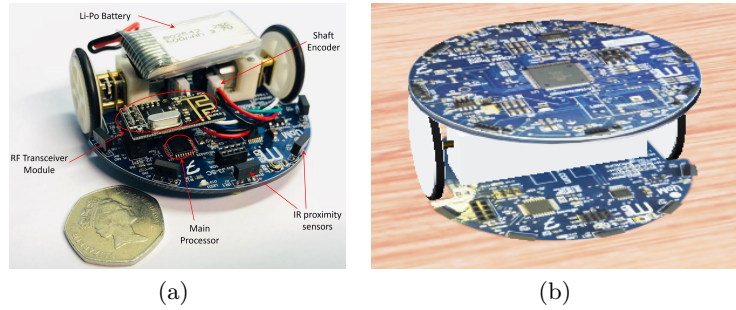


Fig. 3: (a) Monarch, an open-source miniature mobile robot developed for swarm robotic applications and (b) Monarch model in Webots.

paper, we represented three different setups for simulations in both Webots and Matlab, which will be described in the following subsections.

4.1 Setup 1: Linear Motion

The first simulation is executed to prove the linear motion of swarm is achievable thanks to the first modification. The configuration of this setup is similar to Table 1, but the rate of rotation is set to zero, $\omega = 0$. The result of this Matlab simulation is illustrated in the first row of Fig. 5.

4.2 Setup 2: Rotational Motion

Similarly, the second modification is verified by another simulation, in which the setting makes the swarm rotate about its center. The setting of Table 1 is analogous to the current setting, yet the desired linear speed is set to zero, $\|\hat{\mathbf{v}}_d = 0\|$. The second row of Fig. 5 represents a time lapse of this simulation in Matlab.

4.3 Setup 3: Combination of Linear & Rotational Motion

This simulation is conducted in Webots. We considered the robots as independent particles, each of which is determined by its orientation and position in the xy-plane. In this paper, a *supervisor* code is programmed to calculate the position and orientation of robots and dictate them to the corresponding agent. In other words, it is assumed that the controller of robots are ideal and the desired states of robots which supervisor calculates, is the same as the actual state of robots.

The settings of this simulation is placed at Table 1. By defining such settings, we expect the swarm to rotate about its center while moving collectively toward the west direction. To depict the result of the simulation, we saved a log file from Webots, and then plot the position and rotation of each robots in Matlab, results of which are placed in the third row of Fig. 5.

Table 1: AES Simulation Settings

Parameter	Description	Value
v_0	Absolute linear velocity of each agent	0.05 [m/sec]
t_f	Final time of simulation	30 [sec]
N	Number of agents	10*10
d_{init}	Initial distance between two side-by-side agents	0.2 [m]
$\hat{\mathbf{v}}_d$	Desired linear velocity unit vector	$[-1.0, 0.0]^T$ [m/sec]
w_l	Weight for desired linear velocity	0.8
ω	Desired rate of rotation	0.7 [rad/sec]
w_r	Weight for desired rotational velocity	1
D_r	Strength coefficient of measurement noise	0.5
D_θ	Strength coefficient of actuation noise	0.02

5 Results & Discussion

Applying TCACS optimization algorithm on the previously defined problem provided us the tuned parameter of the algorithm for a specific setting of simulation. The setting was considered as Table 1. In the optimization, the maximum number of function evaluations was limited to 300. Finally, the optimized parameters achieved as $\alpha^* = 0.066$, $\beta^* = 0.97$, and $k^* = 1.28$. The shape of swarm was considered as a square with 10 rows and 10 columns of agents.

In order to show the effect of optimized parameters on the performance of flocking, we investigated those objectives with both the optimized and non-optimized set of parameters. Fig. 4-b shows the significant improvements, in comparison with the default parameters Fig. 4-a, which is achieved by optimized parameters. As it is shown, the rate of convergence is increased, so the CM achieved faster. Besides, the total force of the network is decreased, meaning that the shape of the swarm is maintained better.

We also studied the effect of objective weights w_1 , and w_2 on the optimized parameters. Since the behavior of swarm varies for each obtained parameters, we defined two more different weight sets to show the effect of different weights on swarm behavior:

- More emphasis on the first objective, $w_1 = 10/N$, $w_2 = -1$ (Fig. 4-c)
- More emphasis on the second objective, $w_1 = 1/N$, $w_2 = -10$ (Fig. 4-d)

As we expected, the former setting resulted in the more reduction of total force, and the latter one, increased the rate of convergence of alignment, yet made the system more sensitive to noise (the effect of which is shown on the fluctuating behavior of force on Fig. 4-d). The first row of Fig. 5 proves that the initially perturbed formation of the swarm is eliminated, then the collective motion of swarm is achieved. The inducing linear force caused the swarm to move toward the west direction, as it was expected by $\hat{\mathbf{v}}_d$. Due to the robustness of AES to noises, noises were not effective enough to prevent the swarm from flocking. The time history of the second row, for rotational motion, proves that the second

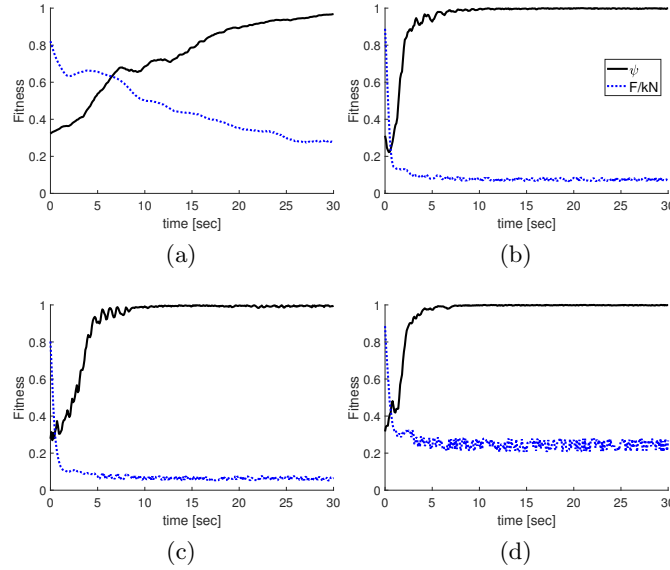


Fig. 4: Effect of Parameter optimization, and Weights on CM: a) Default parameters, b) optimized parameters, c) optimized parameters with emphasis on force objective, d) optimized parameters with emphasis on alignment objective

modification is verified, similarly. In this simulation, the swarm rotates about its center and maintain its shape, meanwhile.

The third row shows that the combination of linear and rotational motions made the swarm to move toward the desired direction, and in the mean time, rotate about its center. The same optimized parameters were used for this simulation, in Webots, proving that similar settings results in similar behavior in Matlab and Webots.

6 Conclusion

We have addressed the parameter optimization and Webots simulations of AES model, which is an elasticity-based approach for achieving CM. Some of its advantages such as not relying on exchanging orientation information, and assuming 2-DoF agents attracted our attention. So, considering our robotic case, we added some modifications helping the swarm of robots to move toward the desired direction and turn about an arbitrary rotation center point. In addition, a new criterion for measuring the alignment of the swarm is defined based on the desired direction of motion.

utilizing TCACS optimization algorithm, we tuned the parameter of AES model for a specific setup containing combination of linear and rotational motion of swarm. The effect of optimization on the behavior of CM, such as increasing

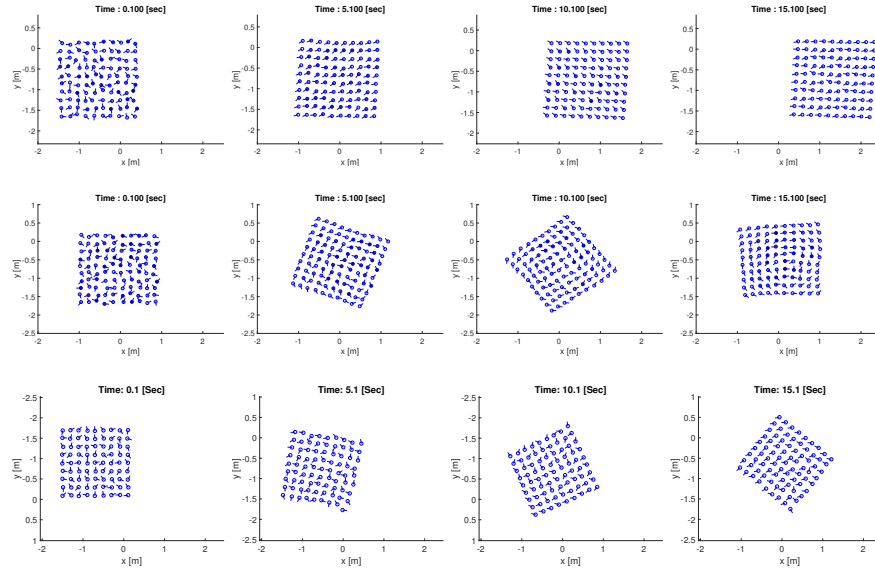


Fig. 5: Simulation Results: First row: Pure linear motion in Matlab, Second row: Pure rotational motion in Matlab, Third row: Linear+rotational motion in Webots

the rate of convergence and decreasing the total force, has been shown. Various weighting strategies for optimization has been studied, too.

Finally, three different scenarios have been conducted in Matlab and Webots. For this end, we have designed a CAD model of Mona robot in Webots and have simulated a supervised program to control agents. The results proved that our modifications are applicable and the optimized parameters improved the performance of the AES mechanism.

References

1. Arboleda-Estudillo, Y., Krieg, M., Stühmer, J., Licata, N.A., Muller, D.J., Heisenberg, C.P.: Movement directionality in collective migration of germ layer progenitors. *Current Biology* **20**(2), 161–169 (2010)
2. Arvin, F., Espinosa, J., Bird, B., West, A., Watson, S., Lennox, B.: Mona: an affordable open-source mobile robot for education and research. *Journal of Intelligent & Robotic Systems* pp. 1–15 (2018)
3. Arvin, F., Turgut, A.E., Krajník, T., Yue, S.: Investigation of cue-based aggregation in static and dynamic environments with a mobile robot swarm. *Adaptive Behavior* **24**(2), 102–118 (2016)
4. Bajec, I.L., Heppner, F.H.: Organized flight in birds. *Animal Behaviour* **78**(4), 777–789 (2009)

5. Ballerini, M., Cabibbo, N., Candelier, R., Cavagna, A., Cisbani, E., Giardina, I., Lecomte, V., Orlandi, A., Parisi, G., Procaccini, A., et al.: Interaction ruling animal collective behavior depends on topological rather than metric distance: Evidence from a field study. *Proceedings of the national academy of sciences* **105**(4), 1232–1237 (2008)
6. Couzin, I.D., Krause, J., James, R., Ruxton, G.D., Franks, N.R.: Collective memory and spatial sorting in animal groups. *Journal of theoretical biology* **218**(1), 1–11 (2002)
7. Ferrante, E., Turgut, A.E., Dorigo, M., Huepe, C.: Collective motion dynamics of active solids and active crystals. *New Journal of Physics* **15**(9), 095011 (2013)
8. Grégoire, G., Chaté, H.: Onset of collective and cohesive motion. *Physical review letters* **92**(2), 025702 (2004)
9. Grossman, D., Aranson, I., Jacob, E.B.: Emergence of agent swarm migration and vortex formation through inelastic collisions. *New Journal of Physics* **10**(2), 023036 (2008)
10. Karimi, A., Nobahari, H., Siarry, P.: Continuous ant colony system and tabu search algorithms hybridized for global minimization of continuous multi-minima functions. *Computational Optimization and Applications* **45**(3), 639–661 (2010)
11. Kudrolli, A., Lumay, G., Volfson, D., Tsimring, L.S.: Swarming and swirling in self-propelled polar granular rods. *Physical review letters* **100**(5), 058001 (2008)
12. Lewandowski, J.R., Sein, J., Blackledge, M., Emsley, L.: Anisotropic collective motion contributes to nuclear spin relaxation in crystalline proteins. *Journal of the American Chemical Society* **132**(4), 1246–1248 (2009)
13. Menzel, A.M., Ohta, T.: Soft deformable self-propelled particles. *EPL (Europhysics Letters)* **99**(5), 58001 (2012)
14. Michel, O.: Cyberbotics ltd. webots: professional mobile robot simulation. *International Journal of Advanced Robotic Systems* **1**(1), 5 (2004)
15. Moussaïd, M., Helbing, D., Theraulaz, G.: How simple rules determine pedestrian behavior and crowd disasters. *Proceedings of the National Academy of Sciences* **108**(17), 6884–6888 (2011)
16. Pourtakdoust, S.H., Nobahari, H.: An extension of ant colony system to continuous optimization problems. In: *International Workshop on Ant Colony Optimization and Swarm Intelligence*. pp. 294–301. Springer (2004)
17. Romanczuk, P., Couzin, I.D., Schimansky-Geier, L.: Collective motion due to individual escape and pursuit response. *Physical Review Letters* **102**(1), 010602 (2009)
18. Schaller, V., Weber, C., Semmrich, C., Frey, E., Bausch, A.R.: Polar patterns of driven filaments. *Nature* **467**(7311), 73 (2010)
19. Siarry, P., Berthiau, G.: Fitting of tabu search to optimize functions of continuous variables. *International journal for numerical methods in engineering* **40**(13), 2449–2457 (1997)
20. Sokolov, A., Aranson, I.S., Kessler, J.O., Goldstein, R.E.: Concentration dependence of the collective dynamics of swimming bacteria. *Physical review letters* **98**(15), 158102 (2007)
21. Suematsu, N.J., Nakata, S., Awazu, A., Nishimori, H.: Collective behavior of inanimate boats. *Physical Review E* **81**(5), 056210 (2010)
22. Vicsek, T.: *Fluctuations and scaling in biology*. Oxford University Press New York (2001)
23. Vicsek, T., Zafeiris, A.: Collective motion. *Physics reports* **517**(3-4), 71–140 (2012)
24. Ward, A.J., Sumpter, D.J., Couzin, I.D., Hart, P.J., Krause, J.: Quorum decision-making facilitates information transfer in fish shoals. *Proceedings of the National Academy of Sciences* **105**(19), 6948–6953 (2008)