Rendezvous Consensus Algorithm Applied to the Location of Possible Victims in Disaster Zones

José León[1,5], Gustavo A. Cardona[2], Luis G. Jaimes[3], Juan M. Calderón[1,4(✉)], and Pablo Ospina Rodriguez[1]

[1] Department of Electronic Engineering, Universidad Santo Tomás, Bogotá, Colombia
{joseleonl,juancalderon,pabloospina}@usantotomas.edu.co
[2] Department of Electrical and Electronics Engineering, Universidad Nacional de Colombia, Bogotá, Colombia
gacardonac@unal.edu.co
[3] Florida Polytechnic University, Lakeland, FL, USA
ljaimes@floridapoly.edu
[4] Bethun-Cookman University, Daytona Beach, FL, USA
calderonj@cookman.edu
[5] Department of Electronic and Telecommunication Engineering, Universidad Católica, Bogotá, Colombia
jleon@ucatolica.edu.co

Abstract. In this paper is presented an alternative to performing the analysis of the sensors in the field of applied cooperative robotics for search and location of disaster victims. This work proposes the use of the Rendezvous algorithm to validate the information coming from the sensors of a multi-robot system. The sensors located in each one of the robotic agents provide a measured value according to the existence or not of victims in the surrounding zone to the robot. Since the information coming from the robots is not the same, however, its belong to the same sensed parameters, the Rendezvous algorithm is used to find a consensus of opinion about the existence of victims. In addition, the swarm of robots uses bio-inspired techniques to generate the navigation algorithm. This navigation algorithm was inspired by the foraging technique used by swarms such as bees, birds or bacteria. The results present some Rendezvous algorithm simulation and robot swarm navigation showing the feasibility of the proposed system.

Keywords: Robot swarm · Rendezvous algorithm
Robot navigation · Bio-inspired systems

1 Introduction

The agencies for the attention of emergencies begin their tasks of locating and recovering victims moments after the occurrence of a disaster. These tasks are

L. Rutkowski et al. (Eds.): ICAISC 2018, LNAI 10842, pp. 700–710, 2018.
https://doi.org/10.1007/978-3-319-91262-2_61

stressful and their development takes hours or even days depending on the magnitude of the disaster. Time is a key factor to avoid loss of life since it is extremely urgent to provide the required medical assistance, but fatigue, instability of the land, climatic conditions and lighting problems make these tasks more complicated.

The agencies for the attention of emergencies begin their tasks of locating and recovering victims moments after the occurrence of a disaster. These tasks are stressful and their development takes hours or even days depending on the magnitude of the disaster. Time is a key factor to avoid loss of life since it is extremely urgent to provide the required medical assistance, but fatigue, instability of the land, climatic conditions and lighting problems make these tasks more complicated.

Mobile robotics in search and rescue applications have contributed in different fields such as recognition of terrain, lifting communications networks that allow the organization of rescue teams, and telemedicine among others. Even sometimes the robotic platform is used to provide instructions about medical procedures to victims while Emergency teams arrive to perform rescue actions.

Cooperative robotics is a viable alternative in search and rescue applications since they offer great advantages, such as the robots perform exploration and reconnaissance tasks in less time without risking the lives of the rescue team. If a robot is lost or stuck, others continue to explore the disaster area as depicted by Yanguas-Rojas [1]. The information provided by the robots is validated with sensor fusion techniques since several robots perform the sensing of common areas, allowing to reach a consensus of the information captured individually. In this way, these systems try to avoid failures due to noise or damage of the sensors, allowing these systems to be more reliable, robust, and precise.

2 Related Work

This section presents some applications of consensus algorithms in cooperative control. In particular, swarm robotics in which the problem of communications in wireless sensor networks is still an open issue. In addition, we present some of the most popular consensus algorithms for robust interaction among agents under different operational conditions.

In terms of position-based consensus, the work of Song *et al.* [2] presents a consensus algorithm with double-integrator dynamics and directed topology. Two types of distributed observer algorithms are proposed to solve the consensus problem by utilizing continuous and intermittent position measurements, respectively. Another research in this area corresponds to the work of [3]. Here the authors present a control scheme capable of solving the leader-follower and the leaderless consensus problems in networks composed of multiple Euler-Lagrange systems.

Another communication issue that raises when applying consensus algorithms on multi-point networks is the communication limitation due to interferences such as static and electric noise, and fluctuations in the bandwidth. In that sense, the work of Chen and Ho [4] tackled this problem in the context consensus for Autonomous Underwater Vehicles (AUVs), where transmission faults are common due to the underwater conditions. His work presents two new consensus techniques which work on the presence of transmission faults with respect to leaderless multiple AUV systems and leader-follower multiple AUV systems.

The use of consensus algorithms raises new challenges on wireless sensor networks. Some of these include energy consumption, unification of sensor readings, handling missing sensor reading values, etc. Research in this area includes the work of Chelbi [5] who focus on energy consumption optimization and the unification of sensor readings from multiple sensors. Here, the consensus algorithm allows obtaining a unique value from a set of different sources while minimizing the effect of noise and corrupt measurements. Here, the authors propose a routing algorithm based on clusters that increase the surviving time of the network. In a similar work Nurellari *et al.* [6] applies consensus algorithm for intrusion detection, are the detection task is distributed through the different nodes of the wireless sensor network. In a first step, the information is disseminated through different channels, the algorithm takes into account the channel capacity. In a second step, the algorithm gathers a global consensus.

3 Performance Evaluation

In order to test the performance of our proposed algorithm, we model an scenario in which a drone swarm is looking for a set of targets on the ground. This could be the case of a rescue mission in which a drone swarm is looking for one or several potential victims on the ground. Thus, we model the target (i.e., victims) and obstacles by using a multi-variable Gaussian distribution. The targets o victims have an attractor effect on the agents which are within a given distance, and obstacles have the opposite effect, namely and a repulsion effect which is used to avoid potential collisions. This surface is defined by $J(x)$, where $x \in \Re^n$, and $J(x)$ is continuous with a finite slope at every point in its domain (i.e., differentiable). Thus, the agents moves toward the negative gradient of $J(x)$ (see Eq. 1) [7,8].

$$- \nabla J(x) = -\frac{\partial J}{\partial x} \tag{1}$$

3.1 Navigation

Even though a swarm of drones seems to be crowded, each member is able to navigate without the interference of any other swarm's member. Thus, avoiding collisions, and keeping the direction and orientation by maintaining always a safe distance from each other. What makes this possible are the constants of attraction and repulsion of the swarm's navigation algorithm. Thus, the attraction constant ensures the swarm stays grouped and moves in the same direction as shown in Eq. 2, while the repulsion constant allows to keep a safe distance among agents in order to avoid collision, see Eq. 3.

$$- k_a(x^i - x^j) \tag{2}$$

Where k_a is the attraction force, this mechanism could be local (e.g., a range in the range of sensing) o global (e.g., an agent can move other group members regardless how far they are each other).

$$k_r exp(\frac{-\frac{1}{2}\|x^i - x^j\|^2}{r_s^2})(x^i - x^j) \tag{3}$$

Where $k_r > 0$, represents the magnitude of repulsion and $r_s > 0$, is the range of repulsion.

3.2 Sub-swarms

Each agent is exposed to different attraction and repulsion forces which help to keep the swarm's cohesion and navigating in the same direction.

The targets generate an attraction force on the closer agents, and thus, this force may modify the normal swarm's behavior by reducing the speed of some agents, namely those closer to that target. Thus, the subset of agents within the range of attraction to the target is separated from the main swarm. This happens because the distance between the main group and those left behind is every time greater, which result in losing communication and breaking the attraction forces that used to keep them together.

Thus, those left behind and with close proximity to the target form a new swarm, this new swarm changes its state to still and send the target's coordinates to the rescue station.

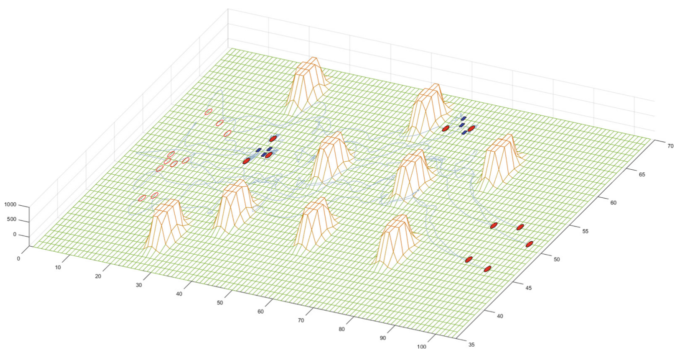


Fig. 1. Sub-swarm [8] (Color figure online)

As can be seen in Fig. 1 the targets are colored in blue, and the swarm agents are colored with red. Around the targets there is a group of agents, those are the ones which were left behind from the main swarm and thus, forming a new small swarm. The members of this small swarm use their sensor readings and the consensus algorithm to determine if there is actually a real target to rescue, and then they proceed to send the target coordinates.

3.3 Sensors

Sensor readings are simulated by using a random normal distribution plus some white noise. These noisy readings are then evaluated by the objective function which determines whether the agent is detecting a potential target. Figure 2 visualize the sensor readings in the target area with five potential targets.

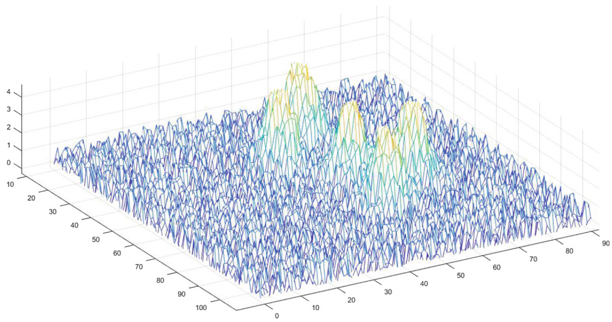


Fig. 2. Sensors

3.4 Consensus

Consensus enables a collective intelligence which allows the swarm to take decisions based on the agents' individual opinions, here, we use consensus among the members of the small swarm, namely the one surrounding a potential target to determine whether or not there is an actual target there.

The algorithm used to reach consensus is [11] which the one that solves the Rendezvous problem which consists in make that several agents meet in given point without knowing its location.

We use Eq. 4 which computes the distance from agent i to the other swarm's members. Thus, the meeting point would be the center of gravity of the swarm [9].

$$\dot{x_i} = \sum_{j \in N_i} (x_j - x_i) \tag{4}$$

For a swarm with many members iterate on Eq. 4 is inefficient, instead we vectorized the computation as show in Eq. 5. This vector is then used as input of Eq. 6 which outputs the swarm's consensus [9].

$$x = \begin{bmatrix} x_1 \\ . \\ . \\ . \\ x_N \end{bmatrix} \tag{5}$$

Here, in Eq. 6 $\dot{x}$ corresponds to rate of change of data, and L to the Laplacian matrix, where is represented the network of the swarm interaction.

$$\dot{x} = -Lx \tag{6}$$

The Laplacian matrix can be found by using Eq. 7 where D is a diagonal matrix as can be seen in Eq. 8 with $d_{in}(k) = \sum_j Adj_{kj}$ and matrix $Adj_{i,j}$ represents the communication network among the swarm agents. Thus, if a member agent i has communication with agent j, the entry (i, j) is one, and zero otherwise [9].

$$L = D - Adj \tag{7}$$

$$D = \begin{bmatrix} d_{in}(1) & & & \\ & \cdot & & \\ & & \cdot & \\ & & & d_{in}(k) \end{bmatrix} \tag{8}$$

Thus, the updated new value which results from the consensus algorithm is stored in x as can be seen in Eq. 9.

$$x = k * \dot{x} + x \tag{9}$$

Here, k is convergence factor. If k is high, we might never reach a consensus given hight level of oscillation. On another hand, if k is too small the algorithm may take too much time to converge. Thus, k is one of the most important parameters of the algorithm.

4 Experiment and Results

In this experiment, the sub-swarm (i.e., the one which identified a potential target) is composed by three agents, sensor readings are in the range of 0, 5. Values below a given threshold, in this case 2.5 indicates that there is not target, and values in between 2.5 and 5 will indicate the presence of a target. Thus, the value of sensor 1, 2, and 3 are 2, 2.8, 1 respectively. See Eq. 10.

$$x = \begin{bmatrix} 2 \\ 2.8 \\ 1 \end{bmatrix} \tag{10}$$

Matrix Adj represents the communication network among the agents, because the sub-swarms are in state still (i.e., no movement) after located the potential target, this will allow members of the sub swarm to communicate with other members of the same sub-swarm, but not with itself. This new definition of communication gives place to the matrix Adj which is define in Eq. 11.

$$Adj = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix} \tag{11}$$

Here, we run the consensus algorithm with 20 iterations, and using a convergence constant $k = 0.1$, the consensus is found after 17 iterations with a final result of 1.93, which indicates that there is not a target in that location. See Fig. 3.

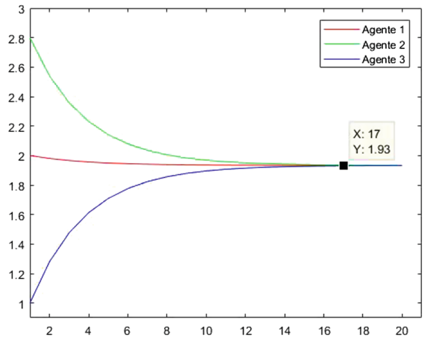


Fig. 3. Test 1

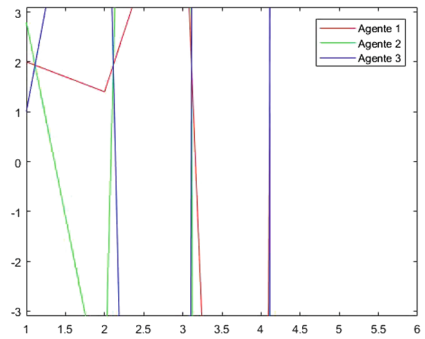


Fig. 4. Test 2

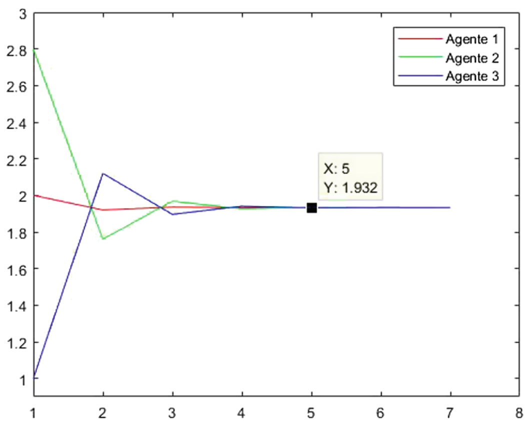
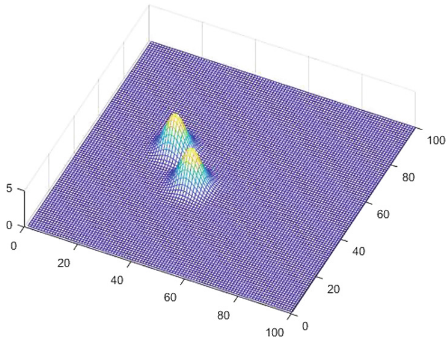
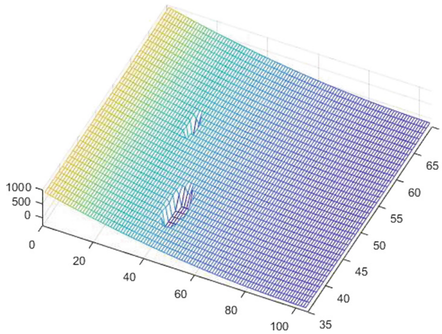


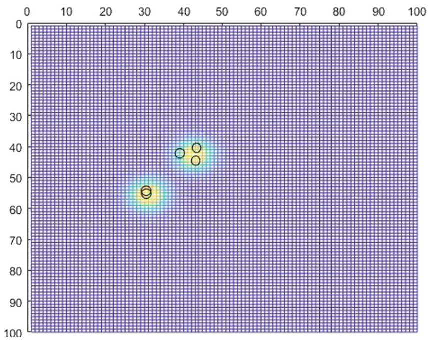
Fig. 5. Test 3



(a)



(b)

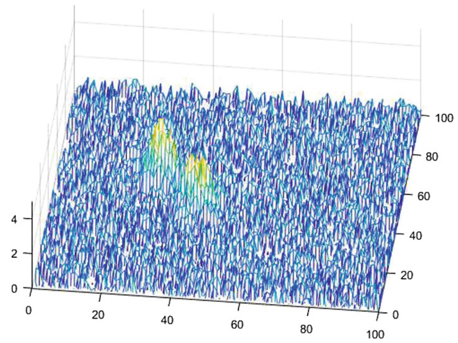


(c)

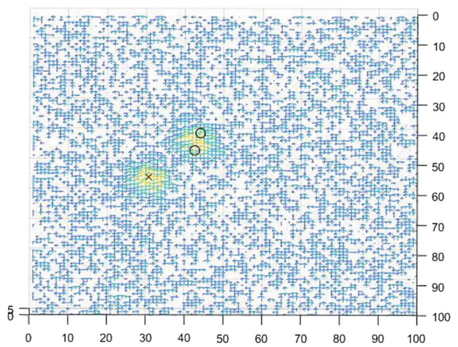
Fig. 6. Test 4

We carried out a second test using the same parameters that the used in first test, but this time using a convergence constant of $k = 3$, using just 5 iteration. Figure 4 shows that the algorithm didn't converge given the value of k and the few number of iterations. A third test was carried out using the same scenario that test 1, and test 2. But this time using a convergence constant of $k = 0.4$, and 7 iterations as shown in Fig. 5. Here, the algorithm found a consensus after iteration 5.

In order to test the performance of the consensus, and navigation algorithms together, we carry out two tests. In the fist test, we simulated an environment with two targets (see Fig. 6b). Here, we did not introduce noise in the sensor readings (Fig. 6a. Around each target we placed a sub-swarm, the first one with three, and two agents respectively. The consensus results in both cases were positive (see Fig. 6c). Here, a positive identification is represented by 0 and a negative with X.



(a)



(b)

Fig. 7. Test 5

In order to evaluate the robustness of system we carried out a second test, this time we added noise to the sensor readings (see Fig. 7a). The consensus in the swarm with less agents resulted negative, while the consensus in sub-swarm with more agents resulted positive (see Fig. 7b).

5 Conclusions and Future Work

This paper presents and consensus algorithm for the identification of targets in the context of swarm robotics. In particular, we present an application of target identification in which a swarm of drones is scanning a target area, looking for potential victims of a disaster, or targets. While the main swarm is moving in a coordinate way scanning the area, a subset of agents whose sensor reading seems to identify a potential target starts to slow down its speed and surrounding that target. This behavior leads to a separation from the main swarm, which results in a loose of communications and a break of the attraction forces that maintain the swarm agents together and navigating in a coordinated way. We carried out a set of experiments to test the ability of the new sub-swarms for the identification of the actual targets. In the first set of the experiment without noise in the sensor readings, all the identification test resulted positive.

We carried out a second set of the experiment using the same set of parameters, but this time adding noise to the sensor reading. Here, parameters such as the number agents in the sub-swarm, the value of convergence constant k, and the number of interactions of the optimization algorithm played a key role in the success of the identification task. In particular, for high values of k the algorithm did not reach a consensus, and for small values of k the algorithm needed a high number of iterations.

Thus, as a future work, we plan to find the range of the parameters that maximize the effectiveness of the combination navigation and target identification. These parameters include the number of agents in the sub-swarm, value of k, the minimum number of iterations before reaching convergence. Another factor might include the relationship between the target area and the minimum number of agents needed per swarm in order to reach a consensus given some level of noise in the agents' sensor readings.

References

1. Yanguas-Rojas, D., Cardona, G.A., Ramirez-Rugeles, J., Mojica-Nava, E.: Victims search, identification, and evacuation with heterogeneous robot networks for search and rescue. In: 2017 IEEE 3rd Colombian Conference on Automatic Control (CCAC), Cartagena, pp. 1–6 (2017). https://doi.org/10.1109/CCAC.2017.8276486
2. Song, Q., Liu, F., Wen, G., Cao, J., Yang, X.: Distributed position-based consensus of second-order multiagent systems with continuous & intermittent communication. IEEE Trans. Cybern. **47**(8), 1860–1871 (2017)
3. Nuno, E.: Consensus of Euler-Lagrange systems using only position measurements. IEEE Trans. Control Netw. Syst. **PP**(99), 1 (2016)

4. Chen, S., Ho, D.W.C.: Consensus control for multiple AUVs under imperfect information caused by communication faults. Inf. Sci. **370**(Suppl. C), 565–577 (2016)
5. Chelbi, S., Duvallet, C., Abdouli, M., Bouaziz, R.: Event-driven wireless sensor networks based on consensus. In: 2016 IEEE/ACS 13th International Conference of Computer Systems and Applications (AICCSA), pp. 1–6 (2016)
6. Nurellari, E., McLernon, D., Ghogho, M.: Distributed two-step quantized fusion rules via consensus algorithm for distributed detection in wireless sensor networks. IEEE Trans. Signal Inf. Process. Netw. **2**(3), 321–335 (2016)
7. Passino, K.M.: Biomimicry for Optimization, Control, and Automation. Springer Science & Business Media, Heidelberg (2004). https://doi.org/10.1007/b138169
8. León, J., Cardona, G.A., Botello, A., Calderón, J.M.: Robot swarms theory applicable to seek and rescue operation. In: Madureira, A., Abraham, A., Gamboa, D., Novais, P. (eds.) ISDA 2016, pp. 1061–1070. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-53480-0_104
9. Mesbahi, M., Egerstedt, M.: Graph Theoretic Methods in Multiagent Networks. Princeton University Press, Princeton (2010)
10. Priolo, A., Gasparri, A., Montijano, E., Sagues, C.: A decentralized algorithm for balancing a strongly connected weighted digraph. In: American Control Conference, pp. 6547–6552 (2013)
11. Lin, J., Morse, A.S., Anderson, B.D.O.: The multi-agent rendezvous problem. Part 2: the asynchronous case. SIAM J. Control Optim. **46**(6), 2120–2147 (2007)