

PAPER • OPEN ACCESS

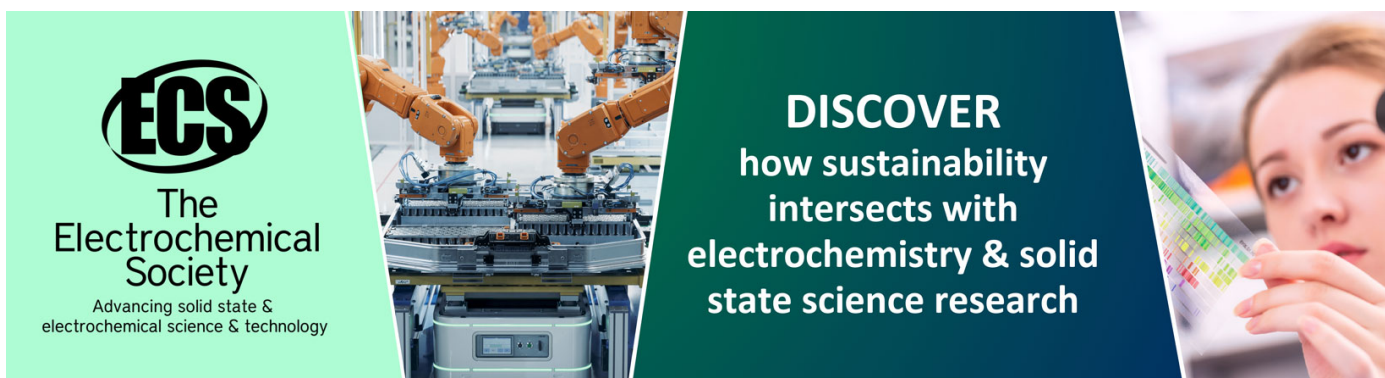
Particle swarm optimization method for the control of a fleet of Unmanned Aerial Vehicles

To cite this article: A Belkadi *et al* 2015 *J. Phys.: Conf. Ser.* **659** 012015

View the [article online](#) for updates and enhancements.

You may also like

- [Application and Typical Case of Unmanned Aerial Vehicles Combined with Building 3D Modeling](#)
Minhui Li, Hang Yin, Dongji Cheng et al.
- [A generic approach for photogrammetric survey using a six-rotor unmanned aerial vehicle](#)
K N Tahar, A Ahmad, W A A W M Akib et al.
- [The Adoption of Unmanned Aerial Vehicles \(UAV\) Technology in the Construction Industry: Construction Stakeholders' Perception](#)
Kong Foo Keong, Chai Chang Sa'ar, Suaathi Kaliannan et al.



ECS
The
Electrochemical
Society
Advancing solid state &
electrochemical science & technology

DISCOVER
how sustainability
intersects with
electrochemistry & solid
state science research

Particle swarm optimization method for the control of a fleet of Unmanned Aerial Vehicles*

A. Belkadi^{1,2}, L. Ciarletta^{3,4}, and D. Theilliol^{1,2}

* Research supported by Conseil Régional de Lorraine, France.

¹ Centre de Recherche en Automatique de Nancy (CRAN), Université de Lorraine, CNRS, 2 avenue de la forêt de Haye F-54516, Vandoeuvre-les-Nancy.

² CNRS, CRAN, UMR 7039, France.

³ Permanent member the Madynes research team of Loria, UMR 7503, France.

⁴ école des Mines Nancy, Université de Lorraine.

E-mail : adel.belkadi@univ-lorraine.fr, laurent.ciarletta@loria.fr,
didier.theilliol@univ-lorraine.fr

Abstract. This paper concerns a control approach of a fleet of Unmanned Aerial Vehicles (UAV) based on virtual leader. Among others, optimization methods are used to develop the virtual leader control approach, particularly the particle swarm optimization method (PSO). The goal is to find optimal positions at each instant of each UAV to guarantee the best performance of a given task by minimizing a predefined objective function. The UAVs are able to organize themselves on a 2D plane in a predefined architecture, following a mission led by a virtual leader and simultaneously avoiding collisions between various vehicles of the group. The global proposed method is independent from the model or the control of a particular UAV. The method is tested in simulation on a group of UAVs whose model is treated as a double integrator. Test results for the different cases are presented.

Keywords : PSO algorithm, Fleet control, Virtual Leader, generating path.

1. Introduction

Recently, various applications of UAVs (Unmanned Aerial Vehicles) are emerging in the military, but also in the civilians areas. Using multiple vehicles fleet can cover large areas of research, surveillance[1], inspection[2] etc .. Recently, the attention of some scientists turned to imitate the behavior of birds in a fleet in order to make an autonomous flying formation [3][4].

Some solutions use leader, and in these approaches, there are various flight training control strategies : Leader-follower[5][6] (hierarchical approach), Virtual Leader [7][8][9] and control based on behavior [10][11] (decentralized approach). The method considered in our approach consists in replacing the head of the formation of the leader-follower approach by a virtual leader. All training entities receive the path of the mission which is considered as the virtual leader itself. A major disadvantage of the classical leader-follower strategy is that the risk of collision between agents increases[6], because



there is no return (feedback) training. In our approach, each agent also receives (and uses) information from its neighbors.

In this article, the objective is to develop a fleet control strategy for a group of UAVs capable of self-organizing and covering a surface in a predefined architecture/topology or to fly in formation, following a mission led by a virtual leader while ensuring collision avoidance between team members. In our case, we postulate that the gathering fleet is a reference generation problem rather than a control design problem. It is formulated as an optimization problem in order to find for each agent of the fleet reference to follow the trajectory at each instant t which minimizes a predefined objective function. Each UAV is assumed to be further controlled by a control law and it can measure the positions of all the agents in his neighborhood thanks to vision techniques and/or wireless communication capabilities. We consider in this paper, a simplified case of figure 2 without data loss communication delay or disruption.

This strategy was introduced in [12]. In our previous work, the objective function has been minimized in a simplified manner. The goal was to choose among a number of reachable points from a UAV, the position that minimized at best the objective function. In order to improve our approach we have chosen to use an optimization method that is more developed and responsive to the command of our fleet. This method is derived from meta-heuristics and is called Particle Swarm Optimization (PSO).

The control systems by Meta-heuristics, such as genetic algorithms[13], ant colony optimization algorithms[14] or by particle swarm optimization [15][16] have been the subject of several studies in the literature. Although genetic algorithms are very effective in finding the global minimum, their complexity and their execution time is their biggest disadvantage. The PSO algorithm has demonstrated very good performance by minimum search term with a minimum calculation time and relative ease of implementation [17]. These performances have oriented us to choose PSO

The paper is organized as follows. Section 2 is dedicated to the problem statement. Then, Section 3 is devoted to the optimization problem and the PSO algorithm description. In Section 4, the performance of the method based on various simulations has been illustrated. Finally, a conclusion and perspectives are presented.

2. Problem statement

The goal of our approach is to get a set of UAVs to fly on a 2D plane following a trajectory considered as a virtual leader while converging to a predefined spatial configuration (a flying pattern) and avoiding collisions.

Let us consider n mobile UAVs operating in space R^2 with $X_i(t)$ the robot position i at time $t > 0$. The system can be described by a weighted graph $G(N, \varepsilon(t), A(t))$ where $N = \{1, \dots, n\}$ and $\varepsilon(t) \in N \times N$ are the set of nodes and the set of arcs respectively of the graph G and $A(t)$ a matrix of $M_n(R)$ of elements in R as $a_{ij}(t) > 0$ if: $(j, i) \in \varepsilon(t)$. At each sample i is an incoming neighbor of j and j is an outgoing neighbor of i .

G is assumed to be an undirected graph such as: $\forall (i, j) \in \varepsilon(t) \Rightarrow (j, i) \in \varepsilon(t)$. In this case the matrix $A(t)$ is symmetrical with $a_{ij}(t) = a_{ji}(t)$.

Let us define for each i node a set of neighbors $\Xi_i(t)$ such that :

$$\Xi_i(t) = \{ j \in N : \|x_j(t) - x_i(t)\| \leq l \} \quad (1)$$

where l defines the scope of the neighborhood.

Let's now consider for simplification, a fully actuated UAV i with dynamic model given by :

$$\dot{X}_i = v_i \quad (2)$$

$$\dot{v}_i = u_i \quad (3)$$

where $X_i = [x_i, y_i]^T$ is the configuration of UAV i , u_i is its control input and v_i its velocity vector. The control law is given by :

$$u_i = \alpha(X_i^d - X_i) + \beta(\dot{X}_i^d - \dot{X}_i) + \ddot{X}_i^d \quad (4)$$

The objective is to achieve a formation flight based on a virtual leader P . All training UAVs receive the trajectory of the mission. This trajectory is that of the virtual leader. The control strategy considered is completely independent from the underlying conception of the individual robot control.

The main idea is to find at each time t a desired reference input $X_i^d \in R^2$ for each UAV i of the fleet based on the desired reference trajectory $P_d(t)$ and the λ positions of the servants in its field of vision $x_j(t)$ with $j \in \Xi_i(t)$ such that $\lim A(t) = A_d$ while ensuring the anti-collision of the UAV with A_d being the desired adjacency matrix. This constraint can be rewritten as :

$$\forall n_i, n_j \in N : \|x_j(t) - x_i(t)\| > c \quad (5)$$

where c is a constant safety distance.

This objective is achieved by minimizing the cost function $\Lambda_i(t)$ for each one of the team member i .

3. Methodology

3.1. Assumptions and Optimization Problem

Given a team of n UAVs operating in space R^2 . The entire system is described by a non-oriented weighted graph $G(N, \varepsilon(t), A(t))$, where $A(t)$ is a $n \times n$ matrix defined positive and symmetric, with $a_{i,j}(t)$ the distance between the node i and j at time t .

The neighbors of a robot i are defined in eq.(1) where $\Xi_i(t)$ is called the metrical neighborhood of the robot i . The control objective is to bring the fleet from an initial configuration $A(t_0)$ to a desired configuration A_d (i.e the desired formation). This problematic can be formulated as an optimization problem.

At first, let us define a cost function $\Lambda_i(t)$ for each UAV i such that:

$$\Lambda_i(t) = \rho \left(\| (P_d(t) - (x_i(t) + h)) \| - a_{ip}(t) \right) + \sum_{j=1}^q \alpha_{ij}(t) \left(\| (x_j(t) - (x_i(t) + h)) \| - a_{ij}(t) \right) \quad (6)$$

with $i \neq j$, $q = \text{card}(\Xi_i(t))$, $h \in R^2$, and $\rho \gg 1$

The main goal is to find the best vector h for each agent i minimizing the cost function $\Lambda_i(t)$ such that:

$$\forall i \in N, \lim_{t \rightarrow \infty} \Lambda_i(t) = 0 \Rightarrow \lim_{t \rightarrow \infty} A(t) = A_d \quad (7)$$

The reference trajectory at time $t + 1$ for the agent i will then be:

$$x_{id}(t + 1) = x_i(t) + h \quad (8)$$

The ρ coefficient in the cost function should be larger than 1. This choice is due to the fact that we favor each agent i position that tends to be in the direction of the virtual leader P affected by the reference trajectory $P_d(t)$.

During the evolution of the robots in R^2 space, non-collision of the agents should be ensured. This constraint, as shown in the previous section, can be written as follows: $\forall n_i \in N, \forall n_j \in h_i : \|x_j(t) - x_i(t)\| > c$, where $c \in R$ is the minimum distance between the two agents i and $j \in N$ do not cross. On matrix $A(t)$, this means that :

$$\forall a_{ij}(t) \in A(t) : a_{ij}(t) > c, \text{ with } i \neq j \quad (9)$$

In order to introduce this condition in the cost function $\Lambda_i(t)$, we define new functions $\alpha_{ij}(t)$ between each agent i and its neighbors j such that :

$$\alpha_{ij}(t) = 1 + \exp\left(\frac{c-a_{ij}(t)}{\sigma}\right) \quad (10)$$

The value of the $\alpha_{ij}(t)$ function depends on the difference between $a_{ij}(t)$ the distance between two Robots i and j with the safety distance c , it is all the greater when $a_{ij}(t) < c$ and converges to the value 1 when $a_{ij}(t) \gg c$.

So, each agent i is more likely to favor the values of h which avoids the need for the distances $a_{ij}(t) < c$ and minimizes at best the cost function $\Lambda_i(t)$, thanks to the PSO algorithm.

3.2. PSO Algorithm

The particle swarm optimization method appeared in 1995 in the U.S.A and was designed by Eberhart and Russel James Kennedy [20]. It is based on the notion of cooperation between agents called "particles", which manage to solve complex problems by exchanging information.

The PSO algorithm is initialized by a random population of potential solutions $x(t)$, interpreted as particles moving in the search space where $V(t)$ represents their speed. Each particle is attracted to its best position discovered in the past noted P_1 as well as to the best position of the particles discovered by its neighborhood P_2 .

The algorithm includes several setting parameters to act on the compromise Exploration - Exploitation. Exploration is the ability to test various regions of space for "looking good" candidate solutions. The Exploitation is the ability to focus research around promising solutions to get as close as possible to the optimum.

The classic PSO algorithm is written as follows [18] :

$$V(t+1) = a V(t) + b_1 r_1 (P_1(t) - x(t)) + b_2 r_2 (P_2(t) - x(t)) \quad (11)$$

$$x(t+1) = x(t) + V(t+1) \quad (12)$$

Where, a is the coefficient of inertia, b_1 and b_2 two real representing the intensity of attraction and r_1, r_2 two random values between 0 and 1.

To simplify the study, we consider the deterministic version of the algorithm, where random numbers are replaced by their average values $1/2$.

With simplifications, the algorithm can be written as:

$$V(t+1) = a V(t) + b (P(t) - x(t)) \quad (13)$$

$$x(t+1) = x(t) + V(t+1) \quad (14)$$

Where :

$$P(t) = \frac{b_1 P_1(t) + b_2 P_2(t)}{b_1 + b_2} \quad \text{and} \quad b = \frac{b_1 + b_2}{2}$$

To make a dynamic analysis of the algorithm, the equations (13) and (14) are rewritten in the matrix form :

$$V(t+1) = -b x(t) + a V(t) + b P(t) \quad (15)$$

$$x(t+1) = x(t) + [-b x(t) + a V(t) + b P(t)] \quad (16)$$

Then :

$$V(t+1) = -bx(t) + aV(t) + bP(t) \quad (17)$$

$$x(t+1) = (1-b)x(t) + aV(t) + bP(t) \quad (18)$$

The equations of the algorithm can then be written in the following matrix form :

$$\begin{pmatrix} x(t+1) \\ V(t+1) \end{pmatrix} = A \begin{pmatrix} x(t) \\ V(t) \end{pmatrix} + B P(t) \quad (19)$$

Where : $\begin{pmatrix} x(t) \\ V(t) \end{pmatrix}$ is the state of the system, consisting of the position of the particle and its velocity, P the system input, $A = \begin{bmatrix} 1-b & a \\ -b & a \end{bmatrix}$ is the dynamical matrix, and $B = \begin{bmatrix} b \\ b \end{bmatrix}$ the input matrix.

The equilibrium point of the system is such that the particle is positioned in P " $x(t) = P$ " and has a zero speed $V(t) = 0$.

Behaviors of the particle depend on the eigenvalues of the matrix A are the solutions of : $\det(\lambda I - A) = 0$, i.e :

$$\lambda^2 - (a-b+1)\lambda + a = 0 \quad (20)$$

The behavior and the convergence of the algorithm depend on the parameters a and b . The analysis of the equation (20) leads to determine the area of convergence of the PSO according to their values.

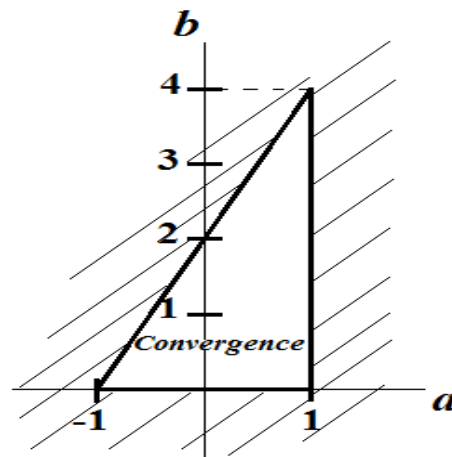


Figure 1. PSO convergence area

Figure 1 shows the values (plain area) that should have the parameters a and b for a convergence of the algorithm [19].

4. Simulation Results

The proposed method for the control of the fleet has been evaluated by the simulation of a group of 6 UAVs. The trajectory of the fictitious agent is defined initially in a fixed position, and secondly in motion. The results are compared with different initial conditions (positions of UAVs).

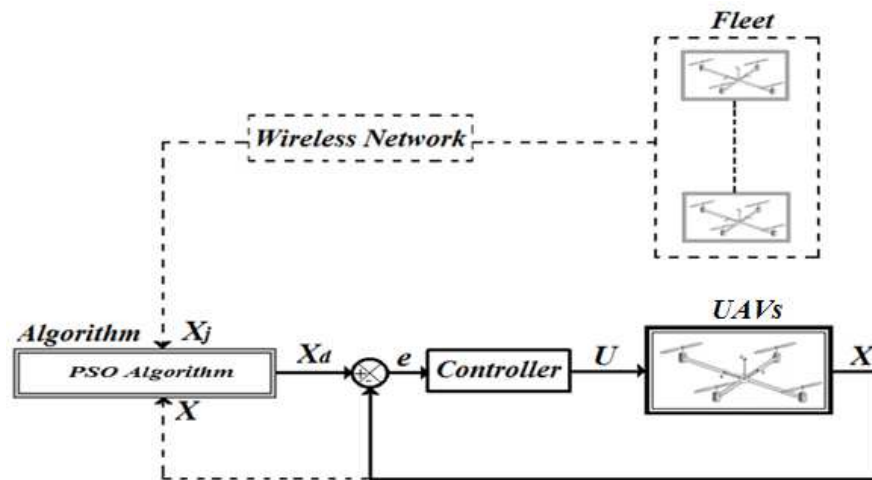


Figure 2. Control scheme of the fleet of UAVs

4.1. Stationary Leader Case

The constant sampling time is set to 10^{-2} seconds. The leader's position is stationary. The desired configuration for the UAVs is set from the desired A_d adjacency matrix.

In this case, the form of a circle is given as the desired configuration and we simulate the behavior of the fleet for different initial conditions.

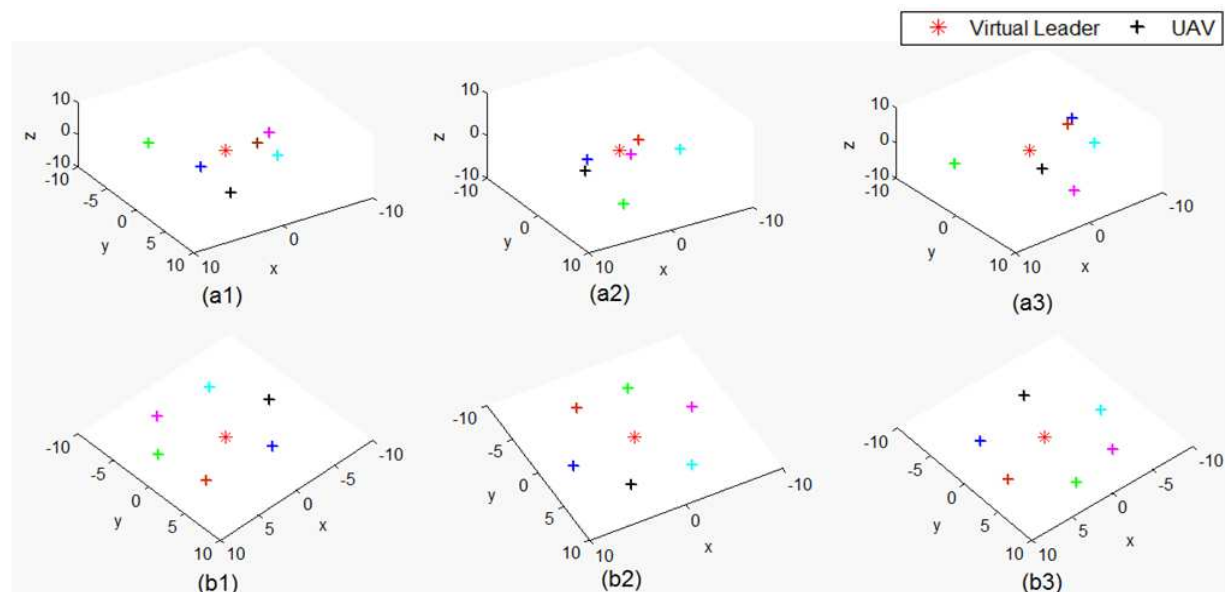


Figure 3. Circle Formation of the UAV Fleet with Stationary Virtual Leader

Figure 3 shows the results obtained b1, b2, b3 with respectively the initial positions a1, a2 and a3, the UAVs fleet is able to reproduce the desired configuration i.e. the shape of a circle with the virtual leader at its center.

To evaluate the evolution of the distances between the UAVs and ensure that there is no collision between the flying robots, changes of the cost function $\Lambda_i(t)$ of each UAV with the initial positions (a1) is plotted in figure 4.

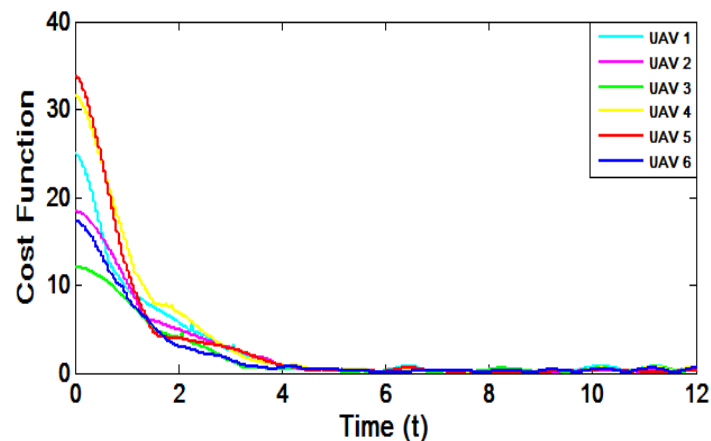


Figure 4. The variation of the cost function for Each UAVs with Stationnary Virtual Leader

The cost function decreases more and more as the fleet converges towards the desired configuration. A collision between two UAVs would appear on the cost function as a great increase in its value according to equation (10). As presented in Figure 4, it should be noticed that the safety distance and scope of the neighborhood is respected throughout the simulation. The performance of the PSO algorithm are good and the system converges after 4s. The simulations are repeated 100 times with different initial conditions. We have considered that the initial distance between two UAVs is greater than the safety distance. The results are equivalent to those obtained for the case a1 with average convergence time equal to 4s.

4.2. Mobile Leader Case

In the second part of the simulation, the leader is now in motion following a straight trajectory.

As in the previous example, various initial conditions a1, a2 and a3 are considered and a form of a circle is given as the desired configuration. The results obtained with the initial conditions a1 are illustrated in the following graphs (Figures 5 to 7).

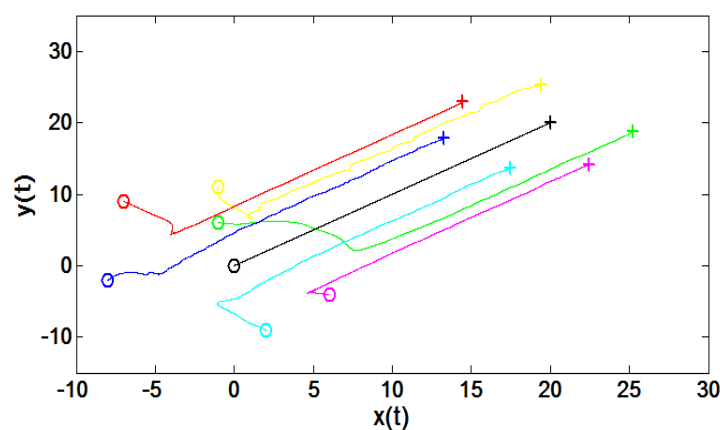


Figure 5. Circle Formation of the UAV Fleet with Mobile Virtual Leader

The figure 5 shows the evolution of the fleet on a 2D space. The desired configuration is a circle as in the previous example. The simulation demonstrates clearly that the fleet converges to the desired configuration while following the trajectory of the virtual leader.

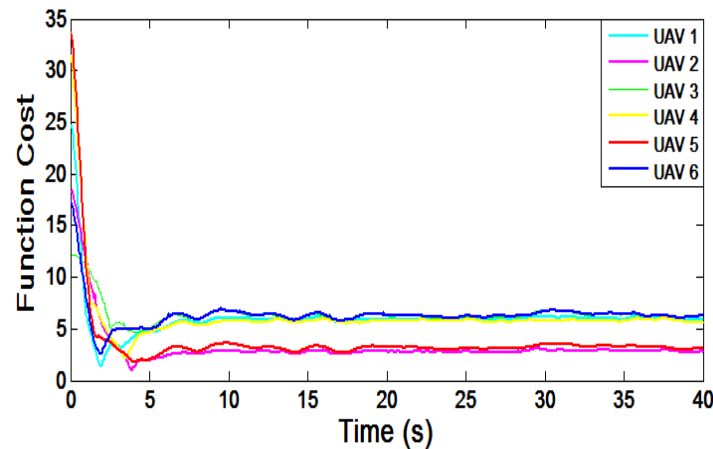


Figure 6. The variation of the cost function for Each UAVs with Mobile Virtual Leader

Figure 6 shows the performance of the PSO algorithm in the case of a moving leader. The convergence speed is reasonable and in the order of 5s. The simulations are also repeated 100 times with different initial conditions and the results are equivalent to the case illustrated in the above example. The approach considered in our simulations is able to reproduce without difficulty geometric configurations and flight training while ensuring the constraint of the anti-collision.

5. Conclusion

The paper presents a fleet control strategy based on a multi-agent system with a virtual leader. The use of the PSO algorithm that is part of optimization algorithms family called Meta-heuristic allowed us to have better results in terms of convergence speed and optimal solution.

This approach has been tested in simulation on a group of UAVs simply modeled as double integrators for our study. In our simulations, both the cases of the virtual leader standing and in motion has been considered.

The results clearly show that the proposed method is effective to cover the surface and the fleet command. The records obtained also show a robust approach against collisions, which was the main drawback of control by the virtual leader in the literature.

A future application of control strategy on non-linear complex systems and implementations on real systems are planned. An improvement would be to introduce the dynamics of controlled systems in the optimization algorithms to better find the most appropriate one for the studied systems solutions.

References

- [1] CHAIMOWICZ, Luiz and KUMAR, Vijay. Aerial shepherds: Coordination among uavs and swarms of robots. In : Distributed Autonomous Robotic Systems 6. Springer Japan, 2007. p. 243-252.
- [2] HERNANDEZ, Andres, COPOT, Cosmin, CERQUERA, Juan, and al. Formation Control of UGVs using an UAV as Remote Vision Sensor*. 2014, Cape Town, South Africa.
- [3] HAFEZ, Ahmed, ISKANDARANI, Mohamad, GIVIGI, Sidney UAVs in Formation and Dynamic Encirclement via Model Predictive Control. In : IFAC World Congress. 2014, Cape Town, South Africa, pp. 1241-1246.
- [4] SASKA, Martin, KASL, Zdenek, and PREUCIL, Libor. Motion planning and control of formations of micro aerial vehicles. In : IFAC World Congress. 2014, Cape Town, South Africa
- [5] GUERRERO, Jose Alfredo and LOZANO, R. UAV Flight formation control. 2012.
- [6] CHIARAMONTI, Martina. Formation control laws for autonomous flight vehicles. In : 2006 14th Mediterranean Conference on Control and Automation. 2006, ANCONA, ITALY. p. 1-5.
- [7] SU, Housheng, WANG, Xiaofan, et LIN, Zongli. Flocking of multi-agents with a virtual

- leader. Automatic Control, IEEE Transactions on, 2009, vol. 54, no 2, p. 293-307.
- [8] LEONARD, Naomi Ehrich et FIORELLI, Edward. Virtual leaders, artificial potentials and coordinated control of groups. In : Decision and Control, 2001. Proceedings of the 40th IEEE Conference on. IEEE, 2001. p. 2968-2973.
 - [9] SHI, Hong, WANG, Long, et CHU, Tianguang. Virtual leader approach to coordinated control of multiple mobile agents with asymmetric interactions. Physica D: Nonlinear Phenomena, 2006, vol. 213, no 1, p. 51-65.
 - [10] OLFATI-SABER, Reza. Flocking for multi-agent dynamic systems: Algorithms and theory. IEEE Transactions on Automatic Control, 2006, vol. 51, no 3, p. 401-420.
 - [11] PARK, Jaemann, KIM, H. Jin, and HA, Seung-Yeal. Cucker-Smale flocking with inter-particle bonding forces. IEEE Transactions on Automatic Control, 2010, vol. 55, no 11, p. 2617-2623.
 - [12] BELKADI, A., CIARLETTA, L., and THEILLIOL, D. UAVs team flight training based on a virtual leader: Application to a fleet of Quadrirotors. In : Unmanned Aircraft Systems (ICUAS), 2015 International Conference on. IEEE, 2015, Denver, CO, USA. p. 1364-1369.
 - [13] DEB, Kalyanmoy, PRATAP, Amrit, AGARWAL, Sameer, et al. A fast and elitist multiobjective genetic algorithm: NSGA-II. Evolutionary Computation, IEEE Transactions on, 2002, vol. 6, no 2, p. 182-197.
 - [14] DORIGO, Marco, BIRATTARI, Mauro, BLUM, Christian, et al. (ed.). Ant Colony Optimization and Swarm Intelligence: 6th International Conference, ANTS 2008, Brussels, Belgium, September 22-24, 2008, Proceedings. Springer, 2008.
 - [15] KENNEDY, James. Particle swarm optimization. In : *Encyclopedia of Machine Learning*. Springer US, 2010. p. 760-766.
 - [16] ALFI, Alireza et MODARES, Hamidreza. System identification and control using adaptive particle swarm optimization. Applied Mathematical Modelling, 2011, vol. 35, no 3, p. 1210-1221.
 - [17] KAO, Chih-Cheng, CHUANG, Chin-Wen, et FUNG, Rong-Fong. The self-tuning PID control in a slider-crank mechanism system by applying particle swarm optimization approach. Mechatronics, 2006, vol. 16, no 8, p. 513-522.
 - [18] SHI, Yuhui et EBERHART, Russell C. Empirical study of particle swarm optimization. In : Evolutionary Computation, 1999. CEC 99. Proceedings of the 1999 Congress on. IEEE, 1999.
 - [19] TRELEA, Ioan Cristian. The particle swarm optimization algorithm: convergence analysis and parameter selection. *Information processing letters*, 2003, vol. 85, no 6, p. 317-325.
 - [20] EBERHART, Russ C. et KENNEDY, James. A new optimizer using particle swarm theory. In : Proceedings of the sixth international symposium on micro machine and human science. 1995. p. 39-43.