Decentralized Formation Shape Control of UAV Swarm using Dynamic Programming

Md Ali Azam[a] and Shankarachary Ragi[a]

[a]Department of Electrical Engineering, South Dakota School of Mines and Technology, Rapid City, SD 57701, USA

ABSTRACT

Formation control of unmanned aerial vehicles (UAVs) has many applications including target tracking, surveillance, terrain mapping, precision agriculture, etc. Although many centralized control methods (single command center/computer controlling the UAVs) exist, there are no standard decentralized control frameworks in the literature. In this paper, we present a novel UAV swarm formation control approach based on a decision theoretic approach. Specifically, we pose the decentralized swarm motion control problem as a Decentralized Markov Decision Process (Dec-MDP). Here, the objective is to drive the swarm from an initial geographical region to another geographical region where the swarm must lie on a certain geometrical surface (e.g., surface of a sphere). As most decision theoretic formulations suffer from the curse of dimensionality, we adapt an approximate dynamic programming method called *nominal belief-state optimization* (NBO) to solve the formation control problem approximately. We perform simulation studies in MATLAB to validate the performance of the algorithms.

Keywords: Swarm intelligence, formation control, decentralized Markov decision process, approximate dynamic programming

1. INTRODUCTION

Unmanned aerial vehicle (UAV) swarm formation control has applications in various fields such as infrastructure inspection[1] and surveillance,[2] target tracking,[3] and precision agriculture. The main objective in these application scenarios is to let the UAVs fly or hover in a certain geometrical formation, e.g., hover at locations lying on the surface of a sphere in a certain geographical region. There are methods existing in the literature to control UAV swarms using centralized methods,[4] where there is command center (centralized system) computing optimal motion commands for the UAVs. Although centralized methods are comparatively easy to develop and implement, but when the swarm is large, the computational complexity for evaluating optimal motion commands is extremely high. This is because the computational complexity in optimizing the UAV motion commands grows exponentially with the number of UAVs in the system. We present a novel decentralized UAV swarm formation control approach.

In this paper, the word *swarm* refers to a collections of UAVs. Each UAV makes decisions on its local kinematic controls, i.e., bank angle and forward acceleration. All UAVs in the system are aware of the global objective, which is arriving at a position on a given geometrical surface in a geographical region. We call the geometrical region in which UAVs are supposed to arrive as *formation shape* which can be any kind of geometrical shape i.e., rectangular, circular and so on. The swarm should complete the global objectives in the shortest time possible while avoiding collisions among the UAVs in the swarm.

We use a dynamic programming approach to solve the decentralized swarm motion control problem. Specifically, we formulate the swarm control problem as a decentralized Markov decision process. As most dynamic programming problems suffer from high dimensionality, we adapt a fast heuristic approach called *nominal belief-state optimization* (NBO) to solve the formation control problem approximately.

This work was supported in part by Air Force Office of Scientific Research under grant FA9550-19-1-0070.

Further author information: (Send correspondence to M.A.A.)

M.A.A.: E-mail: mdali.azam@mines.sdsmt.edu

S.R.: E-mail: shankarachary.ragi@sdsmt.edu

Swarm formation is required in many applications where a collection of UAVs are required to collect sensor data (e.g., images) from a certain geographical region where it may be optimal for the UAVs to lie on a geometrical surface. We develop our methods in 2-D in this study, which will be extended to 3-D in our future studies.

1.1 Literature Review

Taking difficulties and drawbacks of centralized methods into account, a few papers in the literature developed decentralized methods to control UAV swarms. The authors of[5] used the decentralized partially observable Markov decision process (Dec-POMDP) to formulate and solve a target tracking problem with a swarm of decentralized UAVs. As solving decentralized POMDP is very hard, the authors introduced an approximate dynamic programming method called *nominal belief-state optimization* (NBO) to solve the control problem. The authors of[6] used fixed-wing UAV motion models with forward acceleration and bank angle being the UAV motion controls, which are subject to certain constraints. They also account collision avoidance between UAVs and obstacles and among UAVs, and wind disturbance on UAVs. However, their methods were developed for a centralized system, as opposed to our decentralized system studied here. Moreover, their methods were developed for a different application - multi-target tracking.

The authors in[7] propose formation control of a group of UAVs using decentralized Model Predictive Control (MPC). In their work, the UAVs were able to avoid collisions with multiple obstacles in a decentralized manner. They used figure of eight as reference trajectory and result shows UAVs were able to avoid collision with obstacles, and between UAVs. Several recent papers describe formation control of different geometric shapes, e.g. multi-agent circular shape with a leader.[8] The authors of[8] propose centralized formation control, which is not suitable for swarm control when the number of UAVs in the swarm is large.

1.2 Key Contributions

- We formulate the UAV swarm formation control problem as a decentralized Markov decision process (Dec-MDP).
- We extend an approximate dynamic programming method called nominal belief-state optimization (NBO) to solve the formation control problem.
- We perform simulation studies to validate the swarm formation control algorithms developed here.
- We perform numerical studies to quantify the impact of neighborhood threshold on average computation time and average pairwise distance.

The remaining parts of this paper are organized as follows. Section 2 provides the problem specification. We formulate the problem using decentralized Markov decision process in Section 3 followed by the discussion on the NBO approach in Section 4. UAV motion model and kinematic equations are provided in Section 5. In Section 6, we discuss simulation results to evaluate the performance of our method. Finally, we provide conclusions in Section 7.

2. PROBLEM SPECIFICATION

Unmanned aerial vehicles: We assume UAV motion dynamics as described in,[6] where the motion controls are *forward acceleration* and *bank angle*. UAVs are allowed to hover at any location, i.e., the minimum speed limit on the UAVs is zero.

Communications and Sensing: We assume that UAVs are equipped with sensing systems and wireless transceivers using which each UAV learns the exact location and the velocity of the nearest neighboring UAV. Our decentralized method requires only the state of the nearest neighbor to optimize the control commands of the local UAV.

Objective: The goal is to control the swarm (optimizing control commands) in a decentralized manner such that the swarm arrives on a certain known geometrical surface in a certain region in the shortest time possible. The UAVs must complete this objective while avoiding collisions.

3. PROBLEM FORMULATION

We formulate the swarm formation control problem as a decentralized Markov decision process (Dec-MDP). Dec-MDP is a mathematical formulation useful for modeling control problems for decentralized decision making. This formulation has the following advantages: 1) allows us to efficiently utilize the computing resources on-board all the UAVs, 2) requires less computational time compared to a centralized approach, 3) as UAVs are decentralized, point of failure of the entire mission is minimal, 4) decentralized approach provides robustness to addition or deletion of UAVs to the swarm, 5) UAVs do not need to rely on a central command center for evaluating optimal control commands. With Dec-MDP formulation, we can achieve the above features in our swarm control method. We define the key components of Dec-MDP as follows. Here, k represents the discrete-time index.

3.1 Dec-MDP Ingredients

Agents/UAVs: We assume there are N UAVs in our system. The set of UAVs is given by $I = \{1, ..., N\}$. Traditionally, this component is referred to as set of agents or set of independent decision makers. Here, an agent is a UAV.

States: The state of the system s_k includes the locations and velocities of each UAV.

Actions: The actions are the controllable aspects of the system. We define action vector $a_k = (a_k^1, \ldots, a_k^N)$, where a_k^i represents the action vector at UAV i, which includes the forward acceleration and the bank angle for the UAV.

State Transition Law: State transition law describes how the state evolves over time. Specifically, the transition law is a conditional probability distribution of the next state given the current state and the current control actions (assuming Markovian property holds). The transition law is given by $s_{k+1} \sim p_k(\cdot|s_k, a_k)$, where p_k is the conditional probability distribution. Since the state of the system only includes the states of the UAVs, the state transition law is completely determined by the kinematic equations of the UAVs (discussed in the next section). In other words, the transition law is given by $s_{k+1}^i = \psi(s_k^i, a_k^i) + w_k^i, i = 1, \ldots, N$, where s_k^i represents the state of the ith UAV and a_k^i indicates the local kinematic controls (forward acceleration and bank angle) of ith UAV, ψ represents the kinematic motion model as discussed in the next section, and w_k^i represents noise, which is modeled as a zero-mean Gaussian random variable.

Cost Function: The cost function $C(s_k, a_k)$ deals with cost of being in a given state s_k and performing actions a_k. Here, s_k represents the global state, i.e., the state of all the UAVs in the system. Since the problem is decentralized, each UAV only has access to its local state and the state of the nearest neighboring UAV. Let $b_k^i = (s_k^i, s_k^{nn})$ represent that local system state at UAV i, where s_k^{nn} is the state of the nearest neighboring UAV, and $nn \in I\backslash\{i\}$.

Let d^i is the destination location UAV i must reach, and $d_{\text{coll,thresh}}$ is the distance between the UAVs below which the UAVs are considered to be at the risk of collision. We now define the local cost function for UAV i as follows:

$$c(b_k^i, a_k^i, a_k^{nn}) = w_1 \left[\text{dist}(s_k^{i,\text{pos}}, d^i) + \text{dist}(\text{s}_\text{k}^{\text{nn,pos}}, \text{d}^{\text{nn}})\right] +$$
$$w_2 \left[\text{dist}(s_k^i, s_k^{nn})^{-1} \mathbb{I}(\text{dist}\left(s_k^i, s_k^{nn}\right) < d_{\text{coll,thresh}})\right]$$

where $s_k^{i,\text{pos}}$ represents the location of the ith UAV, w_1 and w_2 are weighting parameters, $\text{dist}(a, b)$ represents the distance between locations a and b, and $\mathbb{I}(a)$ is the indicator function, i.e., $\mathbb{I}(a) = 1$ if the argument a is true and 0 otherwise.

By minimizing the above cost function, each UAV optimizes its own control commands and that of its neighbor, but only implement its own local control commands and discards the commands optimizes for its neighbor. The first part of the cost function lets the UAV reach its destination, while the second part minimizes the risk of collisions between UAVs.

The Dec-MDP starts at an initial random state s_0 and the state of the system evolves according to the state-transition law and the control commands applied at each UAV. The overall objective is to optimize the control commands at each UAV i such that the expected cumulative local cost over a horizon H (shown below) is minimized.

$$\min_{\{a_k^i, a_k^{nn}\}, k=0,\dots,H-1} \mathrm{E}\left[\sum_{k=0}^{H-1} c(b_k^i, a_k^i, a_k^{nn}) \middle| b_0^i\right], \tag{1}$$

where b_0^i is the initial local state at UAV i, and the expectation $E[\cdot]$ is over the stochastic evolution of the local state over time (due to the random variables present in the UAV kinematic equations).

4. NBO APPROACH TO SOLVE DEC-MDP

It is well know in the literature that solving Eq. 1 exactly is computationally prohibitive and not practical. For this reason, we extend a heuristic approach called *nominal belief-state optimization* (NBO).[6] As discussed in the previous section, we let a UAV optimize its own and its nearest neighbor's kinematic controls over the time horizon H. Once the UAV calculates local controls for itself and its neighbors, the UAV implement its own controls and discards its neighbors controls at each time step. Since obtaining the expectation in Eq. 1 exactly is not tractable, the NBO approach approximates this expectation by assuming that all the future random variables (over which the expectation is supposed to be evaluated) assume the nominal values, i.e., the mean values. Since we model the above-mentioned random variable as zero-mean Gaussian, the nominal values are simply zeros. In summary, the NBO approach approximates the cumulative cost function in Eq. 1 by replacing the expectation with the random trajectory of the states over time by a sequence of states obtained by replacing future random variables with zeros.

In the NBO method, the objective function at agent i is approximated as follows:

$$J(b_0^i) \approx \sum_{k=0}^{H-1} c(\hat{b}_k^i, a_k^i, a_k^{nn}),$$

where $\hat{b}_1^i, \hat{b}_2^i, \dots, \hat{b}_{H-1}^i$ is a *nominal* local state sequence.

5. UAV MOTION MODEL

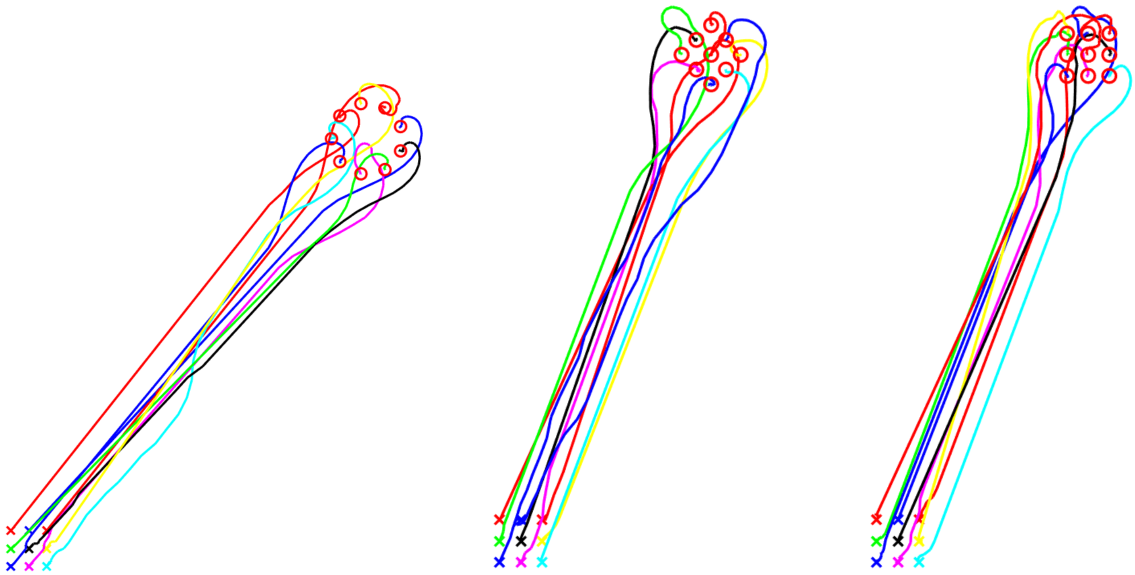
The state of the ith UAV at time k is given by $s_k^i = \left(p_k^i, q_k^i, V_k^i, \theta_k^i\right)$, where (p_k^i, q_k^i) represents the position coordinates, V_k^i represents the speed, and θ_k^i represents the heading angle. The kinematic control action for UAV i is given by $a_k^i = (f_k^i, \phi_k^i)$, where f_k^i is the forward acceleration and ϕ_k^i is the bank angle of the UAV. The kinematic equations of the UAV motion[6] are as follows:

$$\begin{aligned}
V_{k+1}^i &= \left[V_k^i + f_k^i T\right]_{V_{\min}}^{V_{\max}} + w_k^{i,\text{speed}} \\
\theta_{k+1}^i &= \theta_k^i + (gT\tan(\phi_k^i)/V_k^i) + w_k^{i,\text{heading}}, \\
p_{k+1}^i &= p_k^i + V_k^i T\cos(\theta_k^i) + w_k^{i,\text{xpos}}, \\
q_{k+1}^i &= q_k^i + V_k^i T\sin(\theta_k^i) + w_k^{i,\text{ypos}},
\end{aligned}$$

where $[v]_{V_{\min}}^{V_{\max}} = \max\{V_{\min}, \min(V_{\max}, v)\}$, $V_{\min}$ and $V_{\max}$ are the minimum and the maximum limits of each UAV, g is the acceleration due to gravity, T is the length of the time step, and $w_k^{i,\text{speed}}, w_k^{i,\text{heading}}, w_k^{i,\text{xpos}}, w_k^{i,\text{ypos}}$ are the zero-mean Gaussian random variables.

6. SIMULATION RESULTS

We assume that each UAV has its own on-board computer to compute the local optimal control decisions. We implement the above-discussed NBO approach to solve the swarm control problem in MATLAB. We test our methods with three formation shapes - a circular shape, a rectangular shape, and a square shape. The UAVs are aware of the shape dimensions and the exact location of shape. Each UAV randomly picks a location on the formation shape, and uses the NBO approach to arrive at this location. We use MATLAB's *fmincon* to solve the NBO optimization problem. Here, we set the horizon length to $H = 7$ time steps.



(a) Circular formation (b) Rectangle formation (c) Square formation

Figure 1: 9 UAVs converging to the formation shapes using the *Dec-MDP approach*

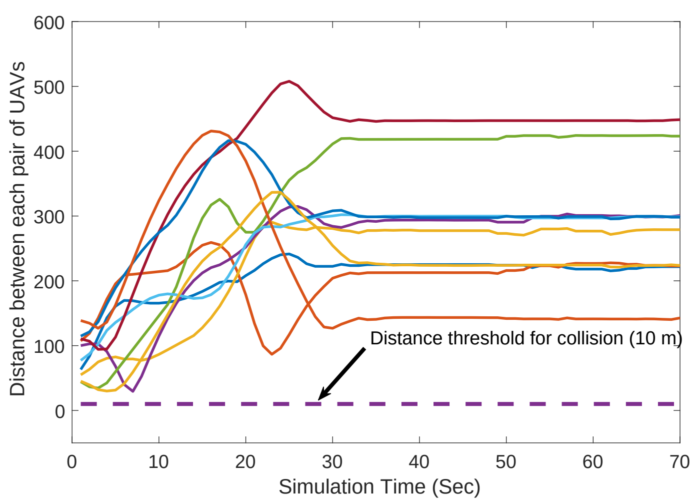


Figure 2: Distance between each pair of UAVs

We define the following metrics to measure the performance of our formation control approach: 1) T_c - *average computation time to evaluate the optimal control commands* and 2) T_f: *time taken for the swarm to arrive on the formation shape*. As a benchmark method, we use a centralized approach to solve the above-discussed swarm formation control problem. In other words, we use a single NBO algorithm, which optimizes the motion control commands for all the UAVs together based on the global state of the system. We implement this centralized algorithm in MATLAB.

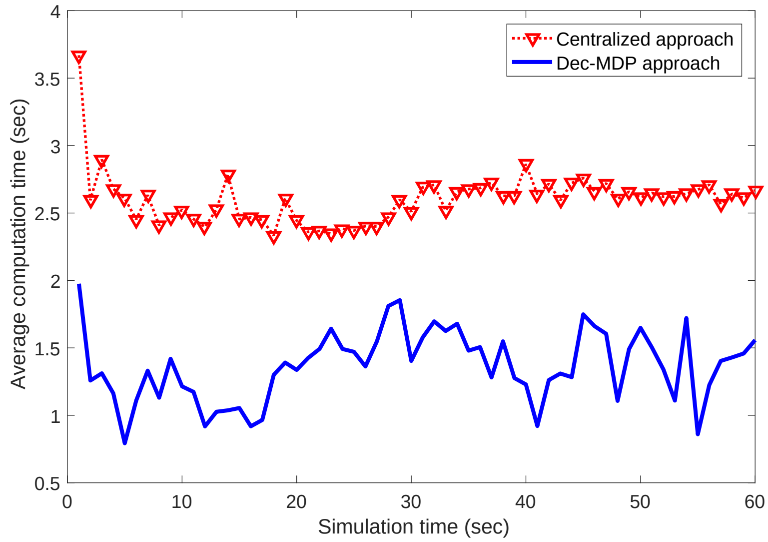


Figure 3: Computation time (T_c): centralized vs decentralized method

	Dec-MDP	Centralized
$T_f(sec)$	16.7	25.98

Table 1: Average time taken by the swarm to arrive at the formation shape.

We implement the Dec-MDP approach with a circular formation shape, a rectangular formation shape, and a square formation shape. The resulting swarm motion is shown in Figures 1a, 1b, and 1c respectively. For the scenario in Figure 1a, we also plot the distance between every pair of UAVs in the swarm as shown in Figure 2. Here, we assume that there is a collision risk between a pair of UAVs when the distance between them is less than 10 m. Clearly, the figures 1, and 2 demonstrate that our decentralized algorithm drives the swarm to the destination while successfully avoiding collisions between the UAVs.

We calculate the T_c and T_f values for both the centralized and the decentralized algorithms for 9 UAVs. Figure 3 and Table I clearly demonstrates that our decentralized method significantly outperforms the centralized method with respect to both the metrics T_c and T_f.

We now compute average computation time and average pairwise distance with respect to neighborhood threshold where each UAV communicates with other UAVs within the radius of neighborhood threshold. If neighborhood threshold is infinity, a UAV can communicate with all other UAVs in the swarm. UAVs optimize its decision together with neighbors which depends on neighborhood threshold and implement its own control. We expect that with the increase of neighborhood threshold, average computation time rises and after certain neighborhood threshold, average computation time saturates. Figure 4 shows average computation time rise until neighborhood threshold reach 240 m and then waves between 20 to 25 sec.

We also expect that with the increase of neighborhood threshold, average pairwise distance drops. The reason we are interested in analyzing average pairwise distance is, we expect the swarm to be as closely as possible while avoiding collision between UAVs. Small average pairwise distance allows the swarm to be more cooperative while saving battery life as communication distance depends on distance between UAVs. Figure 5 and 4 suggest that neighborhood threshold more than 130 m allows UAVs to stay closely in the swarm with reasonable computation cost.

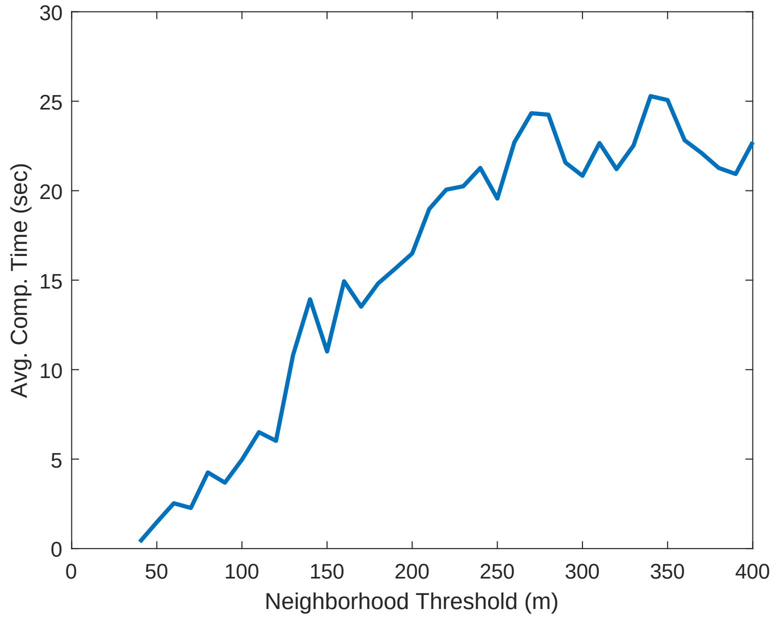


Figure 4: Average computation time with respect to neighborhood threshold

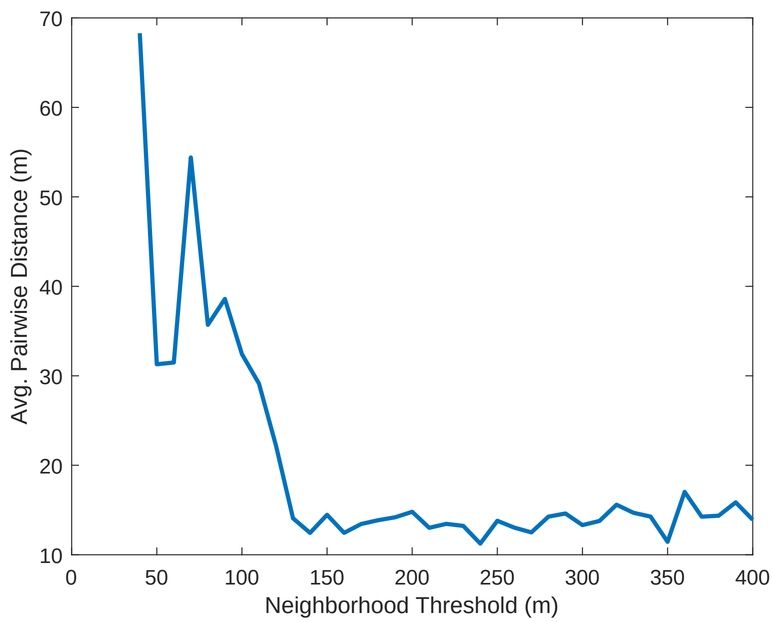


Figure 5: Average pairwise distance with respect to neighborhood threshold

7. CONCLUSIONS

In this study, we designed a decentralized formation control problem to bring a swarm of UAVs from initial position to target location in certain geometrical shape (formation shape) using *decentralized Markov decision process* (Dec-MDP). We extended an approximate dynamic programming method called *nominal belief-state optimization* (NBO) to solve the formation control problem, which we denoted as Dec-MDP. We accounted collision avoidance between UAVs which was clearly described with numerical results in section 6. We used centralized method to compare performance metrics of our Dec-MDP approach. The result shows that the *average computation time* (T_c) and *time taken for the swarm to arrive at the formation shape* T_f of Dec-MDP approach is significantly smaller than that of the centralized method. We also performed numerical studies to quantify the impact of neighborhood threshold on average computation time and average pairwise distance.

REFERENCES

[1] S. Waharte, N. Trigoni, S. Julier. "Coordinated Search with a Swarm of UAVs," *6th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks*, 2009, 10.1109.

[2] D. V. D. Walle, B. Fidan, A. Sutton, C. Yu, and B. D. O. Anderson,"Non-hierarchical UAV Formation Control for Surveillance Tasks," *American Control Conf. Seattle, WA*, pp. 777-782, June 2008.

[3] I. Shames, B. Fidan, and B. D. O. Anderson,"Close Target Reconnaissance using Autonomous UAV Formations," *47th IEEE Conf. Decision and Control, Cancun, Mexico*, pp. 1729-1734, Dec. 2008.

[4] S. Ragi, E. K. P. Chong, "Dynamic UAV Path Planning for Multitargte Tracking," *American Control Conference,* pp. 3845-3850, 2012.

[5] S. Ragi, E. K. P. Chong, "Decentralized Guidance Control of UAVs with Explicit Optimization of Communication," *J Intell Robot Syst*, vol. 73 pp. 811-822, 2014.

[6] S. Ragi and E. K. P. Chong, "UAV Path Planning in a Dynamic Environment via Partially Observable Markov Decision Process," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 49, pp. 2397-2412, 2013.

[7] Í. B. Viana, D. A. D. Santos, L. C. S. Góes, "Formation Control of Multirotor Aerial Vehicles using Decentralized MPC," *J Braz. Soc. Mech. Sci. Eng.*, June 2018, 40:306.

[8] L. Zhao, D. Ma, "Circle Formation Control for Multi-agent Systems with a Leader," *Control Theory Technol.*, vol. 13, pp. 82-88, 2015.