algorithms

MDPI

Article

UAV Formation Shape Control via Decentralized Markov Decision Processes

Md Ali Azam [1], Hans D. Mittelmann [2] and Shankarachary Ragi [1,*]

[1] Electrical Engineering, South Dakota School of Mines and Technology, Rapid City, SD 57701, USA; azam.ete.ruet@gmail.com
[2] School of Mathematical and Statistical Sciences, Arizona State University, Tempe, AZ 85287, USA; mittelmann@asu.edu
* Correspondence: Shankarachary.Ragi@sdsmt.edu

Abstract: In this paper, we present a decentralized unmanned aerial vehicle (UAV) swarm formation control approach based on a decision theoretic approach. Specifically, we pose the UAV swarm motion control problem as a decentralized Markov decision process (Dec-MDP). Here, the goal is to drive the UAV swarm from an initial geographical region to another geographical region where the swarm must form a three-dimensional shape (e.g., surface of a sphere). As most decision-theoretic formulations suffer from the curse of dimensionality, we adapt an existing fast approximate dynamic programming method called *nominal belief-state optimization* (NBO) to approximately solve the formation control problem. We perform numerical studies in MATLAB to validate the performance of the above control algorithms.

Keywords: swarm intelligence; formation control; decentralized Markov decision process; approximate dynamic programming

check for updates

Citation: Azam, M.A.; Mittelmann, H.D.; Ragi, S. UAV Formation Shape Control via Decentralized Markov Decision Processes. *Algorithms* **2021**, *14*, 91. https://doi.org/10.3390/a14030091

Academic Editor: Frank Werner

Received: 11 February 2021
Accepted: 15 March 2021
Published: 17 March 2021

1. Introduction

Unmanned Aerial Vehicle (UAV) swarm formation has applications in many areas of research, such as infrastructure inspection [1], surveillance [2,3], target tracking [4], and precision agriculture [5]. There are existing methods in the literature to control UAV swarms using centralized methods [6–11], where there is a command center (centralized system) computing optimal motion commands for the UAVs. Centralized methods are relatively easy to develop and implement, but computational complexity grows exponentially with the size of the swarm. To address this challenge, we present a decentralized UAV swarm formation control approach using decentralized Markov decision process framework. The main goal this study is to drive the swarm fly and hover in a certain geographical region while forming a certain geometrical shape. The motivation for studying such problems comes from data fusion applications with UAV swarms where the fusion performance depends on the strategic relative separation of the UAVs from each other [12,13]. We previously studied decentralized decision making frameworks for UAV swarm formation in two-dimensional (2D) scenarios [14], while in this study, we decentralized control strategies in three-dimensional (3D) scenarios.

The formation control of vehicle swarms has many applications in areas including infrastructure inspection, precision agriculture, intelligent transportation, and surveillance. In many applications in these domains, strategic placement of the vehicles (forming a certain geometrical shape, e.g., points on the surface of a sphere) can lead to significant gains in data fusion performance due to the different vantage points of the sensors on-board the vehicles observing a target of interest [10]. Suppose the vehicles carry optical cameras generating 2D images of a 3D object, and if the goal is to reconstruct the 3D shape of the object via the 2D images (i.e., tomography-based methods), the strategic placements of the vehicles around the object can have significant impact on the performance of the 3D shape reconstruction.

Different formation control settings have been studied in the past: ground vehicles [15–17], unmanned aerial vehicles (UAVs) [18,19], surface and underwater autonomous vehicles (AUVs) [20,21]. Regardless of settings, there are many different methodologies developed by the researchers to tackle formation control problem, e.g., behavior-based, virtual structure, and leader following. The authors of [22,23] developed a behavior-based approach in which they described the desired behavior for each robot, e.g., collision avoidance, formation keeping, and target seeking. The control commands for the robot is determined by weighing the relative importance of each behavior. The virtual structure approach [24,25] takes a physical object shape as a reference and mimics the formation of that shape. The robots are required to communicate with each other in order to achieve a formation in this approach which requires significant communication costs (e.g., bandwidth). The leader following approach [15] requires a robot, assigned as a leader, which moves according to a predetermined trajectory. The other robots, the followers, are designed to follow the leader, maintaining a desired distance and orientation with respect to the leader. The main drawback of this approach is that the followers are dependent on the leader to achieve the goal (formation). The system may collapse if the leader fails when the leader possibly runs short on power or when the communications link fails. Considering the aforementioned limitations of formation control, which specifically stem from centralized approaches, we develop a decentralized Markov decision process (Dec-MDP)-based formation control approach for a UAV swarm. Our decentralized control strategies are robust to failures of individual UAVs in the swarm and also robust to communications link failures.

Centralized control strategies for UAV swarm control are well studied [7–9,11,26]. For instance, the authors of [6,7] developed UAV control strategies for target tracking in a centralized setting. In centralized systems like these, typically, there exists a notional fusion center (a computing node) that collects and fuses the sensor measurements (e.g., using Bayes' theorem) from all the UAVs and runs a tracking algorithm (e.g., Kalman filter) to maintain and update the estimate of the state of the system. More importantly, the fusion center computes the combined optimal control commands for all the UAVs to maximize the system performance. For instance, the authors of [10] used the notion of fusion center to control fixed-wing UAVs for multitarget tracking while accounting for collision avoidance and wind disturbance on UAVs. Although, these centralized control and fusion strategies are easy to implement, they are computationally expensive especially if the swarm is large. Specifically, the computational complexity increases exponentially with the number of UAVs in the swarm.

To tackle these challenges, a few studies in the literature developed decentralized control strategies [14,26–29]. The authors of [26] 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 difficult (as is the case with solving any decision-theoretic methods), the authors introduced an approximate dynamic programming method called *nominal belief-state optimization* (NBO) to solve the control problem. The authors in [30] developed a UAV formation control approach 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 a figure of eight as a reference trajectory; their results show that the UAVs were able to avoid collision with obstacles and among themselves. Several recent papers describe the formation control of different geometric shapes, e.g., multi-agent circular shape with a leader [9]. The authors of [9] propose centralized formation control, which is not suitable for swarm control when the number of UAVs in the swarm is large. Although decentralized control methods exist in the literature, our method is novel in the sense that each UAV in the swarm optimizes its own control commands and its nearest neighbor's controls over time. Then, each UAV implements its own optimized controls, and discards the neighbor's controls. We anticipate, from this decentralized control optimization approach, a global cooperative behavior among the UAVs emerges mimicking a centralized control approach. The authors of [31] demonstrated a successful use of a distributed UAV control framework for wildfire monitoring while avoiding in-flight collisions. The authors of [32] introduced path tracking and desired formation for networked mobile vehicles using non-linear control theory to maintain the formation in the network. They have showed that path tracking error of each vehicle is reduced to zero and formation is achieved asymptotically. As centralized control strategies suffer from exponential computational complexity and high memory usage, the decentralized control methods are being actively pursued in the context of swarm control, especially when the size of the swarm is large. A survey of these decentralized control strategies can be found in [29].

In this paper, we develop a novel decentralized UAV swarm formation control approach using Dec-MDP formation. In this problem, the goal is to optimize the UAV control decisions (e.g., waypoints) in a decentralized manner, such that the swarm forms a certain geometrical shape while avoiding collisions. We use dynamic programming principles to solve the decentralized swarm motion control problem. As most dynamic programming problems suffer from the curse of dimensionality, we adapt a fast heuristic approach called *nominal belief-state optimization* (NBO) [10,33] to approximately solve the formation control problem. We perform simulation studies to validate our control algorithms and compare their performance with centralized approaches for bench marking the performance.

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 numerical studies in MATLAB to validate the swarm formation control algorithms developed here.
- One of the key contributions of this paper is to induce cooperative behavior among the UAVs in the swarm via the following novel decentralized control optimization strategy:
  - Each UAV i optimizes the control vector $[a_k^i, a_k^{nn}]$ at time k, where a_k^i is the control vector for UAV i, and a_k^{nn} is the control vector for its nearest neighbor.
  - Next, UAV i discards the optimized controls for its neighbor and implements just its own controls a_k^i.
  - Each UAV in the system implements the above approach.

The rest of the paper is organized as follows. Section 2 provides the problem specification and objectives. We also formulate the problem using decentralized Markov decision process in Section 2 followed by the discussion on the NBO approach in Section 3. UAV motion model and dynamics are provided in Section 4. In Section 5, we discuss simulation results to evaluate the performance of our method.

2. Problem Formulation

Unmanned aerial vehicles: We consider quadrotor motion dynamics in 3D, as modeled in [34,35]. In this study, our goal is to optimize the *waypoints* (position coordinates in 3D space) for the quadrotors to guide the UAVs to their destination formation shape while avoiding collisions.

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

Objective: The goal is to control the swarm (optimizing *waypoints*) in a decentralized manner, such that the swarm arrives at a certain pre-determined 3D geometrical surface in the shortest time possible while avoiding collisions.

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 centralized approaches, (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. We define the key components of Dec-MDP as follows. Here, k represents the discrete-time index.

Dec-MDP Ingredients

Agents/UAVs: We assume there are N UAVs in our system. The set of UAVs is given by an index vector $I = \{1, ..., N\}$. This index vectors may be referred to as a set of agents or set of independent decision makers. Here, a UAV can be considered an agent or a decision maker.

States: We model the system dynamics in discrete time, where k represents the time index. The state of the system s_k includes the locations and velocities of all the UAVs in the system.

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 position coordinates in 3D 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 the 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 dynamics of the UAVs (discussed in the next section). In other words, the transition law is given by $s_{k+1}^i = \psi(s_k^i, a_k^i) + \mathcal{W}_k^i, i = 1, \ldots, N$, where s_k^i represents the state of the ith UAV and a_k^i indicates the local dynamic controls (position coordinates) of ith UAV, ψ represents the motion model as discussed in Section 4, and $\mathcal{W}_k^i$ 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 be 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:

$$\begin{aligned} c(b_k^i, a_k^i, a_k^{nn}) = w_1 &\Big[\text{dist}(s_k^{i,\text{pos}}, d^i) + \text{dist}(\mathrm{s_k^{nn,pos}}, \mathrm{d^{nn}})\Big] \\ &+ w_2 \Big[\text{dist}(s_k^i, s_k^{nn})^{-1} \mathbb{I}\Big(\text{dist}(s_k^i, s_k^{nn}) < d_{\text{coll,thresh}}\Big)\Big] \end{aligned} \tag{1}$$

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 optimized 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. 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 dynamic equations).

$$\min_{\{a_k^i, a_k^{nn}\}, k=0,\ldots,H-1} \mathrm{E}\left[\sum_{k=0}^{H-1} c(b_k^i, a_k^i, a_k^{nn}) \middle| b_0^i\right] \tag{2}$$

3. NBO Approach to Solve Dec-MDP

It is well know in the literature that solving Equation (2) exactly is computationally prohibitive and not practical. For this reason, we extend a heuristic approach called *nominal belief-state optimization* (NBO) [10]. As discussed in the previous section, we let a UAV optimize its own and its nearest neighbor's controls over the time horizon H. Once the UAV calculates local controls for itself and its neighbors, the UAV implements its own controls and discards its neighbors controls at each time step. Since obtaining the expectation in Equation (2) 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 Equation (2) 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:

$$J(b_0^i) \approx \sum_{k=0}^{H-1} c(\hat{b}_k^i, a_k^i, a_k^{nn}), \tag{3}$$

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

4. UAV Motion Model

The state of the ith UAV at time k is given by $s_k^i = (x_k^i, y_k^i, z_k^i, \phi_k^i, \theta_k^i, \psi_k^i)$, where (x_k^i, y_k^i, z_k^i) are position coordinates and $[\phi_k^i, \theta_k^i, \psi_k^i] = [$*bank angle, pitch angle, heading angle*$]$ are the Euler angles. The UAV motion dynamics are given by the following equations.

$$\begin{aligned}
u_{k+1} &= T(-g\sin(\theta_k) + r_k v_k - q_k w_k) + u_k + \mathcal{W}_k^u \\
v_{k+1} &= T(g\sin(\phi_k)\cos(\theta_k) - r_k u_k + p_k w_k) + v_k + \mathcal{W}_k^v \\
w_{k+1} &= T\left(\frac{1}{m}(-F_z) + g\cos(\phi_k)\cos(\theta_k) + q_k u_k - p_k v_k\right) + w_k + \mathcal{W}_k^w \\
p_{k+1} &= T\left(\frac{1}{I_{xx}}(L + (I_{yy} - I_{zz})q_k r_k)\right) + p_k + \mathcal{W}_k^p \\
q_{k+1} &= T\left(\frac{1}{I_{yy}}(M + (I_{zz} - I_{xx})p_k r_k)\right) + q_k + \mathcal{W}_k^q \\
r_{k+1} &= T\left(\frac{1}{I_{zz}}(N + (I_{xx} - I_{yy})p_k q_k)\right) + r_k + \mathcal{W}_k^r \\
\phi_{k+1} &= T(p_k + (q_k\sin\phi_k + r_k\cos\phi_k)\tan\theta_k) + \phi_k + \mathcal{W}_k^\phi \\
\theta_{k+1} &= T(q_k\cos\phi_k - r_k\sin\phi_k) + \theta_k + \mathcal{W}_k^\theta \\
\psi_{k+1} &= T((q_k\sin\phi_k + r_k\cos\phi_k)\sec\theta_k) + \psi_k + \mathcal{W}_k^\psi \\
x_{k+1} &= T\Big(c_{\theta_k}c_{\psi_k}u^b + (-c_{\phi_k}s_{\psi_k} + s_{\phi_k}s_{\theta_k}c_{\psi_k})v^b + (s_{\phi_k}s_{\psi_k} + c_{\phi_k}s_{\theta_k}c_{\psi_k})w^b\Big) + x_k + \mathcal{W}_k^x
\end{aligned}$$

$$y_{k+1} = T\Big(c_{\theta_k} s_{\psi_k} u^b + (c_{\phi_k} c_{\psi_k} + s_{\phi_k} s_{\theta_k} s_{\psi_k}) v^b + (-s_{\phi_k} c_{\psi_k} + c_{\phi_k} s_{\theta_k} s_{\psi_k}) w^b\Big) + y_k + \mathcal{W}_k^y$$
$$z_{k+1} = T\Big(-1 * (-s_{\theta_k} u^b + s_{\phi_k} c_{\theta_k} v^b + c_{\phi_k} c_{\theta_k} w^b)\Big) + z_k + \mathcal{W}_k^z$$

where, $\mathcal{W}_k$ is a zero-mean Gaussian random variables, $[u_k, v_k, w_k]$ = [*longitudinal velocity, lateral velocity, normal velocity*] are the linear velocity, and $[p_k, q_k, r_k]$ = [*roll rate, pitch rate, yaw rate*] represent the angular velocity of the vehicle at time k. $[F_x, F_y, F_z]$ are linear translation forces and $[L, M, N]$ are angular moments.

UAV Motion Control

We implement a linear controller [36] to produce the appropriate torque and thrust in order to drive the UAV to the desired state in SO(3), governed by the optimized waypoints. The Figure 1 shows how the waypoints generator works with the controller. We make the following assumptions for the linear controller.

- We linearize the trigonometric functions assuming roll angle ϕ and pitch angle θ small enough, i.e., $\cos\phi = 1$, $\sin\phi = \phi$, $\cos\theta = 1$, $\sin\theta = \theta$
- The angular velocity of the UAV is also considered small enough

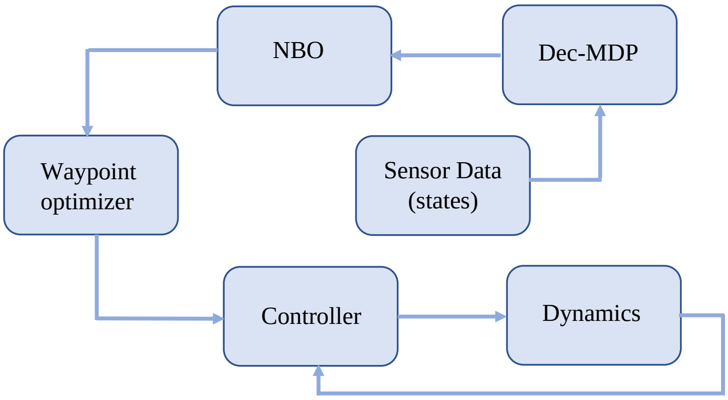

Figure 1. UAV formation shape control architecture.

The linear controller is described extensively in [37,38]. The control problem is to calculate the inputs $u_1 = \sum_{i=1}^{4} F_i$ and u_2 required to track a set of waypoints r_k^w. The input u_2 is given by the following equation.

$$u_2 = \begin{bmatrix} 0 & L & 0 & -L \\ -L & 0 & L & 0 \\ \gamma & \gamma & \gamma & \gamma \end{bmatrix} \begin{bmatrix} F_1 \\ F_2 \\ F_3 \\ F_4 \end{bmatrix}$$

where, $[F_1, F_2, F_3, F_4]$ are propeller forces and γ is the drag coefficient.

Position control. The position control method use the bank and the pitch angles as inputs to drive the position of the UAV. The position controller determines the desired bank angle ϕ^{des} and desired pitch angle θ^{des}. The desired bank and pitch angles are used to calculate the desired speed of the UAV [37].

5. 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 in two scenarios—a spherical shape with and without an obstacle. 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 $H = 3$ time steps.

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.

We implement the Dec-MDP approach with a spherical formation shape with and without an obstacle. The resulting swarm motion is shown in Figure 2 for the spherical formation shape in the absence of any obstacle using the cost function described in Equation (1). The scenario with an obstacle considers the following cost function.

$$\begin{aligned} c(b_k^i, a_k^i, a_k^{nn}) = w_1 &\Big[\mathrm{dist}(s_k^{i,\mathrm{pos}}, d^i) + \mathrm{dist}(\mathrm{s}_\mathrm{k}^{\mathrm{nn,pos}}, \mathrm{d}^{\mathrm{nn}})\Big] \\ &+ w_2\Big[\mathrm{dist}(s_k^i, s_k^{nn})^{-1}\mathbb{I}(\mathrm{dist}\Big(s_k^i, s_k^{nn}) < d_{\mathrm{coll,thresh}}\Big)\Big] \\ &+ w_3\Big[\mathrm{dist}(s_k^i, s_k^{obstacle})^{-1}\mathbb{I}(\mathrm{dist}\Big(s_k^i, s_k^{obstacle}) < d_{\mathrm{coll,obstacle}}\Big)\Big] \end{aligned}$$

where $s_k^{obstacle}$ is the location of an obstacle, $d_{\mathrm{coll,obstacle}}$ is a collision threshold with the obstacle, and w_3 is a weighting parameter. The indicator function $\mathbb{I}(b) = 1$, if the argument b is true and 0 otherwise. The resulting motion of the scenario with the obstacle is shown in Figure 3. For this scenario, we also plot the distance between every pair of UAVs in the swarm, as shown in Figure 4. Here, we assume that there is a collision risk between a pair of UAVs when the distance between them is less than 5 m. Clearly, the Figures 3 and 4 demonstrate that our decentralized algorithm drives the swarm to the destination while successfully avoiding collisions between the UAVs.

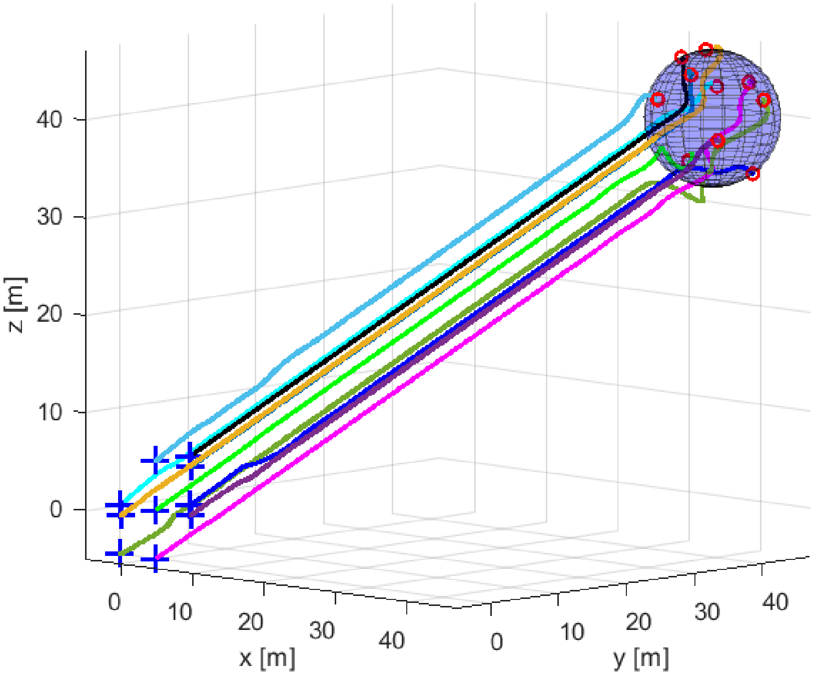

Figure 2. UAV swarm converging to the spherical formation shapes in 3D.

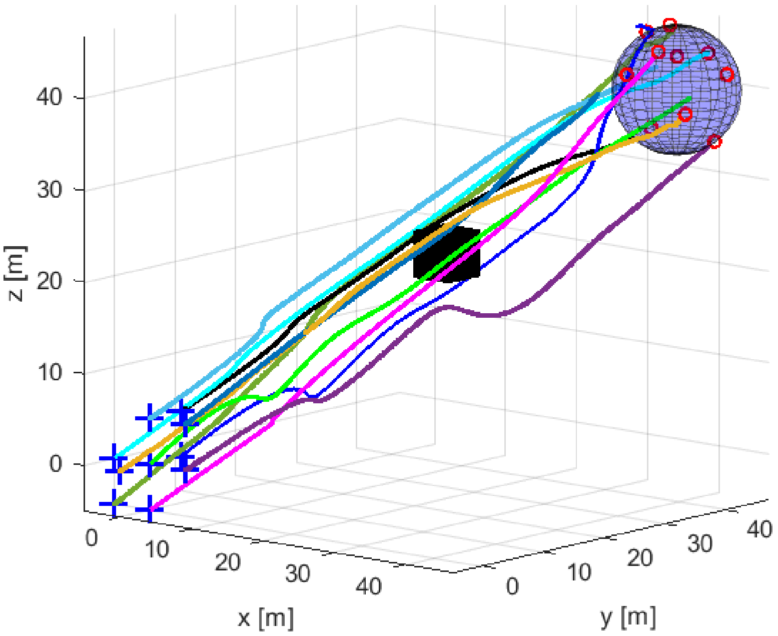

Figure 3. UAV swarm converging to the spherical formation shapes avoiding obstacle.

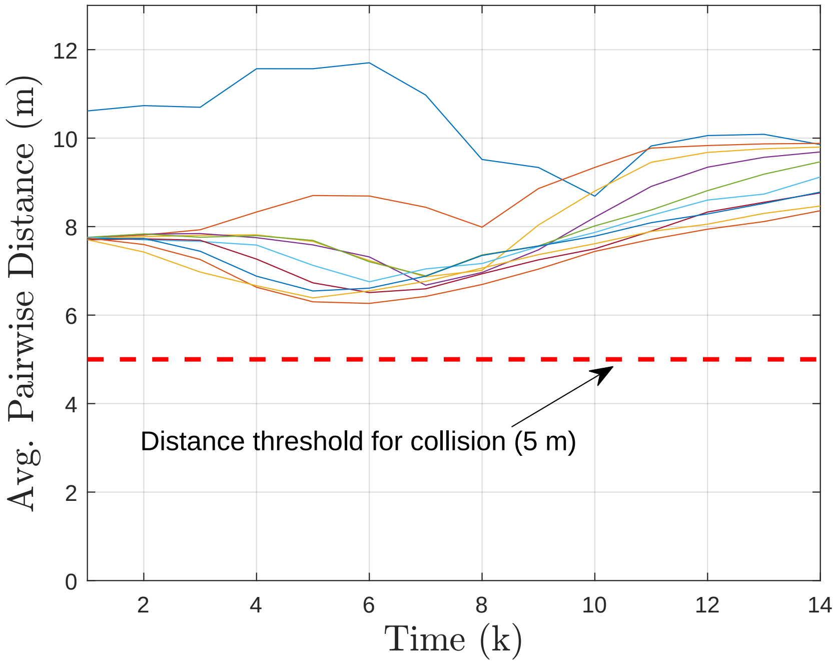

Figure 4. Distance between each pair of UAVs.

We calculate the T_c and T_f values for both the centralized and the decentralized algorithms for 10 UAVs. Figure 5 and Table 1 clearly demonstrate that our decentralized method significantly outperforms the centralized method with respect to both the metrics T_c and T_f.

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

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

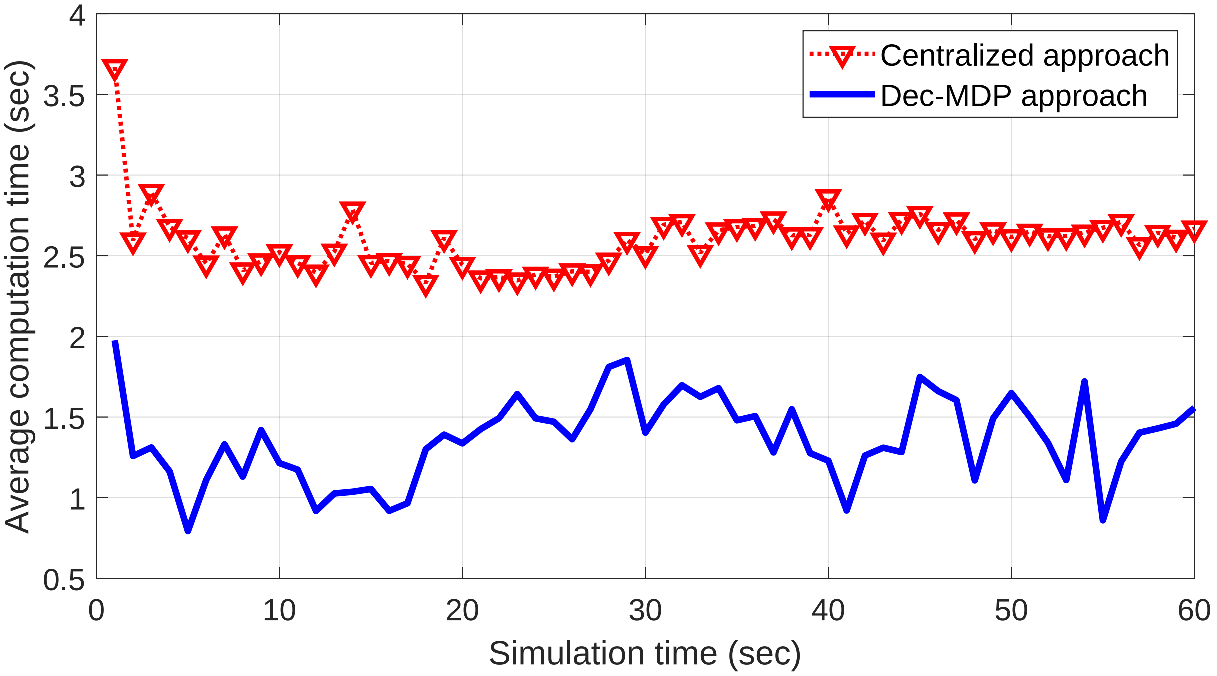

Figure 5. Computation time (T_c): centralized vs. decentralized method.

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 their decisions together with neighbors, which depend on neighborhood threshold and implement its own control. We expect that, with the increase in neighborhood threshold, average computation time rises and, after certain neighborhood threshold, average computation time saturates. Figure 6 shows average computation time rise until neighborhood threshold reach 240 m and then waves between 20 to 25 s.

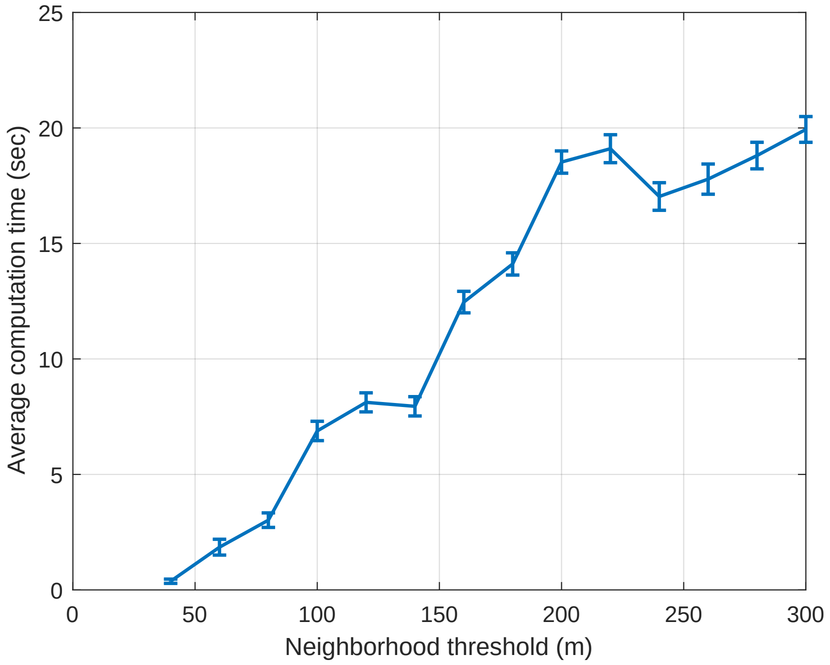

Figure 6. Average computation time with respect to neighborhood threshold.

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 close 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. Figures 6 and 7 suggest that a neighborhood threshold of more than 130 m allows UAVs to stay close in the swarm with reasonable computation cost.

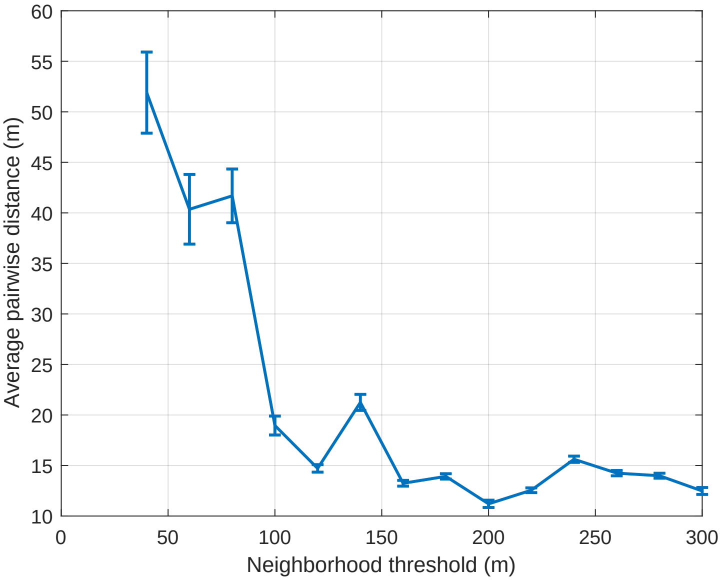

Figure 7. Average pairwise distance with respect to neighborhood threshold.

6. Conclusions

In this paper, we developed decentralized control method for UAVs in the context of formation control. Specifically, we extended a decision-theoretic formulation called *decentralized Markov decision process* (Dec-MDP) to develop near real-time decentralized control methods to drive a UAV swarm from an initial formation to a desired formation in the shortest time possible. As decision-theoretic approaches suffer from the curse of dimensionality, for computational tractability, we extended an approximate dynamic programming method called *nominal belief-state optimization* (NBO) to approximately solve the Dec-MDP. For benchmarking, we also implemented a centralized approach (Markov decision process-based) and compared the performance of our decentralized control methods against the centralized methods. In the context of the formation control problem, our results show that the average computation time for obtaining the optimal controls and the time taken for the swarm to arrive at the formation shape are significantly less with our Dec-MDP approach compared with that of the centralized methods. We also studied the impact of neighborhood threshold on multiple performance metrics in a UAV swarm.

The formation control approach discussed in this thesis can be extended to 3D formation, and these formations can be used to sense the environments for 3D reconstruction of a scene. The vantage points of the UAVs in the swarm in 3D formation can be exploited for the efficient reconstruction of the scene in 3D, while extending tomography-type approaches. The decentralized control strategies presented in this thesis can be extended to control the motion of the UAVs in the swarm to maximize the efficiency of the above 3D scene reconstruction process. These methods have several applications, including the use of drones to map unexplored and unsafe regions (e.g., caves, underground mines, toxic environments).

Author Contributions: Conceptualization, M.A.A., H.D.M. and S.R.; Methodology, M.A.A. and S.R.; Validation, M.A.A. and S.R.; Writing and Editing, M.A.A. and S.R.; and Paper Review, H.D.M. and S.R. All authors have read and agreed to the published version of the manuscript.

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

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

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Waharte, S.; Trigoni, N.; Julier, S. Coordinated Search with a Swarm of UAVs. In Proceedings of the 2009 6th IEEE Annual Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks Workshops, Rome, Italy, 22–26 June 2009; Volume 1109.
2. Walle, D.V.D.; Fidan, B.; Sutton, A.; Yu, C.; Anderson, B.D.O. Non-hierarchical UAV Formation Control for Surveillance Tasks. In Proceedings of the American Control Conference, Seattle, WA, USA, 11–13 June 2008; pp. 777–782.
3. Carthel, C.; Coraluppi, S.; Grignan, P. Multisensor tracking and fusion for maritime surveillance. In Proceedings of the 10th International Conference on Information Fusion, Quebec City, QC, Canada, 9–12 July 2007; pp. 1–6.
4. Shames, I.; Fidan, B.; Anderson, B.D.O. Close Target Reconnaissance using Autonomous UAV Formations. In Proceedings of the 47th IEEE Conference Decision and Control, Cancun, Mexico, 9–11 December 2008; pp. 1729–1734.
5. Vu, Q.; Raković, M.; Delic, V.; Ronzhin, A. Trends in development of UAV-UGV cooperation approaches in precision agriculture. In *International Conference on Interactive Collaborative Robotics*; Springer: Berlin/Heidelberg, Germany, 2018; pp. 213–221.
6. Ragi, S.; Chong, E.K.P. Dynamic UAV Path Planning for Multitargte Tracking. In Proceedings of the American Control Conference, Montreal, QC, Canada, 27–29 June 2012; pp. 3845–3850.
7. Zhan, P.; Casbeer, D.; Swindlehurst, A. A centralized control algorithm for target tracking with UAVs. In Proceedings of the Conference Record of the Thirty-Ninth Asilomar Conference onSignals, Systems and Computers, Monterey, CA, USA, 30 October–2 November 2005; pp. 1148–1152.
8. Qiu, H.; Huang, G.; Gao, J. Centralized multi-sensor multi-target tracking with labeled random finite set. *J. Aerosp. Eng.* **2005**, *231*, 669–676. [CrossRef]
9. Zhao, L.; Ma, D. Circle Formation Control for Multi-agent Systems with a Leader. *Control Theory Technol.* **2015**, *13*, 82–88. [CrossRef]
10. Ragi, S.; Chong, E.K.P. UAV Path Planning in a Dynamic Environment via Partially Observable Markov Decision Process. *IEEE Trans. Aerosp. Electron. Syst.* **2013**, *49*, 2397–2412. [CrossRef]
11. Chong, E.K.P.; Kreucher, C.; Hero, A.O. Partially observable Markov decision process approximations for adaptive sensing. *Disc. Event Dyn. Sys.* **2009**, *19*, 377–422. [CrossRef]
12. Bar-Shalom, Y.; Willett, P.K.; Tian, X. *Tracking and Data Fusion*; YBS Publishing: Storrs, CT, USA, 2011; Volume 11.
13. Shen, D.; Chen, G.; Cruz, J.B.; Blasch, E. A game theoretic data fusion aided path planning approach for cooperative UAV ISR. In Proceedings of the 2008 IEEE Aerospace Conference, Big Sky, MT, USA, 1–8 March 2008; pp. 1–9.
14. Azam, M.A.; Ragi, S. Decentralized formation shape control of UAV swarm using dynamic programming. In Proceedings of the Signal Processing, Sensor/Information Fusion, and Target Recognition XXIX. International Society for Optics and Photonics, Bellingham, WA, USA, 27 April–8 May 2020; Volume 11423, p. 114230I.
15. Das, A.K.; Fierro, R.; Kumar, V.; Ostrowsky, J.P.; Spletzer, J.; Taylor, C. A vision-based formation control framework. *IEEE Trans. Robot. Autom.* **2002**, *18*, 813–825. [CrossRef]
16. Fax, J.A.; Murray, R.M. Information flow and cooperative control of vehicle formations. *IEEE Trans. Autom. Control* **2004**, *49*, 1465–1476. [CrossRef]
17. Ghabcheloo, R.; Pascoal, A.; Silvestre.; Kaminer, I. Coordinated path following control of multiple wheeled robots using linearization techniques. *Int. J. Syst. Sci.* **2006**, *37*, 399–414. [CrossRef]
18. Singh, S.N.; Chandler, P.; Schumacher, C.; Banda, S.; Pachter, M. Adaptive feedback linearizing nonlinear close formation control of UAVs. *Am. Control Conf.* **2000**, *2*, 854–858.
19. Koo, T.J.; Shahruz, S.M. Formation of a group of unmanned aerial vehicles (UAVs). *Am. Control Conf.* **2001**, *1*, 69–74.
20. Edwards, D.B.; Bean, T.A.; Odell, D.L.; Anderson, M.J. A leader–follower algorithm for multiple AUV formations. *IEEE/OES Auton. Underw. Veh.* **2004**, *2*, 40–46.
21. Skjetne, R.; Moi, S.; Fossen, T.I. Nonlinear formation control of marine craft. *IEEE Int. Conf. Decis. Control* **2002**, *2*.
22. Balch, T.; Arkin, R.C. Behavior-based formation control for multirobot teams. *IEEE Trans. Robot. Autom.* **1998**, *14*, 926–939. [CrossRef]
23. Lawton, J.R.; Beard, R.W.; Young, B.J. A decentralized approach to formation maneuvers. *IEEE Trans. Robot. Autom.* **2003**, *19*, 933–941. [CrossRef]
24. Do, K.D.; Pan, J. Nonlinear formation control of unicycle-type mobile robots. *Robot. Auton. Syst.* **2007**, *55*, 191–204. [CrossRef]

25. Lewis, M.A.; Tan, K.H. High precision formation control of mobile robots using virtual structures. *Auton. Robot.* **1997**, *4*, 387–403. [CrossRef]
26. Ragi, S.; Chong, E.K.P. Decentralized Guidance Control of UAVs with Explicit Optimization of Communication. *J. Intell. Robot. Syst.* **2014**, *73*, 811–822. [CrossRef]
27. Kim, Y.; Bang, H. Decentralized control of multiple unmanned aircraft for target tracking and obstacle avoidance. In Proceedings of the 2016 International Conference on Unmanned Aircraft Systems (ICUAS), Arlington, VA, USA, 7–10 June 2016; pp. 327–331.
28. Meng, W.; He, Z.; Su, R.; Shehabinia, A.R.; Lin, L.; Teo, R.; Xie, L. Decentralized control of multi-UAVs for target search, tasking and tracking. *IFAC Proc. Vol.* **2014**, *47*, 10048–10053. [CrossRef]
29. Bakule, L. Decentralized control: An overview. *Elsevier Annu. Rev. Control* **2008**, *32*, 87–98. [CrossRef]
30. Viana, I.B.; Santos, D.A.D.; Goes, L.C.S. Formation Control of Multirotor Aerial Vehicles using Decentralized MPC. *J. Braz. Soc. Mech. Sci. Eng.* **2018**, *40*, 1–12. [CrossRef]
31. Pham, H.X.; La, H.M.; Feil-Seifer, D.; Deans, M. A distributed control framework for a team of unmanned aerial vehicles for dynamic wildfire tracking. In Proceedings of the 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Vancouver, BC, Canada, 24–28 September 2017; pp. 6648–6653.
32. Zhang, Q.; Lapierre, L.; Xiang, X. Distributed Control of Coordinated Path Tracking for Networked Nonholonomic Mobile Vehicles. *IEEE Trans. Ind. Inform.* **2013**, *9*, 472–484. [CrossRef]
33. Miller, S.A.; Harris, Z.A.; Chong, E.K.P. A POMDP framework for coordinated guidance of autonomous UAVs for multitarget tracking. *EURASIP J. Adv. Signal Process.* **2009**, *2009*, 724597. [CrossRef]
34. Schmidt, D. *Modern Flight Dynamics*; McGraw-Hill Higher Education: New York, NY, USA, 2011.
35. Stengel, R.F. *Flight Dynamics*; Princeton University Press: Princeton, NJ, USA, 2015.
36. Kumar, V.; Michael, N. Opportunities and challenges with autonomous micro aerial vehicles. *Int. J. Robot. Res.* **2012**, *31*, 1279–1291. [CrossRef]
37. Michael, N.; Mellinger, D.; Lindsey, Q.; Kumar, V. The grasp multiple micro-uav testbed. *IEEE Robot. Autom. Mag.* **2010**, *17*, 56–65. [CrossRef]
38. Lee, T.; Leok, M.; McClamroch, N.H. Geometric tracking control of a quadrotor UAV on SE (3). In Proceedings of the 49th IEEE Conference on Decision and Control (CDC), Atlanta, GA, USA, 15–17 December 2010; pp. 5420–5425.