IEEE Access
Multidisciplinary ⁝ Rapid Review ⁝ Open Access Journal

Received November 24, 2020, accepted December 6, 2020, date of publication December 11, 2020, date of current version December 28, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3044055

VMCS: Elaborating APF-Based Swarm Intelligence for Mission-Oriented Multi-UV Control

SEONGJOON PARK, (Graduate Student Member, IEEE), HYEONG TAE KIM, AND HWANGNAM KIM, (Member, IEEE)
Department of Electrical Engineering, Korea University, Seoul 02841, Republic of Korea

Corresponding author: Hwangnam Kim (hnkim@korea.ac.kr)

This work was supported by the Unmanned Swarm CPS Research Laboratory Program of Defense Acquisition Program Administration and Agency for Defense Development under Grant UD190029ED.

ABSTRACT This paper addresses a novel approach for multi-agent control systems including Unmanned Vehicles (UV). As UV technology advances, one or a group of UVs can be used in a wide range of industrial or military applications. Centralized monitoring has limitations on various resources (e.g. communication bandwidth, propagation delay, and computational power), but can lead to an optimal solution to the movement of the swarm. The artificial potential field (APF) method is well-known for modeling decentralized behavior, but most APF research only focuses on naive actions such as collision avoidance, flocking, or path planning. In our proposed design Versatile Multi-Vehicle Control System (VMCS), we defined high-level conditions as APF and let UVs perform swarm intelligence in various mission environments. Furthermore, we devised a novel algorithm that controls the UVs' APF topology which can significantly enhance the mission efficiency. We simulated the VMCS in 3D space and showed our scheme can control the dynamic mission scenarios for multi-UV systems.

INDEX TERMS Multi-UV control, network topology, swarm intelligence, artificial potential field.

I. INTRODUCTION

Significant advances in Unmanned Vehicle (UV) technology have increased the possibility of replacing existing industrial components. Research on managing a large number of unmanned vehicles has become a high priority issue. However, in the real world, aligned flocking or shaping an image on the sky [1] (or ground [2]) have been demonstrated only by proactive path-planning, which shows the limitation of multi-UV's empirical application. This lack of practical implementation on a large-scale UV control originates from the difficulty of planning complex paths for UVs that must meet both scalability and optimality.

Commonly, there are two categories to design the multi-UV control system: centralized and distributed. A centralized system places a monitor on the UV network to collect all the data from the UVs and order them directly to carry out their missions. Centralized decision making requires a reduced requirement of the UVs' intelligence and leads to the possibility of finding the global optimum of the solution, but it suffers from the exponentially increased delay of the networking and the decision overhead [3]. In addition, centralized decision making has complex implementation steps because all data are required to make a decision. Collecting, managing, and sending all actions require a fairly complicated process. Sometimes these centralized processes are additionally required to consider time synchronization.

On the other hand, a distributed system yields higher intelligence to individual vehicles, letting each UV understand the current situation and determine its action to complete its mission. This approach does not cost decision delay of UVs since each UV makes a decision by itself. In addition, a distributed system also has strength in computational cost. By simply comparing with the centralized system, it has a lower computational load for decision making. Besides, a distributed system has simplicity in implementation since it does not need to gather all the information on a central station. However, this isolation can lead to the local optima of the solution or increased cost of the system due to the UV's specific requirements (e.g. sensors, networking devices, and computing board).

As one of the distributed multi-agent control mechanism, Artificial Potential Field (APF) has been steadily studied [4]–[7]. APF defines the potential applied to mobile

The associate editor coordinating the review of this manuscript and approving it for publication was Xiaofan He.

UVs and derives location changes based on classical dynamics. APF-based control provides efficient solutions for obstacle avoidance and flock-wise movements such as grouping and herding [8]. However, there are great walls to deploy the APF-based system into the real-life environment.

- Stability was only discussed in the area of mathematical safety. To prove the equilibrium position of the system, large assumptions are made such as UV velocity, available space, and swarm size. Relaxing these assumptions is mathematically challenging, so the APF-based approaches are hard to be directly applied in real-world scenarios.
- Some artificial potentials that require communication between the UVs constrain the network topology because some motion causes them to be undesirably disconnected with their neighbors or backbones.
- Previous APF-based solutions seem only suitable for a few specific applications, such as networking or surveillance.

The unmanned vehicle system can have large functionalities by attaching auxiliary modules (mechanical arms, cameras, lasers, etc.), so the system should return a wide range of actions. For doing so, we propose a novel architecture named *Versatile Multi-Vehicle Control System (VMCS)*, an applicable APF-based multi-UV control system. The core idea of this system is to break away from classical mechanics. In other words, we do not apply the APFs directly into the vehicles, but only *utilize* the direction and magnitude information of the APF calculation. Each vehicle individually calculates its potentials to determine its next operation from the results. In the case of the *move* action, the magnitude of the UV motion is determined by the norm of the potential sum along with the upper bound of the UV's speed.

The former APF-based studies apply each APF directly and derives correspondent actuator control following classical mechanics. This approach helps to shape the UVs' movements smoothly, but there is limitation of action space. The resulting actions of APF analysis are only *movements*, so the APF can only reflect the reason of *movements*. VMCS aims to improve the expandability of APF-based multi-UV control system by additionally mapping procedure letting UVs perform the specific mission. By doing so, VMCS allows the users to design their own mechanism that maps the APF to the actual operations for fulfilling their requirements.

Furthermore, to show the generality of our modeling, we defined a generic structure of the mission that can be applied to various situations. To solve this problem, we designed sample APF models including the potentials each of which maintains the network connectivity, avoids the collision, and performs the mission. To maximize the efficiency of the swarm, we designed an algorithm that controls the network topology. The network topology of the APF-based system is usually predetermined, which make the swarm change its shape *fluidly* by adjusting the potential links. In final, we implemented a whole loop of VMCS, and addressed the numerical studies while comparing with a centralized method by simulation.

Our contribution can be summarized as follows.

- We proposed a scheme to decompose the APF derivation and the operation mapping. Unlike the previous studies utilizing APF as a mean of the devices' movement, our scheme allows the APF to determine *any* types of the devices' operation, such as positioning, power control or networking. In this paper, we simulated our scheme in the multi-UV scenario.
- We proposed a general model of the mission, and designed a set of APFs that models a swarm of UVs to perform a mission. We proved the stability of the APFs in Section IV-D, and evaluated the performance in Section V.
- Overall, we conducted the simulation of our proposed APF-based control scheme in 3D space. From the simulation results, we derived the numerical trends with respect to the system or environmental factors, and showed the persistence of our APF modeling.

The rest of this paper is organized as follows. Section II introduces some researches on multi-UV control systems, and Section III explains the general concept of VMCS. Section IV shows VMCS design including the APF models, topology control algorithm and stability analysis. Section V evaluates the resulting system by numerical simulation with discussions. Section VI concludes the paper.

II. RELATED WORK

As mentioned earlier, the study of using multiple vehicles to perform a variety of missions while considering the issues of optimality and complexity is challenging. Various approaches have been used to study multi-UV control considering the challenge, but studies in distributed control systems have been actively conducted with visible results.

Distributed multi-UV control system: There were several attempts to design the trajectories of multiple mobile agents within constrained environments. Morgan *et al.* [9] proposed a model-predictive control-sequential convex programming algorithm guiding a swarm of spacecrafts. This study assumes communicationally and computationally constrained agents that need to migrate to the other orbit while maintaining the communication with nearby spacecraft and updating the optimal trajectory. Bandyopadhyay *et al.* [10] presented an approach using distributed optimal transport to probabilistic guidance of a swarm of autonomous agents. The feature of this work is an optimal transport algorithm which guarantees faster convergence. Bandyopadhyay *et al.* [11] showed a more general design of distributed control, targeting the formation of large swarms. By adopting Eulerian density-control scheme, each autonomous swarm is probabilistically controlled to handle the loss or addition of the agents. Yu and LaValle [12] solved the path planning problem of multiple robots with minimizing several overheads (arrival time, traveled distance, and so on). The authors exploited the mapping between the problem and the multi-flow network.

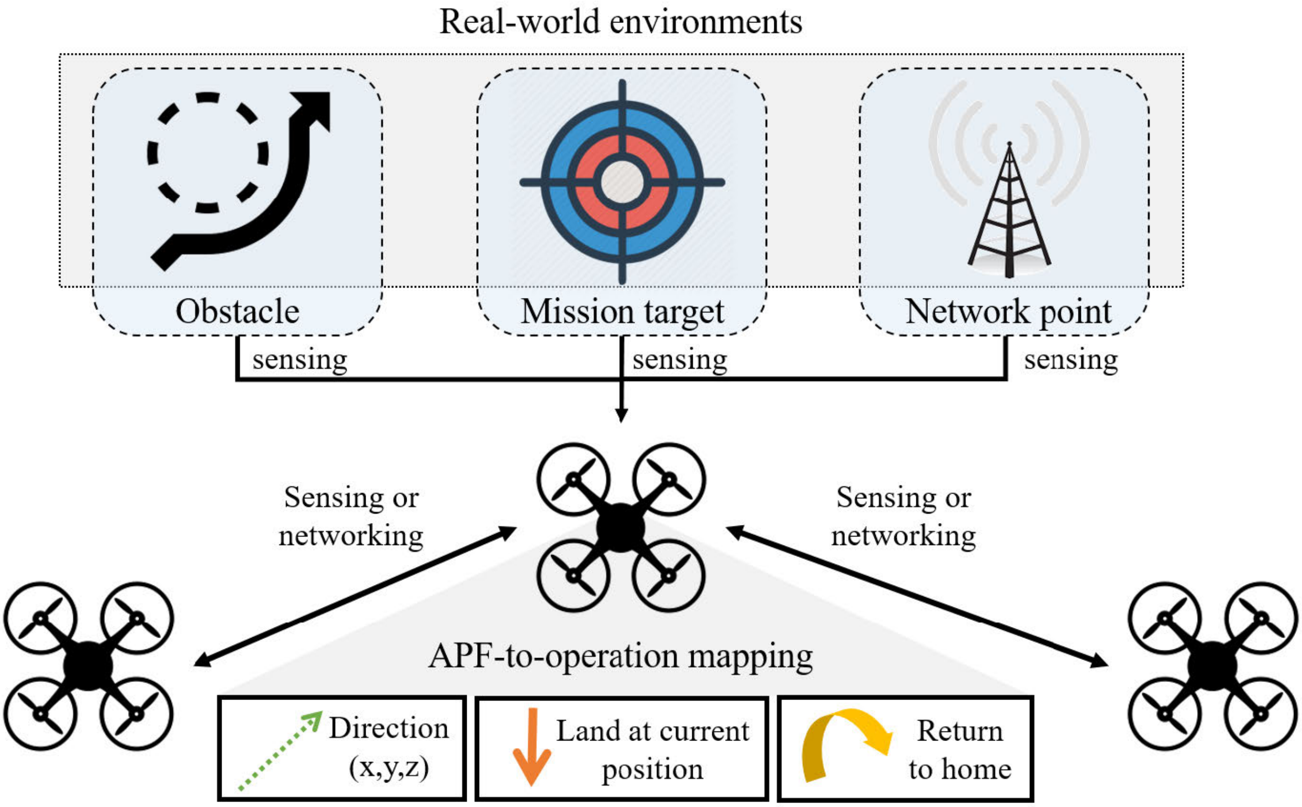


FIGURE 1. VMCS design.

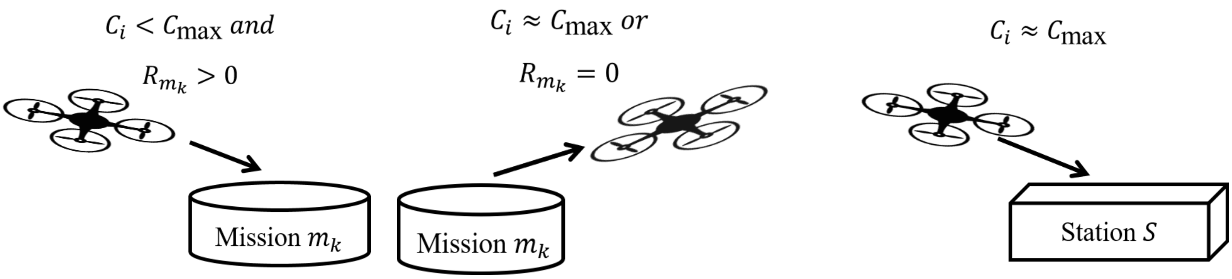
Then, they made an integer linear programming to efficiently solve the issue. The common point from these researches is that the solution is only appropriate for a specific type of environment. In detail, many of these researches only concern and apply their system on their specific conditions, which mostly does not justify the generality on normal multi-UV scenarios. We aimed to operate our multi-UV system in unknown, time variable environment with considerable stability.

APF-based path planning: APF-based mobility control has been studied in decades, from the theoretical designs to empirical approaches. Chen *et al.* [13] and Shah and Vachhani [14] defined attractive and repulsive models between an Unmanned Air Vehicle (UAV) and the target point and solved the optimal trajectory problem on the 3D obstacle map. Huang *et al.* [15] showed APF-Elaborated Resistance Network (APFE-RN) approach to control the 2D vehicle in a road while dividing the planning process into the virtual planning and actual control. Zhou *et al.* [16] solved the local minima problem of APF-based path planning by particle swarm optimization with the tangent vector. APF-based approaches have analyzed the path planning of the robot control with various mathematical tools to reach the optimality, but most of them were focused to achieve collision avoidance in a given obstacle map. In our system, we propose a way to increases the diversity and scalability of the mission by equating the mission efficiency with the basic requirements of the swarm.

III. GENERAL CONCEPT OF VMCS DESIGN

Designing mathematical APF, deploying one or more vehicles and observing their trajectories have been actively conducted in multi-UV control domain [13], [15], [17]. However, drawing APF for the entire map is infeasible in real environments due to the large number of the obstacles and their unlimited shapes. Furthermore, it is often questionable that the UV can fulfill such mobility in practical environments. Our scheme bridges the gap between the APF modeling and the empirical swarm control by abstracting the APFs' roles and then mapping their influences.

Fig. 1 shows the overview of our scheme at a glance. The overall flow of our system follows the general objective of swarm control concept: decentralization and simplification of the decision, and the APF derivation [18]. Each UV senses its surroundings, which may result in the imperfect derivation of the potential vectors. However, we claim that this approach is reasonable in the practical scenario with the possibility of the unknown, mobile obstacles. The unprecedented component proposed in our scheme is called *vector-to-operation mapping*. UVs refer the sum vector of the APFs, and map to the specific operation which may decrease the magnitude



(a) UV enters mission target when available.

(b) UV leaves mission target when unavailable.

(c) UV visits a specific station when capacity is full.

FIGURE 2. Generic mission model.

of the vector. For example, in the case of vector-to-direction mapping can be expressed as

$$\mathcal{D}(x, y, z) = G \times \left(\frac{\mathbf{A}}{|\mathbf{A}|}\right),$$
$$\mathbf{A} = \sum_{\mathbf{f} \in F} \mathbf{f} \ \ and \ \ G = min(|\mathbf{A}|, V_{max}) \qquad (1)$$

where V_{max} refers to the maximum scale of $\mathcal{D}$, $\mathbf{A}$ refers to the total sum of APFs ($\mathbf{f}$), F refers to the set of the APFs defined in VMCS, and G refers to the gain value that applies to the UV's final movement. If we let τ refer to the decision period of each UV, then V_{max} should be set as V_{phymax}/τ, where V_{phymax} refers to the physical maximum velocity of the UV. Consequently, the factor G implicitly represents the UV's velocity, and the factor $\mathbf{A}/|\mathbf{A}|$ represents the UV's direction.

The motivation of our scheme can be summarized into two factors: stability and extensibility. Note that the APFs defined in the existing multi-UV system are *artificial*, so the control modules equipped in the UVs must imitate the force obtained by the fields to balance the whole system. In the vector-to-operation mapping sequence described in Fig. 1, only the commands containing the next destination are determined for the vehicles. This strategy lessens the burden of the control module and contributes to the stability discussed in Section IV-D. Furthermore, existing APF-based approaches can be expanded to more complex swarm, including UAV and Unmanned Ground Vehicle (UGV) with our scheme. To show the generality of our scheme, we assume a simple but largely-applicable scenario where UVs need to perform a set of *missions*. We introduce the detail of the mission and derived VMCS in Section IV-B, and show the performance in Section V.

IV. DESIGNED SYSTEM

This section addresses the detailed design of VMCS. We define the simple APF models utilized in the control decision for the mission performance. Also, we propose a topology control algorithm that dynamically controls the topology of the network to manage the constraints produced by the potentials.

A. MISSION MODEL

To guarantee the applicability of the resulting APFs, we designed a generic mission model for swarming UVs as shown in Fig. 2. The mission model is composed of a set of mission targets M, and a set of UVs U, and a station S used for a return point. Each mission target is referred to m_k $(1 < k \leq |M|)$ and holds a specific amount of the mission load R_{m_k} where $R_{m_k} \geq 0$. Each UV is referred to u_i $(1 < i \leq |U|)$ and carries a specific amount of the mission load C_{u_i} where $0 \leq C_{u_i} \leq C_{max}$, and C_{max} refers to the maximum capacity to carry the result. We assume that each UV knows C_{max} in priori, and has ability to periodically check its remaining capacity. A UV can approach to a mission target, and transport the amount of the mission load from the target to itself at nearby of the target within a pre-defined short distance (Fig. 2a). If the carrying amount of a UV reaches to C_{max}, the UV cannot transport the mission load (Fig. 2c), but it can reset its load at the station (Fig. 2b). The objective of the mission is to collect all load of the scattered mission targets by using a given group of UVs. Furthermore, we apply the connectivity constraint to the mission, so UVs should be connected to a *gateway*, which is in a fixed position.

When the mission starts, a number of the mission targets are scattered in a 3D space with randomized R_{m_k}. Then, a gateway is located at the center of map, and the UVs are randomly located at nearby of the gateway. The UVs know the position of the gateway a priori. Through the mission operation, the UVs and the mission targets periodically broadcast the position of each. The mission is marked as completed after all mission targets are carried by the UVs.

Our proposed scenario can be mapped to a wide range of the swarm applications. For delivery scenario (gathering courier boxes to a station), R_{m_k} refers to the weight of the boxes on the target, C_i refers to the current amount loaded to the i-th UV, then C_{max} can indicate the maximum payload of each UV. In IoT scenario using UAV as a sink [19], R_{m_k} refers to the accumulated data size collected by the sensors around the region, C_i refers to the allocated space of the storage device equipped in i-th UV, then C_{max} can indicate the available maximum data size the UAV can store. For more biological cases, if R_{m_k} is the amount of honey contained in a flower and C_i are the storing amount of i-th bee, then we can design the APF of the flying robots acting as a swarming worker bees.

B. APF MODELS

We propose a mathematical model of APFs that regulates the distributed behavior of multi-UV swarm. Each APF depicts the influence needed to be applied to the UV and induce its position to fulfill the given mission, avoiding unexpected damage or uncontrollable state. We derive $\mathbf{A}$ as a sum of the specific APFs, expressed to

$$\mathbf{A} = G_c\mathbf{F_c} + G_m\mathbf{F_m} + G_n\mathbf{F_n} \tag{2}$$

where $\mathbf{F_c}$ refers to the *collision avoidance potential*, $\mathbf{F_m}$ to the *mission performance potential*, $\mathbf{F_n}$ refers to the *network connectivity potential*, and G_c, G_m and G_n refer to the gain of each potential, respectively. $\mathbf{F_c}$ grants the repulsive force from the nearby obstacles, and $\mathbf{F_m}$ applies the attractive force from the mission targets or the station. $\mathbf{F_n}$ gives the attractive force from the wireless links. Recall that the result of the APF derivation is not directly applied to the UVs' kinematics. Rather, the mapped commands are conducted by its own control module. In addition to the potentials, we consider the case that the UVs are desired to connect to the backbone network since UVs may need to communicate with the operator while performing their mission.

The following subsections describe the aforementioned potential factors in detail. For simplicity, we let P_i and C_i refer to the position and the current load of the u_i, respectively.

1) COLLISION AVOIDANCE POTENTIAL

As most of the existing approaches claimed, UV needs to be away from the obstacles whether they are stationary or mobile. For any detected obstacle including the other UVs, we modeled the collision avoidance potential $\mathbf{f_c}(o_j)$ with the opposite direction to an obstacle (indexed to j), which can be calculated by

$$\mathbf{f_c}(o_j) = -\log\left(\frac{|\mathbf{P_{o_j}} - \mathbf{P_i}|}{D_c}\right)\frac{\mathbf{P_i} - \mathbf{P_{o_j}}}{|\mathbf{P_i} - \mathbf{P_{o_j}}|} \tag{3}$$

where $\mathbf{P_{o_j}}$ refers to the position of the obstacle o_j, and D_c is the maximum obstacle detection distance. Eq. (3) shows the exponential drop when $|\mathbf{P_{o_j}} - \mathbf{P_i}| < D_c$, to let $\mathbf{f_c}(o_j)$ be dominant when the UV approximates to an obstacle.

Note that we assume a case that the detailed map containing the obstacles' position is not given previously. It means, the *surface* of the obstacles can be detected by sensors such as LIDAR, depth camera or any other sensing module. If we let O refer to a set of the detected obstacles within D_c, then we can derive the aggregated obstacle avoidance potential applied to a UV by

$$\mathbf{F_c} = \sum_{o_j \in O} \mathbf{f_c}(\mathbf{o_j}). \tag{4}$$

In addition, O includes a set of UVs E, where $E \subset U$ and $|\mathbf{P_e} - \mathbf{P_i}| < D_c$ for $e \in E$. The positions of the UVs are broadcast as stated in Section IV-A, so the UVs apply repulsive force to each other.

2) MISSION PERFORMANCE POTENTIAL

We modeled $\mathbf{F_m}$ as the aggregated potential of the partial potentials $\mathbf{f_m}$ attracting to the mission targets, and $\mathbf{F_s}$ attracting to the station. The APF of $\mathbf{f_m}$ is calculated to

$$\mathbf{f_m}(m_k) = (D_m - |\mathbf{P_i} - \mathbf{P_{m_k}}|)\frac{\mathbf{P_{m_k}} - \mathbf{P_i}}{|\mathbf{P_{m_k}} - \mathbf{P_i}|} \tag{5}$$

where $\mathbf{P_{m_k}}$ refers to the position of the mission target and D_m refers to the maximum range of the mission target to be attracted, named to *Mission Attraction Range*. The UVs obtain $\mathbf{P_{m_k}}$ from m_k itself, through the position broadcasting mentioned in Section IV-A. $\mathbf{F_s}$ is calculated to

$$\mathbf{F_s} = (e^{C_i - C_{thresh}} - 1)\frac{\mathbf{P_S} - \mathbf{P_i}}{|\mathbf{P_S} - \mathbf{P_i}|} \tag{6}$$

where $\mathbf{P_S}$ refers to the position of the station and C_{thresh} refers to the load threshold of the UV. As shown in Eq. (5), the scale of the potential increases when the mission target is at closer distance. Inverse relationship between the distance and the potential leads to the efficient formation change of the UVs, where each UV approaches to the nearby target. The potential from the station described in Eq. (6) applies when the load of the UV reaches above a threshold C_{thresh}. On our implementation in Section V, we experimentally set C_{thresh} to $0.9C_{max}$.

With this design, UV will be strongly attracted to a station. As regulated in the mission model, the UV filled with the mission load ($C_i \approx C_{max}$) cannot continue the mission. We designed $\mathbf{F_S}$ to fulfill this restriction under any circumstances, which temporarily breaks the UV's stability until C_i resets to 0. Section IV-D addresses the stability of the system in detail.

In summary, $\mathbf{F_m}$ can be calculated to

$$\mathbf{F_m} = \sum_{m_k \in M_i} \mathbf{f_m}(m_k) + \mathbf{F_S} \tag{7}$$

where M_i refers to the set of the remaining mission targets detected by the UV u_i, which means $M_i = \{m_k | (|\mathbf{P_i} - \mathbf{P_{m_k}}|) < D_m, R_{m_k} \neq 0, m_k \in M\}$.

3) NETWORK CONNECTIVITY POTENTIAL

As mentioned earlier, the swarm often needs to maintain the network between the nodes to perform its mission. The potential introduced in this subsection is optional, but we attempted to design a generic one to be able to be utilized at various types of the network topologies. We adopted a graph-based approach to this potential [15] to guarantee the network connectivity of UVs. Let a set N contains the swarm's backbone gateways and the UVs. These backbone gateways can make the UV ad-hoc network to communicate with outer networks. If we let H be the set of the *available* next hops to connect to the backbone network, which satisfies $H \subset N$, then the

Algorithm 1 Topology Control Algorithm

```
1: $H \leftarrow \{g\}$
2: $c \leftarrow 1$
3: while True do
4:   if $|\mathbf{F_n}| > |\mathbf{F_m}| + |\mathbf{F_c}|$ then
5:     if there is a gateway $g$ within the distance $D_n$ then
6:       $c \leftarrow 1$
7:       $H \leftarrow \{g\}$
8:       return
9:     end if
10:    Get a set of the UVs $E$ within distance $D_n$
11:    if $e \in E$ satisfies $c_e < c$ then
12:      $H \leftarrow E$
13:      $c \leftarrow \min(c_e) + 1$
14:    end if
15:  end if
16: end while
```

network connectivity potential $\mathbf{F_n}$ can be simply expressed to

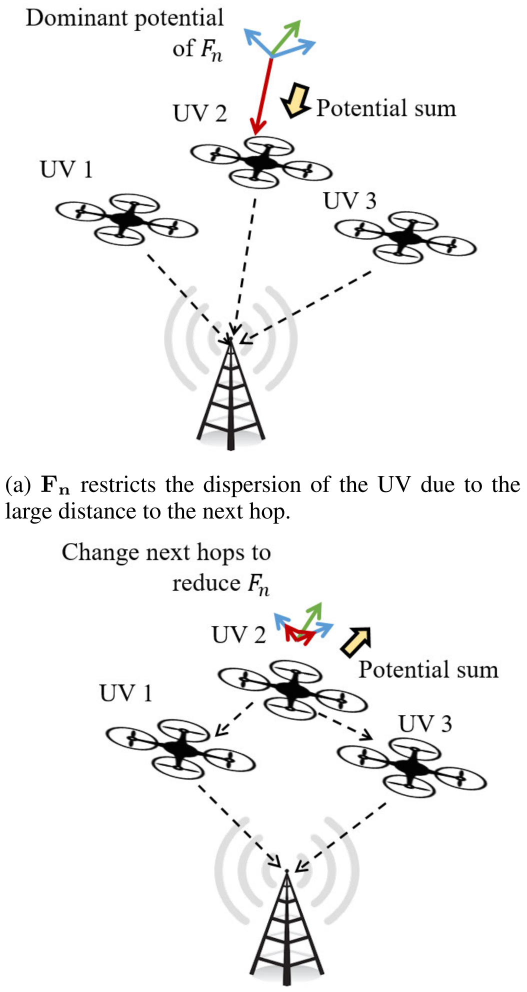
$$\mathbf{F_n} = \sum_{h \in H} e^{|\mathbf{P_h} - \mathbf{P_i}| - D_n} \frac{\mathbf{P_h} - \mathbf{P_i}}{|\mathbf{P_h} - \mathbf{P_i}|} \tag{8}$$

where $\mathbf{P_h}$ refers to the position of the next hop device (or gateway), and D_n refers to the threshold communication range between the wireless links. The UVs obtain $\mathbf{P_h}$ from h itself, through the position broadcasting mentioned in Section IV-A. The magnitude of $\mathbf{F_n}$ increases when the distances from the next hops get larger. Note that $\mathbf{F_n}$ plays the same role with the Reynold's second rule [20], since the UVs follows the nearby other UVs to maintain the network. H is firstly set to the one that includes only the gateway and then modified by the *topology control algorithm*, which detects the possibility to the expansion of the network and attempts to loosen $\mathbf{F_n}$. Section IV-C describes this algorithm in detail.

C. TOPOLOGY CONTROL

Network connectivity guarded by $\mathbf{F_n}$ (Eq. (8)) can result in the undesirable form of swarms if H is fixed. Pre-installing H for each UV can shape the required formation of the swarm, but it brings complex design of the topology and the restricted structure of the swarm. Changing H should be elaborately designed due to the risk of losing the connectivity to the gateway while swarming in the 3D space. The main reason of the difficulty in determining H originates from the independency of the UV's control decision. To guarantee the connectivity to the backbone network, at least one UV needs to be linked within one hop. Thus, changing H must be considered in a group of the nodes. In this section, we propose a novel topology control algorithm based on the APF calculation to make the UVs' movements flexible without connection loss. The main breakthrough of our algorithm is the definition of the H change condition with respect to the relationship of the potentials' magnitude.

Algorithm 1 describes our topology control sequence. Fig. 3 graphically represents the situation where three UVs are connected to the wireless gateway. The red, blue and green arrows on each subfigure represent the potentials $\mathbf{F_n}$, $\mathbf{F_c}$ and $\mathbf{F_m}$, respectively. The yellow bordered arrow represents the direction of the potential sum $\mathbf{A}$. At first, the network starts with the star topology where all UVs refer its next hop to the gateway (Fig. 3a). A variable c refers to the expected hop count of the routing path to the backbone network. In the case of Fig. 3a, UV1 to UV3 have the same value of $c = 1$. Due to the far distance from the gateway, UV2 suffers from the large $\mathbf{F_n}$ and cannot move further to the mission target. If the potentials in UV2 satisfy the condition described in line 4, the APFs imply that the network connectivity potential interrupts the proper deployment of the swarm. Thus, UV2 scans the nearby neighbors which are within its wireless communication range and confirms if one of them guarantees the routing path to the backbone (line 11). If possible, UV2 sets c to the minimum c of the neighbors plus one, and change the next hop to E. Then $\mathbf{F_n}$ is changed and the UV2 can move forward to its desired direction 3b. The resulting value of UV2's c is changed to 2,



(a) $\mathbf{F_n}$ restricts the dispersion of the UV due to the large distance to the next hop.

(b) UV2 changes H and lessen the $\mathbf{F_n}$ to move further.

FIGURE 3. **Topology control example.**

and the rest of the UVs hold their H if all of nearby UVs are $c = 2$.

There are some additional discussions for the topology control algorithm.

- One may wonder why all nearest neighbors are in H when changing the set of next hops. This strategy originates from the philosophy of the network redundancy [21]. As shown in Algorithm 1, any UV in the network can change its c by its nearby UVs. Including all neighbors can increase the robustness by the opportunity to reduce the expected hop count of the UV. However, too intensive redundancy can rather highly restrict the movement of the UVs due to the large F_n after changing the H. On our implementation, we limited the maximum $|H|$ to 2 for flexible swarming.
- From line 5 to 8, we grant the chance to connect to the gateway within a single hop by resetting H and c. This recurrent logic helps to reduce the maximum hop count of the network and routing overhead. However, if all UVs try to reconnect to the gateway, it might be difficult for the formation to escape the star topology. On our implementation, we let UVs try to reconnect to the gateway with 5% probability per loop.

D. STABILITY ANALYSIS

This section analyzes the movement of the VMCS designed at Section IV-B. We adopted the Lyapunov direct method [22] to show the existence of the equilibrium point. At first, we show the stability of a UV in the multi-UV environment, as shown in Fig. 4. In this analysis, we abstract all other UVs and their total APFs into one virtual UV and its APF, and represent all the remaining missions with one virtual mission, as shown in the figure. This abstraction is in the extended line of modeling in Section IV-B and makes the entire scheme intuitively understood. As shown in the figure, a UV changes the direction to the sum of the APFs $\mathbf{F_c}$, $\mathbf{F_m}$ and $\mathbf{F_n}$ while moving to the direction $\mathbf{A}'$, induced by the previous control loop. If a UV reaches an equilibrium point, the resulting operation (Eq. (1)) derived by the APFs let the UV back to the previous position, which can be expressed as

$$\mathbf{A}' = \mathbf{F_c} + \mathbf{F_m} + \mathbf{F_n}. \tag{9}$$

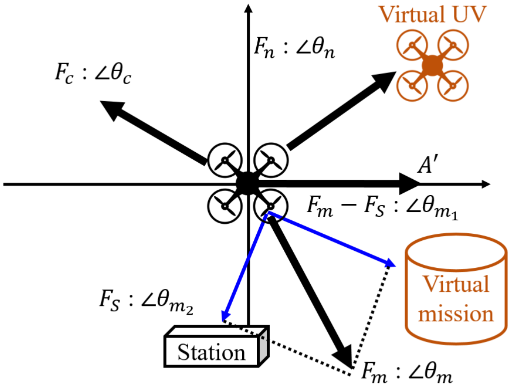


FIGURE 4. APFs applied on UV.

If we divide the above equation into vertical and horizontal components with θ_n, θ_c and θ_m shown in Fig. 4, the result can be expressed to

$$\begin{aligned} \mathbf{F_n}\cos\theta_n + \mathbf{F_c}\cos\theta_c + \mathbf{F_m}\cos\theta_m &= -wa \\ \mathbf{F_n}\sin\theta_n + \mathbf{F_c}\sin\theta_c + \mathbf{F_m}\sin\theta_m &= 0 \end{aligned} \tag{10}$$

where w, a refers to the mass and the acceleration of the UV, respectively. We separately prove the stability of horizontal and vertical components.

For the convenience on the following stability analysis, we express the potentials as

$$\begin{aligned} \mathbf{F_n} &= \alpha(t)\,\mathbf{u_n}(t), \quad \mathbf{F_c} = \beta(t)\,\mathbf{u_c}(t), \; and \\ \mathbf{F_m} &= \gamma_1(t)\mathbf{u_{m_1}}(t) + \gamma_2(t)\mathbf{u_{m_2}}(t) \end{aligned} \tag{11}$$

where $\alpha(t)$, $\beta(t)$, $\gamma_1(t)$ and $\gamma_2(t)$ refer to the magnitude of the $\mathbf{F_n}$, $\mathbf{F_c}$, $\mathbf{F_m} - \mathbf{F_S}$, and $\mathbf{F_S}$, respectively. $\mathbf{u_n}(t)$, $\mathbf{u_c}(t)$, $\mathbf{u_{m_1}}(t)$ and $\mathbf{u_{m_2}}(t)$ refer to the unit vectors directing each potential. To derive the decisive factor from each potential, we revisit Eqs. (4)-(8).

$$let\ r_0(t) := |\mathbf{P_i} - \mathbf{P_{u_j}}| - D_n \ \text{ w.r.t. } t, \quad \alpha(t) = e^{r_0(t)} \tag{12}$$

$$let\ r_1(t) := \frac{|\mathbf{P_i} - \mathbf{P_{u_j}}|}{D_c} \ \text{ w.r.t. } t, \quad \beta(t) = -\log r_1(t) \tag{13}$$

$$let\ r_2(t) := D_m - |\mathbf{P_i} - \mathbf{P_{m_k}}| \ \text{ w.r.t. } t, \quad \gamma_1(t) = r_2(t) \tag{14}$$

$$let\ r_3(t) := C_i - C_{thresh} \ \text{ w.r.t. } t, \quad \gamma_2(t) = e^{r_3(t)} - 1 \tag{15}$$

From the Eqs. (12)-(15), we rewrite horizontal component of Eq. (10),

$$\begin{aligned} w\ddot{x} + e^{r_0(t)}\cos\theta_n x - \log r_1(t)\cos\theta_c x \\ + r_2(t)\cos\theta_{m_1} x + r_3(t)\cos\theta_{m_2} = 0 \end{aligned} \tag{16}$$

where x refers to the horizontal displacement of the UV from the equilibrium point. We define the Lyapunov function $V(x, \dot{x})$ as

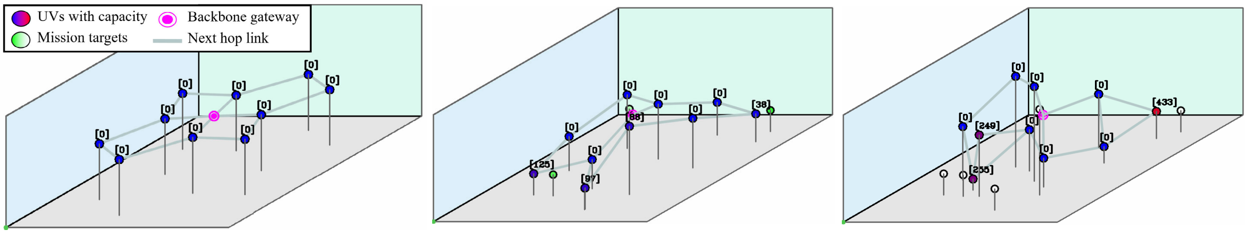
$$V(x,\dot{x}) := \frac{1}{2}w\dot{x}^2 + \frac{1}{2}\left(e^{r_0(t)} - \log r_1(t) + r_2(t)\right)x^2 + r_3(t)x. \tag{17}$$

Then, the derivative of $V(x, \dot{x})$ is

$$\begin{aligned} V'(x,\dot{x}) &= \dot{x}(w\ddot{x} + e^{r_0(t)}x - \log(r_1(t))x + r_2(t)x + r_3(t)) \\ &+ \frac{1}{2}\left(e^{r_0(t)}\dot{r}_0(t) - \frac{1}{r_1(t)}\dot{r}_1(t) + \dot{r}_2(t)\right)x^2 + \dot{r}_3(t)x. \end{aligned} \tag{18}$$

As aforementioned in Section IV-B2, we investigate the case with $C_i < C_{thresh}$. In Eq. (17), since $r_1(t) < D_m$ in Eq. (7), $V(x, \dot{x})$ is positive definite, with equilibrium point $V(0, 0) = 0$. Also, in Eq. (18), the first term is negative since

$$\begin{aligned} & w\ddot{x} + e^{r_0(t)}x - \log r_1(t)x + r_2(t) + r_3(t) \\ & \geq w\ddot{x} + e^{r_0(t)}\cos\theta_n x - \log r_1(t)\cos\theta_c x \\ & \quad + r_2(t)\cos\theta_{m_1} + r_3(t)\cos\theta_{m_2} = 0 \end{aligned} \tag{19}$$



(a) Initial formation with F_n, F_s. (b) Formation in mission with F_m. (c) Final formation after mission ends.

FIGURE 5. Simulation of the proposed APF-based scheme (3D view).

from Eq. (16), and $\dot{x} < 0$ due to the equilibrium. Also, $\dot{r_0}(t) < 0, \dot{r_1}(t) > 0, \dot{r_2}(t) < 0$ from the design of the APFs (Section IV-B), and $\dot{r_3}(T)$ is negligible since $C_i < C_{thresh}$. Thus, $V'(x, \dot{x})$ is negative definite, and the control system of a UV satisfies *Lyapunov stability* in horizontal. Similarly, the system also satisfies Lyapunov stability in vertical since there is only difference in the term $w\ddot{x}$, which results in the same positive/negative definite of the Lyapunov function and its derivative. In general, each UV settles to its equilibrium point while operating, so we can conclude that the multi-UV system controlled by our APF models satisfies Lyapunov stability, which means that there are *equilibrium formations* that the UVs can hold. The position might some cases fall into local minima, however, its formation still satisfies all the constraints and does not violate the overall constructing algorithm in the system. Therefore, we are confident that the VMCS will work stably without any form of failure or divergence.

V. EVALUATION

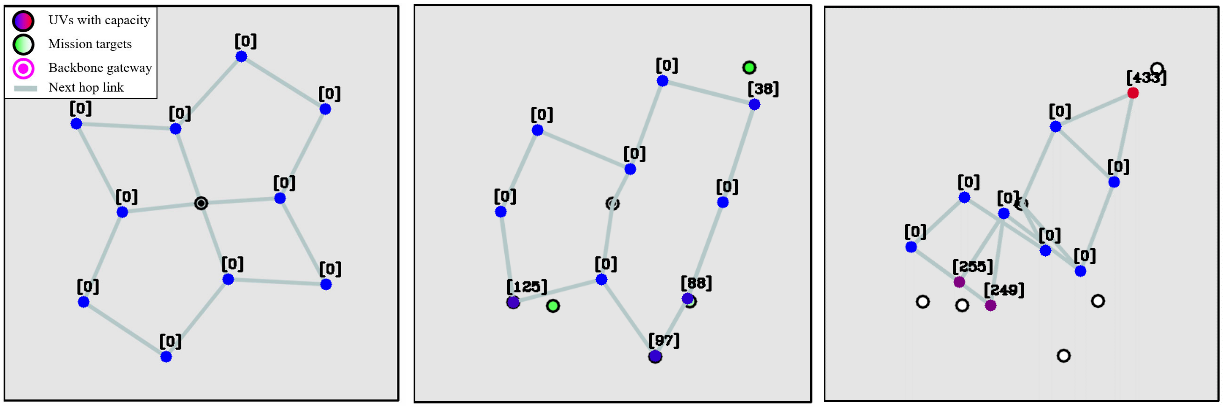
We simulated the VMCS (Section III, IV) by implementing the scheme and APFs in Python. To show our results, we implemented a visualizer as an auxiliary using OpenCV. We simulated our scheme with our APFs using the visualizer. The details on the implementation are as follows.

- We defined a 3D space map with size $100m \times 100m \times 50m$ (width × length × height), and the objects introduced in Section IV-A, such as mission targets, UVs and gateways.
- UVs have 3D mobility in the simulation space. At first, UVs are randomly deployed nearby the gateway. For every 10 ms, each UV calculates $\mathbf{F_c}$, $\mathbf{F_m}$, and $\mathbf{F_n}$ by Eqs. (4)-(8), and determines its next destination from Eq. (1) per loop, where $F = \{\mathbf{F_c}, \mathbf{F_m}, \mathbf{F_n}\}$. Maximum velocity of UVs V_{max} is 20 m/s.
- Mission targets are randomly spread in 3D space, 2s after the drones are created. Mission load R_{m_k} follows uniform random distribution within a range $(100, 500)$ in default. If a UV approaches nearby the mission target, the mission load is transported by 100 units/s. Likewise, if a UV approaches nearby the station, UV immediately releases its load.
- We propose a main evaluation metric as *mission completion time*, which refers to the time the UVs transport all loads of the mission target. In case of the mission failure (due to the unreachable mission target), we set the timeout to 60 s to finish the simulation. From Section V-B to V-D, distribution of mission completion time is measured for each simulation. Additionally, we introduced other metrics each of which is covered in the subsections.

Fig. 5a shows the stabilized formation of 10 UVs where $W_{max} = 30$, before mission targets spawned. As shown in the figure, the swarm forms such a regular shape maintaining similar distances between one another, where $|F_n + F_c| \approx 0$. We claim that this regular shape is advantageous when the mission targets are randomly deployed since the UVs can react quickly in any direction. Also, the UVs are not gathered unnecessarily, so it can detect the targets in further distances. Fig. 5b shows the formation after 5 mission targets deployed. To enhance the visuality of the simulation, we separated the UV icon colors from blue to red: when capacity is almost full, the color changes to red. We captured the simulator right after the mission is completed, where $\sum_{k \in M} R_{m_k} = 0$. As shown in the figure, UVs move to the mission targets while maintaining the network connectivity. Also, we observed that the movements of the UVs are stabilized after all mission requirements reach to 0, as shown in Fig. 5c. Additionally, the numbers denoted to each UVs in the simulation represents the ID number and its current load. The video of our simulation is available at [23].

Note that the APF models shown in Eqs. (4)-(8) are relatively simple and rough, but these potentials cannot dramatically harm the formation due to the mapping procedure described in Eq. (1). This mapping procedure not only contributes to the stability of the system, but extends the area of recognition modeled as APF. From the simulation, we showed that our scheme described in Section III can be utilized to implement practical APF-based systems. Also, we proved that our APF models addressed in Section IV-B stably shape the formation of the UVs, while maintaining the connectivity to the gateway by Algorithm 1.

Fig. 7 graphically represents the trajectories of 10 UVs, with $|M| = 5$. In each figure, green circle indicates the mission target, and magenta circle indicates the gateway. Before the mission targets are scattered, the formation of UVs extend out from the center due to the $\mathbf{F_c}$. After the targets are created,

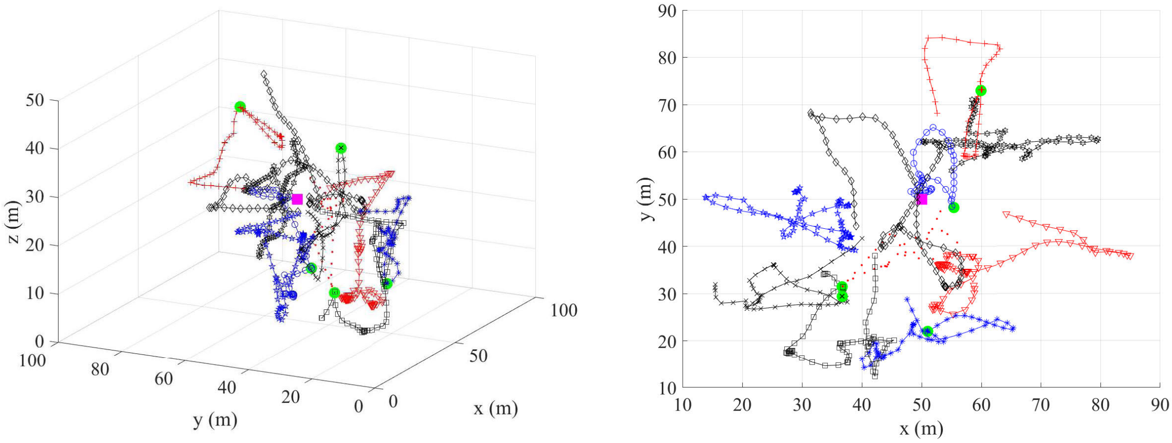


(a) Initial formation with F_n, F_s.

(b) Formation in mission with F_m.

(c) Final formation after mission ends.

FIGURE 6. **Simulation of the proposed APF-based scheme (2D view).**



(a) Trajectories of 10 UVs (3D view).

(b) Trajectories of 10 UVs (2D view).

FIGURE 7. **Trajectories of 10 UVs.**

5 UVs successfully approach to the mission target, whose position is nearby to one of targets. The other 5 UVs firstly move towards the nearest target, but already arrived UVs apply repulsive APF to them. From this figure, we showed that VMCS forms the formation of UVs for mission in a distributed way.

We also investigated the possibility of the collision between UVs and this is represented in Fig. 8. We collected the minimum distance from each UV to the other UVs over time. As shown in the figure, UVs linearly spread for first 1s, and shows variable distances after the mission target spawned. However, until the mission completed, the minimum distance for each UV does not drop into a certain value, which is measured as about 8.5092 m. This investigation shows that UVs movement derived by VMCS maintains the distance between the UVs, by $\mathbf{F_c}$.

The following subsections show numerical evaluations in various cases.

A. COMPARISON WITH CENTRALIZED SCHEME

To justify the usability of the system, we compare VMCS results with another scheme. However, it is notable that VMCS system aims the swarming of multi-UVs in versatile mission scenario, so we proposed a generic mission model and designed APF models for mission performance. Unfortunately, to the best of our knowledge, it was difficult to find corresponding studies that enables the comparative study with our scheme in the same line. We designed our generic model to be specified to various kinds of mission scenarios: surveillance [24], network provision [25], delivery [26]–[28], and so on. Since each scenario has its own metrics for evaluation (e.g., network throughput for [25]) or different constraints (e.g., [26] exploits transit network), it is inappropriate to compare the system with VMCS. Last but not least, our mission model contains network connectivity constraint, but most of path-planning studies do not consider and, and this constraint is considered independently [29], [30].

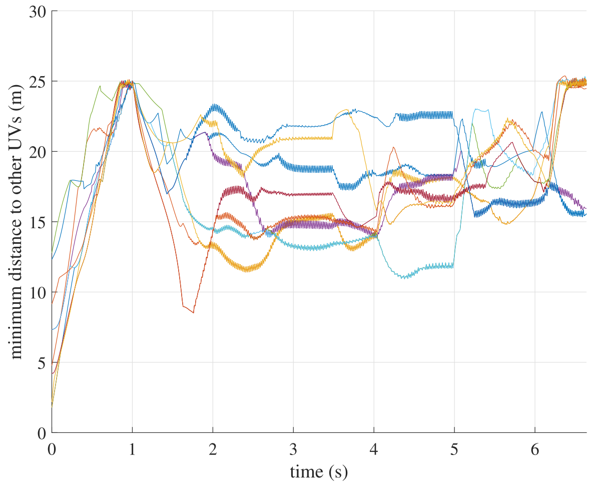


FIGURE 8. Minimum distance between UVs.

Thus, we additionally designed a centralized scheme that a central station sends moving command to the UVs by network.

We designed the centralized multi-UV control mechanism for our mission model as follows.

- Central control station is located at the gateway and firstly allocates the nearest mission target to each UV.
- After the UV finishes the mission at a target, central station allocates another unfinished mission target to the UV.
- If a UV reaches the maximum capacity, the UV returns to the control station. After emptying the load, station reallocates another unfinished mission target to the UV.

The main reason for this reactive assigning approach is for preparing the continuous scenario as addressed in Section V-E. Pre-scheduling mechanism is hard to be applied to the scenario that the mission targets are continuously spawned.

With the above major rules, we added the additional mechanisms to improve the validity of the system.

- **Connectivity**: If a UV moving to the target is determined as disconnected (could not find the route to the central station), the UV temporally holds its position and the central station assigns another UV a networking task. After this newly assigned UV is placed between the station and the UV, the disconnected UV restores its connection and performs the task. Central station preferentially chooses the networking UV that is not assigned the mission target, which can exist after small number of the mission targets left. If there is no unassigned UV, central station randomly chooses a UV.
- **Network delay**: We added the networking delay from the UVs to the central station. When the station commands a UV (for mission targeting or networking), the UV starts the task after the amount of the network delay. However, the network delay is not applied when the UV returns to the station due to the maximum capacity, or the UV holds its position due to the network disconnection.

We calculated the network delay of each communication as

$$NetworkDelay = t_{delay,onehop} \times N_{hops}, \tag{20}$$

where $t_{delay,onehop}$ refers to the one-hop transmission delay of the network. We determined $t_{delay,onehop}$ by exploiting the analytical wireless transmission model in [31].

We simulated both VMCS and the centralized scheme while varying the number of UVs from 10 to 50, and Fig. 9 graphically represents the mission completion time of each. As shown in the figure, VMCS outperforms the centralized scheme, primarily because there is no networking delay. From this evaluation, we showed that VMCS accelerates the mission performance against centralized scheme.

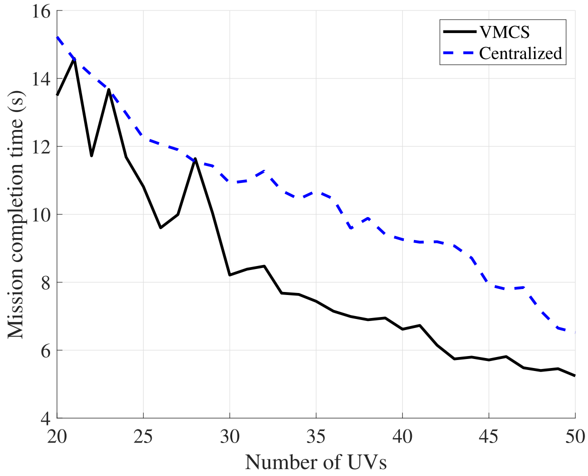


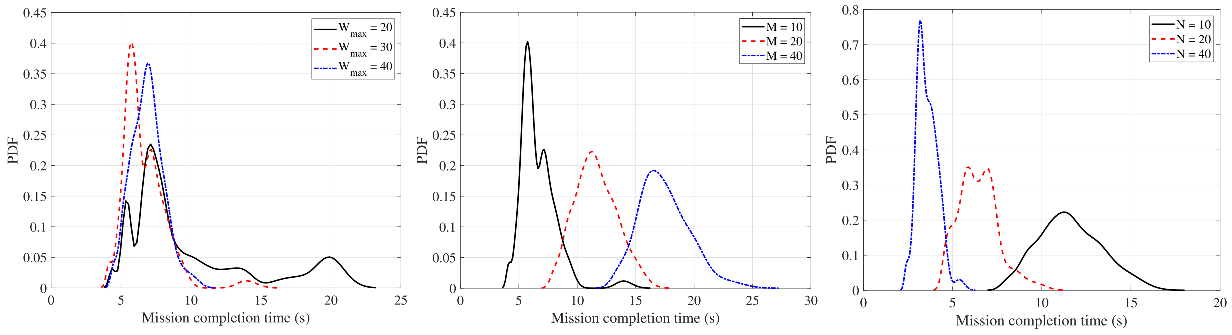
FIGURE 9. Comparison with centralized scheme.

B. COMMUNICATION RANGE

Communication range of the UVs can affect the mission performance since it limits the movement of the UVs more occasionally. Fig. 10 shows the mission completion time while changing the maximum communication range, where $|M| = 10$ and $|N| = 10$. Just to clarify, $|M|$ represents the number of mission targets as described in Section IV-A and $|N|$ represents the number of swarming UVs in the mission. Commonly, it is expected that the longer communication range shows the better performance. However, as shown in the figure, the case of $W_{max} = 30$ outperforms the other cases $W_{max} = 20$ and $W_{max} = 40$. The reason comes from the appliance of F_c. At the case of $W_{max} = 40$, the UVs are spread more widely before the mission starts, which results in the more travel time to the mission targets. From the observation, we can conclude that increasing the communication range does not always result in a better performance.

C. MISSION SIZE

While changing the mission size, W_{max} is set to 30 and $|N|$ is set to 10. As shown in Fig. 10b, increasing mission size



(a) Mission completion time where $W_{max} = \{20, 30, 40\}$ (b) Mission completion time where $|M| = \{10, 20, 40\}$ (c) Mission completion time where $|N| = \{10, 20, 40\}$

FIGURE 10. Mission performance while varying numerical factors.

results in the larger mission completion time. It is notable that the differences of the cases $|M| = 10$ and $|M| = 20$, and the cases $|M| = 20$ and $|M| = 40$ are almost similar. This observation can lead to the conclusion that the mission size and the mission completion time is not linearly proportional. The reason for this is the parallelization of the UVs' actions, where each UV performs the mission with reasonable path. Furthermore, the case of smaller $|M|$ leads to the smaller variance of the mission completion time, especially at the $|M| = 10$. The reason for this is the instant deployment of 10 UVs to 10 targets, which means that VMCS can quickly dispatch the UVs to the targets one by one. When $|M|$ increases, each UV takes more variant time to approach to multiple mission targets, in dynamic changes of formation. From this simulation, we showed that our proposed APF models could achieve the high performance of multi-UV system.

D. SWARM SIZE

We simulated our system while increasing the number of the UVs, which results in Fig. 10c. In this simulation, W_{max} is set to 30 and $|M|$ to 20. As expected, the case of larger $|N|$ outperforms the case of smaller $|N|$. The first reason is the less travel time of UVs, since the UVs are densely swarming at the 3D map. Secondly, when a UV has full capacity ($C_i \approx C_{max}$), there are the other UVs to be replaced and continue the mission. When a UV needs to return to the station, consecutive topology changes can occur due to the large distance from the station. From this simulation, we showed that our topology control algorithm properly changes H while operating a large number of UVs.

E. ROBUSTNESS

We investigated the persistence of our APF models, using our system. We periodically spawned the mission targets per 0.5 s, during 1 hour simulation. We deployed 20 UVs in the map, with $W_{max} = 30$. Then, we measured $K = \sum_{m_k \in M} R_{m_k}$ per 10 ms, which means the current remaining mission load. If the swarm suffers unexpected deadlock or fails to return the station, K will linearly increase due to the accumulated mission targets. Fig. 11 shows the graphical representation of K with respect to the simulation time.[1] As shown in the figure, K oscillates around 600 to 800, and we can estimate that there are about 3 mission targets regularly remaining in the map. The persistence of the system originates from the vector-to-operation mapping which results in the bounded positioning of the UVs, and the stable APFs designed and proved in Section IV-B and IV-D. From the evaluation, we showed that the VMCS can run the general-purpose swarm control system designed with practical potential models.

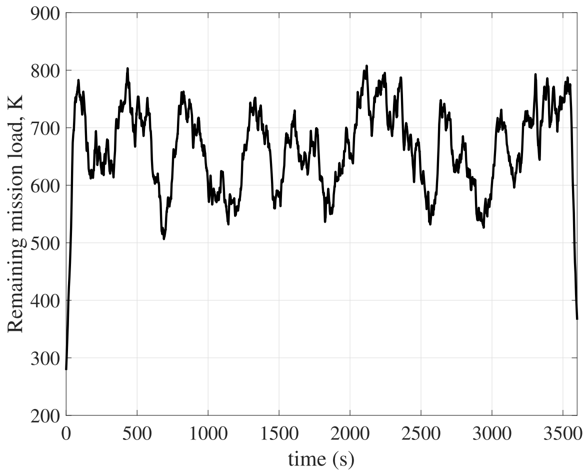


FIGURE 11. *K* with respect to the simulation time.

F. FAIRNESS

Depending on the application, performing the mission can consume the long-term resources such as battery or stock of supply meterials. In this case, it is best to allocate tasks equally to the UVs to maximize the job life. In other word, the system should guarantee *fairness* of the system in mission. Thus, we investigated the amount of work done by each UV. From the scenario of Section V-E, we collected the accumulated amount of work done by each UV in a whole time. Then,

[1] we smoothed the result by windowed average for visibility.

we defined *Throughput* as the amount of the work done per a second, whose unit is in *units/s*. Fig. 12 shows the throughput measured over time. As shown in the figure, although not exactly converges to the same, it can be seen that the UVs converge to almost similar throughput. This result indicates UVs do the mission fairly without idling in the middle of mission performance. The major reason for this fairness is the collaboration of $\mathbf{F_c}$ and $\mathbf{F_m}$. When a target spawned, UVs within D_m gathers by $\mathbf{F_m}$, but with enough distance between them by $\mathbf{F_c}$. Such spaced formation yields opportunity to be assigned to the other targets to the UVs. Even though $\mathbf{F_m}$ applies larger potential to the UVs far from the target, cap of velocity (Eq. (1)) prevents reversed assignment of target. From this evaluation, we showed that our APF model shows fairness in VMCS scheme as UVs are evenly dedicated to the mission performance, even though each UV individually determines its location.

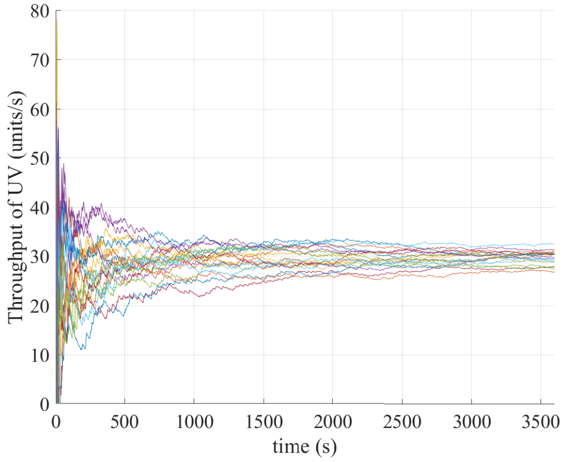


FIGURE 12. Throughput of UVs.

G. DISCUSSION FOR USE CASES

The proposed system does not only contribute to the stability of the control, also allows the system to be applied to various environments other than multi-vehicle control, such as multi-agent air purifier (or conditioner) or multiple speakers on music performance. We introduce some use cases of our scheme.

- **Swarming air purifiers**: Note that existing concept for automated building monitoring system is based on the concept of IoT, where a centralized controller manages the states of the connected devices [32], [33]. By adopting our scheme, air purifier can control its operation by sensing the environment and nearby other purifiers, without communication or network module. APF can be modeled with respect to the dust concentration and the operation status of the other devices at nearby of itself.
- **Power control of wireless sensor network (WSN)**: Life cycle of the wireless sensors is essential due to its complexity in duty cycle scheduling [34]. By designing APF model respect to the traffic ratio and nearby node's status, the node can control its own duty cycle to meet the network's needs. Additionally, our system can be utilized for dynamic control of the transmission power, which can reduce the signal interference in the wireless medium [35].

The sequential action of the APF's vector sum can be *any* of the agent's operation, such as powering submodules, spraying light, networking or signaling as well as movement. By extracting the magnitude of the vector sum, the agent can determine the level of its operation according to the environment.

VI. CONCLUSION

We aimed to present the practical design of the APF-based multi-UV control system for versatile environments. To overcome the limitation of previous APF-based systems' functionality, we decomposed the APF modeling phase and the control decision phase. We designed a generic mission structure that can be applied to the various environments, and formulated the APF for the mission completion. Furthermore, we proposed a topology control algorithm based on our APF models, which increases the flexibility of the UV formulation. We investigated our system in simulation, and the numerical results showed the performance of our system in various condition.

As future work, we can leverage our scheme to construct various kinds of multi-agent control, such as the examples introduced in Section III. Also, we can refine our APF models for better performance, pursuing the optimal formation changes of the swarm. We believe that our studies have the potential to enlarge the applicability of the APF-based approaches. Furthermore, we expected the proposed scheme and generalized approach to be a great inspiration to the area of multi-agent control system research.

ACKNOWLEDGMENT

(Seongjoon Park and Hyeong Tae Kim contributed equally to this work.)

REFERENCES

[1] N. Kshetri and D. Rojas-Torres, "The 2018 winter olympics: A showcase of technological advancement," *IT Prof.*, vol. 20, no. 2, pp. 19–25, Mar. 2018.

[2] M. Rubenstein, A. Cornejo, and R. Nagpal, "Programmable self-assembly in a thousand-robot swarm," *Science*, vol. 345, no. 6198, pp. 795–799, Aug. 2014.

[3] M. Abolhasan, J. Lipman, W. Ni, and B. Hagelstein, "Software-defined wireless networking: Centralized, distributed, or hybrid?" *IEEE Netw.*, vol. 29, no. 4, pp. 32–38, Jul. 2015.

[4] Y. K. Hwang and N. Ahuja, "A potential field approach to path planning," *IEEE Trans. Robot. Autom.*, vol. 8, no. 1, pp. 23–32, Feb. 1992.

[5] C. W. Warren, "Multiple robot path coordination using artificial potential fields," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 1990, pp. 500–505.

[6] C. W. Warren, "Global path planning using artificial potential fields," in *Proc. Int. Conf. Robot. Autom.*, May 1989, pp. 316–321.

[7] N. E. Leonard and E. Fiorelli, "Virtual leaders, artificial potentials and coordinated control of groups," in *Proc. 40th IEEE Conf. Decis. Control*, vol. 3, Dec. 2001, pp. 2968–2973.

[8] E. G. Hernández-Martínez and E. Aranda-Bricaire, "Convergence and collision avoidance in formation control: A survey of the artificial potential functions approach," in *Multi-Agent Systems: Modeling, Control, Programming, Simulations and Applications*. Rijeka, Croatia: InTech, 2011.

[9] D. Morgan, S.-J. Chung, and F. Y. Hadaegh, "Model predictive control of swarms of spacecraft using sequential convex programming," *J. Guid., Control, Dyn.*, vol. 37, no. 6, pp. 1725–1740, Nov. 2014.

[10] S. Bandyopadhyay, S.-J. Chung, and F. Y. Hadaegh, "Probabilistic swarm guidance using optimal transport," in *Proc. IEEE Conf. Control Appl. (CCA)*, Oct. 2014, pp. 498–505.

[11] S. Bandyopadhyay, S.-J. Chung, and F. Y. Hadaegh, "Probabilistic and distributed control of a large-scale swarm of autonomous agents," *IEEE Trans. Robot.*, vol. 33, no. 5, pp. 1103–1123, Oct. 2017.

[12] J. Yu and S. M. LaValle, "Optimal multirobot path planning on graphs: Complete algorithms and effective heuristics," *IEEE Trans. Robot.*, vol. 32, no. 5, pp. 1163–1177, Oct. 2016.

[13] Y.-B. Chen, G.-C. Luo, Y.-S. Mei, J.-Q. Yu, and X.-L. Su, "UAV path planning using artificial potential field method updated by optimal control theory," *Int. J. Syst. Sci.*, vol. 47, no. 6, pp. 1407–1420, Apr. 2016.

[14] D. Shah and L. Vachhani, "Swarm aggregation without communication and global positioning," *IEEE Robot. Autom. Lett.*, vol. 4, no. 2, pp. 886–893, Apr. 2019.

[15] Y. Huang, H. Ding, Y. Zhang, H. Wang, D. Cao, N. Xu, and C. Hu, "A motion planning and tracking framework for autonomous vehicles based on artificial potential field elaborated resistance network approach," *IEEE Trans. Ind. Electron.*, vol. 67, no. 2, pp. 1376–1386, Feb. 2020.

[16] Z. Zhou, J. Wang, Z. Zhu, D. Yang, and J. Wu, "Tangent navigated robot path planning strategy using particle swarm optimized artificial potential field," *Optik*, vol. 158, pp. 639–651, Apr. 2018.

[17] J. Guldner and V. I. Utkin, "Sliding mode control for gradient tracking and robot navigation using artificial potential fields," *IEEE Trans. Robot. Autom.*, vol. 11, no. 2, pp. 247–254, Apr. 1995.

[18] S.-J. Chung, A. A. Paranjape, P. Dames, S. Shen, and V. Kumar, "A survey on aerial swarm robotics," *IEEE Trans. Robot.*, vol. 34, no. 4, pp. 837–855, Aug. 2018.

[19] M. Bae, S. Yoo, J. Jung, S. Park, K. Kim, J. Lee, and H. Kim, "Devising mobile sensing and actuation infrastructure with drones," *Sensors*, vol. 18, no. 3, p. 624, Feb. 2018.

[20] C. W. Reynolds, "Flocks, herds and schools: A distributed behavioral model," in *Proc. 14th Annu. Conf. Comput. Graph. Interact. Techn.*, 1987, pp. 25–34.

[21] Y. Jahir, M. Atiquzzaman, H. Refai, A. Paranjothi, and P. G. LoPresti, "Routing protocols and architecture for disaster area network: A survey," *Ad Hoc Netw.*, vol. 82, pp. 1–14, Jan. 2019.

[22] Y. Li, Y. Chen, and I. Podlubny, "Stability of fractional-order nonlinear dynamic systems: Lyapunov direct method and generalized Mittag–Leffler stability," *Comput. Math. Appl.*, vol. 59, no. 5, pp. 1810–1821, Mar. 2010.

[23] (Sep. 2020). *VMCS: Elaborating APF-Based Swarm Intelligence for Mission-Oriented Multi-UV Control (Video)*. [Online]. Available: https://youtu.be/Ww__X2v9Wu4

[24] J. Jung, S. Yoo, W. La, D. Lee, M. Bae, and H. Kim, "AVSS: Airborne video surveillance system," *Sensors*, vol. 18, no. 6, p. 1939, Jun. 2018.

[25] S. Yoo, J. Jung, A. Y. Chung, K. Kim, J. Lee, S. Park, S. K. Lee, H. K. Lee, and H. Kim, "Empowering drones' teamwork with airborne network," in *Proc. IEEE 31st Int. Conf. Adv. Inf. Netw. Appl. (AINA)*, Mar. 2017, pp. 678–685.

[26] S. Choudhury, K. Solovey, M. J. Kochenderfer, and M. Pavone, "Efficient large-scale multi-drone delivery using transit networks," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2020, pp. 4543–4550.

[27] K. Peng, J. Du, F. Lu, Q. Sun, Y. Dong, P. Zhou, and M. Hu, "A hybrid genetic algorithm on routing and scheduling for vehicle-assisted multi-drone parcel delivery," *IEEE Access*, vol. 7, pp. 49191–49200, 2019.

[28] M. Kim and E. T. Matson, "A cost-optimization model in multi-agent system routing for drone delivery," in *Proc. Int. Conf. Practical Appl. Agents Multi-Agent Syst. (PAAMS)*. Berlin, Germany: Springer, 2017, pp. 40–51.

[29] Z. Zhao and T. Braun, "Topology control and mobility strategy for UAV ad-hoc networks: A survey," in *Proc. Joint ERCIM eMobility MobiSense Workshop*, 2012, pp. 27–32.

[30] S. Park, H. T. Kim, and H. Kim, "Energy-efficient topology control for UAV networks," *Energies*, vol. 12, no. 23, p. 4523, Nov. 2019.

[31] G. Bianchi, "Performance analysis of the IEEE 802.11 distributed coordination function," *IEEE J. Sel. Areas Commun.*, vol. 18, no. 3, pp. 535–547, Mar. 2000.

[32] C. Wei and Y. Li, "Design of energy consumption monitoring and energy-saving management system of intelligent building based on the Internet of Things," in *Proc. Int. Conf. Electron., Commun. Control (ICECC)*, Sep. 2011, pp. 3650–3652.

[33] K. Akkaya, I. Guvenc, R. Aygun, N. Pala, and A. Kadri, "IoT-based occupancy monitoring techniques for energy-efficient smart buildings," in *Proc. IEEE Wireless Commun. Netw. Conf. Workshops (WCNCW)*, Mar. 2015, pp. 58–63.

[34] T. Van Dam and K. Langendoen, "An adaptive energy-efficient mac protocol for wireless sensor networks," in *Proc. 1st Int. Conf. Embedded Netw. Sensor Syst.*, 2003, pp. 171–180.

[35] R. Ramanathan and R. Rosales-Hain, "Topology control of multihop wireless networks using transmit power adjustment," in *Proc. IEEE INFOCOM. Conf. Comput. Communications. 19th Annu. Joint Conf. IEEE Comput. Commun. Societies*, vol. 2, Mar. 2000, pp. 404–413.

SEONGJOON PARK (Graduate Student Member, IEEE) received the B.S.E. degree in electrical engineering from Korea University, Seoul, South Korea, in 2015, where he is currently pursuing the Ph.D. degree with the School of Electrical Engineering. His research interests include community wireless networks, network modeling and simulations, and multiple UAVs applications.

HYEONG TAE KIM received the B.S.E. degree in electrical engineering from Korea University, Seoul, South Korea, in 2019, where he is currently pursuing the M.S.E. degree with the School of Electrical Engineering. His research interests include swarm intelligence and deep neural networks, particularly in reinforcement learning and multi-agent reinforcement learning.

HWANGNAM KIM (Member, IEEE) received the B.S.E. degree in computer engineering from Pusan National University, Busan, South Korea, in 1992, the M.S.E. degree in computer engineering from Seoul National University, Seoul, South Korea, in 1994, and the Ph.D. degree in computer science from the University of Illinois at Urbana–Champaign, in 2004. He is currently a Professor with the School of Electrical Engineering, Korea University, Seoul. His research interests include wireless networks, unmanned aerial systems (UASs), UAS traffic management (UTM), counter UAS systems, the Internet of Things, and cyber physical systems.