Unnamed Systems, Vol. 0, No. 0 (2013) 1–17
© World Scientific Publishing Company

Automated Playbook for UAV Traffic Management based on Spatiotemporal Scenario Data

Chenyuan He [*], Yan Wan [†], and Junfei Xie[‡]

University of Texas at Arlington, Arlington, TX, 76019
San Diego State University, San Diego, CA, 92182

This paper develops a decision framework to automate the playbook for UAS traffic management (UTM) under uncertain environmental conditions based on spatiotemporal scenario data. Motivated by the traditional air traffic management (ATM) which uses the playbook to guide traffic using pre-validated routes under convective weather, the proposed UTM playbook leverages a database to store optimal UAS routes tagged with spatiotemporal wind scenarios to automate the UAS trajectory management. Our perspective is that the UASs, and many other modern systems, operate in spatiotemporally evolving environments, and similar spatiotemporal scenarios are tied with similar management decisions. Motivated by this feature, our automated playbook solution integrates the offline operations, online operations and a database to enable real-time UAS trajectory management decisions. The solution features the use of similarity between spatiotemporal scenarios to retrieve offline decisions as the initial solution for online fine tuning, which significantly shortens the online decision time. A fast query algorithm that exploits the correlation of spatiotemporal scenarios is utilized in the decision framework to quickly retrieve the best offline decisions. The online fine tuning adapts to trajectory deviations and subject to collision avoidance among UASs. The solution is demonstrated using simulation studies, and can be utilized in other applications where quick decisions are desired and spatiotemporal environments play a crucial role in the decision process.

Keywords: UAS traffic management; Playbook; Spatiotemporal scenario data based decision framework.

1. Introduction

Unmanned aircraft system (UAS) technologies have been rapidly developed in recent years and have found applications in a wide range of civilian domains, including goods delivery,[1] infrastructure survelliance,[2] smart farming,[3] on-demand communication,[4,5] and others. In order to integrate UASs into the National Airspace System (NAS), many efforts have been devoted to UAS traffic management (UTM).[6,7] UTM aims to provide safe and efficient low-altitude airspace services, including airspace design, dynamic configuration, route planning and rerouting, geo-fencing, severe weather and wind avoidance, congestion management, terrain avoidance, and separation management.[7,8] Although most of the current studies of UTM are concerned with a low-density airspace, the long-term goal is to enable the safety and efficiency of low-altitude airspace with dense operations of UAS under dynamic weather and other potential hazards.

In this paper, we borrow ideas from traditional air traffic management (ATM) to UTM. The current ATM solutions, developed mainly for commercial airlines, adopt the national playbook[9] to reroute traffic during periods of convective weather or other off-nominal events that affect the coordination of routes. The national playbook is a collection of Severe Weather Avoidance Plan (SWAP) routes that have been pre-validated with involved air route traffic control centers (ARTCCs). Experienced traffic controllers at traffic management units (TMUs) check the national playbook, evaluate the route advisories, and then determine a certain route to implement.

Despite the similarities between UTM and ATM, it is challenging to directly apply ATM solutions to UTM considering the unique characteristics of UTM.[8,10] First, the management solutions need to be automated instead of human controlled considering the high volume of UASs in a dense airspace. Second, UASs are much more sensitive to weather, such as varying wind conditions, due to their small size and low weight. Finer weather resolutions need to be considered. Third, the management solutions need to be

[*] Ph.D. Student, Department of Electrical Engineering. Email: chenyuan.he@mavs.uta.edu.

[†] Professor, Department of Electrical Engineering. Email: yan.wan@uta.edu.

[‡] Assistant professor, Department of Electrical and Computer Engineering. Email: jxie4@sdsu.edu.

adaptive to the dynamic and varying weather conditions. Forth, computational efficiency becomes an issue to resolve with the growing number of UASs in the NAS.

There have been some studies in the literature that investigate the various aspects of UTM. The studies of our interest include the UTM capacity studies,[11] system structures,[8] communication framework,[12] weather service,[13] representative UAS trajectory models,[14] collision avoidance protocols,[15,16] and multi-UAS path planning algorithms.[17–23] In this paper, we propose a spatiotemporal scenario data-driven decision framework which automates the playbook for UTM subject to uncertain environmental conditions. This solution can also be utilized in other applications where real-time decision is desired and spatiotemporal environments play a significant role in the decision process.

The key idea of our proposed spatiotemporal scenario data-driven decision framework is based on the fact that similar spatiotemporal scenarios are tied with similar decision solutions. Broadly, modern systems often operate in spatiotemporally evolving environments, and their dynamics are modulated by environments in a complicated fashion. Such uncertain spatiotemproal environments, if exploited, can improve the efficiency of end-to-end decision-making. To enact the environmental data-driven big data analytics, we earlier systematically investigated a new data type, called the *spatiotemporal scenario data*.[24,25] A data point of spatiotemproal scenario data is composed of a sequence of snapshots, each of which is a spatial map. Different from static data,[26] spatial graphical data,[27] temporal data,[28,29] and spatiotemporal data[30] that have been widely studied in the literature, the spatiotemporal scenario data, unique for physical networked dynamic systems, feature tight correlations across the spatial, temporal, and also spatiotemporal dimensions. In our prior studies, we developed data-driven and model-based algorithms to analyze and process the spatiotemporal scenario data.[24,31–34] Along the data-driven direction, we developed a multi-resolution moving-window approach to capture spatiotemporal correlations and developed a distance measure to quantify the similarity among spatiotemporal scenarios.[24] We also developed an effective similarity search algorithm to quickly query similar spatiotemporal scenarios by exploring the correlations among spatiotemporal dimensions.[32] Along the model-based direction, we adopted a reduced-order stochastic influence model to capture uncertain spatiotemporal weather spread dynamics.[33,34] The sufficient and necessary conditions for the identifiability of the influence model was proposed and estimators were developed to find model parameters.[33] A model-based distance measure[34] based on Kullback-Leibler (KL) divergence was then developed to validate the data-driven distance measure.[24] By utilizing the properties of spatiotemporal scenario data and the corresponding analytical methods we developed, this paper [a] investigates the use of spatiotemporal scenario data query to automate the UTM playbook, and facilitate optimal online decisions for multi-UAS trajectory management under uncertain environmental conditions.

The basic operations of the proposed spatiotemporal scenario data-driven solution for UTM are as follows. First, we construct offline a database of spatiotemporal wind scenarios tagged with their optimal multi-UAS trajectory decisions. Next, based on current wind forecast, a fast query of similar spatiotemporal wind scenario in the database is conducted to retrieve the best offline UAS trajectory decision. The decision is then fine tuned online to obtain the optimal trajectory decision for the current weather scenario with significantly shortened online computing time.

The contributions of this paper are summarized as follows. First, different from the existing studies that treat uncertain wind fields as constants, Gaussian noises or time-varying variables,[26,35,36] we here consider wind fields as stochastic spatiotemporal processes that demonstrate spatial, temporal and spatiotemporal correlations. Second, we efficiently automate the playbook for UTM based on spatiotemporal scenario data. Our automated playbook solution integrates offline operations, online operations and a spatiotemporal scenario database, and significantly accelerates online decisions in adaptation to environmental changes. Database query relies on a spatiotemporal scenario data distance measure and a fast query algorithm that exploits spatiotemporal scenario data correlations. The online tuning aims to guarantee the safety and accuracy of multi-UAS trajectories. Third, the proposed solution framework is flexible and can be easily extended to include more control actions, and more complicated vehicle dynamics and weather conditions. It can also be utilized in other applications where quick decisions are desired in spatiotemporal environments.

The rest of this paper is organized as follows. Section 2 overviews the spatiotemporal scenario data-driven decision framework. Section 3 introduces offline UTM operations, including the spatiotemporal influence model, optimal control-based trajectory management solution, and spatiotemporal scenario database construction. Section 4 introduces online UTM operations, including spatiotemporal scenario data distance measure, fast spatiotemporal scenario data query, online tuning, and database expansion. Section 5 presents simulation studies. Section 6 concludes the paper.

2. Overview of the Spatiotemporal Scenario Data-driven Decision Framework

The spatiotemporal scenario data-driven decision framework is developed based on the assumption that similar spatiotemporal scenarios share similar decision solutions, and hence stored solutions can be leveraged to accelerate online decisions. The framework is composed of three main blocks shown in Figure 1: the offline block, online block and

[a] A brief conference version of this paper can be found in reference.[25]

database block.

The offline block contains three operations. The first operation is to obtain spatiotemporal scenario data from either historical environmental data or environmental simulation models. The second operation is to find an optimal solution for each scenario. The third operation is to construct a database of spatiotemporal scenarios and their corresponding optimal solutions.

The online block also contains three operations. The first operation is to query the spatiotemporal scenario database and search for the scenario that is most similar to the current scenario provided by environmental forecasting tools. The corresponding optimal solution is thus retrieved. Because the current scenario may not be exactly the same as the retrieved scenario during flight, the retrieved solution is then fine tuned online in the second operation. Once the optimal solution to the current scenario is obtained, the solution is pushed to the database in the third operation to expand the solution sets for future use.

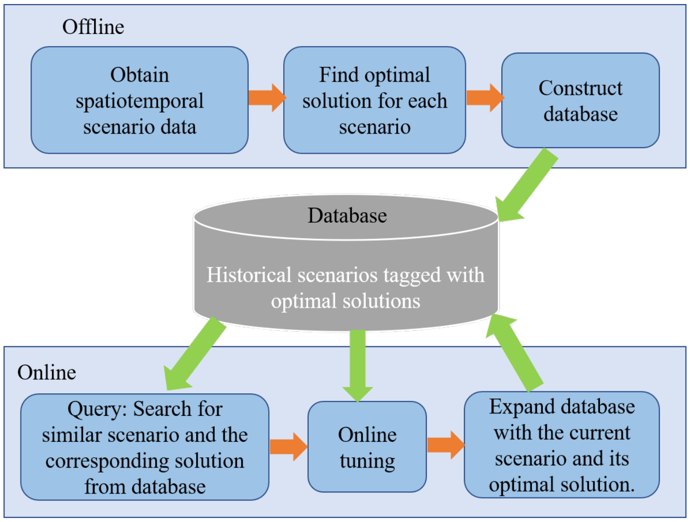

Offline
Obtain spatiotemporal scenario data
Find optimal solution for each scenario
Construct database
Database
Historical scenarios tagged with optimal solutions
Online
Query: Search for similar scenario and the corresponding solution from database
Online tuning
Expand database with the current scenario and its optimal solution.

Fig. 1. Spatiotemporal Scenario Data-driven Decision Solution.

We use the aforementioned spatiotemporal scenario data-driven decision framework to solve multi-UAS trajectory management for UTM as a case study. We adopt the stochastic influence model which captures spatiotemporal spread patterns for the wind fields. We then formulate the multi-UAS trajectory management problem using optimal control in continuous state and action spaces. Each UAS has its own dynamics, starting point and destination. We assume that once the UASs are launched into the air, they navigate at the same altitude, and as such a 2D UAS mobility model is adopted. We adopt the required time of arrival (RTA) for the UTM initiative.[37] We assume that each UAS has a RTA constraint. Our objective is to drive all the UASs to their destinations with minimal total energy consumption, subject to uncertain wind conditions, collision avoidance constraints and RTA constraints. By leveraging the spatiotemporal scenario database and the fast query algorithm, the proposed framework realizes the automated UTM playbook with computational efficiency. Of notice, the proposed framework can be applied to the decision making in other time stringent applications under spatiotemporally varying environments as well.

3. Offline Operations

In this section, we first introduce the generation of spatiotemporal wind scenario data using the stochastic influence model. We then develop the offline optimal multi-UAS trajectory management solution subject to spatiotemporally varying wind conditions and collision avoidance constraints. In the end, we describe the construction of database to store the spatiotemporal wind scenarios tagged with corresponding optimal multi-UAS trajectory solutions.

3.1. *Generation of Spatiotemporal Wind Scenario Data Using the Influence Model*

The spatiotemporal wind scenario database can be constructed from historical wind data or wind field models. We here use the influence model to generate a spatiotemporal wind dataset.[33,34,38] The model captures spatiotemporal correlations by modeling both network- and local- level spatiotemporal spread properties (see Figure 2 for example). Compared to Markov models, the influence model provides a tractable reduced-order representation of a stochastic network and hence is significantly more computationally efficient.

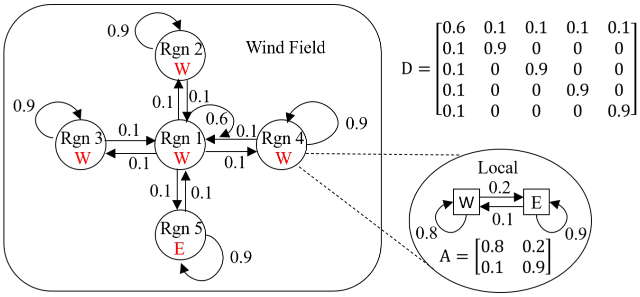
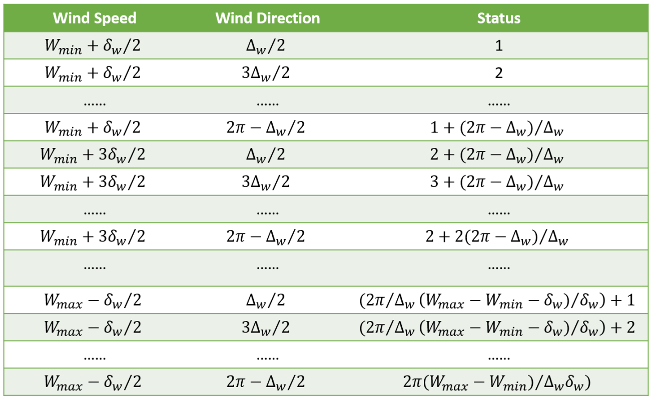


Fig. 2. A simple illustrative example of an influence model with five regions and two wind statuses 'W' and 'E'.

Because the influence model is a discrete time and discrete state model, we first discretize the time and wind conditions. Given a time horizon $[0, T]$, we adopt a sampling period ΔT and obtain a set of discrete time points $\{0, \Delta T, 2\Delta T, \cdots, T\}$. For simplicity, we denote the time step from $(k-1)\Delta T$ to $k\Delta T$ by k, where $k \in$

$\{1, 2, \cdots, \frac{T}{\Delta T}\}$. Consider a wind field of N regions. Each region has its own wind that interacts with other regions stochastically. We describe wind at each region using two elements: speed and direction. Let W_{max} and W_{min} denote the upper and lower bounds of the wind speed respectively. The wind direction ranges from 0 to 2π radius. Let δ_w and Δ_w denote the speed and angle resolutions respectively. The wind speed and direction are discretized accordingly. In order to facilitate the influence model, we encode speed and direction to one wind status by constructing a look-up table (see Figure 3). The wind status for region n at time step k is denoted by a scalar $I_{n,k}$, where $n \in \{1, 2, \cdots, N\}$. The total number of wind statuses M is calculated as $M = \frac{2\pi(W_{max} - W_{min})}{\Delta_w \delta_w}$. We use a length-$M$ row vector $S_n[k]$ to denote the vector form of the wind status of region n at time step k. $S_n[k]$ is filled with zeros except a '1' at the position of $I_{n,k}$. Therefore, $S_n[k]$ and $I_{n,k}$ have a one-to-one mapping relationship. The whole wind field's state $S[k]$ can be denoted by cascading $S_n[k]$ for all $n \in \{1, 2, \cdot, N\}$, i.e., $S[k] = \left[S_1^{\mathrm{T}}[k], S_2^{\mathrm{T}}[k], \cdots, S_N^{\mathrm{T}}[k]\right]^{\mathrm{T}}$, where the superscript $^{\mathrm{T}}$ is the transpose operator.



Wind Speed	Wind Direction	Status
$W_{min} + \delta_w/2$	$\Delta_w/2$	1
$W_{min} + \delta_w/2$	$3\Delta_w/2$	2
……	……	……
$W_{min} + \delta_w/2$	$2\pi - \Delta_w/2$	$1 + (2\pi - \Delta_w)/\Delta_w$
$W_{min} + 3\delta_w/2$	$\Delta_w/2$	$2 + (2\pi - \Delta_w)/\Delta_w$
$W_{min} + 3\delta_w/2$	$3\Delta_w/2$	$3 + (2\pi - \Delta_w)/\Delta_w$
……	……	……
$W_{min} + 3\delta_w/2$	$2\pi - \Delta_w/2$	$2 + 2(2\pi - \Delta_w)/\Delta_w$
……	……	……
$W_{max} - \delta_w/2$	$\Delta_w/2$	$(2\pi/\Delta_w\,(W_{max} - W_{min} - \delta_w)/\delta_w) + 1$
$W_{max} - \delta_w/2$	$3\Delta_w/2$	$(2\pi/\Delta_w\,(W_{max} - W_{min} - \delta_w)/\delta_w) + 2$
……	……	……
$W_{max} - \delta_w/2$	$2\pi - \Delta_w/2$	$2\pi(W_{max} - W_{min})/\Delta_w\delta_w)$

Fig. 3. The construction of the look-up table that encodes wind speed and direction into one wind status.

In the Influence model, the spatiotemporal wind spread dynamics is captured using two matrices, the network influence matrix D and the local Markov chain A. The network influence matrix $D \in R^N$ describes the spread pattern at the network level. Each entry d_{ij} is the probability for the wind status of region i to be influenced by the wind status of region j at the next time step. A local Markov chain $A \in R^{M \times M}$ describes the local influence between a pair of neighboring regions i and j. The entry a_{mn} of A is the probability for region j to be in wind status n at the next time step when region i is in wind status m at the current time step. D and A are both right stochastic matrices.

To use the influence model dynamics to capture uncertain wind dynamics, the probability mass function (PMF) for the wind status of region n at time step k is denoted as a length-M row vector $p_n[k]$. The network's PMF matrix can be represented by cascading $p_n[k]$ in order, $p[k] = \left[p_1^{\mathrm{T}}[k], p_2^{\mathrm{T}}[k], \cdots, p_N^{\mathrm{T}}[k]\right]^{\mathrm{T}}$. For region n, its wind status evolves spatiotemporally based on a quasi-linear combination of the wind statuses of its neighbors and itself, i.e., $p_n[k+1] = \sum_{l=1}^{N} d_{nl} S_l[k] A$. Therefore, the evolution of the wind field can be represented in a matrix multiplication form, i.e., $p[k+1] = DS[k]A$. The wind field's state at time step $k+1$ is randomly realized according to $p[k+1]$, i.e., $S[k+1] = Realize(p[k+1])$.

Each region of the wind field has its own wind status at initial time. The spatiotemporal wind scenario then evolves in time according to the initial wind statuses, the network influence matrix D and the local Markov chain A. Figure 2 shows a simple example to illustrate the influence model dynamics. The example considers a wind field composed of five regions. The wind speeds are constants for all the regions, i.e., $W_{max} = W_{min} = 1\ m/s$. The angle resolution is $\Delta_w = \pi$. Therefore, we have two encoded wind status, denoted as 'W' and 'E' respectively. The wind statuses evolve in time according to the probabilistic interactions among the five regions. Taking region 1 as an example, its wind status is influenced by not only itself, but also the wind statuses of its neighboring regions 2, 3, 4 and 5. When region 1, 2, 3 and 4's wind statuses are 'W', and region 5's wind status is 'E', the probability for region 1 to have wind status 'W' at the next time step is 0.73. This is calculated by summing the probability of 0.48 (0.6×0.8) by region 1 itself, probability of 0.08 (0.1×0.8) from each of region 2, 3 and 4, and probability of 0.01 (0.1×0.1) from region 5.

3.2. *Optimal Multi-UAS Trajectory Management Solution*

We formulate the multi-UAS trajectory management as an optimal control problem in continuous state and action spaces. Consider K UASs in a wind field, each of which has its own starting point and destination, denoted by $L_{i0} = [x_{i0}, y_{i0}]^{\mathrm{T}}$ and $L_{if} = [x_{if}, y_{if}]^{\mathrm{T}}$ respectively, where $i \in \{1, 2, \cdots, K\}$. Let $L_0 = [L_{10}, L_{20}, \cdots, L_{K0}]^{\mathrm{T}}$ and $L_f = [L_{1f}, L_{2f}, \cdots, L_{Kf}]^{\mathrm{T}}$. The control variables for each UAS are thrust u_i and its angle θ_i, where u_i is bounded by the largest thrust u_{max}, i.e., $|u_i| \leq u_{max}$. Let $(x_i(\tau), y_i(\tau))$ and $(v_{xi}(\tau), v_{yi}(\tau))$ denote the position and velocity of UAS i at time τ along the X and Y axes respectively. The corresponding speed and direction of wind for the position of UAS i at time τ are denoted by $W_i(\tau)$ and $\psi_i(\tau)$ respectively. The wind field is generated using the influence model described in Section 3.1. The system state for UAS i is $X_i = [x_i, y_i, v_{xi}, v_{yi}]^{\mathrm{T}}$. Stacking all the X_i, the integrated system state for all the UASs is denoted by $X = [X_1^{\mathrm{T}}, X_2^{\mathrm{T}}, \cdots, X_K^{\mathrm{T}}]^{\mathrm{T}}$. The UAS dynamics $\dot{X}_i(\tau) = f(X_i(\tau), u_i(\tau), \theta_i(\tau), W_i(\tau), \psi_i(\tau), \tau)$ modulated by the spatiotemporally correlated wind subject to collision avoidance can be modeled as:

$$\begin{cases} \dot{x}_i(\tau) = v_{xi}(\tau) + W_i(\tau)\cos\psi_i(\tau) \\ \dot{y}_i(\tau) = v_{yi}(\tau) + W_i(\tau)\sin\psi_i(\tau) \\ \dot{v}_{xi}(\tau) = u_i(\tau)\cos\theta_i(\tau) + \sum_{\substack{j\in\{1,2,\cdots,K\} \\ j\neq i}} \frac{\beta(x_i(\tau)-x_j(\tau))}{r_{ij}(\tau)|x_i(\tau)-x_j(\tau)|} \\ \dot{v}_{yi}(\tau) = u_i(\tau)\sin\theta_i(\tau) + \sum_{\substack{j\in\{1,2,\cdots,K\} \\ j\neq i}} \frac{\beta(y_i(\tau)-y_j(\tau))}{r_{ij}(\tau)|y_i(\tau)-y_j(\tau)|} \\ |u_i(\tau)| \leq u_{max} \end{cases}, \tag{1}$$

where r_{ij} denotes the Euclidean distance between UASs i and j, i.e., $r_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$, and β is a positive constant to account for the importance of collision avoidance. Each UAS emanates a repulsive force to compel the surrounding UASs to move away. Here we adopt inverse proportional functions with respect to r_{ij} to model the repulsive potential field. Other functions such as exponential and Gaussian functions can also be applied.[39] In particular, the accelerations $\frac{\beta(x_i(\tau)-x_j(\tau))}{r_{ij}(\tau)|x_i(\tau)-x_j(\tau)|}$ and $\frac{\beta(y_i(\tau)-y_j(\tau))}{r_{ij}(\tau)|y_i(\tau)-y_j(\tau)|}$ along the X and Y axes are produced by the repulsive force of UAS j for UAS i to avoid collision. $\frac{(x_i(\tau)-x_j(\tau))}{|x_i(\tau)-x_j(\tau)|}$ and $\frac{(y_i(\tau)-y_j(\tau))}{|y_i(\tau)-y_j(\tau)|}$ denote the directions of the repulsive force along X and Y axes respectively. When two UASs are far away from each other, the repulsive fields exert little or no influence on their motion. With the decrease of r_{ij}, the repulsive force that UAS j exerts on i increases to avoid collision.

To improve the UTM, we adopt the RTA for the UTM initiative.[37] We assume each UAS i has a RTA constraint T_i. We aim to find the optimal solutions $u = [u_1, u_2, \cdots, u_K]^{\mathrm{T}}$ and $\theta = [\theta_1, \theta_2, \cdots, \theta_K]^{\mathrm{T}}$ which drive all the UASs from their starting points L_0 to their destinations L_f at their corresponding RTAs and meanwhile minimize the total energy consumption for the UASs subject to wind disturbances and collision avoidance. Mathematically, the optimal control problem is formulated to minimize the following cost function J:

$$\begin{aligned} \min_{u,\theta} J = & \sum_{i=1}^{K} \rho\left((x_i(T_i) - x_{if})^2 + (y_i(T_i) - y_{if})^2\right) \\ & + \alpha_1 \sum_{i=1}^{K} \int_0^{T_i} u_i^2(\tau) d_\tau + \alpha_2 \sum_{i=1}^{K} \int_0^{T_i} \theta_i^2(\tau) d_\tau \\ \mathit{Subject\ to:} & \begin{cases} \dot{X}_i(\tau) = f(X_i(\tau), u_i(\tau), \theta_i(\tau), W_i(\tau), \psi_i(\tau), \tau), \\ [x_i(0), y_i(0)]^{\mathrm{T}} = L_{i0}, \\ [x_i(T_i), y_i(T_i)]^{\mathrm{T}} = L_{if}, \\ r_{ij} \geq R, \\ i, j \in \{1, 2, \cdots, K\}, i \neq j. \end{cases} \end{aligned} \tag{2}$$

where ρ, α_1 and α_2 are weighting coefficients, and R is the pre-specified safety distance.

The corresponding Hamiltonian is:

$$\begin{aligned} H(X(\tau), u(\tau), \theta(\tau), \lambda(\tau), \tau) = & \frac{1}{2}\alpha_1 \sum_{i=1}^{K} u_i^2(\tau) + \frac{1}{2}\alpha_2 \sum_{i=1}^{K} \theta_i^2(\tau) \\ & + \sum_{i=1}^{K} \lambda_{1i}\left(v_{xi}(\tau) + W_i(\tau)\cos\psi_i(\tau)\right) \\ & + \sum_{i=1}^{K} \lambda_{2i}(v_{yi}(\tau) + W_i(\tau)\sin\psi_i(\tau)) \\ & + \sum_{i=1}^{K} \lambda_{3i}\left(u_i(\tau)\cos\theta_i(\tau) + \sum_{\substack{j\in\{1,2,\cdots,K\} \\ j\neq i}} \frac{\beta(x_i(\tau) - x_j(\tau))}{r_{ij}(\tau)|x_i(\tau) - x_j(\tau)|}\right) \\ & + \sum_{i=1}^{K} \lambda_{4i}\left(u_i(\tau)\sin\theta_i(\tau) + \sum_{\substack{j\in\{1,2,\cdots,K\} \\ j\neq i}} \frac{\beta(y_i(\tau) - y_j(\tau))}{r_{ij}(\tau)|y_i(\tau) - y_j(\tau)|}\right), \end{aligned} \tag{3}$$

where $\lambda(\tau)$ is the collection of all the costates, i.e., $\lambda(\tau) = [\lambda_{11}, \lambda_{12}, \cdots, \lambda_{1K}, \lambda_{21}, \cdots, \lambda_{2K}, \lambda_{31}, \cdots, \lambda_{3K}, \lambda_{41}, \cdots, \lambda_{4K}]^{\mathrm{T}}$. The necessary conditions for $u^*(\tau)$ and $\theta^*(\tau)$ to be optimal controls are:

$$\begin{aligned} & \dot{x_i^*} = \frac{\partial H(X^*(\tau), u^*(\tau), \theta^*(\tau), \lambda^*(\tau), \tau)}{\partial \lambda_{1i}}, \\ & \dot{y_i^*} = \frac{\partial H(X^*(\tau), u^*(\tau), \theta^*(\tau), \lambda^*(\tau), \tau)}{\partial \lambda_{2i}}, \\ & \dot{v}_{xi}^* = \frac{\partial H(X^*(\tau), u^*(\tau), \theta^*(\tau), \lambda^*(\tau), \tau)}{\partial \lambda_{3i}}, \\ & \dot{v}_{yi}^* = \frac{\partial H(X^*(\tau), u^*(\tau), \theta^*(\tau), \lambda^*(\tau), \tau)}{\partial \lambda_{4i}}, \\ & \dot{\lambda}_{1i}^* = -\frac{\partial H(X^*(\tau), u^*(\tau), \theta^*(\tau), \lambda^*(\tau), \tau)}{\partial x_i}, \\ & \dot{\lambda}_{2i}^* = -\frac{\partial H(X^*(\tau), u^*(\tau), \theta^*(\tau), \lambda^*(\tau), \tau)}{\partial y_i}, \\ & \dot{\lambda}_{3i}^* = -\frac{\partial H(X^*(\tau), u^*(\tau), \theta^*(\tau), \lambda^*(\tau), \tau)}{\partial v_{xi}}, \\ & \dot{\lambda}_{4i}^* = -\frac{\partial H(X^*(\tau), u^*(\tau), \theta^*(\tau), \lambda^*(\tau), \tau)}{\partial v_{yi}}, \\ & \frac{\partial H(X^*(\tau), u^*(\tau), \theta^*(\tau), \lambda^*(\tau), \tau)}{\partial u_i} = 0, \\ & \frac{\partial H(X^*(\tau), u^*(\tau), \theta^*(\tau), \lambda^*(\tau), \tau)}{\partial \theta_i} = 0, \\ & H(X^*(\tau), u^*(\tau), \theta^*(\tau), \lambda^*(\tau), \tau) \leq H(X^*(\tau), u(\tau), \theta(\tau), \lambda^*(\tau), \tau). \end{aligned} \tag{4}$$

In general, there are two approaches to solve the above optimization problem, indirect methods and direct methods.[40] The indirect methods are based on the necessary conditions derived from the Hamiltonian. They require the computation of gradient and search for the control variables to let the gradient be zero. The indirect methods feature good accuracy and fast convergence speed. However, solving the two-point boundary value problem often requires a priori information about the structure of the solution, which is challenging for complex nonlinear dynamic systems. The direct methods convert the optimal control problem into a nonlinear programming problem and discretize the control variables as piece-wise constants. A priori knowledge about the structure of the solution is not required. The direct methods provide an approximate solution to the original optimization problem.

Here we adopt a widely used direct method for large-scale optimal control problems, the control vector parameterization (CVP), to solve the multi-UAS trajectory management problem.[41] To facilitate the illustration of the proposed decision framework, we assume the RTAs for all the UASs are the same, i.e., $T_i = T$ for all $i \in \{1, 2, \cdots, K\}$. The CVP contains two steps. First, we divide the time horizon $[0, T]$ into $H \geq 1$ control stages, with CVP specified time points $0 < t_1 < t_2 < \cdots < t_H = T$. Second, in each time interval $[t_l, t_{l+1}]$, where $l \in \{1, 2, \cdots, H\}$, the control vector is approximated by a basis function with a limited number of parameters, for example, a low-order B-spline function.[41] Then the system states are integrated forward in time, and the cost at final time T is evaluated and optimized over a number of iterations. Since the optimal control problem is nonconvex and the CVP method is sensitive to initial guesses, we design a restart function to find the global optimum. In particular, we select multiple initial guesses, implement the CVP method, and calculate the cost function for each of them. Then we find the least cost and record the corresponding control strategy as the optimal trajectory solution.

3.3. *Database Construction*

In order to automate the proposed UTM playbook, we construct a database to store the offline spatiotemporal wind scenarios and their corresponding optimal multi-UAS trajectory solutions. The key elements of the database include the wind scenario tag, spatiotemporal wind scenario data, number of UASs, starting points, destinations, optimal management solutions and waypoints for all UASs at the CVP specified time points. The structure of the data type is shown in Figure 4. The waypoint $R_i^{t_l} = [x_i(t_l), y_i(t_l)]^{\mathrm{T}}$ denotes the position of UAS i at the CVP specified time point t_l, where $i \in \{1, 2, \cdots, K\}$ and $l \in \{1, 2, \cdots, H\}$. Here we let $R^{t_l} = [R_1^{t_l}, R_2^{t_l}, \cdots, R_K^{t_l}]^{\mathrm{T}}$ store the waypoints of all UASs at t_l.

4. Online Operations

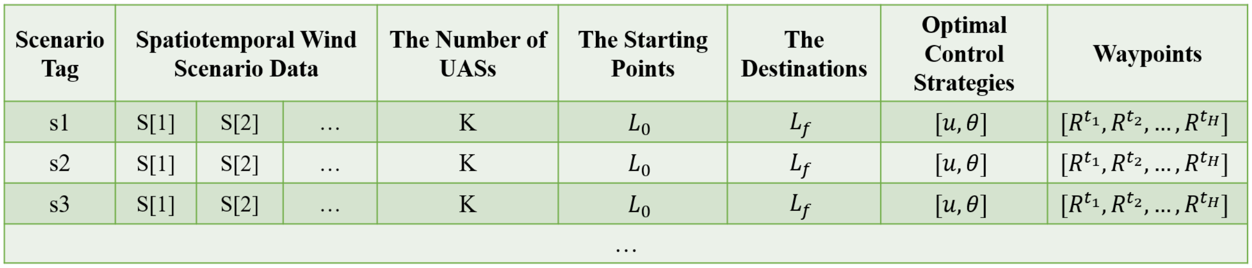
As shown in Figure 1, three main online operations are involved. With the current spatiotemporal wind scenario provided from forecasting tools, we first apply a fast query algorithm to retrieve the most similar scenario and its corresponding optimal multi-UAS management strategy from the database. We then apply online tuning to the retrieved solution to correct trajectory deviation and avoid potential collisions. In the end, we expand the database with the current scenario and its optimal solution. Before we elaborate on the fast query algorithm, let us first introduce a distance measure to be used in the fast query algorithm.

4.1. *Distance Measure For Spatiotemporal Scenario Data*

The distance measure we previously developed[24] quantifies the similarity between spatiotemporal scenario data. It calculates the distance between two spatiotemopral scenarios across spatial, temporal, and also spatiotemporal dimensions. The distance measure has the following features. First, it captures the spatiotemporal correlations for a pair of scenarios by using 3D moving windows of multiple resolutions. Second, the distance measure automatically corrects boundary effects and balances the contributions of all spatial cells and time points. Third, the distance measure is applicable for not only regular-shaped but also irregular-shaped spatial cells.

Consider two spatiotemporal wind scenarios s_a and s_b generated from the influence model introduced in Section 3.1, each of which is composed of N regions and $\frac{T}{\Delta T}$ temporal points. The scalar wind status of s_a at region n and time point k is denoted as $I^a_{n,k}$, where $n \in \{1, 2, \cdots, N\}$ and $k \in \{1, 2, \cdots, \frac{T}{\Delta T}\}$. We use moving windows at mutliple spatial and temporal dimensions to simultaneously scan s_a and s_b, and then compute their similarity. Let $\phi_{n,w}$ and $\phi_{k,h}$ denote the spatial window and temporal window respectively. $\phi_{n,w}$ is a size-w spatial window centered at region n and contains all the regions within $w - 1$ hops to region n. Φ_w is the full set of spatial windows of size w. $\phi_{k,h}$ is a size-h temporal window starting from the time point k and contains the subsequent h time points. Φ_h is the full set of temporal windows of size h. $w \in [1, w_{max}]$ and $h \in [1, h_{max}]$. The maximum sizes of spatial window w_{max} and temporal window h_{max} are selected according to the priori knowledge of spatiotemporal scenario properties. Of note, smaller window size indicates finer resolution, while larger window size indicates coarser resolution. In our previous studies,[24] we found that small w_{max} and h_{max} are typically sufficient for most applications, which alleviate the computational burden. The distance between the two scenarios with fixed spatial window size w and temporal window size h, represented by $\mathcal{D}_{a,b,w,h}$, is calculated by comparing the aggregated region wind statuses,

$$\mathcal{D}_{a,b,w,h} = \sum_{\phi_{n,w} \in \Phi_w} \sum_{\phi_{k,h} \in \Phi_h} \frac{1}{|\phi_{n,w}||\phi_{k,h}||\Phi_h|} \left| \sum_{r \in \phi_{n,w}} \sum_{g \in \phi_{k,h}} \frac{I^a_{r,g}}{\gamma_{r,w}\zeta_{g,h}} - \sum_{r \in \phi_{n,w}} \sum_{g \in \phi_{k,h}} \frac{I^b_{r,g}}{\gamma_{r,w}\zeta_{g,h}} \right|, \tag{5}$$



Scenario Tag	Spatiotemporal Wind Scenario Data			The Number of UASs	The Starting Points	The Destinations	Optimal Control Strategies	Waypoints
s1	S[1]	S[2]	…	K	L_0	L_f	$[u,\theta]$	$[R^{t_1}, R^{t_2}, \ldots, R^{t_H}]$
s2	S[1]	S[2]	…	K	L_0	L_f	$[u,\theta]$	$[R^{t_1}, R^{t_2}, \ldots, R^{t_H}]$
s3	S[1]	S[2]	…	K	L_0	L_f	$[u,\theta]$	$[R^{t_1}, R^{t_2}, \ldots, R^{t_H}]$
…								

Fig. 4. The structure of the multi-UAS trajectory database.

where

$$\gamma_{r,w} = \sum_{\phi_{n,w} \in \{\phi_{n,w} | r \in \phi_{n,w}\}} \frac{1}{|\phi_{n,w}|},$$

$$\zeta_{g,h} = \sum_{\phi_{k,h} \in \{\phi_{k,h} | g \in \phi_{k,h}\}} \frac{|T|}{\Delta T |\phi_{k,h}||\Phi_h|}.$$

$|\cdot|$ denotes the cardinality. $\gamma_{r,w}$, a spatial contribution factor, is used to correct the boundary effect of the spatial cells, so that each spatial cell contributes equally to the distance calculation. The temporal contribution factor $\zeta_{g,h}$ operates in a similar way. Figure 5 shows an illustration of the 3D moving window scanning process.

The total distance between s_a and s_b can be obtained by iterating all the spatial and temporal window sizes in the predefined ranges,

$$\mathcal{D}_{a,b} = \sum_{h=1}^{h_{max}} \sum_{w=1}^{w_{max}} \mathcal{D}_{a,b,w,h} \frac{\sigma_w \alpha_h}{\sum_{h=1}^{h_{max}} \sum_{w=1}^{w_{max}} \sigma_w \alpha_h}, \qquad (6)$$

where $\sigma_w > 0$ and $\alpha_h > 0$ are weighting factors to penalize large spatial and temporal windows, respectively. In general, a larger window size contributes less to the calculation of distance due to its coarser resolution. To reflect that, we select σ_w and α_h to be negative exponential functions as $\sigma_w = e^{-\epsilon(w-1)}$ and $\alpha_h = e^{-\rho(h-1)}$, where ϵ, $\rho \geq 0$. The procedure to calculate the distance matrix for multiple spatiotemporal wind scenarios is summarized in Algorithm 4.1.

Algorithm 4.1. Multi-Resolution Distance Algorithm For Spatiotemporal Wind Scenarios

```
Input: Multiple spatiotemporal wind scenarios $s = [s_1, s_2, \cdots, s_L]$.
Output: Distance matrix $\mathcal{D}$.
for each pair of scenarios $s_i$ and $s_j$ do
  for each pair of spatial resolution $w = 1 : w_{max}$ and temporal resolution $h = 1 : h_{max}$ do
    Calculate the distance $\mathcal{D}_{i,j,w,h}$ with a spatial window size $w$ and a temporal window size $h$ according to (5).
  end for
  Calculate the distance $\mathcal{D}_{i,j}$ between scenarios $s_i$ and $s_j$ using Equation (6).
end for
```

4.2. *Fast Query for Spatiotemporal Scenario Data*

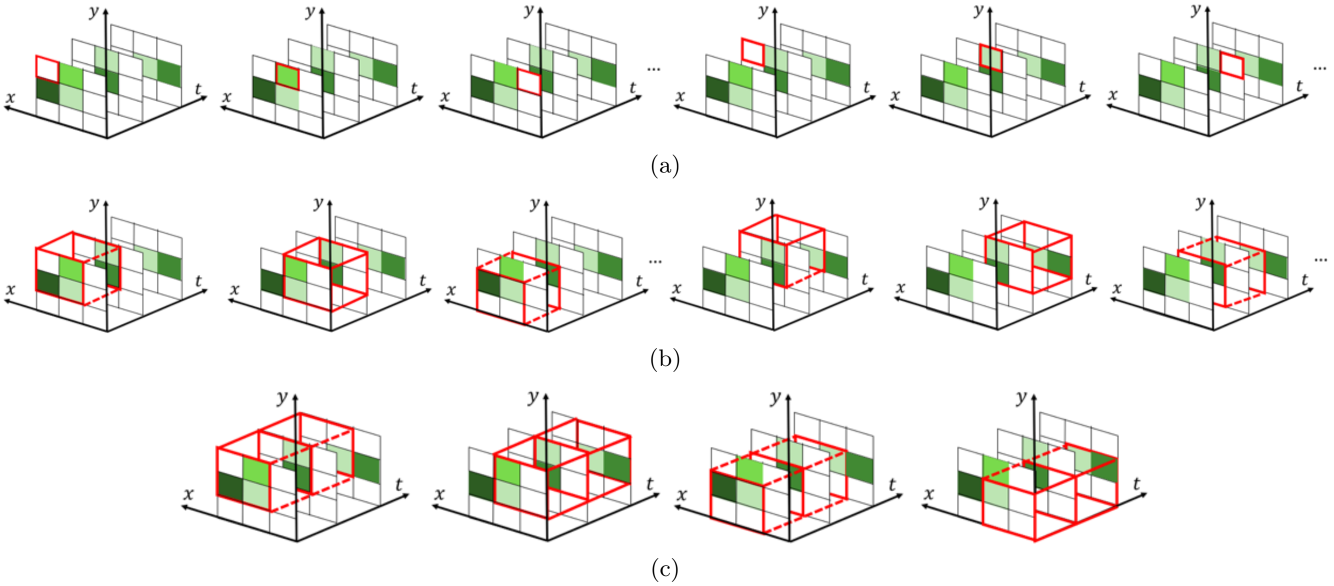
Similar scenarios can be obtained by querying the database to find the stored scenario with the smallest spatiotemporal scenario distance. According to Fig. 4, the size of the database equals the cartesian product of the number of wind scenarios and the number of different UAS settings, which includes the number of UASs and their location distributions. When the number of wind scenarios is large and there exist various UAS settings, the database is large, and this query procedure can be time-consuming. To reduce the computational cost and accelerate the query process, we use a fast query algorithm that we developed in paper.[32] The basic idea of the fast query algorithm is to trim the searching space after each resolution run by exploiting the bounds of the distance measure.

According to (5) and (6), finer resolution leads to larger distance, i.e., $\mathcal{D}_{a,b,w^*,h^*} \leq \mathcal{D}_{a,b} \leq \mathcal{D}_{a,b,1,1}$, where $w^* = N$ and $h^* = \frac{T}{\Delta T}$ are the largest spatial and temporal windows that cover the whole spatial and temporal spaces. Note that w_{max} (or h_{max}) are not necessarily equal to w^* (or h^*). Let I_a denote the total wind intensity of scenario s_a,

$$I_a = \sum_{n \in \{1,2\cdots,N\}} \sum_{k \in \{1,2\cdots,\frac{T}{\Delta T}\}} I^a_{n,k}. \qquad (7)$$

Then $\mathcal{D}_{a,b,w^*,h^*}$ can be computed as $\mathcal{D}_{a,b,w^*,h^*} = \frac{\Delta T |I_a - I_b|}{|T|}$. Let $\overline{\mathcal{D}}^{\mu}_{a,b}$ and $\underline{\mathcal{D}}^{\mu}_{a,b}$ denote the upper bound and the lower bound of the distance between s_a and s_b of spatiotemporal resolution μ respectively, where $\mu \in \{1, 2, \cdots, w_{max}h_{max}\}$. $\overline{\mathcal{D}}^{\mu}_{a,b}$ and $\underline{\mathcal{D}}^{\mu}_{a,b}$ can be calculated iteratively using the bounds of finer resolution $(\mu - 1)$ as follows.[32]

$$\overline{\mathcal{D}}^{\mu}_{a,b} = \begin{cases} \mathcal{D}_{a,b,1,1}, \qquad \text{if} \ \ \mu = 1 \\ \overline{\mathcal{D}}^{\mu-1}_{a,b} + \frac{\delta_{\mu_w} \alpha_{\mu_h}}{\sum_{h=1}^{h_{max}} \sum_{w=1}^{w_{max}} \delta_w \alpha_h} (\mathcal{D}_{a,b,\mu_w,\mu_h} - \mathcal{D}_{a,b,1,1}), \\ \text{if} \ \ \mu \in \{2, 3, \ldots, w_{max}h_{max}\} \end{cases} \qquad (8)$$



(a)

(b)

(c)

Fig. 5. Illustration of the scanning process with window sizes equal to a) $w = 1$ and $h = 1$, b) $w = 2$ and $h = 2$, and c) $w = 2$ and $h = 3$.

$$\underline{\mathcal{D}}^{\mu}_{a,b}=\begin{cases}\mathcal{D}_{a,b,w^*,h^*}+\frac{\delta_1\alpha_1}{\sum_{h=1}^{h_{max}}\sum_{w=1}^{w_{max}}\delta_w\alpha_h}(\mathcal{D}_{a,b,1,1}-\mathcal{D}_{a,b,w^*,h^*}),\\ \text{if } \ \mu=1\\ \underline{\mathcal{D}}^{\mu-1}_{a,b}+\frac{\delta_{\mu_w}\alpha_{\mu_h}}{\sum_{h=1}^{h_{max}}\sum_{w=1}^{w_{max}}\delta_w\alpha_h}(\mathcal{D}_{a,b,\mu_w,\mu_h}-\mathcal{D}_{a,b,w^*,h^*}),\\ \text{if } \ \mu\in\{2,3,\ldots,w_{max}h_{max}\}\end{cases} \tag{9}$$

where μ_w and μ_h denote the spatial window and temporal window corresponding to resolution μ respectively.

Given the current scenario s_C and the spatiotemporal wind scenario database $s = [s_1, s_2, \cdots, s_L]$, the fast query algorithm aiming to find the most similar scenario s_Q to s_C includes the following two steps.

Step 1: Construct a total intensity table $I = [I_1, I_2, \cdots, I_L]$ according to (7). An initial candidate set s_Q is obtained by applying $|I_a - I_C| \leq I_{thrd}$, where I_{thrd} is the threshold that limits the number of retrieved scenarios and is determined by statistical analysis of the database. The initial candidate set s_Q is then fine tuned based on $\overline{\mathcal{D}}^{1}_{a,C}$ and $\underline{\mathcal{D}}^{1}_{a,C}$.

Step 2: Trim s_Q based on $\overline{\mathcal{D}}^{\mu}_{a,C}$ and $\underline{\mathcal{D}}^{\mu}_{a,C}$ at each resolution μ until the most similar scenario is obtained. The algorithm is summarized in Algorithm 4.2. Simulation studies show that by appropriately selecting the threshold I_{thrd}, the fast query algorithm can be tens or hundreds of times faster than exhaustive search.[32]

Algorithm 4.2. Fast Query Algorithm

Input: The current scenario s_C, the spatiotemporal wind scenario database $s = [s_1, s_2, \cdots, s_L]$, and threshold I_{thrd}.

Output: The most similar scenarios $s_Q \subseteq s$ to s_C.

```
1: Construct table $I = [I_1, I_2, \cdots, I_L]^{\mathrm{T}}$ for $s$ according to (7).
2: Calculate $I_C$ according to (7).
3: $s_Q \leftarrow \{s_a\}$, where $s_a \in s$ and $|I_a - I_C| \leq I_{thrd}$.
4: for each $s_a \in s_Q$ do
5:     Calculate $\overline{\mathcal{D}}^{1}_{a,C}$ and $\underline{\mathcal{D}}^{1}_{a,C}$ according to (8) and (9).
6: end for
7: Determine the value of $M_C = \max_{s_a \in s_Q} \overline{\mathcal{D}}^{1}_{a,C}$.
8: while $\underline{\mathcal{D}}^{1}_{a,C} \leq M_C, \forall s_a \in s_Q$ do
9:     Increase the value of $I_{thrd}$ and perform step 3-7.
10: end while
11: $s_Q \leftarrow s_Q \setminus \{s_a\}$, where $\underline{\mathcal{D}}^{1}_{a,C} > M_C$ and $s_a \in s_Q$.
12: for $\mu = 2$ to $w_{max}h_{max}$ do
13:     for each $s_a \in s_Q$ do
14:         Calculate $\overline{\mathcal{D}}^{\mu}_{a,C}$ and $\underline{\mathcal{D}}^{\mu}_{a,C}$.
15:     end for
16:     if $|s_Q| > 1$ then
17:         Determine the value of $M_C = \max_{s_a \in s_Q} \overline{\mathcal{D}}^{\mu}_{a,C}$.
18:         Remove all scenarios $s_a$ that satisfy $\underline{\mathcal{D}}^{\mu}_{a,C} > M_C$ from $s_Q$.
19:     else
20:         Exit from the for loop.
21:     end if
22: end for
23: if $|s_Q| > 1$ then
24:     $s_Q \leftarrow$ the scenario selected from $s_Q$ that have the smallest upper bound values $\overline{\mathcal{D}}^{\mu}_{a,C}$.
25: end if
```

4.3. *Online Tuning and Database Expansion*

The fast query algorithm described in Section 4.2 finds the most similar spatiotemporal wind scenario s_Q in the database to the current wind scenario s_C, together with the optimal multi-UAS trajectory management strategy $[u^Q, \theta^Q]$ under s_Q and the waypoints at the CVP specified time points $R^Q = [R^{Q,t_1}, R^{Q,t_2}, \cdots, R^{Q,t_H}]$. Then we adopt $[u^Q, \theta^Q]$ as the initial solution for UASs under s_C. Because s_Q is not exactly the same as s_C, directly applying $[u^Q, \theta^Q]$ to UASs under s_C may cause deviation or potential collision. To safely and accurately navigate the UASs to their destinations, a simple and fast online tuning is desirable. We here develop an online fine tuning algorithm which includes two procedures, a trajectory correction procedure and a collision avoidance procedure.

Several methods for online trajectory correction can be adopted, including e.g., the robust control-based method,[42,43] stochastic model-based method,[44] and deep learning-based method.[45] In this paper, we develop a data-based trajectory correction procedure which exploits stored single UAS trajectory data. Before we elaborate on the trajectory correction procedure, we first introduce the construction of the single UAS trajectory database s' as follows.

(1) We divide the continuous UAS position space into regularly spaced rectangular cells. Let $[0, L_x]$ and $[0, L_y]$ denote the range on X and Y axes respectively. The resolutions for X and Y axes are set as Δ_x and Δ_y respectively. Hence, there are $\frac{L_x L_y}{\Delta_x \Delta_y}$ rectangular cells in total. The center for the ιth row and κth column cell $[(\iota-1)\Delta_x, \iota\Delta_x] \times [(\kappa-1)\Delta_y, \kappa\Delta_y]$ is denoted as $c_{\iota\kappa} = [(\iota-1)\Delta_x + \frac{\Delta_x}{2}, (\kappa-1)\Delta_y + \frac{\Delta_y}{2}]$, where $\iota \in \{1, 2, \cdots, \frac{L_x}{\Delta_x}\}$ and $\kappa \in \{1, 2, \cdots, \frac{L_y}{\Delta_y}\}$.

(2) For each combination of the wind scenario s_a, cell center $c_{\iota\kappa}$, destination L_{if}, and CVP specified time point t_l, we find the optimal management strategy for a UAS starting at $c_{\iota\kappa}$ at time t_l and arrive at L_{if} at T under wind scenario s_a.

(3) Store all the combinations and their corresponding optimal management strategies and waypoints at the CVP specified time points in the single UAS trajectory database s'.

To summarize, the key elements of s' (See Figure 6) include the scenario tag, spatiotemporal wind scenario data, cell center, destination, the starting time point, optimal management strategies, and waypoints. The size of the single UAS trajectory database s' is calculated as follows:

$$|s'| = LHK \frac{L_x L_y}{\Delta_x \Delta_y}. \tag{10}$$

In the beginning, we apply $[u^Q, \theta^Q]$ to the UASs under the current wind scenario s_C. At each CVP specified time point t_l, where $l \in [1, 2, \cdots, H]$, we record the current waypoints R^{C,t_l} and compare them with the stored waypoints R^{Q,t_l}. If the difference between the stored waypoint and the current waypoint for any UAS i is larger than a pre-specified threshold, i.e., $e_i^{t_l} = |R_i^{C,t_l} - R_i^{Q,t_l}| > \epsilon$, the trajectory correction for UAS i is triggered.

Trajectory Correction Procedure: The trajectory correction repeats the following three steps whenever a deviation occurs at each t_l, until all the UASs arrive at their destinations.

(1) Trim the single UAS trajectory dataset to contain the entries that have the scenario tag s_Q, the starting time point t_l, and the destination L_{if}.

(2) In the trimmed single UAS trajectory dataset, find the entry of which the cell center is closest to the current position of UAS i, i.e., $\min |c_{\iota\kappa} - R^{C,t_l}|$.

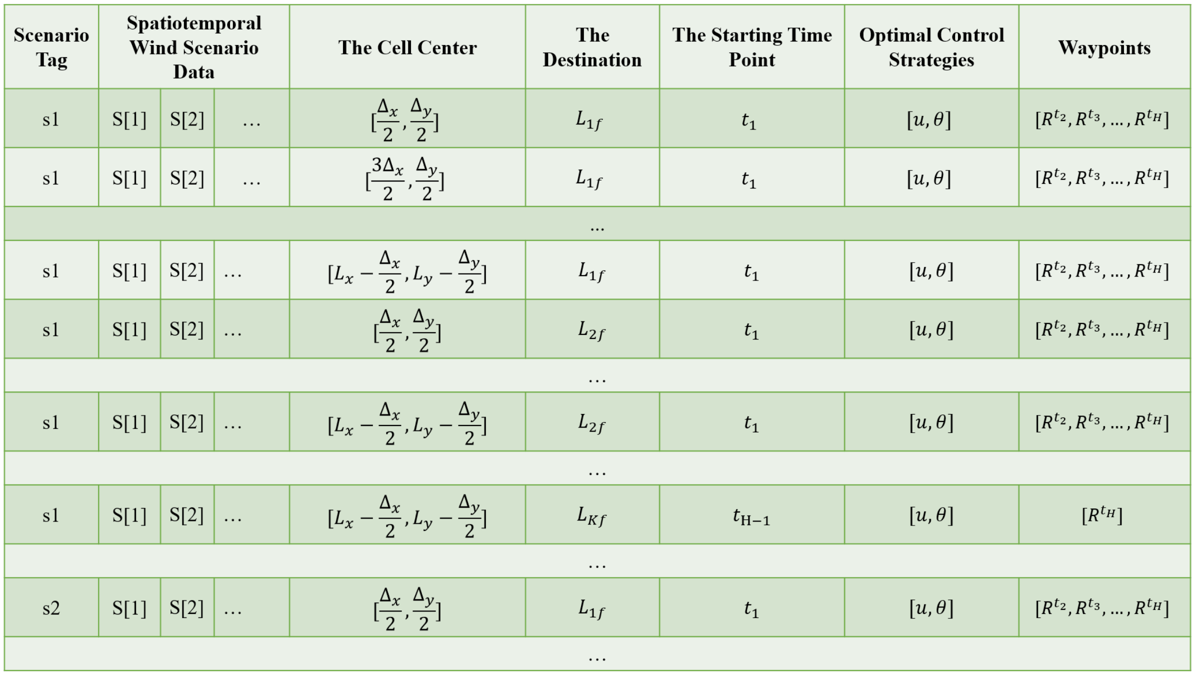
(3) Extract the corresponding optimal single UAS trajectory management strategy and apply it to UAS i.

Collision Avoidance Procedure: To account for the potential collision among UASs, the distance r_{ij} between any pair of UAS i and j is monitored in flight time. If r_{ij} is equal to or smaller than the pre-specified safety distance R, the collision avoidance procedure is triggered. In particular, the optimal management strategies for all the UASs from their current positions to their destinations subject to collision avoidance are recalculated using the CVP method described in Section 3.2. To facilitate the computation in real time, we set the CVP parameters to tolerate a relative low precision for the final results.

The online tuning algorithm is summarized in Algorithm 4.3. We notice that two types of data are needed for the onlline tuning algorithm, including the online data (i.e., the current scenario s_C), and the offline data including the most similar scenario s_Q, the corresponding optimal solution $[u^Q, \theta^Q]$, and the waypoints $R^Q = [R^{Q,t_1}, R^{Q,t_2}, \cdots, R^{Q,t_H}]$.

Once the optimal multi-UAS trajectory strategy $[u^C, \theta^C]$ for the current spatiotemporal wind scenario s_C is obtained, we expand the multi-UAS trajectory database by adding s_C, $[u^C, \theta^C]$, and the waypoints $R^C = [R^{C,t_1}, R^{C,t_2}, \cdots, R^{C,t_H}]$. The single UAS trajectory dataset is expanded in a similar way. Each time a new spatiotemporal wind scenario is obtained, we calculate the single UAS trajectory strategy offline and store the solution in the single UAS trajectory database.

It is worth noting that by combining the offline operations, the database, and the online operations, the optimality of the derived multi-UAS trajectory management solution is guaranteed. Initially, when the offline database is small, it cannot cover various potential wind scenario patterns. In this case, the online tuning plays a major role to find the optimal management solution at the cost of some computation time. The performance can be similar with directly solving the problem using the CVP method which starts with multiple random initial guesses. However, with the expansion of database (i.e., more wind scenarios and their corresponding optimal solutions added to



Scenario Tag	Spatiotemporal Wind Scenario Data			The Cell Center	The Destination	The Starting Time Point	Optimal Control Strategies	Waypoints
s1	S[1]	S[2]	…	$[\frac{\Delta_x}{2},\frac{\Delta_y}{2}]$	L_{1f}	t_1	$[u,\theta]$	$[R^{t_2},R^{t_3},\ldots,R^{t_H}]$
s1	S[1]	S[2]	…	$[\frac{3\Delta_x}{2},\frac{\Delta_y}{2}]$	L_{1f}	t_1	$[u,\theta]$	$[R^{t_2},R^{t_3},\ldots,R^{t_H}]$
…								
s1	S[1]	S[2]	…	$[L_x-\frac{\Delta_x}{2},L_y-\frac{\Delta_y}{2}]$	L_{1f}	t_1	$[u,\theta]$	$[R^{t_2},R^{t_3},\ldots,R^{t_H}]$
s1	S[1]	S[2]	…	$[\frac{\Delta_x}{2},\frac{\Delta_y}{2}]$	L_{2f}	t_1	$[u,\theta]$	$[R^{t_2},R^{t_3},\ldots,R^{t_H}]$
…								
s1	S[1]	S[2]	…	$[L_x-\frac{\Delta_x}{2},L_y-\frac{\Delta_y}{2}]$	L_{2f}	t_1	$[u,\theta]$	$[R^{t_2},R^{t_3},\ldots,R^{t_H}]$
…								
s1	S[1]	S[2]	…	$[L_x-\frac{\Delta_x}{2},L_y-\frac{\Delta_y}{2}]$	L_{Kf}	$t_{\mathrm{H}-1}$	$[u,\theta]$	$[R^{t_H}]$
…								
s2	S[1]	S[2]	…	$[\frac{\Delta_x}{2},\frac{\Delta_y}{2}]$	L_{1f}	t_1	$[u,\theta]$	$[R^{t_2},R^{t_3},\ldots,R^{t_H}]$
…								

Fig. 6. The single UAS trajectory database.

the database), the queried solution is expected to be near-optimal and the decision framework becomes more efficient.

Algorithm 4.3. Online Tuning Algorithm

Input: The current scenario s_C, the most similar scenario s_Q, the corresponding optimal management strategy $[u^Q, \theta^Q]$, the waypoints $R^Q = [R^{Q,t_1}, R^{Q,t_2}, \cdots, R^{Q,t_H}]$, the trajectory deviation threshold ϵ, and the single UAS trajectory dataset s'.

Output: Online optimal management strategy $[u^C, \theta^C]$ for s_C and the corresponding waypoints R^C at the CVP specified time points.

```
for $t \in [0, T]$ do
    Calculate the distance $r_{ij}(t)$ between any two UASs $i$ and $j$ at time $t$ based on $X(t)$, $\forall\ i,\ j \in \{1, 2, \cdots, K\}$ and $i \neq j$.
    if $r_{ij}(t) \leq R$ then
        The collision avoidance procedure is triggered using the CVP method as described in Section 3.2. The optimal management strategy $[u^C(t), \theta^C(t)]$ from the current positions to the destinations $L_f$ is recomputed.
    else
        if $t \in [t_1, t_2, \cdots, t_H]$ and $t = t_l$, where $l \in \{1, 2, \cdots, H\}$ then
            Record the waypoints $R^{C,t_l}$ for all the UASs.
            Calculate the difference between the current and the stored waypoints $e_i^{t_l} = |R_i^{C,t_l} - R_i^{Q,t_l}|$ for each UAS $i$.
            if $e_i^{t_l} > \epsilon$ then
                Query $s'$ and find the entry with the scenario tag $s_Q$, destination $L_{if}$, starting time point $t_l$, and closest cell center to $R^{C,t_l}$, i.e., $\min |c_{\iota\kappa} - R^{C,t_l}|$. Retrieve the corresponding optimal management strategy $[u_i^S(t), \theta_i^S(t)]$.
                $[u_i^C(t), \theta_i^C(t)] \leftarrow [u_i^S(t), \theta_i^S(t)]$.
            else
                $[u_i^C(t), \theta_i^C(t)] \leftarrow [u_i^Q(t), \theta_i^Q(t)]$.
            end if
        end if
    end if
end for
```

5. Simulation Studies

In this section, we demonstrate the proposed spatiotemporal scenario data based decision framework for UTM using simulation studies. We first generate four kinds of spatiotemporal wind spread patterns. Then we apply the CVP method to solve a six-UAS trajectory management problem with different starting points and destinations under various wind scenarios. We store the corresponding optimal solutions and associated waypoints, and construct the database. The performance of the automated multi-UAS data-driven decision framework is then verified using three new representative spatiotemporal wind scenarios.

5.1. *Spatiotemporal Wind Scenarios Generated Using The Influence Model*

We use the influence model to generate 100 spatiotemporal wind scenarios in a wind field of 5×5 regions, each region has an area of $100 \times 100\ m^2$. These wind scenarios are generated using four influence models to capture four spread patterns, i.e., west to east, east to west, north to south, and south to north (See Figure 7 for an example). For each spread pattern, we generate 25 spatiotemporal wind scenarios. Each region is influenced by itself and its neighboring regions. For each region, the wind speed ranges from 1 m/s to 10 m/s with resolution 1 m/s, and the wind direction ranges from 0 to 2π radians with resolution $\frac{1}{36}\pi$ radians. Mathematically, $W_{min} = 1\ m/s$ and $W_{max} = 10\ m/s$, $\delta_w = 1\ m/s$ and $\Delta_w = \frac{1}{36}\pi$ radians. The time horizon is set as $T = 40s$, which is the same as the pre-specified multi-UAS final time T. The sampling period is set to $\Delta T = 1s$.

5.2. *Database Construction*

Consider six UASs operating in the wind field, each of which has its own starting point and destination. UAS 1's starting point is $[0, 0]m$, and destination is $[400, 400]m$. UAS 2's starting point is $[81, 12]m$, and destination is $[250, 330]m$. UAS 3's starting point is $[400, 0]m$, and destination is $[0, 400]m$. UAS 4's starting point is $[400, 200]m$, and destination is $[0, 200]m$. UAS 5's starting point is $[200, 400]m$, and destination is $[200, 0]m$. UAS 6's starting point is $[150, 372]m$, and destination is $[350, 120]m$. The thrust of each UAS is bounded by $u_{max} = 1.5\ m/s^2$. The parameters for CVP are set as follows. $\beta = 1$, $\rho = 100$, $\alpha_1 = 12$, $\alpha_2 = 12$, $H = 5$. The relative error tolerance, absolute error tolerance, termination tolerance on function value and termination tolerance on the current point are all set as 10^{-6} for the offline optimization calculation. For the online collision avoidance procedure, these tolerance values are all set to 10^{-2}. The parameters for single UAS trajectory data are set as $\Delta_x = 15m$ and $\Delta_y = 15m$. The safety distance is set as $R = 35m$. We follow the offline operations to obtain the optimal trajectory solutions and construct the multi-UAS database for the 100 spatiotemporal wind scenarios. Figure 8 shows an example of the optimal six UASs trajectory solution. Figure 8 (a) shows the trajectories of the 6 UASs accurately arriving at their destinations at T. Figure 8 (b) and (c) show the optimal thrusts and their angles respectively. Figure 8 (d) shows the spatiotemporal wind scenario under which the optimal multi-UAS trajectory solution is calculated. It takes $1820s$ for the offline CVP method to compute the optimal multi-UAS trajectory solution, using a Dell XPS 13 laptop with CPU clock time up to 4.9 GHz. The computational time suggests the high computational complexity of multi-UAV trajectory management subject to collision avoidance constraints. The next procedures bring this online by significantly reducing the computation.

5.3. *Performance of the Spatiotemporal Scenario Data-driven Decision Framework For Multi-UAS Trajectory Management*

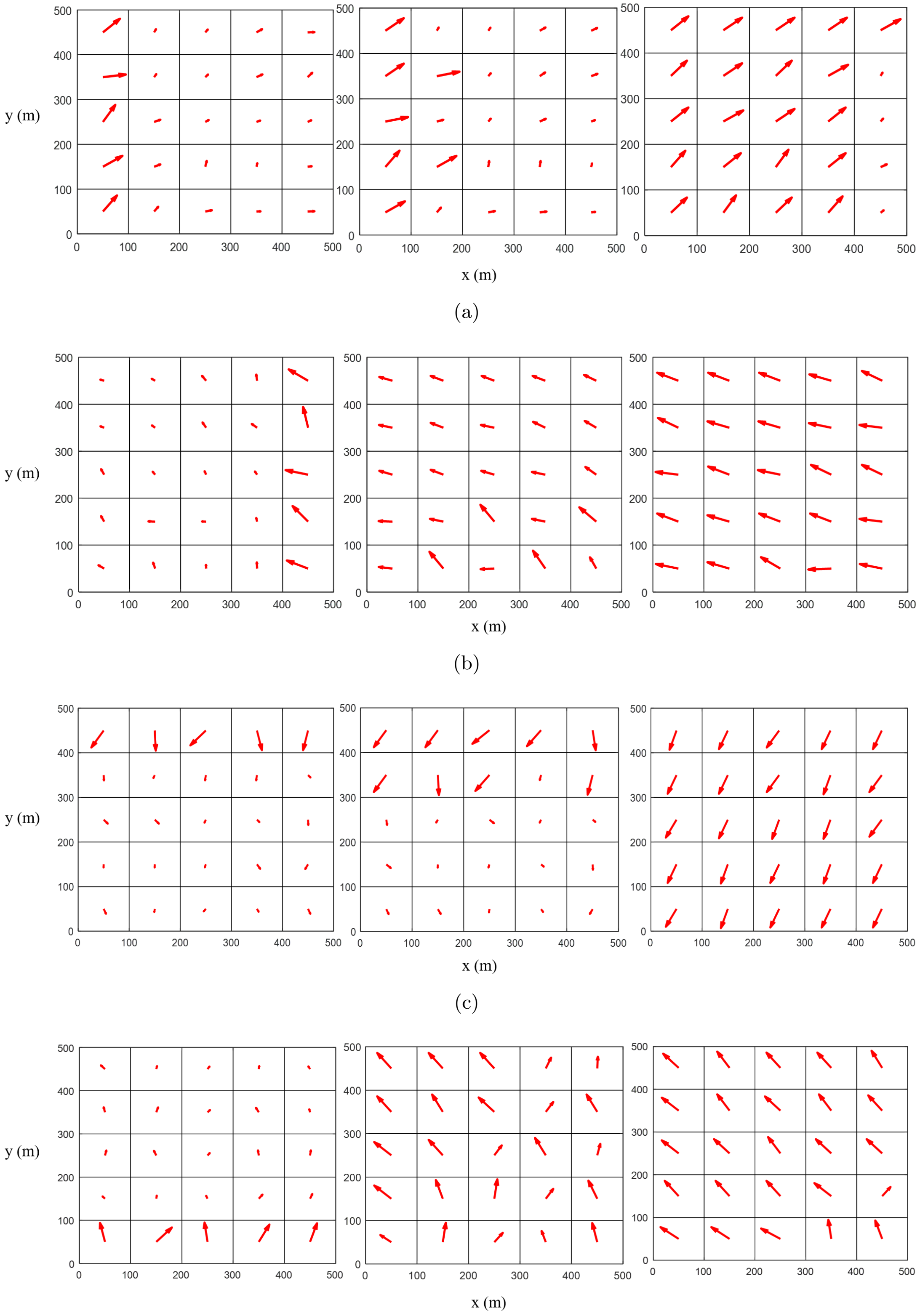
We conduct three representative simulation studies to show the performance of the spatiotemporal scenario data-driven decision framework for multi-UAS trajectory management. In the first case study, we generate a new spatiotemporal wind scenario s_{C1} which is very similar to the spatiotemporal wind scenario s_Q shown in Figure 8 (d). No online tuning operation is needed. In the second case study, the new spatiotemporal wind scenario s_{C2} differs to some degree from the wind scenario s_Q shown in Figure 8 (d), and the trajectory correction procedure for a single UAS is triggered. In the third case study, the new spatiotemporal wind scenario s_{C3} differs significantly from the wind scenario s_Q shown in Figure 8 (d), and the collision avoidance procedure is triggered to avoid potential collision.

5.3.1. *Case 1:*

The new spatiotemporal wind scenario s_{C1} in Figure 9 (d) is generated using the same influence model parameters as s_Q shown in Figure 8 (d). Therefore, s_{C1} is very similar to s_Q. Figure 9 shows s_{C1} and the optimal six UASs trajectory solution for it. Figure 9 (a) shows the trajectories of the six UASs. Compared with Figure 8 (a), we can see that the final position of UAS 3 deviates slightly from its destination, while the other UASs arrive at their destinations with good accuracy. The deviation occurs because s_{C1} is not exactly the same as s_Q. Figure 9 (b) and (c) show the optimal thrusts and their angles. We note that they are exactly the same as in Figure 8 (b) and (c), indicating that no online fine tuning is triggered. The total computation time is $3.8s$ ($0.7s$ for fast query and $3.1s$ for UAS movement simulation), significantly accelerating the online decision procedure.

5.3.2. *Case 2:*

The new spatiotemporal wind scenario s_{C2} in Figure 10 (d) differs to some degree from s_Q shown in Figure 8 (d) because its underlying influence model parameters is updated. In this case, the trajectory correction procedure is triggered during the online tuning. Figure 10 shows s_{C2} and the optimal six UASs trajectory solution for it. Figure 10 (a) shows the trajectories of the six UASs. Compared with Figure 8 (a), we can see that the final positions of all the UASs deviate slightly from their destinations. Figure 10 (b) and (c) show the optimal thrusts and their angles. We note that the angle of the thrust for UAS 2 changes slightly at CVP specified time point $t_3 = 24s$, indicating that the trajectory correction procedure is triggered at that time. The total computation time is $18.7s$, which includes $0.8s$ for fast query, $12s$ for online tuning and $5.9s$ for UASs



(a)

(b)

(c)

(d)

Fig. 7. Examples for four spatiotemporal wind spread patterns. The arrow at each region denotes the wind speed and direction. (a) West to East. (b) East to West. (c) North to South. (d) South to North. Each subfigure contains 3 sample snapshots at $t = 1s$, $t = 18s$, and $t = 35s$.

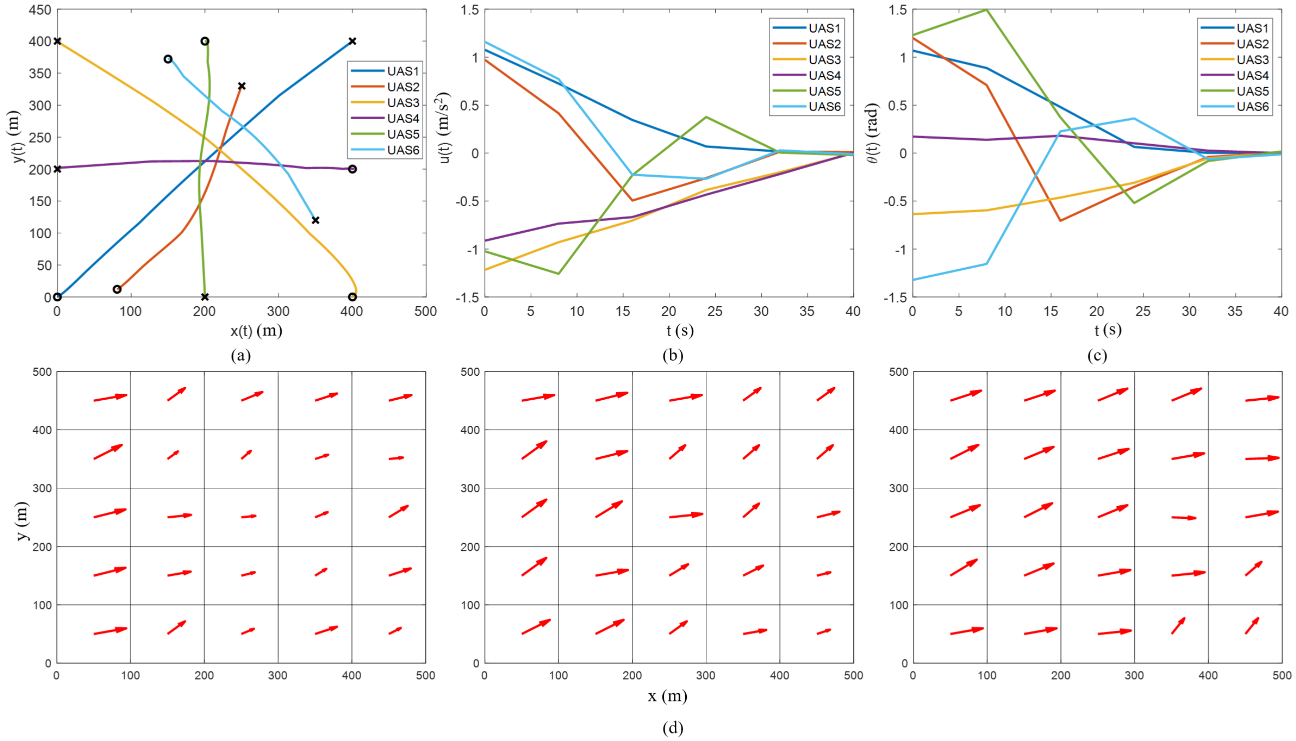


Fig. 8. An example for the optimal six UASs trajectory solution. (a) The trajectories of the 6 UASs. 'Circle' denotes the starting point and 'X' denotes the destination. (b) The thrusts. (c) The angles of the thrusts. (d) The sample snapshots of the spatiotemporal wind scenario at $t = 5s$, $t = 22s$, and $t = 37s$.

movement simulation. The total computation time is longer than Case 1, because correcting the current trajectory introduces additional time.

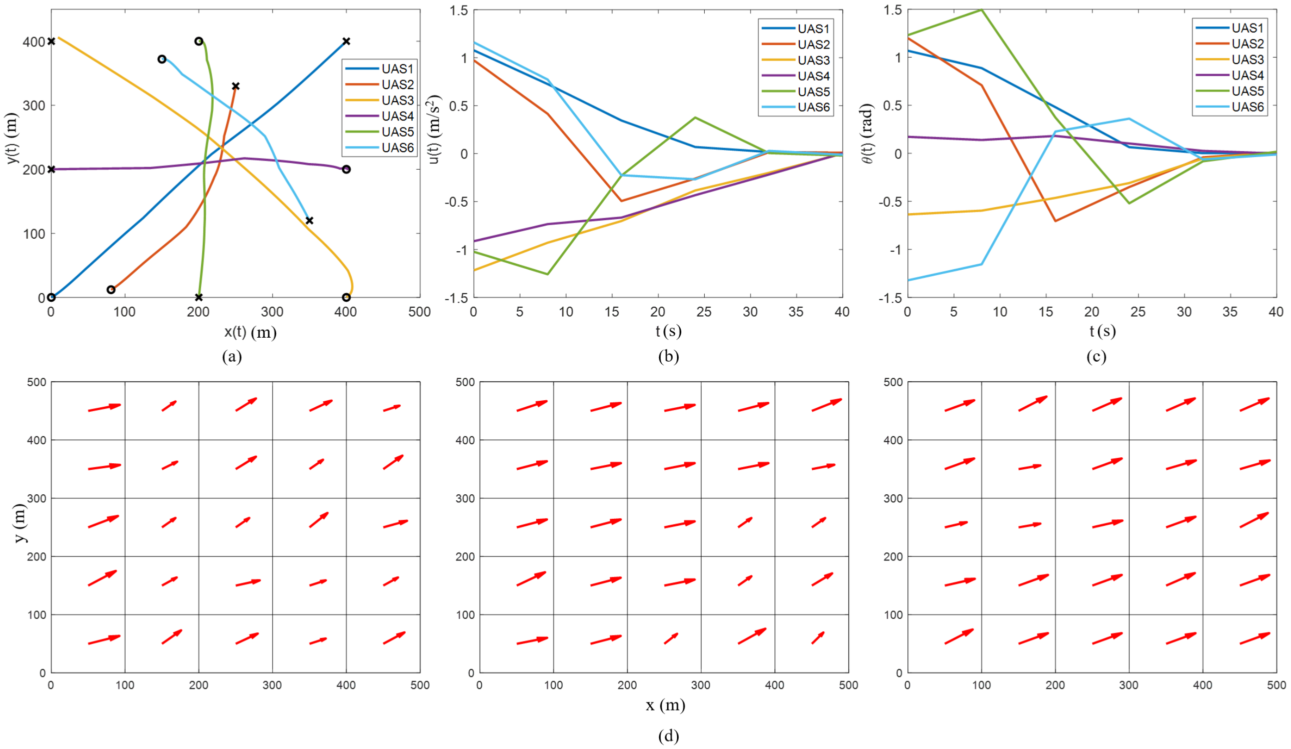
5.3.3. *Case 3:*

The new spatiotemporal wind scenario s_{C3} in Figure 11 (d) differs significantly from s_Q shown in Figure 8 (d). In this case, potential collision exists and the online tuning triggers the collision avoidance procedure. Figure 11 shows s_{C3} and the optimal six UASs trajectory solution for it. Figure 11 (a) shows the trajectories of the six UASs. We see that all the UASs arrive at their destinations with good accuracy. However, their trajectories differ significantly from the trajectories shown in Figure 8 (a). Figure 11 (b) and (c) show the optimal thrusts and their angles. We note visible changes for the management strategies of all the UASs at time $t = 16s$, indicating that the collision avoidance procedure is triggered at that time. The total computation time is $92.7s$ ($0.7s$ for fast query, $85s$ for the online tuning and $7s$ for the UAS movement simulation), longer than Case 1 and Case 2 because the collision avoidance procedure recalculates the optimal management strategy for all the UASs. The computation time is 5% of the total time for the offline optimization ($1820s$). That is because with the help of the queried initial management solution, only the last few control stages of the CVP need to be recalculated, instead of the whole trajectories. Hence we have significant computation reduction.

Combining the simulation results of the three cases, we note that the multi-UAS online trajectory management is significantly expedited by using the offline, online, and database integrated decision framework. In addition, the safety and accuracy are also guaranteed through the adaptive online tuning. With the expansion of database (i.e., more wind scenarios and their corresponding optimal solutions added to the database), the queried solution is expected to be near-optimal and the decision framework becomes more efficient.

6. Conclusions

In this paper, a spatiotemporal scenario data-driven decision framework which automates the playbook of UTM for multi-UAS trajectory management is developed. The proposed framework contains offline operations, online operations and a spatiotemporal scenario database. The offline operations feature an influence model and an optimal trajectory management planner to generate spatiotemporal scenario data, obtain optimal strategies, and construct the database. The online operations feature a fast query algorithm for spatiotemporal scenario data, online tuning



(a) (b) (c) (d)

Fig. 9. The optimal six UASs trajectory solution for s_{C1}. (a) The trajectories of the 6 UASs. (b) The thrusts. (c) The angles of the thrusts. (d) The sample snapshots of s_{C1} at $t = 5s$, $t = 22s$, and $t = 37s$.

that deals with trajectory deviation and potential collision, and database expansion. The offline and online operations are integrated to automate the UTM playbook and provide computationally efficient decisions based on the similarities of spatiotemporal scenarios. The simulation studies show that the online decision time for the multi-UAS trajectory management problem is significantly shortened, while safety and accuracy are also guaranteed. The decision framework can also be utilized in other applications where spatiotemporal environmental impact plays a crucial rule in the decision process. In our future work, we will adopt a more complex 3D dynamic UAS model for the multi-UAS trajectory management problem, study machine learning-based online tuning algorithms, implement the MATLAB prototype using Java, and also consider probabilistic spatiotemporal scenario forecasts.

Acknowledgments

We would like to thank NSF grants 1714519, 1730675, 1724248, 1839804 and 1953049 for the support of this work.

References

[1] F. B. Sorbelli, F. Corò, S. K. Das and C. M. Pinotti, Energy-constrained delivery of goods with drones under varying wind conditions, *IEEE Transactions on Intelligent Transportation Systems* (2020).

[2] S. Sankarasrinivasan, E. Balasubramanian, K. Karthik, U. Chandrasekar and R. Gupta, Health monitoring of civil structures with integrated uav and image processing system, *Procedia Computer Science* **54** (2015) 508–515.

[3] P. Tokekar, J. Vander Hook, D. Mulla and V. Isler, Sensor planning for a symbiotic uav and ugv system for precision agriculture, *IEEE Transactions on Robotics* **32**(6) (2016) 1498–1511.

[4] S. Li, C. He, M. Liu, Y. Wan, Y. Gu, J. Xie, S. Fu and K. Lu, Design and implementation of aerial communication using directional antennas: learning control in unknown communication environments, *IET Control Theory & Applications* **13**(17) (2019) 2906–2916.

[5] G. Skorobogatov, C. Barrado and E. Salamí, Multiple uav systems: a survey, *Unmanned Systems* **8**(02) (2020) 149–169.

[6] T. Jiang, J. Geller, D. Ni and J. Collura, Unmanned aircraft system traffic management: concept of operation and system architecture, *International journal of transportation science and technology* **5**(3) (2016) 123–135.

[7] Unmanned Aircraft Systems (UAS) Traffic Management (UTM) Concept of Operations, V2.0 `https://utm.arc.nasa.gov/docs/`

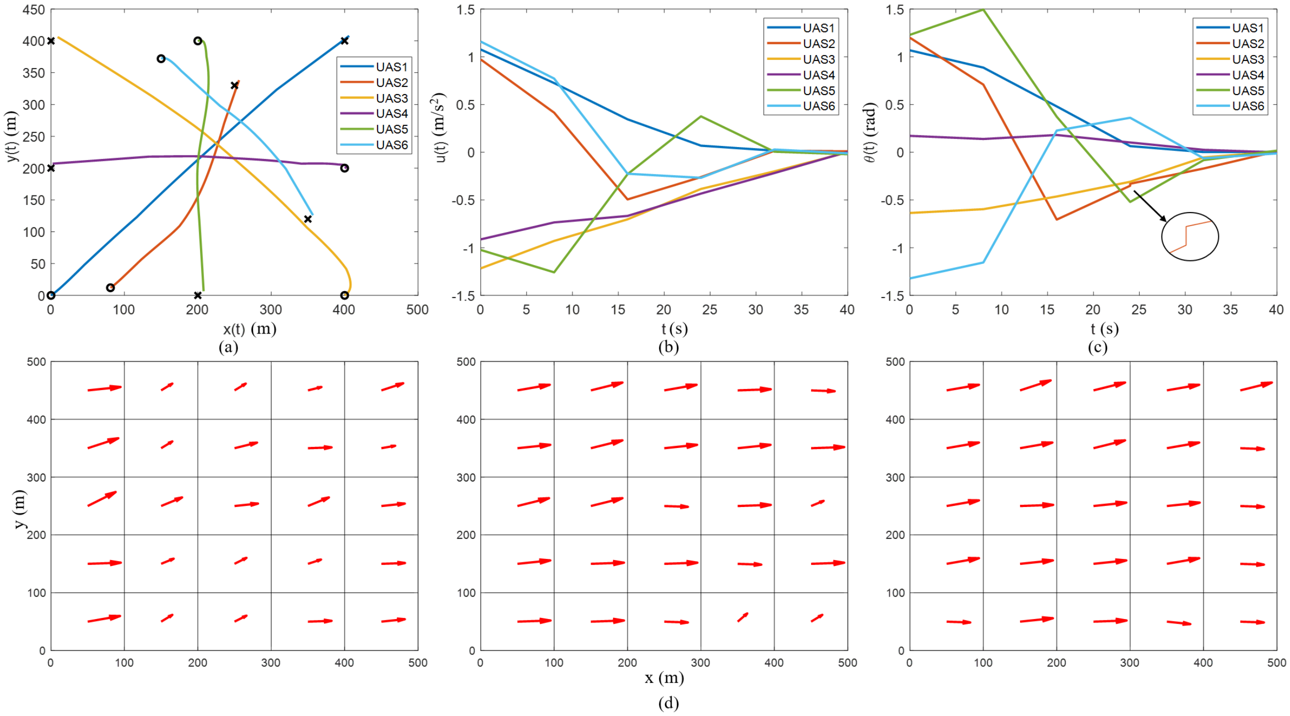


Fig. 10. The optimal six UASs trajectory solution for s_{C2}. (a) The trajectories of the 6 UASs. (b) The thrusts. (c) The angles of the thrusts. (d) The sample snapshots of s_{C2} at $t = 5s$, $t = 22s$, and $t = 37s$.

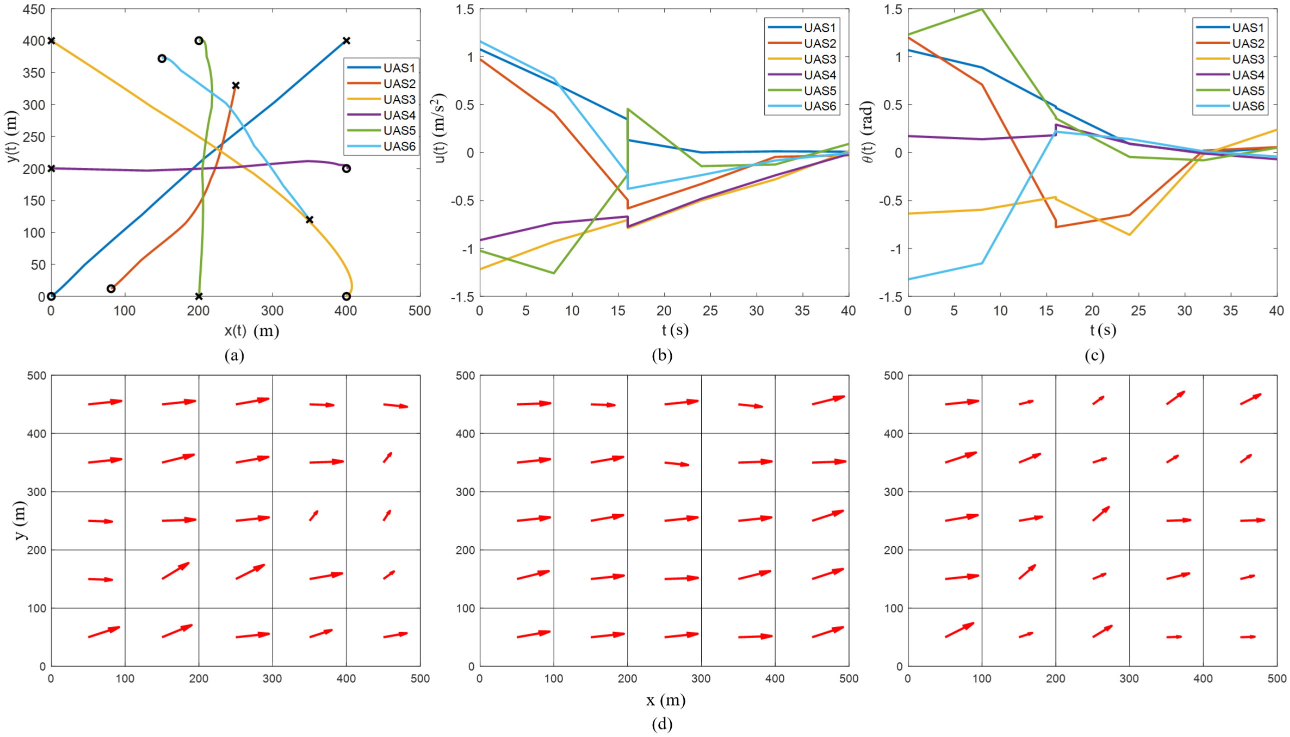


Fig. 11. The optimal six UASs trajectory solution for s_{C3}. (a) The trajectories of the 6 UASs. (b) The thrusts. (c) The angles of the thrusts. (d) The sample snapshots of s_{C3} at $t = 5s$, $t = 22s$, and $t = 37s$.

2020-03-FAA-NextGen-UTM_ConOps_v2.pdf, (2020), Accessed: 2020-03-02.

[8] P. H. Kopardekar, Unmanned aerial system (uas) traffic management (utm): Enabling low-altitude airspace and uas operations (April 2014).

[9] F. A. Administration, National playbook (March 2020).

[10] G. Cai, J. Dias and L. Seneviratne, A survey of small-scale unmanned aerial vehicles: Recent advances and future development trends, *Unmanned Systems* **2**(02) (2014) 175–199.

[11] M. Liu, Y. Wan, F. L. Lewis, E. Atkins and D. O. Wu, Statistical properties and airspace capacity for unmanned aerial vehicle networks subject to sense-and-avoid safety protocols, *IEEE Transactions on Intelligent Transportation Systems* (2020).

[12] R. Guirado, J.-C. Padró, A. Zoroa, J. Olivert, A. Bukva and P. Cavestany, Stratotrans: Unmanned aerial system (uas) 4g communication framework applied on the monitoring of road traffic and linear infrastructure, *Drones* **5**(1) (2021) p. 10.

[13] M. A. Pinheiro, M. Liu, Y. Wan and A. Dogan, On the analysis of on-board sensing and off-board sensing through wireless communication for uav path planning in wind fields, *AIAA Scitech 2019 Forum*, San Diego, California (January 2019).

[14] L. Ren, M. Castillo-Effen, H. Yu, Y. Yoon, T. Nakamura, E. N. Johnson and C. A. Ippolito, Small unmanned aircraft system (suas) trajectory modeling in support of uas traffic management (utm), *17th AIAA Aviation Technology, Integration, and Operations Conference*, (2017), p. 4268.

[15] F. Fabra, C. T. Calafate, J.-C. Cano and P. Manzoni, Mbcap: Mission based collision avoidance protocol for uavs, *2018 IEEE 32nd International Conference on Advanced Information Networking and Applications (AINA)*, IEEE (2018), pp. 579–586.

[16] C. Y. Tan, S. Huang, K. K. Tan, R. S. H. Teo, W. Q. Liu and F. Lin, Collision avoidance design on unmanned aerial vehicle in 3d space, *Unmanned Systems* **6**(04) (2018) 277–295.

[17] O. Souissi, R. Benatitallah, D. Duvivier, A. Artiba, N. Belanger and P. Feyzeau, Path planning: A 2013 survey, *Proceedings of 2013 International Conference on Industrial Engineering and Systems Management (IESM)*, Rabat, Morocco (October 2013).

[18] X. Yang and P. Wei, Scalable multi-agent computational guidance with separation assurance for autonomous urban air mobility operations, *Journal of Guidance, Control, and Dynamics* (2020).

[19] Q. Wang and C. Phillips, Cooperative path-planning for multi-vehicle systems, *electronics* **3**(4) (2014) 636–660.

[20] I. K. Nikolos, K. P. Valavanis, N. C. Tsourveloudis and A. N. Kostaras, Evolutionary algorithm based offline/online path planner for uav navigation, *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* **33**(6) (2003) 898–912.

[21] C. He, Y. Wan, Y. Gu and F. L. Lewis, Integral reinforcement learning-based multi-robot minimum time-energy path planning subject to collision avoidance and unknown environmental disturbances, *IEEE Control Systems Letters* **5**(3) (2020) 983–988.

[22] C. He, Y. Wan, Y. Gu and F. L. Lewis, Integral reinforcement learning-based approximate minimum time-energy path planning in an unknown environment, *International Journal of Robust and Nonlinear Control* (2020).

[23] M. Radmanesh, M. Kumar, P. H. Guentert and M. Sarim, Overview of path-planning and obstacle avoidance algorithms for uavs: a comparative study, *Unmanned systems* **6**(02) (2018) 95–118.

[24] J. Xie, Y. Wan, Y. Zhou, S.-L. Tien, E. P. Vargo, C. Taylor and C. Wanke, Distance measure to cluster spatiotemporal scenarios for strategic air traffic management, *Journal of Aerospace Information Systems* **12**(8) (2015) 545–563.

[25] C. He, Y. Wan and J. Xie, Spatiotemporal scenario data-driven decision for the path planning of multiple uass, *Proceedings of the Fourth Workshop on International Science of Smart City Operations and Platforms Engineering*, (2019), pp. 7–12.

[26] J. A. Guerrero and Y. Bestaoui, Uav path planning for structure inspection in windy environments, *Journal of Intelligent & Robotic Systems* **69**(1-4) (2013) 297–311.

[27] J. Han, M. Kamber and A. K. Tung, Spatial clustering methods in data mining, *Geographic data mining and knowledge discovery* (2001) 188–217.

[28] C. S. Möller-Levet, F. Klawonn, K.-H. Cho and O. Wolkenhauer, Fuzzy clustering of short time-series and unevenly distributed sampling points, *International Symposium on Intelligent Data Analysis*, Berlin, Germany (August 2003).

[29] K. Kalpakis, D. Gada and V. Puttagunta, Distance measures for effective clustering of arima time-series, *Proceedings 2001 IEEE international conference on data mining*, San Jose, California (November 2001).

[30] H. F. Tork, Spatio-temporal clustering methods classification, *Doctoral Symposium on Informatics Engineering*, Porto,Portugal (January 2012).

[31] J. Xie and Y. Wan, Scalable multidimensional uncertainty evaluation approach to strategic air traffic flow management, *AIAA Modeling and Simulation Technologies Conference*, Dallas, TX (June 2015).

[32] J. Xie, A. R. Kothapally, Y. Wan, C. He, C. Taylor and C. Wanke, Similarity search of spatiotemporal scenario data for strategic air traffic management, *Journal of Aerospace Information Systems* (2018, accepted).

[33] C. He, Y. Wan and F. L. Lewis, On the identifiability of the influence model for stochastic spatiotemporal spread processes, *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans* (November submitted, 2018).

[34] C. He and Y. Wan, Clustering stochastic weather sce-

narios using influence model-based distance measures, *AIAA Aviation Conference*, Dallas, TX (June submitted, 2019).

[35] A. Chakrabarty and J. Langelaan, Uav flight path planning in time varying complex wind-fields, *2013 American control conference*, IEEE (2013), pp. 2568–2574.

[36] S. Yang, N. Wei, S. Jeon, R. Bencatel and A. Girard, Real-time optimal path planning and wind estimation using gaussian process regression for precision airdrop, *2017 American Control Conference (ACC)*, IEEE (2017), pp. 2582–2587.

[37] P. Pradeep and P. Wei, Energy-efficient arrival with rta constraint for multirotor evtol in urban air mobility, *Journal of Aerospace Information Systems* **16**(7) (2019) 263–277.

[38] C. Asavathiratham, The influence model: A tractable representation for the dynamics of networked markov chains, PhD thesis, Massachusetts Institute of Technology (2001).

[39] A. Mohamed, J. Ren, A. M. Sharaf and M. EI-Gindy, Optimal path planning for unmanned ground vehicles using potential field method and optimal control method, *International journal of vehicle performance* **4**(1) (2018) 1–14.

[40] G. Huang, Y. Lu and Y. Nan, A survey of numerical algorithms for trajectory optimization of flight vehicles, *Science China Technological Sciences* **55**(9) (2012) 2538–2560.

[41] H. Hadiyanto, D. Esveld, R. Boom, G. Van Straten and A. Van Boxtel, Control vector parameterization with sensitivity based refinement applied to baking optimization, *Food and Bioproducts Processing* **86**(2) (2008) 130–141.

[42] K. Wu, B. Fan and X. Zhang, Trajectory following control of uavs with wind disturbance, *2017 36th Chinese Control Conference (CCC)*, IEEE (2017), pp. 4993–4997.

[43] T. G. McGee and J. K. Hedrick, Path planning and control for multiple point surveillance by an unmanned aircraft in wind, *2006 American Control Conference*, IEEE (2006), pp. 6–pp.

[44] Y. Zhou and G. S. Chirikjian, Planning for noise-induced trajectory bias in nonholonomic robots with uncertainty, *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA'04. 2004*, **5**, IEEE (2004), pp. 4596–4601.

[45] E. J. Shamwell, S. Leung and W. D. Nothwang, Vision-aided absolute trajectory estimation using an unsupervised deep network with online error correction, *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE (2018), pp. 2524–2531.