JOURNAL OF AEROSPACE INFORMATION SYSTEMS
Vol. 13, No. 2, February 2016

Cooperative Task Allocation for Unmanned Vehicles with Communication Delays and Conflict Resolution

Eloy Garcia* and David W. Casbeer†
Air Force Research Laboratory, Wright–Patterson Air Force Base, Ohio 45433

DOI: 10.2514/1.I010218

This paper presents a framework for multi-agent cooperative decision making under communication constraints. Piece-wise optimal decentralized allocation of tasks for a group of unmanned aerial vehicles is considered. Each vehicle estimates the position of all teammates in order to assign tasks based on these estimates, and it also implements event-based broadcasting strategies that allow the group of agents to use communication resources more efficiently. The agents implement a simple decentralized auction scheme in order to resolve possible conflicts. This framework is extended to re-plan assignments when a sequence of tasks is disrupted by wind disturbance. Further, the effects of wind disturbance are also included in the task assignment process by assuming a fixed steady wind value. The overall framework provides a robust and communication-efficient approach for decentralized task allocation in the presence of communication constraints and external uncertainties.

Nomenclature

D	=	total distance from the beginning of mission
d	=	distance between two consecutive tasks
i	=	agent index
j	=	target index
k	=	task assignment stage index
N_t	=	total number of targets
N_u	=	total number of agents
n	=	discrete-time index
$\mathbb{R}$	=	set of real numbers
u	=	control input
v_x, v_y	=	velocities in Cartesian coordinates
x, y	=	positions in Cartesian coordinates
$\boldsymbol{x}^{i,l}$	=	model state of agent l implemented by agent i
$\mathbb{Z}_+$	=	set of positive integers
θ	=	azimuth flight path angle
$\Omega_{\max}$	=	maximum turning rate
ω	=	exogenous input
τ_{il}	=	communication delay from agent i to agent l

I. Introduction

MISSIONS consisting of groups of unmanned aerial vehicles (UAVs) require a high degree of coordination for successfully achieving desired goals. Usually, this coordination has to be reached in the presence of multiple uncertainties in the environment and in the communication channels. In many types of applications, a group of coordinated agents is potentially able to outperform a single or a number of systems operating independently. The specific application considered in this paper involves the decentralized assignment of tasks in which agents can only receive delayed measurements from other vehicles and the environment disturbances are able to disrupt the planned sequence of actions.

Different authors have addressed multi-agent task assignment problems (TAPs) for UAVs. For instance, Dionne and Rabbath [1] presented a robust task assignment algorithm for uncertain environments. Alighanbari and How [2] provided a decentralized task consensus algorithm with asynchronous communication. Decisions to communicate are taken by each agent independently based on the outcomes of assignments using different sets of information. Sujit and Beard [3] addressed the task allocation problem for miniature air vehicles (MAV) that perform search and destroy missions. Challenges particular to this problem are the MAV kinematic constraints and their limited sensor and communication ranges. The TAP is solved by means of a distributed auction scheme. Sujit et al. [4] considered a UAV assignment problem in which each vehicle is able to carry different types of resources and in different quantities; also, to engage a given target, predetermined types and quantities of resources are needed. A particle swarm optimization method is used to solve this complex assignment scenario. Cooperative TAPs have also been considered for perimeter patrol and persistent surveillance. In [5], multi-UAV control and planning for persistent surveillance is studied. One of the features in [5] involves the optimal assignment of UAVs to target spaces in order to perform surveillance tasks. Different challenges in this problem include UAVs with different speeds and minimum turning radii, in addition to dynamically changing environments.

In [6], a coordinated path planning for multiple vehicles in an environment containing multiple radar sites was studied. The main UAV objectives in that reference are the minimization of radar exposure and the simultaneous arrival to the main target by all UAVs. Beard et al. [7] extended the multi-UAV coordination algorithm described in [6] and considered multiple targets. In this case, the UAVs need to select the target to

Presented as Paper 2013-4580 at the AIAA Infotech@Aerospace Conference, Boston, MA, 19–22 August 2013; received 23 December 2013; revision received 7 August 2015; accepted for publication 10 October 2015; published online 21 December 2015.

*Controls Research Scientist (Contractor: Infoscitex Corporation), Control Science Center of Excellence; elgarcia@infoscitex.com. Member AIAA.
†Research Engineer, Control Science Center of Excellence; david.casbeer@us.af.mil. Senior Member AIAA.

visit according to individual and group criteria. A satisficing approach is implemented in that paper in order for the vehicles to get assigned to a particular target while following a path that reduces radar exposure and ensures simultaneous arrival with respect to other UAVs that are assigned to the same target.

The consensus-based bundle algorithm (CBBA) [8] is a decentralized algorithm for task allocation when all-to-all communication among nodes is not possible. It requires from several to many iterations in order for the agents to converge to a list of assigned tasks. In [9], the CBBA is extended to address re-planning in dynamic environments and considering agents with different capabilities. New plans are obtained to reflect the changing nature of the environment; specifically, new tasks need to be accounted for and older or irrelevant tasks are pruned. Johnson et al. [10] used a similar approach, the asynchronous CBBA (ACBBA), for agents that communicate over an asynchronous channel. The ACBBA offers the advantage of local re-planning in which agents are able to decide on individual re-plans that, possibly, affect only a small number of agents. The paper [11] presented a situational awareness algorithm to solve multi-agent task allocation problems. In this approach, agents try to predict what neighbors might bid on, in order to obtain more informed decisions, in a cooperative sense.

The CBBA has also been extended to consider coupled constraints [12,13]. The work in [13], in particular, considers the assignment of tasks with assignment constraints and also with different types of temporal constraints. It addresses similar cases of temporal precedence as in the present paper; however, it is assumed in [13] that assigned tasks will be executed in that order. Because of external uncertainties, this may not be the case. In this paper, we consider wind disturbances. Winds change the time of arrival to perform any given task. Because we want tasks to be performed in certain order, winds will potentially disrupt the desired sequence of tasks. We consider a different approach that predicts possible different executions to what was planned. A call for re-planning is generated when at least one agent detects a possible temporal disruption of scheduled tasks. Our approach also reduces the number of transmitted messages during the task assignment process because each agent assigns tasks based on estimated positions of all other agents. Another extension to the CBBA is provided in [14]. This paper considers the decentralized surveillance problem by a team of robots. Tasks are assigned to each robot with the additional constraint that a subset of the tasks needs to be assigned. This type of tasks is called critical tasks. Binetti et al. [14] use the CBBA incorporating hard constraints in order to ensure that the critical tasks are not left unassigned.

The task allocation problem has been studied in different research areas. For instance, Wellman et al. [15] provided market mechanisms for distributed allocation of resources and using asynchronous communication. Common concerns in distributed TAPs were addressed in this paper, such as minimizing communication overhead, reaching closure within reasonable time and computational cost, and using local information to make some decisions without knowing private information of other agents.

The specific problem studied in the present paper is the wide area search munitions problem [16–18]. This is a special type of TAP in the sense that several tasks need to be performed on the same target and in strict sequential order. Shima et al. [19] showed that, under the presence of communication delays, improved performance can be obtained when each UAV estimates the current positions of the other UAVs in the group. It was also shown in [19] that transmission of measurements can be efficiently scheduled in terms of errors between local estimates and information related to the previous transmission.

The main contributions of this work are as follows: an algorithm for the decentralized computation of piece-wise optimal and conflict-free assignment plans that also reduces interagent communication is proposed. Because of the presence of time-varying communication delays, measurement noise, and wind disturbances, extensions to this algorithm are formulated in order for the UAVs to autonomously decide, at any time during the mission, if the assignment of tasks needs to be executed once again due to the mentioned uncertainties. This approach provides a framework for an assignment of tasks that adapts to the current conditions on the environment and the communication channels. The contents of this paper extend the results in [20].

The present paper is organized as follows: Sec. II describes the problem. Section III presents the approach used in this paper for event-triggered decentralized estimation of vehicle positions and for task assignments. Section IV focuses on estimating and resolving possible conflicts on the assignment of tasks. Re-planning due to the presence of wind disturbances is addressed in Sec. V. In Sec. VI, steady wind disturbance values are considered during the assignment process to help preserve the sequential order of tasks during the mission. Conclusions are presented in Sec. VII.

II. Problem Statement

We consider a group of UAVs

$$U = \{1, 2, \ldots, N_u\} \tag{1}$$

each one modeled using Dubins dynamics:

$$\dot{x}_i = v \cdot \cos\theta_i \tag{2}$$

$$\dot{y}_i = v \cdot \sin\theta_i \tag{3}$$

$$\dot{\theta}_i = \Omega_{\max} \cdot u_i \tag{4}$$

where the control input u_i is restricted to take only on three values:

$$u_i = \{-1, 0, 1\} \tag{5}$$

We assume that the maximum turning rate $\Omega_{\max}$ and the nominal speed v are the same for all UAVs $i \in U$ in order to simplify the notation. However, the framework described in this paper will work in the same way when speeds and maximum turning rates are different for every agent (speeds remain constant). In this case it is only necessary for the agents to have knowledge of these quantities for every other agent in the group. The UAVs' model is a 2-D Dubins vehicle and it is assumed that each UAV is able to maintain constant altitude.

The scenario under consideration is the so-called wide area search munitions [17,18] in which UAVs are required to perform different tasks on stationary targets with known locations. The tasks for each one of the targets include classify, attack, and verify the destruction of such targets.

A searching for targets task is a default mode when the vehicle is not assigned any other task or when it has completed all of its assignments. The tasks classification, attack, and verification of damage need to be performed in sequential manner for any given task and with relative timing constraints resulting in coupled assignments of vehicles performing tasks on the same target. The classification task has an associated heading constraint and it means that the agents performing this task need to approach a target at a specified angle. Attack and verification tasks need only proximity requirements. Attack needs proximity equal to zero and the proximity requirement for verification is a positive constant. Any angle is acceptable for these two tasks.

The problem we concern ourselves is the optimal and conflict-free decentralized computation of assignments in the presence of communication delays, measurement noise, and wind disturbance. By conflict-free, we refer to the correct assignment of tasks to UAVs in which a given task needs to be assigned to one and only one UAV and the tasks on the same target need to be performed in certain order. The term does not refer to aircraft collision avoidance. UAV collision avoidance capabilities are assumed in this paper. The decentralized characteristic of the algorithm is necessary because all-to-all communication among agents is not assumed. Instead, agents will communicate only with a subset of agents or neighbors. Each agent will estimate the position of every other agent and obtain a list of assignments for the group. Local estimated positions as well as the estimates of other agent positions are affected by zero mean white sensor noise. The estimated positions of other agents are also affected by additional uncertainties such as communication delays, lack of continuous communication among agents, and wind disturbance. Potential conflicts in the list of assignments will be flagged by any given agent that requires every agent to respond with a bid in order to solve the estimated conflict. Each agent's bid will contain the agent ID, the task, and the target he is placing its bid on, and the corresponding estimated cost. Transmission of this information is subject to time-varying and bounded communication delays. The cost function to be minimized is the cumulative distance the vehicles travel in order to perform all required tasks:

$$J = \sum_{i=1}^{N_u} D_i > 0 \tag{6}$$

In addition, a coordinated assignment plan is required in order to guarantee that every task will be performed only once. This is to prevent, for instance, that the same task on the same target is carried over by two different vehicles or that conflict in the plan occurs such that a given task is never performed. A different type of uncertainty is considered in the form of wind disturbance. The main implications of nonzero wind are the possible disruption of the sequential tasks when they are performed.

Consider a group of agents in Eq. (1) and a set of targets $T = \{1, 2, \ldots, N_t\}$. Let N_m be the number of tasks to be performed at each target. In this paper $N_m = 3$, representing the tasks classify, attack, and verify. Let us associate each task with an integer value m as follows: classify $\{m = 1\}$, attack $\{m = 2\}$, verify $\{m = 3\}$. Let $N_s = N_t N_m$ be the number of single assignments and $S = \{1, 2, \ldots, N_s\}$ represents the set of stages. At each stage $k \in S$ one and only one task is to be assigned. The stages do not represent real-time intervals but only iterations used in the algorithm for the assignment of tasks in such a way that at the end of the N_s iteration (stage) all tasks have been assigned to the group of agents. Define

$$g_{i,j,k} \in \{0, 1\} \tag{7}$$

as a decision variable that is equal to 1 if UAV $i \in U$ performs a task on target $j \in T$ at stage $k \in S$ and it is 0 otherwise. Let $\bar{g}_k = [i, j]$ such that $g_{i,j,k} = 1$. The list

$$G_k = \{\bar{g}_1; \bar{g}_2, \ldots, \bar{g}_k\} \tag{8}$$

represents the set of assignments up to, and including, stage k. Let $d_{i,j,k}^{G_k}$ be the nominal distance to be traveled by UAV i to perform a task on target j at stage k. At the time of execution of the task assignment algorithm, for the first task assigned to agent i, $d_{i,j,k}^{G_k}$ represents the distance from the current position to the specified coordinates to perform that particular task. For subsequent tasks scheduled during the same execution of the plan, $d_{i,j,k}^{G_k}$ represents the distance from the coordinates where the previous task is to be performed to the new coordinates corresponding to the following assigned task. These distances represent the optimal Dubins trajectories between a pair of points. Optimal Dubins trajectory means that the distance traveled by the UAV between two points is minimized subject to the maximum turning rate of the vehicle and to the departure and arrival heading constraints. The mathematical formulation of the cooperative multiple TAP can be expressed as follows:

$$\min\left(J = \sum_{i=1}^{N_u}\sum_{j=1}^{N_t}\sum_{k=1}^{N_s} d_{i,j,k}^{G_k} g_{i,j,k}\right) \tag{9}$$

subject to

$$\sum_{i=1}^{N_u}\sum_{j=1}^{N_t} g_{i,j,k} = 1, \qquad k \in S \tag{10}$$

$$\sum_{i=1}^{N_u}\sum_{k=1}^{N_s} g_{i,j,k} = N_m, \qquad j \in T \tag{11}$$

The constraints guarantee that exactly one task is assigned at any given stage [Eq. (10)] and that on each target exactly N_m tasks are performed [Eq. (11)]. There is an additional constraint that significantly increases the complexity of this optimization problem. This constraint imposes a specific order on the tasks performed on the same target; that is, for a given target $j \in T$ the different tasks $m(j)$ should be assigned in stages that satisfy the following: if $m(j) = 1$ is assigned at stage $k \in S$, $m(j) = 2$ is assigned at stage $k' \in S$, and $m(j) = 3$ is assigned at stage $k'' \in S$, then $k < k' < k''$. This constraint means that classification of a target should be performed before attacking the same target and attacking should be performed before verification takes place. This type of optimization problem was thoroughly discussed in [17,18].

III. Decentralized Estimation

This section represents an extension to the solution in [19]. Information about current positions of other vehicles is needed by each UAV in order to design the best strategy, the one that minimizes [Eq. (6)]. Dynamical models of all UAVs are implemented by each agent $i \in U$, including a model of its own dynamics. Models of other vehicles provide estimates of their current positions to agent i. The local model of agent i is used by the same agent to estimate the error between its current position measurement and the position of agent i estimated by the remaining agents. In other words, the purpose of the local model is to determine the next broadcasting time instant based on an event-triggered strategy. The model is updated when the local agent i broadcasts a measurement update. The events are generated only when the position error is larger than a predetermined constant threshold. Additionally and in contrast to [19], every agent uses an estimate of the communication delay for each measurement update that it receives in order to obtain a better estimate of the current position of the agent which transmitted that measurement.

To run a decentralized task allocation algorithm based on distance-to-task costs, each agent needs an estimate of current positions of teammates. It is also desired to avoid continuous or frequent interagent communication. At a reasonable increased computation cost, each agent will implement models of all vehicle dynamics, including itself. Models of teammates provide the estimated positions needed to solve the TAP problem and the model of the local agent provides a way for an efficient implementation of a broadcasting strategy based on events [21–24]. Consider the vehicle dynamics (2–4) in the following form:

$$\dot{\boldsymbol{x}}_i = A\boldsymbol{x}_i + B(\boldsymbol{x}_i)u_i + F\omega_i \tag{12}$$

where $\boldsymbol{x}_i = [\,x_i \quad y_i \quad v_{xi} \quad v_{yi}\,]^T$, ω_i represents exogenous disturbances mainly associated to winds, and

$$A = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \qquad B(\boldsymbol{x}_i) = \begin{bmatrix} 0 \\ 0 \\ -\Omega_{\max} v_{yi} \\ \Omega_{\max} v_{xi} \end{bmatrix}, \qquad F = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \tag{13}$$

A discretized representation of each vehicle dynamics is used for estimation purposes in which the discretization period is equivalent to the measurement period and denoted by T_1,

$$\boldsymbol{x}_i[n+1] = \Phi\boldsymbol{x}_i[n] + B_i[n]u_i[n] + \Gamma\omega_i[n] \quad \boldsymbol{z}_i[n] = [x_i[n]\, y_i[n]]^T + \boldsymbol{v}_i[n] = H\boldsymbol{x}_i[n] + \boldsymbol{v}_i[n] \tag{14}$$

where

$$\Phi = \begin{bmatrix} 1 & 0 & T_1 & 0 \\ 0 & 1 & 0 & T_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \qquad B_i[n] = \begin{bmatrix} -\frac{1}{2}\Omega_{\max,i} v_{yi}[n] T_1^2 \\ \frac{1}{2}\Omega_{\max,i} v_{xi}[n] T_1^2 \\ -\Omega_{\max,i} v_{yi}[n] T_1 \\ \Omega_{\max,i} v_{xi}[n] T_1 \end{bmatrix}, \qquad \Gamma = \begin{bmatrix} T_1 & 0 \\ 0 & T_1 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}, \qquad H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \tag{15}$$

and $\boldsymbol{v}_i[n] \in \mathbb{R}^2$ is a zero mean white measurement noise with positive definite covariance matrix R_i. Each agent $i \in U$ implements a Kalman filter in order to estimate $\boldsymbol{x}_i$. Let us denote this estimated variable as $\hat{\boldsymbol{x}}_i$. The Kalman filter equations are given by

$$\begin{aligned} P_i^-[n] &= \Phi P_i^+[n-1]\Phi^T + Q_{aw} \\ K_i[n] &= P_i^-[n]H^T(HP_i^-[n]H^T + R_i)^{-1} \\ \hat{\boldsymbol{x}}_i^-[n] &= \Phi\hat{\boldsymbol{x}}_i^+[n-1] + \hat{B}_i[n-1]u_i[n-1] \\ \hat{\boldsymbol{x}}_i^+[n] &= \hat{\boldsymbol{x}}_i^-[n] + K_i[n](z_i[n] - H\hat{\boldsymbol{x}}_i^-[n]) \\ P_i^+[n] &= (I - K_i[n]H)P_i^-[n](I - K_i[n]H)^T + K_i[n]R_iK_i[n]^T \end{aligned} \tag{16}$$

where $\hat{\boldsymbol{x}}_i^-[n]$ represents the a priori state estimate and $\hat{\boldsymbol{x}}_i^+[n]$ the a posteriori state estimate of $\boldsymbol{x}_i[n]$. $P_i^-[n]$ and $P_i^+[n]$ denote the covariance of the estimation error of $\hat{\boldsymbol{x}}_i^-[n]$ and $\hat{\boldsymbol{x}}_i^+[n]$, respectively. The time-varying matrix $\hat{B}_i[n]$ is given by

$$\hat{B}_i[n-1] = \begin{bmatrix} -\frac{1}{2}\Omega_{\max,i}\hat{v}_{yi}[n-1]T_1^2 \\ \frac{1}{2}\Omega_{\max,i}\hat{v}_{xi}[n-1]T_1^2 \\ -\Omega_{\max,i}\hat{v}_{yi}[n-1]T_1 \\ \Omega_{\max,i}\hat{v}_{xi}[n-1]T_1 \end{bmatrix} \tag{17}$$

where $\hat{v}_{xi}$ and $\hat{v}_{yi}$ are contained in $\hat{\boldsymbol{x}}_i$ and denote the estimate of v_{xi} and v_{yi}, respectively.

Remark. The input disturbance ω is not necessarily Gaussian or zero-mean. We use the first and second moments to model the input disturbance ω. A small artificial noise is used for the implementation of the filters with positive definite covariance matrix Q_{aw}. This means that even though a Gaussian input noise does not exist we still use a positive definite input covariance. The real ω represents wind disturbances and its effects can be seen in the time to reach a destination of each agent according to the direction of travel. It is assumed in this paper that each UAV can track a predetermined flyable trajectory with certain accuracy in the presence of winds. However, the time to fly a given trajectory will be different than the nominal predicted time (under zero wind). Differences between the real and nominal flight times can cause important disruptions in the mission because tasks on the same target need to be performed in strict order. Because it is difficult to model the effect of winds with time-varying magnitudes and directions, we implement a re-planning scheme in Sec. V. The agents execute partial or total assignments in the adaptive scheme only when the time differences may disrupt the sequential order of tasks performed on the same target.

Each agent $i \in U$ also implements models of all agents in the network given by

$$\boldsymbol{x}^{i,l}[n+1] = \Phi\boldsymbol{x}^{i,l}[n] + B^{i,l}[n]u^{i,l}[n] \tag{18}$$

where $\boldsymbol{x}^{i,l}$ represents a model of the state of agent l implemented by agent i, and $u^{i,l}$ is the control input of agent l known to agent i. Note that the model equations use $B^{i,l}$, which is constructed using the corresponding elements of the state $\boldsymbol{x}^{i,l}$. Define the local state error

$$e_i[n] = \hat{\boldsymbol{x}}_i[n] - \boldsymbol{x}^{i,i}[n] \tag{19}$$

An error event is triggered when the elements of the error corresponding to the positions are large compared with a specified threshold:

$$e_i[n]^T E e_i[n] > \epsilon_1 \tag{20}$$

where $E = \text{diag}\{1, 1, 0, 0\}$ and $\epsilon_1 > 0$ is the chosen threshold. When Eq. (20) holds at some time n_i, agent i broadcasts to its neighbors a time-stamped measurement vector containing its current measurement $\hat{\boldsymbol{x}}_i[n_i]$ and its current control action $u_i[n_i]$. It also resets its own model; that is, $\boldsymbol{x}^{i,i}[n_i] = \hat{\boldsymbol{x}}_i[n_i]$, which makes the local state error equal to zero, $e_i[n_i] = 0$. Neighbors of agent i, that is, agents that can receive measurements from agent i directly, receive the transmitted information after some delay, that is, at some time $n_i + \tau_{il}[n_i]$, where $\tau_{il}[n_i]$ represents a time-varying delay associated with the transmitted information at time n_i from agent i to agent l. The delayed information is used to obtain, instantaneously, an estimate of the current state based on knowledge of the delay by agent l, $\hat{\tau}_{il}[n_i]$, which is different in general from the real delay $\tau_{il}[n_i]$ because exact clock synchronization between agents is not assumed. Agent l updates its model state of agent i as follows:

$$\boldsymbol{x}^{l,i}[n_i + \tau_{il}[n_i]] = \bar{\boldsymbol{x}}^{l,i}[n_i + \hat{\tau}_{il}[n_i]] \tag{21}$$

where $\bar{\boldsymbol{x}}^{l,i}[n_i + \hat{\tau}_{il}[n_i]]$ is found by time propagating the received values $\hat{\boldsymbol{x}}_i[n_i]$ and $u_i[n_i]$ according to the estimated delay $\hat{\tau}_{il}[n_i]$ and the nominal vehicle dynamics; $\bar{\boldsymbol{x}}^{l,i}[n_i + \hat{\tau}_{il}[n_i]]$ can be found instantaneously once the update is received. Agent l retransmits this information so that other teammates may also update their models of agent i. If an update needs to be retransmitted in order to reach every node, the delay is greater, in general, because one-link delays are added for every retransmission. In general, these agents will receive information with additional delay, but this is not a constraint because they can estimate the total delay from the source agent and obtain current estimates of that agent's state.

Remark. By retransmitting received measurements it is not necessary that all agents can communicate with each other (complete communication graph) but only that the agents form an undirected connected graph at every time n.

The filtering method used in this paper represents a simple time propagation approach in the absence of new measurements. Such methods have been employed in event-triggered control and estimation such as in [25]. Filtering techniques for delayed information have been formally studied in different papers. For example, the paper [26] addresses the problem of fusing displacement measurements with position estimates for robot localization. A "stochastic cloning" approach is used to analyze the interdependency of these two variables, which are updated at different rates and are given at different time instants. In [27], delayed information is used to reset the accumulated navigation error of unmanned underwater vehicles by using scanned images and matching the estimated positions with respect to a previously visited site.

IV. Decentralized Task Allocation

In this section we consider the initial execution of the TAP algorithm. The assignment at each stage is found based on a cost matrix that evaluates the expected cost of each vehicle to perform each one of the current tasks (at the given stage). Because of communication delays and other uncertainties, agents may arrive at different assignment plans. The main extension to the work presented in [19] that is proposed in this paper is an algorithm for estimation and resolution of possible conflicts. The idea is based on generating new events (to broadcast measurements) when entries of the cost matrix are close to the minimum at any given stage of the optimization problem. To resolve an estimated conflict, agents bid on their best task at that particular stage. Their bids represent their cumulative cost on the task they are bidding on. Because these are real numbers representing the expected distance to travel to perform previous tasks (if already scheduled) and the conflicted task, the probability of agents bidding exactly the same cost is very low. This approach results in a trade-off between reducing interagent communication and achieving a conflict-free assignment plan. This trade-off is reflected on the selection of the new threshold applied to the cost matrix entries. In the case that two or more agents bid using exactly the same cumulative cost, several choices can be implemented in order to break ties. A simple solution is to use agents' ID; for instance, if a tie exists in a conflict resolution stage, the agent with lowest ID wins the assignment.

Each agent is in charge of assigning all existing tasks to the fleet. The approach followed in this section to solve Eqs. (9–11) produces piece-wise optimal trajectories with respect to the Dubins dynamics (2–4). At each stage $k \in S$ a matrix of costs is generated by each agent containing all cumulative distances $D^{G_{ki}}_{i,j,k}$ up to and including stage k. The minimum entry in the matrix at stage k is selected as the assigned task and the costs are updated according to the selection in order to obtain new costs for the next stage. An agent $i \in U$ computes these costs according to

$$D^{G_{ki}}_{i,j,k} = f(\hat{\boldsymbol{x}}_i[n], G_{ki}) \tag{22}$$

for itself and

$$D^{G_{ki}}_{l,j,k} = f(\boldsymbol{x}^{i,l}[n], G_{ki}) \tag{23}$$

for the rest of the agents. These costs are computed using the path optimization function of the MULTIUAV2 simulation [28]. The current UAV position and heading is used to predict an optimal trajectory to every target. The obtained distance is used to fill all columns corresponding to the particular UAV. Corresponding waypoints for each optimal trajectory are also generated and those corresponding to winning tasks by the local UAV are stored for actual guidance when the tasks are actually performed. The classification task requires a predetermined heading. The attack and verification tasks do not have this constraint.

The dependence of the costs on the local variables $\hat{\boldsymbol{x}}_i[n], \boldsymbol{x}^{i,l}[n]$ is made explicit in both Eqs. (22) and (23). This means that the cost matrices at the same stage for some or all agents could be different resulting in a different list of assigned tasks G_{ki}; that is, conflicts may be introduced during the assignment process. The dependence of the costs on the local assignment list G_{ki} is also made explicit in Eqs. (22) and (23).

One solution is to communicate at every stage and arrive at the same assignments. This is undesired, especially for large number of UAVs and targets, due to the limitations and constraints of the communication channel. The approach in this paper is to predict conflicts and only establish communication to resolve potential conflicts. The approach to follow in this section is to implement a second threshold that will be used during the assignment stages and will be applied to the entries of the cost matrix. This threshold is used during the assignment algorithm. The previous

threshold, which is given by Eq. (20), is used outside the assignment algorithm and it is checked consistently to keep the estimate of teammate locations viable. Good estimates will yield less conflicts here in the assignment problem.

Every agent $i \in U$ first chooses the minimum entry of its cost matrix at every stage $k \in S$ defined by $\min 1_{i,j}(D^{G_{ki}}_{i,j,k})$ for $i \in U$, $j \in T$. Before proceeding to the next stage, the agent will search for the second minimum entry, which is defined by $\min 2_{i,j}(D^{G_{ki}}_{i,j,k})$ for $i \in U$, $j \in T$, and compare as follows:

$$\min 2_{i,j}(D^{G_{ki}}_{i,j,k}) - \min 1_{i,j}(D^{G_{ki}}_{i,j,k}) < \epsilon_2 \tag{24}$$

where $\epsilon_2 > 0$ provides a threshold for estimating potential conflicts. A smaller threshold reduces communication because fewer conflicts will be flagged but the probability of missing real conflicts increases. On the other hand, a large threshold will reduce the probability of missing real conflicts but communication will be increased because false alarms, that is, stages where no conflicts are present, will be flagged more frequently. An optimal choice of ϵ_2 is difficult to obtain due to its dependence on the sizes of measurement noise, communication delays, and wind disturbance as well as on the accuracy of clock synchronization and the choice of ϵ_1.

When Eq. (24) holds a potential conflict is flagged. The local agent broadcasts a bid vector along with the measurement vector. The bid vector is given by

$$\Delta_i = [\, i \quad j \quad m \quad k \quad \beta_i \,]^T \tag{25}$$

where $i, j, m, k \in \mathbb{Z}_+$ represent the vehicle ID, the target on which vehicle i is placing its bid, the task it is scheduled on that target, and the stage when it found a potential conflict, respectively. $\beta_i = \min_j(D^{G_{ki}}_{i,j,k})$ is the local bid for that particular conflict. This means that the agent finds its best assignment for the stage, regardless if the agent is involved or not in the conflict it detects. Then the agent waits to collect bids from all other agents. If one or more agent did not detect a particular conflict, they will respond when they receive a bid vector by generating and transmitting a bid vector corresponding to the same stage. The stage when a conflict was flagged is known to the receiving agent because this information is included in the bid vector. After each agent collects all bid vectors, it finds immediately the winning bid and the associated winning assignment (embedded in the bid vector). The agents then proceed to finish the assignment of tasks.

Example 1. The proposed algorithm has been tested in simulations and it produces conflict-free assignments under some simplifying assumptions and for sufficiently large cost matrix thresholds. This example represents a single run of the proposed method (for each run the positions of the targets and the initial position of the UAVs change randomly). The target coordinates in this run were $j_1 = (9876, 8043)$, $j_2 = (12,087, 19,879)$, $j_3 = (16,901, 10,034)$, $j_4 = (21,769, 17,034)$. The nominal speed is $v = 300$ ft/s, the nominal maximum turning radius is $\Omega_{\max} = 0.15$ rad/s, and $T_1 = 0.02$ s. These nominal quantities are the same for all vehicles $i \in U$.

In this example a total of three conflicts were estimated. The first conflict occurred at stage $k = 2$ of the assignment process (at stage $k = 1$ all agents assigned agent 2 to perform task 1 on target 2). Table 1 provides the conflict details for each agent. This can be considered as a "false conflict" (or false alarm) because all agents could have assigned agent 1 to perform task 2 on target 2. However, the cost that each agent obtained for agent 1 to perform task 2 on target 2 was very close to the one corresponding to agent 2 to perform task 2 on target 2 and that is the reason that a conflict event is triggered. The threshold used in this example to trigger a conflict was $\epsilon_2 = 200$. The costs estimated for every agent involved in the detected conflict are also shown in Table 1.

As explained before, each agent finds its own best cost and transmits this information, which is the bid vector (25). The bid vectors corresponding to this conflict are

$$\Delta_1 = [\,1 \quad 2 \quad 2 \quad 2 \quad 11,802.84\,]^T \quad \Delta_2 = [\,2 \quad 2 \quad 2 \quad 2 \quad 11,915.31\,]^T \quad \Delta_3 = [\,3 \quad 1 \quad 1 \quad 2 \quad 12,132.27\,]^T \tag{26}$$

Agents 1 and 2 are obviously bidding on target 2, while agent 3 is bidding on its preferred target at the current stage, which, for this example, is target 1 (to perform task 1). Once all agents receive all bid vectors, they can all immediately agree on the assignment agent 1 to perform task 2 on target 2 because $\beta_1 = 11,802.84$ is the winning bid.

The second conflict is more interesting. This conflict was estimated at stage $k = 5$. Table 2 provides the conflict details for each agent. One can clearly see that there is a conflict because agent 1 assigns itself to perform task 1 on target 4. However, agents 2 and 3 assign the same task on the same target to agent 2. As this conflict is detected by all agents then all of them transmit bid vectors. The bid vectors corresponding to this conflict are

$$\Delta_1 = [\,1 \quad 4 \quad 1 \quad 5 \quad 19,958.85\,]^T \quad \Delta_2 = [\,2 \quad 4 \quad 1 \quad 5 \quad 19,894.02\,]^T \quad \Delta_3 = [\,3 \quad 2 \quad 3 \quad 5 \quad 25,112.4\,]^T \tag{27}$$

After all agents share this information, all of them agree on the assignment agent 2 to perform task 1 on target 4 because $\beta_2 = 19,894.02$ is the winning bid.

The last conflict occurred at stage $k = 6$. Table 3 provides the conflict details for each agent. This is another false conflict that was estimated only by agents 2 and 3. Even though agent 1 did not estimate the conflict, it responds the call for resolving the conflict and finds its best bid. The bid vectors corresponding to this conflict are

Table 1 Conflict estimated at stage $k = 2$

	Conflicts			
Conflict detected by agent	UAV	Target	Task	Cost (ft)
1	1	2	2	11,802.84
	2	2	2	11,893.55
2	1	2	2	11,817.14
	2	2	2	11,915.31
3	1	2	2	11,825.06
	2	2	2	11,902.67

Every pair of rows represents the conflict details detected by each agent.

Table 2 Conflict estimated at stage $k = 5$

	Conflicts			
Conflict detected by agent	UAV	Target	Task	Cost (ft)
1	1	4	1	19,958.85
	2	4	1	19,962.61
2	2	4	1	19,894.02
	1	4	1	19,943.37
3	2	4	1	19,901.38
	1	4	1	19,962.78

Every pair of rows represents the conflict details detected by each agent.

Table 3 Conflict estimated at stage $k = 6$

	Conflicts			
Conflict detected by agent	UAV	Target	Task	Cost (ft)
1	No conflict detected			
2	1	4	2	21,375.03
	2	4	2	21,570.51
3	1	4	2	21,364.98
	2	4	2	21,522.71

Every pair of rows represents the conflict details detected by each agent.

$$\Delta_1 = [1 \quad 4 \quad 2 \quad 6 \quad 21,284.67]^T \quad \Delta_2 = [2 \quad 4 \quad 2 \quad 6 \quad 21,570.51]^T \quad \Delta_3 = [3 \quad 2 \quad 3 \quad 6 \quad 24,818.25]^T \tag{28}$$

After all agents share this information, all of them agree on the assignment agent 1 to perform task 2 on target 4 because $\beta_1 = 21,284.67$ is the winning bid.

Figure 1 shows the complete trajectories of the UAVs. The agents resolve the conflicts based on the corresponding bids and the assignment continues from that point forward using the recent position updates. A maximum delay of 1 s is implemented. A double-maximum delay (2 s) can be incurred when not all agents estimate the same conflicts; that is, agents that are unaware of a particular conflict respond after 1 s and all agents collect all bids after 2 s. The final assignment plan is shown in Table 4.

V. Adaptive Re-Planning

One important aspect of the algorithm presented in the previous section is that the mission can be planned in advance under nominal conditions. However, the presence of winds will disrupt the sequence of scheduled tasks in the sense that UAVs may arrive significantly sooner or later than their estimated time of arrival (ETA) to the next task under nominal conditions. At each target there is a strict order on how the tasks have to be performed. Classification(1) first, attack(2) second, and verification(3) last. During the execution of the mission the effects of wind disturbances may cause the agents to perform these three tasks on a given target in an order different to 1-2-3. For instance, consider the time instants when the UAVs provide service to the different targets in Example 1 under zero wind (nominal conditions) and under nonzero wind. This is shown in Fig. 2. Under zero wind the targets receive the scheduled services in sequential manner: classification(1), attack(2), and verification(3). For example, the service times under nominal conditions for target 4 were approximately 75, 80, and 120 s. However, we can see that under some nonzero wind the attack task on target 4 was performed before classification. These actions occurred around the simulation interval 75 and 80 s. One way to avoid this situation is to estimate possible disruptions on the sequence of performed tasks on any given target and re-plan as necessary.

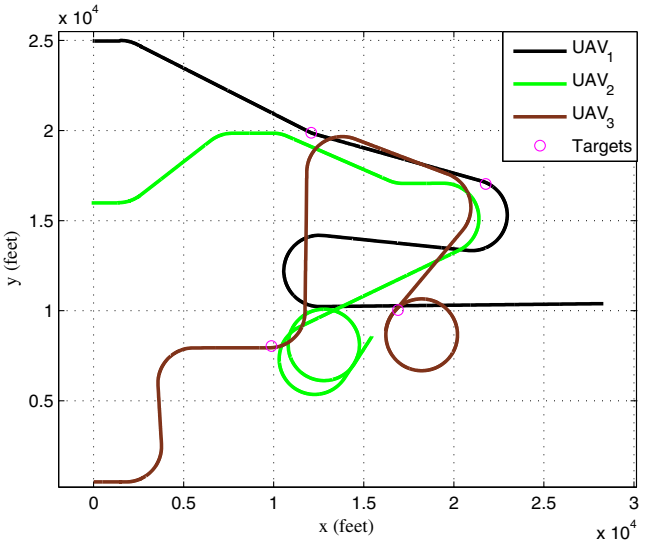


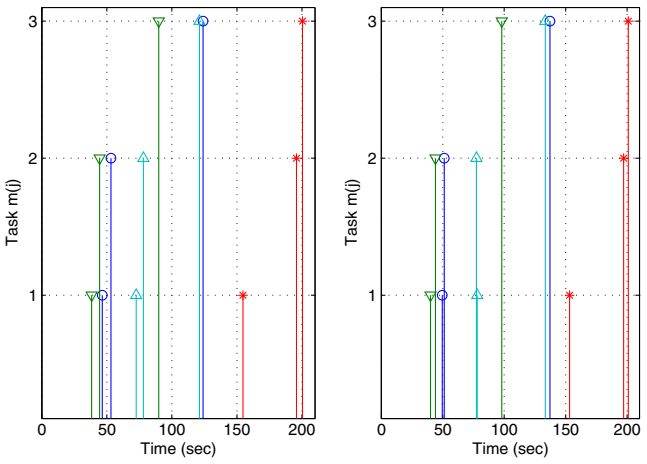
Fig. 1 Example UAV trajectories and target positions.

Table 4 List of assignments in Example 1

UAV	Target	Task
2	2	1
1	2	2
3	1	1
3	1	2
2	4	1
1	4	2
3	2	3
3	4	3
2	1	3
1	3	1
3	3	2
2	3	3

The extension proposed in this section is for each agent to keep track of its real flight times and the difference with respect to ETAs. Because the ETAs for all agents can be computed by every agent, then it is only necessary for each agent to transmit its time difference every time that a measurement update is triggered. An additional function has been added to the TAP algorithm in order to decide if the mission has to be re-planned at some particular time because of possible disorder on coupled tasks, that is, those tasks performed on the same target. Every agent computes its real flight time and estimates the real flight times for every other agent based on the last received time difference from each agent. Let $t_n(i)$ be the nominal flight time of agent i and $t_r(i)$ be the current real flight time. Define the time difference $t_d(i) = t_r(i) - t_n(i)$. These variables are used by each agent to predict possible plan disruptions according to Table 5, where $m_i(j)$ is used to denote the task $\{1,2,3\}$ that agent i is scheduled to perform on target j. The expression $t_r(i, j, m)$ represents the predicted real flight time, based on current data, for agent i to reach target j in order to perform task m. The decision variable $R_p \in \{0, 1\}$ is equal to one when the local agent decides that a re-planning is necessary and 0 if the current plan will not be disrupted by wind disturbances. If some agent decides that a re-planning is necessary, then it transmits a measurement update along with an order of re-planning that it will be followed by a new assignment process from that point forward by all agents. The threshold $\epsilon_3 > 0$ provides a measure of a tolerance when comparing flight times. This decision process only considers the next task to do. In detail, when a given agent finishes its current task, it starts traveling toward the next target for which it is scheduled to perform some task. Only at this time instant (right after finishing the current task) the agent compares the time flights t_r following the rules in Table 5 and considering only its next task. This process will be repeated by every agent after finishing every one of the assigned tasks. Note that the agents do not attempt to predict disruptions beyond the next task because wind effects on the plan are generally more unpredictable as one considers a longer time horizon.

Example 2 (Monte Carlo Simulations). We have chosen two sets of thresholds and run Monte Carlo simulations (100 runs for each set). For each run the target positions and the initial conditions of the UAVs are chosen randomly. The wind values are random for each run as well with bounded magnitude less or equal than 10 ft/s. The nominal speed and maximum turning radius are as given in Example 1. Note that the real speed and turning radius may deviate from the nominal ones as a consequence of wind disturbances. The measurement noise is a zero-mean Gaussian noise with $R = 100I$. Relevant data from these simulations are shown in Table 6. The first set of parameters provides a great reduction on the total number of transmissions (updates) by all agents but the performance is poor in terms of detecting conflicts. All types of conflicts include those generated at the initial planning and every re-planning due to wind. Undetected conflicts include those conflicts that were not detected even if they were never executed because in a re-planning phase they were corrected. The second set of parameters provides a much better performance in terms of detecting true conflicts and greatly reducing the number of undetected conflicts. This behavior is obtained at the cost of a substantial increase of false conflicts; that is, one or more agents trigger a bidding process for stages in which all agents assigned the same task. This selection of thresholds translates into an increased number of updates. The number of estimated disordered tasks also increased but the number of executed disordered tasks did not decrease. Further analysis needs to be performed in terms of selection of threshold ϵ_3 in order to reduce the number of tasks that are performed in incorrect order. A different approach to deal with this type of uncertainties that can be considered is to analyze the robustness of solutions to uncertainty in the path lengths and target locations similar to the work in [29].



a) Zero wind **b) Non−zero wind**

Fig. 2 Service times in Example 1. Target 1 (○), target 2 (▽), target 3(⋆), and target 4 (△). Tasks: classify: $m(j) = 1$; attack: $m(j) = 2$; verify: $m(j) = 3$. In (b) the wind causes an order disruption of tasks on target 4.

Table 5 Agent i decision rules for re-planning

t_d	Next task $m_i(j)$	Agent l doing task $m_l(j)$	Compare
	1	2	$t_r(i,j,m)+\epsilon_3 > t_r(l,j,m+1)$
+	2	3	Yes: $R_p=1$; no: $R_p=0$
	3		$R_p=0$
	1		$R_p=0$
−	2	1	$t_r(i,j,m)-\epsilon_3 < t_r(l,j,m-1)$
	3	2	Yes: $R_p=1$; no: $R_p=0$

Table 6 Results of simulations in Example 2

Thresholds and observations	Set 1	Set 2
ϵ_1	1000	200
ϵ_2	100	1000
ϵ_3	1	5
Conflicts detected	0.35	1.29
False conflicts	0.74	6.77
Conflicts undetected	1.04	0.23
Mission time	300.16	317.25
Estimated disordered tasks	0.73	1.78
Disordered tasks executed	0.25	0.27
Total number of updates	82.02	180.8

VI. Planning with Nonzero Steady Wind

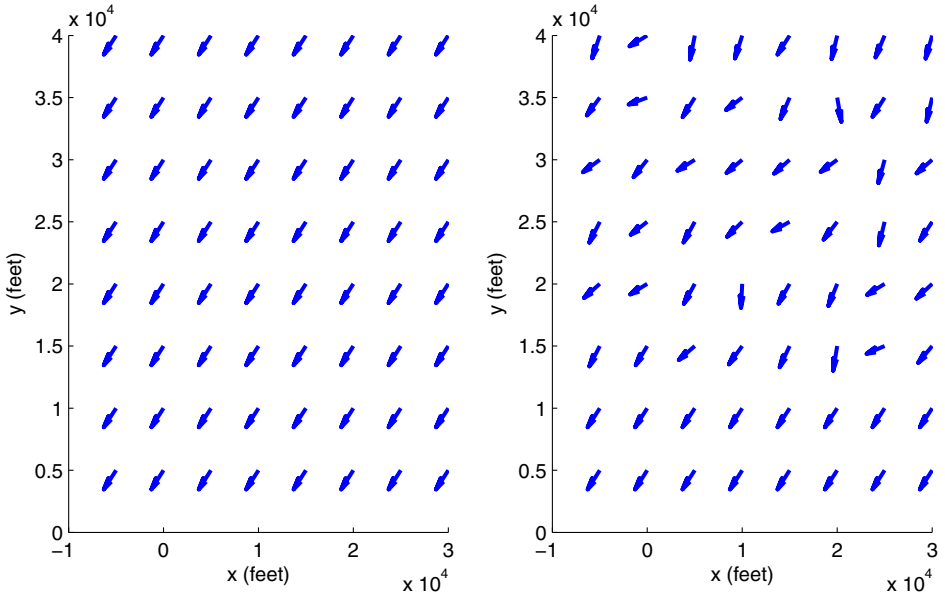
Previously, the assignment of tasks was planned assuming zero wind conditions and a re-planning order was issued when any agent detected a possible disruption of the tasks performed on any given target. The zero wind assumption simplifies the assignment process and possible bidding situations by using the cumulative distance values $D^{G_k}_{i,j,k}$ to determine the best assignments at every stage of the assignment process. However, re-planning requests will arise in general, and a re-planning process increases communication between UAVs.

In this section we consider a decentralized task assignment method that takes into account a nominal nonzero wind value (current at the time of making assignments) in order to reduce the instances that a re-planning request needs to be issued. This new way of determining the assignments does not use the distances $D^{G_k}_{i,j,k}$ but it tries to predict the time it takes every vehicle to complete possible assignments in the presence of a nominal nonzero wind value. In this case we use the ETAs of each UAV conditioned on the nominal nonzero wind value to find the best list of assignments. The real wind conditions during the mission will change around the steady wind values. Figure 3a shows an example of a steady wind vector field as it is assumed by the UAVs at the beginning of the mission when the assignments are planned and based on the wind measurements by the UAVs at that particular time. However, the real wind vector field is allowed to vary spatially and temporally; an example of such changes on the assumed wind vector field is shown in Fig. 3b.

The re-planning function described in Sec. V is still used to detect possible disruptions on the sequence of scheduled tasks. However, we expect that fewer disruptions on the execution of tasks will appear when the real wind values remain close to the assumed steady wind value ω_0 and using ω_0 in the assignment process, compared with the case when the assignments are made assuming zero wind.

The effects of wind on the planned trajectories can be described by

$$\dot{x}_i = v_{gx_i} = v \cdot \cos\theta_i + \omega_x \tag{29}$$



a) Nominal windfield **b) Real windfield**

Fig. 3 Nominal and real wind vector fields over an area of interest.

$$\dot{y}_i = v_{gy_i} = v \cdot \sin\theta_i + \omega_y \tag{30}$$

where ω_x represents the wind component in the x coordinate and ω_y represents the wind component in the y coordinate. To follow the planned trajectories and to reach the generated waypoints in the presence of winds, one can use a controller for path following in the presence of winds as described in [30,31]. A key characteristic of these type of controllers is that UAVs are able to follow the specified waypoints but the time to arrive at each waypoint under nonzero wind conditions will be different from flying with zero wind.

To incorporate the nonzero steady wind feature into the task assignment procedure, we change the type of costs used to generate the cost matrices at every stage. Instead of generating the cost matrices using cumulative distances, we now use cumulative estimated times of arrival. These estimated times are obtained in a way that reflect the effect of a nonzero steady wind on the planned trajectories. Because waypoints associated to all possible trajectories are also obtained when computing cumulative distances in Sec. IV, then it is possible to readjust the corresponding ETAs. This readjustment is done by predicting the steady wind effect on the planned trajectory according to the direction of travel at every segment of the planned trajectory. For instance, when considering straight line segments of a planned trajectory, we obtain $v_{g_i}(\theta_i, \omega_0) = (v_{gx_i}^2 + v_{gy_i}^2)^{1/2}$, which is constant for the line segment because ω_0 is constant and θ_i is also constant when traveling over a straight line segment. Then, the speed $v_{g_i}(\theta_i, \omega_0)$ and the length of the segment are used to estimate the corresponding ETA. This is a computationally simple way of approximating the ETAs to each waypoint corresponding to straight line segments of a trajectory associated to Eqs. (29) and (30).

The same simple approach can also be used to approximate the ETAs when following circular arcs. In this case we evaluate the speeds $v_{g_i}(\theta_i, \omega_0)$ at small straight lines on the circular arc. Similarly, the time to travel over each portion of the circular arc is estimated based on the approximated speed at that arc interval and the length of the same arc interval. Because, in general, the real wind vector field will be different from the assumed steady wind, an exact value of ETAs is not necessary and the implementation of this simple method to approximate the ETAs provides a significant improvement in reduction of re-planning requests.

Note that the method used in this section for planning trajectories is a simple approach that is not an optimal path planning algorithm for nonzero winds. This method consists on obtaining optimal, under zero-wind, trajectories and adjust the ETAs according to some assumed steady wind. The real wind can be different from the assumed one as it is commonly the case in real applications. Thus, we focus on the design of robust controllers that can deal with wind uncertainties.

The matrix of costs that is generated by each agent at each stage $k \in S$ now contains all cumulative ETAs (conditioned on the assumed steady wind) up to and including stage k. The cumulative ETAs are represented by $\mathcal{T}_{i,j,k}^{G_{ki}}$; that is, $\mathcal{T}_{i,j,k}^{G_{ki}}$ is the cumulative ETA of UAV i to perform a task on target j at stage k and according to the local assignment list G_{ki}. Similarly, the minimum entry in the matrix corresponding to stage k is selected as the assigned task and the costs are updated according to the selection in order to obtain new costs for the next stage. An agent $i \in U$ computes these costs according to

$$\mathcal{T}_{i,j,k}^{G_{ki}} = f_{\mathcal{T}}(\hat{\boldsymbol{x}}_i[n], G_{ki}) \tag{31}$$

for itself and

$$\mathcal{T}_{l,j,k}^{G_{ki}} = f_{\mathcal{T}}(\boldsymbol{x}^{i,l}[n], G_{ki}) \tag{32}$$

for the rest of the agents. The dependence of the costs on the local variables $\hat{\boldsymbol{x}}_i[n], \boldsymbol{x}^{i,l}[n]$ is also made explicit in both Eqs. (31) and (32) as it was in the case of the costs [Eq. (22) and (23)].

Conflicts during the assignment process can also occur and they are estimated in a similar way as it was done in Sec. IV. Every agent $i \in U$ first chooses the minimum entry of its cost matrix at every stage $k \in S$ defined by $\min\ 1_{i,j}(\mathcal{T}_{i,j,k}^{G_{ki}})$ for $i \in U, j \in T$. Before proceeding to the next stage, the agent will search for the second minimum entry, which is defined by $\min\ 2_{i,j}(\mathcal{T}_{i,j,k}^{G_{ki}})$ for $i \in U, j \in T$, and compare as follows:

$$\min\ 2_{i,j}(\mathcal{T}_{i,j,k}^{G_{ki}}) - \min\ 1_{i,j}(\mathcal{T}_{i,j,k}^{G_{ki}}) < \epsilon_{\mathcal{T}} \tag{33}$$

where $\epsilon_{\mathcal{T}} > 0$ provides a threshold for estimating potential conflicts. The threshold $\epsilon_{\mathcal{T}}$ plays the same role as ϵ_2 but it is chosen according to the cumulative ETAs instead of the cumulative distances.

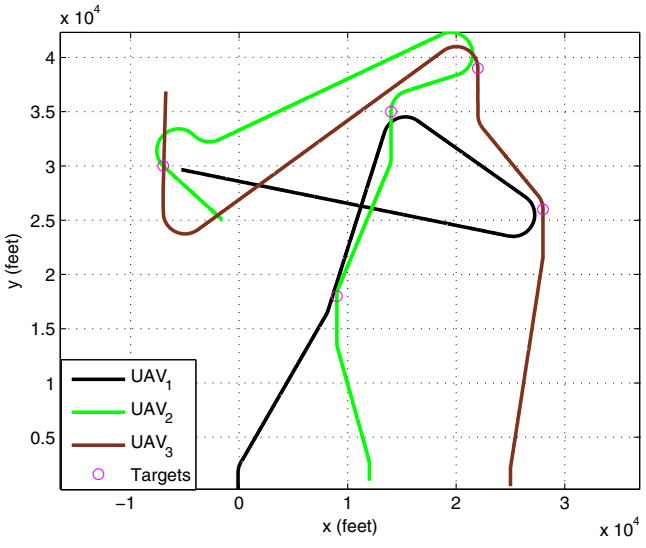


Fig. 4 Example UAV trajectories and target positions for nonzero nominal planned wind.

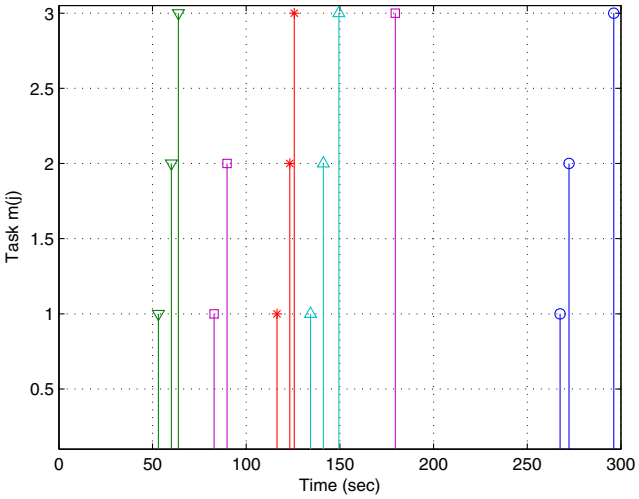


Fig. 5 Service times in Example 3. Target 1 (○), target 2 (∇), target 3 (⋆), target 4 (Δ), and target 5 (□). Tasks: classify: $m(j) = 1$; attack: $m(j) = 2$; verify: $m(j) = 3$.

Table 7 List of assignments in Example 3

UAV	Target	Task
2	2	1
2	2	2
1	2	3
3	5	1
3	5	2
2	3	1
2	3	2
1	3	3
3	4	1
3	4	2
2	4	3
1	5	3
3	1	1
2	1	2
1	1	3

When Eq. (33) holds, a potential conflict is flagged. The local agent broadcasts a bid vector along with the measurement vector. The bid vector is given by Eq. (25), in which $i, j, m, k \in \mathbb{Z}_+$ represent the vehicle ID, the target on which vehicle i is placing its bid, the task it is scheduled on that target, and the stage when it found a potential conflict, respectively. Now we have $\beta_i = \min_j(\mathcal{T}_{i,j,k}^{G_{ki}})$.

Example 3. Consider three UAVs approaching, from the south, an area of interest. The initial coordinates of each vehicle are $i_1 = (-100, 200)$, $i_2 = (12,000, 1000)$, $i_3 = (25,000, 500)$. In the area of interest there are five potential targets. The target coordinates in this execution are $j_1 = (-7000, 30,000)$, $j_2 = (9000, 18,000)$, $j_3 = (14,000, 35,000)$, $j_4 = (22,000, 39,000)$, $j_5 = (28,000, 26,000)$. The nominal speed is $v = 300$ ft/s, the nominal maximum turning radius is $\Omega_{\max} = 0.15$ rad/s, and $T_1 = 0.02$ s. These nominal quantities are the same for all vehicles $i \in U$. Consider a steady wind value (expressed in Cartesian coordinates) of $\omega_0 = [-10 \quad -12]$ ft/s while the real wind varies in magnitude and direction around ω_0. The classification process now requires an approaching heading of -90 deg.

Figure 4 shows the complete trajectories of the UAVs for this example, in which the path following controller described in [31] was implemented in the simulation of the UAV trajectories. The assignment plan is shown in Table 7. The service times are shown in Fig. 5. Note that the sequence of tasks to be performed at each target was carried over in the same order as planned and in the presence of significant wind disturbance. The re-planning function of Sec. V was still used but the variables $t_d(i)$, for $i \in U$, always remained very close to zero. This means that the difference between the real flight times and the nominal flight times (that were predicted based on the assumed steady wind value) was close to zero, for every vehicle, and disruptions did not appear in the execution of tasks performed on each target.

VII. Conclusions

This work provides a combined estimation and conflict resolution method for unmanned aerial vehicle task allocation that considerably reduces interagent communication compared with similar work. It extends the results in [19] to add a conflict resolution phase and obtain a conflict-free assignment plan. Additionally, a three-event strategy has been implemented in order to transmit information and help in decision making. The first event determines asynchronous transmission of position measurements and current control input. The second event detects and solves potential conflicts in the assignment process. The last event mandates a new assignment process to be executed when wind disturbance may disrupt the sequential order of coupled tasks. A final function associated to this approach allows to consider the effects of wind disturbances during the planning process. Simulation examples were provided that show the robustness and performance of this scheme under communication delays, measurement noise, and wind disturbance.

Acknowledgment

This work has been supported in part by Air Force Office of Scientific Research (AFOSR) Laboratory Research Initiation Request (LRIR) No. 12RB07COR.

References

[1] Dionne, D., and Rabbath, C. A., "Multi-UAV Decentralized Task Allocation with Intermittent Communications: The DTC Algorithm," *American Control Conference*, IEEE Publ., Piscataway, NJ, 2007, pp. 5406–5411.

[2] Alighanbari, M., and How, J. P., "A Robust Approach to the UAV Task Assignment Problem," *International Journal of Robust and Nonlinear Control*, Vol. 12, No. 12, 2008, pp. 118–134.
doi:10.1002/(ISSN)1099-1239

[3] Sujit, P. B., and Beard, R., "Multiple MAV Task Allocation Using Distributed Auctions," *AIAA Guidance, Navigation, and Control Conference and Exhibit*, AIAA Paper 2007-6452, 2007.

[4] Sujit, P. B., George, J. M., and Beard, R., "Multiple UAV Task Allocation Using Particle Swarm Optimization," *AIAA Guidance, Navigation, and Control Conference and Exhibit*, AIAA Paper 2008-6837, 2008.

[5] Nigam, N., "Dynamic Replanning for Multi-UAV Persistent Surveillance," *AIAA Guidance, Navigation, and Control Conference*, AIAA Paper 2013-4887, 2013.

[6] Chandler, P. R., Rasmussen, S., and Pachter, M., "UAV Cooperative Path Planning," *AIAA Guidance, Navigation, and Control Conference*, AIAA Paper 2000-4370, 2000, pp. 1255–1265.

[7] Beard, R. W., McLain, T. W., Goodrich, M., and Anderson, E. P., "Coordinated Target Assignement and Intercept for Unmanned Air Vehicles," *IEEE Transactions on Robotics and Automation*, Vol. 12, No. 6, 2002, pp. 911–922.
doi:10.1109/TRA.2002.805653

[8] Choi, H. L., Brunet, L., and How, J. P., "Consensus-Based Decentralized Auctions for Robust Task Allocation," *IEEE Transactions on Robotics*, Vol. 25, No. 4, 2009, pp. 912–926.
doi:10.1109/TRO.2009.2022423

[9] Ponda, S. S., Redding, J., Choi, H. L., How, J. P., Vavrina, M., and Vian, J., "Decentralized Planning for Complex Missions with Dynamic Communication Constraints," *American Control Conference*, IEEE Publ., Piscataway, NJ, 2010, pp. 3998–4003.

[10] Johnson, L. B., Ponda, S. S., Choi, H. L., and How, J. P., "Asynchronous Decentralized Task Allocation for Dynamic Environments," *Proceedings of the AIAA Infotech@Aerospace Conference*, AIAA Paper 2011-1441, 2011.

[11] Johnson, L. B., Choi, H. L., and How, J. P., "Hybrid Information and Plan Consensus in Distributed Task Allocation," *AIAA Guidance, Navigation, and Control Conference*, AIAA Paper 2013-4888, 2013.

[12] Choi, H. L., Whitten, A. K., and How, J. P., "Decentralized Task Allocation for Heterogeneous Teams with Cooperation Constraints," *American Control Conference*, IEEE Publ., Piscataway, NJ, 2010, pp. 3057–3062.

[13] Whitten, A. K., Choi, H. L., Johnson, L. B., and How, J. P., "Decentralized Task Allocation with Coupled Constraints in Complex Missions," *American Control Conference*, IEEE Publ., Piscataway, NJ, 2011, pp. 1642–1649.

[14] Binetti, G., Naso, D., and Turchiano, B., "Decentralized Task Allocation for Surveillance Systems with Critical Tasks," *Robotics and Autonomous Systems*, Vol. 61, No. 12, 2013, pp. 1653–1664.
doi:10.1016/j.robot.2013.06.007

[15] Wellman, M. P., Walsh, W. E., Wurman, P. R., and MacKie-Mason, J. K., "Auction Protocols for Decentralized Scheduling," *Games and Economic Behavior*, Vol. 35, Nos. 1–2, 2001, pp. 271–303.
doi:10.1006/game.2000.0822

[16] Mitchell, J. W., Schumacher, C., Chandler, P. R., and Rasmussen, S. J., "Communication Delays in the Cooperative Control of Wide Area Search Munitions via Iterative Network Flow," DTIC Document, Tech. Rep., 2003.

[17] Schumacher, C., Chandler, P., and Rasmussen, S. J., "Task Allocation for Wide Area Search Munitions," *American Control Conference*, IEEE Publ., Piscataway, NJ, 2002, pp. 1917–1922.

[18] Schumacher, C., Chandler, P., Rasmussen, S. J., and Walker, D., "Task Allocation for Wide Area Search Munitions with Variable Path Length," *American Control Conference*, IEEE Publ., Piscataway, NJ, 2003, pp. 3472–3477.

[19] Shima, T., Rasmussen, S. J., and Chandler, P., "UAV Team Decision and Control Using Efficient Collaborative Estimation," *ASME Journal of Dynamic Systems, Measurement, and Control*, Vol. 129, No. 5, 2007, pp. 609–619.
doi:10.1115/1.2764504

[20] Garcia, E., and Casbeer, D. W., "UAV Cooperative Task Allocation with Communication Delays and Conflict Resolution," *AIAA Infotech@Aerospace Conference*, AIAA Paper 2013-4580, 2013.

[21] Dimarogonas, D. V., Frazzoli, E., and Johansson, K. H., "Distributed Event-Triggered Control for Multi-Agent Systems," *IEEE Transactions on Automatic Control*, Vol. 57, No. 5, 2012, pp. 1291–1297.
doi:10.1109/TAC.2011.2174666

[22] Garcia, E., and Antsaklis, P. J., "Decentralized Model-Based Event-Triggered Control of Networked Systems," *American Control Conference*, IEEE Publ., Piscataway, NJ, 2012, pp. 6485–6490.

[23] Garcia, E., and Antsaklis, P., "Model-Based Event-Triggered Control for Systems with Quantization and Time-Varying Network Delays," *IEEE Transactions on Automatic Control*, Vol. 58, No. 12, 2013, pp. 422–434.
doi:10.1109/TAC.2012.2211411

[24] Garcia, E., Cao, Y., Yu, H., Antsaklis, P. J., and Casbeer, D. W., "Decentralized Event-Triggered Cooperative Control with Limited Communication," *International Journal of Control*, Vol. 86, No. 9, 2013, pp. 1479–1488.
doi:10.1080/00207179.2013.787647

[25] Li, L., and Lemmon, M., "Event-Triggered Output Feedback Control of Finite Horizon Discrete-Time Multi-Dimensional Linear Processes," *49th IEEE Conference on Decision and Control (CDC)*, IEEE Publ., Piscataway, NJ, 2010, pp. 3221–3226.

[26] Roumeliotis, S. I., and Burdick, J. W., "Stochastic Cloning: A Generalized Framework for Processing Relative State Measurements," *IEEE International Conference on Robotics and Automation, 2002. Proceedings. ICRA '02*, Vol. 2, IEEE Publ., Piscataway, NJ, 2002, pp. 1788–1795.

[27] Eustice, R. M., Singh, H., and Leonard, J. J., "Exactly Sparse Delayed-State Filters for View-Based Slam," *IEEE Transactions on Robotics*, Vol. 22, No. 6, 2006, pp. 1100–1114.
doi:10.1109/TRO.2006.886264

[28] Rasmussen, S. J., Mitchell, J. W., Schulz, C., Schumacher, C. J., and Chandler, P., "A Multiple UAV Simulation for Researchers," *AIAA Modeling and Simulation Technologies Conference*, AIAA Paper 2003-5684, 2003.

[29] Niendorf, M., Kabamba, P. T., and Girard, A. R., "Stability of Solutions to Classes of Traveling Salesman Problems," *IEEE Transactions on Cybernetics* (in press).
doi:10.1109/TCYB.2015.2418737

[30] Nelson, D. R., Barber, D. B., McLain, T. W., and Beard, R. W., "Vector Field Path Following for Miniature Air Vehicles," *IEEE Transactions on Robotics*, Vol. 23, No. 3, 2007, pp. 519–529.
doi:10.1109/TRO.2007.898976

[31] Paley, D. A., and Peterson, C., "Stabilization of Collective Motion in a Time-Invariant Flowfield," *AIAA Journal of Guidance, Control, and Dynamics*, Vol. 32, No. 3, 2009, pp. 771–779.
doi:10.2514/1.40636

H. Balakrishnan
Associate Editor