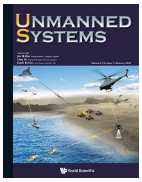
Unmanned Systems, Vol. 1, No. 1 (2013) 41–58

DOI: 10.1142/S2301385013500039

UNMANNED SYSTEMS
http://www.worldscientific.com/worldscinet/us



Dynamic Mission Planning for Communication Control in Multiple Unmanned Aircraft Teams

Andrew N. Kopeikin*,†, Sameera S. Ponda*,‡, Luke B. Johnson*,§, Jonathan P. How*,¶

Department of Aeronautics and Astronautics, MIT, Cambridge, MA

A multi-UAV system relies on communications to operate. Failure to communicate remotely sensed mission data to the base may render the system ineffective, and the inability to exchange command and control messages can lead to system failures. This paper describes a unique method to control network communications through distributed task allocation to engage under-utilized UAVs to serve as communication relays and to ensure that the network supports mission tasks. This work builds upon a distributed algorithm previously developed by the authors, CBBA with Relays, which uses task assignment information, including task location and proposed execution time, to predict the network topology and plan support using relays. By explicitly coupling task assignment and relay creation processes, the team is able to optimize the use of agents to address the needs of dynamic complex missions. In this work, the algorithm is extended to explicitly consider realistic network communication dynamics, including path loss, stochastic fading, and information routing. Simulation and flight test results validate the proposed approach, demonstrating that the algorithm ensures both data-rate and interconnectivity bit-error-rate requirements during task execution.

Keywords: Multi-agent control; multi-UAV team; distributed control; UAV communication network; communication relays; UAV flight test; network capacity.

1. Introduction

As unmanned systems technology progresses, teams of multiple unmanned air vehicles (UAVs) will play an increasingly important role in a broader range of applications. These "real-world" scenarios involve executing complex missions where the number, status, and types of tasks, as well as the environment vary dynamically. As such, a multi-UAV system must continuously assess its capabilities and properly allocate resources to overcome changes and maximize performance. A significant challenge in this process involves maintaining proper communications to execute the mission. For multi-UAV teams to cooperate, the communication network must exchange command and control messages, and when necessary, remotely sensed data (e.g., live video). Failures in communication can render the multi-UAV system ineffective or can lead to dangerous system failures and unintended consequences.[1] This is particularly true in distributed systems, where these messages enable vehicle control and team decision making. For example, inadequate team control can lead to formation instability,[2] and poor decision making may prevent the team from reaching consensus on a plan, leading to unwanted consequences.[3]

Because UAVs are highly mobile, information is commonly exchanged using wireless communications. Signals with encoded messages travel between transmitting and receiving radio modules over wireless channels. The quality of the channel is fundamentally based on the strength of the signal at the receiver compared to noise, or Signal to Noise Ratio (SNR).[4] This channel quality drives the probability that information transmitted will successfully be received, and affects the rate at which information can be exchanged. Since wireless channels generally degrade with increasing distance and obstacles in the line of sight, controlling the network topology in multi-vehicle systems involves properly positioning agents to support the network. Common

Received 1 February 2013; Revised 5 March 2013; Accepted 31 March 2013; Published 12 June 2013. This paper was recommended for publication in revised form by the Editors-in-Chief.
Email Addresses: †*kopeikin@alum.mit.edu*, ‡*sponda@alum.mit.edu*, §*lbj16@mit.edu*, ¶*jhow@mit.edu*

ways to do this include planning the motion of agents using gradient ascent,[5] potential fields,[6,7] reactive control,[8] and adaptive strategies[9] to steer vehicles to stay within communication range of each other. Another method is to deploy agents designated as mobile communication relays to support a network of sensor nodes, using both graph theoretic[10] and network optimization schemes.[11] In previous work, we proposed a unique task allocation strategy for network communication control in a multi-agent team using relay tasks (CBBA with Relays),[12] demonstrating that, through cooperative planning, agents can be dynamically assigned to take on mission tasks or support network requirements as relays to improve global mission performance. This approach is fundamentally different from most other multi-vehicle network control methods where roles are typically predesignated. By explicitly coupling the task assignment and relay planning processes, CBBA with Relays can dynamically optimize the use of agents to better address current mission needs leading to improved performance and added flexibility in real-time dynamic mission scenarios.

This paper extends the CBBA with Relays algorithm proposed in [12] to include realistic networking dynamics such as path loss, fading, and information routing. In particular, this paper proposes extensions to address the following communication performance considerations: (1) data-rate, (2) data routing, (3) inter-agent bit-error-rate (BER), and (4) uncertainty. The paper proceeds as follows: Sec. 2 describes the task assignment formulation, background algorithms, and communication models employed in this work; Sec. 3 describes the extensions to the CBBA with Relays algorithm to control the network considering realistic communication constraints; Sec. 4 offers methods to estimate the communication environment and plan adaptively given uncertainty; Sec. 5 presents experimental results for both simulation and outdoor flight tests, demonstrating the performance of the proposed algorithms; and finally Sec. 6 provides closing remarks and future work directions.

2. Problem Statement

The objective in this paper is to enable a multi-UAV team to cooperatively execute a mission and to ensure that its wireless network supports the transmission of sensor data and command and control (C2) messages. This is illustrated by the notional cooperative aerial surveying mission shown in Fig. 1.

2.1. *Network model*

A fundamental factor in network wireless channel performance is the SNR. In general, channels degrade with increasing inter-node distance due to path loss. Signal shadowing occurs due to obstacles in the environment such as buildings, mountains, or even the platform vehicle itself. Finally, multipath replicas of the signal reflecting and scattering off these obstacles constructively or destructively interfere with each other at the receiver.[13] A practical way to model SNR in uncertain environments is to vary SNR (γ) according to a lognormal distribution with variance σ^2 which depends on the density of obstacles in the environment to capture both shadowing and multipath fading effects.[4] The SNR is conveniently expressed in decibels by Eq. (1):

$$\gamma_{dB} = 10\log\left(\frac{P}{N_o W}\right) + K_{dB} - 10\alpha\log\left(\frac{d}{d_o}\right) - \mathcal{N}(0, \sigma^2_{dB}), \tag{1}$$

where P is the transmission power, $\frac{N_0}{2}$ is the power spectral density of the environment noise, W is the bandwidth of the signal, K_{dB} is a gain based on equipment characteristics, d is the Euclidean distance between the transmitter and receiver, and α is the path loss exponent which equals 2 in free space, and up to 6 in environments congested with obstacles.[4,14]

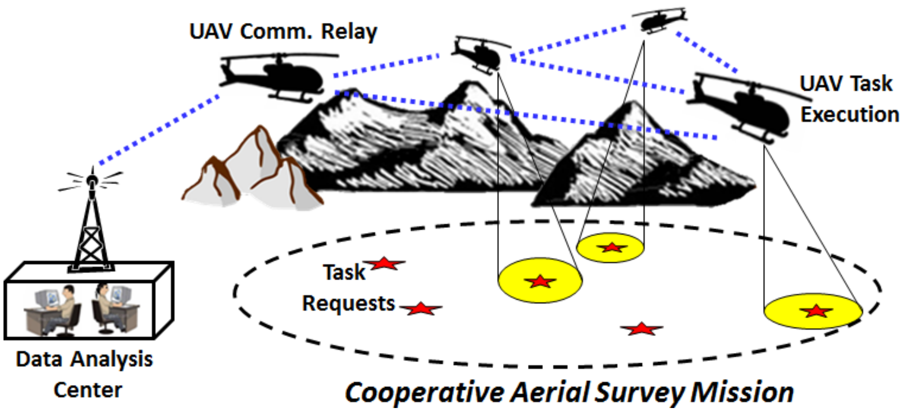


Fig. 1. (Color online) Cooperative aerial surveying mission with a multi-UAV team.

A multi-UAV system network typically needs to support two types of data. First, sensor data may need to be transmitted to designated analysis centers.[1] Communication requirements depend on the data type, but live-video, for instance, requires minimal delay, high data-rates, and has some fault tolerance.[15] As such, the *Shannon Capacity* described by Eq. (2) as a function of γ is used as a representative performance indicator to determine whether a channel satisfies data-rate requirements.

$$u = W \log_2(1+\gamma), \tag{2}$$

The second data type consists of C2 messages including state information (telemetry), observations of the world (e.g., estimated target location), and control data (e.g., waypoints or task allocation). These messages have low bandwidth requirements, but must be exchanged with minimal delay and error for effective team coordination. This study therefore, employs the BER as a metric for C2 messaging success. BER is the probability an information bit will be dropped, and is defined generally by Eq. (3).[4] To evaluate network performance, the locations of UAVs are used to determine the expected SNR of wireless channels between agents and predict the available data-rate and BER for information transfer between them.

$$\mathtt{BER} = Q(\sqrt{2\gamma}), \tag{3}$$

where

$$Q(z) = p(X \geq z) = \int_z^\infty \frac{1}{\sqrt{2\pi}} e^{-x^2/2} dx.$$

Information must be routed along links with sufficient capacities, and arrive at its destination with acceptable error-rates (BER), as dictated by data requirements. Information routing protocols in *ad-hoc* wireless networks consistent with a distributed multi-UAV architecture often use shortest path routing mechanisms, such as AODV.[16] A node which needs to transmit broadcasts a "request to send" message which identifies the destination node. Neighbors receive the message, compute a cost metric (see Sec. 3), and rebroadcast the message. The process continues, where cost is computed at every hop, until the destination is reached. Each node stores its lowest cost originator. Once a designated amount of time has elapsed, the destination node sends a unicast response to its lowest cost originator, which similarly relays the response to backtrack the lowest cost route.[16] These protocols are implemented in practice because of their rapid response to changes in wireless channel performance and topology. A significant drawback to them, however, is that information from different nodes is routed mostly independently of other nodes, and tends to use the same high performance routes leading to congestion and unfairness in node usage.[17]

2.2. *Distributed task allocation*

The complex missions envisioned to be executed by autonomous teams of robotic agents will require agents to have different roles and responsibilities during execution. The task allocation problem for a team of N_a heterogeneous unmanned systems and N_t tasks can be written as a mixed-integer optimization problem

$$\begin{aligned} \underset{\mathbf{x},\boldsymbol{\tau}}{\operatorname{argmax}} \; & \sum_{i=1}^{N_a} \left(\sum_{j=1}^{N_t} s_{ij}(\mathbf{p}_i(\mathbf{x}_i, \boldsymbol{\tau}_i)) x_{ij} \right) \\ \text{s. t.} \; & \sum_{j=1}^{N_t} x_{ij} \leq L_i, \quad \forall i \in \mathcal{I} \\ & \sum_{i=1}^{N_a} x_{ij} \leq 1, \quad \forall j \in \mathcal{J} \\ & x_{ij} \in \{0,1\}, \tau_{ij} \in \{\mathbb{R}^+ \cup \emptyset\}, \forall (i,j) \in \mathcal{I} \times \mathcal{J}, \end{aligned} \tag{4}$$

where $x_{ij} = 1$ if agent i is assigned to task j, and $\mathbf{x}_i$ is its vector of assignments. Vector $\mathbf{p}_i$ represents the *path* for agent i, a sequence of tasks listed in the order to be executed at times in $\boldsymbol{\tau}_i$. The length of $\mathbf{p}_i$ is limited by L_i, the planning horizon. The objective in Eq. (4) is to maximize the sum of the reward for each agent. The score s_{ij} that agent i obtains by performing task j is a function of the value of the task, the cost of execution (e.g., fuel consumed), and the execution time τ_{ij} compared to the optimal task execution time.[12]

The numerous possible combinations of agent to task assignments and the inherent interdependencies in score from executing tasks in different orders makes this problem difficult to solve for $L_i > 1$ (NP-hard). A variety of optimization algorithms and frameworks have been developed to solve this problem for a team of unmanned vehicles.[18] Centralized planners solving a Constraint Satisfaction Problem (CSP) or using dynamic programming can provide the optimal task assignment solution, but usually do so in an unreasonable amount of time given the dynamic nature of these complex missions.[19] Approximate solvers are also available and can be quite fast. For instance, a process algebra framework can model interactions between tasks and agents to optimize task allocation using an efficient state-space search algorithm.[20] Similarly, evolutionary optimization and genetic algorithms can rapidly steer a search of possible assignments to a good solution for large teams of UAVs involved in ISR operations.[21] The problem formulation can also be relaxed and then solved using a mixed integer linear Programming approach to find an optimized task allocation.[22] While these centralized planning methods generate good approximate solutions, they require a central node to have full situational awareness of all agents in the team.

In dynamic environments, however, distributed task planning frameworks offer several benefits for multi-agent teams.[19] Distributed frameworks reduce the need to communicate excessive amounts of state information to a ground control station, and can mitigate latency effects by enabling agents to leverage their immediate local situational awareness when generating task assignments. The main challenges associated with distributed planning are computing high performance task allocations and ensuring that all agents in the team have a consistent assignment. One method to ensure deconflicted solutions is to perform centralized space partitioning prior to planning and then allow agents to make distributed decisions in their areas of responsibility.[23] Another approach is to use *implicit coordination*, where agents reach consensus on the global situational awareness of the team, and then each solve the same problem using the centralized techniques listed above,[24–26] but this strategy usually involves significant communication overhead costs. Other popular methods to solve the task assignment problem include distributed auction algorithms which provide efficient approximate solutions while not requiring consistent situational awareness across the team. The Consensus-Based Bundle Algorithm (CBBA)[27] is one such auction protocol that provides provably good approximate solutions for heterogeneous distributed multi-agent multi-task allocation problems. Each CBBA iteration is composed of two phases: (1) a bundle building phase, where each agent greedily places bids on desired tasks according to its capabilities and state; and (2) a consensus phase, where agent bid information is shared, and conflicts in task assignment are identified and resolved. The algorithm iterates between these two phases until it converges to a conflict-free plan.[27] The CBBA algorithm runs in polynomial time and has demonstrated good scalability with increasing numbers of agents and tasks. It has also been experimentally demonstrated in numerous applications with teams of heterogeneous unmanned systems.[3,12,28]

2.3. *Distributed task allocation for network control*

A complicating factor not considered by the CBBA algorithm is that the multi-UAV system needs to meet communication requirements as described in Sec. 2.1 to operate effectively. Communication constraints can be explicitly considered in the planning process. Task allocation information, such as task locations and planned execution times, can be leveraged by the agents to predict the network topology at execution and identify communication constraint violations. In a previous installment of this work,[12] we developed the CBBA with Relays algorithm as an outer-loop to CBBA to prevent network disconnects. When potential disconnects are identified in a CBBA task allocation, relay tasks are created to repair the network. The CBBA algorithm then iterates over the task allocation including the relay tasks. If agents can accommodate relay tasks into their plan they place appropriate bids. Following each iteration of the algorithm, unassigned relays are deleted and tasks dependent on the deleted relays are placed in the bidding agent's forbidden task list to ensure that the CBBA with Relays algorithm eventually converges on a task assignment solution.[12]

In this paper, the previously proposed CBBA with Relays algorithm is further developed to consider realistic networking dynamics as described in Sec. 2.1. These include data-rate, BER, information routing, and uncertainty. The objective is to prevent data transmission bottlenecks and poor interconnectivity which can affect the performance of the multi-UAV system (see Fig. 2). The models listed above, along with known UAV positions, can be used to determine the expected SNR of wireless channels between agents, and to predict the available data-rate and BER for information transfer between them. The resulting communication graph can then be used to predict how information is routed through the network, as will be discussed later in Secs. 3.2 and 3.3. These models for wireless communications assume

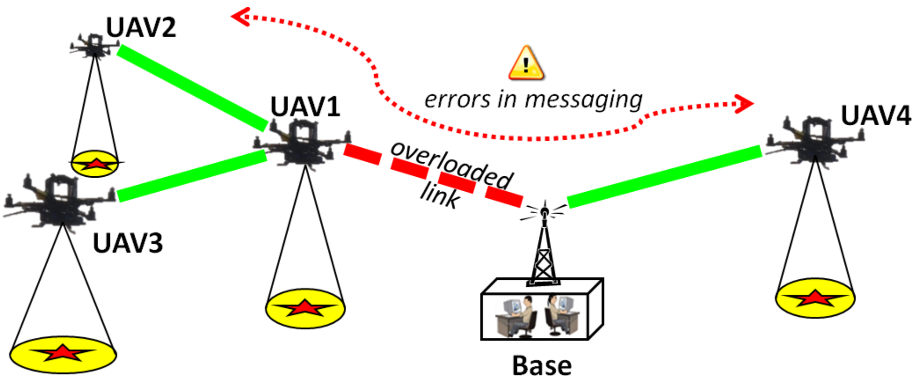


Fig. 2. (Color online) Overloaded and poorly interconnected multi-UAV network.

the following:

- There are two different communication systems on each agent, one to handle sensor data (video link) and one for command and control (C2 link). This is a common implementation on modern UAV systems,[29] and allows separate optimized routing for both types of messages. For simplicity in this work, both communication systems share the same parameters and numerical values but are routed differently.
- Communication devices can transmit and receive data at the same time. This is commonly achieved by incorporating multiple interfaces to the physical layer.[30]
- Wireless channel frequencies are efficiently assigned.
- There are no delays in transmissions, and the multiple nodes do not interfere with each other.

3. Task Allocation for Network Control with Realistic Communications

3.1. *General framework approach*

This section describes a planning framework modified from the original CBBA with Relays algorithm[12] which can handle the communication dynamics discussed in Sec. 2.1. The extended CBBA with Relays algorithm is summarized in Algorithm 1 and illustrated in Fig. 3 for convenience. The following is a detailed description of the key algorithmic pieces (referencing Fig. 3):

- Block 1: The algorithm begins with the CBBA task assignment over all tasks and relays (created in the previous iteration).
- Block 2: Relays which are unassigned, or now *irrelevant*, are removed and their dependencies dropped as described in the original implementation (see Prune-Task-Space Algorithm and Keep-Task($j_{\text{inf}}, \mathcal{A}$) function in [12]). To accomplish this, each relay tracks the set of assignments used in the network topology prediction at the time of its creation. If any of those assignments have changed, for example if another agent out-bids one of the tasks to execute it early, the relay becomes irrelevant and is deleted (Algorithm 1, lines 8–13).
- Block 3: For each assigned task j, agent i assigned to it predicts the network topology $\mathcal{G}$ using: (1) CBBA information containing the set of committed tasks and their start times, and (2) that agent's current belief of the communication environment parameters $\mathcal{C}_i$ as described in Sec. 4 (Algorithm 1, line 18).
- Block 4: The set of tasks $\mathcal{J}_{\text{disconnected}}$ which create network requirement violations, or *disconnects*, are identified. All other tasks in this network prediction $\mathcal{G}$ are therefore, connected and allowed to be executed without relays assigned to them. This step varies depending on which type of communication requirement is used for planning. For instance, the method to find data-rate overload *disconnects* $\mathcal{J}_{\text{disconnected}}$ with AODV routing is outlined in Algorithm 3. Different types of communication requirements are discussed in Sec. 3.2 and 3.3.
- Block 5: Given the network $\mathcal{G}$ and its disconnected tasks $\mathcal{J}_{\text{disconnected}}$, relays are sequentially created in a greedy value optimization scheme, described in Algorithm 2.

This process iterates until Algorithm 1 converges, which occurs when all assignments are supported by the network, no new relay tasks are created, and the assignment is consistent with the previous iteration.

The objective of Algorithm 2 is to create relays conducive to high-reward achievable assignments in the next CBBA iteration. This does not necessarily mean reconnecting all tasks using the minimum number of relays. Reward is maximized by performing high value tasks on time while incurring minimal cost. Relay tasks alone do not generate any reward since their agents are not generating mission data, however, they do incur costs. As such, it is only beneficial to execute relay tasks if the sum of the reward they enable through reconnected tasks is greater than the sum of (1) the cost they incur, (2) the reward opportunity passed up by their agents, and (3) the reward opportunity available to reconnected agents by executing connected tasks instead. Determining the optimal relay creation based on the needs

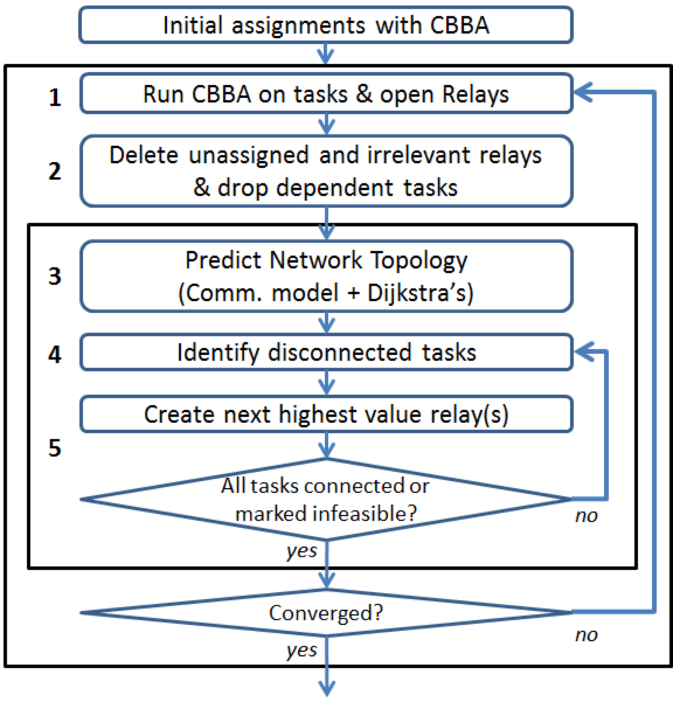


Fig. 3. Improved general approach to CBBA with relays to handle realistic communication requirements.

Algorithm 1 EXTENDED CBBA-RELAYS($\mathcal{I}$, $\mathcal{J}$, $\mathcal{J}_{\text{active}}$)

```
J_i = J, ∀i ∈ I; R = ∅; J̄ = {J_1, ..., J_{N_a}, R}
while ¬ converged do
  A ← CBBA(J̄) (See [27] for details)                               Block 1
  for (r ∈ R) do
    if (r ∉ A) then
      (J̄', A') ← PRUNE-TASK-SPACE(r, J̄, A)
    else
      for (a ∈ A_r') do
        if (a ∉ A) then
          J_{Winning−Agent(a)} ← J_{Winning−Agent(a)} \ {j}             Block 2
          (J̄', A') ← PRUNE-TASK-SPACE(r, J̄, A)
        end if
      end for
    end if
  end for
  for (j ∈ A') do
    C ← C_{Assigned−Agent(j)}
    G ← Predict-Topology(J̄', A', C)                                  Block 3
    J_disconnected ← FIND-DISCONNECTS(J_active, G)                     Block 4, see Alg. 3
    (J̄'', J_infeasible) ← PLACE-RELAYS(J̄', J_disconnected, G, C)      Block 5
    for (j_inf ∈ J_infeasible) do
      keep ← Keep-Task(j_inf, A)
      if ¬keep then
        A ← A \ {j}
        J_{Winning−Agent(j_inf)} ← J_{Winning−Agent(j_inf)} \ {j_inf}
      end if
    end for
  end for
  if (J̄'' = J̄) & (A' = A) then
    converged ← true
  end if
  J̄ ← J̄'' and A ← A'
end while
return A
```


of the dynamic network scales exponentially in the number of agents and tasks, and quickly becomes intractable. Instead, Algorithm 2 provides an efficient approximate solution by sequentially choosing the next best task j in $\mathcal{J}_{\text{disconnected}}$ to reconnect using relays. For each j, it computes optimized relay locations to satisfy communication requirements (Algorithm 2, line 10). Relays are then temporarily added to the network graph $\mathcal{G}$ and the FIND-DISCONNECTS function from Algorithm 1 then determines which additional tasks from $\mathcal{J}_{\text{disconnected}}$ are now reconnected. The value of the created relays is then determined using

$$v_r = K_v \sum_{j}^{\mathcal{J}_{\text{depend}}(r)} \frac{s_{ij}}{N_r{}'(j)}, \tag{5}$$

where $\mathcal{J}_{\text{depend}}(r)$ indicates the set of tasks j dependent on relay r, s_{ij} is the predicted reward for j from CBBA (as determined by the winning agent i), $N_r{}'(j)$ is the number of other relays j depends on, and K_v is a constant multiplicative scaling factor. For all disconnected tasks, the algorithm finds the highest value task to reconnect and adds that task with other reconnected tasks $\mathcal{J}_{\text{connected}}$, and its relays $\mathcal{R}'$, to the network graph $\mathcal{G}$. These will be added to the CBBA auction for task allocation. This process is repeated until all tasks are reconnected or designated as infeasible. Labeling tasks as infeasible eliminates task assignment options which are bound to fail but may cause additional CBBA iterations. Relays created in this process are then awarded a *virtual value* determined by Eq. (5)

Algorithm 2 PLACE-RELAYS($\bar{\mathcal{J}}'$,$\mathcal{J}_{\text{disconnected}}$,$\mathcal{G}$,$\mathcal{C}$)

$\mathcal{R} = \emptyset$
$\mathcal{J}_{\text{infeasible}} = \emptyset$
$\mathcal{J}_{\text{active}} = (j \in \mathcal{G})$ & $(j \notin \mathcal{J}_{\text{disconnected}})$
while $\neg$ *done* **do**
  $v_{\max} = 0$
  $newRelay = false$
  for $(j \in \mathcal{J}_{\text{disconnected}})$ **do**
    ${\mathcal{J}_{\text{disc}}}' \leftarrow \mathcal{J}_{\text{disconnected}} \setminus j$
    $\mathcal{J}_{\text{connect}} \leftarrow j$
    $(\mathcal{J}_{\text{infeasible}},\mathcal{R}',\mathcal{G}') \leftarrow$ FIX-DISCONNECT($\mathcal{G}$,j,$\mathcal{J}_{\text{infeasible}}$,$\mathcal{J}_{\text{active}}$,$\mathcal{C}$)
    ${\mathcal{J}_{\text{active}}}' \leftarrow \mathcal{J}_{\text{active}} + \mathcal{R}' + j$
    ${\mathcal{J}_{\text{disc}}}'' \leftarrow$ FIND-DISCONNECTS($\bar{\mathcal{J}}'$,${\mathcal{J}_{\text{active}}}'$,$\mathcal{G}'$, $\mathcal{C}$)
    $\mathcal{J}_{\text{connect}} \leftarrow \mathcal{J}_{\text{connect}} + \{(j' \in {\mathcal{J}_{\text{disc}}}')\&(j' \notin {\mathcal{J}_{\text{disc}}}'')\}$
    $(\mathcal{R}',\mathcal{J}_{\text{connect}}) \leftarrow$ Find-Dependencies($\mathcal{G}'$)
    $v \leftarrow$ Compute-Value($\mathcal{R}'$, $\mathcal{J}_{\text{connect}}$)
    if $(v > v_{\max})$ **then**
      $v_{\max} = v$
      ${\mathcal{R}_{\max}}' \leftarrow \mathcal{R}'$
      $\mathcal{G}_{\max} \leftarrow \mathcal{G}'$
      $\mathcal{J}_{\text{connect}-\max} \leftarrow \mathcal{J}_{\text{connect}}$
      $newRelay = true$
    end if
  end for
  if $newRelay$ **then**
    $\bar{\mathcal{J}}' \leftarrow \bar{\mathcal{J}}' + \{{\mathcal{R}_{\max}}', \mathcal{J}_{\text{connect}-\max}\}$
    $\mathcal{J}_{\text{active}} \leftarrow \mathcal{J}_{\text{active}} + \{{\mathcal{R}_{\max}}', \mathcal{J}_{\text{connect}-\max}\}$
    $\mathcal{J}_{\text{disconnected}} \leftarrow \mathcal{J}_{\text{disconnected}} \setminus \{\mathcal{J}_{\text{connect}-\max}\}$
    $\mathcal{J}_{\text{infeasible}} = \emptyset$
    $\mathcal{G} \leftarrow \mathcal{G}'$
  else
    $done = true$
  end if
end while
return $(\bar{\mathcal{J}}',\mathcal{J}_{\text{disconnected}})$

recomputed after all relays have been created. This incentivizes agents to bid on relay tasks during the CBBA task assignment. The desired cooperative team behavior can be adjusted through the choice of K_v in Eq. (5).

The CBBA with Relays algorithm presented in this section guarantees convergence and offers real-time performance as demonstrated in Sec. 5. The following two subsections describe how the framework is tailored to address data-rate (Sec. 3.2) and error-rate requirements (Sec. 3.3). In both cases, the planner runs Algorithm 1, but identifies network violations (line 19) and creates relays (line 20) differently.

3.2. *Supporting data-rate requirements*

In order to transmit live sensor data back to the base, the multi-UAV network must ensure information routes have sufficient end-to-end channel capacity at the time of execution. The algorithmic approach in Sec. 3.1 is tailored based on the data-rate requirements and information routing protocols. Many protocols exist, along with many variations, each with strengths and weaknesses.[16,17,30–35] As such, this section includes two different representative routing strategies: (1) the model of a protocol representative of shortest path routing schemes commonly used in decentralized wireless networks, and (2) a protocol providing the theoretical optimal routing solution. The purpose is to show the commonality in the planning strategies and provide performance bounds of the system as better future protocols become available.

3.2.1. *Realistic routing*

Information routing protocols in *ad-hoc* wireless networks consistent with a distributed multi-UAV architecture often use shortest path routing mechanisms, such as *Ad-hoc* On-Demand Distance Vector (AODV).[16] AODV employs local information at every node to dynamically compute the shortest transmission route according to an established metric. This protocol can rapidly adapt to changes in topology, but also tends to converge to commonly used shortest routes in multi-user scenarios leading to data bottlenecks.[17] The metric employed in this work to find the *shortest* transmission path is a representation of the *airtime* metric popular for these applications and described by Eq. (6),[36]

$$C_{Aij} = \left(O + \frac{B_t}{f}\right)\frac{1}{1 - p_{\text{pkt}}}, \tag{6}$$

where p_{pkt} is the packet loss ratio (based on BER), O is overhead, B_t is the test-frame length, and f is the information flow data-rate (based on capacity). Here, parameters are set to: $O = 0$, $B_t = 1$, $p_{\text{pkt}} = 0$, and $f = u$, which is the wireless channel data-rate capacity. As such, the cost metric $C_{A_{ij}}$ for transmitting between any two nodes i and j is simply $C_{A_{ij}} = \frac{1}{u_{ij}}$. To transmit from node i to k, the protocol finds the path which minimizes $\sum C_{A_{ij}}$ over any number of hops. This identifies routes with high data-rates, and simultaneously minimizes the number of hops to destination, a desirable behavior in high-bandwidth transfers.[4]

The predicted network graph $\mathcal{G}$ at the time of task execution (see Algorithm 1, line 18) is used as input to simulate information routing. In this simulation, route discovery is performed using the well known Dijkstra's shortest path algorithm.[37] Flows are then summed over each generated path to determine the channel capacity requirement between each node as:

$$u_{\text{req}ij} = \sum_k f_{ki} + f_{\text{task}_i}. \tag{7}$$

Predicted data-rate violations are identified when $u_{\text{req}ij} > u_{ij}$. This information is then used by the planner to decide: (1) which tasks are allowed to be executed given the present configuration, and (2) how best to create relays to support disconnected tasks. This is performed using Algorithm 3 described as follows, which is called in Algorithm 1, line 19.

Choosing which tasks should be allowed versus dropped if data-rate overloads exist (Fig. 2) is not trivial. Since the mission objective is to maximize team reward, a selection process is required to find the combination of proposed tasks which provides the highest reward without violating data-rate requirements. To avoid an exhaustive search over the set of all tasks, Algorithm 3 leverages the Minimum Spanning Tree (MST) network graph structure produced by AODV routing. The tree is rooted at the base and each node sends its data to a single other node. The algorithm therefore, greedily prunes lowest value leaves (or end node tasks) that are up-link of edges with data-rate violations (Algorithm 3, line 9). If all end-nodes are connected tasks being executed (in $\mathcal{J}_{\text{active}}$) which were carried over from a

Algorithm 3 FIND-DISCONNECTS-DATARATE-AODV($\mathcal{J}_{\text{active}}$, $\mathcal{G}$)

1: $\mathcal{J}_{\text{disconnected}} \leftarrow j \notin \mathcal{G}_{\text{sub-base}}$
2: $\mathcal{G}' \leftarrow \mathcal{G} \setminus \mathcal{J}_{\text{disconnected}}$
3: $(U_{\text{req}}, \mathcal{P}) = \text{ROUTE-DATA}(\mathcal{G}', \text{AODV})$
4: $\mathcal{J}_{\text{m}} \leftarrow \arg\left(u_{\text{req}_{\text{m,n}}} > u_{\text{m,n}}\right)$
5: **while** $\neg(\mathcal{J}_{\text{m}} = \emptyset)$ **do**
6:     $m = \arg\max\left(\mathcal{P}_{\text{j,base}}\right), \forall j \in \mathcal{J}_{\text{m}}$
7:     **while** $u_{\text{req}_{\text{m,n}}} > u_{\text{m,n}}$ **do**
8:         $\mathcal{J}_{\text{candidate}} \leftarrow \text{Find-Children}(m, \mathcal{P}, \mathcal{J}_{\text{active}})$
9:         $j_{\text{del}} = \text{Select-Node-to-Delete}(\mathcal{J}_{\text{candidate}})$
10:         $\mathcal{J}_{\text{disconnected}} \leftarrow \mathcal{J}_{\text{disconnected}} + j_{\text{del}}$
11:         $\mathcal{G}' \leftarrow \mathcal{G}' \setminus \{j_{\text{del}}\}$
12:         $(U_{\text{req}}, \mathcal{P}) = \text{ROUTE-DATA}(\mathcal{G}', \text{AODV})$
13:     **end while**
14:     $\mathcal{J}_{\text{m}} \leftarrow \arg\left(u_{\text{req}_{\text{m,n}}} > u_{\text{m,n}}\right)$
15: **end while**
16: **return** $(\mathcal{J}_{\text{disconnected}})$

previous planning cycle, the algorithm prunes the furthest (number of hops) and lowest value task up-link of a violation not in $\mathcal{J}_{\text{active}}$. In this case, routing must be recomputed with the remaining graph. This process repeats until all remaining tasks are supported by the network.

Once the set of disconnected tasks is identified, the planning framework uses Algorithm 2 to reconnect them with relays. Line 10 in Algorithm 2 uses the known required link capacity $u_{\text{req}ij}$ to satisfy flow requirements on the lowest cost route, along with Eqs. (1) and (2) to solve for the maximum inter-node distance $d_{\max}$ to meet the requirement. The number of required relays and equally spaced positions is computed ($N_r = \lceil \frac{d_{ij}}{d_{\max}} \rceil$). Once relays are created, information routing is recomputed to identify possible new link violations. This process repeats until all tasks in the configuration are connected or designated infeasible. Ultimately, the *virtual value* of all relay tasks created in this iteration are evaluated using Eq. (5), and added to the CBBA auction.

3.2.2. *Optimized routing*

Shortest path routing protocols, such as AODV, have limitations,[17] and better performing protocols are being developed.[38–40] In order to provide a performance bound of the planning framework independent of the intricacies of specific routing protocols, a mechanism simulating optimal data routing is proposed. This simulated optimal protocol assumes perfect real-time knowledge of the global network topology, and precise flow control node to node. These assumptions distinguish this optimal routing protocol from the previous AODV model:

- A transmitting node can split its data flow to go out to multiple other nodes, rather than sending all its data to a single other node as in AODV.
- The optimal protocol simultaneously optimizes transmission requirements of all active nodes to distribute information flow through available links. In AODV, each node routes its information on its best path independently from other nodes, leading to possible bottlenecks.

The routing mechanism is formulated as a Linear Program (LP) solving a network flow optimization problem[37] as follows

$$\arg\min_{f} \sum_i \sum_j \left(c_{ij} f_{ij} + M \frac{1}{u_{ij}} \epsilon_{ij} \right). \tag{8}$$

$$\text{s.t.} \sum_j f_{ij} - \sum_j f_{ji} = \begin{cases} f_{\text{task}_i} & \text{if } i \text{ is an active task} \\ 0 & \text{if } i \text{ is a relay} \\ -\sum_i f_{\text{task}_i} & \text{if } i \text{ is the base.} \end{cases} \tag{9}$$

$$\sum_j \frac{f_{ij}}{u_{ij}} - \epsilon_{ij} \le 1. \tag{10}$$

$$0 \le f_{ij} \le C u_{ij}. \tag{11}$$

$$0 \le \epsilon_{ij} \le C. \tag{12}$$

This solves the minimum cost flow f to transmit all node data to the base. Slack variable ϵ allows the network to *overload* its link capacities as needed to reach a solution. Here, $\sum_j \epsilon_{ij}$ is the percent overload of an agent, and identifies network data-rate bottlenecks. In this study, the cost function is simply set to $c_{ij} = 1$ to incentivize reducing the number of hops from a node to the base.[38] The cost function for ϵ, $\frac{M}{u_{ij}}$ deters links from overloading unnecessarily by setting M to a large value, but incentivize fewer stronger links to be overloaded if needed.

The first LP constraint in Eq. (9) describes the conservation of flow principle true for all closed networks.[37] Relay tasks do not generate flow of their own and as such $f_{\text{task}_{\text{relay}}} = 0$. The base station is treated as a sink for all incoming flows $\sum_i f_{\text{task}_i}$. The next constraint, Eq. (10), describes the total transmission constraint of each node. If a node transmits at full available data-rate to a single other node, it is fully utilizing its resources to do so. In order to split its flow out, it must reduce its transmission rate to the first node to award channel utilization resources to others. Finally Eqs. (11) and (12) constrain flows and overloads to be positive and bounded by capacity, with C as a constant to tune the maximum percent overload allowed on each link.

To ensure the multi-UAV team collaboratively executes the mission while satisfying data-rate requirements, the planner runs a modified version of Algorithm 3, also called in Algorithm 1, line 19. First, for planning purposes only, a spanning tree is created using the largest flow out of each node. This helps identify nodes which contribute most to overloaded links by summing slack values ϵ along their path to the base. A greedy process again is used to sequentially prune nodes until the network is supported, starting with end nodes. Network routing is recomputed after each node. To minimize the number of recomputations, the next node to delete is not selected simply based on lowest value, but also by considering a weighting factor of its total slack contribution, or overload. The remainder of this selection process is identical to that presented in Algorithm 3. Once disconnected tasks are identified, relays are created using the process as presented in the previous subsection with some notable differences. First, since network routing here does not consist of a spanning tree, the search space for optimal relay placement is reduced by connecting disconnected tasks directly to the base. This generally requires more relays at each placement, but also reconnects more tasks per step since data is efficiently routed in a convex task space. Second, relay and task dependencies cannot be determined explicitly based on routing paths since all links are inter-dependent. As such, relays and tasks added at every step in the algorithm are dependent on relays created in previous steps.

3.3. *Supporting error-rate requirements*

In addition to satisfying data-rate requirements, the multi-UAV network must also ensure that agents meet BER requirements to adequately exchange C2 messages. Routing for these messages is assumed to be independent of routing sensor data back to the base. As data travels over multiple hops to its destination, each hop has a probability of error. Over n hops, the probability of error is $p_e = 1 - ((1 - p_{e1})(1 - p_{e2}) \cdots (1 - p_{en}))$ (see Fig. 4).[15] Since error probabilities are multiplicative over multiple hops, the additive cost metric of each hop is set to $C_{\text{ber}} = -\ln(1 - \mathtt{BER}_{ij})$.[9] Dijkstra's Algorithm is used again to generate the minimum cost routes between all agents. BER violations occur if any pair of agents i and j have $\mathtt{BER}_{ij} > \mathtt{BER}_{\text{thresh}}$ which is the threshold maximum allowable error rate for the system to be effective.

Once BER violations are identified, the framework greedily prunes disconnected tasks through a modified version of Algorithm 3. First, all tasks which violate BER with the base are set as disconnected. Next, the algorithm considers the data-rate routing MST, and prunes its low value end-nodes with BER violations in an effort to avoid changes in satisfactory data-routing. Finally, if violations are still present, the algorithm greedily selects low value nodes to remove. Disconnected tasks are then greedily repaired by sequentially adding relays along the computed shortest BER path between the two neighboring nodes with the weakest link until the requirement is met. This process is run in series while ensuring data-rate requirements are met for all agents executing tasks at the time of execution. The resulting relay tasks are then added to the CBBA auction to allow under-utilized agents to support the network.

This process does not ensure, however, that other agents which are not busy executing tasks at the time of network prediction are connected. In order to drive these other agents, which are either idle or traveling to their next task, to an interconnected state, a decentralized reactive motion control policy is implemented. Its objectives are to minimize additional fuel costs incurred from extra travel, but also ensure that agents can satisfy their proposed plans. Each

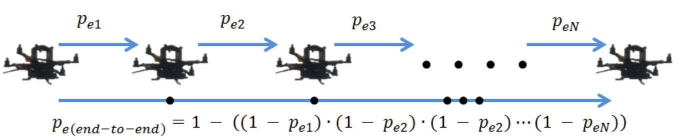


Fig. 4. (Color online) Probability of messaging error hop to hop.

agent i uses its current knowledge of other agent locations and communication environment parameters to locally estimate the network topology and BER values with other agents. It determines if it is at risk of violating BER with any other agent m if $\mathtt{BER}_{\mathtt{im}} > \mathtt{BER}_{\mathtt{plan}}$, where $\mathtt{BER}_{\mathtt{plan}}$ is a tunable planning threshold less than $\mathtt{BER}_{\mathtt{thresh}}$. It then plans to reconnect with the agent most at risk of being disconnected with, above this threshold, through the estimated minimum BER route. It computes its minimum distance $d_{\text{min}_{ij}}$ to the next agent j in the route needed for $\mathtt{BER}_{\mathtt{im}} = \mathtt{BER}_{\mathtt{plan}}$ and starts traveling to meet that requirement. This is dynamically updated to steer agent i. Because agents' plans are for discrete task execution times instead of continuous mission time, it is not possible to guarantee that all agents will be interconnected at all times. However, results in Sec. 5 show good performance of this strategy.

4. Planning Under Uncertainty

In realistic environments, shadowing and multipath fading lead to uncertainty in wireless channel performance. In addition, dynamic obstacles, noise, and interference from other sources can further vary network performance. This uncertainty must be taken into consideration when planning to ensure that mission execution is robust to degradations in the environment.

The SNR, γ_{dB}, the fundamental parameter in wireless channel capacity and BER performance is commonly modeled with normally distributed shadowing and fading distortions (see Eq. (1)). If the fading distribution variance σ_{dB}^2 is known, conservatism can be added to the planning process to drive the system performance to a desirable confidence level. This type of *risk adjustment* is often employed to add robustness in planning for multi-agent systems.[19] The SNR used to predict the network, identify disconnected tasks, create relays, and plan connectivity motion can be degraded by a certain amount to ensure a confidence level of its minimum performance. In Eq. (4), K_m is used as a multiplier of σ_{dB} to tune the confidence level. $K_m = 0$ is the center of the normal distribution, and as such offers 50% confidence that $\gamma_{ij} > \gamma_{\text{plan}}$, whereas $K_m = 1$ offers 84% confidence.

$$\gamma_{\text{plan}\,dB} = 10 \log\left(\frac{P}{N_o W}\right) + K_{dB} - 10\alpha \log\left(\frac{d}{d_o}\right) - (K_m \sigma_{dB}). \qquad (13)$$

A common problem encountered in field operations involves predicting the performance of uncertain wireless communications.[8,41,42] Often the attenuation, fading, and interference levels are initially unknown, and can vary spatially and temporally during mission execution.[43] Previous efforts have attempted to sample the communication environment *a priori* to construct a *radio-map* indicating areas of high network performance. However, these were then undercut by changes in performance during actual mission execution.[8] Therefore, this work employs a dynamic real-time communication model estimation method for the multi-UAV system to initially learn the communication environment, sense changes during execution, and adapt the planning strategy accordingly.

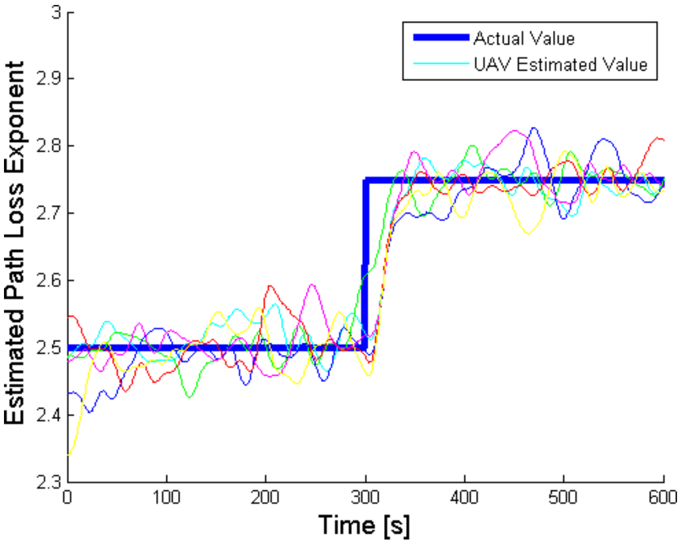
A Least-Squares (LS) estimation method obtained from and described in [9] was adapted for this application. It assumes that over the course of mission execution, agents collect and share SNR samples γ_s with each other, and know each others' positions and therefore, relative distances d_s to other agents. These measurements are then used to estimate the global environment communication parameters for planning purposes. The SNR sample γ_s can be rewritten as Eq. (14), where Φ is defined in Eq. (15), and $\gamma_{f_s dB} \sim \mathcal{N}(0, {\sigma_{dB}}^2)$ is the stochastic contribution to SNR due to fading in that sample.

$$\gamma_{sdB} = \Phi - 10\alpha \log\left(\frac{d_s}{d_o}\right) - \gamma_{f_s dB}. \quad (14)$$

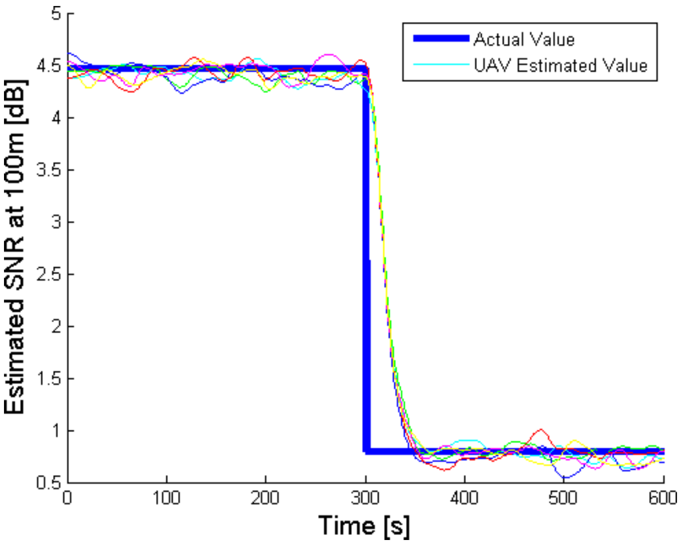
$$\Phi = 10 \log\left(\frac{P}{N_o W}\right) + K_{dB}. \quad (15)$$

It is reasonably assumed that each agent knows the transmission power P, bandwidth W, and equipment gain K_{dB} used in the sample. However, environment parameters such as the path loss exponent α, fading variance ${\sigma_{dB}}^2$, and noise power density N_0 are unknown, may vary, and are estimated through LS estimation.[9]

A weighted moving average filter is applied to the estimated values to smooth out disturbances and allow values to vary over time with changes in the environment. As implemented, each agent performs this estimation over a different set of measurements in a fully decentralized setting, and may therefore have different parameter estimates from other agents. These estimates are then used in the following stages of planning: network topology prediction (Fig. 3, Block 3), identification of disconnected tasks (Block 4), relay placement (Block 5), and reactive motion control planning to maintain inter-agent BER. Figure 5 shows the estimation profile of the three initially unknown communication parameters over the course of a 600 s mission with six UAVs. Path loss α, Fading variance γ_{dB}, and Noise Floor N_0 are increased by 10% halfway through the scenario. Estimates of these parameters are used to estimate expected SNR performance for task allocation planning. Figure 5(b) shows that the agents all quickly assess and adapt to the change in communication environment, which improves system performance as demonstrated in Sec. 5.



(a) Path loss (α)



(b) SNR (γ_{dB}) at 100 m

Fig. 5. (Color online) Estimation of simulated communication environment parameters with $N = 6$ agents.

5. Experiments & Results

A full system framework was developed to conduct multi-UAV missions with this algorithm both in simulation and flight tests. The system consists of multiple modules operating in the MATLAB environment, many modified from previous efforts.[12] Framework components are illustrated in Fig. 6, and described as follows:

- Simulator Mission Manager: This module runs the mission, primarily by creating and managing tasks. These are created at uniformly distributed random locations, times,

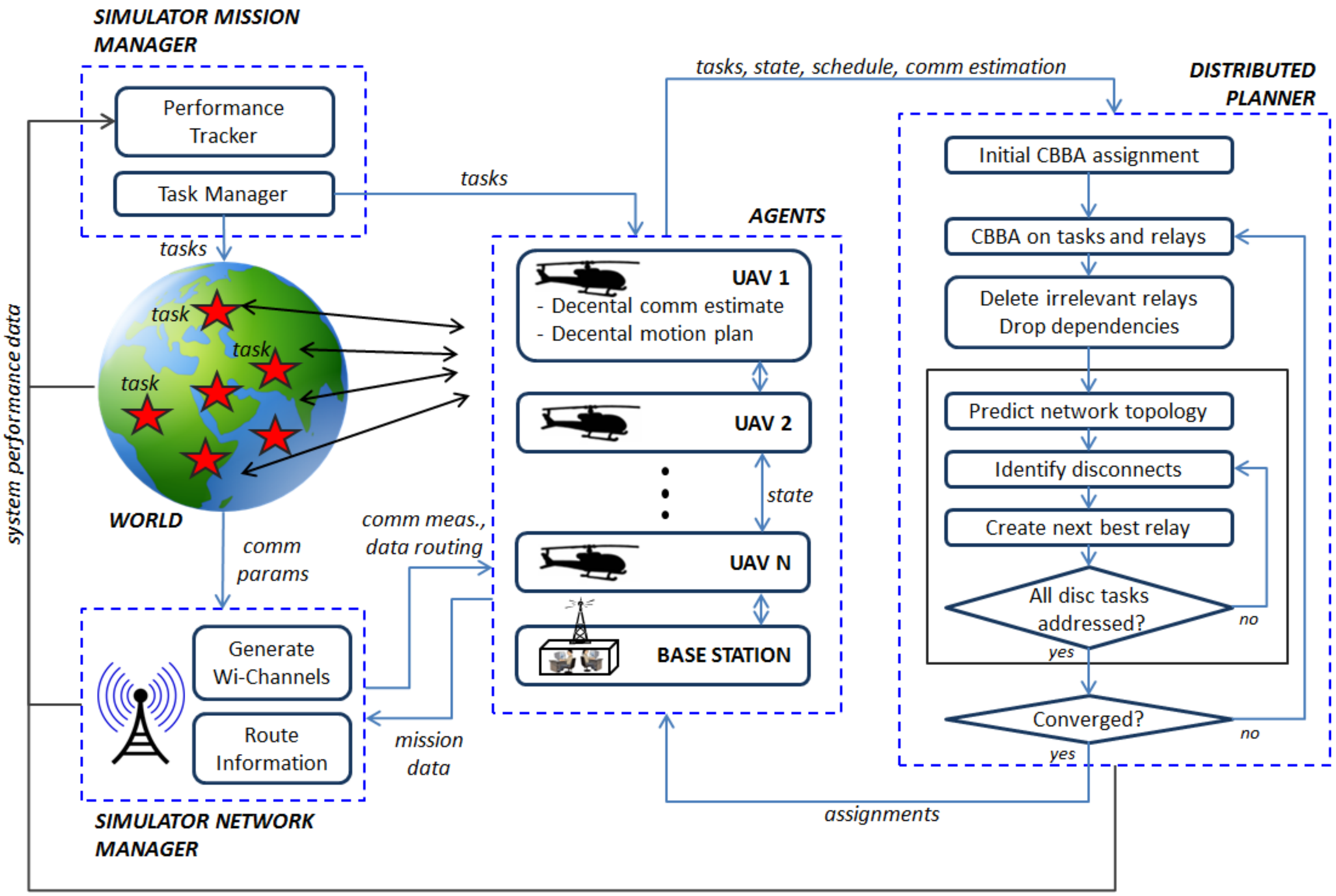


Fig. 6. (Color online) System framework block diagram.

and windows of validity. The value of each task is scaled linearly as a function of its distance from the base. The purpose for this setting is to incentivize agents to co-operate to execute more difficult tasks from the perspective of network connectivity.

- Agents: This module is distributed into submodules for each agent, which each simulates its agent state, estimates the communication environment, and controls motion. This module interfaces with UAV autopilots during flight testing.
- Distributed Planner: This module runs the task assignment routine using the distributed CBBA with Relays algorithm.
- Simulator Network Manager: This module generates wireless communication channels using the models in Sec. 2.1 and manages routing information according to the protocols presented in Secs. 3.2 and 3.3. The separation of the planning module and the network manager module is key. The planner uses the model of the communication environment estimated by the planning agent to formulate a task assignment plan. It does not however, actively control network routing during actual task execution, and the network could therefore, behave differently than planned. This is important for two reasons: (1) wireless channels may vary significantly over short time-scales so the ability to adapt to these unplanned changes through routing is important,[4] and more importantly (2) the multi-vehicle system may not have dynamic control of information routing and is therefore at the mercy of the protocol behavior.

5.1. *Simulation — comparing planning strategies*

A simulation experiment was conducted to evaluate the proposed framework planning for different communication requirements. The mission scenario consisted of gathering information in the environment using small UAVs. Tasks were created at uniformly distributed random locations and times. The task value was scaled linearly as a function of its distance from the base (value = 0 at base versus 100 at furthest location). This relates well to real-world remote sensing objectives which often consist of surveying out-of-reach areas. Tasks further away offer more reward, but require agents to cooperate to execute them successfully. The expected SNR at a distance of $R_{\text{comm}} = 30\%$ the environment size was used to establish the required task data-rate and the single hop BER threshold. Reward was only generated when data-rate requirements to the base were supported at each time step. In other words, if a specific task was only supported 50% of the time during its execution, it received 50% of its reward.

Four planning strategies were evaluated. These consisted in planning to:

- Meet simplified *connectivity* requirements only, not considering complex networking dynamics.
- Meet data-rate requirements considering the AODV routing model (Sec. 3.2).
- Meet data-rate requirements, but considering the LP based optimal routing algorithm instead.
- Simultaneously meet data-rate and BER using relays and reactive motion policy (Sec. 3.3).

To compare the planning effectiveness, these experiments assumed a deterministic communication environment with no fading ($\sigma_{dB} = 0$). The experiment varied the team size from 2 to 10 UAVs, and 30 scenario trials were executed per data point.

Figure 7 shows the performance results of the experiments. All strategies performed similarly for small counts of agents. However, as the team size grows, teams considering connectivity only, and not the more complex networking dynamics, quickly overload the network with their plans. As can be seen in Fig. 7(b), agents planning for connectivity only without stricter requirements attempt more and higher value tasks. However, Fig. 7(c) shows these plans are too ambitious and often lead to data-rates not supported by the network (as low as 38.6% average supportability for teams of 10 UAVs) which results in significant mission performance degradations compared to other strategies considering data-rate.

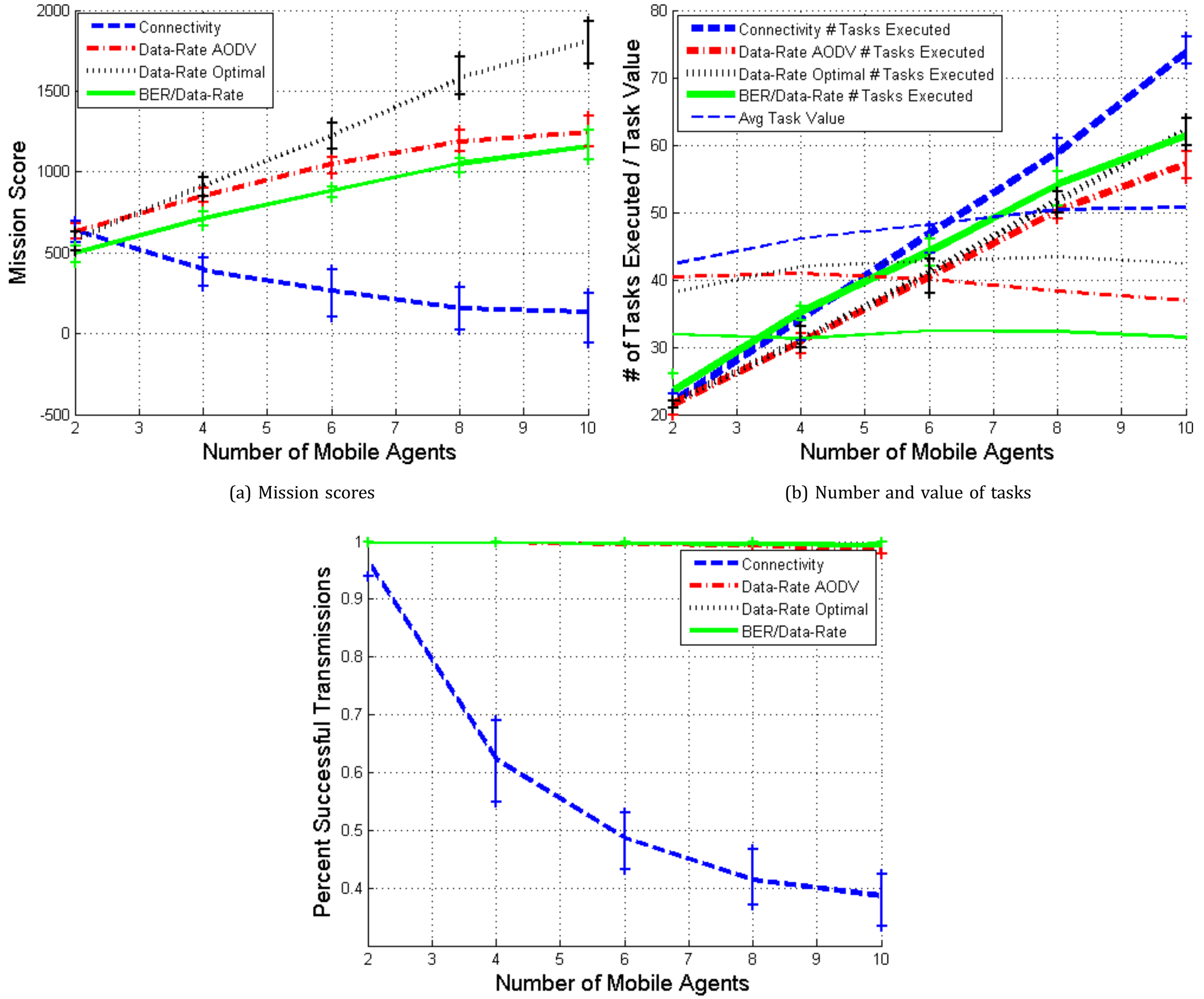
Although data-rates are better supported when the planning algorithm specifically takes it into account, Fig. 7(c) reveals a limitation of this control framework. Since network prediction is performed at discrete task execution times, as opposed to continuous time, the network is vulnerable to changes in routing caused by agents not active during this prediction (such as agents traveling to tasks). While network supportability and plans are still good, with 98.4% average network support in the **worst** case, perfect performance guarantees are not possible with this framework and real-world protocols.

Finally, while not plotted here, the experiment showed the performance of the system maintaining inter-agent connectivity. Teams not considering inter-agent BER requirements spend significant amounts of time disconnected from one another during the mission. On the other hand, if BER is considered as a requirement in addition to connectivity or data-rate alone, and inactive or traveling agents plan their motion properly, the team can maintain good inter-connectivity performance (98.2% in the worst case for small teams, and 99.95% with larger teams).

5.2. *Simulation — uncertainty and adaptive strategy*

Another simulation was performed to evaluate the ability of agents to estimate unknown and possibly changing communication parameters and adapt the planning strategy accordingly (see Sec. 4). Two sets of trials are presented: (1) the communication environment improves half way through the mission (α, σ_{dB}, N_0 decrease 10%), and (2) the communication environment instead degrades (α, σ_{dB}, N_0 increase 10%). Each trial is executed with a team of six UAVs unaware of the communication parameters, planning for data-rate using the AODV routing model.

Figure 8 shows an averaged mission execution profile with three settings: (1) agents have perfect up-to-date knowledge of the communication parameters (black line), (2) agents estimate parameters (blue line), and (3) agents have perfect knowledge of initial parameters but are unaware of the change (red line). The wavy shape of the reward profile is due to the 60 s replan cycle, when agents tend to travel to new tasks temporarily decreasing the score. Results clearly show the ability of the team to estimate changes in the environment and adapt the planning strategy to improve performance. In Fig. 8(a) the communication environment improves at 300 s allowing agents to achieve more tasks of higher reward. Teams which do not adapt to this setting (red) maintain the original strategy and marginally improve the rate of reward gain because of more successful communication rates. On the other hand, teams equipped with perfect knowledge (black) adapt their planning to be more ambitious which generates increased reward. Teams estimating these parameters (blue) closely follow the trends obtained with perfect knowledge with a short lag due to estimation and the planning cycle, and clearly outperform teams which do not adapt. Similarly in Fig. 8(b) the communication environment degrades at 300 s. Teams which do not adapt degrade in performance because of increased network violations, whereas teams with perfect knowledge propose more conservative plans and are able to keep generating reward. Once again, the teams estimating these parameters are able to closely follow the perfect knowledge case with a short lag, and outperform teams which do not adapt.



(a) Mission scores

(b) Number and value of tasks

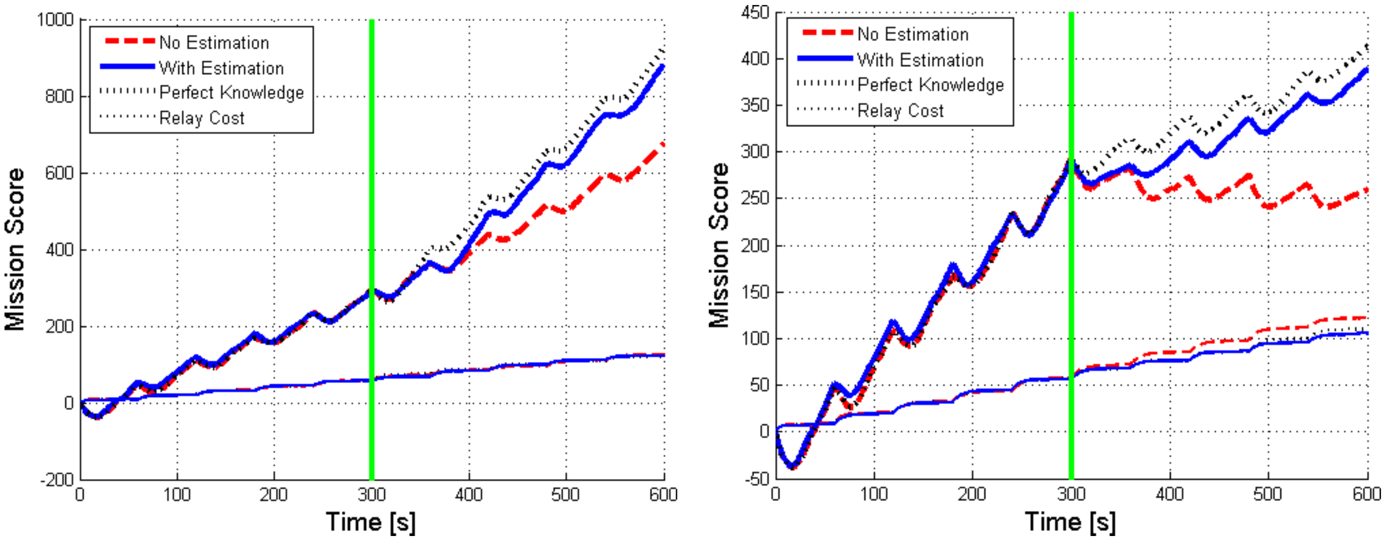
(c) Data-rate support success rate

Fig. 7. (Color online) Simulation experiment results.

5.3. *Outdoor flight testing — comparing planning strategies*

In order to demonstrate the algorithm in a more operationally realistic environment a field experiment was designed to show the capability of CBBA with relays in outdoor flight testing. There are several challenges that ultimately benefits to implementing a distributed multi-agent system in an outdoor uncontrolled environment: (1) it demonstrates real-time functionality of the system, (2) the system is subjected to nonlinear dynamics not considered in simulation due to imperfect modeling and environmental effects, and (3) it evaluates the robustness of the framework to system degradations and uncertainties which can significantly hinder the system from behaving as intended.

A similar scenario as used in the simulation experiments was employed in these flight experiments with a team of three Ascending Technologies Quadrotor UAVs (see Fig. 9). Three missions were executed to evaluate the different planning strategies described above: considering (1) connectivity only, (2) data-rate requirements, and (3) joint data-rate and BER requirements. The theoretical LP based optimal routing algorithm was not exercised in flight test



(a) Parameters improve 10% at 300 s

(b) Parameters degrade 10% at 300 s

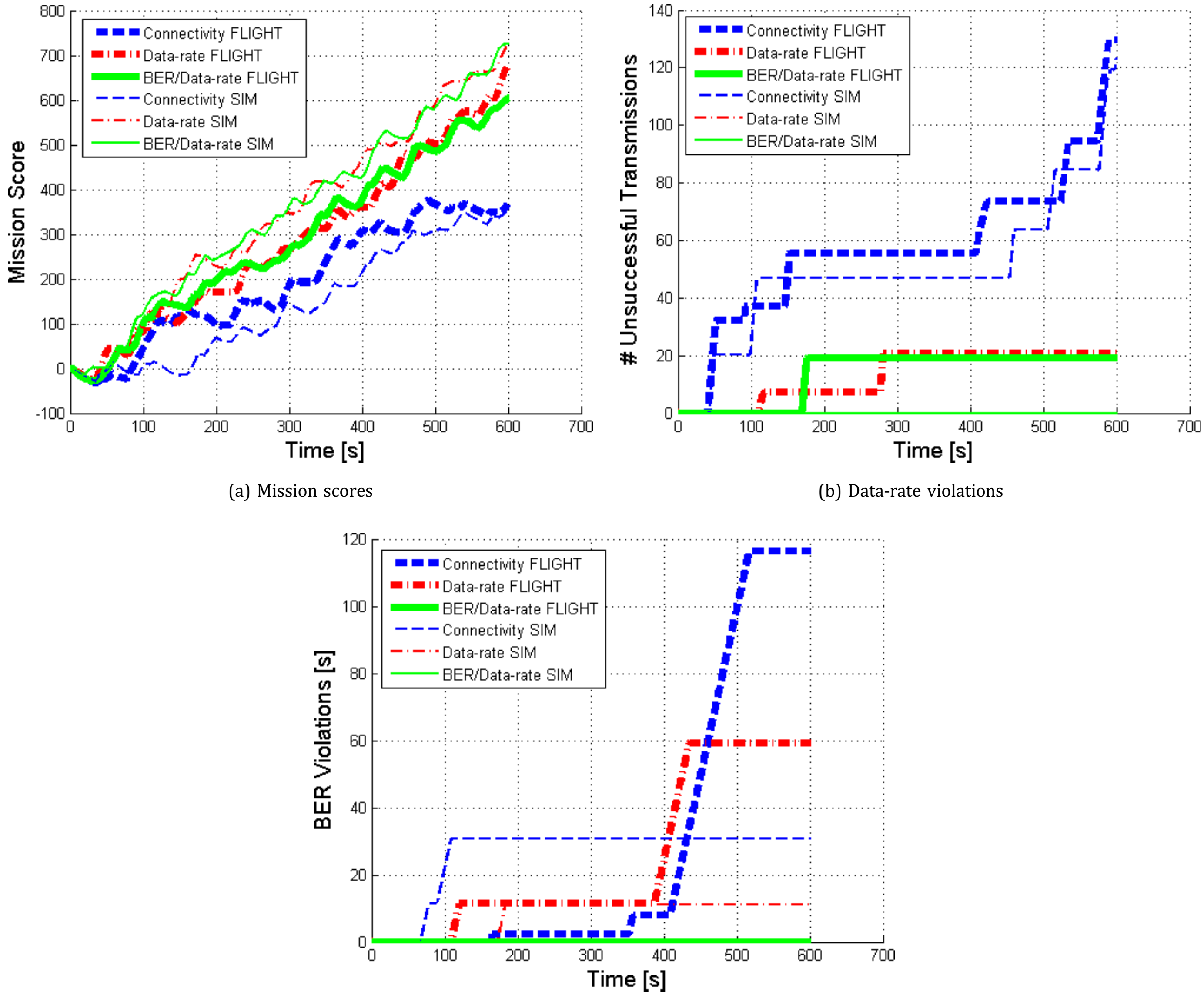
Fig. 8. (Color online) Adaptive planning to changes in the communication environment.

Fig. 9. Quadrotor UAVs executing flight test.

due to its slower runtime. Flight testing was conducted in restricted airspace. Interested readers are referred to our previous work for a description of the hardware implementation architecture (see [44]). The communication network in this flight test is simulated using the model presented in Sec. 2.

Results in Fig. 10 compare the planning strategies for the three flights. All three strategies are also compared to their respective performance predicted in simulation. Results first show that the two strategies planning to meet data-rate requirements outperform the original CBBA with Relays algorithm considering connectivity only which results in frequent link capacity overloads and missed reward. In addition, the green line in Fig. 10(c) indicates that the framework successfully maintains interconnectivity between nodes through relay placement and using motion control, as opposed to the two other strategies.

Interestingly though, results in Fig. 10(b) reveal three instances of overloaded capacity using the two strategies which specifically plan for this requirement (each step up in the graph represents a separate event) which were not predicted in simulation. In all three instances, an agent traveled slower than predicted due to head wind and arrived at its task late. In each case the task was still in its window of validity and agents started late anyways. Shortly after, other agents start other tasks which had not considered the late task still being active during network prediction in planning. This led to the three unpredicted



(a) Mission scores

(b) Data-rate violations

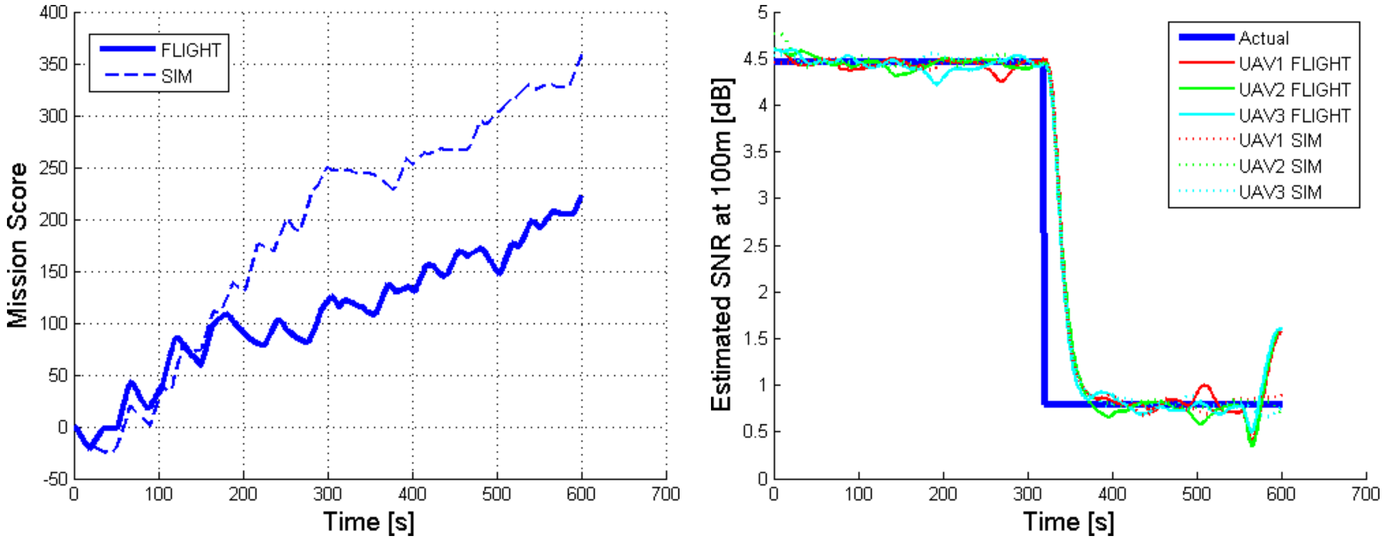
(c) BER violations

Fig. 10. (Color online) Outdoor flight network control performance comparison using three different planning strategies.

overloads which were directly responsible for reducing the mission score in data-rate and joint BER/data-rate planning strategies compared to their simulated runs. Different solutions are proposed to fix this issue, but need further evaluation: (1) the planner could use more conservative velocities in planning, (2) network predictions could include time window buffers to account for possible task delays, and (3) agents could be forced to stop executing or not execute a task and stay put in the event of a late arrival. Overall though, the results show the algorithm generally works well and as intended, and improves the performance of the multi-UAV team subject to more realistic communication requirements.

5.4. *Outdoor flight testing — uncertainty and adaptive planning*

A similar flight test experiment was conducted to evaluate the ability of agents to estimate unknown communication parameters, plan under uncertainty, and adapt to changes in the environment. Figure 11(a) shows a noticeable mission score performance difference between simulation and flight test. The difference can be attributed again to differences in predicted versus actual arrival times at the tasks. In this experiment, agents experienced significant delays due to wind, and arrived late to six of the tasks during the mission. On two occasions, the delays were enough to make the task



(a) Mission score

(b) SNR (γ_{dB}) at 100 m

Fig. 11. (Color online) Uncertainty and adaptive planning in flight test.

invalid and therefore, force assigned agents to skip on to the next item on their list. As such, UAVs incurred cost for a plan they could not realize in time.

On the other hand, Fig. 11(b) indicates that the UAVs were successfully able to estimate the communication parameters. They detect the sudden 10% parameter degradation in the environment at 320 s in the mission, and refine their predicted SNR values used in planning within 52 s. A significant erroneous deviation in estimates can be seen during flight test at 550 s. Here, two UAVs became positioned very close to one another due to randomly generated and assigned tasks. This small distance created a singularity in the SNR samples generated by the small team which led to the degraded estimate. In future implementations, these values can be identified and filtered out prior to the estimation.

6. Conclusion

The framework proposed in this paper enables teams of multiple UAVs to cooperatively execute complex scenarios using dynamic mission planning to support communication requirements. The distributed algorithm uses task assignment information, including task location and proposed execution time, to predict the network topology and plan support using relays. By explicitly coupling the task assignment and relay creation processes the team is able to optimize the use of agents to address the needs of dynamic complex missions. This also ensures that the resulting network topology and information routing support data-rate and BER required for the system to be effective. It also enables the team to estimate the expected performance of the communication network, detect changes in performance due to the operating environment, and adapt the planning strategy accordingly.

Future work will include incorporating robust planning strategies being investigated in [45] to plan for the uncertainty in vehicle dynamics which led to network violations in flight testing. In addition, a next step needed in this work is to validate the performance of the system in flight test with actual networking modules instead of a simulated network.

Acknowledgments

This research acknowledges support from Aurora Flight Sciences and Mike Klinker during field testing, and was financially supported by the Aurora Fellowship program.

References

[1] N. Jodeh and M. Mears, An overview of the cooperative operations in urban terrain (counter) program, in *AIAA Guidance, Navigation, and Control Conf.* (*GNC*), August 2008.

[2] M. Schwager, N. Michael, V. Kumar and D. Rus, Time scales and stability in networked multi-robot systits, in *IEEE Int. Conf. Robotics and Automation* (*ICRA*), May 2011.

[3] S. S. Ponda, J. Redding, H.-L. Choi, J. P. How, M. A. Vavrina and J. Vian, Decentralized planning for complex missions with dynamic communication constraints, in *American Control Conf.* (*ACC*), Baltimore, MD, July 2010.

[4] A. Goldsmith, *Wireless Communications* (Cambridge University Press, New York, NY, 2005).

[5] C. Dixon and E. W. Frew, Maintaining optimal communication chains in robotic sensor networks using mobility control, *Mobi. Net. Appl.* **14** (3) (2009) 281–291.

[6] N. Bezzo and R. Fierro, Decentralized connectivity and user localization via wireless robotic networks, in *IEEE Globecom*, Decitber 2011.

[7] N. Godditeier, S. Rohde, J. Pojda and C. Wietfeld, Evaluation of potential fields mobility strategies for aerial network provisioning, in *IEEE Globecom*, Decitber 2011.

[8] M. A. Hsieh, A. Cowley, R. V. Kumar and C. J. Taylor, Maintaining network connectivity and performance in robot teams, *IEEE J.* **25** (1–2) (2008) 111–131.

[9] Y. Yan and Y. Mostofi, Robotic router formation — a bit error rate approach, in *IEEE Military Communications Conf.* 2010.

[10] R. Olfati-Saber, J. Alex Fax and R. M. Murray, Consensus and cooperation in networked multi-agent systems, *IEEE Proc.* **95**(1) (2007) 215–233.

[11] E. Craparo, J. P. How and E. Modiano, Throughput optimization in mobile backbone networks, in *SIAM Conf. Optimization*, May 2008.

[12] S. S. Ponda, L. B. Johnson, A. N. Kopeikin, H.-L. Choi and J. P. How, Distributed planning strategies to ensure network connectivity for dynamic heterogeneous teams, *IEEE J. Selec. Areas Commun.* **30**(5) (2012) 861–869.

[13] Y. Mostofi, A. Gonzales Ruiz, A. Ghaffarkhah and D. Li, Characterization and modeling of wireless channels for networked robotic and control system — a comprehensive overview, in *IEEE Int. Conf. Intelligent Robots and System* (2009).

[14] P. Santi, *Topology Control in Wireless Ad Hoc and Sensor Networks* (John Wiley and Sons, Chichester, U.K., 2005).

[15] E. Modiano, 16.36: Communication systits and networks, Lecture Notes (Massachusetts Institute of Technology, Cambridge, MA, 2012).

[16] C. Perkins and E. Belding-Royer, *Ad-hoc* on-ditand distance vector (aodv) routing, in *IEEE Workshop on Mobile Computing System*, February 1999.

[17] T. Salonidis, M. Garetto, A. Saha and E. Knightly, Identifying high throughput paths in 802.11 mesh networks: A model-based approach, in *IEEE Int. Conf. Network Protocols*, Octoboer 2007.

[18] A. S. Clare, M. L. Cummings and L. F. Bertuccelli, Identifying suitable algorithms for human-computer collaborative scheduling of multiple unmanned vehicles, in *Accepted at AIAA Aerospace Sciences Meeting* (2012).

[19] S. S. Ponda, L. B. Johnson, A. Geramifard and J. P. How, in *Handbook of Unmanned Aerial Vehicles*, Cooperative mission planning for multi-UAV teams, in (Springer, 2012) (to appear).

[20] S. Karaman, M. Faied, E. Frazzoli and A. Girard, Specification and planning of interactive UAV missions in adversarial environments, in *AIAA Guidance, Navigation, and Control Conf.* (*GNC*) (2009).

[21] J. Boskovic, N. Knoebel, N. Moshtagh, J. Amin and G. Larson, Collaborative mission planning and autonomous control technology (compact) system employing swarms of UAVs, in *AIAA Guidance, Navigation, and Control Conf.* (*GNC*) (2009).

[22] O. Toupet and J. How, Collaborative sensor fusion and managitent for multiple UAVs, in *AIAA Infotech @ Aerospace Conf.* March 2011. (AIAA-2011-1488).

[23] J. Wood and J. K. Hedrick, Space partitioning and classification for multi-target search and tracking by heterogeneous unmanned aerial system teams, in *AIAA Infotech @ Aerospace Conf.* (2011).

[24] W. Ren and R. W. Beard, Consensus seeking in multiagent systits under dynamically changing interaction topologies, *IEEE Trans. Auto. Control* **50**(5) (2005) 655–661.

[25] R. Olfati-Saber and R. M. Murray, Consensus problits in networks of agents with switching topology and time-delays, *IEEE Trans. Auto. Control* **49**(9) (2004) 1520–1533.

[26] M. Alighanbari, L. F. Bertuccelli and J. P. How, A robust approach to the UAV task assignment problem, in *IEEE Conf. Decision and Control* (*CDC*), 13–15 December 2006, pp. 5935–5940.

[27] H. -L. Choi, L. Brunet and J. P. How, Consensus-based decentralized auctions for robust task allocation, *IEEE Trans. Robotics* **25**(4) (2009) 912–926.

[28] A. N. Kopeikin, A. Clare, O. Toupet, J. P. How and M. L. Cummings, Flight testing a heterogeneous multi-uav systit with human supervision, in *AIAA Guidance, Navigation, and Control Conf.* (*GNC*), August 2012.

[29] C. Marshall, M. Mears and S. Rasmussen, 2010 ice-t cooperative control flight testing, in *AIAA Infotech @ Aerospace Conference*, April 2011.

[30] F. Kaabi and F. Filali, Channel allocation and routing in wireless mesh networks: A survey and qualitative comparison between schites, *Int. J. Wireless and Mobile Networks* **2**(1) (2010) 104–111.

[31] G. R. Hiertz, D. Denteneer, S. Max, R. Taori, J. Cardona, L. Berlitann and B. Walke, Ieee 802.11s: The wlan mesh standard, *IEEE Wireless Commun.* **2** (2010) 104–111.

[32] C. Perkins and E. Belding-Royer, Dynamic source routing in *ad-hoc* wireless networks (Kluwer Academic, 1996).

[33] C. Adjih, A. Laouiti, P. Minet, P. Muhlethaler, A. Qayyum and L. Viennot, The optimised routing protocol for mobile *ad-hoc* networks: protocol specification — Report n5145, Technical report, Institut National de Recherche en Informatique et en Automatique, France (2004).

[34] R. G. Garroppo, S. Giordano and L. Tavanti, Experimental evaluation of two open source solutions for wireless mesh routing at layer two, in *Int. Symp. Wireless Pervasive Computing* (2010).

[35] D. Decouto, D. Aguayo, J. Bicket and R. Morris, A high-throughput path metric for multi-hop wireless routing, *Wire. Net.* Springer Science and Business Media Inc. **11** (2005) 419–434.

[36] A. Ribeiro, Notes on implitenting a ieee 802.11s mesh point, in *Euro NGI Workshop, Barcelona, Spain*, January 2008.

[37] D. Bertsimas and J. N. Tsitsiklis, *Introduction to Linear Optimization* (Athena Scientific, Belmont, MA, 1997).

[38] S. Toumpis and S. Gitzenis, in *IEEE INFOCOM*, April 2009.

[39] L. Xiao, M. Johansson and S. P. Boyd, Simultaneous routing and resource allocation via dual decomposition, *IEEE Trans. Commun.* **52** (7) (2004) 1136–1144.

[40] R. L. Cruz and A. V. Santhanam, Optimal routing, link scheduling and power control in multihop wireless networks, in *IEEE INFOCOM*, April 2003.

[41] C. Dixon, *Controlled Mobility of Unmanned Aircraft Chains to Optimize Network Capacity in Realistic Communication Environments*, PhD Thesis, University of Colorado at Boulder, Boulder, CO (2010).

[42] D. Tardioli, A. R. Mosteo, L. Riazuelo, J. L. Villarroel and L. Montano, Enforcing network connectivity in robot team missions, *The Int. J. Robot. Res.* **29**(4) (2010) 460–480.

[43] N. Wagle and E. Frew, Spatio-titporal characterization of airborne radio frequency environments, in *IEEE Globecom*, Decitber 2011.

[44] A. N. Kopeikin, S. S. Ponda, L. B. Johnson, O. Toupet and J. P. How, Real-time dynamic planning to maintain network connectivity in a team of heterogeneous unmanned systits, in *Wi-UAV 2011, 2nd Int. Workshop on Wireless Networking for Unmanned Autonomous Vehicles at the IEEE GlobeComm Conf.* December 2011.

[45] S. S. Ponda, L. B. Johnson and J. P. How, Distributed chance-constrained task allocation for autonomous multi-agent teams, in *American Control Conf.* (*ACC*), June 2012.