

SYNCHRONOUS AND ASYNCHRONOUS COMMUNICATION EFFECTS ON THE COOPERATIVE CONTROL OF UNINHABITED AERIAL VEHICLES*

Jason W. Mitchell[†]
Emergent Space Technologies, Inc.
Greenbelt, MD 20770-6334

Steven J. Rasmussen[‡]
General Dynamics AIES
Wright-Patterson AFB, OH 45433-7531

Joshua D. Redding[§]
Mechanical Engineering
Brigham Young University
Provo, UT 84602

Phillip R. Chandler[¶]
Control Sciences Division
Air Force Research Laboratory (AFRL/VACA)
Wright-Patterson AFB, OH 45433-7531

Abstract

Synchronous and asynchronous communication effects are considered in the cooperative control of uninhabited aerial vehicles where resource allocation is performed by an iterative network flow. We briefly outline both the single and iterative network flow assignment algorithms and their communication requirements. Then, using the abstracted communication framework found in the Air Force Research Laboratory MultiUAV simulator, a model is constructed to investigate the effects of fully synchronous or a limited form of asynchronous communication, in the form of message loss, on mission performance. These effects are observed in a sequence of vehicle-target scenarios where the task assignment is implemented as a redundant, centralized optimization.

Keywords: cooperative control, uninhabited aerial vehicles, communication delay, synchronous, asynchronous, AFRL/VA MultiUAV.

Introduction

COORDINATION and cooperation between uninhabited aerial vehicles (UAVs) has the potential to significantly improve their effectiveness in many situations. For the typical tasks that these vehicles must perform, i.e. search, detection, classification, attack, and verification, explicit vehicle cooperation may be required to meet specific objectives. Thus, the ability to communicate information between vehicles becomes essential and provides an opportunity to enhance overall capability.

While vehicle communications provide the opportunity to enhance performance, it is not without cost. Frequently, control algorithms are designed without regard to their associated communication needs or effects. For the control system designer, such treatment is undertaken to reduce algorithmic complexity and obtain a manageable result. Consequently, communication constraints and their effect on the control algorithms are quantified a posteriori. As an example of this design strategy, consider two previously studied approaches that quickly produce near-optimal single task assignments,^{1,2} and more recently, the near-optimal assignment of a sequence of tasks using an iterative network flow model.³ These studies did not specifically consider communication constraints or the potential effect on the performance of the cooperative control algorithms. One recent study has investigated fixed communication delays using MultiUAV2,⁴ however it was of limited scope and did not include fully synchronized but delayed communication, Mitchell et al.⁵

In this work, synchronous and asynchronous communication effects are considered in the cooperative control of UAVs with resource allocation performed by an iterative network flow. Specifically, we use a Monte-Carlo approach to investigate the effect of communication delays on mission performance for the two differing modes of decision operation, viz. fully synchronous and limited asynchronous. In the first mode, the vehicles are required to have a fully synchronized perception of their environment before making decisions. In the second mode, planning decisions are made after waiting a specified time-out period for communicated responses, while vehicles are subject to *message loss*. The resulting effects are observed

*This material is declared a work of the U.S. Government and is not subject to copyright protection in the United States.

[†]AIAA member, Aerospace Scientist.

[‡]AIAA member, Senior Aerospace Engineer.

[§]Graduate Student.

[¶]AIAA member, Senior Aerospace Engineer.

in a sequence of vehicle-target scenarios where the task assignment is implemented as a redundant, centralized optimization (RCO). We compare the performance of the two operational modes for the vehicle-target scenarios based on the following averages: task transition time, number of kills and verifies completed per mission.

Initially, we outline the single and iterative network flow assignment algorithms and their communication requirements. Then, we briefly describe the MultiUAV2 simulator,^{6,7} and its framework that models vehicle-to-vehicle communication.

Background

We begin with a short description of MultiUAV2 including the general mission scenario architecture. This is followed by a succinct outline of the network flow task allocation algorithm, and its communication messaging requirements.

Simulation Framework

The MultiUAV2 simulator⁷ is capable of simulating multiple UAVs which cooperate to accomplish a predefined mission. Individually, the vehicles are capable of searching for, recognizing, attacking, and verifying targets.

The purpose of the simulator is to provide an environment that researchers can use to implement and analyze cooperative control algorithms. The simulation is built using a hierarchical decomposition where inter-vehicle communication is explicitly modeled. MultiUAV2 includes visualization tools and provides links to external programs for post-processing analysis. Each of the vehicle simulations include six-degree-of-freedom dynamics and embedded flight software (EFS). The EFS consists of a collection of *managers* or *agents* that control situational awareness and responses of the vehicles. In addition, the vehicle model includes an autopilot that provides waypoint navigation capability. With the original release, MultiUAV⁶ could simulate a maximum of eight (8) vehicles and ten (10) targets, however recent work eases the previous burden of extending these limits. The individual managers contained within the vehicles include: Tactical Maneuvering, Sensor, Target, Cooperation, Route, and Weapons. At the top level, these managers are coded as SIMULINK models, with supporting code written in both MATLAB script and C++.

General Scenario

Let us consider a set of N simultaneously deployed vehicles indexed by $i \in \mathbb{Z}[1, N]$. The targets are categorized according to the value associated with their destruction. The individual targets are indexed by j as they are found via search, so that we find $j \in \mathbb{Z}[1, M]$ with V_j as the value of target j . The vehicles are provided no precise a priori information about the total number of targets or their initial locations. Potential target information is obtained by the vehicles via either on-board Automatic Target Recognition (ATR) methodologies or communicated information from other off-board sensor platform simulations. Once successfully classified as a target, the attack vehicle is selected. Upon reaching the selected target, the vehicle releases its munition and is subsequently declared an unavailable asset, i.e. attack is a terminal task. Finally, the selected target must be verified as destroyed to complete the target specific task chain.

Throughout the simulation, at each target state change or task failure, a resource allocation algorithm is executed to compute task assignments. The resulting assignment is sub-optimal. However, Rasmussen et al.⁸ have shown that these assignments are typically near-optimal in an average sense.

Models

The essential models used in the MultiUAV2 simulation are described in the following.

Target Recognition

The on-board ATR process is modeled using a system that provides a probability that the target has been correctly classified. The probability of a successful classification is based on the viewing angle of the vehicle relative to the target, Rasmussen et al.⁷ For this exercise, the possibility of incorrect identification is not modeled, however targets are not attacked unless a 90% probability of correct identification is estimated. Further details of the ATR methodology can be found in Chandler and Pachter,⁹ with a detailed discussion available in Chandler and Pachter.¹⁰

In addition to the on-board ATR process, information potential target information can be communicated to the vehicles from other off-board sensor simulations. One example of an off-board

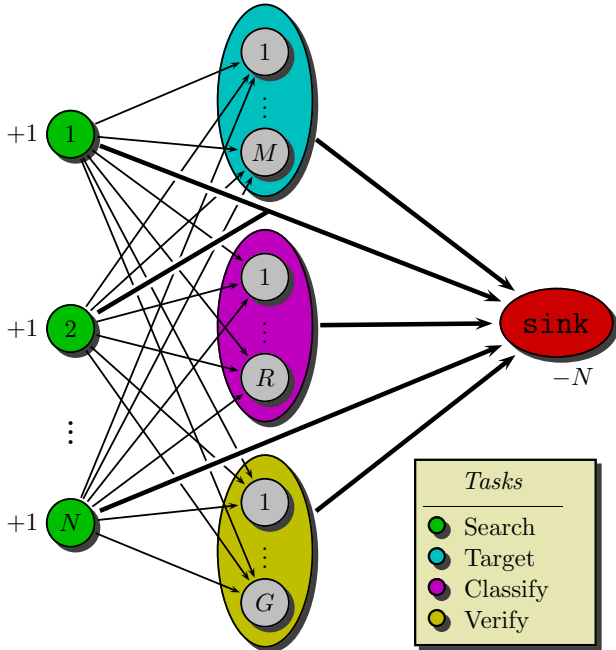


Fig. 1 Network flow diagram.

sensor scenario could be an aircraft loitering at some higher altitude and using a different suite of sensors, e.g. GlobalHawk¹¹ or SensorCraft.¹²

Task Allocation by Network Optimization

The weapon system allocation is treated as follows: individual vehicles are discrete, single unit supplies executing tasks that correspond to flows on arcs through the network, with the ultimate disposition of the vehicles representing the demand. Thus, the flows are zero (0) or one (1). We assume that each vehicle operates independently, and makes decisions when new information is received. These decisions are determined by the solution of the network optimization model. The receipt of new target information triggers the formulation and solving of a new optimization problem that reflects current conditions, thus achieving feedback action. At any point in time, the database on-board each vehicle contains a *target* set, consisting of indices, types and locations for targets that have been classified above the probability threshold. There is also a *speculative* set, consisting of indices, types and locations for potential targets that have been detected, but are classified below the probability threshold and thus require further inspection before striking.

The network flow model, seen in Fig. 1, is demand driven. The sink node at the right exerts a demand-pull of N units, causing the nodes on

the left to flow through the network. In the middle layer, the top M nodes represent all of the successfully classified targets, and thus are ready to be attacked. An arc exists from a specific vehicle node to a target node if and only if it is a feasible vehicle/target pair. At a minimum, the feasibility requirement would mean that there is sufficient fuel remaining to strike the target if so tasked. Other feasibility conditions could also be considered, e.g. heterogeneous weapons or sensing platforms, poor look-angles. The center R nodes of the middle layer represent potential targets that have been detected, but do not meet the minimum classification probability. We call them *speculatives*. The minimum feasibility requirement to connect a vehicle/speculative pair is sufficient fuel for the vehicle to deploy its sensor to elevate the classification probability. The lower-tier G nodes model alternatives for verification of targets that have been struck. Finally, each node in the vehicle set on the left has a direct arc to the far right node labeled sink , modeling the option of continuing to search. The capacities on the arcs from the target and speculative sets are fixed at one (1). From the integrality property, flow values are constrained to be either zero (0) or one (1). Each unit of flow along an arc has a benefit which is an expected future value. The optimal solution maximizes total value. For a more detailed discussion, including the issue of the *benefit calculation*, see Schumacher et al.³

Single Pass Network Flow

Single task assignment in MultiUAV² is formulated as the capacitated transshipment problem (CTP).¹ Due to the special structure of the problem, there will be an entirely integer optimal solution.² Thus, solutions to this problem pose a small computational burden, making it feasible for implementation on the processors likely to be available on inexpensive UAVs.

Iterative Network Flow

Due to the integrality property, it is not normally possible to simultaneously assign multiple vehicles to a single target, or multiple targets to a single vehicle. However, using the network assignment iteratively, *tours* of multiple assignments can be generated.³ This is done by solving the initial assignment problem once, and only finalizing the assignment with the shortest estimated

arrival time. The assignment problem can then be updated assuming that assignment is performed, updating target and vehicle states, and running the assignment again. This iteration can be repeated until all of the vehicles have been assigned terminal tasks, or until all of the target assignments have been fully distributed. The target assignments are complete when classification, attack, and verification tasks have been assigned for all known targets. Assignments must be recomputed if a new target is found or a vehicle fails to complete an assigned task.

Information Requirements

The implementation of the task allocation algorithms outlined above requires communication of information between vehicles. Since both algorithms discussed here make use of network flow, the necessary information is common between them. The overarching optimization problem can be characterized as both centralized and redundant, i.e. each vehicle computes its own network flow.

Momentarily disregarding communication issues, the RCO problem requires a synchronized database of target and vehicle state information. With this, each vehicle computes the benefits for the arcs in the network, and solves an optimization problem to maximize the total benefit. From Mitchell et al.,¹³ the MultiUAV2 network flow implementation requires, in general, the following communicated information: ATR data; target and vehicle positions; target, vehicle, and task status; and vehicle trajectory waypoints.

Communication

The communication simulation used in this work is very similar to that used in Mitchell et al.,^{5,13} except that messages are subject to additional effects in the simulated vehicle communication at each major model update. At the present time, the major model update occurs at 10 Hz. This fairly coarse grained update is necessary to maintain a reasonable run-time for individual scenarios to complete, in a larger Monte-Carlo sense, on a desktop/personal computer. The minor model update, which controls the vehicle dynamics and other underlying subsystems, is scheduled at 100 Hz.

The *broadcast* communication model is implicitly assumed for the vehicle communication.

While not specifically targeted to address a particular physical implementation, such a choice encompasses the notions of time-division multiple access (TDMA) or time-division multiplexing (TDM) communication, e.g. Link-16¹⁴ or Link-11, respectively.

General Effects

In general, each vehicle waits two (2) major model updates to collect vehicle responses before solving a network flow problem to determine task assignments^{||}. This is taken to be a representative combination of both communication and processing delay in an ideal environment.

For the case of fully synchronized operation, the team has a single *bad actor* as a member whose delay is fixed for a given scenario. As a result, the team must wait before making decisions and executing a plan. This approach ensures that, in the presence of finite delays and no information loss, all known tasks will be completed, however the total mission time must grow accordingly. This penalty of increased mission time may or may not be acceptable in terms of overall performance and depends strongly on the total delay incurred.

In the case of asynchronous operation, the team again contains a single bad actor that is subject to degraded communication quality. Specifically, the bad actor drops certain incoming messages. As in the nominal simulation mode, the cooperating vehicles will only wait for the prescribed number of major model updates before planning their task assignments and executing the resulting plan. As a consequence, the bad actor may not be included in the new plans of the remaining vehicles. Likewise, the bad actor may not include other vehicles in its plan. Consequently, the vehicles' differing perceptions of their environment make it difficult to guarantee that all tasks will be completed. In addition, we are unable to guarantee that redundant tasks will not occur, increasing the likelihood of inefficient resource usage.

Effects Based Modeling

In previous releases of MultiUAV, each vehicle's Cooperation Manager provided for the accumulation of planning information based on simple delay timer implemented as a discrete unit-nominal transfer-function of the desired degree.

^{||}This delay is necessary to break algebraic-loops in the SIMULINK portions of MultiUAV2.

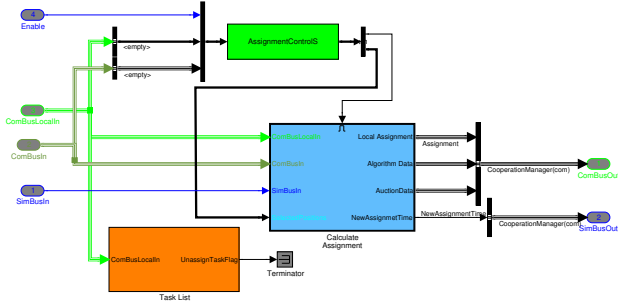


Fig. 2 Cooperation Manager subsystem for Multi-UAV².

Thus, the timer values could be set as desired to implicitly introduce communication or processing delays. This implementation, however, failed to meet more strenuous requirements on the Cooperation Manager levied by other growing simulation needs, e.g. unified treatment of assignment algorithms, auction control, grouped synchronous behaviours, etc.

Synchronized Perception

To meet the requirements mentioned above, a more flexible component was developed, called **AssignmentControls**, and currently exists as a MATLAB s-function that is responsible for managing more advanced planning management such as message invoked planning, auction coordination, and the traditional timer controlled planning; Fig. 2. The algorithm uses vehicle properties associated with queuing delay, assignment processing delay, and communication delay to decide when to plan, and when to execute the computed plan.

Message Loss

The asynchronous communication component implemented for this study focused on a general effects based method of simulating message loss due to hardware/software malfunction, antenna masking, unfriendly electromagnetic environment, weather, etc. In addition to the vehicle properties mentioned above for synchronized perception, we add a property representing a vehicle's incoming communication link quality. This link quality is treated as a percentage loss for incoming messages. Since this property is local to each vehicle, it can be manipulated throughout the simulation to represent communication link quality in response to changes in the vehicle's environment.

Simulation

In this work, we investigate the effect of requiring fully synchronous communication or allowing limited asynchronous operation, in the form of message loss, on the performance of cooperating UAVs that use a iterative network flow of depth three (3) for task allocation. To study this, a Monte-Carlo approach is taken, consisting of fifty (50) individual simulations,⁸ each with a maximum mission time of $t_f = 200$ s.

Individual scenarios are composed of four (4) vehicles with three (3) targets distributed inside an area of responsibility (AoR) of approximately 16 mi². The vehicle properties are: constant velocity of 370 ft/s or approximately mach 0.33, constant altitude of 675 ft, minimum turn radius of 2000 ft, rectangular sensor footprint of approximately w:2000 ft × d:600 ft, trailing edge sensor stand-off of approximately 2500 ft, and sufficient fuel for the required search operation. The ATR sensors are subject to a roll limit of 30°.

Since search is not the focus of this study, vehicles begin in a line formation, and initially follow a preprogrammed *zamboni race* search pattern through the AoR. The targets are uniformly distributed throughout the domain and oriented with uniformly random pose-angles.

A total of nine (9) Monte-Carlo scenarios are compared. Their designation and bad actor properties are as follows:

- B_ℓ : baseline simulation, best link quality, no bad actors;
- S_0 : as B_ℓ , plan delay of 2 sec;
- S_1 : as B_ℓ , plan delay of 8 sec;
- S_2 : as B_ℓ , plan delay of 24 sec;
- D_0 : as B_ℓ , link quality at 0%;
- D_1 : as B_ℓ , link quality at 25%;
- D_2 : as B_ℓ , link quality at 50%;
- D_3 : as B_ℓ , link quality at 75%;
- D_4 : as B_ℓ , link quality at 95%.

In general, targets can take on the states: undetected, detected, classified, attacked, killed, verified. In the following, however, we are only interested in detected (\mathcal{D}), classified (\mathcal{C}), killed (\mathcal{K}), and verified (\mathcal{V}).

Results

The results of the simulations described in the previous sections are summarized in Figs. 3–7.

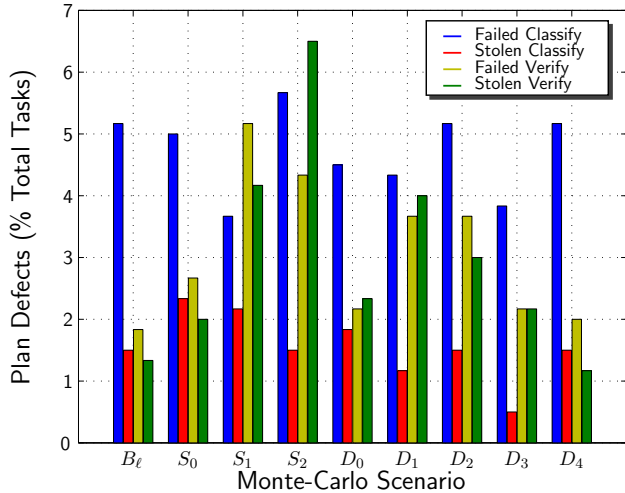


Fig. 3 Planning defects as a percentage of the total nominal tasks.

Planning Defects

Fig. 3 shows the number of planning defects occurring as a percentage of the nominal 600 total tasks per scenario, i.e. 4 tasks per target, 3 targets per mission, 50 missions per scenario. We distinguish between *failed* tasks and *stolen* tasks for both classification and verification.

Failed Tasks

A failed classification task indicates that a sensor pass on a known target failed to produce a combined ATR value of sufficient confidence to conclusively determine the target’s state. This is the result of viewing the target at a poor look-angle based on the ATR template. Individually, targets are oriented uniformly randomly and the resulting defects do not appear to be strongly correlated within or between the S and D scenarios.

While classification failures do not appear to be correlated, failed verifies display some scenario related non-uniform behaviour. At present, classification failures only occur if the predicted timing window is not met; currently verifications do not use an ATR template and any sensor pass will produce positive verification. We notice that for synchronized operation, there is a steady increase in failures with the exception of S_2 . This is related to the vehicle’s ability to reach the task in the time estimated using potentially stale position information about other vehicles. A rationale for the reduction in S_2 could be that as the synchronization time becomes a significant percentage of the time required to traverse the AoR, the vehicles

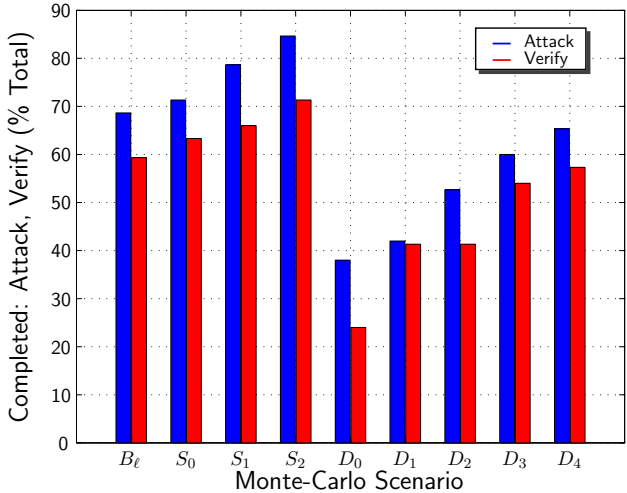


Fig. 4 Number of completed kills and verifies as percentage of total.

obtain a more complete picture of the battlespace, thus improving decisions. This invokes the gaming trade-off between the benefit derived from acting quickly with limited information versus delaying action to obtain additional information. A full consideration of such a trade-off is beyond the scope of this work. However, as Fig. 3 indicates, to reduce task failures, it is best to act as quickly as possible with the available information.

Stolen Tasks

A stolen task indicates that an unplanned sensor passed over a known target. Stolen classification defects occur when planning cannot react quickly enough to changes in the environment and is related to proximity of targets in time and space.¹⁵ In the S scenarios, we see that these defects decrease as the synchronization time increases. This is a consequence of moving away from targets clustered in space or time before executing a plan. In the D scenarios, there does not appear to be sufficient information to identify a trend or declare the result uniformly random with reasonable certainty.

Stolen verification tasks for the S scenarios show a significant relative increase as the synchronization time increases. It is not immediately clear why this occurs. One possible explanation could be that as the vehicles return to service targets, target clutter that would have been seen in stolen classifications is moved to verification. In the case of the D scenarios, there is a marked decrease in stolen verifications. Reliable information

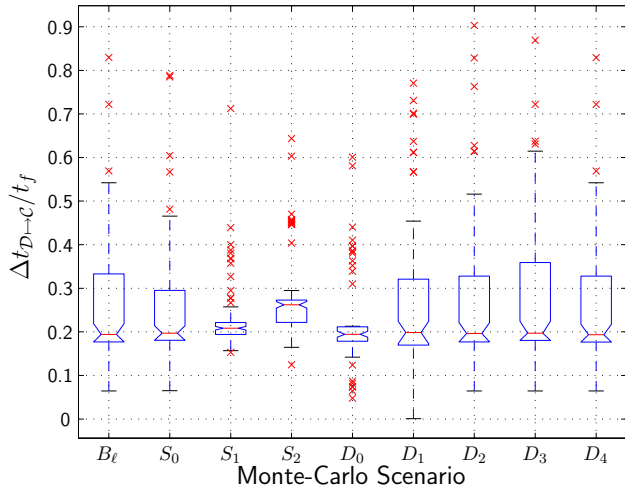


Fig. 5 Comparison of transition times: detect to classify.

about teammates provides for better cooperation, as is clearly demonstrated in Fig. 3.

Task Completion

The tasks of interest here are number of completed kill and number of completed verifies. This information is summarized in Fig. 4. We see that S and D scenarios, the synchronization delay and increased communication link quality, respectively, improves completed tasks. Scenario S_2 shows an approximately 15% improvements over the baseline scenario B_ℓ .

For D_0 , we have one bad actor that receives no information from his teammates, and we see that the mission performance is quite poor compared to B_ℓ . However, with D_3 , increasing the communication link quality to 75% produces a significant increase in completed tasks compared to D_0 . Finally, with D_4 , we see an approximately 25% increase in kill performance and nearly 30% improvement in completed verifies. These D scenarios clearly demonstrate improved efficiency cooperation provides in executing a mission, and underscore the need for reliable communication.

Task Transition Times

The task transition time statistics for $\mathcal{D} \mapsto \mathcal{C}$, Fig. 5; $\mathcal{C} \mapsto \mathcal{K}$, Fig. 6; $\mathcal{K} \mapsto \mathcal{V}$, Fig. 7. Each task transition time is expressed as a fraction of the maximum simulation time, $t_f = 200$ sec. The notched box plots have whiskers at the extreme values that are within $1.5 \times$ the interquartile range (IQR) of each box. This visualization provides a convenient graphical t -test for comparison of

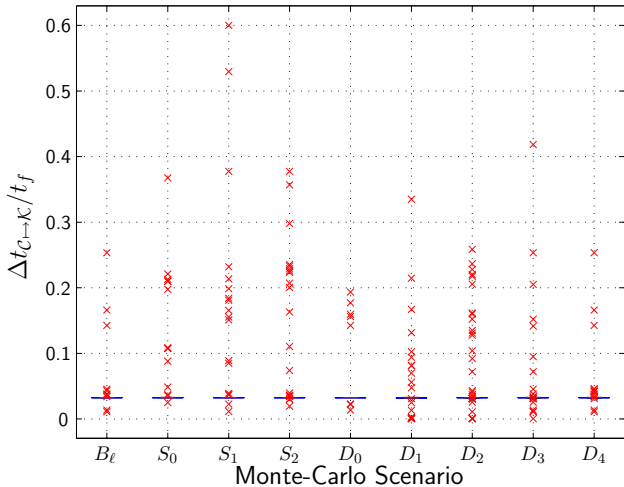


Fig. 6 Comparison of transition times: classify to kill.

the data distributions. Any common notch overlap between two boxes indicates that the medians agree with a 95% confidence level.

$\mathcal{D} \mapsto \mathcal{C}$:

From Fig. 5, we see that the medians for S_0 and S_1 are in agreement with B_ℓ , however, the median transition time for S_2 shows a measurable increase. S_2 also indicates a skew shift to reduced variability in the upper quartile. The median transition time for $\mathcal{D} \mapsto \mathcal{C}$ is significantly affected only in S_2 . While S_1 's median is slightly larger than for S_0 , it does not appear to be outside the confidence interval. This fact, along with a tighter IQR, indicates that S_1 provides the best choice to minimize this S scenario transition time.

The D scenarios are quite similar even when the communication link quality is zero. All the medians agree within the significance level, and the only variability appears in the IQR for D_0 . For D , we could presumably choose D_2 or D_4 , as they appear very similar. In fact, the D_i indicate that moderate communication to the bad actor is worse than even no communication. Of course, we prefer D_4 for the other benefits it can provide to cooperation. Initially, D_0 appears very attractive, but we must remember that, in this case, we are likely to have inefficient usage of resources as the bad actor may visit a target without regard to its teammates actions due to its lack of inbound communication.

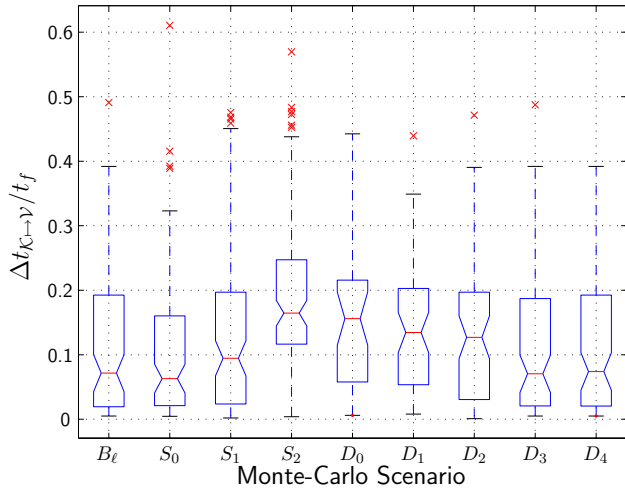


Fig. 7 Comparison of transition times: kill to verify.

$\mathcal{C} \mapsto \mathcal{K}$:

Transition times $\mathcal{C} \mapsto \mathcal{K}$ are shown in Fig. 6. It is immediately clear that this transition time is generally quite small in comparison to t_f , and is the result of the small trailing edge sensor stand-off distance. The medians agree across B_ℓ , S , and D scenarios. For both S and D scenarios, to reduce the maximum transition time, it is better to have less or more of synchronization delay or communication, respectively, than to have a moderate amount of either. As with the previous transition, we discount D_0 because of the lack of teammate consideration.

One interesting aspect to notice here is that our previous choice of S_1 for best $\mathcal{D} \mapsto \mathcal{C}$ median transition time is the worst choice of the S scenarios to reduce $\mathcal{C} \mapsto \mathcal{K}$ median transition time.

$\mathcal{K} \mapsto \mathcal{V}$:

Transition times for $\mathcal{K} \mapsto \mathcal{V}$ are given in Fig. 7. For S scenarios, the median comparison is similar to that of the $\mathcal{D} \mapsto \mathcal{C}$ transition time: only the median of S_2 differs significantly from the other S scenarios and B_ℓ . Given the median increase of S_2 and the reduced IQR of S_0 , the median S scenario $\mathcal{K} \mapsto \mathcal{V}$ transition time is best reduced by S_0 .

The D scenario transition times are clearly aided by the increased quality of communication. In this case, D_4 is the best choice to reduce the both median and maximum transition time.

Conclusions

In this work, we investigated the effects of synchronous and asynchronous communication on the cooperative control of uninhabited aerial vehicles where resource allocation was performed by an iterative network flow. We briefly outlined both the single and iterative network flow assignment algorithms and their communication requirements. Then, using the abstracted communication framework found in the Air Force Research Laboratory MultiUAV2 simulator, a model was constructed to investigate the effects of fully synchronous and a limited form of asynchronous communication, in the form of message loss, on mission performance. These effects were observed in a sequence of vehicle-target scenarios where the task assignment was implemented as a redundant, centralized optimization.

As this work confirms, it is difficult, in general, to optimize the performance of a single mission parameter without impacting others. In fact, favoring one mission parameter, e.g. minimum median transition time from detection to classification, may saddle the evaluator with worst case performance in another parameter; smallest maximum classify to kill transition time, continuing the previous example. To clarify these issues, we have shown that we can measure mission performance parameters in the context of evaluating cooperative control algorithms, and that the framework provided by MultiUAV2 permits overall mission goals and the performance of cooperative control algorithms to be analyzed together using multiple objectives.

In the case of synchronization delay, the extreme values of delay generally performed better in reducing median task transition times. The increased synchronization delay improved task completion, but produced mixed results in planning defects. For the single bad actor with varying communication link quality, more reliable communication provided reduced median task transition times, and clearly demonstrated the value of communication and cooperation in improving task completion efficiency. Unlike the synchronous case, more reliable communication produced less ambiguous results in reducing planning defects, and significantly improved verification defects as compared to little or no reliable communication.

Acknowledgement

The authors would like to thank Lt. David Carlos of AFRL/VACA for his assistance in generating portions of the data used in this work.

References

- ¹Schumacher, C. J., Chandler, P. R., and Rasmussen, S. J., "Task Allocation for Wide Area Search Munitions Via Network Flow Optimization," *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, 2001.
- ²Nygaard, K. E., Chandler, P. R., and Pachter, M., "Dynamic Network Flow Optimization Models for Air Vehicle Resource Allocation," *Proceedings of the Automatic Control Conference*, 2001.
- ³Schumacher, C. J., Chandler, P. R., and Rasmussen, S. J., "Task Allocation for Wide Area Search Munitions Via Iterative Network Flow Optimization," *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, 2002.
- ⁴Rasmussen, S. J., *MultiUAV Users Manual*, U.S. Air Force Research Laboratory (AFRL/VACA), 2nd ed., 2004, AFRL-VA-WP-TR-2004-????
- ⁵Mitchell, J. W., Schumacher, C. J., Chandler, P. R., and Rasmussen, S. J., "Communication Delays in the Cooperative Control of Wide Area Search Munitions via Iterative Network Flow," *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, 2003.
- ⁶Rasmussen, S. J. and Chandler, P. R., "MultiUAV: A Multiple UAV Simulation for Investigation of Cooperative Control," *Proceedings of the Winter Simulation Conference*, 2002.
- ⁷Rasmussen, S. J., Mitchell, J. W., Schulz, C., Schumacher, C. J., and Chandler, P. R., "A Multiple UAV Simulation for Researchers," *Proceedings of the AIAA Modeling and Simulation Technologies Conference*, 2003.
- ⁸Rasmussen, S. J., Chandler, P. R., Mitchell, J. W., Schumacher, C. J., and Sparks, A. G., "Optimal vs. Heuristic Assignment of Cooperative Autonomous Unmanned Air Vehicles," *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, 2003.
- ⁹Chandler, P. R. and Pachter, M. N., "UAV Cooperative Classification," *Workshop on Cooperative Control and Optimization*, Kluwer Academic Publishers, 2001.
- ¹⁰Chandler, P. R. and Pachter, M. N., "Hierarchical Control of Autonomous Teams," *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, 2001.
- ¹¹U.S. Space and Naval Warfare Systems Command, *GlobalHawk*, Jun 2004, <http://tinyurl.com/3wmmmp>.
- ¹²U.S. Air Force Research Laboratory, Air Vehicles Directorate, *SensorCraft*, Jun 2004, <http://tinyurl.com/57nod>.
- ¹³Mitchell, J. W. and Sparks, A. G., "Communication Issues in the Cooperative Control of Unmanned Aerial Vehicles," *Proceedings of the Forty-First Annual Allerton Conference on Communication, Control, & Computing*, 2003.
- ¹⁴U.S. Naval Academy, *What is Link-16?*, May 2004, <http://tinyurl.com/yqppjm>.
- ¹⁵Mitchell, J. W., Rasmussen, S. J., and Sparks, A. G., "Effects of Target Arrival Rate on Mission Performance of Cooperatively Controlled UAVs with Communication Constraints," *Proceedings of the 43rd IEEE Conference on Decision and Control*, 2004, (to appear).
- ¹⁶Ford, Jr., L. and Fulkerson, D., *Flows in Networks*, Princeton University Press, Princeton, NJ, 1962.
- ¹⁷Mitchell, J. W., Rasmussen, S. J., and Sparks, A. G., "Communication Requirements in the Cooperative Control of Wide Area Search Munitions Using Iterative Network Flow," *Proceedings of the Fourth International Conference on Cooperative Control and Optimization*, 2003.