TASK-BASED INTERFACES FOR DECENTRALIZED MULTIPLE UNMANNED VEHICLE CONTROL

Andrew S. Clare[*] and Mary L. Cummings[†]

Enhanced autonomy in Unmanned Vehicles (UV) has given human operators the ability to move from teleoperation to supervisory control of single vehicles, and now multi-vehicle coordination. This research seeks to leverage task-based interfaces, where the human operator guides a fleet of decentralized UVs via high-level goals as opposed to individual vehicle control. In such decentralized control architectures, each vehicle computes its locally best plan to accomplish the mission goals with shared information. The results of two experiments are described where 62 participants performed multi-UV missions in an existing decentralized multiple unmanned vehicle simulation environment under increasing task load. Results suggest that a system which uses a task-based interface and decentralized control algorithms may be robust to task load increases by mitigating operator cognitive overload.

INTRODUCTION

A future concept of operations for controlling Unmanned Vehicles (UVs) is one of a single, operator supervising multiple, heterogeneous (air, sea, land) UVs.[1] Many modern UVs can execute basic operational and navigational tasks autonomously and can collaborate with other UVs to complete higher level tasks, such as surveying a designated area.[2, 3] Numerous automated path-planning and scheduling algorithms have been developed recently to aid operators with scheduling tasks for multiple UVs.[4-10] In the presence of unknown variables, possibly inaccurate information, and changing environments, however, automated scheduling algorithms do not always perform well. Though fast and able to handle complex computation far better than humans, computer optimization algorithms are notoriously "brittle" in that they can only take into account those quantifiable variables identified in the design stages that were deemed to be critical.[11, 12] Thus, human management of automated planners is crucial and operators will need to comprehend a large amount of information while under time pressure to make effective decisions in dynamic environments.

One significant concern in this concept of operations is the potential high workload for the operator, and possible negative performance consequences. Previous research in human supervisory control of multiple unmanned vehicles has shown that under centralized, vehicle-based control, increasing task load can lead to operator cognitive overload and performance degradation.[13, 14] To address this concern, we seek to leverage task-based interfaces, where the human operator

[*] Doctoral Student, MIT Aeronautics and Astronautics Department, 77 Massachusetts Avenue, Cambridge MA 02139.
[†] Associate Professor, Director of the MIT Humans and Automation Lab, 77 Massachusetts Avenue, Cambridge MA 02139.

only guides the high-level goals of the team of UVs (as opposed to guiding each individual vehicle), and a decentralized control architecture, where each vehicle computes its locally best plan to accomplish the mission goals with shared information. As opposed to a centralized algorithm, where all decisions are made by a single agent, the decentralized framework is robust to a single point of failure, since no single agent is globally planning for the fleet.[15] Plans can be carried out even if communication links with the human operator are intermittent or lost. The architecture is scalable, since adding additional agents also adds computational capability.[16]

The set of experiments described in this paper investigates whether task-based decentralized control can mitigate cognitive overload under increasing task load. The rest of this paper is organized as follows. The Experimental Test Bed section describes the multiple unmanned vehicle simulation environment with a task-based control interface and decentralized algorithms for vehicle routing and task allocation used in this research. The Methodology section describes the set of experiments run with different task loads to compare operator workload and system performance. Finally, the results of the experiments are analyzed and conclusions are drawn.

EXPERIMENTAL TEST BED

This effort utilized a collaborative, multiple UV simulation environment called Onboard Planning System for UVs Supporting Expeditionary Reconnaissance and Surveillance (OPS-USERS), which leverages decentralized algorithms for vehicle routing and task allocation. This simulation environment functions as a computer simulation but also supports actual flight and ground capabilities.[17] All the decision support displays described here have operated actual small air and ground UVs in real-time.

Operators controlled multiple, heterogeneous UVs for the purpose of searching the area of interest for new targets, tracking targets, and approving weapons launch. All targets were initially hidden, but once a target was found, it was designated as hostile, unknown, or friendly, and given a priority level by the user. Hostile targets were tracked by one or more of the vehicles until they were destroyed by a Weaponized Unmanned Aerial Vehicle (WUAV). Operators had to approve all weapon launches. Unknown targets were revisited as often as possible, tracking target movement. A primary assumption was that operators had minimal time to interact with the displays due to other mission-related tasks. Icons represent vehicles, targets, and tasks, and the symbols are consistent with MIL-STD 2525[18].

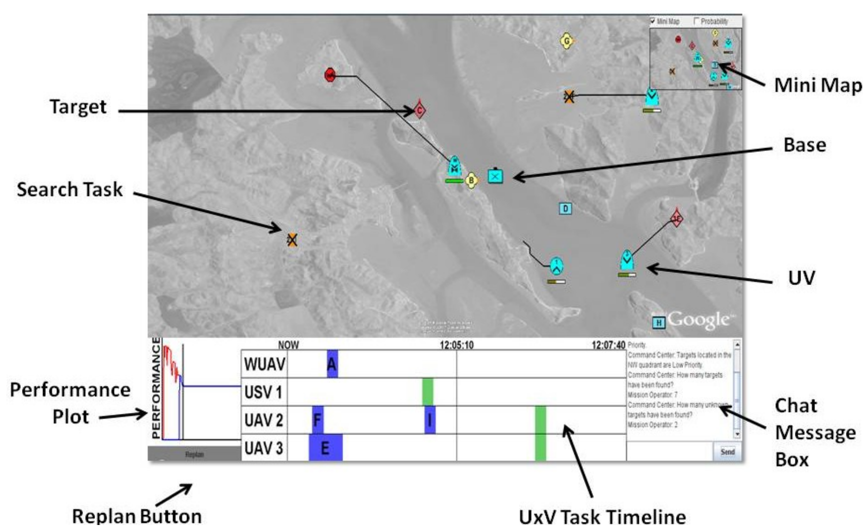


Figure 1. The Map Display.

Provided with intelligence via a text messaging "chat" box, the operator had the ability to re-designate unknown targets or create search tasks for emergent targets. The primary interface used by the operator is a Map Display, shown in Figure 1. Operators had two exclusive tasks that could not be performed by automation: target identification and approval of all WUAV weapon launches. Operators created search tasks, which dictated on the map those areas the UVs should specifically search. Operators also had scheduling tasks in that they dictated when certain tasks should occur. These were performed in collaboration with the automation such that when the planner recommended schedules, operators accepted, rejected, or modified these plans.

A task-based, decentralized implementation was chosen for the planner to allow rapid reaction to changes in the environment.[15] The decentralized task planner used in OPS-USERS is the Consensus Based Bundle Algorithm (CBBA), a decentralized, polynomial-time, market-based protocol that can generate new schedules on the order of seconds.[16] The human operator provides high-level *task-based* control, as opposed to more low-level *vehicle-based* control, by approving which tasks should be completed by the vehicles. The list of operator-approved tasks is referred to as a *strategic-level plan*.

In such architectures, operators do not directly individually task a single vehicle. When appropriate, the decentralized task planner can modify the *tactical-level plan* (at the vehicle level) without human intervention, which includes changing the task assignment without affecting the overall plan quality (i.e., agents switch tasks). The CBBA algorithm is able to make these local repairs faster through inter-agent communication than it could if it had to wait for the next update from the human operator. The architecture is scalable, since adding additional agents also adds computational capability, and the decentralized framework is robust to a single point of failure, since no single agent is globally planning for the fleet.[16]

Operators were shown the results of the scheduling algorithm through a decision support interface, called the Schedule Comparison Tool (SCT), shown in Figure 2. The display showed the high-level performance metrics of each schedule, as well as unassigned high, medium, and low priority tasks that could not be completed by one or more of the vehicles. If the operator was unhappy with the automation-generated schedule, he or she could create new tasks or ask the automation to prioritize a particular task, in effect forcing the decentralized algorithms to re-allocate the tasks across the UVs. Details of the interface design and usability testing are provided in previous research.[19]

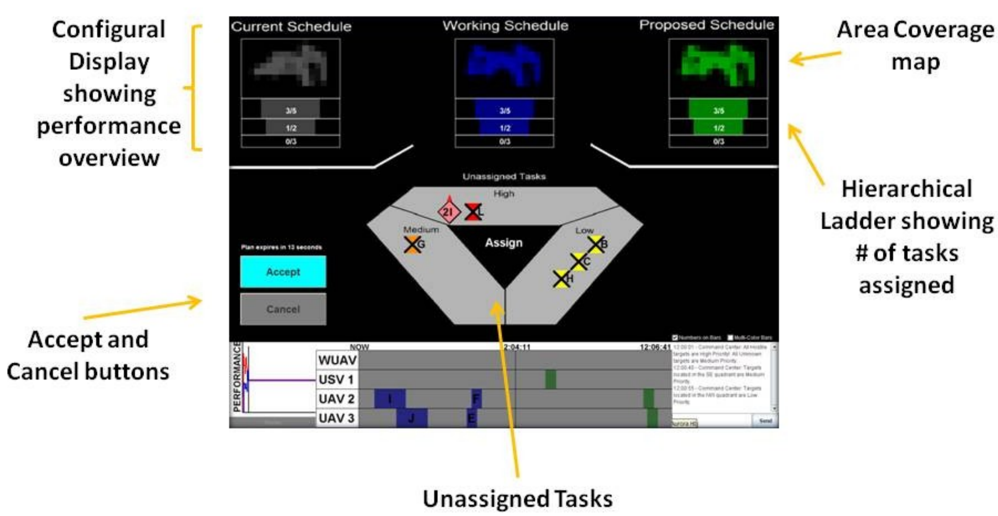


Figure 2. The Schedule Comparison Tool (SCT).

METHODOLOGY

Task load is defined as the level of tasking that an operator is asked to perform. In contrast, operator workload is defined as the mental resource demand experienced by the operator as a function of task load.[20] We hypothesize that under increasing task load, operators controlling a team of UVs can avoid high mental workload (cognitive overload) by manipulating the strategic-level plan of the UVs with the assistance of decentralized planning algorithms. The human operator provides high-level *task-based* control, as opposed to more low-level *vehicle-based* control, since the operator cannot directly individually task a single vehicle.

To examine this hypothesis, we compared results from a previous moderate task load study with 31 participants,[21] to a new high task load experiment with 31 different participants. Both studies had the same objective to search for new targets, track those targets, and destroy hostile targets with 3 heterogeneous UVs and a WUAV, and all scenarios lasted 10 minutes. To provide additional control for task load in each study, operators were prompted to view automation-generated schedules as seen in Figure 2 at prescribed intervals of either 30 or 45 seconds. Changing the rate of prompts to use the SCT modulates the task load of the operator, such that 30s replan intervals induce higher workload than the 45s intervals. These intervals have been validated in a previous study.[21] The replanning prompt was given through the green illumination of the replan button (Figure 1) and an aural replan alert sounded when a schedule was available that the automation deemed better than the current schedule. This Replan Prompt Interval served as one of the independent variables in the two experiments.

Increasing Task Load Levels between the experiments served as the other independent variable, with a high and a medium task load experiment. The high task load experiment had double the number of targets as the medium (20 vs. 10), 11 of which were hidden from the UVs at first and became available to be found throughout the scenario. In addition, the UVs traveled 5 times faster in the high task load experiment and operators received 71% more chat messages that either provided intelligence information updates or Situational Awareness (SA) questions requiring operator response. The average number of tasks required to be done per minute for the 4 scenarios (two task load levels at two replan intervals) are shown in Table 1, showing that the high task load experiment had 90% higher task load than the medium task load experiment.

Table 1. Average Tasks Required Per Minute Across the Four Scenarios.

Task Load Level	Replan Prompt Interval (s)	Average Tasks Required Per Minute
Medium	45	3.96
	30	4.05
High	45	7.30
	30	7.90

In order to familiarize each subject with the interfaces in Figures 1 and 2, a self-paced, slide-based tutorial was provided, which typically took subjects approximately twenty minutes to complete. Then, subjects had a ten-minute practice session during which the experimenter walked the subject through all the necessary functions to use the interface and to develop schedules before accepting them. Each subject was given the opportunity to ask the experimenter questions regarding the interface and mission during the tutorial and practice session.

Each subject experienced the different Replan Prompting Intervals in a counterbalanced and randomized order to control for learning effects. The total subject population of both experiments consisted of 62 subjects: 44 men and 18 women. Ages ranged from 19 to 67 years with a mean of 25.3 years and standard deviation of 7.7 years.

RESULTS

An Analysis of Variance (ANOVA) model was used for parametric dependent variables (α = 0.05). For dependent variables that did not meet ANOVA assumptions, non-parametric analyses were used.

Workload Metrics

Workload was measured via a utilization metric (i.e., percent busy time) because utilization has proven to be sensitive to changes in workload in similar multiple tasking, time-pressured scenarios.[22, 23] Operators were considered "busy" when performing one of the following tasks: creating search tasks to specify locations on the map where UVs must search for targets; identifying targets by looking at the imagery and designating a target type and priority level; approving weapons launches on hostile targets; chat messaging with the virtual, remote command center; and replanning in the SCT.

As expected, there was a significant difference in utilization between the Task Load Levels ($F(1, 120) = 252.732$, $p < 0.001$) and also a significant difference between the Replan Prompting Intervals, $F(1, 120) = 4.312$, $p = 0.040$. Operators were working harder under higher task loads, as shown in Figure 3, with a 48% difference between the medium and high Task Load Level experiments. It should be noted that despite task load nearly doubling, operators never achieved complete cognitive overload (defined for this paper as 100% utilization, or percent busy time) during the mission, with a maximum utilization of 89.8%. Thus, even though their task load increased by 90%, their workload only increased by 48%.

Finally, operators provided subjective ratings of their workload after each scenario on a Likert scale from 1-5. The average rating was 2.8 in the medium task load experiment and 3.6 in the high task load experiment, a significant difference with non-parametric testing ($p < 0.001$). These results confirm that operators recognized that they were working harder under higher task load.

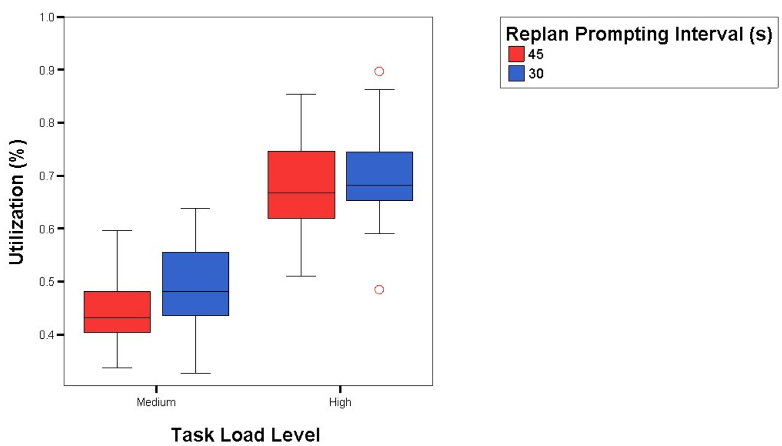


Figure 3. Utilization Over Increasing Task Loads.

System Performance Metrics

The mission performance dependent variables included two weighted performance scores, a Target Finding Score (TFS) and a Hostile Destruction Score (HDS) that accounted for differences in available targets and different speeds and distances covered by the UVs in the two different experiments. Equations 1 and 2 show the formulas for calculating the TFS and HDS respectively:

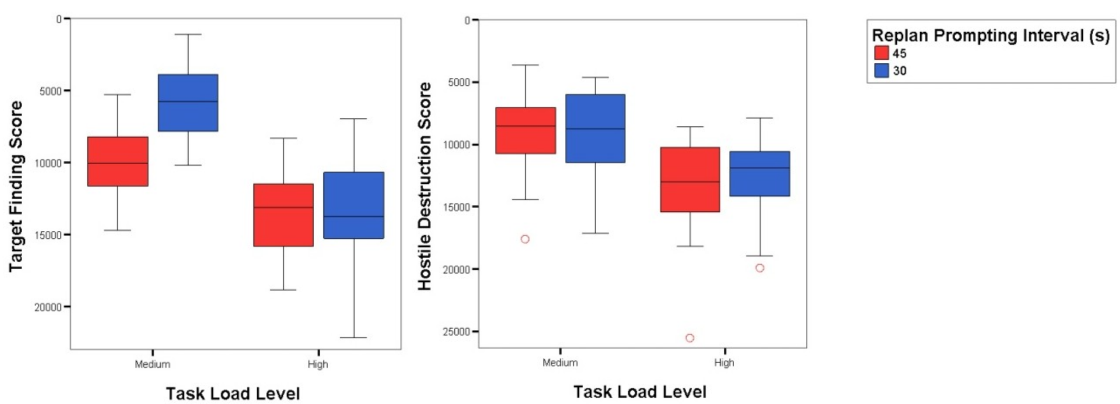
$$\frac{\boldsymbol{nv}}{\mathrm{T}^*}\left[\frac{\sum_{i=1}^{\mathrm{T}^*} \boldsymbol{f_i}}{\boldsymbol{T}}\right] \quad (1)$$

$$\frac{nv}{\mathrm{H}^*}\left[\frac{\sum_{i=1}^{\mathrm{H}^*} d_i - a_i}{H}\right] \quad (2)$$

Where:

- n = Total Number of Vehicles
- v = Speed of Vehicles
- T* = total number of targets available in simulation (10 in medium task load, 20 in high task load)
- H* = total number of hostile targets available in simulation (5 or 10 targets)
- T = total number of targets found during simulation
- H = total number of hostile targets destroyed during simulation
- f_i = Time in seconds that target i was found (set to 0 if never found)
- a_i = Available time = MAX(time target was designated as hostile via chat; time target was uncloaked)
- d_i = Time in seconds that a hostile target was destroyed (set to 0 if never destroyed)

These new metrics were developed to measure the combined human-computer system's efficiency at either finding or destroying targets. The metrics represent the average distances traveled to find or destroy a target, normalized by the target density of the scenario. A lower score indicates better performance in either finding targets or destroying hostile targets. The results for both TFS and HDS are shown in Figure 4, showing that there were significant decreases in performance with higher task load.



(a) Target Finding Score (TFS) **(b) Hostile Destruction Score (HDS)**

Figure 4. System Performance Metrics Over Increasing Task Loads.

Significant differences were found in TFS for both Task Load Level ($F(1, 120) = 128.334$, $p < 0.001$) and Replan Prompting Interval ($F(1, 120) = 15.363$, $p < 0.001$). There was also a significant interaction effect for TFS between Task Load Level and Replan Prompting Interval, $F(1, 120) = 17.935$, $p < 0.001$. This is a notable result, in that the TFS improved significantly when moving from the 45 second Replan Prompting Interval to the 30 second Replan Prompting Interval in the medium Task Load Level experiment, but the same results did not occur in the high Task Load Level experiment. A comparison of the data from the 30 and 45 second Replan Prompting Interval scenarios in the medium Task Load Level experiment showed that operators found fewer targets in the 30 second Interval scenario (51% vs. 62% for the 30s and 45s intervals respectively), but the average time to find those few targets was lower (79 seconds into the simulation on average compared to 156 seconds). Operators in the 30s interval found targets early in the simulation and did not find many more later on, while operators in the 45s interval continued to find targets throughout the simulation. This difference in the average time to find a target did not occur in the high Task Load Level experiment because of the higher target density (20 targets vs. 10 targets, which caused the average time to find the target to drop to ~110s).

For HDS, there was a significant difference between Task Load Levels ($F(1, 120) = 39.637$, $p < 0.001$), but no significant differences between Replan Prompting Intervals ($F(1, 120) = 0.079$, $p = 0.779$). While the task load was a primary performance driver for HDS, the rate at which the replanning requests alerted the operator did not affect performance. This is expected since there were fewer destroy events as compared to search events.

As expected, performance scores decreased with higher task load. The TFS decreased on average 42% from the medium to high Task Load experiments and HDS decreased 30%. While these are fairly substantial performance decrements, it should be considered that again, task load increased by 90%, so there was not an associated linear decrease in performance. A previous experiment with centralized, vehicle-based control has shown average decrements of 28% in mission performance and 32% in system performance efficiency over an increase in task load of only 43%.[13] Thus, the task-based, decentralized control showed comparable performance decrements to centralized, vehicle-based control despite a more than double increase in task load.

CONCLUSION

A set of experiments was conducted to determine whether a task-based control interface and a decentralized UV architecture could mitigate cognitive overload and performance degradation under increasing task load. Under task-based, decentralized control, a human operator provides high-level control by approving which tasks should be completed by the team of vehicles without directly individually tasking a particular vehicle. Then the decentralized network of vehicles chooses how to allocate the approved tasks among themselves and can make tactical-level changes on their own, such as switching tasks. The architecture is scalable, since adding additional agents also adds computational capability, and the decentralized framework is robust to a single point of failure, since no single agent is globally planning for the fleet.

Experimental results showed that with a task-based, decentralized control system, increasing task load by 90% led to an operator workload increase of only 48%. The increase was not as dramatic as might be expected from almost doubling the task load of the operator. Also, cognitive overload (defined for this paper as 100% utilization, or percent busy time) was not seen, suggesting that a task-based, decentralized control system may be robust to high task load situations. Additionally, with the 90% increase in task load, system performance decreased by only 30-42%, so there was not an associated linear decrease in performance.

These results have important implications for the types of interfaces and algorithms that should be employed in future UV systems. A task-based, decentralized control system can possibly achieve superior workload mitigation under high task loads with comparable system performance as compared to a vehicle-based, centralized control system. Preventing high workload situations in a command and control environment is crucial for maintaining system performance and preventing costly or deadly errors. Also, the scalability and robustness to failure of decentralized networks will be essential for larger fleets of coordinated UVs in the future. Task-based control interfaces and decentralized algorithms for vehicle routing and task allocation hold significant promise for future human supervisory control systems.

ACKNOWLEDGMENTS

This research is funded by the Office of Naval Research (ONR) and Aurora Flight Sciences. Andrew Clare was supported by a National Defense Science and Engineering Graduate Fellowship for part of this effort. Olivier Toupet of Aurora Flight Sciences provided extensive support to generate the algorithms that run the simulation. Professor Jonathan How and the members of the MIT Aerospace Controls Laboratory provided experimentation test bed support. Benjamin Roth and Caroline Kurtz assisted with running experiments. Armen Mkrtchyan and Kristopher Thornburg assisted with data analysis.

REFERENCES

[1]Naval Studies Board, "Autonomous Vehicles in Support of Naval Operations," National Research Council, Washington D.C., 2005.

[2]Alighanbari, M., and How, J. P., "Robust Decentralized Task Assignment for Cooperative UAVs," *AIAA Guidance, Navigation, and Control Conference and Exhibit*, Keystone, CO, 2006.

[3]Bertuccelli, L. F., Choi, H.-L., Cho, P., and How, J. P., "Real-Time Multi-UAV Task Assignment in Dynamic and Uncertain Environments," *AIAA Guidance, Navigation, and Control Conference*, Chicago, IL, 2009.

[4]Boskovic, J., Knoebel, N., Moshtagh, N., Amin, J., and Larson, G., "Collaborative Mission Planning & Autonomous Control Technology (CoMPACT) System Employing Swarms of UAVs," *AIAA Guidance, Navigation, and Control Conference*, Chicago, IL, 2009.

[5]Karaman, S., Faied, M., Frazzoli, E., and Girard, A., "Specification and Planning of Interactive UAV Missions in Adversarial Environments," *AIAA Guidance, Navigation, and Control Conference*, Chicago, IL, 2009.

[6]Franke, J., Moffitt, V., Housten, D., Clark, J., and Lizardi, A., "ICARUS: The Construction of and Lessons Learned from a General-Purpose Autonomy System," *AIAA Infotech@Aerospace Conference*, 2009.

[7]Bieniawski, S., Pigg, P., Vian, J., Bethke, B., and How, J., "Exploring Health-Enabled Mission Concepts in the Vehicle Swarm Technology Lab," *AIAA Infotech@Aerospace Conference* Seattle, WA, 2009.

[8]Casbeer, D., and Argyle, M., "Hierarchical Distributed Task Assignment for UAV Teams," *AIAA Infotech@Aerospace Conference*, St. Louis, MO, 2011.

[9]Toupet, O., and How, J., "Collaborative Sensor Fusion and Management for Multiple UAVs," *AIAA Infotech@Aerospace Conference*, St. Louis, MO, 2011.

[10]Wood, J., and Hedrick, J. K., "Space Partitioning and Classification for Multi-target Search and Tracking by Heterogeneous Unmanned Aerial System Teams," *AIAA Infotech@Aerospace Conference*, St. Louis, MO, 2011.

[11]Silverman, B. G., "Human-Computer Collaboration," *Human-Computer Interaction,* Vol. 7, No. 2, 1992, pp. 165-196.

[12]Smith, P., McCoy, E., and Layton, C., "Brittleness in the Design of Cooperative Problem-solving Systems: The Effects on User Performance," *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans,* Vol. 27, No. 3, 1997, pp. 360-371.

[13]Nehme, C., "Modeling Human Supervisory Control in Heterogeneous Unmanned Vehicle Systems," Ph.D Dissertation, Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, Cambridge, MA, 2009.

[14]Nehme, C., Mekdeci, B., Crandall, J. W., and Cummings, M. L., "The Impact of Heterogeneity on Operator Performance in Futuristic Unmanned Vehicle Systems," *The International C2 Journal,* Vol. 2, No. 2, 2008, pp. 1-30.

[15]Whitten, A. K., "Decentralized Planning for Autonomous Agents Cooperating in Complex Missions," Masters of Science Dissertation, Massachusetts Institute of Technology, Cambridge, MA, 2010.

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

[17]How, J. P., Fraser, C., Kulling, K. C., Bertuccelli, L. F., Toupet, O., Brunet, L., Bachrach, A., and Roy, N., "Increasing Autonomy of UAVs," *IEEE Robotics & Automation Magazine,* Vol. 16, No. 2, 2009, pp. 43-51.

[18]"Interface Standard, Common Warfighting Symbology," MIL-STD-2525B, 30 January, 1999.

[19]Fisher, S., "Replan Understanding for Heterogenous Unmanned Vehicle Teams," Masters of Engineering Dissertation, Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, MA, 2008.

[20]Wickens, C. D., and Hollands, J. G., *Engineering Psychology and Human Performance*, 3rd ed., Prentice Hall, Upper Saddle River, N.J., 2000.

[21]Cummings, M. L., Clare, A. S., and Hart, C. S., "The Role of Human-Automation Consensus in Multiple Unmanned Vehicle Scheduling," *Human Factors,* Vol. 52, No. 1, 2010, pp. 17-27.

[22]Cummings, M. L., and Guerlain, S., "Developing Operator Capacity Estimates for Supervisory Control of Autonomous Vehicles," *Human Factors,* Vol. 49, No. 1, 2007, pp. 1-15.

[23]Cummings, M. L., and Nehme, C. E., "Modeling the Impact of Workload in Network Centric Supervisory Control Settings," *2nd Annual Sustaining Performance Under Stress Symposium*. College Park, MD, 2009.