6-2014

Task Versus Vehicle-based Control Paradigms in Multiple Unmanned Vehicle Supervision by a Single Operator

Mary L. Cummings
Duke University

Luca F. Bertuccelli
United Technologies Research Center

Jamie C. Macbeth
Massachusetts Institute of Technology, jmacbeth@smith.edu

Amit Surana
United Technologies Research Center

Task Versus Vehicle-Based Control Paradigms in Multiple Unmanned Vehicle Supervision by a Single Operator

Mary L. Cummings, *Senior Member, IEEE*, Luca F. Bertucelli, Jamie Macbeth, *Member, IEEE*, and Amit Surana

***Abstract*—There has recently been a significant amount of activity in developing supervisory control algorithms for multiple unmanned aerial vehicle operation by a single operator. While previous work has demonstrated the favorable impacts that arise in the introduction of increasingly sophisticated autonomy algorithms, little work has performed an explicit comparison of different types of multiple unmanned vehicle control architectures on operator performance and workload. This paper compares a vehicle-based paradigm (where a single operator individually assigns tasks to unmanned assets) to a task-based paradigm (where the operator generates a task list, which is then given to the group of vehicles that determine how to best divide the tasks among themselves.) The results demonstrate significant advantages in using a task-based paradigm for both overall performance and robustness to increased workload. This effort also demonstrated that while previous video gaming experience mattered for performance, the degree of experience that demonstrated benefit was minimal. Further work should focus on designing a flexible automated system that allows operators to focus on a primary goal, but also facilitate lower level control when needed without degradation in performance.**

***Index Terms*—Autonomy, centralized, decentralized, drones, human performance, scheduling, unmanned vehicles (UVs), video gaming.**

I. Introduction

UNMANNED VEHICLES (UVs) have become commonplace across many domains. Recent advances in UV technology have made possible new envisioned applications that have the potential of revolutionizing the kinds of missions that can be performed in both military and civilian domains. For example, unmanned resources are now being called upon to move beyond conventional unmanned aerial vehicle (UAV) applications of surveillance and now commonplace missions include cargo resupply [1], oil pipeline inspection [2], and search and rescue [3].

For current day military operations, UAVs require human guidance to varying degrees and often through several operators. Conventional piloting skills have been replaced by point-and-click control so that traditional pilots are no longer needed to control such systems. Currently, while one operator supervises the actual flight activity of the UAV (the "pilot" who is responsible for stable flight and navigation), the other operator (the "sensor operator") typically monitors the UAVs sensors and is responsible for the payload, such as a camera, and coordinates with the "pilot" so that he or she can maneuver the UAV for the best system response.

There has been significant recent research activity attempting to streamline UAV operations and reduce staffing in order to invert the current many-to-one ratio of operators to vehicles. This is important not just for military operations, but also for future commercial operations where air traffic controllers will direct both manned and unmanned aircraft. This means that all the functions of maintaining balanced flight, navigation, and payload management, currently split across both the pilot and the sensor operator for traditional systems, must be combined into one with significant automation in order for one person to control multiple UAVs.

One of the key enablers in these systems is the increasing autonomy in the vehicles themselves. In typical UAVs, low-level control loops are now closed by conventional autopilots, freeing the mission crew from mundane low-level control tasks, which allows them to focus on more high-level mission relevant tasks such as resource allocation and sensor management. Such advanced automation allows for force multiplication in that it is now possible for one operator can control more than one UAV at a time (including multiple heterogeneous UVs, which is a stated future vision for the US military [4]).

Such increases in autonomy and shift in operator responsibility inevitably result in new design challenges. The design of the underlying architecture across UVs and between the human is critical to determine the functions that will be carried out by computers and humans, respectively. Moreover, the underlying architecture, including communications, tasking, and decision authority, will significantly influence operator workload such that if the human is required to do too many tasks by a network of multiple vehicles, overall system performance could suffer.

One commonly found architecture in the literature is a centralized approach, also known as vehicle-based control, where one operator serially controls multiple UVs. An alternate

Manuscript received July 3, 2013; revised November 21, 2013; accepted January 18, 2014. This work was supported by the United Technologies Research Center and the Office of Naval Research. This paper was recommended by Associate Editor B. M. Mettler.

M. L. Cummings is with the Department of Mechanical Engineering, Duke University, Durham, NC 27708 USA (e-mail: m.cummings@duke.edu).

L. F. Bertuccelli and A. Surana are with the United Technologies Research Center, East Hartford, CT 06118 USA (e-mail: bertuclf@utrc.utc.com; suranaa@utrc.utc.com).

J. Macbeth is with the Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, Cambridge, MA 02139 USA (e-mail: jmacbeth@mit.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/THMS.2014.2304962

architecture is task-based control, where operators interact with an overall automated mission and payload manager, which coordinates a set of tasks for a group of UVs that typically communicate with one another in a decentralized fashion, instead of having the operator individually tasking each vehicle. In a task-based architecture, operators convey high-level goals to an automated mission manager (such as requesting that an area be searched), which then allows the UVs to coordinate across the group to determine how to assign particular tasks, which may be dynamic.

In other words, in a vehicle-based control, an operator must assign tasks to individual vehicles, while in task-based control, the operator generates a task list, which is then given to the group of vehicles that determine through a decentralized approach how to best divide the tasks among themselves.

A task-based architecture provides substantial benefit over a vehicle-based one in that the operator and his or her ground control station does not become a single point of failure. For example, because a network of decentralized UVs communicates the tasks across the network, if one vehicle breaks down, another can take its place. Another advantage is that the task-based system is robust to lapses in operator situation awareness and delays since UVs do not necessarily have to wait for commands. However, emergent UV behavior in such systems can be suboptimal and confusing for an operator, and it could be difficult for operators to correct problems unless they have the ability to understand emergent behavior and then execute the necessary commands to correct the system.

A task-based architecture may allow operators to control more UVs in a network since they only issue high-level goal-based commands instead of micromanaging attributes of individual UVs; however, no studies to date have examined the costs and benefits of vehicle-based versus task-based control from the operator's perspective in a controlled setting. Thus, the goal of this paper is to assess these two different architecture design approaches and understand the implications of their use in a simulated, multi-UV mission scenario.

We show through an experimental comparison of the two architectures that the vehicle-based approach, while more familiar to current-day operators, resulted in lower performance. The task-based approach enabled operators to focus on the primary goal, while being able to achieve higher rates of task completion. Both vehicle and task-based approaches demonstrated a clear diminishing return in total number of tasks processed as the number of UVs increased, but that operators interacting with a task-based architecture were more robust to increases in workload than those in the vehicle-based condition.

This paper is subdivided in the following sections. Section II discusses relevant background work in human supervisory control; Section III discusses the vehicle/task-based approaches and discusses the experimental setup; Section IV discusses the experimental results, and we conclude in Section V.

II. Background on MultiUAV Supervisory Control

There has been a significant volume of literature addressing the issue of human supervisory control in multiple UAV control. It should be noted that there are no operational systems in existence today where one operator controls multiple UVs. Currently, several are in development by both the US, e.g., [5] and other countries' defense departments, e.g., [6]; therefore, these efforts are still in early research and development stages.

Current key research topics revolve around the topics of human performance and limitations [7], [8], design strategies to manage potentially high workload in a centralized multiple UAV setting [9], and the inherent tradeoff between operator workload and performance [10]. In these areas, much research has revolved around understanding the upper limits of how many UAVs can be adequately controlled by a single operator, and understanding the associated ramifications in operator workload and situation awareness.

Other work has been devoted to quantifying the role of supportive algorithms at different layers in the human decision making chain for multiple UVs. For example, Bellingham looked at the computational issues of cooperative path planning for multiple UAVs in dynamic environments using receding horizon control [11]. However, other research has demonstrated that while performance improvements can arise from a human interacting with a randomized algorithm, workload can increase as a result in using such nondeterministic algorithms [12].

In terms of designing these underlying algorithms, Savla *et al.* [13] and Srivastava *et al.* [14] investigated different kinds of receding horizon control approaches in a vehicle-based control setting, in which attention allocation algorithms optimized operator tasks for a finite length of time in the future, and then replanned based on new and updated tasks in the queue. Key results demonstrated that such attention algorithms can optimize tasking given workload considerations and delays in tasks.

Other related research has included allowing operators or automation to requeue difficult tasks in an attempt to either satisfy or optimize a schedule [15], with no clear performance advantage by either approach. However, using a decentralized single operator, multiple UV simulation where vehicles effectively "bid" for tasks in order to optimize mission performance, allowing a single operator to coach a scheduling algorithm resulted in significantly enhanced system performance [16].

While the literature has introduced different analyses on different algorithms for both vehicle- and task-based allocation, a key consistent gap has been a lack of direct comparison between the two approaches. This gap raises the following important research question: how does the incorporation of a higher level of automation resource allocation algorithm impact the workload and performance of the operator in both a task-based and vehicle-based paradigm? The next section discusses the experimental approach developed to address this question.

III. Task- versus Vehicle-Based Experiment

An experimental comparison of the two architectures, vehicle-based, and task-based approaches under increasing task load was conducted to examine the previously raised issues.

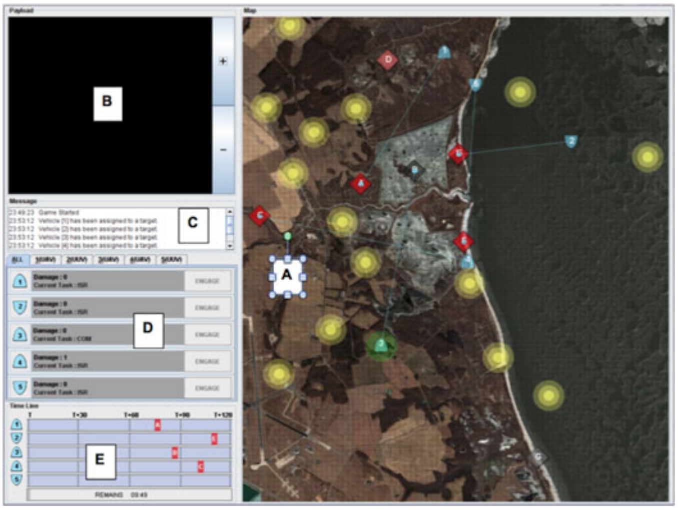

Fig. 1. Vehicle-based control RESCHU interface (A: map, B: camera window, C: message box, D: vehicle control panel, E: timeline).

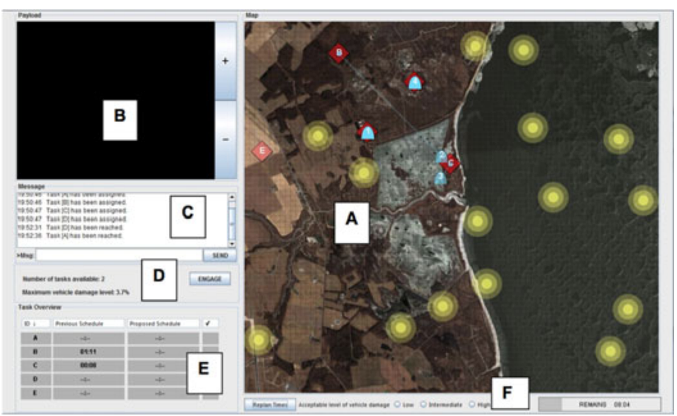

Fig. 2. Task-based RESCHU interface (A: map, B: camera window, C: message box, D: engagement panel, E: task overview, F: replan panel).

A. Experiment Test Bed

The test bed used for this experiment is called Research Environment for Supervisory Control of Heterogeneous Unmanned vehicles (RESCHU), which allows a single operator the ability to control multiple UVs in a search and identify task in an urban coastal and inland setting. Two versions of RESCHU were implemented for this study to reflect the vehicle (see Fig. 1) and task-based (see Fig. 2) forms of control, but there exists common elements in both versions. Both interfaces have a map (A), a window that shows a camera image for target identification (B), and a message board (C). The map shows different vehicles [both UAVs and unmanned underwater vehicles (represented by the bullet shapes)], hazard areas, targets (diamonds), and the paths of the vehicles. These UVs incur damage when intersecting any of the circular threat areas on the map. The threat areas occasionally appear and disappear, creating the need for dynamic path planning for the vehicles.

In both versions, once a vehicle reaches a target, an image search task corresponding to that vehicle becomes available to the operator in the camera window (see B in Figs. 1 and 2). The operator selects an engage button to search this simulated video feed to visually identify an object of interest, such as a car or a geographic feature, by panning and scanning through the image and then clicking the object specified in the search task description for designation. The name of the object to find is given to the operator by a simulated supervisor in the message window (see C in Figs. 1 and 2). In both versions, after this search task is completed, a new target is automatically assigned to the vehicle, and the vehicle begins moving along a straight line path to that target.

The main differences between the vehicle-based and task-based interfaces, discussed next, are in the method of target assignment and path planning for the vehicles.

1) Vehicle-Based RESCHU: The vehicle-based RESCHU interface, in addition to the map and camera window, has a vehicle control panel (see D in Fig. 1) and a timeline (see E in Fig. 1). In the vehicle window, there is a tab for each operator's vehicle showing health, status, and damage level. The timeline informs the operator of the time when a vehicle will reach a certain waypoint or target. The timeline displays the estimated time of arrival to waypoints and to targets, as well as the total amount of time remaining in the simulation.

In Fig. 1, the vehicles are controlled by clicking on the map. When the experiment starts, each vehicle is assigned randomly to a target, which is not necessarily the closest target to the vehicle. Vehicles can be selected by either clicking on the vehicle on the map or by selecting the corresponding tab in the vehicle window. The operator can choose to either "change the goal" or "add waypoint" for each vehicle by clicking on the map in the desired location. The vehicle visits waypoints first before moving to the goal. The creation of waypoints is also used to maneuver individual vehicles to avoid hazard areas, which can appear with no warning. When selected, the vehicle on the map flashes in a different color than the rest, and the corresponding tab in the vehicle window is highlighted.

Once the visual task engage button is clicked, the map locks, meaning that the operator cannot control the vehicles anymore, but the vehicles will keep moving according to their paths. Thus once started, the visual search task must be completed. Only when the operator has submitted an object, either correctly or incorrectly, the search and identify task ends. The operator then returns to controlling the vehicles by changing goals and adding waypoints for each vehicle individually until a new search and identify task is engaged.

2) Task-Based RESCHU: The interface for the task-based version of RESCHU differs from vehicle-based RESCHU in that it has an engagement panel that summarizes available engagement tasks, a task overview panel for estimated times for vehicles to arrive at their targets according to schedules created by the automated planner (an A^* algorithm), and a replan panel that requests an overall new plan.

The visual task engagement panel (see D in Fig. 2.) allows the operator to see a quick summary of how many visual engagement tasks are available for all the UVs, as well as the damage level of the UV with the highest damage. It also contains an "ENGAGE" button for the image analysis task, and when selected, the image tasks are prioritized in a first in-first out order.

For the task-based interface, the system decides which image to present to the operator, whereas in the vehicle-based interface, the operator selects which task (linked to a specific vehicle) to process.

The replan panel (see F in Fig. 2) consists of a "replan times" button, which causes the A* planner to replan every vehicle to its closest target. In the vehicle-based system, the operator has to look at each vehicle, judge whether the current intended target met all the prespecified goals, and then redirect each vehicle to a new goal as desired, including inserting waypoints to avoid existing or pop-up threat areas. Thus, each replan is essentially a locally-optimized solution, led by the human.

In the task-based system, when the operator selects the "replan times" button, a global plan is proposed by the automation that attempts to optimize the number of vehicles versus the remaining targets, as well as account for threats in the route planning. Thus, the task-based system relies more heavily on automation for route planning, threat avoidance, resource allocation, and scheduling.

When the "replan times" button is pressed, the task overview panel (see E in Fig. 2) shows the proposed new times to reach that target, as compared with the previous schedule plan. If the operator approves of the new plan (either for an individual task or an entirely new schedule), he or she then clicks in the last approval (checkmark) column in the task overview panel. This is, in effect, the task list where the operator determines the queue for which engagement tasks he or she would like to see. Therefore, for the task-based interface, the operator can approve only the task list for when various targets/images gets processed, but they have no control over which vehicle does the task.

The replan panel also includes a damage acceptance level (high, medium, low), which lets operators express to the automation how much risk they are willing to accept in terms of avoiding pop up threat areas for all vehicles. This is in an indirect way to control path planning. Before selecting the "Replan Times" button, the vehicle damage acceptance level must first be chosen. If "Low" is chosen, paths planned by the automation will completely avoid the hazard areas. If "Intermediate" is selected, the path can cross the outer areas of the hazard area. If "High" is chosen, the path can cross through any part of a hazard area. In the vehicle-based interface, operators create waypoints for individual vehicles to circumvent threat areas to varying degrees of risk acceptance.

In the task-based version of RESCHU, there is no explicit timeline like there is in the vehicle-based version, but this same information is embedded in the task overview panel. There is also no tab for each vehicle showing health, status, and damage level, but the vehicle with the highest vehicle damage is shown in the engagement panel.

B. Participants and Procedure

A total of 60 participants took part in the experiment (43 males and 17 females). A total of 46 participants were student at a major East Coast university, and the remaining were working professionals. Out of the remaining 14 participants, a total of seven participants categorized themselves as an engineer, scientist, software developer, or researcher; one participant was in the financial industry; one person was an entrepreneur; one was a graphic designer; three were administrators; and the remaining participant declined a category. In the subsequent data analysis, four participants' data were dropped because of data recording issues.

The participants were assigned either a task-based or vehicle-based control mode. After filling out a demographic survey and completing a self-paced slide presentation, each participant was given a self-paced practice session, with guidance from the experimenter on achieving the goals of maximum target identification and minimal damage. No weight for these two variables was given. Subjects were paid \$15, and a \$200 gift certificate was awarded to the top scorer of all participants of the study, which was determined by rank-ordering participants separately by target identifications and vehicle damage. The winner was the subject with the least sum of these two ranks. No score was explicitly shown to the participants in the interface.

Each participant supervised four, eight, and twelve vehicles in three separate scenarios, and the order in which the participant controlled the different UAVs was randomized and counterbalanced. Since one of the hypotheses was that a task-based interface would allow operators to control a greater number of vehicles without significant increase in workload, the task load was varied via the four, eight, and twelve vehicle factor.

Each scenario lasted 10 min, and the participants were instructed to maximize the number of targets correctly identified, while also ensuring that the vehicles minimized any flight into hazard zones. As the operator completed an imagery task associated with a target, an additional task was inserted, keeping the total number of targets equal to the number of vehicles. Each experiment in total lasted about ~1 h.

IV. Results and Discussion

The experiment was originally designed as a 2 (task versus vehicle-based architecture) × 3 (four, eight, and twelve vehicles) mixed factorial study. However, in the subsequent analysis of the data, an unexpected factor level emerged. For the task- and vehicle-based architecture factor levels, a distinct group of operators elected to never replan the vehicles, completely (but sometimes erroneously) trusting the automation to always find the best paths and best vehicle-target assignments. This was discovered because no replans in either architecture were recorded by this group for either interface. Thus, these operators ($N = 10$) focused exclusively on image analysis, effectively attempting to game the system to maximize their number of targets found.

When the demographic surveys were analyzed, this group of overtrusters was found to have significantly less video gaming experience that the participants in the task and vehicle factor levels. On average, the participants in the task and vehicle based conditions reported that they were light to moderate gamers (Mean (M) = 2.25 on a Likert scale of 1–5, with 5 representing daily gaming, standard deviation (SD) = 1.33). In contrast, the overtrusters reported a 1.1 mean rating (SD = 0.32), meaning little to no experience gaming. This was statistically significant in a one way analysis of variance (ANOVA, $p = .035$)

Because of this clearly identifiable category, the experiment design was changed post-hoc to reflect a 3 × 3 study, where the architecture level was augmented with a third category of "overtrusters". Arguably, this data could have been thrown out because the participants did not follow the rules of the experiment, but given that this type of automation bias has been seen repeatedly in similar experimental and real-world settings [17], these participants' data were preserved for comparative purposes.

A. Performance

Performance was assessed in two dimensions. First, participants were told to maximize the total number of targets correctly identified over the course of a single mission. However, they were also told to avoid damage. Therefore, two metrics give insight into overall performance, which is the number of targets correctly identified and the damage incurred over the course of a mission. We hypothesized that participants in the task-based condition would perform better in both cases since they had to spend less time determining which target to engage (since the automation did this), which also gave them more time to ensure vehicles avoided threat areas.

1) Targets Correctly Identified: Age was a significant covariate for number of target correctly identified (Pearson $R = -0.36$, $p < .001$, which is a moderate to strong correlation in human factors studies).

The resulting 3 × 3 repeated measures ANOVA with age as a covariate demonstrated that there was no significant effect for number of vehicles under control (i.e., no statistical difference in whether participants controlled four, eight, and twelve vehicles, but there was a significant effect for the architecture ($F(2,47) = 8.44$, $p = .001$). Fig. 3 demonstrates that as hypothesized, those participants in the task-based control ($M = 17.0$, SD $= 6.3$) outperformed their vehicle-based counterparts (with a Bonferroni pairwise significant result, $M = 15.0$, SD $= 4.3$, $p < .001$).

However, the overtrusting participants who appeared to try to game the system fell between these two groups, indicating that it was not an effective strategy to ignore vehicle path planning but that it did help them somewhat in terms of identifying more targets ($M = 16.0$, SD $= 6.2$). In order to assess this impact of this choice, the damaged incurred by each vehicle was evaluated, as this was directly affected by path planning, discussed in the next section.

One possible confound existed in that operators in the vehicle-based condition, once in the visual engagement window, had to commit to an answer about the location of the target before returning to the replanning task. In the task-based condition, operators could select the "replan times" button at any time, so in theory, they may not have been constrained to the decision point in the visual search tasks. Thus, we examined the average time operators in both conditions spent in the replanning task. If this design feature provided the task-based operators an advantage, we would expect to see more time spent in the replanning condition, but replanning times between the two conditions was statistically no different ($M = 26$ s SD $= 16.3$ s, $p = 0.281$).

One interesting overall trend to note is that while not statistically significant, the inverted U relationship in the data for Fig. 3(a) suggest that participants performed best given eight vehicles to manage, which is nearly identical to similar results found for control of entirely different vehicles but with similar levels of autonomy [18]. This approximate inverted U shape in Fig. 3(a) is likely an inherent task efficiency/capacity characteristic.

2) Damage: Age was also a significant covariate for the damage metric, which was measured as percentage of time in a threat area. The same statistical model as in the number of targets correctly identified was used, also yielding a nonsignificant result for the number of vehicles controlled, but a significant one for the architecture ($F(2, 47) = 10.74$, $p < .001$). Fig. 3(b) illustrates that indeed, for the overtrusting group, their strategy in ignoring the path planning element caused them to have significantly more damage than in the other groups ($M = 8.6$ SD $= 5.0$, Bonferroni pairwise comparisons $p = .008$ and $< .001$, respectively, for overtrusting-vehicle and overtrusting-task comparisons.) While there was no statistical difference between the task and vehicle comparisons ($M = 5.2$ SD $= 4.5$), it is interesting that the vehicle-based participants experienced the least vehicle damage, likely because they were focused on individual vehicles.

3) Risk Modulation: The damage metric in conjunction with the replanning metric reported earlier provide insight as to the risk modulation in each group. The vehicle-based control group modulated risk by replanning each of the vehicles around obstacles or to new targets. The task-based group mitigated risk via the "replan times" function, which allowed them to set their degree of acceptable risk by telling the automation to change any routing of the UAVs if they came to close to a threat. For both the damage and replanning metrics, there was no statistical difference between the two conditions, suggesting that the participants were able to adapt in both environments.

We further examined the use of the low, intermediate, and high buttons that let those in the task-based condition modulate their risk. Forty-three percent of thirty participants in this condition always set low for every replanning event, one person always set high, and the remainder used a mix of low, intermediate, and high. Only 27% of participants ever used the high risk button, and this high risk group was statistically no different from the others in terms of performance and damage incurred. However, interestingly, this high risk group had significantly higher utilization rates than the others who gravitated to low and intermediate risk ($F(1,49) = 5.439$, $p = 0.024$, High Risk $M = 0.80$ SD $= 0.06$, Low and Moderate Risk $M = 0.74$ SD $= 0.04$). This group preferred to work at a higher pace, suggesting that such measures may be useful in determining risk profiles but this deserves further research.

One striking result, particularly in light of risk mitigation, is that our hypothesis that overtrusters would do more poorly in this measure was correct; therefore, this group was unable to adapt to the risk, because they essentially ignored the need to replan. The only objective performance measure where task-based

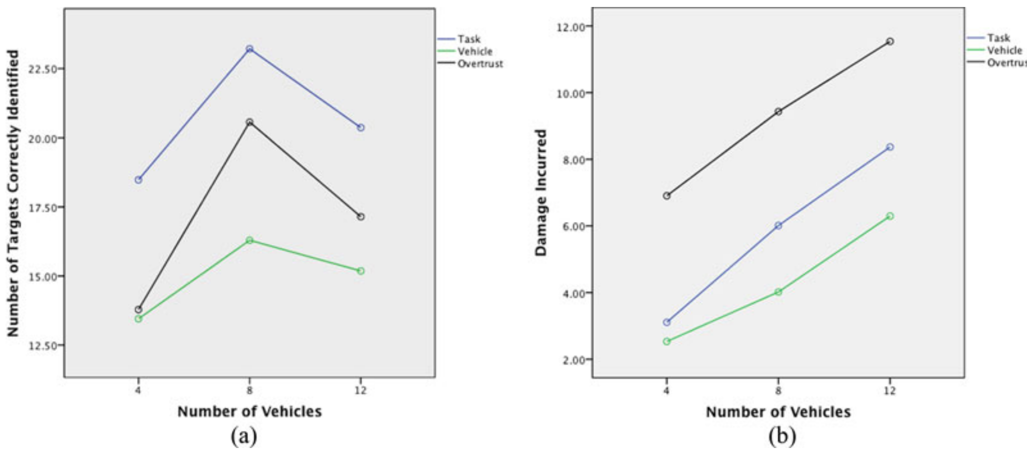

Fig. 3. Performance results for: (a) number of targets correctly identified and (b) damage Incurred.

participants did better than the vehicle-based was in number of targets correctly identified.

B. Workload

1) Subjective Workload: Participants were asked at the end of each test session to rate their mental workload on a Likert scale from 1–5 (with 5 representing maximum mental workload). Unlike previous results, the number of vehicles was significant with the 3 × 3 ANOVA (F(2, 52) = 49.2, $p < .001$). However, architecture was not significant ($M = 3.7$, SD = 0.9) and for all three groups (task, vehicle, and overtrust) participants tightly clustered around a subjective assessment of 3.1 for four vehicles (SD = 0.7), 3.8 for eight vehicles (SD = 0.8), and 4.2 for twelve vehicles (SD = 0.8). While participants were somewhat sensitive to the increase in taskload, it is important to note that even with the maximum number of vehicles (12), participants felt they were working at high, but not extreme levels of workload.

2) Objective Workload: To assess workload objectively, we measured utilization, which is the percent busy time of each participant. Busy time was calculated as any time the operator was interacting with any screen element including searching in the image pane or replanning the paths of the vehicles. While monitoring the interface is also cognitively consuming, percent busy time is an objective, clearly identifiable measure but likely underestimates overall workload. However, utilization has been shown to be an effective objective workload metric that correlates well with subjective measures of workload [18], [19].

We hypothesized that workload should be more robust to the increases in task load and that utilization and subjective workload would not increase at the same rate for task-based control than it did for vehicle-based control.

The 3 × 3 ANOVA yielded significance for both factor levels [architecture ($F(2,48) = 16.5$, $p < .001$) and number of vehicles ($F(2,48) = 125.9$, $p < .001$) Task $M_4 = 0.64$ $SD_4 = 0.09$, Vehicle $M_4 = 0.61$ $SD_4 = 0.07$, Overtrust $M_4 = 0.66$ $SD_4 = 0.10$; Task $M_8 = 0.83$ $SD_8 = 0.06$, Vehicle $M_8 = 0.71$ $SD_8 = 0.05$, Overtrust $M_8 = 0.79$ $SD_8 = 0.08$; Task $M_{12} = 0.87$ $SD_{12} = 0.04$, Vehicle $M_{12} = 0.76$ $SD_{12} = 0.07$, Overtrust $M_{12} = 0.86$ $SD_{12} = 0.05$]. Fig. 4 demonstrates that the increasing number of vehicles expectedly caused participants to spend objectively more time on tasks, but unexpectedly caused participants in the task and overtrust conditions to spend significantly more time interacting with the interface than in the vehicle condition (Bonferroni pairwise comparison p values for task and overtrust ($p = .935$) and for task and overtrust as compared with vehicle ($p > .001$)).

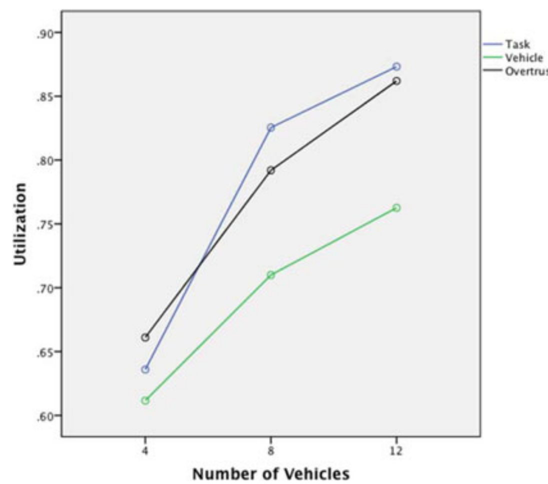

Fig. 4. Utilization results.

These results indicate that counter to our hypothesis about workload regulation, the participants in the task-based architecture and the overtrusters pushed themselves to process more targets [as evidenced by the results in Fig. 3(a)]. Therefore, it appears that the task-based interface and the overtrusters who never replanned felt they had the capacity to do more, while those in the vehicle-based scenario consistently worked at lower levels.

As indicated by the number of targets correctly found [see Fig. 3(a)], the ability to do more worked well for the task-based participants, but not as well for the overtrusters who ignored the need to replan vehicle routes around threat areas. And those directly controlling vehicles were not working as hard in terms of interacting with the interface, which suggest that they spent more

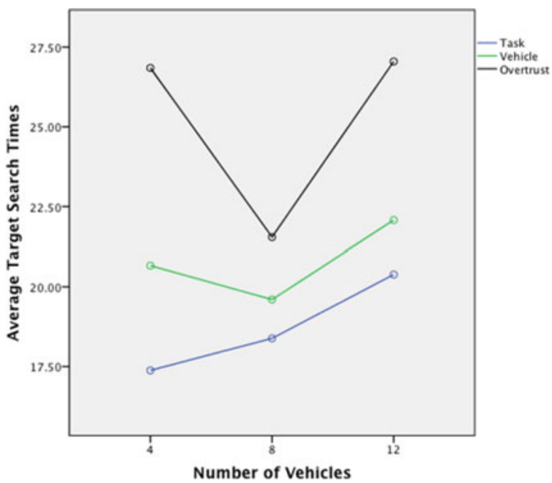

Fig. 5. Average target search times.

time searching the interface elements to support their various tasks.

Another interesting result from Fig. 4 is that the participants in the task-based and overtrust architectures control pushed utilization beyond the generally accepted utilization limits of ∼70% [20], although subjects did not necessarily find this level unacceptable as indicated by their subjective assessments as noted in the previous sections.

3) Efficiency and Search Times: Given that the overall task was heavily dependent on how many targets were identified, how much time, on average, spent searching for a target could give insight to workload management strategies. Using the same 3 × 3 repeated measures ANOVA, how much time was spent in each target search task was evaluated with significant results for the architecture factor level (F(2,48) = 4.7, p = .013) and the number of vehicles (F(2,48) = 6.054, p = .003, Task M_4 = 17.4 s SD_4 = 6.0 s, Vehicle M_4 = 20.7 s SD_4 = 6.2 s, Overtrust M_4 = 26.7 s SD_4 = 11.3 s; Task M_8 = 18.4 s SD_8 = 5.4 s, Vehicle M_8 = 19.6 s SD_8 = 4.1 s, Overtrust M_8 = 21.5 s SD_8 = 9.4 s; Task M_{12} = 20.4 s SD_{12} = 5.9 s, Vehicle M_{12} = 22.1 s SD_{12} = 5.6 s, Overtrust M_{12} = 27.1 s SD_{12} = 9.0 s). Fig. 5 demonstrates that the overtrusters spent the most time searching each image on average, which curiously dropped by a full 5 s in the eight vehicle condition as compared with the four and 12 cases.

In addition, the overtrusters also spent significantly more time searching than those in the task- and vehicle-based conditions (Bonferroni p = .010). The task and vehicle-based search times were statistically no different (p = .458). Thus, it appears that the overtrusters took more time but were not as efficient as the those participants in the task-based interface, who took much less time and correctly identified more targets [see Fig. 3(a)].

V. Conclusions and Future Work

The purpose of this effort was to evaluate the impact of a task-based control multiple UAV architecture as compared with vehicle-based control in terms of human performance. We hypothesized that given the same task load levels across the two different architectures, those operators with task-based control would have more time for target searching and vehicle path planning since they could do this at the aggregate level as opposed to directing individual vehicles. Such an interface that aggregates information allows operators to be goal focused, without having to spend additional time gathering information about individual entities.

Given a common experiment test bed, the task-based architecture led to higher performance scores in terms of number of targets correctly identified, but unexpectedly higher levels of objective workload. This partially explains why operators in this condition fared so much better in terms of the number of correctly identified targets. Operators in the task-based condition pushed themselves to work harder. At their peak performance in the eight vehicle condition, operators in the task-based condition were utilized ∼82%. This is substantially higher than the 70% heuristic often cited [20], which suggests that this threshold is dependent on the control architecture. However, the overall average utilization was 74% (SD = 11%), which demonstrates that this heuristic may be good enough for average populations (which is often how such heuristics are formed), but not necessarily reflective of the best performers.

On the other metrics, the task-based architecture was not statistically different from the vehicle-based architecture, meaning both sustained the same damage, and operators experienced similar subjective levels of workload as well as search times. It appears the task-based operators did better overall because they were able to more accurately identify targets, with only minimal interaction with the aggregate route planner for replans. It is possible that since vehicle-based operators divided their attention between individual vehicles, their image searches were less fruitful, even although they took, on average, the same time as the task-based operators. This should be investigated in future research, which can look at eye tracking to determine if the scans of vehicles and/or the image searching is inefficient.

The emergence and performance of the overtrusters deserves further attention. Overtrusters ignored the vehicle route replanning functions in both control architectures, effectively over relying on the automation to manage the vehicles appropriately. This overreliance is linked to inexperience but what is interesting is the degree of difference between the overtrusters who did not replan in any interface and reported little to no video game playing, and the remainder of the participants who reported monthly occasional game playing. This was effectively only one level difference on a Likert scale, but clearly affected the overall results. While recent research suggests intensive gamers have superior visual sensitivity than nongamers [21], this research suggests that such a positive benefit can occur at less intense skill levels.

In addition, the overtrusters, while performing the worst in terms of vehicle damage, performed better than the vehicle-based performers in terms of number of targets accurately identified. This raises the question of training. It is possible that had the overtrusters been given targeted training, they could have substantially improved performance. We leave this to further work, since training efficiency has major implications for costs of operating unmanned vehicles.

While there were clearly strong elements of the task-based interface for multiple unmanned vehicle control, this study does not suggest this should be the only architecture. Indeed, it is likely that for operational systems, some hybrid mix would be needed, and the degree of autonomy in the system would drive the balance.

As our study demonstrated, when not required to divide attention across multiple entities and information is aggregated for presentation, operators can perform well, but this comes at a loss of control and possibly the ability to manage contingencies and unexpected situations. Further work is needed to determine how to design a flexible system that both allows operators to focus on a primary goal, but then drill down into the details when needed without degradation in performance.

ACKNOWLEDGMENT

We would like to thank A. Mkrtchyan and M. Weehuizen for their support in data gathering and analysis. The reviewers' comments also greatly improved this work.

REFERENCES

[1] Reuters. US Marines extend K-MAX unmanned helicopter's use in Afghanistan, May 19, 2013.

[2] D. Hausamann, W. Zirnig, and G. Schreier, "Monitoring of gas transmission pipelines—a customer driven civil UAV application," presented at the in 5th ONERA-DLR Aerosp. Symp., Toulouse, France, 2003.

[3] J. Weidlich,. Aerial drone locates Sask. man injured in rollover crash, May 19, 2013

[4] Office of the Secretary of Defense, "FY2009–2034 unmanned systems integrated roadmap," *Department of Defense*, Washington DC2009.

[5] A. J. Rowe, K. K. Liggett, and J. E. Davis, Vigilant Spirit Control Station: A Research Testbed For Multi-Uas Supervisory Control Interfaces. H. E. Directorate, Wright-Patterson AFB, OH Air Force Research Laboratory, 2009.

[6] T. Chen, D. Campbell, G. Coppin, M. Mooij, and F. Gonzalez, "A capability framework vialisation for multiple heterogeneous UAVS: From a mission commander's perpsective," in *Proc. 28th Int. Congr. Aeronautical Sci.*, Brisbane, Australia, 2012, pp. 1–10.

[7] S. R. Dixon and C. D. Wickens, "Control of multiple-UAVs: A workload analysis," presented at the 12th Int. Symp. Aviation Psychol., Dayton, OH, USA, 2003.

[8] C. A. Miller, "Modeling human workload limitations on multiple UAV control," in *Proc. 48th Annu. Meet. Human Factors Ergonomics Soc.*, New Orleans, LA, USA, 2004, pp. 526–527.

[9] C. A. Miller, R. P. Goldman, H. B. Funk, P. Wu, and B. B. Pate, "A playbook approach to variable autonomy control: Application for control of multiple, heterogeneous unmanned air vehicles," presented at the Am. Helicopter Soc. 60th Annu. Forum, Baltimore, MD, USA, 2004.

[10] M. L. Cummings, C. E. Nehme, and J. W. Crandall, "Predicting operator capacity for supervisory control of multiple UAVs," in *Studies in Computational Intelligence*, J. S. Chahk, L. C. Jain, A. Mizutani, and M. Sato-Ilic, Eds. New York, NY, USA: Springer, 2007.

[11] J. S. Bellingham, "Coordination and control of UAV fleets using mixed-integer linear programming," in *Masters, Aeronautics, and Astronautics*. Cambridge, MA, USA: MIT, 2002.

[12] A. Caves, "Human-Automation collaborative RRT for UAV mission path planning," *Electrical Engineering and Computer Science*, Cambridge, MA, USA: MIT, 2010.

[13] K. Savla, C. E. Nehme, E. Frazzoli, and T. Temple, "On efficient cooperative strategies between UAVs and humans in a dynamic environment," presented at the AIAA Guid., Navigat., Control Conf., Honolulu, HI, USA, 2008.

[14] V. Srivastava, A. Surana, and F. Bullo, "Adaptive attention allocation in human–robot systems," in *Proc. Amer. Control Conf.*, Montreal, Canada, 2012, pp. 2767–2774.

[15] L. F. Bertuccelli and M. L. Cummings, "Operator choice modeling for collaborative UAV visual search tasks," *IEEE Syst., Man, Cybern. A, Syst., Humans*, vol. 42, no. 5, pp. 1088–1099, Sep. 2012.

[16] M. L. Cummings, J. How, A. Whitten, and O. Toupet, "The impact of human-automation collaboration in decentralized multiple unmanned vehicle control," *Proc. IEEE*, vol. 100, no. 3, pp. 660–671, Mar. 2012.

[17] M. L. Cummings, "Automation bias in intelligent time critical decision support systems," presented at the AIAA 3rd Intell. Syst. Conf., Chicago, IL, USA, 2004.

[18] M. L. Cummings and S. Guerlain, "Developing operator capacity estimates for supervisory control of autonomous vehicles," *Human Factors*, vol. 49, pp. 1–15, 2007.

[19] B. Donmez, C. Nehme, and M. L. Cummings, "Modeling workload impact in multiple unmanned vehicle supervisory control," *IEEE Syst., Man, Cybern. A, Syst. Humans*, vol. 40, no. 6, pp. 1180–1190, Nov. 2010.

[20] W. B. Rouse, *Systems Engineering Models of Human–Machine Interaction*. New York, NY, USA: North Holland, 1983.

[21] L. G. Appelbaum, M. S. Cain, E. F. Darling, and S. R. Mitroff, "Action video game playing is associated with improved visual sensitivity, but not alterations in visual sensory memory," *Attention, Perception, Psychophysics*, vol. 75, no. 6, pp. 1161–1167, 2013.

Mary L. Cummings (SM'03) received the B.S. degree in mathematics from the United States Naval Academy, Annapolis, MD, USA, in 1988, the M.S. degree in space systems engineering from the Naval Postgraduate School, Monterey, CA, USA, in 1994, and the Ph.D. degree in systems engineering from the University of Virginia, Charlottesville, VA, USA, in 2004.

She was a Naval Officer and a Military Pilot from 1988 to 1999, and was one of the Navy's first female fighter pilots. She is currently an Associate Professor with the Department of Mechanical Engineering and Materials Science, Duke University, Durham, NC, USA. Her research interests include human-unmanned vehicle interaction, human-autonomous system collaboration, human-systems engineering, public policy implications of unmanned vehicles, and the ethical and social impact of technology.

Luca F. Bertuccelli received the Bachelor's degree in aeronautical and astronautical engineering from Purdue University, West Lafayette, IN, USA, and the S.M. and Ph.D. degrees in aeronautics and astronautics (major area of controls / estimation, and minor area in human factors) from the Massachusetts Institute of Technology (MIT), Cambridge, MA, USA, where his research work was in the area of planning algorithms for complex aerospace systems and human supervisory control.

He is currently a Staff Research Engineer in human–machine interaction / user experience with the Systems Department, United Technologies Research Center (UTRC), East Hartford, CT, USA. He joined the UTRC after being a Postdoctoral Associate at MIT from 2009 to 2011 in the Humans and Automation Lab (Department of Aeronautics and Astronautics). He is a former active duty Air Force Officer, where he served as a Test Engineer at Eglin AFB, FL, USA.

Jamie C. Macbeth (M'12) received the B.S. degree in mathematical physics from Brown University, Providence, RI, USA, the M.S. degree in physics from Stanford University, Stanford, CA, USA, and the Ph.D. degree in computer science from the University of California, Los Angeles, CA.

He is currently a Research Affiliate with the Humans and Automation Laboratory, Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, Cambridge, MA, USA. His research interests include human–computer interaction, human factors of software engineering, and natural language understanding. He is a Member of the ACM.

Amit Surana received the Bachelor's degree in mechanical engineering from Indian Institute of Technology, Bombay, India, in 2000, the M.S. degree in mechanical engineering and the M.A. degree in mathematics both from Pennsylvania State University, University Park, PA, USA, in 2002 and 2003, respectively, and the Ph.D. degree in mechanical engineering from Massachusetts Institute of Technology, Cambridge, MA, USA, in 2007.

He is currently a Staff Research Engineer in system dynamics and optimization group with the systems department at United Technologies Research Center, East Hartford, CT, USA. His research interests include applied dynamical system and controls, decision making under uncertainty, planning and learning in multiagent systems, and human supervisory control.