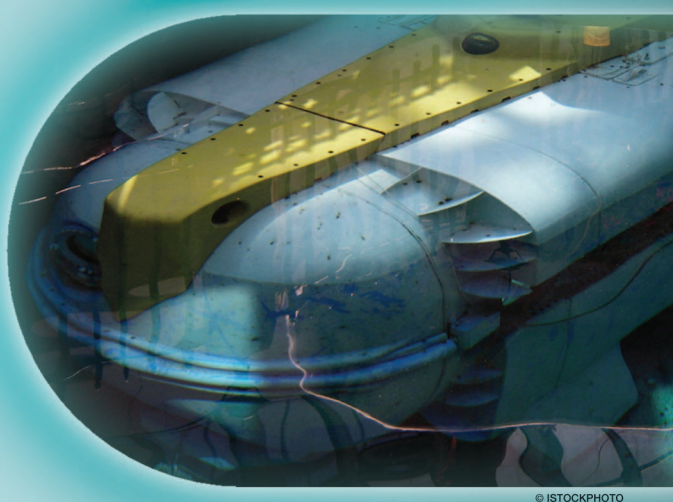
UAV Task Assignment

An Experimental Demonstration with Integrated Health Monitoring

BY BRETT BETHKE, MARIO VALENTI, AND JONATHAN P. HOW



Mobile Multirobot Systems

Unmanned aerial vehicles (UAVs) are becoming vital warfare and homeland security platforms because they have the potential to significantly reduce cost and risk to human life while amplifying warfighter and first-responder capabilities. To date, these vehicles have been operated in real missions with some success, but there remain challenging barriers to achieving the future vision of multiple UAVs operating cooperatively with other manned and unmanned vehicles in national airspace and beyond [1]. Among these is the problem of developing efficient and effective algorithms for simultaneously controlling and coordinating the actions of multiple autonomous vehicles in a dynamic environment. A particular concern is that off-nominal conditions or degraded components could reduce the capabilities of these UAVs to accomplish the mission objectives.

This article builds on the very active area of planning and control for autonomous multiagent systems (see [2] and [3] and the references therein). In principle, some of the issues raised in this problem are similar to questions arising in manufacturing systems [4], [5] and air transportation [6]–[8]. In addition, similar problems have been investigated under the Defense Advanced Research Projects Agency sponsored mixed initiative control of teams of autonomous agents [9]–[11]. While these efforts have made significant progress in understanding how to handle some of the complexity inherent in multiagent problems, the research in this article considers issues related to how vehicle health (e.g., fuel management and vehicle failures) affects the real-time mission planning (e.g., the task assignment). This work represents a step toward enabling robust decision making for distributed autonomous UAVs by improving the team's operational reliability and capabilities through better system self-awareness and adaptive mission planning.

The proposed methods for solving the overall multiagent problem typically involve formulating several smaller subproblems, each of which is simpler and, therefore, easier to solve [12]. One such solution architecture is shown in Figure 1, in which a number of components are combined to achieve the overall goals of the mission. The mission planning component is the highest level in the system. It keeps track of the mission objectives and generates tasks, which are discrete actions whose completion will aid the overall accomplishment of the mission. Examples of tasks include searching for, identifying, or tracking an object of interest. The mission planner provides the list of tasks to the task assignment component, which decides which of the available vehicles should perform each task based on the information about the tasks and the capabilities of the vehicles. Once the assignments have been made, they are sent to the trajectory designer, which plans feasible trajectories for each vehicle. The output of the trajectory designer is a sequence of waypoints for each vehicle to follow. These waypoints are sent to the vehicle controllers, which compute the actual controls needed to follow the waypoint plans.

Digital Object Identifier 10.1109/M-RA.2007.914931

1070-9932/08/$25.00©2008 IEEE

Knowledge of the fuel state of the vehicle is important to be able to estimate the remaining useful flight time of the vehicle.

Inherent in each of the components in the architecture is a set of interconnected models used to predict future system behavior. For example, the controller contains a model of the control input dynamics of the vehicle, while the task assignment component contains a model of the performance each vehicle can be expected to produce if assigned to a given task. In the most general sense, system actions are selected by searching for actions that lead to desirable, predicted outcomes as given by the system models. Clearly, the performance of the system, therefore, depends heavily on the accuracy of these models.

One strategy for improving the accuracy of the models is to include additional feedback loops that provide information that can be used to adjust the models in real time. The amount, type, and quality of feedback information that each component receives plays a large role in how effectively the system can deal with dynamically changing factors in the environment, mission objectives, and state of the vehicles. Intuitively, feedback is necessary wherever there is uncertainty in the system, so that the initial plan of action made by each of the components of the planner can be modified when changes occur. Uncertainty may be present at all levels of the planning architecture as a result of incomplete knowledge of many factors, such as actuator performance at the control level, dynamic constraints at the trajectory design level, sensor health at the task assignment level, and long-term maintenance needs at the mission management level.

This article focuses on the health management problem at the task assignment level, developing a feedback mechanism for the performance model used by the task assignment algorithm. The assignment problem has been studied extensively [13]–[15]. However, most of the work done to date has used only a static vehicle performance model, making it difficult for these approaches to adapt to unexpected changes, such as sensor failures, during the course of the mission. The goal of this article is to develop a feedback loop that uses health state information to update the performance model in real time. By updating the performance model of an already existing algorithm, previous work on the task assignment problem can be leveraged and extended without requiring the modification of the existing algorithm. Its performance can be improved only by improving the quality of information available to make assignments.

Selection of Performance Model

The selection of the performance model incorporating health state information about the vehicle is clearly an important aspect of the feedback design. The particular details of the model depend on the mission problem in question and the vehicle hardware being used. However, there are a number of classes of general features that may be appropriate to be included in a performance model.

Vehicle Translational Dynamics

At the level of the task assignment problem, the vehicle dynamics are usually abstracted as being first order with a maximum speed $v_{\max}$. This abstraction allows the task assignment algorithm to capture important aspects of the vehicles' performance (in particular, how long they can be expected to take to reach a particular task), while being sufficiently simple to allow computational tractability. Recall that the trajectory planning and control levels below the task assignment level are responsible for carrying out those lower-level functions, allowing this simplification to be made. Note also that this is the model used in most of the previous work on task assignment.

Propulsion System State

The vehicle propulsion system may be abstracted as an entity that enables the vehicle to move at the maximum speed $v_{\max}$. Health feedback about the propulsion system may dynamically modify $v_{\max}$ to reflect the state of the propulsion system. For example, knowledge of a failing motor may cause $v_{\max}$ to decrease from its nominal value.

Fuel State

Knowledge of the fuel state of the vehicle is important to be able to estimate the remaining useful flight time of the vehicle. The performance model should include an estimator that performs the remaining flight time calculation based on the remaining fuel, average fuel consumption rates, and perhaps other environmental factors. Use of this information allows the task assignment algorithm to safely make assignments while ensuring that vehicles can return to the base before running out of fuel.

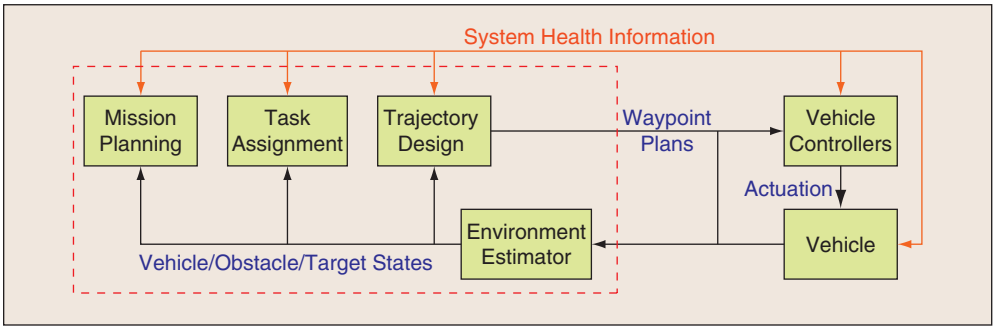


Figure 1. *Overall autonomous mission system architecture.*

Sensor States

The current performance level of any sensing system onboard the vehicle should be included in the model if they are required to carry out tasks. For example, if an onboard camera is to be used

for a surveillance task, the state of the camera (e.g., quality of the video signal) should be accounted for in the model.

Communication System State

Communication with other vehicles is often a requirement to enable vehicles to coordinate their actions with each other or relay messages to a distant ground station. Accounting for a vehicle's current estimated transmission and reception distances may allow the tasking system to avoid sending a vehicle to a location where it will be out of the communication range.

Modification of RHTA to Include Health Feedback: An Example

For the purposes of illustration, an example of incorporating a simple health feedback loop in the receding horizon task assignment (RHTA) algorithm is presented here. Briefly, the RHTA algorithm works as follows (for more details, see Alighanbari, 2004; Algorithm 2.3.1 [16]). Given the set of tasks W, distances between tasks $d(i, j)$, and vehicles V, RHTA enumerates all possible task sequences of specified length n_c. These sequences are called petals. The value of each petal is estimated as

$$S_{\text{vp}} = \sum \lambda^{T_{\text{ip}}} s_{\text{wd}},$$

where T_{ip} is the time at which task i is completed in petal p, s_{wd} is the task value, and λ is a time discount factor. Given the values of all the petals S_{vp}, RHTA solves the following optimization problem to select the optimal petal for each UAV:

$$\begin{aligned} \max\ & J = \sum_{v=1}^{N_v} \sum_{p=1}^{N_{\text{vp}}} S_{\text{vp}} x_{\text{vp}} \\ \text{subject to}\ & \sum_{v=1}^{N_v} \sum_{p=1}^{N_{\text{vp}}} A_{\text{vp}i} x_{\text{vp}} \le 1, \quad x_{\text{vp}} \in \{0, 1\} \\ & \sum_{p=1}^{N_{\text{vp}}} x_{\text{vp}} = 1, \quad \forall\ v \in \{1, \ldots, N_v\}. \end{aligned}$$

Here, x_{vp} is a binary variable that is equal to 1 if the pth petal is selected and 0 if not, and $A_{\text{vp}i}$ equals 1 if task i is visited by vehicle v in petal p and 0 otherwise.

In the example, health state information is represented by adding a fuel state to the vehicle model. In this case, the fuel model is straightforward.

- The vehicle's fuel level f_i decreases at a constant rate k_{fuel} anytime the vehicle is flying.
- If f_i reaches zero before the vehicle refuels, the vehicle crashes and is lost.
- In addition, the occurrence of failures is modeled as a Poisson process with time intensity ρ_f; when a failure occurs, the rate of fuel burn increases to $k_{\text{fuel,failure}} > k_{\text{fuel}}$. Thus, this failure mode increases the rate at which fuel is burned (and, thus, decreases the time a vehicle can complete tasks).

Due to the inclusion of randomly occurring failures, the fuel model is able to capture some of the uncertainty in the health state

Unmanned aerial vehicles (UAVs) are becoming vital warfare and homeland security platforms.

of the vehicle. If a failure occurs, the optimal task assignment may change due to the fact that the failed vehicle may no longer be able to service its assigned task. When this happens, the task assignment algorithm must be able to calculate the new optimal solution, subject to the new constraint imposed by the failure.

To handle these types of scenarios, the RHTA algorithm was extended to include the fuel state in the vehicle model. This was accomplished by including an estimate of each vehicle's operational radius, which is defined as $r_i \equiv v_{\max}\ (f_i / k_{\text{fuel}})$. The quantity r_i represents the maximum distance a vehicle can fly given its current fuel state, before running out of fuel. This information can be used to effectively prune the list of petals that RHTA considers to ensure that the vehicle can always safely return to the base before its fuel is exhausted. Specifically, the following constraint was added to the RHTA optimization problem:

$$L_i + d(w_{n_c}, x_{\text{base}}) \le r_i.$$

Here, $d(w_{n_c}, x_{\text{base}})$ represents the normal Euclidean distance between the last waypoint in the petal and the base, and

$$L_i = d(v, w_1) + \sum_{j=2}^{n_c} d(w_{j-1}, w_j)$$

is the total length of the petal. The constraint effective rejects a petal if the length of the petal plus the distance from the terminal waypoint w_{n_c} to base is greater than the current operational radius of the vehicle. This ensures that the vehicle visits only waypoints that allow it to return safely to the base.

With this extension, RHTA will assign a vehicle to return to the base when every possible permutation of waypoints is rejected by the pruning criterion. Thus, this method provides a simple rule that determines when a vehicle should return to the base for refueling since it cannot safely service any of the remaining tasks. Note that this method can create some problems if the above rule is followed too strictly since too many vehicles may be sent back to the base unnecessarily (i.e., when they still have large operational radii) if there are few or no active tasks. This problem can be solved by inserting artificial loiter tasks (w_{loiter}, p_{loiter}) into W. These tasks are treated in the same way as real tasks by the RHTA algorithm, but their purpose is to force the vehicles to remain in advantageous areas.

Simulation Results

A multivehicle mission simulation was developed to test the task assignment algorithms. This simulation includes a base location and a number of vehicles (20 were simulated in the following tests), as well as a mechanism to randomly generate tasks and vehicle failures. The simulation runs RHTA to repeatedly assign tasks to vehicles and simulate the resulting system response.

The mission was to be carried out over a period of time longer than the flight endurance of the UAVs being used.

There are two metrics of performance calculated in the simulation: the average time it took to service each task (response time) and how many vehicles were lost during the mission (vehicle loss occurs when a vehicle runs out of fuel before returning to the base).

Simulation results are shown in Figure 2. The first test used RHTA in its original form. Since unmodified RHTA does not account for vehicle failures, it will command a failed vehicle to continue toward its original target despite the risk that it may run out of fuel and crash before returning to the base. The performance of unmodified RHTA results in an average service time of 21.3 s, and a vehicle loss rate of 25%.

The second test used the modified form of RHTA, which proactively recalls failed vehicles to the base while quickly reassigning a new, healthy vehicle to the task, using the idea of the operational radius discussed previously.

The results in Figure 2 clearly show that the modified RHTA provides a faster average response time due to its proactive reassignment behavior. The improvement in response time is about 18%, which is significant considering that the speed of the vehicles has not been changed, only the way they are assigned. In addition, the vehicle loss rate is significantly reduced (by 20%) because failed vehicles are automatically returned to the base instead of continuing toward their assigned tasks.

Flight Results

A set of experiments incorporating all aspects of the work presented thus far was conducted to demonstrate a complete, fully autonomous, persistent search and track mission on MIT's RAVEN (Real-time indoor Autonomous Vehicle test Environment) platform [17]. In these experiments, the UAVs used were Draganfly V Ti Pro R/C helicopters (see Figure 3). The mission goals were to search for, detect, estimate, and track an unknown number of ground vehicles in a predefined search region. The mission was to be carried out over a period of time longer than the flight endurance of the UAVs being used (around 5–10 min, depending on the charge of the battery), necessitating the coordination of multiple UAVs coming in and out of the flight area as required to maintain coverage. Finally, active health monitoring was required to detect and adapt to potential vehicle camera failures during the test.

To carry out the mission, a cooperative vision-based target estimation and tracking system [18], [19] was combined with the modified RHTA algorithm. Furthermore, the RHTA tasking system was interfaced to an autonomous mission system [12] that employed battery monitors to estimate the time of flight remaining for each UAV in the search area and handled requests by the tasking system to activate vehicles for use in the search or tracking activities.

The experiment setup is shown in Figure 3. Three UAVs are initially stationed at their base location at the far north end of the flight area, while two ground vehicles are positioned at random locations in the southern region. For these experiments, one of the vehicles was positioned on top of a box, while the other was located on the ground and was free to move.

The progression of the mission is according to the following sequences.

1) At the beginning of the test, the tasking system requests a single UAV from the mission system.
2) Once the requested UAV is airborne, the tasking system commands this UAV to begin an area search. During this initial detection phase, the UAV keeps track of how many distinct targets it has detected so far and stores them in a target list. The detection phase lasts for 2 min.
3) After the detection phase ends, the tasking system requests another UAV from the mission system.

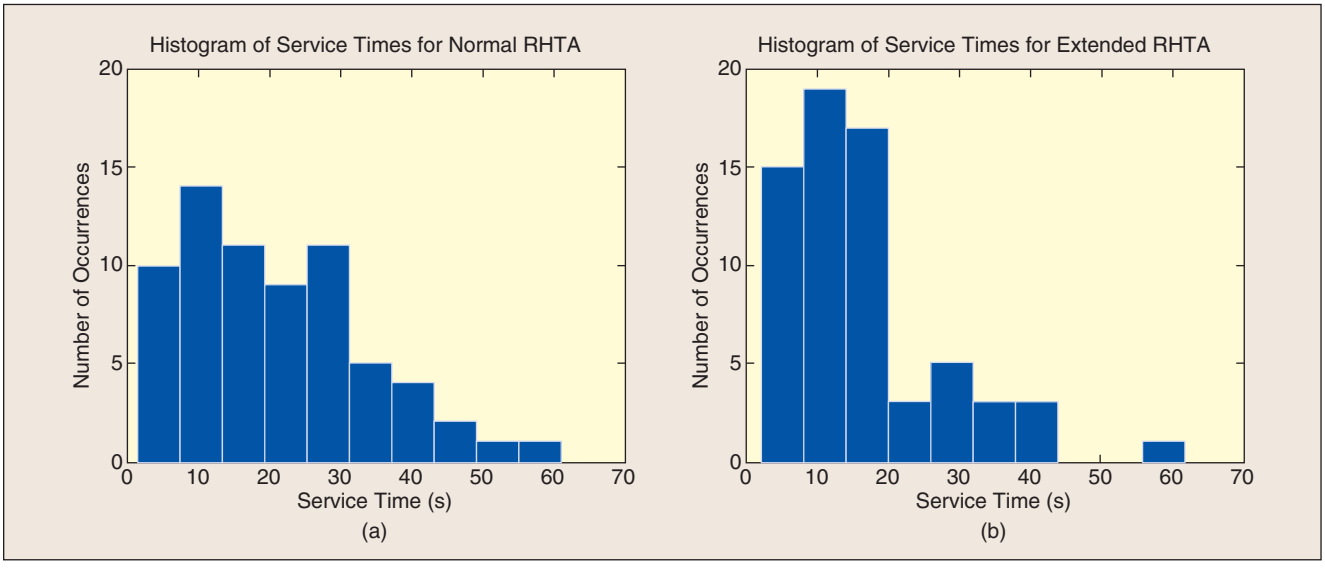


Figure 2. *Simulation results: (a) Normal RHTA: median service time, 18.8 s; average service time, 21.3 s; vehicles lost, 5 of 20 (25.0%). (b) Extended RHTA: median service time, 14.0 s; average service time, 17.4 s; vehicles lost, 1 of 20 (5%).*

4) Once the second UAV is airborne, the system enters the tracking phase. The tasking system commands the second UAV into the search area so that there are now two UAVs in the area. Together, these two UAVs sequentially visit each location in the target list found during the detection phase. The UAVs spend 1 min at each location before moving on to the next. If there is a target at the given location when the UAVs arrive, they begin tracking the target. Additionally, although the tracking logic is designed to prevent collisions between the vehicles, a potential function-based method is used to ensure an additional level of safety. If a UAV comes too close to another UAV or an obstacle in the environment, it is repelled away by seeking to move to an area of lower potential.
5) At any point in the mission, the tasking or mission systems may determine that a particular UAV needs to return to the base. The reason for this may be either that the UAV is getting low on the remaining battery lifetime or that the UAV's camera has failed or is performing poorly. In either case, when a return-to-base condition is detected, the tasking system sends a sequence of waypoints to the UAV to command it back to the base. Once at the base location, the mission system lands the UAV and schedules any necessary refuelling or maintenance. At the same time, another UAV is launched and sent to the search area. In this manner, the mission is able to continue as UAVs cycle in and out.
6) The mission continues until a preset mission time expires or the human operator stops the mission. For these experiments, the mission time was 11 min.

In the detection phase, a single vehicle explored the search area and detected the presence of two ground vehicles. Figure 4 shows an early segment of the tracking phase after the second UAV had entered the search area. In this phase, the two UAVs estimated and tracked the position of the eastern ground vehicle using the vision tracking system [18], [19].

Figure 5 shows the time history of the mission for all the three UAVs used in the experiment. At t = 0, UAV 1 is taking off and surveying the area. It then requests a second vehicle for support at t = 182 s, and UAV 2 takes off and begins assisting in tracking targets. At t = 304 s, UAV 1 receives a low battery warning and returns to base, while UAV 3 takes off to replace UAV 1. At t = 433 s, UAV 3 experiences a simulated camera failure. The system detects the failure and sends UAV 3 back to base while commanding UAV 1 to take off again. The mission ends at t = 650 s. At several points during the mission, UAVs were successfully changed out because of low-battery states. In addition, a simulated camera failure during the tracking phase of the mission resulted in the failed vehicle returning to the base and a replacement vehicle being sent out. Due to these correct system responses, the goals of the

The RHTA algorithm was extended to include the fuel state in the vehicle model.

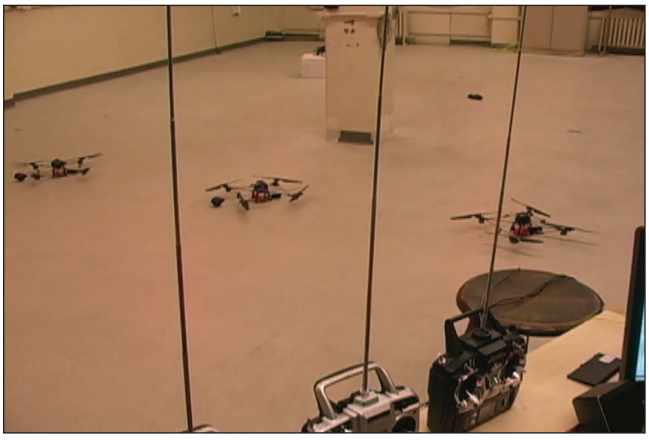


Figure 3. *Persistent search and track mission setup.*

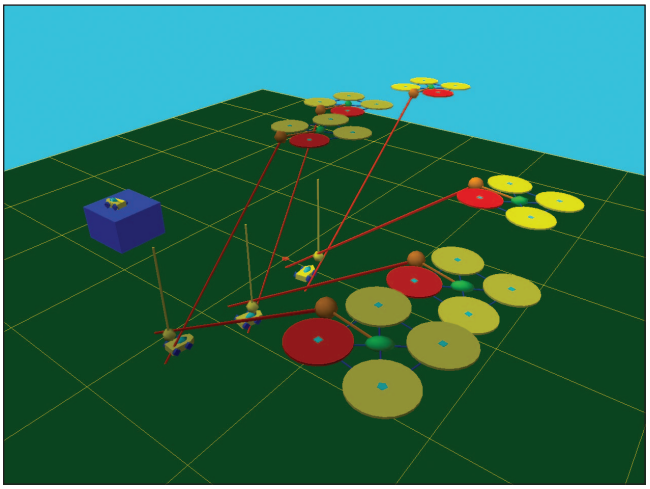


Figure 4. *Time-lapse image of one phase of the persistent mission showing cooperative tracking of a moving ground vehicle using two UAVs.*

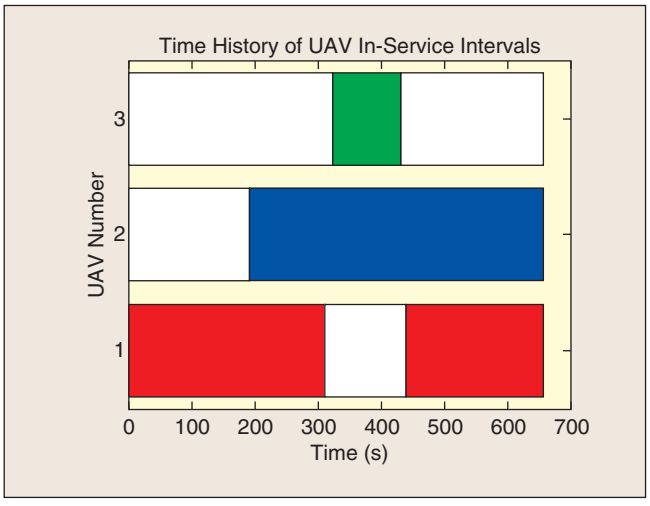


Figure 5. *Time history of the persistent surveillance mission. Colored blocks indicate times when that UAV was actively flying in support of accomplishing the mission.*

overall mission were able to be accomplished continuously over the course of the mission.

Conclusions

The health-aware task assignment algorithm developed in this article was demonstrated to be effective both in simulation and actual flight experiments. These initial results are very promising; however, more can be done in the health management problem in terms of accounting for other types of health states (sensor performance and control actuator failure modes). Furthermore, an important concept in the health management problem is to provide a robust performance in the face of uncertainty. Future work will focus on embedding more sophisticated stochastic models of numerous health states (including fuel usage and sensor performance) into the problem formulation and devising techniques to maximize performance while being robust to the uncertainty inherent in the problem.

Acknowledgments

We would like to thank Jim McGrew, Brandon Luders, Josh Redding, and Spencer Ahrens at the Massachusetts Institute of Technology (MIT), for their assistance in this research. Brett Bethke is sponsored by the Hertz Foundation and the American Society for Engineering Education. The research has been supported also by the Boeing Company and AFOSR grant FA9550-04-1-0458.

Keywords

UAVs, task assignment, health management, autonomous systems.

References

[1] Office of the Secretary of Defense. (2005). Unmanned Aircraft Systems Roadmap. [Online]. Available: http://www.acq.osd.mil/usd/Roadmap Final2.pdf

[2] H. Paruanak, S. Brueckner, and J. Odell, "Swarming coordination of multiple UAV's for collaborative sensing," in *Proc. 2nd AIAA Unmanned Unlimited Systems, Technologies, and Operations Aerospace Conf.*, San Diego, CA, Sept. 2003, p. 6525.

[3] P. Gaudiano, B. Shargel, E. Bonabeau, and B. Clough, "Control of UAV SWARMS: What the bugs can teach us," in *Proc. 2nd AIAA Unmanned Unlimited Systems, Technologies, and Operations Aerospace Conf.*, San Diego, CA, Sept. 2003, p. 6624.

[4] D. P. Bertsekas, *Dynamic Programming and Optimal Control*. Belmont, MA: Athena Scientific, 2000.

[5] J. C. Hartman and J. Ban, "The series-parallel replacement problem," *Robot. Comput. Integr. Manuf.*, vol. 18, nos. 3–4, pp. 215–221, 2002.

[6] D. Bertsimas and S. S. Patterson, "The air traffic flow management problem with enroute capacities," *Operat. Res.*, vol. 46, no. 3, pp. 406–422, 1998.

[7] K. Andersson, W. Hall, S. Atkins, and E. Feron, "Optimization-based analysis of collaborative airport arrival planning," *Transport. Sci.*, vol. 37, no. 4, pp. 422–433, 2003.

[8] J. Rosenberger, E. Johnson, and G. Nemhauser, "Rerouting aircraft for airline recovery," *Transport. Sci.*, vol. 37, no. 4, pp. 408–421, 2003.

[9] J. Bellingham, M. Tillerson, M. Alighanbari, and J. How, "Cooperative path planning for multiple UAVs in dynamic and uncertain environments," in *Proc. 41st IEEE Conf. Decision and Control*, Las Vegas, NV, Dec. 2002, vol. 3, pp. 2816–2822.

[10] C. Johnson, "Inverting the control ratio: human control of large autonomous teams," presented at the Int. Conf. on Autonomous Agents and Multiagent Systems: Workshop on Humans and Multi-Agent Systems Melbourne, Australia, July 2003.

[11] J. Wohletz, D. Castanon, and M. Curry, "Closed-loop control for joint air operations," in *Proc. American Control Conf.*, Arlington, VA, June 2007, vol. 6, pp. 4699–4704.

[12] M. Valenti, B. Bethke, J. How, D. Pucci de Farias, and J. Vian, "Embedding health management into mission tasking for UAV teams," presented at the American Controls Conf., New York, NY, June 2007.

[13] R. W. Beard, T. W. McLain, M. A. Goodrich, and E. P. Anderson, "Coordinated target assignment and intercept for unmanned air vehicles," *IEEE Trans. Robot. Automat.*, vol. 18, pp. 911–922, 2002.

[14] C. Schumacher, P. R. Chandler, S. J. Rasmussen, and D. Walker, "Task allocation for wide area search munitions with variable path length," in *Proc. 2003 American Control Conf.*, Denver, CO, June 2003, vol. 4, pp. 3472–3477.

[15] E. Frazzoli and F. Bullo, "Decentralized algorithms for vehicle routing in a stochastic time-varying environment," presented at the IEEE Conf. Decision and Control, Dec. 2004.

[16] M. Alighanbari, "Task assignment algorithms for teams of UAVs in dynamic environments," M.S. thesis, Dept. Aeronaut. Astronaut., Massachusetts Institute of Technology, Cambridge, 2004.

[17] M. Valenti, B. Bethke, G. Fiore, J. How, and E. Feron, "Indoor multi-vehicle flight testbed for fault detection, isolation, and recovery," in *Proc. AIAA Guidance, Navigation, and Control Conference and Exhibit*, Keystone, CO, Aug. 2006, p. 6200.

[18] B. Bethke, M. Valenti, and J. How, "Cooperative vision based estimation and tracking using multiple UAVs," in *Proc. Conf. Cooperative Control and Optimization*, Gainesville, FL, Jan. 2007, pp. 179–189.

[19] B. Bethke, "Persistent vision-based search and track using multiple UAVs," M.S. thesis, Massachusetts Institute of Technology, Cambridge, 2007.

Brett Bethke is a research assistant at the Aerospace Controls Laboratory at MIT, Cambridge. He received his bachelors degree in physics and aerospace engineering from MIT and a fellowship from the Hertz Foundation and the American Society for Engineering Education. In June 2007, he received his master's degree in aeronautics and astronautics and is currently pursuing his Ph.D. degree in the Aeronautics and Astronautics Department at MIT.

Mario Valenti received his bachelor's degree in electrical and mechanical engineering from the Temple University, Philadelphia, Pennsylvania, and his master's degree in electrical engineering and computer science from MIT, Cambridge. He graduated with a Ph.D. in electrical engineering and computer science from MIT in June 2007.

Jonathan P. How is a professor in the Department of Aeronautics and Astronautics, MIT, Cambridge, where he leads the Aerospace Controls Laboratory. He graduated from the University of Toronto with a B.Sc. in aerospace engineering and received the M.Sc. and Ph.D. degrees in aeronautics and astronautics from MIT in 1989 and 1993, respectively. He is a Senior Member of the IEEE.

Address for Correspondence: Jonathan P. How, Aerospace Controls Laboratory, Massachusetts Institute of Technology, 77 Massachusetts Ave, MIT Rm 33-326, Cambridge, MA 02139 USA. E-mail: jhow@mit.edu.