Increasing autonomy of UAVs

Citation	How, J.P. et al. “Increasing autonomy of UAVs.” Robotics & Automation Magazine, IEEE 16.2 (2009): 43-51. © 2009 Institute of Electrical and Electronics Engineers
As Published	http://dx.doi.org/10.1109/MRA.2009.932530
Publisher	Institute of Electrical and Electronics Engineers
Version	Final published version
Citable link	http://hdl.handle.net/1721.1/51874
Terms of Use	Article is made available in accordance with the publisher's policy and may be subject to US copyright law. Please refer to the publisher's site for terms of use.

Unmanned aerial vehicles (UAVs) are acquiring an increased level of autonomy as more complex mission scenarios are envisioned [1]. For example, UAVs are being used for intelligence, surveillance, and reconnaissance missions as well as to assist humans in the detection and localization of wildfires [2], tracking of moving vehicles along roads [3], [4], and performing border patrol missions [5]. A critical component for networks of autonomous vehicles is the ability to detect and localize targets of interest in a dynamic and unknown environment. The success of these missions hinges on the ability of the algorithms to appropriately handle the uncertainty in the information of the dynamic environment and the ability to cope with the potentially large amounts of communicated data that will need to be broadcast to synchronize information across networks of vehicles. Because of their relative simplicity, centralized mission management algorithms have previously been developed to create a conflict-free task assignment (TA) across all vehicles. However, these algorithms are often slow to react to changes in the fleet and environment and require high bandwidth communication to ensure a consistent situational awareness (SA) from distributed sensors and also to transmit detailed plans back to those sensors.

More recently, decentralized decision-making algorithms have been proposed [6]–[8] that reduce the amount of communication required between agents and improve the robustness and reactive ability of the overall system to bandwidth limitations and fleet, mission, and environmental variations. These methods focus on individual agents generating and maintaining their own SA and TA, relying on periodic intervehicle

© U.S. AIR FORCE PHOTO/MASTER SGT. ROBERT W. VALENCA

Digital Object Identifier 10.1109/MRA.2009.932530

Increasing Autonomy of UAVs

Decentralized CSAT Mission Management Algorithm

BY JONATHAN P. HOW, CAMERON FRASER, KARL C. KULLING, LUCA F. BERTUCCELLI, OLIVIER TOUPET, LUC BRUNET, ABE BACHRACH, AND NICHOLAS ROY

1070-9932/09/$25.00©2009 IEEE

Unmanned aerial vehicles are acquiring an increased level of autonomy as more complex mission scenarios are envisioned.

communication to resolve conflicts. Although many researchers [9]–[11] have examined algorithms for distributed TA problems, few (such as [12]) have included strong experimental results demonstrating the implementation of these algorithms in tightly coupled missions such as search and track applications. Some of the key challenges posed by these missions include the interdependence between the vehicles, the tasks, the high data rates required to maintain a good estimate of the target state, and broad area search. Much of the previous work has primarily emphasized one aspect of the mission (such as search [13], [14] or track [15], [16]), but little work has addressed a more synergistic combination of the two operating modes. A Bayesian framework for search and track was developed in [17] but did not consider the multivehicle task allocation problem. Elston and Frew [18] developed a hierarchical approach to the coordinated search and track mission, but based their revisit times on mean target motion, rather than explicitly propagating the entire covariance matrix.

The cooperative search, acquisition, and track (CSAT) mission requires an allocation of UAV assets to the potentially conflicting objectives of searching and tracking. Although the searching component encourages exploration of the environment and maximizes the probability of finding unknown targets, the tracking objective requires a vehicle to persistently focus on a single target. A successful mission will necessarily tradeoff between these two modes, because it is generally undesirable to be in only a search or track mode throughout the course of the mission. Striking the right balance between these two objectives is of key importance for overall mission effectiveness, since search must be performed throughout the course of the entire mission. One of the key results of this article is that the dynamic transition between searching and tracking can arise naturally from the problem specification rather than being a behavior that is artificially encoded in the problem statement.

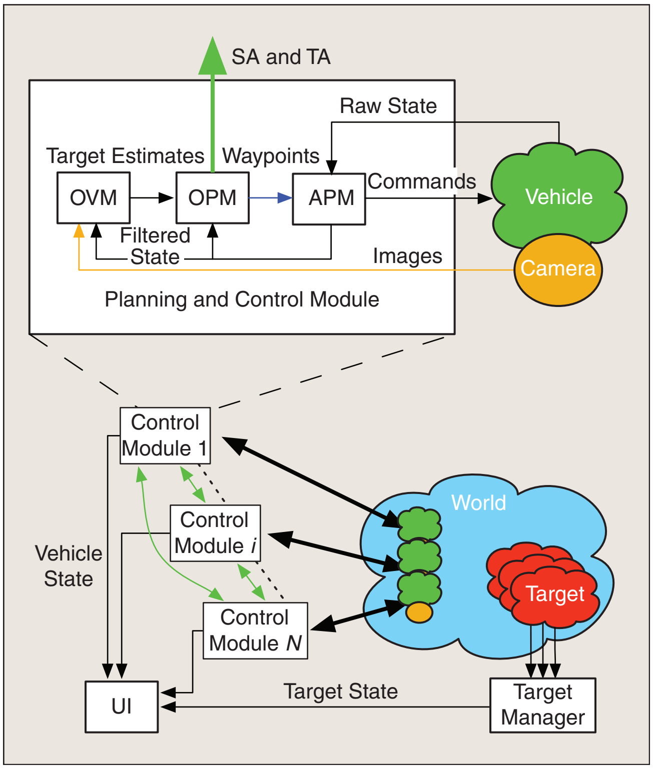


Figure 1. *CSAT architecture block diagram.*

This article presents the results of a collaborative effort between Massachusetts Institute of Technology (MIT) and Aurora Flight Sciences to develop a new integrated architecture that combines search and track and solves a challenging multivehicle, dynamic target resource allocation problem in real time and in the presence of uncertainty. Thus, multiple UAVs are used to collaboratively search the environment and keep track of any targets found. UAVs persist in a baseline search mode, and the transition to a tracking mode, either once a new target is detected or when the uncertainty in a target that has been previously detected (but not currently being tracked), exceeds a desired threshold. This approach uses true-to-life vision in the loop to task a fleet of UAVs to efficiently search for and track a large number of dynamic targets. The experimental demonstrations of the system are conducted in the real-time indoor autonomous vehicle test environment (RAVEN) testbed [19], [20] that is part of the MIT Aerospace Controls Laboratory. The following sections provide a richer description of the underlying architecture of our CSAT implementation.

Architecture

To achieve the goal of a decentralized search and track mission, the CSAT algorithm has been divided into a number of modules that communicate with each other over a network. Figure 1 shows the architecture for several vehicles and several targets. Each vehicle runs three modules onboard [the onboard vision module (OVM), the onboard planning module (OPM), and the autopilot module (APM)], which work together to perform the sensing, planning, and control of each vehicle. The OVM takes in images from the vehicle's camera and provides a state estimate to the OPM for each target that it detects in the image. The OPM then either updates its own target state estimate based on the new information or propagates the estimate and uncertainty of the target if no measurements are available. If the vehicle does not find the target after a planned revisit, then the OPM reverts back to searching for the target but aided by the knowledge of the target's last known position and velocity. The OPM uses these estimates and creates a plan for the vehicle to search for or track targets as appropriate. The waypoints generated by the OPM are sent to the APM, which implements the plan by interfacing to the low-level controller on the flight vehicle. The APM also manages the vehicle's state estimate and distributes it to the other modules as necessary. The target manager (TM) module generates commands for the targets and gathers their actual state information (rather than the estimate provided by the OVM), which are used as truth data for display in the user interface (UI) and for subsequent analysis. The UI receives data from the OPM, APM, and TM and displays it in a bird's-eye view of the operations area.

For each new vehicle, an additional instance of the OVM, OPM, and APM is executed. Since the vehicles are operating in a distributed manner, each UAV will only have direct access to its local information, which can lead to conflicts in the SA. To mitigate the effects of conflicting information, the OPMs on the vehicles communicate with each other to achieve global consensus on their plans and to coordinate their search and track efforts. This setup not only allows each vehicle to run its own algorithms but allows separation of functions within the vehicle itself. For example, the high-level planner is a separate module from the low-level autopilot. The various modules communicate with each other over a transmission control protocol (TCP)/Internet protocol (IP) network, which allows the modules to run on separate computers or simply as separate processes on the same computer. This modular approach adds additional robustness to the system, allowing for the overall system to continue to execute the mission even if one or many modules fail.

Onboard Planning Module

The OPM is the vehicle's high-level path and task planner. It assigns the vehicle to search for targets (whose number and location are unknown until they are found) or to track known targets. Whenever a vehicle is searching, the OPM uses a set of probability distributions to guide the vehicle along an optimal search path that maximizes the likelihood of detecting a target [13], [21]–[24]. Once a target is found and classified, it must then be periodically revisited to maintain an up-to-date estimate of its position and velocity [25], [26]. Between revisits, the OPM determines whether other tasks need to be executed or if the vehicle should resume searching the local area. This decision is made by a decentralized TA algorithm [7], [8] that continuously runs within the OPM on each vehicle and ensures that as many tasks as possible are executed without conflicting assignments. The following will first describe the search and track behaviors followed by how the two are allocated by the tasking algorithm within the OPM.

Search

To be able to search effectively [13], [21]–[24], the OPM maintains information about what areas of the operational map have previously been observed. It does this by maintaining a set of probability maps, where each cell (x, y) in the map M_i has a probability of containing a target $P_t^i(x, y)$, at some time t. (All $P_t^i(x, y) = 0$ for (x, y) in an obstacle.) A generic map is maintained for each potential target environment encountered, such as land or water environments, and is initialized to represent any a priori knowledge of the unknown targets' position distributions. In addition, a new map is generated for each target that had been found at one point but has since been lost. This approach assumes that there may always be at least one additional undiscovered target in each environment type beyond those that have already been found. Thus, the generic search map provides the OPM with a constant incentive to continue searching the environment even if many targets have already been discovered and are successfully being tracked, ensuring that the UAVs maintain a nominal level of exploration of the environment.

Multiple UAVs are used to collaboratively search the environment and keep track of any targets found.

When in search mode, the OPM uses the combined probability maps to determine a finite-horizon path that maximizes the sum of the probabilities that the sensor footprint will cover. This path generation scheme is based on a breadth-first tree search with limited depth and turn constraints and includes not only the search path but also the route to the next task, if applicable. If other UAVs are in the area, they will hierarchically coordinate their search paths so that they avoid searching the same area twice.

If a previously discovered target is lost, it is converted to a search target with an associated, newly created search map. This map is initialized with a nonzero probability, only within an estimated reachable region based on the target's last known position and velocity and is thereafter propagated based on the last estimate of the target's speed through the equations given later. To account for dynamic search targets that have not been found yet, each generic search map is associated with a phantom search target with its own velocity estimate to be used to determine how quickly uncertainty diffuses back into previously explored space. The probability diffusion update for each cell (x, y) in map M_i at time t is given by the two-stage process

$$\begin{aligned} P_t^i(x, y) = {} & (P_{t-1}^i(x, y) + P_{t-1}^i(x', y') \\ & \times P(x, y | x', y', v_i))(1 - sc), \end{aligned} \qquad (1)$$

$$P_t^i(x, y) \leftarrow \frac{P_t^i(x, y)}{\sum\limits_{\hat{x}, \hat{y} \in M_i} P_t^i(\hat{x}, \hat{y})}, \qquad (2)$$

where $P(x, y | x', y', v_i)$ is the probability that a target transitions from (x', y') to (x, y) given the search target velocity v_i, and sc is the percentage of the cell that is covered by the sensor at time t. The first equation is the diffusion process while the second ensures that the sum of the probabilities is unity. By diffusing the probabilities as such, the OPM promotes searching for dynamic targets that may have traveled back into previously explored territory.

Track

Once a target is successfully detected by the vision module, it is classified and tracked by a UAV that has the required capability. The OVM provides target state measurements to update the OPM's internal estimator. In our experiments, a Kalman filter was used under the assumption that the dynamic system is linear [25]. Target tracks are maintained by recursively updating the state estimates $\hat{X}_{k+1|k}$ using a kinematic model A

of the target motion with additive noise $w_k \sim \mathcal{N}(0, Q)$ to capture any unmodeled dynamics because of this simplification

$$\begin{aligned}\text{True Model: } & X_{k+1} = AX_k + Bw_k,\\ \text{Estimate: } & \hat{X}_{k+1|k} = A\hat{X}_{k|k}.\end{aligned} \tag{3}$$

Once the uncertainty in the target estimate has been reduced through tracking measurements, the UAV can temporarily leave the target to execute other tasks and be confident that the target will be relocated upon its return. This permits each vehicle to complete multiple tasks even if the number of available tasks exceeds the number of capable agents.

In order to determine the necessary revisit time, Algorithm 1 is run for each track-capable UAV (because of possibly differing sensor footprints). The target estimate and error covariance are both propagated forward until a scaled representation of the covariance ellipse no longer can be contained within the vehicle's sensor footprint. (This forward propagation uses the existing process noise to propagate the error covariance by recursively using the prediction step of the Kalman filter and is measurement independent.) The scaling multiplier used, denoted n_σ in Algorithm 1, can be thought to represent a desired confidence level on finding the target at the revisit time and location, with higher values leading to more conservative revisit times. UAVs with different sensor footprints will in general predict a unique revisit time based on their physical sensor properties. A track task is then created to visit the target at the revisit time and the target's propagated position using a vehicle capable of tracking.

When a vehicle is assigned to track a target, the OPM generates a path that coordinates the vehicle arrival time at the revisit location to match the predicted revisit time. Upon UAV arrival at the desired location, the target may or may not be within the UAV's field of view. In the first case, the vehicle overflies the intended target for a predetermined time, keeping the target in its field of view and updates its position and velocity estimates. The specified track time is an empirically chosen value that was determined to be long enough to obtain a reliable state estimate of the target, though both the tracking trajectory and duration can be modified to incorporate any desired tracking algorithm. In the second case, however, the target is declared lost and a new search has to be initialized, as described in the "Search" subsection of the "Onboard Planning Module" section. If the target is once again found, then the search probability map is removed and a new revisit location and time is calculated.

Algorithm 1. Revisit time and location calculation

```
$k \leftarrow 0$
Initialize process noise covariance $Q$
Initialize state estimate $\hat{X}_k \leftarrow \hat{X}_0$
Initialize error covariance $P_k \leftarrow P_0$
Initialize characteristic size $\varphi_k \leftarrow \pi\sqrt{|P_k|}$
while $n_\sigma \varphi_k < \varphi_{\text{sensor}}$ do
    $\hat{X}_{k+1} \leftarrow A\hat{X}_k$
    $P_{k+1} \leftarrow AP_kA^T + BQB^T$
    $\varphi_{k+1} \leftarrow \pi\sqrt{|P_{k+1}|}$
    $k \leftarrow k + 1$
end while
return Revisit time $= k\Delta t$
```

TA Algorithm

Given the updated search probabilities maps and target estimates, the OPM can then decide whether to search regions of the map that have a high likelihood of containing targets or execute existing track tasks. In our framework, search is considered a spare time strategy rather than a task. This means that the vehicles search for targets when they are not assigned to a track task or when their next track task is far enough in the future that they can search in the intervening time. This approach assumes that, given the choice between keeping track of a known, nontrivial target and searching, it is more beneficial to follow the targets that have already been found.

The OPM uses a modification of a multiagent TA algorithm introduced in [7] and [8], called the consensus-based bundle algorithm (CBBA). CBBA is a cooperative, low-communication-bandwidth iterative auction approach that uses two phases to achieve a conflict-free TA. In the first phase, each vehicle generates a single-ordered bundle of tasks by greedily selecting tasks for itself. The second phase resolves inconsistent or conflicting assignments through heuristic methods and improves the global reward through the bidding process. The implementation of CBBA in the OPM executes these two phases continuously and concurrently at each time step, allowing the algorithm to rapidly adjust to changes in the network and environment. See [7] and [8] for additional details.

Onboard Vision Module

Images captured from the onboard camera are loaded and analyzed by the vision processing unit using OpenCV [27]. The received image is converted from a red-green-blue (RGB) format to a hue-lightness-saturation (HLS) format for easier color separation. Then, given the expected color ranges for each target, a detection algorithm determines which pixels fall within the range of colors for each target, and a smoothing function is applied to each color blob to locate its centroid. The location of the target in the image plane is then projected to the inertial world frame using a calibrated pinhole camera model, assuming that targets exist on the ground plane ($z = 0$). This estimate of the targets' locations is then fed into a particle filter to smooth the measurement before transmission to the OPM.

Target state estimation relies on principles from particle filtering [28]. Upon receiving a (noisy) measurement of the location of the target in the inertial world frame, each particle's location is updated using the kinematic motion model of the target. The particles are then reweighted based on their distance from the target location as measured from the camera image, and importance resampling is performed on the set of particles [28]. Particles with low weight are rejected, and new particles are generated. At this point, the set of particles should approximate the distribution of possible target states, and the weighted mean value is transmitted to the OPM as the new target measurement.

Autopilot Module

The APM acts as the interface between the low-level vehicle controller and the rest of the CSAT architecture. In simulation modes, it also simulates vehicle dynamics. Specifically, the autopilot maintains the vehicle state estimate, provides guidance to fly the vehicle along the waypoints provided by the OPM, and monitors the health of the vehicle, including fuel status.

The APM's open architecture allows it to accept state estimate input from various sources depending on the situation. For example, in simulation, it simply takes the state estimate from the simulated dynamics, while in flight experiments it might use state estimates from onboard or offboard sensors. The APM can also perform additional filtering on the state information. This state estimate is then distributed to the other modules that need the estimate, including the OPM and the OVM, for use in planning and target estimation.

The APM periodically receives a list of waypoints from the OPM that describe the planned path over a short time horizon. If the APM is using simulated dynamics, it generates appropriate steering commands using the nonlinear control law developed in [29]. If a separate vehicle controller with waypoint following ability is used, such as in the RAVEN testbed, then the APM only keeps track of which waypoint the vehicle should fly to next using the logic developed in [30] and sends that waypoint to the vehicle controller. Waypoints are specified as a position, a time at which the vehicle should reach that location, and a type, such as flyby, flyover, or stop, which specifies when the vehicle can continue on to the next waypoint.

Flight Experiments

Indoor Flight Tests

The RAVEN System

Experimental trials of the presented CSAT formulation were conducted in the MIT's RAVEN [19], [20], a multivehicle platform allowing for rapid prototyping of high-level mission management algorithms. This capability is achieved by using a very accurate Vicon MX motion capture system [31] to produce high bandwidth state estimates of numerous aerial and ground vehicles, as well as in-house vehicle controllers to provide low-level control and stabilization of the vehicle hardware.

The motion capture system detects lightweight reflective dots on the vehicles and uses these to calculate the vehicles' position and orientation within the 25 by 30 ft test room. This data is transmitted via Ethernet to each vehicle's ground-based control computer, which in turn commands its vehicle through a commercial off-the-shelf (COTS) radio control (R/C) transmitter [19], [20]. Along those same lines, the OPM and OVM modules are also run offboard, allowing the use of COTS vehicle hardware with minimal requirements for onboard computational capacity. This offboard computation replicates the exact type of computation that would be performed onboard each vehicle, and it is performed offboard simply to ease the integration process given the payload restrictions of the current vehicles.

Test Vehicles

While the RAVEN testbed allows for a wide range of vehicle types to be used, the room space constraints and prior proven vehicle performance led us to select the Hummingbird quad rotor produced by Ascending Technologies [32] as the aerial vehicles (see Figure 2). The particular model used can stabilize the vehicle attitude using onboard sensors and microcontrollers while an associated midlevel control program running on one of the RAVEN vehicle computers generates attitude commands to control the position of the vehicle. This vehicle wrapper code is the link between the Vicon motion capture state estimates, the CSAT APM, and the vehicle itself. The wrapper implements a simple linear quadratic regulator (LQR) controller to follow a reference trajectory generated by an internal waypoint follower [20]. The APM sends activate, takeoff, land, and waypoint commands to the wrapper, which are then converted to the relevant control signals to send to the vehicle.

The Hummingbirds are modified with the COTS wireless fidelity (Wi-Fi) enabled Panasonic BL-C131A network camera to provide the vehicles with visual detection capabilities. The camera is networked with the OVM module of the CSAT framework. The internal mechanisms to control the pan and the tilt of this camera are removed to reduce the overall payload weight, making the current hardware configuration to constrain the cameras to a fixed orientation looking vertically downward from the vehicle. This is a 1/6 in CMOS sensor, with approximately 320,000 pixels that provides a footprint of 1.0 m width by 0.75 m height at an altitude of 1.2 m.

Results

This section presents some the results of the CSAT architecture flown in RAVEN. The indoor flight experiments used five targets and three UAVs. The number and composition of vehicles for the flight tests is limited only by the physical size of our indoor test environment; the planning algorithm can return

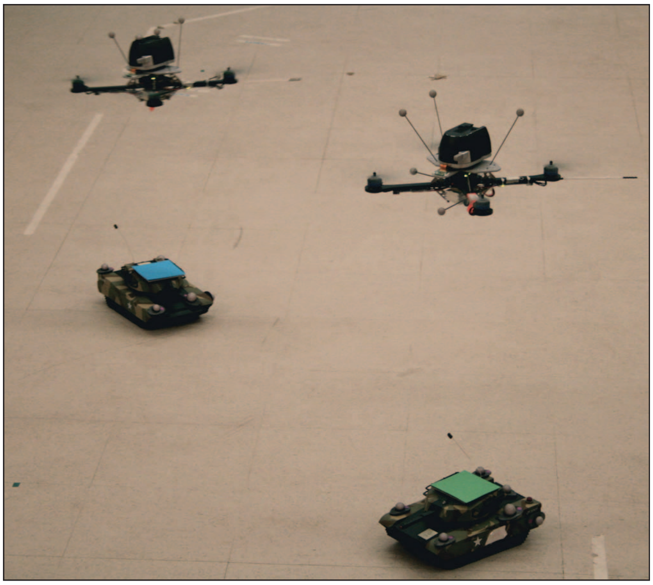


Figure 2. *Hummingbird UAVs performing a coordinated search and track task on tank targets.*

assignments with as many as ten UAVs and 50 waypoints in real time (in the order of seconds).

To scale the experiment to the size of the environment, the UAVs were flown with a maximum speed of 0.2 m/s at an altitude of 1.2 m. Some of the targets were stationary, while the dynamic targets were controlled autonomously and by R/C and designed for a nominal speed of 0.05 m/s. All were assigned the same score for the tests in Scenario I, while in Scenario II, one target was designated a higher value than the others.

Scenario I: Effect of Estimated Noise Covariance *Q*

This scenario considered a set of three UAVs and five target hardware experiments in which the goal was to investigate the sensitivity of the CSAT planner to the choice of the process noise covariance Q. Under the assumption of a target motion model driven by process noise, the noise covariance is used to determine the revisit time for the tracking exercise. The process noise covariance is an effective tuning parameter for the CSAT controller that must be chosen carefully, based on anticipated target maneuverability. Thus, the true Q will inherently vary between vehicles, specifically between static and mobile targets, according to the vehicle dynamics.

Two choices of Q were made in this scenario resulting in two distinct experiments (the scaling factor $n_\sigma = 1$ for both experiments). The results are shown in Figure 3. Figure 3(a) shows experiments with low covariance ($Q = 0.001 \times I_{2\times 2}\,\mathrm{m}^2/\mathrm{s}^4$) for all targets, and Figure 3(b) shows experiments with high covariance ($Q = 0.05 \times I_{2\times 2}\,\mathrm{m}^2/\mathrm{s}^4$) for all targets. In the first experiment, a low value of Q was used, implying that the targets being tracked were assumed not to be very maneuverable. Because of this, the UAVs track the targets occasionally, but spend most of their time searching. Consequently, the cumulative area searched quickly approaches and eventually reaches 100%. Conversely, in the second experiment, where a high value of Q was used, the motion of the targets is assumed to be very uncertain and, consequently, the targets must be revisited often. Once a UAV finds a target, it spends most of its time in tracking mode and only has a few seconds of search between each revisit. This is reflected in the cumulative search area, which quickly plateaus once the first UAV begins tracking and never reaches the same level as in the low Q experiment. Furthermore, because the UAVs in the high Q situation have so little time to search between revisits, they can never wander far from their assigned targets. As a result, they will have difficulty finding any new targets entering the operations area. Essentially, the area searched after tracking has begun is much lower in the second case than in the first.

Even in the restricted testing space, we have been able to experimentally show that a small change in the choice of the process noise Q can lead to significant variations in the tracking strategies. In particular, if the target dynamics is unknown but likely very erratic, as would be the case with highly maneuverable targets, it is probably a good choice to select a high Q and emphasize tracking

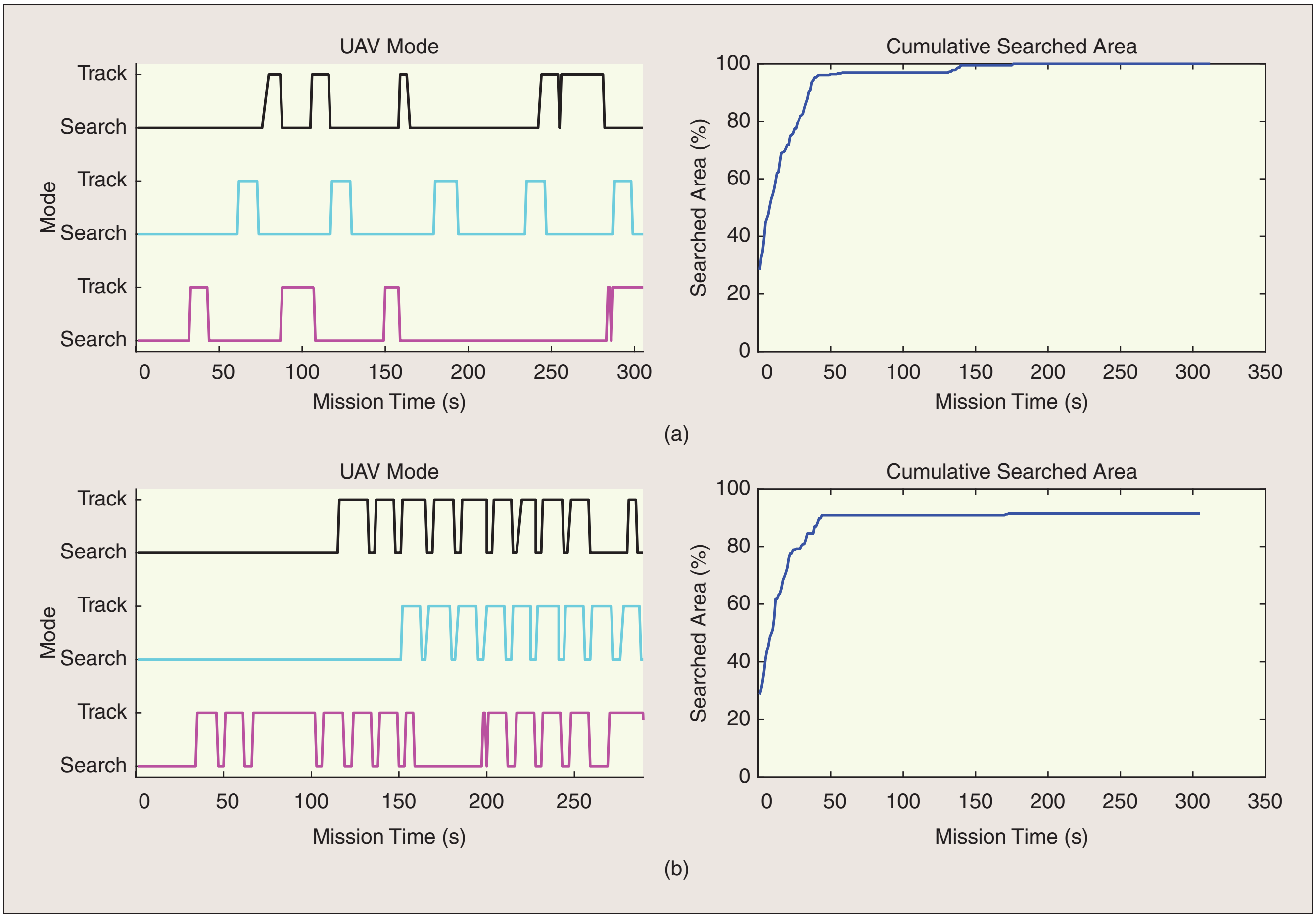


Figure 3. UAV mode and search area comparison with (a) low Q value and (b) high Q value.

at the expense of search performance. Conversely, if the choice of the dynamic model is assumed very accurate or the expected targets are not very maneuverable, then a matching low Q value will allow the agents to execute other tasks while proceeding with the track. Thus, it is very beneficial to the overall execution of the CSAT mission to choose a representative Q for each target to best tradeoff between the two competing tasks.

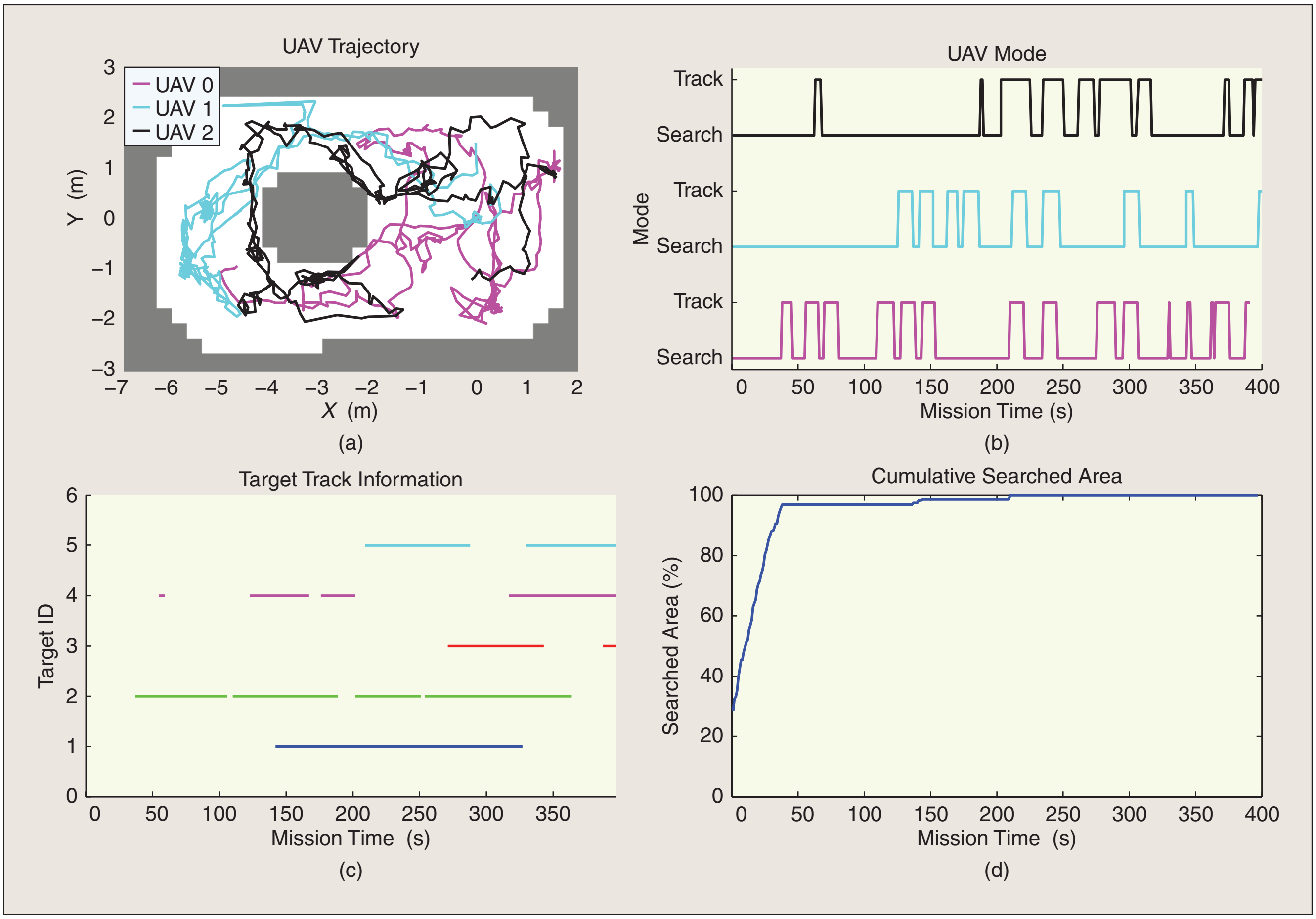
The CSAT mission requires an allocation of UAV assets to the potentially conflicting objectives of searching and tracking.

Scenario II: Three UAVs and Five Targets

This scenario demonstrated a multivehicle, multitarget mission with three autonomous UAVs and five targets (two of which were dynamic). Using results from the first scenario as a guide, the process noise covariance matrices were set to $Q = 0.001 \times I_{2\times 2}$ m^2/s^4 for any target with measured velocity less than 0.005 m/s, and $Q = 0.05 \times I_{2\times 2}$ m^2/s^4 for those with velocities over 0.005 m/s. Also, Target 2 (green) was given a higher tracking score and desired confidence level ($n_\sigma > 1$), specifying that it is a high-priority target that the agents need to track if found. The CSAT planner's search map was initialized with a uniform prior distribution of target locations. All vehicles in this experiment were eligible to both search and track targets.

Figure 4 shows a summary of the mission. In general, this mission shows a good balance between searching and tracking, as well as alternating between tracking erratic dynamic targets and static targets. Figure 4(a) shows the trajectory of the three UAVs (the apparent noise in the paths is due to perturbations from the downwash effects of the multiple UAVs rather than the algorithm). We can see from the overlapping trajectories that the UAVs are using a fluid search method rather than partitioning the operations area or flying fixed Zamboni patterns. The advantage of this approach is that the team is inherently flexible and the agents can explore regions of high uncertainty regardless of their location, as opposed to remaining constrained to local areas or inefficient search paths. Additionally, as agents are sent to complete other tasks, they can handoff their original search regions to the remaining nearby searching vehicles.

Figure 4(b) shows the mode (search or track) of each UAV. Since the environment has a mix of dynamic targets (with high covariance) and static targets (with low covariance), the UAVs exhibit both short and long revisit times. This permits agents to naturally execute target handoffs if one agent needs to



(a)

(b)

(c)

(d)

Figure 4. CSAT mission output.

service another target or refuel. This plot also demonstrates the natural shift of focus from primarily searching early on to primarily tracking as more targets are found.

Figure 4(c) and (d) demonstrates this shift even more clearly. In Figure 4(d), it can be seen that the UAVs have enough time to search, despite the track tasks, close to 100% of the map. Figure 4(c) displays a line when the target estimate of each target is within a specified threshold of the true position. All five targets were found, tracked, and revisited during the mission. As was intended, UAV 2 (in green) is tracked more frequently because of its higher priority and higher desired revisit confidence. This results in the UAVs maintaining a better estimate of that target than of the other targets, though the agents were all able to contribute to the search and track components for the duration of the mission.

Conclusions

This article has presented a tightly integrated systems architecture for a decentralized CSAT mission management algorithm and demonstrated successful implementation in actual hardware flight tests. This CSAT architecture allows each UAV to accomplish a combined search and track mission by conceptualizing the searching aspect as a spare time strategy to be executed optimally over a short time horizon when the agents are not actively tracking a vehicle. This presented a balance between the two conflicting search and track modes and allowed the mission to achieve more than simply searching or tracking alone.

Using the RAVEN testbed, we demonstrated this system in action using three agents and both static and dynamic targets. The first set of tests showed the importance of accurately representing the expected maneuverability of the targets in the revisit time calculation. The results demonstrated that the presented system achieves a successful balance between searching unexplored regions of the environment and tracking known targets. The second set of flight tests demonstrated a mission using static and dynamic targets with varying levels of uncertainty about the target models and nonuniform scoring. This second mission demonstrated the flexibility of the presented CSAT formulation and its ability to deal with various search and track requirements even in environments where the track targets outnumber the sensing vehicles.

Our immediate future work is dedicated to the identification of the process noise covariance Q using adaptive estimation techniques so that it does not need to be specified a priori. Also, we intend to extend the OVM to account for target recognition using other features such as target shape or velocity.

Acknowledgments

This research has been supported by Aurora Flight Sciences through Air Force Office of Scientific Research Small Business Technology Transfer Program (AFOSR STTR) contract FA9550-06-C-0088 (Program Manager Dr. Jim Paduano) by MIT AFOSR grant FA9550-08-1-0086 and by the Boeing Company (Program Manager Dr. John Vian at the Boeing Phantom Works).

Keywords

Unmanned aircraft systems, cooperative control, coordinated search and track, experimental demonstrations.

References

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

[2] D. W. Casbeer, D. B. Kingston, R. W. Beard, T. W. McLain, S. M. Li, and R. Mehra, "Cooperative forest fire surveillance using a team of small unmanned air vehicles," *Int. J. Syst. Sci.*, vol. 37, no. 6, pp. 351–360, 2006.

[3] E. Frew, T. McGee, Z. Kim, X. Xiao, S. Jackson, M. Morimoto, S. Rathinam, J. Padial, and R. Sengupta, "Vision-based road-following using a small autonomous aircraft," in *Proc. 2004 IEEE Aerospace Conf.*, Big Sky, MT, Mar. 2004, pp. 3006–3015.

[4] M. Quigley, M. Goodrich, S. Griffiths, A. Eldgredge, and R. Beard, "Target acquisition, localization, and surveillance using a fixed-wing mini-UAV and gimbaled camera," in *Proc. 2005 IEEE Int. Conf. Robotics and Automation*, Barcelona, Spain, Apr. 2005, pp. 2600–2605.

[5] A. Girard, A. Howell, and K. Hedrick, "Border patrol and surveillance missions using multiple unmanned air vehicles," in *Proc. IEEE Conf. Decision and Control*, 2004, pp. 620–625.

[6] M. Alighanbari and J. How, "Decentralized task assignment for unmanned aerial vehicles," in *Proc. European Control Conf. Decision and Control CDC-ECC*, 2005, pp. 5668–5673.

[7] L. Brunet, H. L. Choi, J. P. How, and M. Alighanbari, "Consensus-based auction approaches for decentralized task assignment," in *Proc. AIAA Conf. Guidance, Navigation and Control*, 2008, Paper 2008-6839.

[8] H.-L. Choi, L. Brunet, and J. P. How, "Consensus-based decentralized auctions for robust task allocation," *IEEE Trans. Robot.*, to be published.

[9] D. Dionne and C. A. Rabbath, "Multi-UAV decentralized task allocation with intermittent communications: The DTC algorithm," in *Proc. Amer. Control Conf.*, 2007, pp. 5406–5411.

[10] M. Flint, T. Khovanova, and M. Curry, "Decentralized control using global optimization," in *Proc. AIAA*, 2007, Paper 2007-2906.

[11] A. Ryan, J. Tisdale, M. Godwin, D. Coatta, D. Nguyen, S. Spry, R. Sengupta, and J. K. Hedrick, "Decentralized control of unmanned aerial vehicle collaborative sensing missions," in *Proc. Amer. Control Conf.*, 2007, pp. 4672–4677.

[12] B. Grocholsky, J. Kellar, V. Kumar, and G. Pappas, "Cooperative air and ground surveillance," *IEEE Robot. Automat. Mag.*, vol. 13, no. 3, pp. 16–25, 2006.

[13] W. Wong, F. Bourgault, and T. Furukawa, "Multi-vehicle Bayesian search for multiple lost targets," in *Proc. IEEE Int. Conf. Robotics and Automation*, 2005, pp. 3169–3174.

[14] Y. Yang, A. Minai, and M. Polycarpou, "Evidential map-building approaches for multi-UAV cooperative search," in *Proc. Amer. Control Conf.*, 2005, pp. 116–121.

[15] P. Yang, R. Freeman, and K. Lynch, "Distributed cooperative active sensing using consensus filters," in *Proc. IEEE Int. Conf. Robotics and Automation*, 2007, pp. 405–410.

[16] A. Makarenko and H. Durrant-Whyte, "Decentralized data fusion and control in active sensor networks," in *Proc. Int. Conf. Information Fusion*, 2004.

[17] T. Furukawa, F. Bourgault, B. Lavis, and H. Durrant-Whyte, "Recursive Bayesian search-and-tracking using coordinated UAVs for lost targets," in *Proc. 2006 IEEE Int. Conf. Robotics and Automation, 2006 (ICRA 2006)*, May 2006, pp. 2521–2526.

[18] J. Elston and E. Frew, "Hierarchical distributed control for search and tracking by heterogeneous aerial robot networks," in *Proc. IEEE Int. Conf. Robotics and Automation (ICRA 2008)*, 2008, pp. 170–175.

[19] 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 Conf. and Exhibit*, Keystone, CO, Aug. 2006, Paper 2006-6200.

[20] J. How, B. Bethke, A. Frank, D. Dale, and J. Vian, "Real-time indoor autonomous vehicle test environment," *IEEE Control Syst. Mag.*, vol. 28, no. 2, pp. 51–64, Apr. 2008.

[21] F. Bourgault, T. Furukawa, and H. F. Durrant-Whyte, "Process model, constraints, and the coordinated search strategy," in *Proc. IEEE Int. Conf. Robotics and Automation*, 2004, pp. 5256–5261.

[22] F. Bourgault, T. Furukawa, and H. F. Durrant-Whyte, “Decentralized Bayesian negotiation for cooperative search,” in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, 2004, pp. 2681–2686.

[23] S. Brown, “Optimal search for a moving target in discrete time and space,” *Oper. Res.*, vol. 28, no. 6, pp. 1275–1289, 1980.

[24] L. D. Stone, *Theory of Optimal Search*, vol. 118, *Mathematics in Science and Engineering*. New York: Academic Press, 1975.

[25] Y. B. Shalom, X. R. Li, and T. Kirubarajan, *Estimation with Applications to Tracking and Navigation*. New York: Wiley Interscience, 2001.

[26] T. Kirubarajan, Y. Bar-Shalom, K. R. Pattipati, I. Kadar, B. Abrams, and E. Eadan, “Tracking ground targets with road constraints using an IMM estimator,” in *Proc. IEEE Aerospace Conf.*, 1998, pp. 5–12.

[27] Intel Corporation. (2007). OpenCV computer vision library [Online]. Available: http://www.intel.com/technology/computing/opencv/

[28] S. Arulampalam, N. Gordon, and B. Ristic, *Beyond the Kalman Filter: Particle Filters for Tracking Applications*. Boston, MA: Artech House, 2004.

[29] S. Park, J. Deyst, and J. P. How, “Performance and Lyapunov stability of a nonlinear path following guidance method,” *AIAA J. Guid., Control Dyn.*, vol. 30, no. 6, pp. 1718–1728, Nov. 2007.

[30] G. Ducard, K. Kulling, and H. P. Geering, “A simple and adaptive online path planning system for a UAV,” in *Proc. 15th Mediterranean Conf. Control and Automation*, 2007, pp. 1–6.

[31] Vicon(2006, July).Vicon MX systems [Online]. Available: http://www.vicon.com/products/viconmx.html

[32] A. Technologies (2008). Asctec Hummingbird helicopter [Online]. Available: http://www.asctec.de

Jonathan P. How received his B.A.Sc. degree from the University of Toronto in 1987 and his S.M. and Ph.D. degrees in aeronautics and astronautics from MIT in 1990 and 1993, respectively. He then studied for two years at MIT as a postdoctoral associate for the Middeck Active Control Experiment (MACE) that flew onboard the Space Shuttle Endeavour in March 1995. Before joining MIT in 2000, he was an assistant professor in the Department of Aeronautics and Astronautics at Stanford University. He is currently a professor in the Department of Aeronautics and Astronautics at MIT. Current research interests include robust coordination and control of autonomous vehicles in dynamic uncertain environments. He was the recipient of the 2002 Institute of Navigation Burka Award, an associate fellow of AIAA, and a Senior Member of the IEEE.

Cameron Fraser received his M.Sc. degree in the Aeronautics and Astronautics Department at MIT and bachelors of applied science from the University of Toronto in the Aerospace option of the Division of Engineering Science in 2007. Before working with the Aerospace Controls Lab at MIT, he spent two summers developing tidal energy technology with the Naval Architecture Lab at UBC on a grant from the Natural Sciences and Engineering Research Council of Canada.

Karl C. Kulling received his M.Sc. degree in the Aeronautics and Astronautics Department at MIT and his B.Sc. degree from the Aeronautics and Astronautics Department at MIT in 2007 and studied abroad at ETH Zurich for a semester in 2006. Before joining the Aerospace Controls Lab, he worked as an intern at Avidyne Corporation developing and testing electronic flight displays. He is currently an Aurora Flight Sciences Fellow and will be joining Aurora full time after graduation.

Luca F. Bertuccelli received his B.S. degree in aeronautical and astronautical engineering from Purdue University and his M.S. and Ph.D. degrees in aeronautics and astronautics from MIT. He is currently a postdoctoral associate at MIT in the Department of Aeronautics and Astronautics. His research interests include robust decision making and human-in-the-loop planning.

Olivier Toupet received his M.Sc. degree in aeronautical engineering at SUPAERO, France, in 2004, and his M.Sc. degree in aeronautics and astronautics at MIT in January 2006. Upon graduation, he worked at Nascent Technology Corporation as a lead aerospace engineer involved with flight dynamics and control work, simulation, software development, flight control hardware integration, and autonomous helicopter flight tests. He then joined the R&D center of Aurora Flight Sciences in November 2006 as a research engineer in the Autonomy, Controls and Estimation (ACE) division. He is currently an autonomy engineer at Aurora Flight Sciences, R&D Center, Cambridge, Massachusetts. He has since been the principal investigator and program manager on various STTR and SBIR programs, most of them related to multivehicle coordinated decision control and sensor fusion.

Luc Brunet received his B.Eng degree with high distinction in aerospace engineering from Carleton University in 2006 and his S.M. degree in aeronautics and astronautics from the MIT in 2008. He is currently a collaborative robotics specialist for Frontline Robotics in Ottawa, Ontario, Canada.

Abe Bachrach received his B.Sc. degree from the Electrical Engineering and Computer Science Department at the University of California, Berkeley. He is currently a Ph.D. degree candidate in the Department of Electrical Engineering and Computer Science at MIT.

Nicholas Roy received his Ph.D. degree in robotics from Carnegie Mellon University in 2003. He is the Boeing assistant professor in the Department of Aeronautics and Astronautics and a member of the Computer Science and Artificial Intelligence Laboratory (CSAIL) at MIT. His research interests include autonomous systems, decision-making under uncertainty, machine learning, mobile robotics, and human–computer interaction.

Address for Correspondence: Jonathan P. How, 77 Massachusetts Avenue, Room 33-326, Cambridge, MA 02139, USA. E-mail: jhow@mit.edu.