Design, Integration, and Flight Test Results for an Autonomous Surveillance Helicopter

M. Whalley
mwhalley@mail.arc.nasa.gov
Army/NASA Rotorcraft Division
Aeroflightdynamics Directorate (AMRDEC)
US Army Research, Development,
and Engineering Command
Ames Research Center, CA

M. Freed
michael.a.freed@nasa.gov
Computational Sciences Division
NASA Ames Research Center, CA

R. Harris
rharris@mail.arc.nasa.gov
QSS, Computational Sciences Division
NASA Ames Research Center, CA

M. Takahashi
mtakahashi@mail.arc.nasa.gov
QSS, Computational Sciences Division
NASA Ames Research Center, CA

G. Schulein
gschulein@mail.arc.nasa.gov
San Jose State Foundation
Army/NASA Rotorcraft Division
NASA Ames Research Center, CA

J. Howlett
jhowlett@mail.arc.nasa.gov
Army/NASA Rotorcraft Division
Aeroflightdynamics Directorate (AMRDEC)
US Army Research, Development,
and Engineering Command
Ames Research Center, CA

Theoretical development and practical implementation of an autonomous surveillance helicopter is described. The autonomous surveillance planning problem for multiple, varying targets-of-interest is defined. An example solution is presented that uses a 2-opt approach incorporating a math model of the vehicle and repeated visits to the targets. A methodology for evaluating surveillance algorithms is described and demonstrated by comparing human performance with the 2-opt approach. A helicopter research platform developed for the purpose of demonstrating autonomous behaviors, including surveillance, is described. The 2-opt algorithm, a reactive planner, Apex, and obstacle avoidance route planning are integrated into the research helicopter and test flown. Finally, flight test results are reported for the surveillance concepts and algorithms developed to-date.

NOTATION

ARP	Autonomous Rotorcraft Project
ASPP	Airborne Sureveillance Planning Problem
DART	Disaster Assistance and Rescue Team
DOMS	Distributed Open Messaging System
ECI	Expected cost of ignorance
E	Cost-imposing event
GSO	Ground Station perator
HIL	Hardware-in-the-loop
$p(t)$	Probability density function
OFRP	Obstacle Field Route Planner
PDL	Procedure Definition Language
TOI	Target-of-interest
UAV	Unmanned Aerial Vehicle
URL	Uniform Resource Locator

Presented at The AHS International Specialists' Meeting on Unmanned Rotorcraft, Mesa, AZ, Jan. 2005.

INTRODUCTION

One of the earliest applications of powered air vehicles was gathering information about conditions on the ground, exploiting the relatively high speed and broad view offered by these machines to, e.g., provide guidance to World War I artillery units and track enemy movements. Similar surveillance applications quickly emerged in other areas such as security, land management and scientific research. Unmanned aerial vehicles (UAVs) have dramatically increased the availability and usefulness of aircraft as information-gathering platforms (Ref. 1). As UAV technologies improve and the number of such vehicles increases, costs will come to reflect economy of scale and decreased weight and complexity made possible by not having to support human occupants. Reduced cost should, in turn, make UAVs available to a wider and less specialized set of users and for increasingly diverse purposes. This presents a challenge: how best to accommodate increasingly diverse missions and users of UAV-based surveillance platforms.

Surveillance is often a lengthy, repetitive and largely uneventful process that strains human vigilance and morale, making it an excellent application for autonomy. However, performing surveillance autonomously is challenging for several reasons. One is that, for surveillance of multiple, spatially-separated targets, deciding where to observe next and what observation actions to perform is inherently a difficult planning and scheduling problem. This problem, termed here as the "Airborne Surveillance Planning Problem" (ASPP) differs in significant ways from planning and scheduling problems typically addressed in operations research and artificial intelligence. This work has therefore involved not only devising new algorithms, but also creating a theoretical foundation and implementing a test bed for evaluating surveillance decision methods. Moreover, it is recognize that the development of robust ASPP algorithms is likely to be a gradual process, with existing methods likely to perform well in some conditions and poorly in others. Thus, part of this effort has been to create a way to formally characterize the conditions in which surveillance algorithm degrades to a point where human intervention becomes desirable.

This work on autonomous airborne surveillance is part of a larger effort, the NASA/Army Autonomous Rotorcraft Project (ARP, Ref. 2). The project addresses a range of challenges with the goal of producing a practical helicopter-based observation platform that is versatile enough to support both NASA science missions and military surveillance missions. In this paper, the foundational work in addressing the ASPP is reviewed, and then discuss ARP system development efforts and flight test results supporting realization of the project's overall goal.

THE AIRBORNE SURVEILLANCE PLANNING PROBLEM

Surveillance tasks involve repeated or continuous observation intended to maintain awareness of some entity or geographical area. Because surveillance generally takes place over a lengthy period, it is particularly appropriate to carry out these missions with autonomous vehicles rather than, as is now the norm, with humans as remote operators. In surveilling multiple, spatially-separated sites, an autononous aircraft needs to repeatedly decide where to go next and what observation activities to perform when it gets there. However, choosing to observe one site has a cascading effect on both the time-cost and desirability of all subsequent observation tasks. This coupling of current and future choices suggests considering each observation to be part of a plan or schedule whose utility can be compared to possible alternatives.

Unlike typical planning and scheduling problems, such as the traveling salesman problem, the surveillance problem requires allowing for the possibility that some sites should be visited more often than others due to differences in their importance and in the rates at which observed information becomes obsolete. And, given an overall goal of maximizing the value of information returned to the user, a surveillance planner should, in many cases, omit visits to some (possibly most) sites entirely in order to observe the most important ones at a higher rate. Surveillance planning thus combines task ordering with task selection, a combination notorious for increasing the computational complexity of any solution (Ref. 3).

Numerous factors, some specific to air vehicles, should be taken into account by a surveillance algorithm. For instance, evaluating the desirability of a candidate site will typically require estimating how long it will take to get to the site and take any needed measurements. However, even assuming that the site is at a known, fixed location, UAV-relevant variables such as wind speed, vehicle payload, and routing for obstacle avoidance can all have a significant impact on travel time. Other factors affect what kind of approach will be effective in comparing alternatives. For instance, an algorithm that works well when the number of surveillance sites is small (say, five) may not work well when the number of sites is an order of magnitude larger. Similarly, algorithms that attempt to take advantage of problem structure – e.g., spatial "clumping" of sites or non-uniform distribution of event probabilities across these sites – will not be effective in problem instances lacking these structural features. A useful solution will need to deal with the range of qualitatively distinct surveillance mission definitions that might be encountered in operational contexts.

A Decision-Theoretic Approach To Autonomous Surveillance

A key part of our effort is a framework for evaluating surveillance decision performance in a wide range of mission scenarios. Like Massios, Dorst and Voorback (Ref. 4), a decision-theoretic approach is taken in defining the surveillance problem. The value of making an observation at a particular time and place, then returning that information to a user, depends on the kinds of events that might be observed and the value the user places on knowing about them. As the value of information often depends on when the user receives it (e.g. it is better to be informed of a break-in as it is beginning than long after the thief has escaped), surveillance decisions should take into account temporal characteristics of the task environment such as the likelihood of an interesting event occurring over a given interval and the change over time in the value of observing that event after it occurs. This approach treats observations as boundaries on time-intervals in which the user has been ignorant of events occurring at a given site (target). The expected cost of ignorance (ECI) for a given target over a given interval is:

$$\mathrm{ECI}(t_1, t_2) = \int_{t=t_1}^{t_2} p(t) \cdot \mathrm{cost}(t_2 - t)dt$$

where (t_1,t_2) is the interval between observations measured from some mission start time t_0, $p(t)$ is the probability density function for the occurrence of some cost-imposing event E (e.g. a break-in) and cost(d) is a function describing the expected cost imposed by E as a function of the time from occurrence to intervention. The expected cost of ignorance is thus the sum, for all points in the interval, of the probability of the event occurring at that point times the expected cost if it occurs at that point. With cost and probability functions appropriate to model events of type E, the total cost of a given surveillance schedule is the sum of ECIs for all inter-observation intervals for all targets. The value of observations resulting from following the schedule is the worst-case schedule cost (no observations at all) minus the cost of the selected schedule. The goal of a surveillance algorithm is to maximize that value. See Freed et al. (Ref. 5) for more detail on this approach.

Using this formulation of the ASPP problem requires temporally structured models of the environment. For instance, an autonomous surveillance agent might be given a set of observation goals that include checking for fires in a particular region of forest. Quantifying the expected cost of remaining ignorant of the state of that site requires a model of how a fire there might spread and how costly the user considers a fire that has spread for a given interval. To characterize the mounting cost of a forest fire, the user might realize that fires tend to start slowly and then spread rapidly, but eventually slow down as they consume the fuel in a region and approach natural boundaries. A mathematician might choose to model this as an appropriately parameterized sigmoid function.

However, an autonomous vehicle in daily use for fire surveillance is likely to be attended by people who know a lot about forests and forest fires, but (perhaps) are not mathematicians. This highlights one of several issues that need to be addressed to make an autonomous surveillance capability contribute effectively to operations within a larger human-machine system (Ref. 6). In particular, human users must be able to construct environment models and periodically modify them as conditions in the environment evolve.

Another issue arises for users making reconnaissance requests that must be executed against the "background" surveillance task. A requestor might mean "drop everything and go perform this observation" but might also be making a more nuanced request such as "make this observation at the most convenient time in the current schedule". To handle these different kinds of requests, the system needs to know what degrees of freedom exist in scheduling the proposed action and to insure that the user understands and accepts the observation delays that will result from carrying it out.

Evaluating Surveillance Algorithms

The decision-theoretic approach described above defines a method for evaluating how well a surveillance algorithm performs in a specific scenario. Understanding the general strengths and weaknesses of an algorithm requires applying the approach to a range of representative mission types and comparing the results to different algorithms. And, since there may be missions in which humans can outperform all known algorithms, human performance may be tested in the same way using the same example missions. One result of such testing is that the autonomous control software on-board a surveillance UAV could match its current mission to the most similar of the evaluated missions, then decide which algorithm is likely to be most effective and whether humans are likely to perform better than any available algorithm. In addition, if the mission changes while in flight – e.g. if surveillance targets are added or deleted – the system would have a principled and empirically justified basis for switching to a different algorithm or calling for human assistance.

A characterization and evaluation methodology has been selected to represent a significant portion of the diversity of surveillance missions in realistic NASA and Army contexts. Five independent variables with three values each are used:

1. Number of observation targets: 4, 8 or 16
2. Spatial separation of targets: 0.2, 2, or 20 percent of vehicle range
3. Spatial distribution: uniform, globular, 2-cluster (Fig. 1)
4. ECI maximum cost distribution: fixed, uniform, clustered
5. ECI rate distribution: fixed, uniform, clustered

Defining the problem this way yields 243 separate cases (see Ref. 6 for more detail).

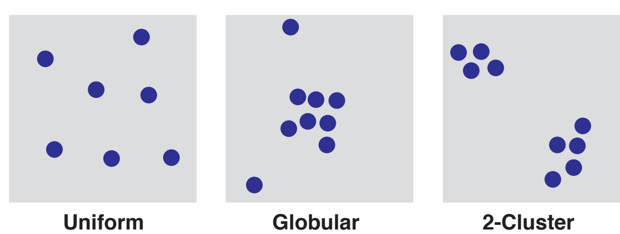


Figure 1. Spatial distribution examples.

2-opt Algorithm Comparative Analysis

As a candidate autonomous surveillance decision-method, a variant of the 2-opt algorithm for the Traveling Salesman Problem was used (Ref. 7). The 2-opt algorithm starts with a random tour of the targets and then iteratively finds and applies a tour-improving exchange of segments until no further improvements can be found (Fig. 2). The basic algorithm was modified to generate repeating sequences and to use the flight-identified UAV math model to compute travel time between targets. Schedules for each scenario generated by this method were evaluated using the ECI approach outlined above.

2-opt planning performance was compared with human subject performance as an initial study into the use of the ASPP characterization. Human surveillance performance was based on data from five subjects. Each scenario was depicted as a map (Fig. 3) with observation targets color-coded to indicate cost-rate (urgency) and shape-coded to indicate maximum cost (importance). The start/end point (home) was depicted as a distinctive icon and spatial scale was represented by a scale in the lower right-hand corner. The subject used a mouse to select and modify a route.

The amount of time taken to select each route was displayed, though no time limit was enforced. The subject was very familiar with the surveillance task but was not given training on effective strategy. Comparison between human and algorithm performance (Table 1) showed a number of trends including the following:

1. The subjects generally out-performed the 2-opt algorithm when the scale was large, especially when there were a large number of targets.
2. The subjects performed especially well when there was a lot of structure to reason about (sets of targets "clumped" together).

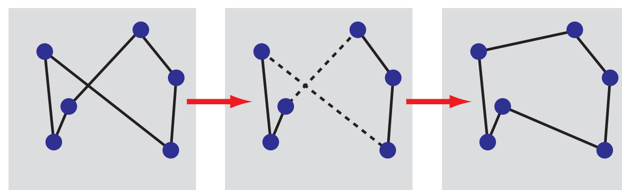


Figure 2. 2-opt solution method.

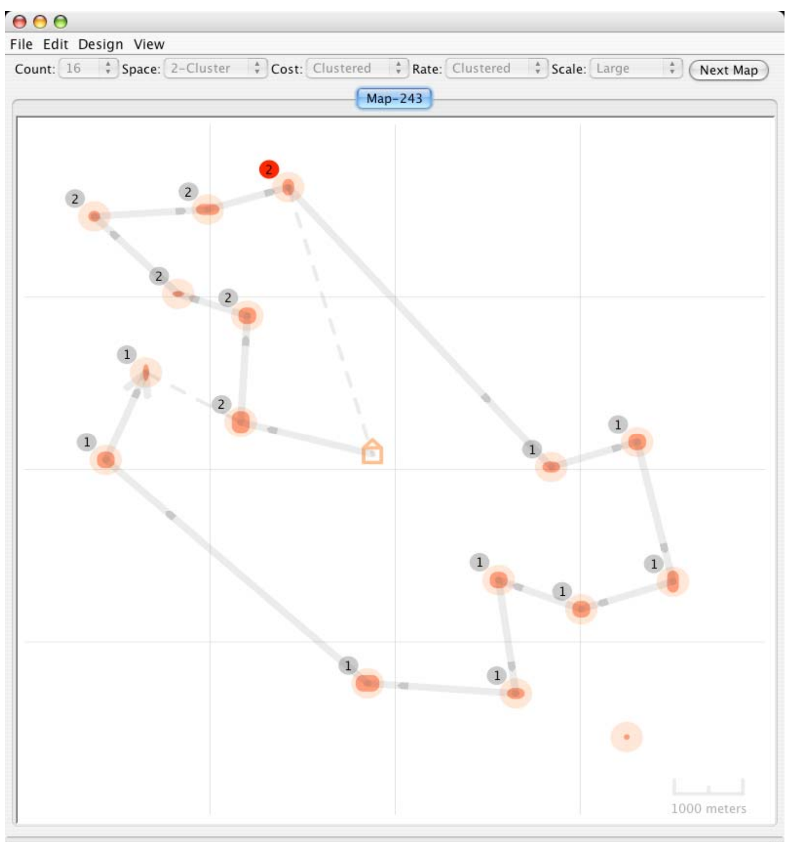


Figure 3. Target display used in human performance testing; symbol height denotes importance, symbol width denotes obsolescence rate; numbers indicate number of visits during mission.

Table 1. Percentage difference in performance between 2-opt and human-directed surveillance. Positive values indicate advantage for 2-opt.

			4			4 Overall	8			8 Overall	16			16 Overall
Scale	Rate	Cost	2-Cluster	Globular	Uniform		2-Cluster	Globular	Uniform		2-Cluster	Globular	Uniform	
Large	Clustered	Clustered	-3			-1	2	-1	-8	-3	-5	-11	-7	-8
		Fixed	-3			-1	-2	-2	-4	-3	2	-11	1	-3
		Uniform	-4		-1	-1	-3	1	-6	-3	-7	-22	-8	-12
	Clustered Overall		-3			-1	-1	-1	-6	-3	-3	-14	-5	-8
	Fixed	Clustered	-3			-1	-2		-6	-3	-8	-1	-7	-5
		Fixed	-3			-1	-2	-1	-3	-2	1	-10	1	-3
		Uniform	-5		-1	-2	-3	1	-5	-2	-7	-18	-8	-11
	Fixed Overall		-4			-1	-2		-4	-2	-5	-10	-5	-6
	Uniform	Clustered	-4	-1	1	-1	-3	-2	-9	-5	-5	-12	-1	-6
		Fixed	-4	-1	1	-1	-2	-3	-2	-2		-12		-4
		Uniform	-5	-3	-2	-3	-3	2	-7	-3	-9	-15	-10	-11
	Uniform Overall		-4	-2		-2	-3	-1	-6	-3	-5	-13	-4	-7
Large Overall			-4	-1		-2	-2	-1	-6	-3	-4	-12	-4	-7
Medium	Clustered	Clustered						1		1	3	2	1	2
		Fixed					2			1	4	2		2
		Uniform									1	5		2
	Clustered Overall						1	1		1	3	3	1	2
	Fixed	Clustered						1			1	2	1	1
		Fixed									1	1		1
		Uniform			2				2	1	1	3	2	2
	Fixed Overall				1				1	1	1	2	1	1
	Uniform	Clustered					1			1	2	3	3	3
		Fixed			1				2	1	3	2	1	2
		Uniform	-1	3	1	1			1	1	2	1	1	1
	Uniform Overall			1	1		1		1	1	2	2	2	2
Medium Overall							1		1	1	2	2	1	2
Small	Clustered	Clustered												
		Fixed									2			1
		Uniform		4		1					1			1
	Clustered Overall			1							1			1
	Fixed	Clustered									1			
		Fixed												
		Uniform									1			
	Fixed Overall													
	Uniform	Clustered									1		4	2
		Fixed							2	1	1			1
		Uniform							3	1	1	3		1
	Uniform Overall								2	1	1	1	1	1
Small Overall									1		1		1	1

What's missing from this analysis is any measure of time to complete the plan or subject workload in doing so. Anecdotal evidence suggests that the subjects who participated in study found the work truly dull and laborious. Even so, it seems clear that it will be advantageous to build into any algorithmic surveillance the capability of having a human operator resolve conflicts or "tweak" the plan as necessary.

Future studies will include assessment of additional algorithms with the intent of having different algorithms that can be brought to bear in different situations. Also, including a larger number of human subjects provided with training and decision aids will enable a more reliable and detailed assessment of relative strengths and weaknesses.

IMPLEMENTATION – THE AUTONOMOUS ROTORCRAFT PROJECT

Field-testing is an indispensable component of our research. Real-world uncertainty quickly reveals the shortcomings of autonomy concepts as well as overall system weaknesses. To this end, ARP has been developing an autonomous helicopter system, described in Ref. 2, for conducting this and other UAV research.

There are a host of subjects to be considered under the topic of developing an autonomous helicopter; e.g., imaging sensor design, payload stabilization, communication bandwidth and robustness, flight control and disturbance rejection, vehicle performance and reliability, obstacle detection, operator interface design, etc., all of which must be up to the task before considering the implementation of a practical autonomous capability. Through a series of developmental tests, ARP has developed a capable, flexible, and reliable UAV system that permits easy implementation and evaluation of autonomy concepts. This section provides a description of the major elements of the ARP UAV system, the implementation of the surveillance algorithms and autonomous behaviors described above, and the lessons learned from field-testing.

Hardware Description

A Yamaha RMAX helicopter is used as the demonstration platform (Fig. 4). The RMAX was originally developed for remote control agricultural seeding and spraying but has been adapted for use as an autonomy demonstration platform. The aircraft is capable of approximately one hour of hover flight duration with a 65 lb payload. The maximum speed of the vehicle is approximately 40 kt.

Figure 4. ARP RMAX research aircraft.

Avionics payload

The RMAX has been modified to include an avionics payload which carries a navigation and flight control computer, experimentation computer (typically used for vision processing), inertial measurement unit, GPS receiver, and radio communications equipment. The payload was designed for simple maintenance and to be easily transferred between aircraft. A mobile ground station provides support and acts as a base of operations for the testing. Figures 5-8 show some of the main hardware elements of the ARP UAV system.

Figure 5. Avionics payload.

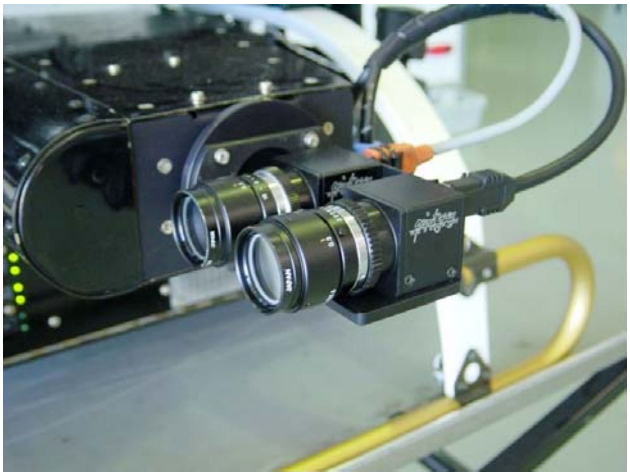


Figure 6. Camera system detail; monochrome stereo (for passive ranging) and color IEEE-1394 cameras on left wingtip.

Figure 7. Modified SICK scanning laser (for active ranging).

Figure 8. Mobile ground station.

IEEE-1394 camera system

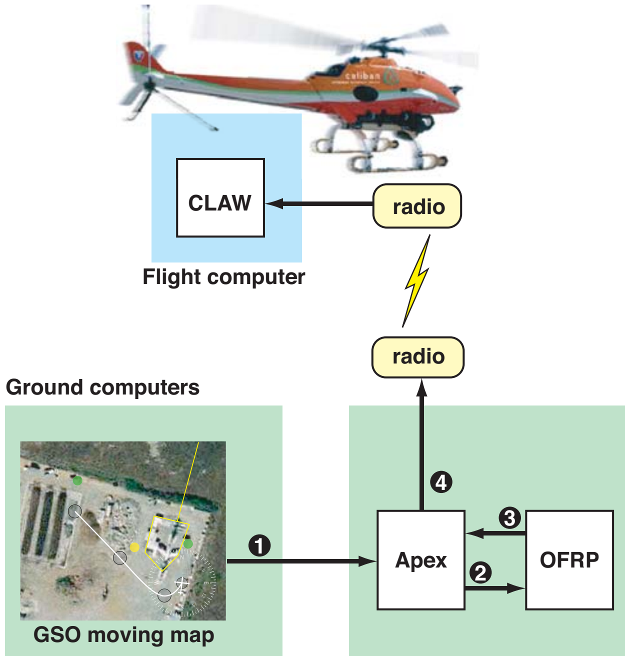
Separate from the avionics payload is a vibration-isolated stub wing on which various cameras can be mounted. A tilting mechanism supports a pair of monochrome 640x480 resolution progressive-scan IEEE-1394 cameras (Fig. 6). These stereo cameras have a one-meter baseline to provide accurate passive-ranging of obstacles at distances sufficient for path re-planning. The tilting mechanism provides +10 to –100 degrees of pitch travel. Cross-shafting provides sufficient stiffness to ensure the stereo cameras maintain proper alignment throughout their range of travel and under vehicle vibratory loads. Any process can interrogate or reposition the stereo camera tilting system using a simple DOMS message (see below). A color 640x480 resolution IEEE-1394 camera mounted alongside the left monochrome camera provides real-time progressive-scan streaming imagery to the ground at 10 fps. Communication with and power to the cameras is achieved via the IEEE-1394 connection. Compact 8-mm fixed-focal-length C-Mount lenses provide approximately 45x36 degrees field-of-view. Images are gathered simultaneously from the cameras at a rate of 30 fps and stored on-board. A video server software system has been developed to ensure that any process can access the camera imagery when needed; e.g. the stereo passive ranging system, a monocular tracker, and a video compression and downlink system can make simultaneous use of the imagery.

SICK scanning laser

Mounted under the nose of the aircraft is a SICK PLS scanning laser used for obstacle detection and high-resolution mapping. The sensor has been remanufactured and lightened from 9.9 lb to 3.6 lb for use on the helicopter. The mounting system provides vibration isolation. The device provides centimeter-accuracy range measurement every one degree over a field-of-view of 180 degrees. Scans are performed at 75 Hz. The maximum range of the sensor is 80 meters. The sensor elevation is easily repositioned depending on the research requirements; e.g., downward for high-resolution mapping or forward for (2-D) obstacle detection.

System Architecture and Software Elements

Figure 9 shows the major system elements and interconnections relevant to the surveillance algorithm work. In this case the planning software is run on one of the ground computers. However, the messaging system makes the physical location of any particular process arbitrary. Detailed descriptions of the individual elements are provided below.



1 doms:///rcTarget/map/target
2 doms:///rcWaypoint/apex/raw-waypoint
3 doms:///rcWaypoint/oapp/safe-waypoint
4 doms:///rcWaypoint/apex/waypoint

Figure 9. System architecture and DOMS messages relevant to surveillance task.

Distributed Open Messaging System (DOMS)

ARP requires a communication standard that can cope with the intensive data flow between the wide variety of processes. Flight control, path generation, video processing, health monitoring, and mission planning all have different needs with respect to data communication bandwidth, synchronization, and quality. To meet this need, on-board and telemetry information exchange is performed using the ARP-developed Distributed Open Messaging System (DOMS).

DOMS uses a publish-subscribe style message passing communications architecture. Publish-subscribe message passing is the preferred way to handle data flow from multiple asynchronous sources as is commonly found in robotics applications. The publish-subscribe technique allows data to be exchanged with little or no information about other processes in the system. DOMS supports multi-cast messages which pass the same data to multiple computers with a single transmission. DOMS also supports communications on a single computer or even within a program. A transport daemon handles routing between multiple computers. A uniform resource locator (URL) is used to describe message subscriptions, publications, and transport routes. Typical DOMS URLs are shown in Fig. 9.

Apex – A Coupled Reactive Planner

High-level autonomous control is provided by Apex, a reactive, procedure-based planner/scheduler used for mission-level task execution, navigation, response to vehicle health and safety contingencies and interaction with human users. The 2-opt surveillance scheduler acts as an expert called upon by Apex to perform the scheduling task. Surveillance scheduling in a realistically dynamic mission context – i.e., where flight conditions and user needs can change often and unexpectedly – is seen as a special case of the problem of multitask management, a central Apex capability and research focus (Refs. 8 and 9). Though the approach it incorporates has proven effective for some relatively complex tasks (Refs. 10 and 11) the surveillance problem has proven much more demanding.

The core element of Apex is a reactive planning algorithm that selects actions based partly on a library of stored partial plans. Such planning algorithms are considered reactive, because decisions about the next course of action evolve as new decision-relevant information becomes available. For example, reconnaissance of a particular location might be delayed in response to hazardous weather conditions or, alternately, increased in urgency if weather conditions are likely to make the route hazardous later. Similarly, a decision regarding how to get to the location might be made (or changed) at any time in the course of carrying out the overall plan based on changes in the probable locations of hazards and information opportunities.

Apex synthesizes a course of action mainly by linking together elemental procedures expressed in Procedure Definition Language (PDL), a notation developed specifically for the Apex reactive planner. A PDL procedure consists of at least an index clause and one or more step clauses. The index uniquely identifies the procedure and describes a class of goals for which the procedure is intended. Each step clause describes a subtask or auxiliary activity prescribed by the procedure. Steps are not necessarily carried out in the order listed or even in a sequence. Instead, they are assumed to be concurrently executable unless otherwise specified.

ASPP behaviors added to Apex

Simulation testing led to a few behavior enhancements to the system to increase flexibility. It was clear early on that performing the surveillance task goes well beyond simply flying to the targets in the correct order. The additional behaviors that needed to be integrated into Apex to handle these tasks are described below.

Selecting the best vantage point for viewing a target involves several factors. The angle of the sun can produce undesirable effects such as lens flare, poor dynamic separation, etc. in an electro-optical sensor. Wind angle produces undesirable aircraft motion when blowing from the right due to tail rotor wake re-ingestion (the RMAX main rotor turns clockwise). Proximity to obstacles must also be considered.

The sun angle was easily incorporated into the selection of a vantage point through the use of a time-based sun azimuth and elevation model. The vantage point was positioned such that the aircraft viewed the target from the direction of the sun. Also, the vantage point was selected so as to maintain a safe distance from known obstacles. It was also selected such that there would be no obstacles in its line-of-sight to the target.

The ground station operator (GSO) was given the ability to change the heading from which the vantage point viewed the target. This added flexibility improved the situational awareness of the GSO once the aircraft had reached the computed vantage point.

Although algorithmically correct, having the surveillance routing immediately redirect the vehicle – in effect, change its mind – every time a new target was introduced was judged to be disruptive. Given the relatively short travel time between targets, it was decided that the aircraft should finish its journey and observation of the current target before introducing a route update. However, new obstacle edges always resulted in a route recalculation to ensure there was no conflict with the current path.

Obstacle Field Route Planner (OFRP)

A key feature of the ARP platform is the ability to perform route planning for obstacle avoidance. ARP employs the Obstacle Field Route Planner (OFRP, Ref 12). The OFRP algorithm provides a 2-D solution using a four phase approach: Voronoi graph generation from obstacle edges (Fig. 10), graph culling using binary space partitioning, shortest path search using Eppstein's search method (Ref. 13), and path smoothing using binary space partitioning again. For this test, obstacle locations were known beforehand and predefined on the 2-D map of the test site.

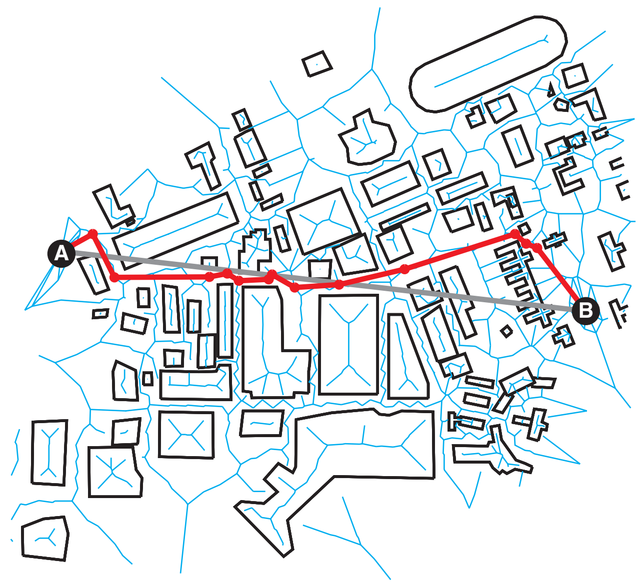


Figure 10. Obstacle Field Route Planner output; obstacles (black), Voronoi graph (blue), and resulting 2-D path (red).

Control Law Software

The Control Law (CLAW) provides attitude stabilization and waypoint guidance control. The top level topology of the control law is shown in Fig. 11.

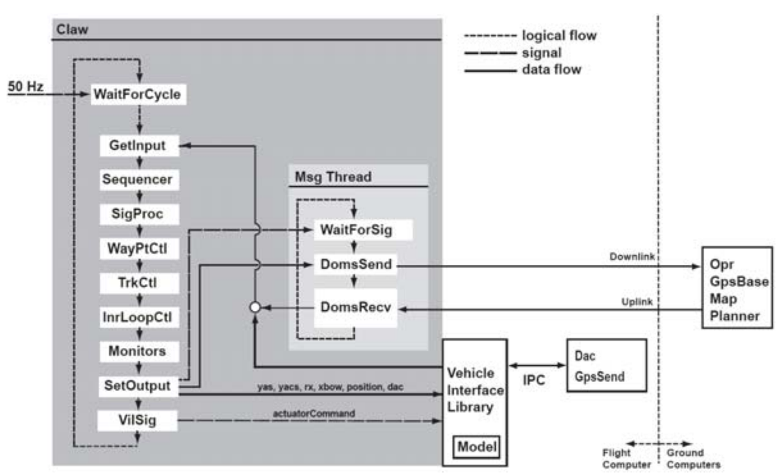


Figure 11. Flight control law topology.

The waypoint controller uses the vehicle state estimates and the waypoint data from the Apex reactive planner and commands the tracking control. Waypoints arrive in the controller and are converted into a stream of commanded positions at the control cycle time. This process is handled by the path smoother and the path follower.

The Path Smoother accepts a three-dimensional list of waypoints and returns a larger list of waypoints that describe a desired smooth path between the original waypoints (Fig. 12). Each original waypoint has a radius which is used to construct a corridor that the smooth path is allowed to exist in. The returned path is smoothed in three dimensions. After a smooth path is generated, a velocity profile for that path is calculated, taking into account user-supplied values for maximum bank angle and cruise speed.

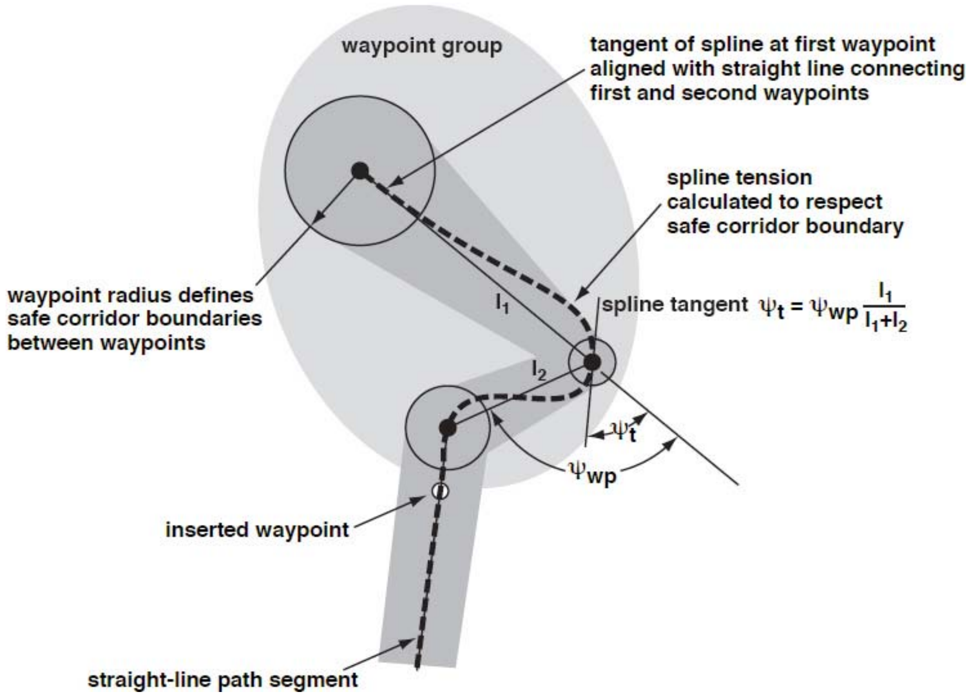


Fig 12. Kochanek-Bartels spline path smoother employed by waypoint-following flight control laws.

One key feature of the flight control system is the ability to command heading independent of the path following. This was exploited by Apex to point the vehicle and its cameras at the target-of-interest (TOI) whenever the vehicle was within 25 m of the TOI and below 3 m/sec total airspeed. This gave the GSO a view of the TOI as the aircraft was arriving at and departing from the vantage point.

Surveillance Task Sequence

The resource described above were used to construct a surveillance mission. This mission consisted of the following sequence of events:

1. The Ground Station Operator (GSO) initiated the scenario by clicking locations on the moving map display to indicate TOI locations.
2. Apex would compute a vantage point for each TOI and then call the 2-opt algorithm to sequence the targets for surveillance. The route was expressed as a series of waypoints.
3. The series of waypoints computed by Apex were processed by the obstacle field route planner (OFRP) which would insert any necessary additional waypoints to route the aircraft around known obstacles.
4. The final, expanded list of waypoints was transmitted to the aircraft, which would fly the commanded route, stopping for five seconds at each vantage point to return real-time imagery of the TOI.

The aircraft continued to compute and fly a route, repeating observations of the TOIs. This would continue unless the TOIs were deleted by the GSO.

The flexibility and robustness of the route planner was demonstrated by having the GSO periodically insert or delete TOIs on-the-fly, which triggered an automatic update of the route. Similarly, virtual obstacles were

Figure 13. Still image from ground station status video display (clockwise from upper left): 3D graphical display showing aircraft and desired flight path; video downlink; moving map showing aircraft, targets, and obstacle locations; ground tracking camera.

periodically added (simulating new obstacle detection information) also resulting in an update of the route as needed.

Computation of the waypoints and transmission to the aircraft (steps 2-4) happened automatically once the GSO selected TOIs on the moving map. Figure 13 shows a screen capture of the moving map illustrating the TOIs, waypoints, virtual obstacles, and flight path.

Hardware-in-the-loop testing

Before performing the surveillance mission in-flight, extensive hardware-in-the-loop (HIL) verification, validation, and integration testing of the project software was conducted. A highly accurate math model representation of the RMAX is embedded in CLAW for precisely this reason. The model contains the key components that significantly affect the quality of the feedback including a dynamic linear math model identified from flight, non-linear kinematics, position and rate limited actuators, transport delay, sensor noise, and sensor quantization effects. The actuator commands are sent to the internal model as well as the vehicle actuators.

FLIGHT TEST RESULTS

Test flights were performed at the NASA Ames Disaster Assistance and Rescue Team (DART) Collapsed Structure Rescue Training Site (Figs. 14 and 15). The DART site contains a large (simulated) collapsed building, rubble piles, and a pair of towers that must be avoided in flight. TOIs were selected on either side of the structure thus forcing the aircraft to plan routes around it to gain a vantage point. All flight was conducted at a constant altitude of approximately 10 m above ground level and at a maximum speed of 5 m/sec.

Figure 14. DART Collapsed Building Training Center.

Figure 15. Flight testing at the DART site.

In general, flight-testing proceeded as anticipated from HIL testing. The vantage point selection, sequencing of waypoints, and route planning proceeded without incident. The flight control laws held the path error to within 10 cm even when having to cope with significant turbulence downwind of the towers. Computation of the target sequence, obstacle-free route planning, and path smoothing was typically accomplished in tens of milliseconds. Imagery returned by the digital video system was clear and smooth without placing an excessive burden on the telemetry system.

CONCLUDING REMARKS

Significant progress has been made in the development of an autonomous surveillance helicopter as summarized below:

1. A helicopter research platform has been developed for the purpose of demonstrating autonomous surveillance as well as other autonomous helicopter behaviors. Flight testing of the surveillance concepts and algorithms developed to-date has demonstrated the real-world practicality of these methods.
2. The surveillance problem has been defined using a decision-theoretic approach. This method attempts to maximize the difference between the expected cost of ignorance of not visiting any targets at all, and the cost of visiting the targets in a chosen sequence.
3. A 2-opt approach to solving the problem has been developed that incorporates a dynamic model of the vehicle and repeated visits to targets-of-interest. Additional behaviors have been linked to this core planning capability through the integration and extension of the Apex reactive planner
4. A methodology for evaluating surveillance algorithms has been developed and demonstrated by comparing human performance with the 2-opt approach. This technique is directed at developing an objective rationale for choosing the most appropriate surveillance algorithm for a given situation.

FUTURE WORK

The project focus so far has been on the theoretical underpinnings and practical implementation of surveillance decision-making. Work will continue on the development of additional surveillance algorithms to be evaluated against the framework described. This will enable the selection of the algorithm appropriate to the task at hand thereby increasing the robustness of the overall system. Also of interest is the human interaction problem with autonomous systems and the elicitation of utility knowledge from human experts in defining the ECI function for a given situation. Development of the research platform will continue with particular emphasis on the subjects of obstacle detection and representation through active and passive means, route planning and reactive maneuvering for obstacle avoidance, take-off and landing at non-cooperative sites, and aggressive maneuvering.

ACKNOWLEDGEMENTS

This work was supported by the NASA CICT/Intelligent Systems program. The NASA CICT program was established in 2001 to ensure NASA's continuing leadership in developing and deploying key enabling technologies for a broad range of mission-critical tasks. Support was also provided by the Army Aeroflightdynamics Directorate (AFDD) and the Flight Projects Office of that same organization.

REFERENCES

1. Office of the Secretary of Defense, “Unmanned Aerial Vehicles Roadmap 2002-2027,” December 2002.
2. Whalley, et al, “The NASA/Army Autonomous Rotorcraft Project,” Proceedings of the American Helicopter Society 59th Annual Forum, Phoenix, AZ, May 2003.
3. Demeulemeester, E. and Herroelen, W., Project Scheduling, A Research Handbook, Kluwer Academic Publishers, Norwell, MA, 2002.
4. Massios, N., Dorst, L. and Voorbraak, F., “A Strategy for Robot Surveillance Using the Hierarchical Structure of the Environment,” Proceedings of the IJCAI Workshop on Reasoning with Uncertainty in Robotics, 43-50, Seattle, WA, August 2001.
5. Freed, M., Harris, R. and Shafto, M., “Human-Interaction Challenges in UAV-Based Autonomous Surveillance,” Proceedings of the American Association for Artificial Intelligence Spring Symposium, Stanford, CA, March 2004.
6. Freed, M. and Shafto, M., “Human versus Autonomous Control of UAV Surveillance” in proceedings of the AIAA 1st Intelligent Systems Technical Conference, Chicago, IL, September 2004.
7. Reinelt, G., “The Travelling Salesman. Computational Solutions for TSP Applications,” Vol. 840 of Lecture Notes in Computer Science, Springer-Verlag, Berlin Heidelberg New York, 1994.
8. Freed, M., “Managing Multiple Tasks in Complex, Dynamic Environments,” Proceedings of the Fifteenth National Conference on Artificial Intelligence (AAAI-98), Madison, WI, July 1998.
9. Freed, M., “Reactive Prioritization,” Proceedings of the 2nd NASA International Workshop on Planning and Scheduling for Space. San Francisco, CA, March 2000.
10. Freed, M., Matessa, M., Remington, R. and Vera, A., “How Apex Automates CPM-GOMS,” Proceedings of the Fifth International Conference on Cognitive Modeling. Bamberg, Germany, April 2003.
11. John, B., Vera, A., Matessa, M., Freed, M. and Remington, R., “Automating CPM-GOMS,” Proceedings of CHI 2002 Conference on Human Factors in Computing Systems, Minneapolis, MN, April 2002.
12. Howlett, J., Schulein, G., and Mansur, H., “A Practical Approach to Obstacle Field Route Planning for Unmanned Rotorcraft,” in proceedings of the American Helicopter Society 60th Annual Forum, Baltimore, Maryland, June 2004.
13. Eppstein, D., “Finding the *k* Shortest Paths,” SIAM Journal on Computing, Vol. 28, (2), pp. 652–673, 1998.