AN ARCHITECTURE FOR UAV TEAM CONTROL

Sivakumar Rathinam [*] **Marco Zennaro** [**]
Tony Mak [***] **Raja Sengupta** [****]

[*] *Graduate Student, CEE Systems, Berkeley*
[**] *Graduate Student, CEE Systems, Berkeley*
[***] *Systems Analyst, PATH, Berkeley*
[****] *Assistant Professor, CEE Systems, Berkeley*

Abstract: Recent years has seen a widespread interest in the use of Unmanned aircraft vehicles for military applications. These UAV's can be used in many applications such as surveillance, information gathering, suppression of enemy defenses, air to air combat, mapping buildings and facilities etc. In this paper, we present an architecture with the necessary algorithms that we have implemented to control a team of UAVs to search for targets such as SAMs, ground troops, artillery, tanks etc in a given region.

Keywords: UAV, search, architecture

1. INTRODUCTION

Unmanned Aerial Vehicles (UAV's) has received significant attention in recent years for military applications {Pachter, 98} {Bortoff, 99} {Mclain, 99} {McLain, 00} {Chandler, 01} {Nygard, 01} {Passino, 01} {Lee, 03}. The motivation behind this interest is to realize a vision where these vehicles cooperatively accomplish missions such as search and attack {Murphey, 99} {Polycarpou, 01} {Beard, 02} {Jaques, 03} {Bellingham, 03}. The reason for the widespread interest are the many advantages that the unmanned vehicles have over manned vehicles such as reduced human risk, manoeuvrability, and superior cooperation. In this paper, we primarily concentrate on the search mission and describe a architecture with the algorithms that we have implemented to realize the same. A detailed review of the search literature is presented in {Polycarpou, 01}. Our contributions of this work compared to the previous work are:

- Modularity with which the different types of sensors and the strategies are coupled in the architecture.
- A safe flight path planning algorithm that allows sufficient time for the sensors for basic operations such as imaging and image processing. Also this algorithm satisfies the the yaw rate and sensor range constraints of the vehicle.

1.1 *Basic Approach*

The primary step before developing search strategies and its architecture is to first come up with a way of representing the threats and to answer what kind of information is shared between the vehicles. To address these questions, we use a probability map to represent the threats. As the vehicles move in the unknown regions, the sensors collect new information (location and the type of targets) about the environment. This new information is updated with the probablity map using bayes rule. The probability maps of all the different types of the threats is used to calculate the risk map which indicates the risk of being shot at any given point in the desired search region. This risk map is shared by all the vehicles and is used for path planning. Following the path based on the partially known risk map, would damage or destroy the vehicles if they enter regions where Surface to Air Missiles (SAMs) are present. Therefore, we reduce the forward speed of the vehicles by refining the nominal path, thus allowing sufficient time for image processing and target recognition. This refinement of the nominal path is basically done by a safe flight design which is discussed in the later sections.

Two strategies for navigating in an unknown region are outlined in this paper: one for reaching a destination and an other for searching a given area. All the ideas presented above forms the backbone for the UAV team architecture. Ideally, what we want is an architecture that allows human interface at each level of control and also can automatically make its decisions if required as to which speed mode or sensor choice to use to suitably realize a mission. The architecture should allow an human interface to pick and choose the regions that is required to be searched.

The paper is organized as follows: The problem setup which includes the capabilities of the various sensors, threat models, assumptions on the motion of the vehicle is presented in 2. The threat map representation and the risk function calculation is presented in section 3. Safe flight algorithm is discussed in 4. Section 5 and 4 presents the two main search strategies. The main architecture is explained in section 6 and the properties are discussed in 6.5. Simulation results are presented in section 7 and the paper concludes with comments and future work.

2. PROBLEM SETUP

2.1 Sensors and their capabilities

Sensors such as Electro-optical(EO), Infrared(IR) and Synthetic Aperture Radar Systems can be used onboard of these vehicles to collect information about the partially known or unknown environments. The information that we are interested from these Surveillance and Reconnaissance (S&R) systems are the type of threats and the location of the threats. Each of these sensors has its own advantages and disadvantages and cannot be used for all applications. {Leachtenauer, 01} proposes a guideline in the selection of a sensor for a S&R application. This is presented in figure 1. Retrieving this useful information requires a sufficient amount of processing time (varies for different sensors) for performing basic operations

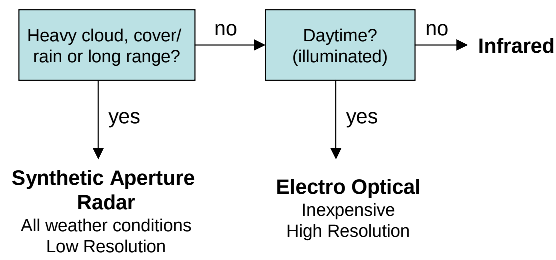

Fig. 1. General guidelines for sensor selection {Leachtenauer, 01}.

such as image formation and image processing. In this paper we assume that if sufficient amount of time is spent on imaging and processing an area, then the probability of detecting a target located in that area is 1.

2.2 Threat Model

In this work, we assume that the only threat that can destroy the vehicles are the stationary SAM launchers. We also assume that the range of the search radars of the launchers are larger than its fire control range. Therefore, as soon as the vehicles enter their search radar zone, they are locked for destruction and the weapons with its tracking radars are instantly fired when the vehicles enter the fire control range. Hence, a vehicle is instantly destroyed when they fall in the range of the fire control radars of the SAMs. The only way to keep the vehicles safe is to avoid the threat regions of all the launchers.

2.3 Assumptions on the Motion of the UAV

The vehicles are assumed to fly at a constant height. A UAV at any time t can be specified by its coordinates {x(t),y(t),θ(t)}. We treat each UAV as a Dubins car (simple stick model), travelling at a constant speed with a bound on its yaw rate. Let v denote the velocity of the UAV. Let ω represent the bound on the yaw rate of the UAV. The motion of the UAV is governed by the following equations:

$$\dot{x(t)} = v \cos \theta(t) \tag{1}$$

$$\dot{y(t)} = v \sin \theta(t) \tag{2}$$

$$\dot{\theta(t)} = \Omega \quad where \quad \Omega \; \epsilon \; [-\omega, +\omega] \tag{3}$$

3. THREAT REPRESENTATION

The targets consists of different types of targets such as Surface to Air Missile Launchers (SAMs), Surface to surface missiles, tanks, etc. A set of N targets is represented as:

$$Targets = \{target_1 = (type_1, (x_1, y_1)), .. \\ , ..., target_N = (type_N, (x_N, y_N))\} \quad (4)$$

Here, the $target_i$ is of $type_i$ and is situated at a location (x_i, y_i). Let the targets be distributed over k areas $A_1, A_2, ..., A_k$. N_{tj} targets of type t are assumed to be independently and uniformly distributed over Area A_j. The information about the targets is represented as a probability distribution of the random variables $N, type_i, \{x_i, y_i\}$. This probability distribution of targets is expressed as:

$$P_{threat}(Targets) = \prod_t \prod_{j=1}^{k} \prod_{i=1}^{N_{tj}} p_{tj}(type = t, (x_i, y_i)) P_{tj}(N_{tj}) \quad (5)$$

in which,

$$p_{tj}(type = t, (x_i, y_i)) = \begin{cases} |A_j|^{-1}, & (x_i, y_i) \in A_j \\ 0. & otherwise \end{cases} \quad (6)$$

At the start of the mission, this information about the targets or the probability distribution is also referred as the Initial Preparation of the Battlefield (IPB). As the vehicles move through the region, sensors on the vehicles gather information about the targets. We use Bayes rule the update the distribution with the incoming sensor information.

3.1 Risk Function Calculation

The instantaneous risk function $r(x, y, P)$ at any point (x, y) given the probability distribution P of the targets is given by:

$$r(x, y, P) = \sum_{j=1}^{k} \sum_t \sum_{N_{tj}=0}^{\infty} \sum_{n=1} N_{tj} \left\{ \int_{A_j} f_t(|(x, y) - (x_n, y_n)|) |A_j|^{-1} dx_n dy_n \right\} \quad (7)$$

where, the function f_t is chosen as follows: $f_t(r) = 1$ if $r \leq R_L$ and $f_t(r) = 0$ if $r > R_L$.

4. SAFE FLIGHT DESIGN

A detailed description of a general safe flight algorithm is presented in {Rathinam, 04}. Here, we just present the gist of designing a path for a vehicle required to travel from waypoint A to waypoint B'. As the vehicle moves from A to B', let the sensor on the vehicle collect information about the area $S_1S_2S_3S_4$ as shown in the figure 2. If the vehicle travels straight from A to B', since the forward speed of the vehicle is generally large ($200\frac{km}{hr}$), the sensor on the vehicle may not have enough time to process the information. Therefore, the vehicle may not be able to take any evasive action in the presence of any SAM launcher in the area $S_1S_2S_3S_4$. So, the vehicle has the risk of being shot. To avoid this, we increase the length of the path from A to B' to give sufficient time for the sensor to process the information. This is done by piecing together a series of semi circles and half ellipses as shown in figure 3. The minor axis x and the major axis y of the ellipses are the variables and are chosen to satisfy all the required constraints. Let us denote the generated path from A to B with n such semicircles and half ellipses be denoted by $G(A, B, n)$. The values of x and y are chosen to satisfy the following constraints:

- *Minimum curvature constraint*: Choosing $y = \sqrt{xr}$ where $r = \frac{v}{\omega}$ is the radius of the semicircle satisfies this constraint. Refer to the appendix for the proof.
- *Sensor timing constraint*: As shown in the figure 4, the sensor on the UAV, as it moves from A to B', is required to collect and process the information of the targets present in the area $S_1S_2S_3S_4$. The constraint here is that, the flight time required to travel from A to B' should be sufficient enough for the sensor to process the collected information. Let R_L be the firing range (km)of the SAM launchers present in the area of interest. Let τ indicate the image processing rate per unit area ($\frac{hr}{km^2}$) of the sensor on the UAV. Then,

$$2(R_L + d)\tau \leq \frac{(\pi r + L(x, y))}{2v(x - r)} \quad (8)$$

  where, $L(x, y)$ indicates the length of the half ellipse.
- *Sensor range constraint*: The range of the sensor R_S should be large enough (>length PS_2 in figure 4)for the sensor on the UAV to reach the entire area $S_1S_2S_3S_4$. That is,

$$(R_L + y + 2x)^2 + (R_L + 2y)^2 \leq R_S^2 \quad (9)$$

5. SEARCH STRATEGIES

5.1 Strategic Search

The objective in strategic search is to find a feasible path (if there exists one), devoid of threats to a given destination. At the start of the mission, only partial or no information is known about the location of the launchers. {Stentz, 96} proposed a efficient path planning algorithm for vehicles for

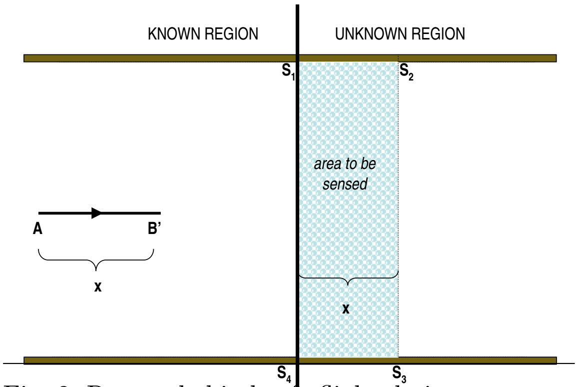

Fig. 2. Reason behind safe flight design.

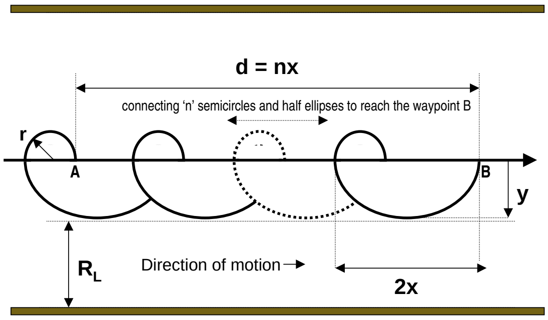

Fig. 3. Connecting semicircles and half ellipses to augment the length of the path.

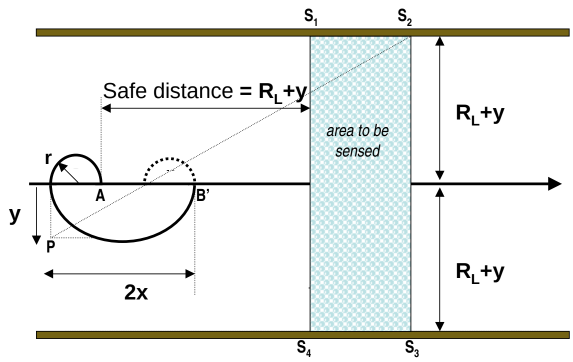

Fig. 4. Scanning area for the vehicle as it moves forward.

reaching a destination in partially known environments. The algorithm is functionally equal to the brute-force optimal strategy that is presented as follows:

- Follow the minimum risk path using the information known at the start.
- If any obstacle is identified, update the map with the collected information and follow the new minimum risk path to the destination (from the current location). Repeat this step until you reach the destination or all the paths to the destination are blocked by obstacles

The above algorithm has the completeness property that the vehicle will find a feasible path to the destination if there exists one.

5.2 Threat Search

The aim of the threat search is to find all the SAMs in a given area. For this search, we use a adaptive space filling algorithm as follows:

- Generate a space filling curve that covers the given area
- Fly the generated curve until one of the following happens:
  - · If a new threat is observed, then fly the minimum risk path to the next destination on the curve. Repeat this process until all the destination points in the curve are used.
  - · Final destination on the curve is reached.

6. THE UAV TEAM ARCHITECTURE

6.1 Architecture overview and design guidelines

The system goal is to employ a team of UAVs to search collaboratively an unknown environment while avoiding obstacles and threats. The UAVs have different capabilities (e.g. different sensors accuracy and type, speed, endurance), that have to be used efficiently to complete the mission. Given the nature of the scenario we design the system to have the following four properties: modularity, safety, resilience, scalability. The system consists of heterogeneous components, e.g. different UAVs equipped with different sensors. We want this heterogeneousness to be transparent to the system. In order to achieve this goal we made the design modular: we decompose the system into logical blocks (e.g. sensor controller, UAV autopilot) and we define the interfaces between them. In this particular scenario the set of available resources dynamically changes: resources may be added on the run, while other may be destroyed or damaged. The system must be able to cope with these continuous changes. It has to be scalable, in order to be able to operate with all the available resources and accommodate additional resources. It has to be resilient to failures in order to continue to operate even when some of the resources are destroyed or damaged. The system performances should degrade gracefully. Scalability and resilience are addressed using a layered architecture that offers different level of control. Since the environment is unknown and potentially unsafe, the system should try to minimize destruction or damage to the UAVs. Moreover the output of the available sensors is subjects to errors and long processing delay. The sensor inaccuracy problem has

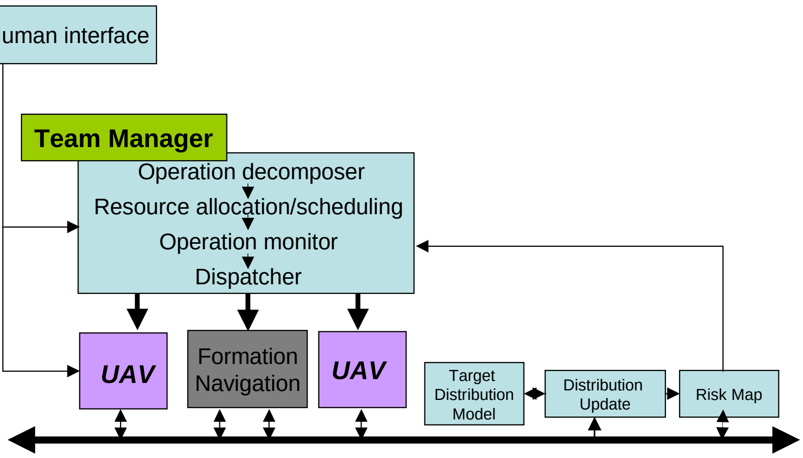

Fig. 5. Overview of the Architecture.

been addressed by sharing the threat information among the members of the team. The processing delay problem has been addressed by having every single UAV adopting a safe flight strategy (as described in section 4). The high level structure of the system architecture is given in figure 5. The system consists on three main components: the Team Manager, the UAV Managers and the Sensor Information Processing Unit(Risk map, Target distribution modul e, Prob update module). The UAV Managers offers a higher level of control of a single UAV to the Team Manager: instead of flying the UAV waypoint by waypoint, we can command it to execute a more complex task (e.g. flying to an area while avoiding obstacles or search an area for threats). The Sensor Information Processing Unit is used to share and fuse the information gathered by different sensors from different UAVs. The Team Manager block coordinates and monitors the team members, and it offers a higher level of control. Instead of controlling every single UAV using tasks we can assign to the team a mission (e.g. efficiently search an area or look for a minimum risk corridor).

6.2 Team Manager

The Team Manager offers a higher level of control above the single UAV task control level. It allows the user to control a team of UAVs to collaboratively perform a mission, such as target search in an unknown environment and safe corridor discovery. The Team Manager components is also given in figure 5. A mission is broken down by the Operation Decomposer into a sequence of tasks. The resource allocation block then allocates the available resources to perform these tasks. The dispatcher assigns the tasks to the allocated resources. The Operation Monitor monitors the system status. If some of the resources are destroyed, or new ones become available, it prompts the Resource Allocator to redistribute the task load among the new set of resources. This component allows the system to dynamically adapt to changes.

6.3 UAV Manager

The current available systems, for example the Cloudcap Piccolo system offer a low level of control, usually a waypoint navigation control without obstacle and threat avoidance. The UAV manager is built on top of such systems in order to provide a higher level of control as well as obstacle avoidance. The structure of the UAV manager is described in figure 6. The Sensor Manager deals with the complexity of the sensors using the information provided by the Dynamic Path Planer (e.g. the direction and the speed of the motion). It takes care of the sensor aiming and ensures that the sweeping pattern does not leave behind unsearched areas. The autopilot deals with the waypoint navigation (without obstacle avoidance). An example of such an Autopilot is the just mentioned Cloudcap Piccolo system. On top of it, we built a safe controller. The Dynamic Path Planer (DPP) creates a minimum risk path (using the risk map) between the current position and the destination given by the task manager, avoiding known obstacles and threats. The task manager provides a task execution service to the above layers.

If any threat is detected, the safe controller stops the autopilot from following the assigned path and interrupts the Dynamic Path Planner for a revised nominal path. Meanwhile, the safe controller

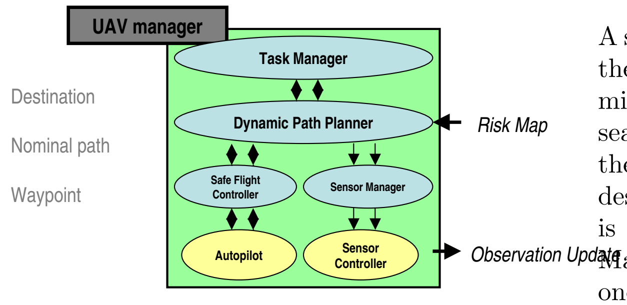

Fig. 6. UAV Manager

generates dummy paths that forces the vehicle to just fly around a fixed point until it receives a new nominal path from the DPP to follow. This interrupt from the safe controller ensures that irrespective of the decisions that are taken at the higher levels of the architecture, it always controls the safety of the vehicle.

6.4 Architecture implementation using an example

Lets consider an example of a search area mission. The following are the break up of the functions of each block in the proposed architecture:

- The *Operation decomposer* produces the following list of tasks: fly to the area to be searched, search the area and fly back home.
- The *Resource Allocator* assigns each of these tasks to the team of UAVs available. The following three tasks are assigned to every UAV: a fly to to the search area, a search area task, where the area to be searched is a fraction of the total area to be searched and a fly to the base. If any of the UAVs are destroyed or damaged, the *Operation Monitor* prompts the Resource Allocator to redistribute the areas to be searched among the remaining UAVs.
- Each *UAV manager* is assigned with an area to searched. The *Task Manager* produces a set of way points to be followed based on a adaptive space filling algorithm for the given search area.
- *Dynamic path planner* generates the minimum risk path (nominal path) to any destination way point specified by the task manager. The *safe flight controller* refines the path with the safe flight design.
- Apart from the functions that the *sensor manager* performs as mentioned before, a human interface is provided to pick the type of sensor to be used in the mission. SAR sensors can be first used to identify all the main threats that can destroy the vehicle such as SAMs. Once they are identified, then EO sensors can be used to classify targets such as trucks or buses accurately.

A simulation of the above search mission to find all the SAMs is shown in figure 7. One of the other missions that we implemented was the strategic search mission. This mission would be useful when the aim is to find a feasible safe path to the destination. In this mission, the only real part that is different from the search mission is the Task Manager. In this case, the Task Manager has just one destination point to reach. A simulation of this mission is shown in figure 8. The following subsection lists the properties of the architecture that we presented above.

6.5 Properties of the Architecture

The interrupt that the safe controller has basically provides the completeness property of the strategic search that if a there exists a safe path, the components in the architecture will guide the vehicle to find that path. The safe controller by refining the nominal path to the destination provides sufficient time for the sensors to process the information about the scanned region in front of the vehicles for threats. If the sensors report any presence of SAMs, then the entire mission is immediately stopped and the interrupt is passed on to the Dynamic path planner. The Dynamic path planner then again probes the sensor information processing unit for the latest risk map with the new updates. The DPP uses this new risk map to find a feasible nominal path and this process repeats itself according to the strategy outlined in section 5. Thus the completeness property of the strategy is satisfied. Also, the presented architecture has a property of information adequacy. That is, the safe controller generates a path such that the information that is gathered by flying the refined safe path is atleast equal to the information that would have been gathered by flying the nominal path generated by DPP. The next section discusses the Mission control interface that we built for the architecture.

7. SIMULATION

An implementation of the proposed architecture has been developed on Mixed Initiative Control for Automata-teams (MICA) Open Experimental Platform (OEP) Simulator. We implemented also a Mission Control interface to interact with the system. The interface is shown in figure 9.

The user specifies a mission (currently only a search area mission is implemented) and its parameters (e.g. the area to be searched, some constraints on the sensor type to use, etc).Then

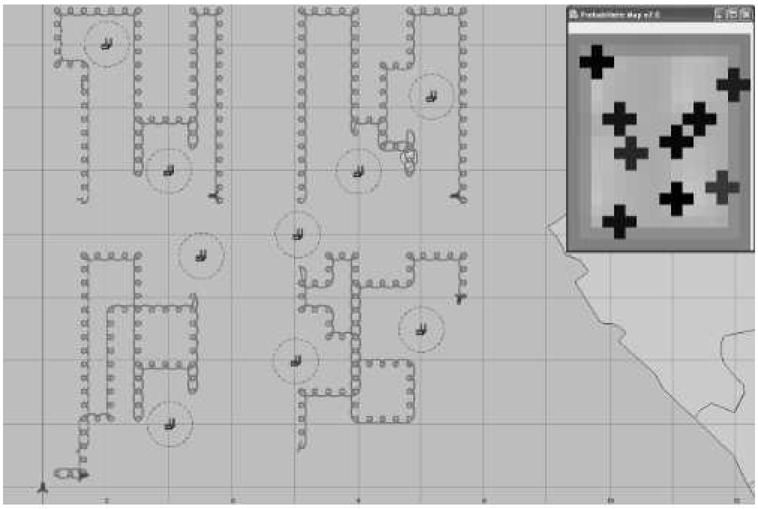

Fig. 7. Search mission

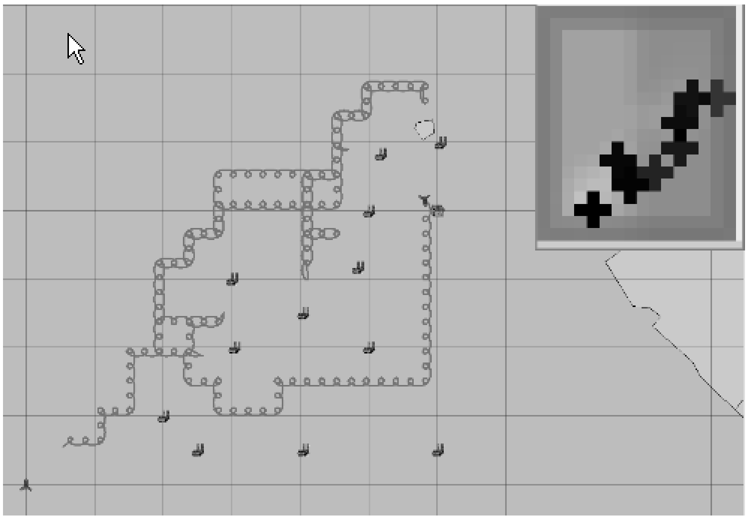

Fig. 8. Strategic search

he/she monitors the mission progress. The user interface displays the location of the UAVs and of the detected objects (in the current applications trucks, SAM and busses) in real-time. When an object is detected but cannot be classified (because, for example, SAR sensors cannot distinguish between truck and busses since they belong to the same visually similar objects group), the user can interactively dispatch a new UAV equipped with EO sensors for classification. Again, the results are displayed in real time on the user interface.

8. CONCLUSION

An architecture with the algorithms and its implementation for controlling a team of vehicles for the search mission was presented in the paper. Some of the assumptions (which also are the drawbacks) of the current work are as follows:

- Sensors are perfect or the probability of detecting a SAM launcher is 1 if sufficient amount of time is spent.
- A simple stick model is used for the kinematics of the UAV and only 2-D motion is considered.
- The sensor control onboard UAV's are capable of scanning a specified part of an area irrespective of the direction of the UAV.

The following are the future directions of this current work

- The resource allocation during cooperation between the vehicles has not been solved.

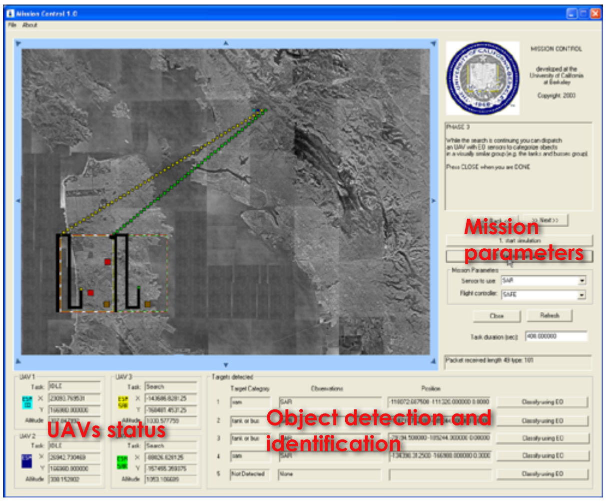

Fig. 9. Mission Control user interface

Also, the question as to how to redistribute the resources when the vehicles are shot has not been answered

- The information structure in the presented architecture was centralized. In other words, the information about the entire region or the risk map is known to all the vehicles. So when large number of vehicles are operating, the aim now is to make this information structure decentralized.

9. ACKNOWLEDGEMENTS

The authors thank Prof Pravin Varaiya, University of California, Berkeley and Prof Joao Sousa, Universidade do Porto, Portugal for their useful comments. The research was supported in part by the Boeing Mica Program - grant #F33615-01-C-3150 and by ONR AINS Program - grant #25833.

10. APPENDIX

Lemma:

Assume x $\geq$ y. If x $\geq \frac{v}{\omega}$ and if y $\geq \sqrt{\frac{xv}{\omega}}$, the yaw rate constraint is always satisfied.

Proof: The speed of a UAV is a constant and is always along the direction of the path. Let the ellipse be represented in a parametric form by (X(t),Y(t)), where X(t) = $x \cos t$,Y(t) = $y \sin t$ and t varies in the interval $[0, 2\pi]$. Also let the radius of curvature and the angular velocity at any point be denoted by $\rho(t)$ and $\dot{\Phi}(t)$. Since the velocity vector of the vehicle is always tangent to the path of the vehicle and its magnitude is a constant, the following claim is true:

$$\max_{\forall t \in [0,2\pi]} \dot{\Phi}(t) \sim \min_{\forall t \in [0,2\pi]} \rho(t) \tag{10}$$

The radius of curvature is given by,

$$\rho(t) = \frac{(X'^2 + Y'^2)^{\frac{3}{2}}}{X'Y'' - Y'X''} \tag{11}$$

where X' = $\frac{dx}{dt}$ and Y' = $\frac{dy}{dt}$. Hence by substituting the parametric forms for X and Y, we have,

$$\rho(t) = \frac{(x^2 \sin t^2 + y^2 \cos t^2)^{\frac{3}{2}}}{xy} \tag{12}$$

It is trivial to see that the radius of curvature is the same at all the points if x = y. $\rho(t)$ is minimum at t $=\{0, \pi\}$ if x > y. The minimum value for x $\geq$ y is equal is equal to $\frac{y^2}{x}$. Hence for the yaw rate condition to be satisfied, x $\geq \frac{v}{\omega}$ and $\frac{y^2}{a} \geq \frac{v}{\omega}$. Hence proved.

11. REFERENCES

Beard R.W., T.W. McLain, M. Goodrich, and E.P. Anderson, "Coordinated target assignment and interceptfor unmanned air vehicles", in *IEEE Transactions On Robotics and Automation*, 18:911 922, 2002.

Bellingham J., M. Tillerson, A. Richards, and J. How, "Multi-task allocation and path planning for cooperative UAVs", in *Cooperative Control: Models, Applications and Algorithms*, Editors - S. Butenko, R. Murphey, and P.M. Pardalos, Kluwer Academic Publishers, pages 2341, 2003.

Bortoff S.A., "Path-planning for unmanned air vehicles ", Summer report, AFRL, 1999.

Chandler P., S. Rasmussen and M. Pachter, "UAV cooperative control," in *Proceedings of American Control Conference*, Arlington, VA, June 25-27 2001.

Dubins L.E., "On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents", in *American Journal of Mathematics*, vol 79, Issue 3, July 1957, pp. 487-516.

Hwang Y.K., N. Ahuja, Gross Motion Planning - A Survey, ACM Computing Surveys 24, no. 3, pp. 219-291, September 1992.

Jacques D.R., "Search, classification and attack decisions for cooperative wide area search munitions", in *Cooperative Control: Models, Applications and Algorithms*, Editors - S. Butenko, R. Murphey, and P.M. Pardalos, Kluwer Academic Publishers, pp 7593, 2003.

Latombe J.C., *Robot Motion Planning*, Kluwer Academic Publishers, Norwell, Mass. 1991.

Laumond J.-P., *Robot motion planning and control*, Springer-Verlag, 1998.

Leachtenauer J.C. and R.G. Driggers, *Surveillance and Reconnaissance Imaging Systems - Modelling and Performance Prediction*, Massachusetts: Artech House, 2001.

Lee, Huang, Vaughn, Xiao, Hedrick, Zennaro, Sengupta, *Startegies of Path-Planning for a UAV to Track a Ground Vehicle*, AINS Conference 2003

Lumelsky V. and A. Stepanov , "Dynamic path planning for a mobile automaton with limited information on the environment," in *IEEE Transactions on Automatic Control*, vol. 31, no. 11, pp. 1058-1063, 1986.

Mclain T.W., "Coordinated control of unmanned aircraft vehicles", Summer report, AFRL, 1999.

McLain T.,"Cooperative rendezvous of multiple unmanned air vehicles," in AIAA GNC 2000, Denver, CO, Aug 2000, Paper no. AIAA 2000-4369.

Murphey R.A., "An approximate algorithm for a weapon target assignment stochastic program, in *Approximation and Complexity in Numerical Optimization: Continuous and Discrete Problems*, P.M. Pardalos, Ed., Boston: Kluwer Academic Publishers, pp. 1-2, 1999.

Nygard K.E., P.R. Chandler and M. Pachter,"Dynamic network flow optimization models for air vehicle resource allocation," in *American Control Conference 2001*, Arlington, VA.

Pachter M. and P. Chandler, "Challenges of autonomous control", IEEE Control Systems Magazine, pp. 92-97, April 1998.

Passino K.M., M.M. Polycarpou, D. Jacques and M. Pachter, "Distributed cooperation and control for autonomous air vehicles", in *The Workshop for Cooperative Control and Optimization*, Gainsville, FL, December 3-5 2000.

Polycarpou M.M., Y. Yang, and K. Passino, A cooperative search framework for distributed agents, in *Proc.IEEE ISIC2001*, pp 16, 2001.

Rao N., S. Hareti, W. Shi, and S. Iyengar, *Robot navigation in unknown terrains : Introductory survey of non-heuristic algorithms*, Technical Report ORNL/TM-12410, Oak Ridge National Laboratory, 1993.

Rathinam S. and R. Sengupta, "A Safe Flight Algorithm for Unmanned Aerial Vehicles", to be presented in *IEEE Aerospace Conference*, Montana, 2004.

Stentz A.,"Map-based Strategies for Robot Navigation in Unknown Environments", in *Planning with Incomplete Information for Robot Problems: Papers from the 1996 AAAI Spring Symposium*, AAAI Press, Menlo Park, California, pp. 10-116, 1996.