

REPORT DOCUMENTATION PAGEForm Approved
OMB NO. 0704-0188

Public Reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comment regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. AGENCY USE ONLY (Leave Blank)		2. REPORT DATE Jan 1, 2002		3. REPORT TYPE AND DATES COVERED Technical Report March 1, 2001- January 1, 2002	
4. TITLE AND SUBTITLE Hierarchical System for Multiple-Agent Scenarios				5. FUNDING NUMBERS DAAH04-96-1-0341	
6. AUTHOR(S) H.J.Kim, D.H. Shim, S.Rashid and S. Sastry					
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Electronics Research Laboratory University of California Berkeley, CA 94720				8. PERFORMING ORGANIZATION REPORT NUMBER	
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) U. S. Army Research Office P.O. Box 12211 Research Triangle Park, NC 27709-2211				10. SPONSORING / MONITORING AGENCY REPORT NUMBER 35873-MA-MUR 177	
11. SUPPLEMENTARY NOTES The views, opinions and/or findings contained in this report are those of the author(s) and should not be construed as an official Department of the Army position, policy or decision, unless so designated by other documentation.					
12 a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution unlimited.				12 b. DISTRIBUTION CODE	
13. ABSTRACT (Maximum 200 words) This paper presents a hierarchical architecture for the coordination and control of autonomous agents performing intelligent team operations. Each team, consisting of multiple aerial and ground vehicles, uses a coordinated strategy through communication via a wireless network. As an exemplary case study, a pursuit-evasion scenario is developed. This paper also introduces the experimental setup for aerial and ground-based autonomous agents. The proposed scheme is currently under development for near future experiments.					
14. SUBJECT TERMS Hierarchical system; Multiple-agents scenarios.				15. NUMBER OF PAGES 8 pp	
				16. PRICE CODE	
17. SECURITY CLASSIFICATION OR REPORT UNCLASSIFIED	18. SECURITY CLASSIFICATION ON THIS PAGE UNCLASSIFIED	19. SECURITY CLASSIFICATION OF ABSTRACT UNCLASSIFIED	20. LIMITATION OF ABSTRACT UL		

NSN 7540-01-280-5500

Standard Form 298 (Rev.2-89)
Prescribed by ANSI Std. Z39-18
298-102

20030514 130

Hierarchical System for Multiple-agent Scenarios

H. Jin Kim*, David H. Shim*, Shahid Rashid**, Shankar Sastry***

*Graduate Students, Department of Mechanical Engineering
University of California at Berkeley

** Graduate Student, Department of Electrical Engineering and Computer Science
University of California at Berkeley

***Professor, Department of Electrical Engineering and Computer Science
University of California at Berkeley
{jin, hcshim, rashid, sastry}@robotics.eecs.berkeley.edu

ABSTRACT

This paper presents a hierarchical architecture for the coordination and control of autonomous agents performing intelligent team operations. Each team, consisting of multiple aerial and ground vehicles, uses a coordinated strategy through communication via a wireless network. As an exemplary case study, a pursuit-evasion scenario is developed. This paper also introduces the experimental setup for aerial and ground-based autonomous agents. The proposed scheme is currently under development for near future experiments.

1. Introduction

Intelligent multi-agent systems have been of great interest recently because they offer rich sets of challenging questions to address: optimization, coordination, fault detection-tolerance, stability, and communication among them. The *BErkeley AeRobot (BEAR)* project at University of California at Berkeley started as development of a single autonomous Unmanned Aerial Vehicle (UAV) and now one of current areas of the research activities is the creation of an intelligent network of ground and aerial vehicles performing coordinated operations. The goal of this research is the organization of multiple autonomous agents into integrated and intelligent systems with reduced control and cognition complexity, fault-tolerance, adaptivity to changes in task and environment, modularity and scalability to perform complex missions efficiently. This project encompasses diverse active research topics: 1) multi-agent coordination, 2) hybrid system synthesis and verification, 3) communication, 4)

navigation and 5) vehicular system identification and control synthesis.

Among many scenarios, the pursuit-evasion or search-rescue mission is particularly attractive since it addresses most of the interesting issues aforementioned. One team of agents play *pursuers* or *rescuers* trying to catch a team of evading agents, which move around in random or intelligent manner, or locate the objects of interest while minimizing some cost function.

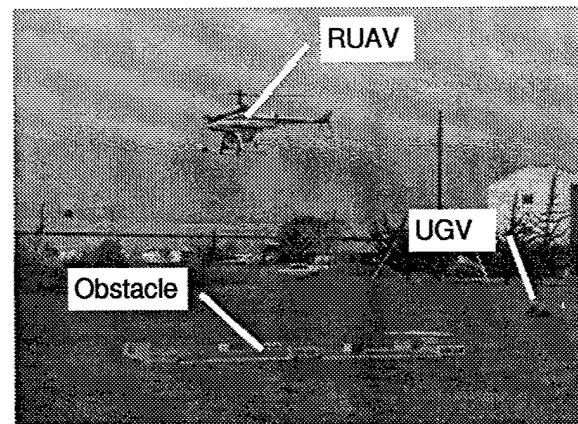


Figure 1. Berkeley testbed for pursuit-evasion game

In our implementation, UAVs are pursuers, which build a probabilistic map of the field using a vision system in real-time and assume a flight profile to capture the evaders. In doing so, UAVs are required to fly through the given waypoints or hover over certain points. Rotorcraft-based UAVs (RUAVs) are very attractive for this application because of their maneuverability. Unmanned ground vehicles (UGVs) play the role of evaders, traveling on the surface of

the arena, following a certain motion rule. Depending on the scenario, some UGVs can act as pursuers in a cooperative network with UAVs.

Wireless communication serves as the backbone to connect individual agents for exchanging necessary information. While the current setup assumes a fixed number of communication nodes throughout the mission, more advanced wireless network architectures using dynamic clustering are currently under development.

In this paper, we introduce the research activities for the development of multi-agent coordination systems, emphasizing the system architecture and realization of pursuit-evasion or search-rescue scenarios.

2. BEAR System Architecture

Our architecture of multi-agent distributed system is inspired by *hierarchical hybrid systems* [7]. This will allow integration and synchronization of global plans from local intentions of each agent in the perspective of multiple-UAV/UGV operations. Each UAV consists of the base airframe and the integrated avionics systems, which includes path planning and flight control system, navigational sensors, and a communication module (Figure 2). UGVs are similarly constructed. Strategy planning and probabilistic map building of the environment may be done on-board or in ground station depending on the choice of informational centralization.

2.1 Strategy Planning

The strategic planner is responsible for overall planning for the execution of mission. It decomposes a mission into a sequence of waypoints. In addition, it acknowledges the completion of a subtask, such as arrival at a waypoint, and schedules the next one. When the tactical layer notifies this layer of the potential conflict, it will generate proper constraints for conflict-free trajectories. In multi-agent systems, this layer allocates resource needed to accomplish the mission efficiently.

In our pursuit-evasion or search-rescue type scenarios, the proper policy is employed to generate waypoints for each agent in this layer. Using the collected information by the map builder, the strategy planner calculates the tactical movements of the pursuing agents at next time frame and distributes them through the wireless communication network.

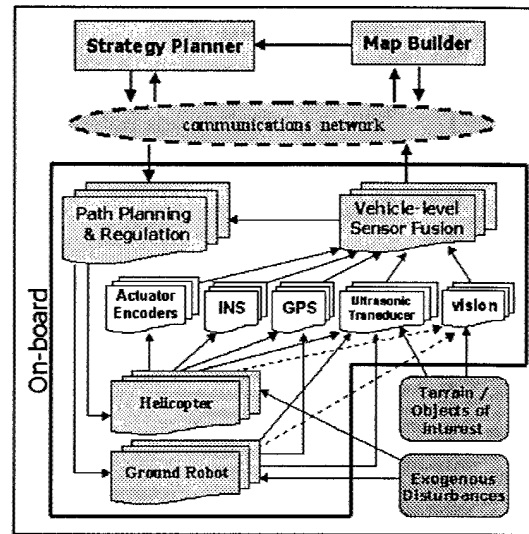


Figure 2. BEAR System Architecture

2.2 Path Planning and Regulation Layer

This layer, composed of the trajectory planner and vehicle regulation, resides on each agent to generate a realizable path for each agent to follow and control the host vehicle to track the given path.

The trajectory planner designs a realizable trajectory for each agent and associated flight modes, based on a detailed dynamic model of the RUAV and the trajectory from tactical planner. Different flight modes such as take-off, hover, cruise, turn, etc. may lead to multiple sets of control laws. Moreover, the resulting trajectory is given to the regulation layer to directly control the dynamics of each agent. Thus, the transfer between controllers is desired to be bumpless and the issue of actuator saturation should be considered in generating trajectory constraints.

The motion request by Strategy planner is transmitted to this layer via wireless communication network. The motion request is cast in the form of *Vehicle Control Language (VCL)*, a human-understandable script language proposed by Shim[10]. This structure delivers the navigation information including coordinates of target waypoint, type of waypoints and other requirements. Based on the contents of VCL, PPR determines the feasible flight mode, generate reference trajectory in realtime and feed it to the integrated regulation layer.

Regulation layer plays the important role to stabilize the inherently unstable dynamics of the host vehicle and track the given trajectory. The underlying feedback controller is currently based on classical multi-loop SISO controllers as shown in Figure 3.

The feedback compensation gains are determined applying classical controller design framework to the LTI model for hover [10].

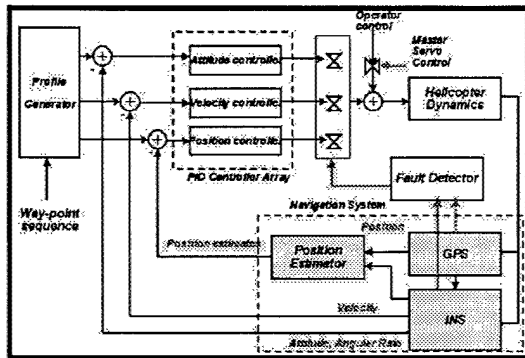


Figure 3. The proposed controller architecture using SISO multi-loop controllers

2.3 Vision Systems of RUAV

The onboard vision system of RUAV consists of color CCD camera, color tracking devices and dedicated vision processing computer. The current vision processing system uses color segmentation to identify objects from camera images. In this scheme, the ground agents are marked with certain color of high contrast with background. The vision computer computes the localized coordinate of the target and then converts it into the global coordinates using the position and attitude information estimated by the onboard vision system. The estimated positions of the evaders are then reported to the map builder via wireless communication network.

2.4 Map Building

Based on sensory information, this layer dynamically builds a representation of features of the environment that is relevant for the navigation of agents. This map, which contains observation and possible positions of objects of interest, will be sent to the strategy planner and used as a basis for planning and performing of tasks. Depending on the scenarios and the computational load of the onboard computers, this is done either on each agent or centrally by our ground station.

2.5 Communication Network

The multi-agent scenario requires communication channel among the participating agents. Since the participating agents are moving in the arena freely, wireless communication is preferred unquestionably. The scenario that the agents in each group exchange the position information requires peer-to-peer

communication setup than one-to-one format, which is typical setup for serial communication. In this research, Lucent Technology's Orinoco system is chosen as the communication backbone. Orinoco system supports TCP/UDP/IP in user-selectable speeds from 2Mbps to 11Mbps in 2.4GHz band.

In our small-sized test field, we notice no packet loss, but can expect imperfect communication between agents in a real scenario. One of the active areas of research within the group is on how to share information and coordinate actions with an unreliable communication channel.

2.7 RUAV Platform

An RUAV should be able to maneuver through the given waypoints while searching for ground agents using vision processor. Small-size helicopters are chosen for the aerial agents because of their flexible maneuverability such as vertical take-off/landing, hovering, side-step flight and forward flight. The capability of hovering and low velocity forward/lateral maneuver is very valuable when they need to track ground-based agents. They can be also operated in a relatively small area because they do not need a runway to take off and land.

Four different sizes of helicopters are considered for RUAV application: Yamaha industrial helicopter RMAX and R-50, Bergen Industrial Twin, and Kyosho Concept 60. Among these, Yamaha helicopters are used as the aerial agents for PEG because of their sufficient payload and reliability. The aerial agents are equipped with autopilot system and vision processing computer. The autopilot system is divided into navigation sensor suite and the flight computer. The navigation sensor suite consists of inertial navigation system (INS), global positioning system (GPS), and ultrasonic height sensors. For INS, Boeing DQI-NP is chosen for the built-in data processing capability, compact size and GPS-integrability option. NovAtel GPS MillenRT-2 provides the position information updated at 4HZ with approximately 2 cm. Ultrasonic sensors provide the local altitude information valuable for autonomous take-off and landing stage.

Flight control computer (FCC) is Intel Pentium-based IBM-PC compatible system in PC104 industrial standard. FCC is in charge of sensor management, control command generation and wireless communication. The onboard flight control software is running on QNX realtime operating system. Control outputs for four channels, main rotor

collective pitch, tail rotor collective pitch, main rotor longitudinal and lateral cyclic pitch, are computed using a programmed control law for stabilization and tracking of the host vehicle. FCC also downloads the current flight status using Orinoco wireless LAN at 2.4GHz.

2.8 Ground Robots

Since the experimental set-up requires the UGVs to be operated outdoors to interact with UAVs, Activmedia Pioneer 2-ATs were chosen. These rugged UGVs are four wheel drive, differential skid-steering robots designed for all-terrain operations.

For self-localization or position estimation of robots, the usual approach is based on a combination of dead reckoning with periodical compensation using external information to keep the accuracy of position from gradually decaying. This external information is obtained from active/passive landmarks or from the matching between a global map and the information provided by agents. However, we are mainly interested in operations in which *a priori* environmental information is unavailable, so GPS is chosen as the primary navigation sensor. Other components for sensing and navigation include position encoders, digital compass, and range-finding ultrasonic sonar transducers.

Currently the UGVs are programmed for navigation in an outdoor environment using Saphira motion control software. The strategy planner can access the integrated onboard PC through RS232. Then the Saphira OS accepts the motion command from the upper layer and steers the host vehicle to the desired position by transmitting appropriate motor commands to the robot. The robot has two independent control channels for transition and rotation and commands to control them can be issued and executed concurrently.

Shaphira also has several functions to look at the raw sonar readings and determine if an obstacle is near the robot. These detection functions either look at a rectangular region in the vicinity of the robot or a portion of a half-plane.

3. Multi-player Pursuit-Evasion Game

To validate the proposed architecture, a particular kind of game scenario is conceived in which a group of *pursuers* are attempting to capture another group of *evaders* within a fixed and unknown *arena* which

may contain fixed obstacles. The discrete-time game is implemented algorithmically on a discrete map over which the pursuers assign a probability of evader occupation. The pursuers use their observations at each time instant $t \in \mathcal{T} = \{1, 2, \dots\}$ to update the perceived state of the arena and then predict the state of the arena, particularly the location of the evaders, at the next time instant.

We denote by $y(t)$ the set of measurements taken by the pursuers at each time $t \in \mathcal{T}$. Every $y(t)$ is assumed to be a random variable with values in a measurement space \mathcal{Y} . Each control action $u(t) \in \mathcal{U}$ is a function of $Y_t \equiv \{y(1), \dots, y(t)\}$, the sequence of measurements taken up to time t . By the pursuit policy we mean the function $g: \mathcal{Y}^* \rightarrow \mathcal{U}$ such that

$$u(t+1) = g(Y_t)$$

for each $t \in \mathcal{T}$, where \mathcal{Y}^* is the set of all finite sequences of \mathcal{Y} . We say that an evader was found at time $t \in \mathcal{T}$ when one of the pursuers is located at a cell for which the conditional probability of the evader being there, given Y_t , exceeds a certain threshold. T^o represents the first time instant at which one of the evaders is found. $E_{\bar{g}}[T^o]$, the expected value of T^o under a specific pursuit policy $\bar{g}: \mathcal{Y}^* \rightarrow \mathcal{U}$ provides a good measure of the performance of \bar{g} . However, since the dependence of $E_{\bar{g}}[T^o]$ on \bar{g} is very complex, finding an optimal policy that minimizes $E_{\bar{g}}[T^o]$ is difficult and not suited for real-time applications. In this research, we will concentrate on suboptimal policies and compare the performance of different strategies with three pursuers and varying number of evaders.

Suppose that n_p pursuers try to find a single evader. Then, at each $t \in \mathcal{T}$, the position of the pursuers $\mathbf{x}_p(t) \equiv \{\mathbf{x}_1(t), \dots, \mathbf{x}_{n_p}(t)\} \in \mathcal{X}^{n_p}$ and the position of the evader $\mathbf{x}_e(t) \in \mathcal{X}$ can be considered as random variables. If a model for the motion of the evader is assumed to be known, for $\forall x \in \mathcal{X}, Y_t(t) \in \mathcal{Y}^*$,

$$p_e(x, Y_t) = P_e(\mathbf{x}_e(t+1) = x | Y_t = Y_t)$$

can be computed recursively as a deterministic function of the last measurement $y(t)$ in Y_t ,

and $p_e(x, Y_{t-1})$, where Y_{t-1} is the first $t-1$ elements of Y_t . This conditional probability will be used to generate pursuit policy $u(t)$, as to be explained later.

Once waypoints for movement are planned as $u(t)$, a realizable trajectory is generated by incorporating the continuous-time dynamic vehicle model and regulation layer. The game terminates when the pursuers capture all the evaders in the arena.

Among the many development tools used to simulate hybrid behavior, two different simulation environments are adopted in this research: Hybrid System Tool Interchange Format (SHIFT) and MatLAB/Simulink. Although SHIFT offers a number of features ideal for the simulation of a dynamically evolving hybrid network, MatLAB/Simulink is also used in this research because of its fast computation and convenient graphical user interface. Figure 4 shows all components of the system described in Section 2 built for simulation.

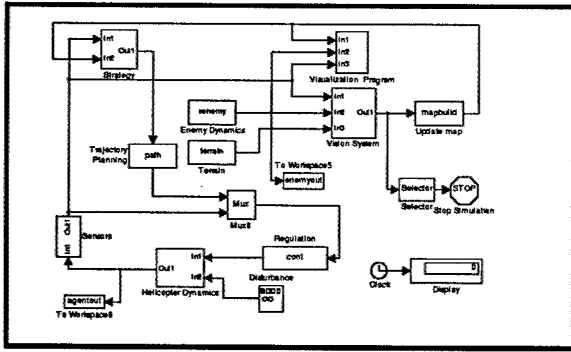


Figure 4. Hierarchical Structure Implemented in Matlab/Simulink

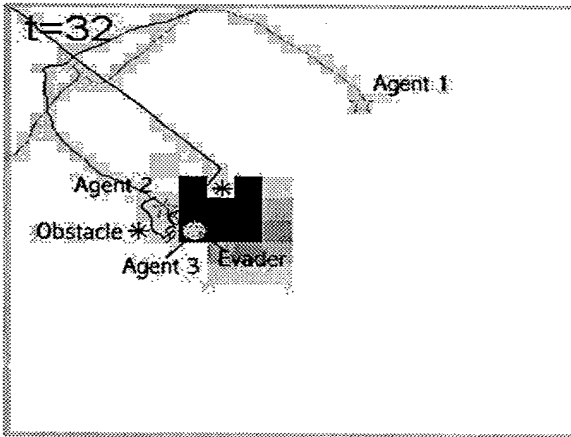


Figure 5. Trajectory of three agents until they capture the evader

Figure 5 shows the trail of three pursuers in $100m \times 100m$ arena until they capture an evader, when the capture is defined as collocation of a pursuer and an evader. Figure 6 shows the display environment for the visualization purpose of our scenarios.

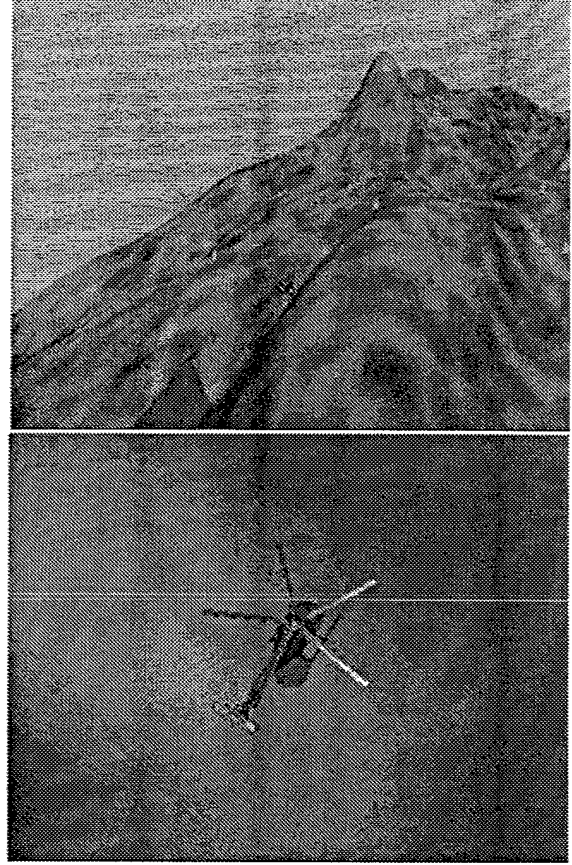


Figure 6. Simulation Display

An agent following an efficient strategy would be expected to perform thorough local searches, but at the same time adapt on a global scale to information supplied by other agents. A classical search strategy is an A*-type search, whereby each agent moves towards the global location which has the highest discounted probability of being occupied by an evader. The discounting factor is designed to be proportional to the distance from the agent to each location. When $x_p(t)$ denotes the position of pursuers at time t , this strategy can be expressed as

$$s_A = \arg \min_{\{v_1, \dots, v_{n_p}\} \in \mathcal{Z}(x_p(t))} d\left\{v_1, \dots, v_{n_p}\right\}, \arg \max_{\{w_1, \dots, w_{n_e}\} \in \mathcal{Z}} \left(\frac{\sum_{k=1}^{n_e} p_e(w_k, Y)}{d(x_p(t), \{w_1, \dots, w_{n_e}\})} \right)$$

,where d is the distance function according to the movement of pursuers and $\mathcal{Z}(x(t)) \subset \mathcal{Z}$ is the set of

cells reachable from $x(t)$ in a single step, i.e. for every $x = \{x_1, \dots, x_{n_p}\} \in \mathcal{X}^{n_p}$,

$$d(\{x_1, \dots, x_{n_p}\}, \{w_1, \dots, w_{n_p}\}) = 1 \\ \forall \{w_1, \dots, w_{n_p}\} \in \mathcal{Z}(\{x_1, \dots, x_{n_p}\})$$

Another obvious candidate for a search policy is just a simple greedy strategy, whereby each agent moves to the cell within its range that has the highest probability of containing an evader at the next instant, i.e.,

$$g_B = \arg \max_{\{v_1, \dots, v_{n_p}\} \in \mathcal{Z}(x(t))} \sum_{k=1}^{n_p} p_e(v_k, Y).$$

The former policy g_A leads to relatively poor performance as the agents do not make complete searches of the map, but rather traverse the map back and forth frequently. Moreover, without the proper coordination, pursuers tend to move toward the same place, losing the advantage of having multiple agents covering a wider region (Figure 7).

In [8], it was shown that the expected time needed to find the evader is finite under greedy strategy g_B . In fact, a simple greedy strategy does have good performance and computationally efficient as its computational cost depends only on $\mathcal{Z}(x(t))$ rather than the entire \mathcal{Z} . The drawback to the pure greedy policy, though, is that (even in the perfect communication case) each agent effectively works on its own and does not take advantage of information gathered by other agents. Considering that our interest is in the hierarchical and team-wise approach to the pursuit-evasion problem, a pure greedy strategy is not very appealing either (Figure 8).

One combined policy found to be especially effective is where agents in general follow a greedy strategy, but (a subset of the agents) are 'dispatched', i.e. follow a trajectory, to locations where enemies are thought to have been seen. This policy utilizes the thorough local search provided by a greedy strategy, yet still allows agents to react to information provided by other agents (Figure 9).

One of the extensions to the standard pursuit-evasion game we have considered is the inclusion of a supervisory agent [9]. This supervisory agent may have more accurate sensors and a longer range of vision, but cannot capture the evaders. These supervisory agents can be thought of as roughly analogous to satellites or Airborne Warning and Control Systems (AWACS) platforms in the modern

battlefield. These supervisory agents traverse the map in a pre-defined manner and report enemy sightings. If an agent is following a pure greedy policy, though, he will effectively ignore these updates and no improvement in performance is noticed when a supervisor is added to agents using pure greedy strategies. The greedy/dispatch strategy mentioned in the previous paragraph lends itself to a scenario with a supervisor quite naturally, and an improvement in performance is noted versus the case without a supervisor (Figure 7-9). Figures 7-10 show the data all averaged from 100 runs. In Figures 7-9, five left columns have no AWACS while the five right columns have one AWACS.

Figure 10 shows a comparison of the three aforementioned policies when 'intelligence' is added to the evaders. In an attempt to add more realism, the evaders were designed to hide so as not to be seen by the supervisory agents. Once an evader was spotted by a supervisory agent, it would hide near an obstacle until the supervisor had moved itself out of range. The initial spotting of the evader was sufficient, though, to dispatch a pursuer to the area, and in this scenario, our combined policy shows its true advantage over the pure greedy policy.

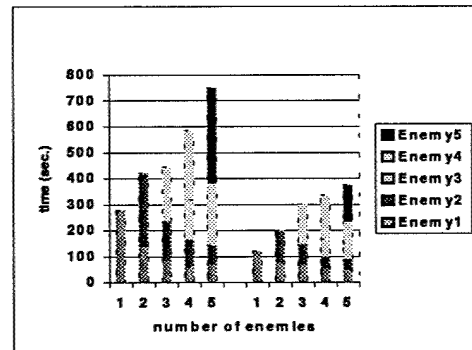


Figure 7. A* Policy

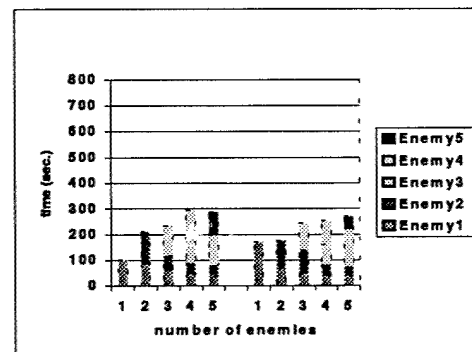


Figure 8. Greedy Policy

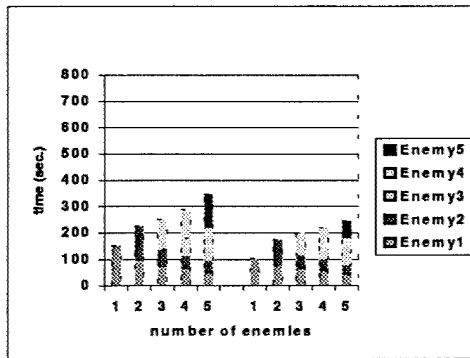


Figure 9. Combined Policy

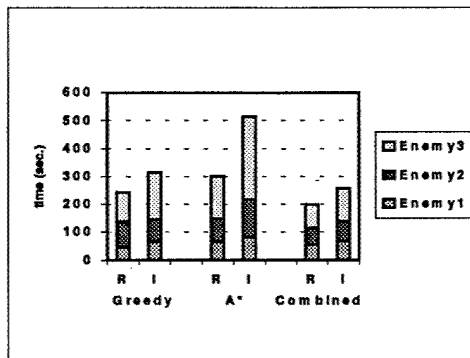


Figure 10. Comparison of the three policies for three random(R) and intelligent(I) evaders

Our research has focused primarily on the coordination of the movement of agents, but recently an emphasis has been placed on the coordination of information as well. The previous simulations all assumed perfect communication between all agents, i.e. each agent would update its own map with the sensor reading of all the agents and thus all agents effectively shared one complete map. A much more realistic scenario is the situation of imperfect communication between agents. Communication may be disrupted or corrupted for numerous reasons including propagation delays, the physical location of agents, loss of an agent, or jamming. The design of a network and communication structure that will be robust to errors and loss of communication between agents is of extreme interest. A strategy without communication between agents would intuitively be expected to give poor performance and this has indeed been confirmed by simulation. Various protocols for information sharing as well as flexible network structures are being explored to facilitate the creation of robust communications. The movement strategies will also be tuned to correspond to the reliability of information shared between agents.

4. Conclusion

This paper formulates a setup for the development of pursuit-evasion games using multiple UAVs and UGVs. The development of control and coordination algorithms and their test in simulation have been completed and the development of the physical testbeds is as well nearing completion. The future emphasis of this research will be on the investigation of more robust methodologies for complex and unreliable configurations. The implementation of the algorithms and software on the UAVs and UGVs using the outlined methodology will be followed by the completion of an actual field exercise.

5. Acknowledgments

This research was supported by ARO DAAH04-96-1-0341, ONR N00014-97-1-0946, DARPA F33615-98-C-3614, and Honeywell subcontract on DARPA N66001-99-C-8510.

6. References

- [1] R.W. Prouty, *Helicopter Performance, Stability and Control*, Krieger Publishing Company, 1995.
- [2] B. Mettler, M. B. Tischler, T. Kanade, "System Identification of Small-Size Unmanned Helicopter Dynamics," *American Helicopter Society 55th Forum*, Montreal, Quebec, Canada, May 1999.
- [3] H. Shim, T. J. Koo, F. Hoffmann, S. Sastry, "A Comprehensive Study of Control Design for an Autonomous Helicopter," *37th IEEE Conference on Decision and Control*, pp. 3653-3658, 1998.
- [4] C. P. Sanders, P. A. DeBitetto, E. Feron, H. F. Vuong, N. Leveson, "Hierarchical Control of Small Autonomous Helicopters," *37th IEEE Conference on Decision and Control*, pp. 3629-3634, 1998.
- [5] J. E. Corban, A. J. Calise, J. V. R. Prasad, "Implementation of Adaptive Nonlinear Control for Flight Test on an Unmanned Helicopter," *37th IEEE Conference on Decision and Control*, pp. 3641-3646, 1998.
- [6] L. J. Ljung, *Matlab System Identification Toolbox User's Guide*, The Math Works, Inc., 1997.

[7] C. Tomlin, G. Pappas, J. Lygeros, D. Godbole, S. Sastry, "A Next Generation Architecture for Air Traffic Management Systems," Memorandum No. UCB/ERL M97/7, Univ. of Cal. Berkeley, 1997.

[8] J. Hespanha, H. J. Kim, S. Sastry, "Multiple-Agent Probabilistic Pursuit-Evasion Games," 39th *IEEE Conference on Decision and Control*, pp. 3511-3516, 1999.

[9] H. J. Kim, S. Rashid "Multiple-Agent Probabilistic Pursuit-Evasion Games in 2.5 D," in *Hybrid Systems: Modeling, Analysis and Control*, Memorandum No. UCB/ERL M99/34, Univ. of Cal. Berkeley, 1999.

[10] D. H. Shim, H. J. Kim, S. Sastry, "Hierarchical Control System Synthesis for Rotorcraft-based Unmanned Aerial Vehicles," *AIAA Guidance, Navigation, and Control Conference*, Denver, 2000.