A Dynamic Multi-Robot Control System for Probabilistic Search and Rescue

Thesis proposal

For the degree of Master of Science in Computer Science

At Southern Connecticut State University

Richard S. Stephens

December 2003

Thesis Advisor: Dr. Hrvoje Podnar

Major-Field Approval – The advisor and the department chairperson

Advisor	Date
Chairperson	Date

A. Title

A Dynamic Multi-Robot Control System for Probabilistic Search and Rescue

B. Statement of Purpose

Search is an integral part of many robotic applications ranging from planetary exploration, examination and assessment of hazardous environments, rescue operations and urban warfare, to domestic applications. Robots provide a means to minimize human exposure to harmful and possibly life-threatening situations while providing a mechanism to perform potentially life-saving operations. The use of robotic platforms in treacherous environments such as space, sea, and hostile zones may in fact become a necessity in present day society as exemplified by the current environment in the Middle East where men and women are being placed in harms way on a daily basis.

This research examines and develops a multi-robot control system that is adaptive, fault tolerant, scalable, and serves as a computationally feasible dynamic solution to both search and

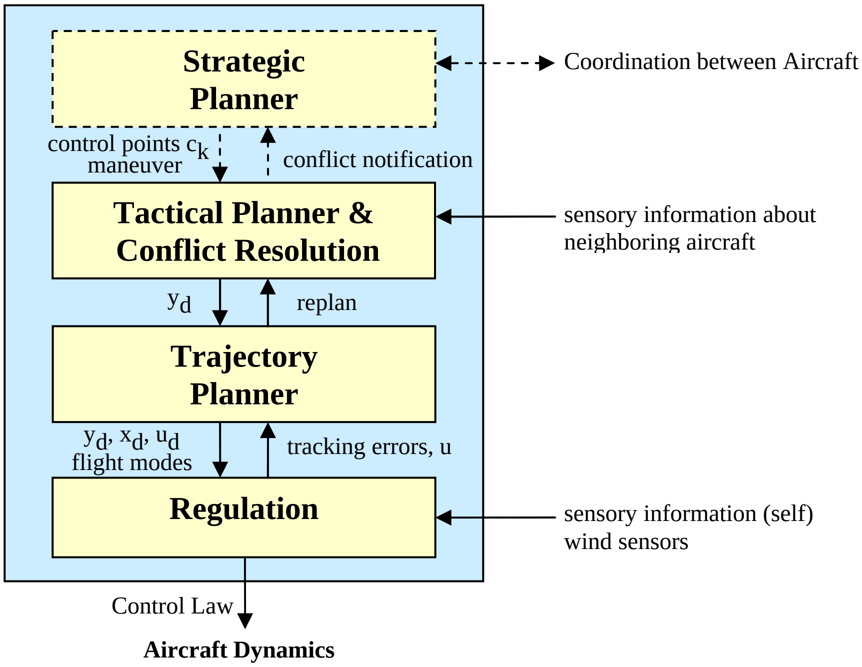

Figure 1. FVMS Structure

rescue and pursuit-evasion problems. The system framework will be a distributed hierarchical architecture that segments tasks into modules based on the Flight Vehicle Management System (FVMS) design **(**Sastry, Meyer, Tomlin, Lygeros, Godbole, & Pappas, 1995**)** shown in Figure 1. The modular architecture lends itself to scalability and allows one to implement a "divide-and-conquer" approach that simplifies the development and integration of a complex system.

In addition to enabling dynamic pursuit and search algorithms to be used in the Coordination and Policy Planning module, two additional policies will be implemented, namely, global-max and local-max. Both of these policies fall into the category of greedy policy in that they make the best choice at each time step. These policies are well documented and will be used to establish a consistent environment for comparison between existing systems and the proposed system. Time in steps to completion of goal will be measured for each methodology and compared against established results. The dynamic nature and capabilities of the algorithm will add new functionality and potentially improve upon the current performance characteristics of multi-robot control systems designed, for example, to find a victim, locate a gas-leak or unexploded mine, or explore a threatening environment while keeping people safe.

C. Literature Review And Current State-Of-The-Art

The classical approach to pursuit-evasion has been to build a map of the environment and then execute the system in the known environment. Several methods for map building have been proposed in (Thrun, Burgard, & Fox, 1998) and its references. Most of these techniques have been based on Bayesian estimation and implemented with Extended Kalman Filters. The problem with this approach has been that it is time consuming and computationally expensive even when limited to two dimensions (Deng, Kameda, & Papadimitriou, 1998).

Hespanha, Kim, and Sastry (1999) have developed a single probabilistic framework that combines the pursuit-evasion game with map building using a single evader. Multiple evaders were considered and simple vision-based algorithms for evader detection along with navigation, communication, and sensing routines were implemented in (Vidal, Rashid, Sharp, Shakernia, Kim, & Sastry, 2001). Intelligent evaders that actively avoid pursuers using a dynamic programming solution to a Stackelberg equilibrium of a partial information Markov process was proposed in (Hespanha, Prandini, & Sastry, 2000). Also, a computer vision approach utilizing optical flow to determine the number of evaders along with their position and orientation was explored in (Vidal & Sastry, 2002).

The multi-agent pursuit-evasion scenario considered for this project falls within the framework of multi-robot systems. A considerable amount of literature has been developed in this field. Systems dealing with machine learning were explored in (Stone & Veloso, 2000), hybrid control algorithms and distributed sensor fusion (Timofeev, Kolushev, & Bogdanov, 1999), localization techniques (Thrun et al, 1998) and (Roumeliotis & Bekey, 2000) and formation control (Desai, Kumar, & Ostrowski, 1999). Another area of interest in multi-robot systems research is robot soccer. Systems utilizing a centralized coordination approach are examined in (Han & Veloso, 1998), while decentralized coordination systems are studied in (Kitano, Asada, Noda, & Matsubara, 1998) and (Asada, Uchibe, & Hosoda, 1999).

To reduce computational complexity and calculation time, a probabilistic approach to pursuit-evasion including theoretic foundations was developed in (Hespanha, Kim, & Sastry, 1999) and (Vidal, Shakernia, Kim, Shim, & Sastry, 2002). The dependence equations for an optimal pursuit policy are very complex. Therefore, a pair of efficiently computable sub-optimal greedy pursuit policies called global-max and local-max was developed in (Vidal et al, 2002) and it was shown that the global-max policy outperforms the local-max policy in realistic situations.

D. Methodology

- *Single Robot Control System:*

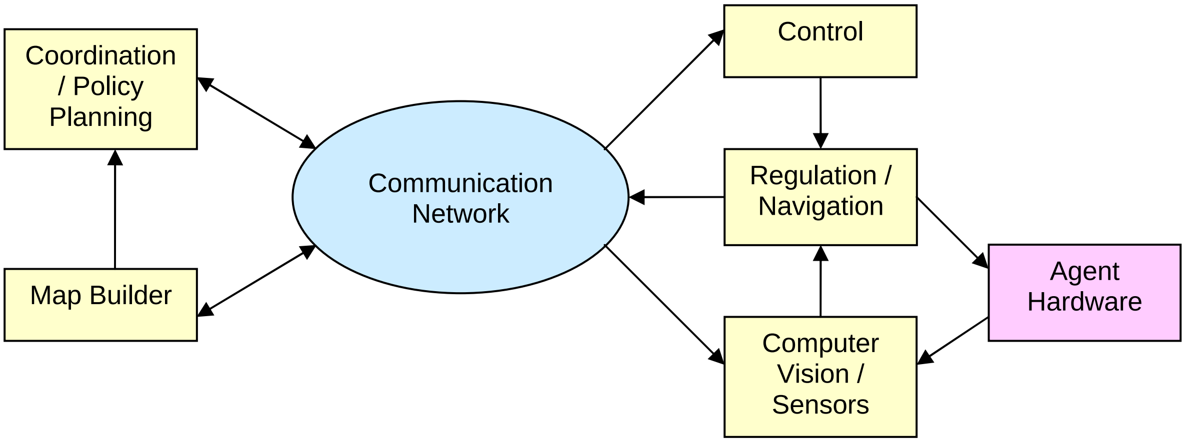

Figure 2. System Architecture for Single Hierarchical Robot Control System

The system architecture, depicted in Figure 2, is a hybrid hierarchical distributed architecture that segments the system into separate task specific modules. The following modules will be implemented as part of this research:

- *Coordination / Policy Planning (Strategic Planner)*

The *Coordination and Policy Planning* (CPP) module is at the top of the FVMS architecture and is responsible for the planning and execution of the central mission. The CPP instructs the Control module to execute the necessary commands to carry out the pursuit or search policy and accomplish the mission. The CCP also communicates with other individuals in the group to coordinate missions that require multiple participants.

- *Control (Tactical Planner & Conflict Resolution)*

The *Control* module is responsible for the coordination and execution of various behaviors such as searching an area or possibly approaching a way-point. The Control module should override the CCP instructions in safety critical circumstances such as collision avoidance. Once the safety issue has been resolved, control returns to the behavior.

- *Regulation / Navigation (Regulation / Trajectory Planner)*

The *Regulation and Navigation* (RN) module is responsible for translating the commands from the Control module into basic commands used to manipulate the robots, such as forward, turn left, turn right, stop, etc. The RN should ensure safe transitions between motion states.

- *Computer Vision/Sensing*

The *Computer Vision and Sensing* module is responsible for sensing the robot's environment. A wide variety of sensors, including cameras, thermal detectors, sonar, and lasers, can be implemented on robotic platforms and this module would need to integrate to the specific sensing device. For this project, a simple vision algorithm will be simulated.

- *Communication*

The *Communication* module is responsible for passing information between different modules on the same robot as well as inter-robot communication between CCP modules. A variety of different networking strategies may be implemented including direct robot-to-robot wireless, Ethernet and satellite links.

- *Map-Builder*

The *Map-Builder* module is responsible for constructing a map of the unknown environment based on the robot's sensor readings. Map information may be shared between robots over the Communication network based on mission requirements.

Together, these modules comprise the basic logical and intelligence components of the robotic platforms. A benefit to the modular design is that modules can be modified, removed or added to the design to accommodate enhancements or new technology without significantly affecting the remainder of the application.

Since each module is responsible for a specific task and redundant modules may be implemented, the system is extensible, scalable, adaptive, and fault tolerant; characteristics that are highly desirable if not necessary in a real-world system. The modular design also allows one to adopt a divide-and-conquer strategy which simplifies the development of complex systems as well as the solution to complex problems. Modules will be implemented in MATLAB and/or C++ depending upon the specific task and its environment.

MATLAB was chosen because it has a large user community spread across industry, government, and academia, and is recognized as the standard for technical computing. The MATLAB environment combines mathematical computing and built-in interfaces while allowing integration of external routines written in C, C++, Fortran, and Java. Finally, MATLAB applications can be entirely converted to C and C++ applications using the MATLAB compiler and therefore can be used on a wide variety of platforms. These reasons, among others, make MATLAB an ideal choice for this and other technical problems (MATLAB 6.5.1, n.d.).

- *Multi-Robot Control System:*

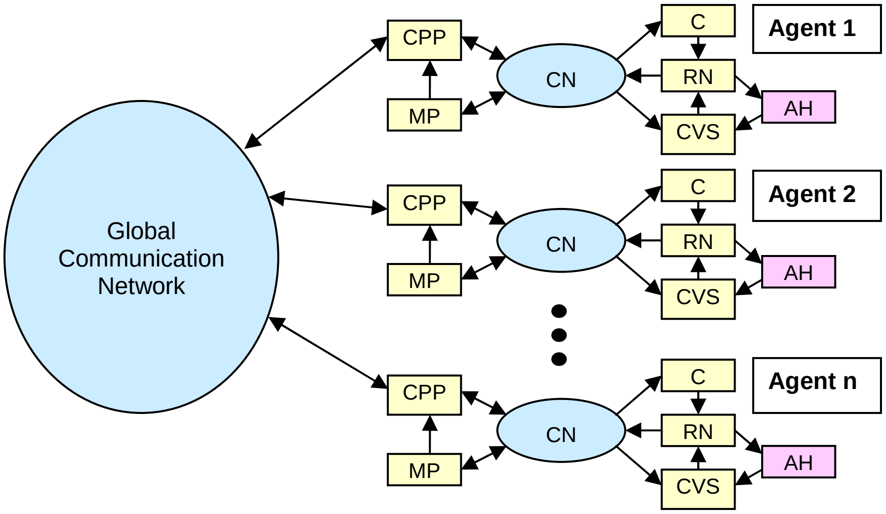

Figure 3. Global System Architecture for Distributed Hierarchical Multi-Robot Control

When the mission requires the use of multiple robots, the system architecture depicted in Figure 3 is utilized. This architecture augments the hybrid hierarchical distributed architecture of the single robot control system by allowing individual members to communicate with each other through their CPP module thereby enhancing the team's ability while maintaining a distributed nature. Since each unit retains its autonomy, if communication is disrupted or a member is lost, the remainder of the team will continue thereby alleviating a common point of failure. This high-level architecture extends the concepts developed at the single robot level into a global distributed system that is also redundant, extensible, scalable, adaptive, and fault tolerant.

- *Measurements:*

To establish a consistent environment for comparison, both the global-max and the local-max policies will be implemented in addition to developing the dynamic policy. Time in steps to completion of goal will be measured for each methodology and compared against established results.

E. Contributions

Robotic platforms provide a means to minimize human exposure to hazardous conditions. This research extends current probabilistic multi-agent systems by examining a dynamic search policy which is expected to demonstrate higher performance than established local search policies while maintaining greater computational feasibility than global search policies. Furthermore, robotic systems such as the one considered in this research have applications virtually anywhere human lives are put at risk. Potential areas that can benefit from robotic platforms include but are not limited to space, land, and sea exploration, hazardous environment assessment, urban warfare, search and rescue, and terrain mapping applications.

In addition to robotic applications in general, the problem considered in this research is applicable to several other disciplines as well, including mapping and exploration, distributed computing, computer vision, communications, networking, multi-agent coordination, electronic sensing, navigation control and regulation. Techniques used in this research can even cross disciplines into modeling biological systems such as human/insect vision (Webb & Harrison, 2000) and the immune system (Thayer & Singh, 2002).

F. References

Asada, M., Uchibe, E., & Hosoda, K. (1999). Cooperative behavior acquisition for mobile robots in dynamically changing real worlds via vision-based reinforcement learning and development. *Artificial Intelligence, 110,* 275-292.

Deng, X., Kameda, T., & Papadimitriou, C. (1998, March). How to learn an unknown environment I: The rectilinear case. *Journal of the ACM, 45, 2,* 215-245.

Desai, J., Kumar, V., & Ostrowski, J. (1999). Control of changes in formation for a team of mobile robots. *Proc. of IEEE Conf. on Robotics and Automation,* 1556-1561.

Han, K., & Veloso, M. (1998). Perception, reasoning and learning of multiple agent systems for robot soccer. *Proc. of IEEE Conf. on Robotics and Automation*, *4*, 3510-3515.

Hespanha, J., Kim, J., & Sastry, S. (1999). Multiple-agent probabilistic pursuit-evasion games. *Proc. of 38th IEEE Conf. on Decision and Control*, *3*, 2432-2437.

Hespanha, J., Prandini, M., & Sastry, S. (2000). Probabilistic pursuit-evasion games: a one-step Nash approach. *Proc. of 39th IEEE Conf. on Decision and Control*, 2272-2277.

Kitano, H., Asada, M., Noda, I., & Matsubara, H. (1998). Robocup: Robot world cup. *IEEE Robotics and Automation Magazine, 5, 3,* 30-36.

MATLAB 6.5.1. (n.d.). Retrieved December 31, 2003, from http://www.mathworks.com/products/matlab/description1.jsp

Roumeliotis, S., & Bekey, G. (2000). Collective localization: a distributed Kalman Filter approach to localization of groups of mobile robots. *Proc. of IEEE ICRA*, 2958-2965.

Sastry, S., Meyer, G., Tomlin, C., Lygeros, J., Godbole, D., & Pappas, G. (1995). Hybrid Control in Air Traffic Management Systems. *IEEE Conference on Decision and Control,* 1478-1483.

Stone, P., & Veloso, M. (2000). Multiagent systems: A survey from a machine learning perspective. *Autonomous Robots, 8, 3,* 345-383.

Thayer, S.M., & Singh, S.P.N. (2002). Development of an immunology-based multi-robot coordination algorithm for exploration and mapping domains. *IEEE/RSJ International Conference on Intelligent Robots and Systems*, *3,* 2735-2739.

Thrun, S., Burgard, W., & Fox, D. (1998). A probabilistic approach to concurrent mapping and localization for mobile robots. *Machine Learning & Autonomous Robots, 31, 5,* 1-25.

Timofeev, A., Kolushev, F., & Bogdanov, A. (1999). Hybrid algorithms of multi-agent control of mobile robots. *Int. Joint Conference on Neural Networks*, *6,* 4115-4118.

Vidal, R., Rashid, S., Sharp, C., Shakernia, O., Kim, J., & Sastry, S. (2001). Pursuit-evasion games with unmanned ground and aerial vehicles. *Proc. of IEEE ICRA, 3,* 2948-2955.

Vidal, R., & Sastry, S. (2002). Vision based detection of autonomous vehicles for pursuit-evasion games. *Proc. of IFAC World Congress on Automatic Control*.

Vidal, R., Shakernia, O., Kim, H.J., Shim, D.H., & Sastry, S. (2002). Probabilistic pursuit-evasion games: theory, implementation, and experimental evaluation. *IEEE Transactions on Robotics and Automation, 18, 5,* 662-669.

Webb, B., & Harrison, R. (2000). Eyes and ears: combining sensory motor systems modeled on insect physiology. *Proceedings IEEE International Conference on Robotics and Automation, 4,* 3913 -3918.