Phase I Final Report

Topic #: STTR-016

Project Title: Perception-based Co-evolutionary Reinforcement Learning for UAV Sensor Allocation

Contract #: N00014-02-M-0265

Company Name: Intelligent Inference Systems Corp

Point of Contact: Dr. Hamid R. Berenji

Address: 333 W. Maude Ave., Suite 105
Sunnyvale, CA 94085

Phone: (408) 730-8345

Fax (Optional): (408) 730-8550

E-mail (Optional): berenji@iiscorp.com

Web-site (Optional): iiscorp.com

Government Technical POC (Point of Contact):
Name: Dr. Allen Moshfegh

Activity: Office of Naval Research

Phone #: (703) 696-7954

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*	**2. REPORT DATE**	**3. DATES COVERED** *(From - To)*
02-00-2003	Final Report, Phase I	July 1, 2002 to Feb 3, 2003

4. TITLE AND SUBTITLE
Perception-based Co-evolutionary Reinforcement Learning for UAV Sensor Allocation

5a. CONTRACT NUMBER
N00014-02-M-0265

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)
Hamid R. Berenji, David Vengerov, Jayesh Ametha

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Intelligent Inference Systems Corp
333 W. Maude Ave., Suite 105
Sunnyvale, CA 94085

8. PERFORMING ORGANIZATION REPORT NUMBER
IIS-03-01

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Office of Naval Research
ATTN: Dr. Allen Moshfegh
800 Quincy Street
Arlington, VA 22217-5660

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSORING/MONITORING AGENCY REPORT NUMBER

12. DISTRIBUTION AVAILABILITY STATEMENT
Unclassified/Unlimmited

13. SUPPLEMENTARY NOTES

14. ABSTRACT
In this project, we have formulated the problem of sensor allocation in a team of UAVs within a mathematical programming framework. A Perception-based reasoning approach based on co-evolutionary reinforcement learning was developed for jointly addressing sensor allocation on each individual UAV and allocation of a team of UAVs in the geographical search space. An elaborate problem setup was simulated and experimented with, for testing and analysis of this framework using the Player-Stage multi-agent simulator. This simulator was developed jointly at the USC Robotics Research Lab and HRL Labs.The experimental results demonstrated a very strong performance of our methodology for UAV sensor allocation problem domains. Our results indicate that not only it is feasible to use perception-based reinforcement learning for this problem but it is an adequate solution for many typical UAV teams.

15. SUBJECT TERMS
Reinforcement Learning, Genetic Algorithms, Sensor Allocation, Unmanned Aerial Vehicles (UAVs), Neural Networks, Dynamic Programming

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON**
			Unclassified/Unlimmited	14	Dr. Hamid R. Berenji
a. REPORT UU	**b. ABSTRACT** UU	**c. THIS PAGE** UU			**19b. TELEPONE NUMBER (*Include area code*)** (408) 730-8345

Executive Summary

The current approaches for management of teams of UAVs are mostly inadequate due to their requirement of centralized control. The other approaches that attempt de-centralized and cooperative control mostly fail due to the curse of dimensionality of this problem. In order to meet this big challenge, a completely new and revolutionary approach is required. In phase I of this STTR project, we proposed to study the feasibility of using a distributed reinforcement learning approach for sensor allocation in a team of UAVs. A Perception-based reasoning framework based on reinforcement learning approach was developed for jointly addressing sensor allocation on each individual UAV and allocation of a team of UAVs in the geographical search space. An elaborate problem setup was simulated and experimented with, for testing and analysis of this framework using the Player-Stage multi-agent simulator. The experimental results demonstrated a very strong performance of our methodology for UAV sensor allocation problem domains. Our results indicate that not only it is feasible to use perception based reinforcement learning for this problem but it is an adequate solution for many typical UAV team problems.

Introduction

UAVs have recently gained a lot of attention as natural candidates for various applications where human intervention is considered difficult or dangerous. There are well-established feedback control techniques for stabilizing and controlling UAV motors for achieving the chosen motion objectives. However, the high-level issues of making strategic decisions about UAV motion and sensor management have only been addressed heuristically on a case-by-case basis. We have developed a general Perception-based Reinforcement Learning (PRL) framework for jointly addressing these issues in dynamic and unpredictable environments.

A mission by a team of UAVs requires sensor allocation at two levels: the individual UAV and the team-wide allocation. At the individual level, the total available resources (such as time or stored energy) of UAV's sensors needs to be dynamically allocated for tracking various targets. At the team level, each UAV can be treated as a composite sensor, and these sensors need to be allocated to different regions of the search space in proportion to the density and importance of targets there, in order to satisfy the team level goals.

The biggest challenge in applying any reinforcement learning algorithm to the UAV surveillance problem is in differentiating whether a successful mission is due to a good motion policy or a good sensor management policy. Instead of using a single policy for decision making that involves both sensor management and UAV allocation together, we proposed to solve this problem by jointly learning two different but complementary policies that work towards a common goal. By assigning the same reward function to both policies, the co-evolutionary process is guaranteed to converge, since both learning updates will take turns in improving the same objective.

In Phase I of this project we have been able to successfully achieve the following technical objectives:

- Determining the feasibility of combining Perception-based reasoning with reinforcement learning in modeling UAV problem scenarios
- Determining the feasibility of applying PRL for both individual sensor allocation and for team level UAV allocation, based on potential field method
- Evaluated the co-evolutionary adaptation of individual and team-level UAV sensor allocation policies

In this report, we first describe the details of the problem formulation followed by the methodology developed, experimental setup, and the simulations conducted in the course of this project. We then summarize the results, present our conclusions and plans for phase II.

Problem formulation

We have developed a mathematical programming framework, which allows finding sensor allocation policies that are optimal not only for individual UAVs but also for the multi-UAV team as a whole. The objective of each UAV is to choose actions a_t in consecutive time periods t = 0, 1, 2, … so as to maximize the expected value of the discounted sum of future rewards:

$$\max_{a_t, t=0,1,\ldots} E\{\sum_{t=0}^{\infty} g^t r_t(s_t, a_t)\}, \quad (1)$$

subject to the constraint on the sequence of states s_t:

$$s_{t+1} = f(s_t, a_t), \quad (2)$$

where γ is the discounting factor and $r_t(s_t, a_t)$ is the reward received at time t in state s_t after taking the action a_t.

We chose a very general reward function, which reflects simultaneously many of the problem complexities that we would like the team of UAVs to optimize. The reward received by the kth UAV for tracking all targets within its sensor range (e.g. its field of vision) after having aligned itself with target j is given by:

$$r_{kj} = \sum_{n=1}^{N} \left(\frac{V_n}{1+d_{kn}^2} \right) \left(\frac{\frac{1}{1+d_{kn}^2}}{\sum_{m=1}^{M} \frac{1}{1+d_{mn}^2}} \right), \quad (3)$$

where N is the total number of targets within its field of vision, once aligned with target j, M is the number of UAVs tracking target n, d_{kn} is the distance between UAV k and target n, and V_n is the value of target n. The above form of the reward function allows UAVs to learn the sensor allocation policy that tends to track targets that have higher values, that are closer to the UAV, targets that are bunched together, and that do not cause increased competition among UAVs for rewards.

The action a_t of each UAV needs to optimize simultaneously two aspects of the reward function. On the one hand, the UAV needs to be close to the high-valued targets in order to maximize the first term of the reward function, $\frac{V_n}{1+d_{kn}^2}$. On the other hand, the team-level optimality of the multi-UAV sensor allocation to multiple targets requires that UAVs do not duplicate each other's efforts by allocating all of their sensors to the single highest-valued target while leaving other targets unattended. The second component of the reward function, $\frac{1/(1+d_{kn}^2)}{\sum_{m=1}^{M} 1/(1+d_{mn}^2)}$, attempts to prevent such a behavior by rewarding each UAV more for tracking targets that do not have many other UAVs around them. Therefore, the action a_t of each UAV has two components: individual decision of

choosing the target with which to align the sensors and the team-level decision of where to position itself in the metric space inhabited by other UAVs. Because of the different physical nature of these components, different parts of the state s_t will be most relevant for making each decision.

The complete state s_t of each UAV can be described as a vector of dimension $2L+P$, where L is the total number of targets currently visible by the UAV and P is the total number of UAVs in the team. The first $2L$ components of the state vector come from the need of tracking for each observed target its distance and its estimated value. The remaining P components come from the need of tracking the distances to all other UAVs. Unfortunately, the approach of developing policies based on the complete state vector is clearly not scalable and will require an unreasonable amount of time to learn an optimal policy that is sensitive to all possible configurations of the large number of state variables. Moreover, as the number of targets present in the environment or the number of UAVs present in that locality changes, the dimension of the state vector will change and all UAV policies will need to be re-learned.

Therefore, in order to make UAVs learn policies that are both tractable and robust to changes in the number of targets or UAVs observed, we proposed an approach where each UAV observes only the most relevant parts of the complete state vector at each time slot t when making the individual sensor allocation decision and then making the team-level motion decision.

In order to extract the most relevant information from the high-dimensional state vector, we used the potential field approach for compactly encoding information about location of multiple objects. That is, since the presence of each object – target or UAV – is important only in its local neighborhood, we treated it as a potential charge, whose value decays with the squared distance from it. With this view, we found the following two variables to be sufficient for learning the direction of change in the individual sensor allocation component of the action a_t, which leads to maximization of the reward $r_t(s_t, a_t)$:

1. $s[1] = \sum_{n=1}^{N} \frac{V_n}{1+d_{kn}^2}$
2. $s[2] = \sum_{m=1}^{P} \frac{1}{1+d_{jm}^2}$,

where P is the total number of other UAVs and d_{jm} is the distance between target j and UAV m. The variable $s[1]$ represents the sum of potentials of all targets that it can expect to track if it aligns with the jth target, while $s[2]$ represents the sum of potentials of all other UAVs near the jth target. The individual level decision process of each UAV consists of sequentially computing the utilities of all targets based on the above two state variables and then aligning its sensors with the highest utility target.

Using the same potential field approach, we found the following two variables to be sufficient for learning the direction of change in the team-level motion component of the action a_t, which leads to maximization of the reward $r_t(s_t, a_t)$:

1. $x[1]$ = Target potential
2. $x[2]$ = UAV potential

The gradient of the "Target potential" determines for each UAV the direction of motion leading to the greatest concentration of targets. All else being equal, this should be the preferred direction of motion. The gradient of the "UAV potential" determines for each UAV the direction of motion leading to the greatest concentration of other UAVs. All else being equal, this should be the least preferred direction of motion, so as to cause least competition over available targets. The team-level motion strategy for each UAV will help it to tradeoff the target and the UAV potentials at future locations in various circumstances.

More specifically, a target j contributes the following amount to the target potential at location i in the world:

$$P_{ij} = \frac{V_j}{1 + d_{ij}^2}, \tag{4}$$

where V_j is the value of the jth target and d_{ij} is the distance between the target and the considered location. A similar formula holds for computing the UAV potential, except that the potential sources are other UAVs and $V_j = 1$ is the value assigned to each UAV. Different values may be assigned to different UAVs in case of a heterogeneous set of UAVs that have different capabilities. The variable x_1 is the sum of potential of all targets at the considered location for the UAV, and x_2 is the sum of potential of all other UAVs.

To complete the mathematical programming formulation of the multi-UAV sensor allocation policy, the state transition function f needs to be specified in equation (2). Due to the complexity of the considered problem, this function cannot be expressed analytically. However, it can be simulated by following the motion strategy of all UAVs as well as by simulating appearance and disappearance of targets in the search area. Fortunately, the simulation-based description of the state transition targets is sufficient for reinforcement learning algorithms to learn how to iteratively improve the actions a_t in order to maximize the reward function $r_t(s_t, a_t)$.

Solution Methodology

The decision variables used by UAVs are assigned to a number of categories depending on the level of granularity intended for it. For example, in the simplified case, we can have two categories, SMALL and LARGE as shown in Figure 1. Each state variable will be SMALL to a certain degree and LARGE to a certain degree, according to the value of these linguistic categories at each point in space. If only two categories are used, then the following rules will be used by each UAV for evaluating the utility of aligning its sensors with each of the targets:

IF (s_1 is SMALL) and (s_2 is SMALL) then (Q is Q_1)
IF (s_1 is SMALL) and (s_2 is LARGE) then (Q is Q_2)
IF (s_1 is LARGE) and (s_2 is SMALL) then (Q is Q_3)
IF (s_1 is LARGE) and (s_2 is LARGE) then (Q is Q_4),

with the values Q_1, ...,Q_4 tuned by the reinforcement learning algorithm presented in *FuzzIEEE 2003* paper in Appendix A. The final utility of each target is computed as

$$Q = \frac{\sum_{i=1}^{4} Q_i w_i(s_1, s_2)}{\sum_{i=1}^{4} w_i(s_1, s_2)}, \quad (5)$$

where $w_i(s_1,s_2)$ is computed as $w^i(s) = \prod_{j=1}^{K} m_{S_j^i}(s_j)$ and $m_{S_j^i}(s_j)$ is the degree of membership of the state variable s_j to the fuzzy category S_j^i. In our simplified example, S_1^i = SMALL and S_2^i = LARGE as shown in Figure 1.

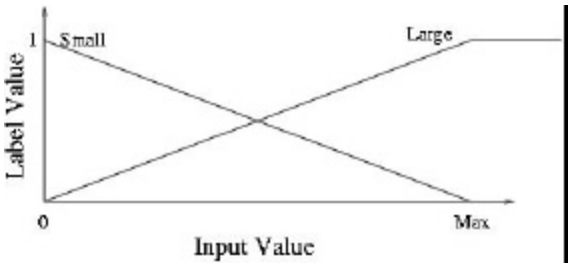

Figure 1. Generalized labels used by UAVs.

While preferable actions can be suggested by a remote human operator in simple cases, more complex scenarios requiring evaluating the tradeoff between distance, value, density of targets and vicinity of UAVs have to be learned by the UAV in the context of a particular mission.

To illustrate the approach for defining the team-level motion policy, assume that the motion state variables x_1 and x_2 are separated into two categories: SMALL and LARGE. The UAV selects one of 8 possible locations to move to, which are uniformly spaced around its current location. The preference of each location (its Q-value) is computed using the following perception rulebase:

IF (x_1 is SMALL) and (x_2 is SMALL) then (Q is Q_1)
IF (x_1 is SMALL) and (x_2 is LARGE) then (Q is Q_2)
IF (x_1 is LARGE) and (x_2 is SMALL) then (Q is Q_3)
IF (x_1 is LARGE) and (x_2 is LARGE) then (Q is Q_4)

with the values Q_1, ...,Q_4 once again tuned by reinforcement learning and the final utility computed according to equation (5).

The overall learning procedure of a UAV is as follows. First, the UAV selects its next location by choosing the one with the highest Q-value with probability p and choosing any other location at random with probability 1-p. After arriving at the new location, the UAV chooses its next target toward which its sensors should be pointed. The target with the highest Q-value (from the individual sensor allocation policy) is chosen with probability q and any other target is chosen at random with probability 1-q. Once the motion and rotation phases have been accomplished, the UAV computes its one-step reward according to equation (3) and uses it for updating the Q-values of both perception rulebases. After updating the Q-values, the UAV selects a new location and sensor alignment direction, and the cycle repeats.

The potential field motion planning strategy is fully distributed and robust to any changes in the environment. The decisions of each agent change gradually as the environment changes without the need for a complete ``re-planning" of classical planning strategies.

Note that the individual and team sensor allocation policies are interdependent. If one of the policies makes a suboptimal decision, it may adversely affect the common reward obtained, hence affecting the other policy even if the later had made an optimal decision for itself. For example, the sensor allocation policy may correctly choose a valuable target that is nearby. But if the UAV allocation policy chooses to move away from it, then the reward obtained by sensor allocation policy is less than it expected. In another case, the UAV may move correctly toward a location of high target potential, but if due to its sensor policy it fails to track those targets, the UAV allocation policy may receive a reduced reward for a good decision.

Since both UAV allocation policy and sensor allocation policy have the same common goals: tracking higher valued targets, closer targets, targets surrounded by other nearby targets, and tendency to reduce competition with other UAVs, we showed that they can tune themselves in a co-evolutionary manner. In more complex problems, the goals of the two policies may be only partially overlapping. For example, the sensor allocation policy may not consider presence of other UAVs while selecting targets to align with. In such a case, it will not be concerned with reducing competition with other UAVs for reward, unlike the UAV allocation policy, hence forming a more challenging learning problem.

Experimental Results

The software applications used in this work, are ‘Player’ and ‘Stage’ that were developed jointly at the USC Robotics Research Lab and HRL Labs and are freely available under the GNU General Public License from http://playerstage.sourceforge.net.

Player is a multi-threaded robot device server. It gives simple and complete control over the physical sensors and actuators on a mobile robot. When Player is running on a physical robot, a client control program connects to it via a standard TCP socket, and

communication is accomplished by the sending and receiving of some of a small set of simple messages. Player is designed to be language and platform independent. The client program can run on any machine that has network connectivity to your robot, and it can be written in any language that can open and control a TCP socket.

Player is also designed to support virtually any number of clients. In other words the robots can "see" through each other's eyes. Any client can connect to and read sensor data from (and even write motor commands to) any instance of Player on any robot. The Player interface is used verbatim by the 'Stage' multi-robot simulator. This means that our control program for the simulator can be used without any changes on the real robots.

Stage simulates a population of mobile robots, sensors and objects in a two-dimensional bitmapped environment. Objects can be placed arbitrarily in the environment, and can act as obstacles, or targets depending on our problem. Environments can be constructed by simply drawing it in any form such as Adobe PhotoShop software.

Stage is designed to support research into multi-agent autonomous systems, so it provides fairly simple, computationally cheap models of lots of devices rather than attempting to emulate any device with great fidelity. Stage provides populations of virtual devices for Player. One can write robot controllers (such as our fuzzy rule base) as 'clients' to the Player 'server'. Typically, clients cannot tell the difference between the real robot devices and their simulated Stage equivalents. Clients developed using Stage would work with little or no modification with the real robots and vice versa. Thus Stage allows rapid prototyping of controllers destined for real robots. Stage also allows experiments with realistic robot devices one might not have physically. Various sensors and actuators are provided in the simulation, including sonar, scanning laser range-finders, camera, GPS, among others.

A bounded environment with no physical obstacles was chosen for clarity of our results. We used a simple 2D square shaped environment of length 2 units with no physical obstacles. In our experimental setup, 3 UAVs move in this environment attempting to track 6 moving targets. Targets have different values represented by their color. Size of a UAV and targets is 0.05 units and 0.025 units respectively.

The targets use sonar sensors to detect UAVs around them. If a UAV comes closer than a pre-selected minimum threshold distance to any target, the target moves in the exact opposite direction from the UAV in order to avoid it. The UAVs can move over each other and over the targets in the 2D simulator, which makes our simulation more realistic for the UAV domain.

Each UAV has the following set of simulated sensors:

1. Sony EVID30 pan-tilt-zoom camera set to a range of 60 degrees, with ACTS -- a fast color segmentation program to identify color of targets coming in the camera's range
2. SICK LMS-200 laser rangefinder to measure distance to other targets or UAVs
3. GPS device to exactly locate its own position in the environment with respect to a fixed reference point.

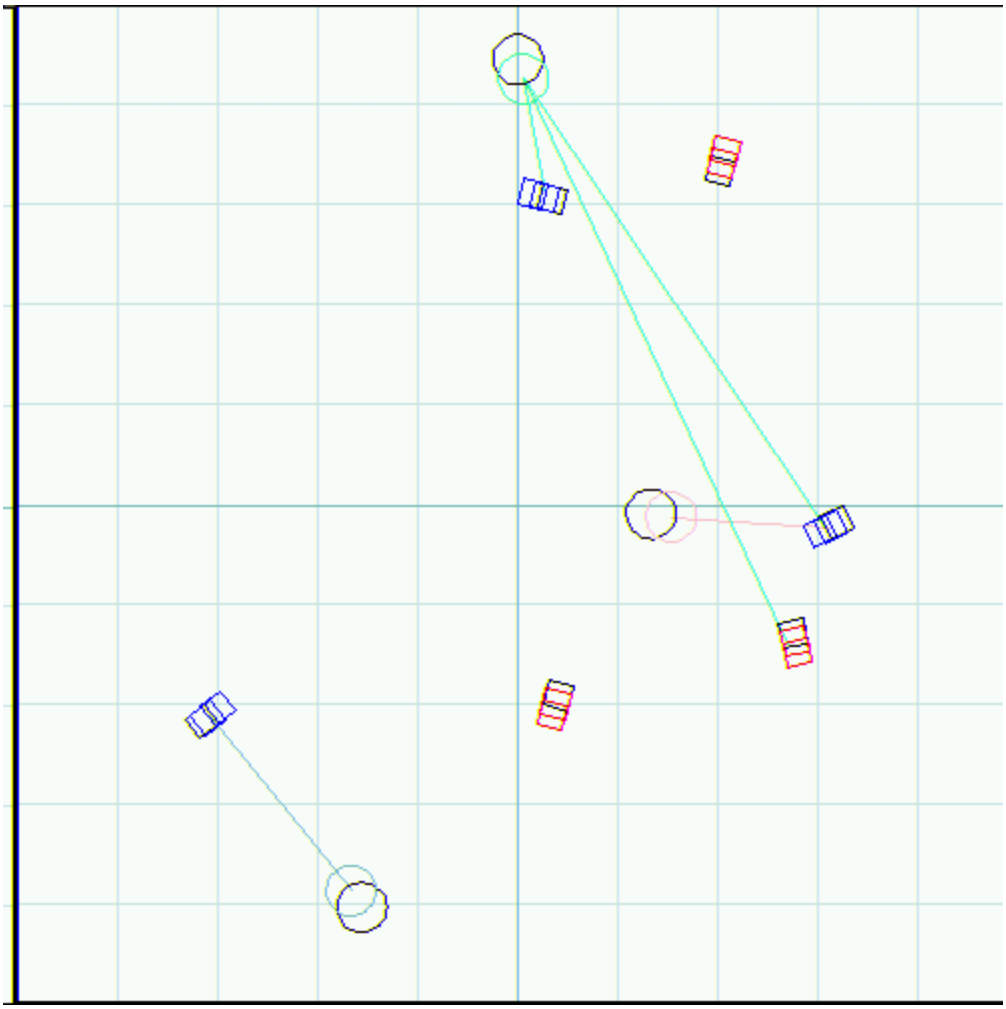

Figure 2. UAV team simulation on Player-Stage

Figure 2 depicts a still picture of the Player-Stage simulation setup. The circular objects simulate the UAVs. The square shaped smaller objects are the targets. Targets have different colors that represent their inherent values. In our simulation, the blue colored targets are three times more valuable than the red colored targets. The colored rays emerging from the UAVs represent which target(s) that UAV is presently tracking.

In the beginning of our simulation process, antecedent labels for the fuzzy rules used by each UAV had to be created. In order to accomplish this, a simple simulation is used where the UAVs and targets move in the environment, and data corresponding to the state variables is collected. The UAVs tend to avoid other UAVs, while targets tend to avoid UAVs and each other. Based on the range and distribution of data obtained, each state variable was categorized into 2 labels, LOW and HIGH. Since two state variables were used by the individual sensor allocation policy as well as by the UAV motion policy, 4 rules were created for each of them.

A sensible set of initial Q-values (q1, q2, q3 and q4) was assigned to each of the policies, and performance of the UAVs measured as the average reward was evaluated. The antecedent labels were then manually tuned to improve the average reward, while ensuring that all rules are triggered to similar cumulative levels for effective learning during the training phase. This tuning could have also been done using our co-evolutionary reinforcement learning algorithm at the expense of adding extra complexity to the problem.

The TD(λ) version of the discounted Q-learning was used for updating the consequent labels of the fuzzy rules (Q-values). Training runs had a fixed duration of 30 minutes, which translates into about 1500 steps for one UAV. The UAVs and targets were placed in the environment randomly at the start of each simulation. After every 60 seconds, they were re-randomized to ensure that all possible states have been visited adequately. Both co-evolutionary policies used the same one step reward function, as given by equation (1). 30 different training runs each of length (30 minutes or 5000 steps) were experimented with, to optimize the values of learning rate, discounting factor, and randomness decay rate, so as to obtain the best-learnt policies within the set framework.

All 3 UAVs used and updated the two policies individually and asynchronously. The learning proceeded from a completely uninformed situation where all Q values are set to 0. In the beginning, each UAV used high exploration probabilities *p* and *q*. However, over time both *p* and *q* approached 0, and each UAV tended to choose actions that have the highest Q value.

	l =**0**	l =**0.5**	l =**0.9**
Before learning	1.1	1.1	1.1
After learning	2.55	2.52	2.25

Table 1. Average reward of the final policy learned by UAVs.

Table I, shows the values of the average reward received by the UAVs during the testing phase for various values of the TD-parameter λ. The table also shows the values of the average reward for the initial policy that used Q-values equal to 0. The following values for the key parameters were used:

- Learning rate α for both policies: 0.75.
- Discounting factor γ: 0.9
- Randomness decay rate: 0.998

As table I shows, the UAVs significantly improved their performance as a result of learning with our co-evolutionary algorithm. The decrease in performance for higher values of λ is most likely caused by the fact that as time separation between actions and rewards increases, the connection between them decreases faster than in a single-policy reinforcement learning due to the presence of a second policy, which is also changing with time.

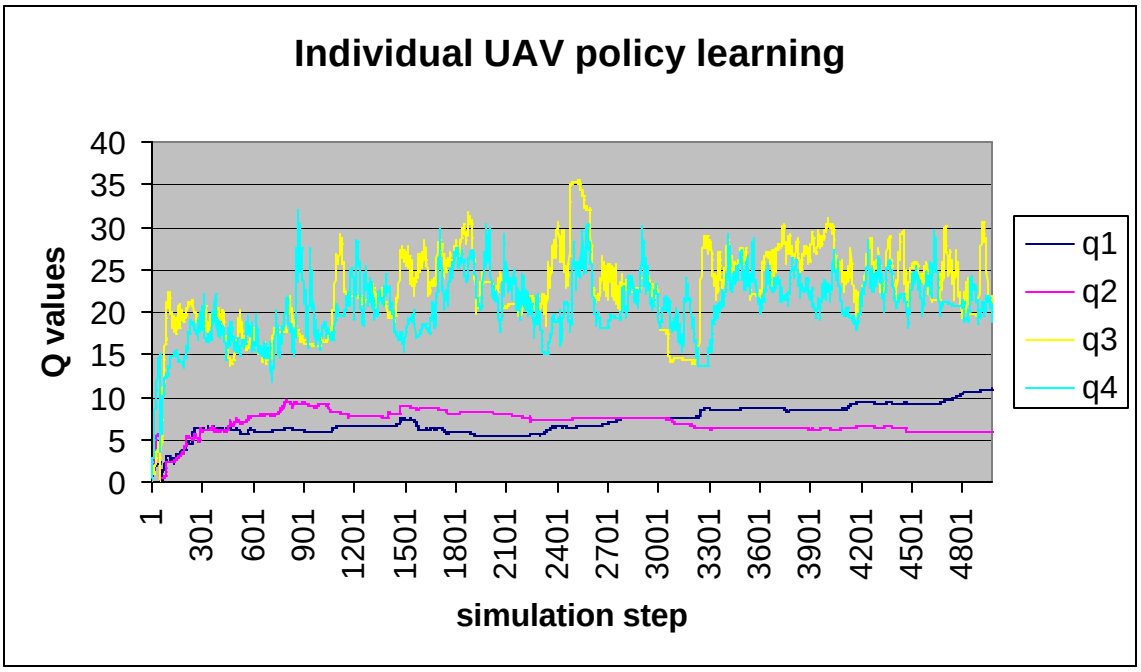

Figure 3. Plot of Q-value learning during training run for individual UAV sensor allocation

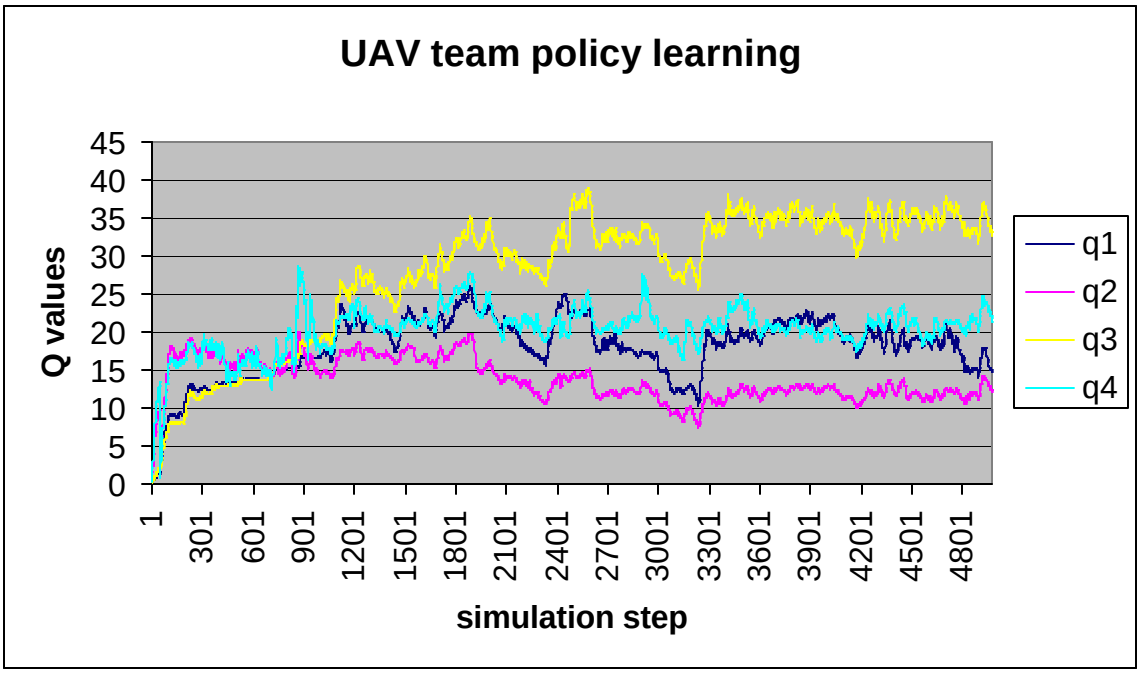

Figure 4. Plot of Q-value learning during training run for team-level UAV allocation

Summary of Phase I results

Our results indicate that not only it is feasible to use perception based reinforcement learning for this problem but it is an adequate solution for many typical UAV team problems. Using the Stage/Player Simulator, we tested the scenario with 3 UAVs and 6 moving targets. A more detailed hardware and software prototype demonstrations are planned for Phase II.

References

1. Phillip Chandler and Meir Pachter, "Hierarchical Control for Autonomous Teams," Proceedings of the 2001 AIAA Guidance, Navigation, and Control Conference, Montreal, Quebec, Canada, August, 2001.
2. Phillip Chandler, Meir Pachter, and Steven Rasmussen, "UAV Cooperative Control," Proceedings of the 2001 American Control Conference, Arlington, VA, June, 2001.
3. Phillip Chandler, Meir Pachter, Dharba Swaroop, Jeffery Fowler, Jason Howlett, Steven Rasmussen, Corey Schumacher, and Kendall Nygard, "Complexity in UAV Cooperative Control," Proceedings of the 2002 American Control Conference, Anchorage, Alaska, May, 2002.
4. J. Alexander Fax and Richard Murray, "Information Flow and Cooperative Control of Vehicle Formations," Proceedings of the 2002 IFAC World Congress, Barcelona, Spain, July, 2002.
5. Dimitris Hristu and Kristi Morgansen, "Limited communication control," *Systems and Control Letters*, Vol. 37, pp.193-205, 1999.
6. R. S. Sutton and A. G. Barto. *Reinforcement Learning: An Introduction*, MIT Press, 1998.
7. Christopher J. C. H. Watkins. *Learning from Delayed Rewards*. PhD thesis, King's College, Cambridge, UK, 1989.
8. D. Bertsekas and J. Tsitsiklis. *Neuro-Dynamic Programming*, Athena Scientific, 2000.
9. Kaixin Xu, Xiaoyan Hong, and Mario Gerla "Landmark Routing in Ad Hoc Networks with Mobile Backbones." *To appear in Journal of Parallel and Distributed Computing (JPDC), Special Issues on Ad Hoc Networks, 2002*
10. David A. Vengerov, Nicholas Bambos, Hamid R. Berenji. (2002) "Adaptive Learning Scheme for Power Control in Wireless Networks." In Proceedings of the 2002 Fall Vehicular Technology Conference (VTC).
11. David A. Vengerov, Hamid R. Berenji, Alexander B. Vengerov. (2002) "Emergent Coordination Among Fuzzy Reinforcement Learning Agents." A book chapter in *Soft Computing Agents: A New Perspective for Dynamic Information Systems*, International Series "Frontiers in Artificial Intelligence and Application" by IOS Press. Editor: V. Loia.
12. David A. Vengerov, Hamid R. Berenji, Alexander B. Vengerov (2002) Adaptive Coordination Among Fuzzy Reinforcement Learning Agents Performing Distributed Dynamic Load Balancing, In proceedings of the 11th IEEE International Conference on Fuzzy Systems (FUZZ-IEEE '02), pp. 179-184.
13. Hamid R. Berenji, David A. Vengerov. (2002) "A Convergent Actor Critic Based Fuzzy Reinforcement Learning Algorithm with Application to Power Management of Wireless Transmitters." Accepted for publication in the *IEEE Transactions on Fuzzy Systems* on October 12, 2002.

Appendix A

Perception-based Reinforcement Learning for Sensor Allocation in Unmanned Aerial Vehicles, to appear in the proceedings of the 20003 IEEE conference on Fuzzy Systems, May 2003.

Acrobat Document