Building Adaptive Computer Generated Forces: The Effect of Increasing Task Reactivity on Human and Machine Control Abilities

Magdalena D. Bugajska
Alan C. Schultz
J. Gregory Trafton
Shaun Gittens
Naval Research Lab
Washington, DC 20375
{magda, schultz, trafton, gittens}@aic.nrl.navy.mil

Farilee Mintz
ITT Industries, AES Division
Alexandria, VA 22303
mintz@aic.nrl.navy.mil

Abstract

Computer Generated Forces (CGF), in order to be effective training tools, must exhibit robust, challenging, as well as realistic behaviors. CGF tasks usually have both cognitive and reactive aspects to them. The reactivity has to co-exist with the "higher-level" cognitive activities like planning and strategy assessment, in the system that interacts with the environment. The overall purpose of our research is to merge a machine-learning algorithm (SAMUEL, an evolutionary algorithm-based rule learning system) with a cognitive model (ACT-R) into a system where the learning algorithm handles the reactive aspects of the task and provides an adaptation mechanism, and where the behavior's realism is constrained by the cognitive model. Such a system would learn through experience so that it can adapt to changes in adversaries' strategies and capabilities, to present human opponents with more exciting, varied, yet realistic training situations. This preliminary work presents an initial examination of effect of the changes in task reactivity on the human and SAMUEL control abilities.

1 INTRODUCTION

To be effective training tools, Computer Generated Forces (CGF) must exhibit cognitively plausible behaviors. In addition, they should not appear to be overly predictable and instead should exhibit adaptability in their behavior. This adaptability, of course, must not go beyond the bounds of realism.

The overall purpose of our research is to merge SAMUEL (an evolutionary algorithm-based rule learning system) (Grefenstette, et al. 1990) with a cognitive model in a system where the learning algorithm handles reactive aspects of the task and provides an adaptation mechanism, and where the behavior's realism is constrained by the cognitive model. Such a system would learn through experience so that it can adapt to changes in adversaries' strategies and capabilities, to present human opponents with more exciting, varied, yet realistic training situations.

This study looked at the effect that increasing the reactivity of the task has on human and a machine controller's performance. Our premise is that people are more sensitive to additional reactivity constraints than the machine learning algorithms. This paper presents the task and domain details as well as other experimental details of this study. The results supporting our premise are presented as well.

2 BEHAVIOR REPRESENTATIONS: LOW-LEVEL REACTIVITY AND HIGH-LEVEL COGNITION

How should Computer Generated Forces (CGF) be controlled? Many aspects of CGF tasks have highly reactive aspects to them (e.g., observing and responding to multiple simultaneous information sources while piloting an airplane). In addition, reactivity can be a critical aspect of performance when there are many individuals agents being controlled. This reactivity, however, must be combined with "higher-level" cognitive activities like planning and strategy assessment. Additionally, reactivity and planning activities must coexist in a single system that interacts realistically with the environment. In this paper, we explore how a reactive system and how people deal with different levels of reactivity. In a later part of this project, we will explore how a cognitive architecture (ACT-R) is able to deal with

both reactivity and higher-level cognitive aspects of a task.

Other cognitive models support reactive or perceptual/motor components. Soar can support reactive models (Laird, et al. 2000), (Nielsen, et al. 2000); however, a fixed decision cycle is not guaranteed. In Samuel, a defined, fixed decision cycle time is guaranteed, and a decision will be given each decision step. ACT-R/PM (Byrne, et al. 1998) adds a perceptual motor component to ACT-R (Anderson, et al. 1998). However, it does not give us the separation of components that allows for measuring the contributions of the reactive component. ACT-R/PM is an integrated cognitive architecture that allows low-level perceptual and motor activities to be used and controlled by full-scale productions. ACT-R/PM has an excellent integrated approach, but because we are specifically interested in reactive behavior, we have decided to explore the reactive and high-level cognitive aspects in different ways.

Our reactive system, Samuel, learns stimulus-response (S-R) rules that implement behaviors (Grefenstette, et al. 1990). The Samuel system's S-R rule representation is derived from behaviorist tradition. For example, Samuel's S-R rules do not use cognitive representation at all: there is no representation of goals, schema, memory structures, etc. The condition side of Samuel's rules match to the environment (or sensors), and the action side of Samuel's rules attempt to change the state of the world through actions. Samuel's strength lies in its ability to learn relatively simple condition-action rules to solve complex tasks using evolutionary algorithms and other learning methods. In addition, Samuel allows for parallel execution of sets of these S-R rules, thereby making possible the implementation of different behaviors.

Evolutionary algorithm-based reinforcement learning systems (Moriarty, et al. 1999) like Samuel are good at learning reactive strategies for sequential decision problems, but cannot take advantage of the higher level information that facilitates cognition, while cognitive models allow good representations of high level planning tasks, but are not typically as good at reactive skill learning. Our hypothesis is that an integration of these two approaches will create a system that combines the best of both reactivity and high-level cognition (e.g. planning), with learning at both the reactive level and at the cognitive level.

The evolutionary computation (EC) community has been interested in comparing evolutionary algorithms' performance to human performance (Koza, et al. 2000), (Luke, et al. 1998). In these studies, when only the reactivity of the task is increased, the EC-based SAMUEL algorithm significantly outperforms human performance in these tasks, as reported in this paper. Although we expect the humans to outperform the learning algorithm when the high-level cognitive aspects of the task are increased, in the final hybrid system, we predict the overall performance will be similar to a human in realism of behavior (a requirement of these types of systems), while exhibiting adaptability and robustness, which is required for capable computer-generated forces.

3 EXPERIMENTAL DOMAIN

To investigate these issues we have created a distributed Micro Air Vehicle (MAV) task. In the MAV task, groups of MAVs cooperate to perform reconnaissance. In this research, we assume each vehicle can detect certain ground features below the vehicle, and can detect obstacles, including other MAVs, within a defined range. As a group, the MAVs need to maximize the information gain, concentrating on areas of more importance, and minimizing duplication of effort. In previous work, we successfully used genetic algorithms (GAs) to evolve MAV control rule sets that could accomplish the above surveillance task (Wu, et al. 1999).

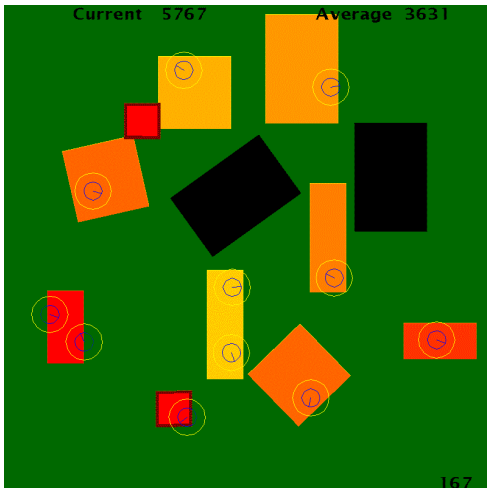

Figure 1 Snapshot of MAVSIM interface showing the God's Eye View of the environment. Regions of interest (yellow to red rectangles), obstacles (black obstacles) and MAVs (circles) are shown.

This domain allows us to vary both, cognitive and reactive aspects of the task independent of each other. The reactivity of the task can be increased by increasing the MAV group size as well as increasing the number of obstacles in the environment. The reactivity of the task is also dependent on the control mode used to control the MAV agents. In the current implementation, there are couple different control modes: Go-Direction and Move-To-Goal. The more reactive Go-Direction control mode allows the user to specify a direction for the agent to move in; the agent proceeds in the specified direction until a new command is issued. The Move-To-Goal control mode makes the task less reactive by allowing the controller to specify a goal location to which MAV is to proceed; the agent moves in straight line to the location and hovers over the goal until a new command is issued.

In order to investigate the sensitivity of the human and machine controllers to changes in the reactivity we manipulated the following aspects of the environment:

- The MAV group size.
- Number of obstacles.
- Human control modes

Future experiments will investigate more sophisticated aspects of reactivity including: the speed and maneuverability of the MAVs, and the speed and number of dynamic objects on the ground as well as cognitive aspects of the task including introducing spatial and temporal patterns to the environment.

3.1 SIMULATION

The Micro Air Vehicle Simulator (MAVSIM) includes a simple 2D model of the MAV's motion, sensors, and the environment. The motion model allows for calculating the agent's position at any time step given translation and turning rates and collisions between agents themselves as well as environment obstacles. The sensors currently modeled include a range sensor, whose output is a floating-point value representing distance to the nearest obstacle or fellow MAV, and a "vision" sensor, which provides the information about the ground features beneath the vehicle. The vision sensor determines the interest level of features within this area, and returns both the value of the highest interest area within the sensor area and a direction to that highest valued area. The MAV's environment consists of static as well as dynamic regions of varied military interest which model real world features such as roads, buildings, ground vehicles, etc. MAVs could be permanently destroyed in two different ways: they could leave the flight zone (i.e., fly off the screen), or they could collide with fellow MAVs or obstacles.

For this study, the environment whose example is shown in Figure 1, contained both, static and dynamic ground features with interest values between 0 (no interest) and 10 (highly interesting). It also contained between two and three obstacles. A number of predefined features and the MAVs were randomly positioned throughout in the environment at the beginning of each experimental trial.

The goal of the task was to maximize the score, which was determined as follows. Each MAV's instantaneous value was equal to the sensed area weighted by the interest of the visible regions within the sensor. If the sensor only partially covered an area on interest, it would a lower value than if it sat completely over the area of interest. The average score is the total value of all sensors averaged over time. Note that an area of interest could not be accumulated by more than one MAV in the same instant of time, i.e. only one MAV could get credit if two or more sensors overlapped on some portion of an area of interest.

4 EXPERIMENTS

4.1 HUMAN CONTROLLER

Ten researchers from the U.S. Naval Research Laboratory (NRL) served as participants in this study. Their education ranged from college graduate to Ph.D. Participants were given a short description of the task (Section 3), the general makeup of the environments, and were instructed on how to control the MAVs. They then practiced on a training session that lasted from 5-10 minutes. Following the training session, the participants went through six simulation sessions lasting five minutes each during which data was collected.

The human participants could judge the level of interest in the objects by their color, which ranged from a pale yellow to a bright red. The amount of red indicated the level of interest, with bright red areas mapping to an interest level of 10. The world began with no objects being visible to the participants. As a MAV flew around, the world underneath it became visible. Thus, a MAV flying over something like a building would see the object appear underneath it.

The human controlled MAVs by mouse manipulation. In the Move-To-Goal control mode, in order to move a MAV to a particular location, the participant clicked on the MAV, and then dragged it to the desired location. When the MAV arrived at the location, it hovered over that area. In the Go-Direction control mode, the user also used the same manipulation, but the MAV would continue in the direction defined by the mouse gesture until it left the flight zone at which time it could no longer be controlled.

The human participants, in addition to the average score, were also given a metric of the instantaneous total of all sensors. They could use this to make decisions about their current positioning of the MAVs.

4.2 SAMUEL CONTROLLER

SAMUEL is a machine learning system that uses evolutionary algorithms (GAs), reinforcement learning, and Lamarckian learning to solve sequential decision problems. The Lamarckian operators (e.g. specialization and generalization) modify decision rules based on observed interaction with the task environment. SAMUEL is designed for problems in which feedback is delayed (payoff occurs only at the end of an episode that spans many decision steps). This learning system has been previously used to learn behaviors such as navigation and collision avoidance for an autonomous underwater vehicle (Schultz 1991), shepherding (Schultz, et al. 1996), and tracking and herding for mobile robots. The original system implementation is described in detail in (Grefenstette, et al. 1990).

SAMUEL implements behaviors as a collection of stimulus-response rules. Each stimulus-response rule consists of conditions that match against the current

sensors of the autonomous vehicle, and an action that defines action to be performed by it. An example of a rule might be:

```
RULE 4
IF   range2 > 25
     AND  range5 > 0
     AND  camera1_interest > 1
     THEN SET turn = 45
```

This rule should be interpreted as follows: if the MAV's range sensor 2 is returning a value greater than 25 units, the range sensor 5 is sensing something, and the MAV is over a region of interest, the MAV should turn 45 degrees. Each rule has an associated strength with it as well as number of other statistics. During each decision cycle, all the rules that match the current state are identified. Conflicts are resolved in favor of rules with higher strength. Rule strength is updated based on the reward received after each training episode.

The MAVSIM as described above was used to model the MAVs, their sensors, and the environment. Each MAV (radius of 15.0 units) was equipped with a "vision" sensor, which returned the highest interest value within sensing range (0.0 – 30.0 in 5.0 units increments) as well as the bearing (angle relative to heading between –180.0 and 180.0 degrees in 10-degree increments) to the biggest visible area of that interest. Each agent was also equipped with eight range sensors with a 45-degree beam width and range between 0.0 and 50.0 units in 5.0-unit increments. Agents moved with a constant speed of 5.0 units per decision cycle. In order to control the MAV, the turn rate value between –180.0 and 180.0 degrees in 45-degree increments is specified for each decision cycle. The number of MAV agents and their configurations were held constant throughout the experiments. All the MAVs were controlled using the same behavior, which was currently being evaluated by SAMUEL.

For each simulation run (an episode), 1-3 predefined regions and 2-6 MAVs were randomly placed in the environment. The environment size varied between 0.5 and 0.3 of the size of the environment used by the human controllers.

Each learning evaluation consisted of a maximum of 150 decision cycles at the end of which the behavior was evaluated. The fitness function used in this study is defined as a weight based on the region's interest, sum of the regions surveyed by the group of MAVs over time. The value is normalized as a percentage of maximum possible payoff which is calculated as the weighted sum of the highest interest areas equal to the total area covered by the MAVs' sensor. SAMUEL's condition values included *range0* - *range7* representing MAV's range sensor readings, *camera1_interest*, which stored the highest interest value currently within sensing range of the "vision" sensor, and *camera1_direction*, which represented the bearing to the area of the highest value. In addition, SAMUEL was allowed to store some previous step information such as previously seen highest interest region (*prev_camera1_interest*), the bearing to the previously seen highest interest area (*prev_camera1_direction*), as well as the previous turning rate (*prev_turn_angle*). The *turn_rate* action attribute, which specified the MAV's turning angle per decision cycle, was the only action attribute in the system.

The learning experiment was allowed to run for 100 generations with a population of 100 rulebases. For each single evaluation 40 runs of the simulator were performed in order to provide the learning system with statistics about rulebase's performance for Lamarckian learning, rule strength updates, as well as the genetic algorithm. The system was initialized with 18 heterogeneous sets of rules, which implemented behaviors ranging from simple environment independent, reactive behaviors such as random walk and obstacle avoidance to behaviors, which incorporated domain knowledge.

5 RESULTS

In this section, the results of the study of the effect that increasing the task reactivity has on the human and SAMUEL controller's ability to perform the task, are discussed. The SAMUEL learning results are also presented.

5.1 LEARNING RESULTS

Every generation, the best ruleset (based on average performance measure) was evaluated 100 times in different, randomly generated environments. The values of these evaluations are plotted in Figure 2. As seen in this figure, the performance of the best behavior was approximately 54%, which shows a significant performance improvement from the initial behavior.

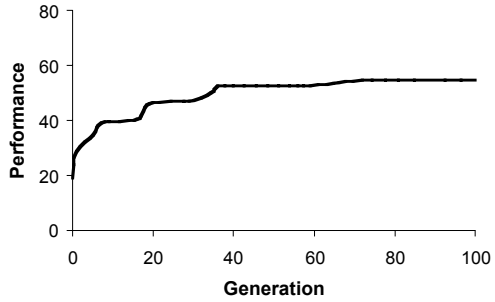

Figure 2 Average performance (over 100 trials) of the best individuals throughout generations tested in learning environment (solid black line).

After learning, the performance of the best ruleset was evaluated in the same environment as used by the human controllers.

5.2 HUMAN AND SAMUEL CONTROLLER PERFORMANCE RESULTS

The task performance measures used to evaluate the controllers included:

- MAV Survival Rate. The MAV survival rate was calculated as the number of MAVs controllable at the end of experimental trial. The higher survival rates signified a better performance.
- Total Score. The total score was calculated as average MAV group score over the length of the experimental trial (Section 3.1). The higher total score signified a better performance.

The controllers were evaluated using these metrics in varying levels of reactivity. For this study, the human controller performance was compared against SAMUEL performance of the task across the following reactivity conditions:

- MAV group size. There were to group sizes considered. The smaller MAV group (lower reactivity level) consisted of six agents while the larger MAV group consisted of sixteen agents (higher reactivity level).
- Human controller control mode. The human participants were able to control the MAV agent using either Go Direction control mode which increased the reactivity level of the task, but was essential the same as the control mode use by SAMUEL, or the Move To Goal control mode which decreased the reactivity of the task (Section 3).

Figures 3 and 4 summarize the results of the study for all the controllers across all the reactivity conditions.

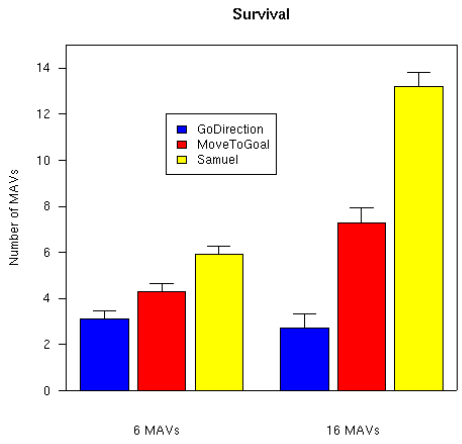

Figure 3 The MAV survival rate. The graph shows the performance as measured by the number of MAVs surviving at the end of the trial across different controllers (from the left: Human w/ Go-Direction, Human w/ Move-To-Goal, and SAMUEL) across different MAV group sizes (from the left: 6 and 16 MAVs). The error bars show standard error of means.

Figure 3 presents the performance as measured by the number of MAVs surviving at the end of the trial across different controllers using different MAV group sizes. Within the human controller experiments, two main effects can be seen. First, the number of MAV agents controllable at the end of the trial was proportional to the initial number of MAVs in the group. In addition, the number of surviving agents was greater when human controllers used the less reactive Move-To-Goal control mode then when the more reactive Go-Direction control mode was being used. More importantly, an interaction effect can be observed as well. Independent of the initial MAV group size, the human controllers using Go-Direction control mode (increased reactivity level) were able to control only three MAVs on average. When the human controllers' performance is compared to that of the SAMUEL controller, it can easily be seen that SAMUEL, which is essentially using the Go-Direction control mode, is able to control more agents and hence it easily outperformed the human controller in all conditions; SAMUEL even outperformed the human controller using the less reactive Move-To-Goal control mode.

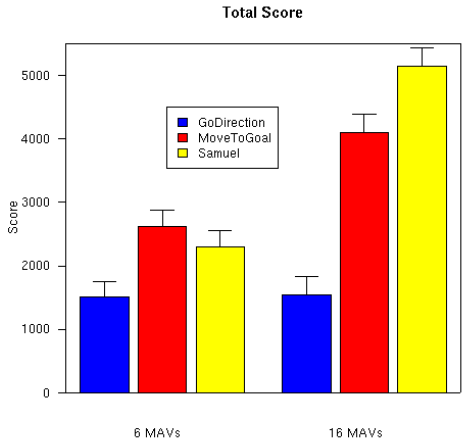

Figure 4 The total score. The graph shows the performance as measured by the total score across different controllers (from the left: Human w/ Go-Direction, Human w/ Move-To-Goal, and SAMUEL) across different MAV group sizes (from the left: 6 and 16 MAVs). The error bars show standard error of means.

Figure 4 presents the performance as measured by the total score across different controllers using different MAV group sizes. Within the human controller experiments, patterns similar to the MAV survival rates can be observed. The first main effect shows that the human controller can obtain a higher total score when controlling agents using the less reactive Move-To-Goal control mode. The second observed main effect shows

that a higher score is obtained when the initial group size is larger. The interaction effect can also be observed. Because the human controller using the Go-Direction control mode is able to control only three MAVs on average, the total score obtained under this condition is independent of the initial size of the MAV group. When the human controllers' performance is compared to that of the SAMUEL controller, it can be seen that SAMUEL again out performed or performed no worse than the human controller in all conditions; SAMUEL easily outperformed the human controller using the Move-To-Goal control mode when the number of agents was larger, and it performed on par with the human controllers when a smaller group of MAVs was controlled.

6 CONCLUSIONS AND FUTURE WORK

A study was presented that showed that while people are very sensitive to increases in reactivity, a genetic algorithm based system, SAMUEL, is not as sensitive. This tentatively suggests that a genetic algorithm should be able to assist or take over aspects of increasing reactivity in computer generated forces paradigms.

Even though SAMUEL handles reactivity very well, it is not expected to be able to handle higher-level cognitive aspects of the task such as planning and strategy assessment. Following study will focus on the effects that varying cognitive aspects of the task has on human and genetic algorithm controller's ability to perform the task. Finally, a cognitive model of human controller performance will be created and merged with SAMUEL into a hybrid system for realistic, robust, and more challenging Computer Generated Forces.

Acknowledgments

This work was supported by the Office of Naval Research.

References

J. R. Anderson and C. Lebiere (1998). The Atomic Components of Thought. Mahwah, NJ: Lawrence Erlbaum.

M. D. Byrne, & J. R. Anderson (1998). "Perception and action," in J. R. Anderson and C. Lebiere (Eds.) The Atomic Components of Thought, (pp. 167-200). Mahwah, NJ: Lawrence Erlbaum.

J. Grefenstette, C. L. Ramsey and A. C. Schultz (1990). "Learning sequential decision rules using simulation models and competition," Machine Learning, 5(4), 355-381, October 1990, Kluwer.

Koza, John R., Keane, Martin A., Yu, Jessen, Bennett, Forrest H III, and Mydlowec, William. 2000. "Automatic creation of human-competitive programs and controllers by means of genetic programming," Genetic Programming and Evolvable Machines. 1 (1 -2) 121 - 164.

J. Laird (2000). "An exploration into Computer Games and Computer Generated Forces," proceedings of the Ninth Conference on Computer Generated Forces. Orlando, FL.

Luke, S., C. Hohn, J. Farris, G. Jackson, and J. Hendler. 1998. Co-evolving Soccer Softbot Team Coordination with Genetic Programming. In RoboCup-97: Robot Soccer World Cup I (Lecture Notes in Artificial Intelligence No. 1395), H. Kitano, ed. Berlin: Springer-Verlag. 398-411.

D. E. Moriarty, A. C. Schultz, and J. J. Grefenstette (1999). "Evolutionary Algorithms for Reinforcement Learning," Journal of Artificial Intelligence Research, 11: 199 –229, 1999.

P. Nielsen, D. Smoot, R. Martinez, & JD Dennison (2000). "Participation of TacAIR-Soar in RoadRunner and Coyote Exercises at Air Force Research Lab, Mesa, AZ," proceedings of the Ninth Conference on Computer Generated Forces. Orlando, FL.

A. C. Schultz (1991). "Using a Genetic Algorithm to Learn Strategies for Collision Avoidance and Local Navigation," Proceedings of the Seventh International Symposium on Unmanned Untethered Submersible Technology, 213-225, University of New Hampshire, September 23-25, 1991.

A. C. Schultz, J. J. Grefenstette, and W. Adams (1996). "Robo-Shepherd: Learning Complex Robotic Behaviors," Proc. Of the International Symposium on Robotics and Automation, Robotics and Manufacturing: Recent Trends in Research and Applications, Volume 6, (Eds. Mohammad Jamshidi, Francois Pin, and Pierre Dauchez), ASME Press: New York, 763-768, May, 1996.

A. Wu, A. Schultz, & A. Agah (1999). "Evolving Control for Distributed Micro Air Vehicles," proc. Of the IEEE Conference on Computational Intelligence in Robotics and Automation, IEEE, 174-179, November 1999.