An investigation of how humans and machines deal with increases in reactivity

Alan C. Schultz
United States Naval Research Laboratory

Magdalena Dagmara Bugajska
United States Naval Research Laboratory

Farilee Mintz
Booz Allen Hamilton

An Investigation of How Humans and Machines Deal with Increases in Reactivity

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

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

Keywords:
reactivity, machine learning, cognition, evolutionary algorithms

ABSTRACT: *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). Also, reactivity can be a critical aspect of performance when there are many individual agents being controlled. This reactivity, however, must be combined with "higher-level" cognitive activities like planning and strategy assessment. Finally, reactivity and planning activities must coexist in a single system that interacts realistically with the environment. This preliminary work presents an initial examination of reactivity in SAMUEL agents and humans.*

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, much like a pilot would. This adaptability, of course, must not go beyond the bounds of realism.

The overall purpose of our research is to add learning and adaptation mechanisms to our CGF models. Our general approach is to combine reactive behavioral models with cognitive models. The cognitive models allow realistic behavior; the reactive behaviors allow us to adapt lower-level behaviors to achieve adaptability.

This paper, which reports our initial work, describes a pilot study which was designed to help determine the bounds and experimental setup for the rest of our research. These initial studies give interesting insights in several areas.

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). Also, 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.

Our reactive system, Samuel, uses stimulus-response (S-R) rules to implement behaviors [1]. 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 it's 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 [2] 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.

Why separate the cognitive from the reactive component? We wish to understand the interaction between the reactive and cognitive components. With two distinct models, we are able to more precisely measure the contribution of each to the total ability of the system. Also, learning at the reactive and cognitive levels may be quite different, and implementation of the learning system is simplified with separate models.

To investigate these issues we have created a distributed Micro Air Vehicle (MAVs) 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 [3].

This work presents an initial examination of reactivity in Samuel agents and humans. Our premise is that people will be sensitive to additional reactivity constraints, while Samuel agents will be less sensitive.

Our reactivity manipulation was extremely simple: The number of MAVs that needed to be controlled. 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.

We will first describe the basic experiment as well as a pilot test of human performance. We will show how human participants do seem to be sensitive to an increased level of reactivity.

3. Related Work

Other cognitive models support reactive or perceptual/motor components. Soar can support reactive models ([4], [5]); 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 [6] adds a perceptual motor component to ACT-R ([7]). 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.

4. Human Controller Experiments

4.1 Participants

Five researchers from the U.S. Naval Research Laboratory (NRL) served as participants in this pilot study. Their education ranged from college graduate to Ph.D.

4.2 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. 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 interest which model real world features such as roads, buildings, ground vehicles, etc., although in this study we only consider static features.

4.3 Test Environments

Ten different environments of varied complexity were created for this pilot study. The MAVs' flight zone for all the environments was restricted to an 800 x 800 unit area. Objects on the ground can be classified as to their level of "interest" with a value between 0 (no interest) and 10 (highly interesting). In the less complex environments, a set of five predefined regions of interest varying from 3 to 9 were randomly

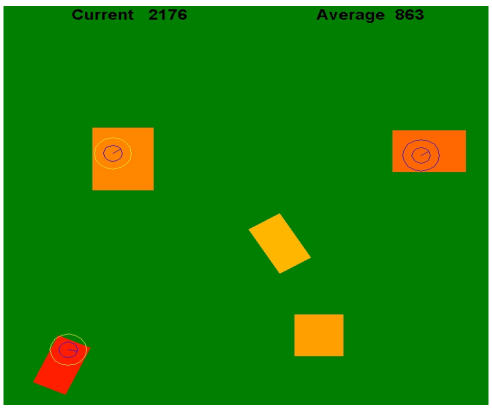

Figure 4.1 MAVSIM showing an environment of lower complexity. The rectangles are "buildings" and the circles are MAVs. The original display is in color.

positioned throughout the area to simulate the building structures in the flight zone. The more complex environments included the set of five regions described above as well as two additional regions of interest between 3 and 9, and a predefined complex shape of value 2, which simulated roads in the flight zone. Only the orientation and location of the regions changed, not their size or shape. At the beginning of each simulation run, all the MAVs were hovering on the left edge of the flight zone. Figures 4.1 and 4.2 show examples of simple and complex environments, respectively. 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.

4.4 Scoring an Episode

The participants were told to maximize their score, which was determined as follows. Each MAV's instantaneous value is equal to the sensed area weighted by the interest of the visible regions within

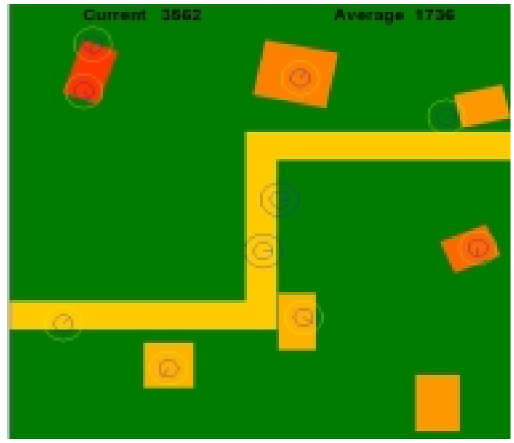

Figure 4.2 MAVSIM showing a more complex environment. The rectangles are "buidlings" and the lightest colored bar is a "road" (though nothing traveled on the road during these experiments) and the circles are MAVs. The original display is in color.

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. The 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.

4.5 Human Control of the Vehicles

MAVs were controlled by mouse manipulation. In order to move a MAV to a particular location, the participant left-clicked on the MAV, and then dragged it to the desired location. When the MAV arrived at the location, it hovered over that area. A MAV could also be directed by clicking on the rightmost mouse button. In this case, 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. MAVs could be permanently destroyed in two different ways: they could leave the flight zone (i.e., fly off the screen), or two or more MAVs could collide, destroying all MAVs involved in the collision. All MAVs moved continuously. When the simulator first started, all MAVs were set to orbit on the far left side of the screen.

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.

4.6 Design

Participants were tested on a sample of six environments chosen randomly for each participant from the set of all possible environments as described previously. We manipulated reactivity by increasing the number of MAVs the participant had to control. In the Low Reactivity condition, participants had to control three MAVs at once. In the High Reactivity condition, participants had to control ten MAVs at once.

4.7 Measures

We examined three total measures: the total score (as described above), the number of control strokes per MAV, and the average score of a single MAV. Total score will allow us to examine how participants performed overall. The number of commands per MAV was calculated as the total number of commands via mouse-clicks issued to each MAV divided by the number of MAVs. The average score of each MAV was calculated as the sum of the average scores of each MAV over time divided by the total number of MAVs. The latter two measures will allow us to measure reactivity.

We also collected protocol data [8] which will not be discussed in this report.

4.8 Procedure

Participants were given a short description of the task, 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.

4.9 Results

We first determined if the difference in complexity of the environments had any effect on the participants' scores. The complexity of the worlds did not seem to play a major role in the scores, $F(1,4) = 3.0$, $MSE=211053$, $p > .10$. For all later analyses, we will collapse across this variable. Also, participants did not crash many MAVs. Excluding participant 2 (the outlier), only 2 MAVs were lost throughout the session. Thus, participants seemed able to use and control their MAVs with reasonable success.

Next, we analyzed overall score and performance of each participant. As Figure 3 shows, when participants controlled more MAVs, they scored much better than when they controlled fewer MAVs, $F(1,4) = 24.2$, $MSE=3980359$, $p < .005$. This finding makes a great deal of sense: the more MAVs the user had, the greater the amount of interesting areas which could be monitored by MAVs increased, and thus the bigger the possible (and actual) score.

As Figure 4.3 shows, there is an obvious outlier in the pilot data. Since we will be examining within subject effects, we kept this participant in the dataset, though removing this outlier does not change the pattern or significance of any of the reported results.

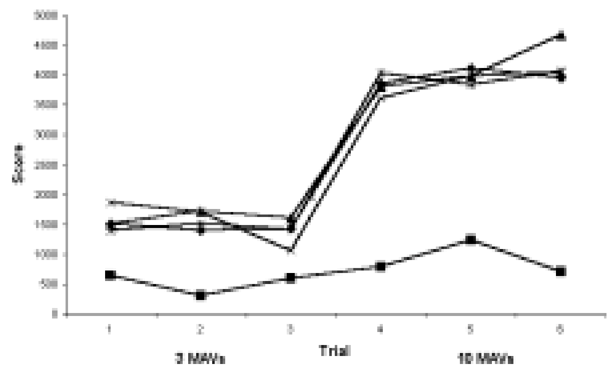

Figure 4.3 The scores for each participant across trials in this experiment. Each line is a different participant.

Next, we wanted to examine reactivity of the participants. Two obvious variables to examine were the number of commands issued to each MAV and each MAV's score. As described above, participants were able to control their MAVs and increase their total score with more MAVs. But how did the average of the MAV's scores change in a more reactive environment?

Participants in the Low Reactivity condition issued many more commands to the individual MAVs than they did in the High Reactivity condition. As Figure 4.4 suggests, this effect is robust, even with the small number of participants, $F(1,4)=44.5$, $MSE=36.9$, $p < .001$. Thus, when participants had to control more MAVs, they issued fewer commands to each MAV than when they needed to control fewer MAVs.

As Figure 4.5 shows, when participants were in the Low Reactivity condition, they had a higher average

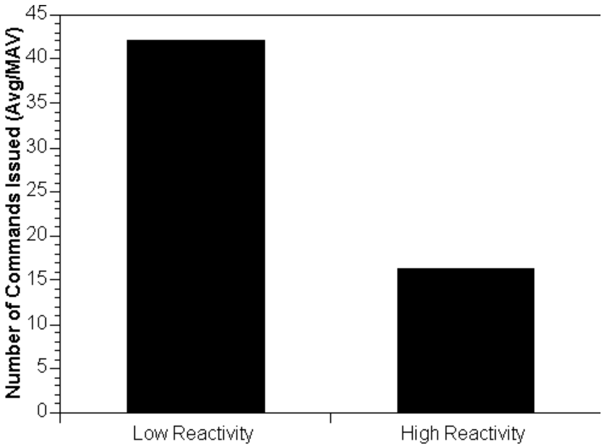

Figure 4.4 The average number of commands given to each MAV in the Low and High reactivity conditions.

score for each MAV than when they were in the High Reactivity condition, $F(1,4)=87.4$, $MSE=2404$, $p < .001$.

4.10 Discussion

Participants were able to obtain a higher score when they had access to more MAVs. However, more MAVs came at an increased reactivity cost: fewer commands given and a lower score for each MAV. There are, however, explanations other than an increase in reactivity to explain these findings. It

Figure 4.5 Average score per MAV for low and high reactivity conditions for human controllers.

could be, for example, the MAVs in the low reactivity condition had to explore more of the area, and this additional exploration required more commands. Also, the difference in the scores could be accounted for by assuming in the Low reactivity condition each MAV was able to "fit" on a building by itself, while in the High reactivity condition, MAVs had to either double up, (which reduced the score because only one MAV would get credit for a single feature at the same time), or be satisfied with a lower interest region. These issues will be explored in a later experiment.

5. SAMUEL Experiment

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 [9], shepherding [10], and tracking and herding for mobile robots. The original system implementation is described in detail in [1].

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.

5.1 Experimental Design

This section describes the methodology used for learning experiments performed to evolve a stimulus-response rule-based controller for the MAVs for the task of multi-agent large-area surveillance.

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 8 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), a constant number of predefined regions were randomly placed with a random orientation in the environment. The predefined features were only a subset of features used for implementing environments described earlier for human experiments, and included an 80x80 region of interest 4.0, a 100x60 region of interest 9.0, and a 50x270 region of interest 2.0. In addition, the size of the environment was reduced to 270.0 x 270.0 (about 1/3 of the original size). For these experiments, on the beginning of each trial, a group of four MAVs was placed in the same position and orientation on the left-most edge of the world. In order to confine the MAVs to the flight zone, a perimeter was placed around it. The perimeter was visible to the range sensor and permanently disabled the MAV, which crossed it.

Each learning evaluation consisted of a maximum of 150 decision cycles at the end of which the behavior was evaluated. If a MAV collided with an obstacle or a fellow MAV the episode terminated immediately. 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, which for the environments used during learning was equal to 3009. 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. 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 a set of rules, which implemented an environment independent random walk.

After learning, the best rule set was tested on the superset of the predefined environments, the human controllers used. For each of the ten possible environments, an average score and number of lost MAVs was obtained by running ten independent simulations during which SAMUEL controlled a group of MAVs. The performance was evaluated with groups of both three and ten MAVs giving us a total of 20 data points. The averages were also calculated for each of the reactivity conditions by averaging the average scores and MAVs collision statistics across the number of environments in each reactivity condition. Finally, the average score per MAV was then calculated by dividing the average score for each reactivity condition across all the environments by the number of MAVs in the group.

5.2 Results

Every generation, the best ruleset (based on average performance measure) was evaluated 100 times in different randomly generated environments. The values of these evaluation are plotted in Figure 5.1. As seen in this figure, the performance of the best behavior was about ~55% which shows a significant performance improvement from the initial behavior.

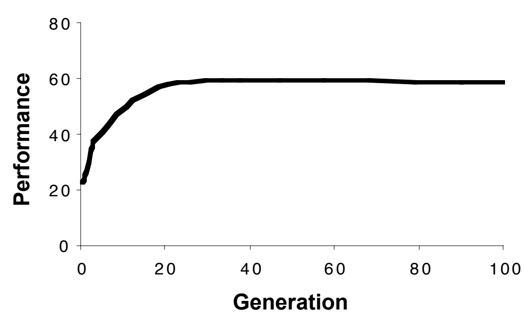

Figure 5.1 Average performance (over 100 trials) of the best individuals throughout generations tested in learning environment (solid black line).

The same metrics as in the experiments with human controllers were used to evaluate the SAMUEL's performance. The average number of commands per MAV, which for SAMUEL is defined by the number of decisions cycles, is independent of environment conditions and was held constant at 150. SAMUEL, as Figure 5.2 shows, scored higher when the number of

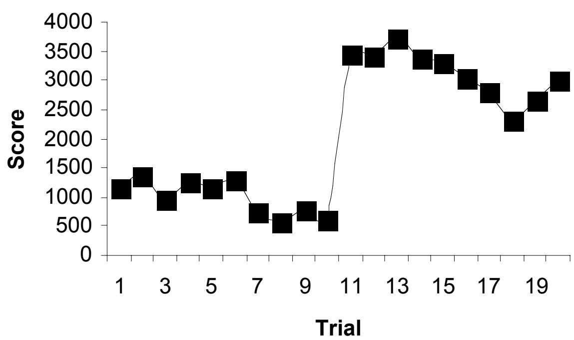

Figure 5.2 Average score (over 10 trials) for each environment for low (1-10) and high (11-20) reactivity levels.

controlled MAVs was higher. This result is consistent with how the human controllers performed. Thus, SAMUEL was able to score better when it had more MAVs to control, $F(1,18)=166.9$, $MSE=134116$, $p < .001$.

Recall that our hypothesis was that people would have problems with increased levels of reactivity and that SAMUEL would not. Interestingly, SAMUEL was able to deal with an increased level of reactivity in some ways, but had problems with more reactivity in others.

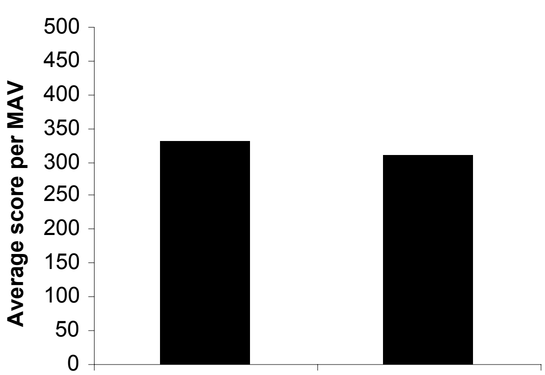

Figure 5.3 Average score per MAV for Low and High rectivity conditions.

SAMUEL seemed to have difficulty with an increase in reactivity, as shown by the increase in number of crashes that SAMUEL had. SAMUEL lost more MAVs in the High reactivity condition than in the Low reactivity condition (.42 vs. 0), $F(1,16)=14.2$, $MSE=.062$, $p < .005$. It should be noted, however, that while this difference is statistically significant, it is a very small effect.

Did SAMUEL show a difference in reactivity as determined by the individual scores of the MAVs? As Figure 5.3 shows, there was no difference in the average score of the MAVs, $F(1,16)< 1$, $MSE=3170$. This finding shows that SAMUEL is able to score the same amount on average with its MAVs, whether it is controlling only 3 MAVs or 10, even though it lost more MAVs due to crashes when it had to control 10 MAVs at once.

5.3 Discussion

Our hypothesis was that SAMUEL would deal better with increased levels of reactivity, while the human participants would pay a performance price with increased reactivity. The data collected for this study as presented in Sections 4.9 and 5.**3**, partially supports this hypothesis. In this section, the possible reasons for this outcome are discussed in this section.

The computational complexities of the MAVSIM as well as the internal characteristics of the SAMUEL forced us to design simpler and smaller learning environments, decrease the number of the MAVs in a group, and decrease the mission time by a factor of 8 as discussed in Section 5.2. This could have had adverse effects on the performance of evolved behaviors such as lower reactive abilities due to limited practice. Thus, when it was expected to maneuver with many more MAVs (the experimental environments), it had more crashes.

SAMUEL's performance of the task could have been also adversely affected by the MAVs' sensors (Section 5.2). The information given to SAMUEL was a fraction of information observable by the human controller. SAMUEL was given limited range information and even more limited information as to the interest and direction of the regions below the MAVs. It was not given any temporal information such as previously seen regions or any information about the MAVs' current states. All of that could have lead to a much harder problem than initially expected.

For the learning experiments discussed here SAMUEL's initial population was seeded with a ruleset of several rules, which implemented a random walk behavior. There are many different (although not necessarily better or worse) ways of initializing the population for this specific task. It is possible that a more domain specific initial behavior would have resulted in a better final behavior.

6. Conclusions

We have presented a pilot experiment that showed that people seem to be sensitive to increases in reactivity. We also showed that a genetic algorithm based system also was minimally sensitive to increases in reactivity.

The only way that SAMUEL showed sensitivity to an increase in reactivity was through a small increase in the number of crashes. This increase in number of MAV crashes was so small that it did not seem to affect the average score in the task. Further, the non-difference in average score casts doubt on one of the possibilities offered for the reactivity difference found in the human experiment. We suggested that one possibility for the different reactivity scores of the human participants was that the MAVs had to "double up" in the more reactive condition and not in the less reactive condition. Because SAMUEL did not show this difference, it suggests that SAMUEL was simply better at controlling the MAVs in a more reactive environment.

We should note that, in general, the human participant's score was rather better than SAMUEL's. We find this quite interesting and are exploring ways of increasing SAMUEL's behavior to increase its score.

In some ways, people's sensitivity to reactivity is surprising because we did not increase the reactivity by very much. In future experiments we plan on increasing the reactive aspects of this task much more.

These two experiments suggest that people are sensitive to differing levels of reactivity, while genetic algorithms are much less sensitive. This sensitivity in both cases needs to be explored more, but we can tentatively suggest that a genetic algorithm may be able to assist or take over aspects of increasing reactivity in computer generated forces paradigms.

7. Acknowledgments

This work was supported by the Office of Naval Research.

8. References

[1] 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.

[2] 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.

[3] 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.

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

[5] 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.

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

[7] J. R. Anderson and C. Lebiere (1998). <u>The Atomic Components of Thought.</u> Mahwah, NJ: Lawrence Erlbaum.

[8] K. A. Ericcson and H. A. Simon (1993). <u>Protocol analysis: Verbal reports as data.</u> Cambridge, MA: MIT Press.

[9] 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.

[10] 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.

Author Biographies

GREG TRAFTON is a cognitive scientist at the Naval Research Laboratory. His research interests are in cognitive modeling, the cognition of complex visualizations, and how people deal with interruptions. He received his B.S. from Trinity University in computer science/psychology and his MA and Ph.D in psychology from Princeton University.

ALAN C. SCHULTZ is the head of the Intelligent Systems Section of the Navy Center for Applied Research in Artificial Intelligence, part of the Naval Research Laboratory. His research interests are in genetic algorithms, robotics, machine learning, and human-robot interfaces. He received his BA in communications from the American University and a MS in computer science from George Mason University. He is a member of the IEEE, IEEE Computer Society, and the AAAI.

MAGDA BUGAJSKA is a computer scientist in the Intelligent Systems Section of the Navy Center for Applied Research in Artificial Intelligence. Her research interests are in evolutionary computation and robotics. She received her BS in computer science with minor in AI and Robotics from the Colorado School of Mines. She is a member of the AAAI.

SHAUN GITTENS is a PH.D. student at the University of Maryland currently working at Naval Research Laboratory.

FARILEE MINTZ is an ITT Industries AES Division researcher for the Naval Research Laboratory. She received her BA in psychology from The University of Michigan.