Applying Artificial Intelligence to Training Air Combat Maneuvering: The Potential, The Pitfalls, The Products

Richard A. Thurman Ph.D.
Air Force Armstrong Laboratory
Aircrew Training Research Division
Williams AFB AZ 85240

ABSTRACT

While nothing in the foreseeable future appears capable of replacing the requirement of actual aircraft flight in developing Air Combat Maneuvering (ACM) proficiency, the technology is now available to augment actual flight hours with meaningful training from computer-based simulations. At the Aircrew Training Research Division of the Armstrong Laboratory we have been developing several Artificial Intelligence (AI) based approaches to augmenting pilot training in simulation-based ACM. Because air-to-air combat is such a fast moving, complex task, automating (through AI) such tasks as performance measurement and assessment can provide a very important enhancement to a simulation. In addition, using AI techniques to create "smart bogeys" can provide a real boost in the training capabilities of a simulation. In this paper we describe our efforts to create three AI systems (two based upon rule-based production system technology and one based upon artificial neural systems technology) and detail their strengths and weaknesses in providing pilot training in ACM. Particular emphasis is given to the lessons we have learned in applying AI in this rich, fast moving, and complex task environment.

ABOUT THE AUTHOR

Richard Thurman is a research psychologist at the Air Force Armstrong Laboratory, Aircrew Training Research Division. After completion of undergraduate work in psychology, he received a a Ph.D. in Instructional Science from Brigham Young University. Previous to working for the Armstrong Lab, Dr. Thurman held R&D posts at Link Flight Simulation and at General Electric. His current research interests include instructional simulations, AI based training applications and virtual environment research and evaluation.

Applying Artificial Intelligence to Training Air Combat Maneuvering: The Potential, The Pitfalls, The Products

Richard A. Thurman Ph.D.
Air Force Armstrong Laboratory
Aircrew Training Research Division
Williams AFB AZ 85240

INTRODUCTION

Air combat training has a very high priority throughout an Air Force fighter pilot's career. Major Air Force time and expense is dedicated to ensuring that a fighter pilot is prepared to meet the challenges of air-to-air combat. All involved in air combat training realize that real air-to-air combat exposure can be acquired only during times of open conflict. During peacetime, Air Combat Maneuvering (ACM) is usually practiced through simulations of wartime conditions.

Two types of ACM simulation are available for pilots. The first type of simulation is aircraft based, and is represented by the Air Force Air Combat Maneuvering Instrumentation (ACMI). In ACMI training systems, instrumented aircraft are flown against each other while the ACMI records and displays aircraft engagements for later replay and debrief. Sensors continuously track and download to computers on the ground such variables as aircraft position, altitude, and attitude, as well as kill probabilities for each shot. Ground-based computers store the data for subsequent debriefing. The debrief is usually provided on ground-based computer graphics stations which show various views of the combat situation as well as some form of performance measurement.

The second type of simulation is the ground-based simulator. These typically take the form of an aircraft cockpit enclosed in a large domed display screen. Aircrews "fight" an adversary which is displayed on the surface of the dome. The adversary aircraft is "flown" either by an instructor at the simulator control console, another pilot in a networked simulator, or driven by a computer. System computers track variables such as aircraft positions, states, control activity and weapons releases during a scenario for subsequent debriefing.

Both aircraft and ground-based simulations require that some form of feedback to pilots (usually in the form of a debrief) take place. One can see that because air-to-air combat is such a fast moving, complex task, automated performance assessment and feedback can be a powerful addition to the simulator system. In fact for over a decade, automated performance measurement systems have been a required feature of many modern training simulators (Semple et al., 1981). However, there have been occasions where they were less than beneficial due to poor design or inconsistent application (Vreuls and Obermayer, 1985). Never-the-less a real obligation which training and performance technologists have, is the obligation to provide pilots with quick and appropriate assessments of their performance.

ACM PERFORMANCE ASSESSMENT IS NOT EASY

Trying to determine how well or how poorly a pilot is performing air-to-air combat is not an easy task. Kelly (1988) indicated some of the major challenges facing performance technologists as they attempt to come up with measures of ACM performance. He determined that three key factors make ACM performance measurement hard. These factors are; (a) no fixed profile, (b) ACM is highly reactive in nature, and (c) differing aircraft require differing ACM tactics.

No Profile

According to Kelly, most empirical aircrew performance measurement work has been conducted for instrument flight or other specific maneuvers where the desired flight profile can be established rather easily before hand. In these situations, performance can be easily described and criteria can be established through observing deviations from the desired profile. In ACM, because there is no such thing as a fixed profile. Performance criteria are ambiguous and constantly shifting.

ACM is Reactive

ACM is a highly dynamic and reactive activity. Kelly states:

> The dynamic relationship between the various aircraft in an ACM engagement is constantly changing as each pilot maneuvers in response to one or more other pilots. The resulting performance is a composite of the behavior of multiple pilots in multiple aircraft with multiple and often mutually exclusive objectives. It is impossible to obtain a pure measure of the performance of a single pilot in isolation form the others. (p. 497)

Different Tactics For Different Aircraft

Tactics are quite different for different pairings of aircraft. Subsequently pilot behavior will change as different aircraft are encountered. Often measures are taken which compare the maneuvering performance of one pilot's aircraft against the design capability of another's. However, even this technique fails to capture the essence of ACM such as "gamesmanship" and "intimidation."

OUR PROGRAM OBJECTIVES

At the Aircrew Training Research Division of the Armstrong Laboratory we have been developing several Artificial Intelligence (AI) based approaches to augmenting pilot training in simulation-based ACM. Because air-to-air combat is such a fast moving, complex task, automating (through AI) such tasks as performance measurement and assessment can provide a very important enhancement to a simulation. In addition, using AI techniques to create "smart bogeys" can provide a real boost in the training capabilities of a simulation.

This program grew out of a Program Research and Development Announcement (PRDA) which appeared in the Commerce Business Daily of 2 January 1988. The announcement stated "The purpose of this effort is to develop and validate an expert model of pilot decision making in air combat maneuvering (ACM). ... The system shall be capable of providing ACM decision-making training at the transition or continuation level ..." Basically, we were interested in answering three questions; (a) could expert system technology be applied when monitoring pilot performance, (b) could an AI based model of pilot decision making be created, and (c) could an AI based system provide pilots with diagnostic performance feedback. In addition we were interested in creating tools for future research in these three areas.

Several organizations submitted technical proposals for our review and evaluation. Based upon such criteria as technical expertise in the field, understanding of the problem, and soundness of the approach, several separate proposals were selected. Three are selected for discussion in this paper. These proposals were submitted by Ball Systems, Merit Technology, and Vreuls Research Corporation. Two of the projects used an expert system (rule based) approach (Vreuls & Merit), while the other used a neural network approach (Ball Systems).

Following is a review of each of the projects, the products emerging from them, and an assessment of the relative strengths and weaknesses of each product.

RULE-BASED APPROACHES TO ACM TRAINING

In the world of artificial intelligence, the term "rule" has a more restricted meaning than it does in ordinary language. Here it refers to a formal way of representing recommendations, directives, or strategies. Rules are usually stated in the form of IF-THEN statements like those shown below.

[1] **If** speed advantage of ownship over the target ship is greater than 110
Then display "Speed not adjusted properly."

[2] **If** GCI is not available
and FORMATION OF THE ENEMY is "Bearing/ Echelon"
and FIRST TARGETATTACKED is #1
and CLOSEST TARGET is #1
Then set SELECTED CORRECT TARGET to true

In a rule-based expert system, the domain knowledge is embodied as sets of rules that are verified against an assembly of facts or knowledge about the present situation. When the IF portion of a rule is fulfilled by the facts, the execution designated in the THEN segment is performed. When this happens, the rule is said to fire or execute.

Almost all rule-based expert systems have an inference engine. An inference engine contains an interpreter that decides how to apply the rules to infer new knowledge and some sort of scheduler that decides the order in which the rules should be applied (Waterman 1986). It is the inference engine, in combination with the rule base which makes the expert system "artificially intelligent."

Following are two systems which use a rule-based expert system to train ACM skills.

OBSERVING SYSTEM FOR CRITIQUE, ADVISE AND REVIEW

Vreuls Research Corporation (VRC) created an expert system which would provide a critique of a pilot's decisions about and ability to perform a "baseline stern conversion intercept" (Gray & Edwards, 1991; Edwards & Hubbard, 1991). Their system, the Observing System for Critique, Advise and Review (OSCAR) is designed to be used in a debrief setting, after a student pilot has flown an intercept.

The Critiquing Approach

A usual approach to designing an expert system is to create a program which simulates the expert's decision-making process. In contrast, a critiquing approach does not attempt to replicate an expert's decision making process. Instead, the system critiques it, discussing the pros and cons of the student pilot's approach and compares and contrasts it with alternatives which might be reasonable or preferred.

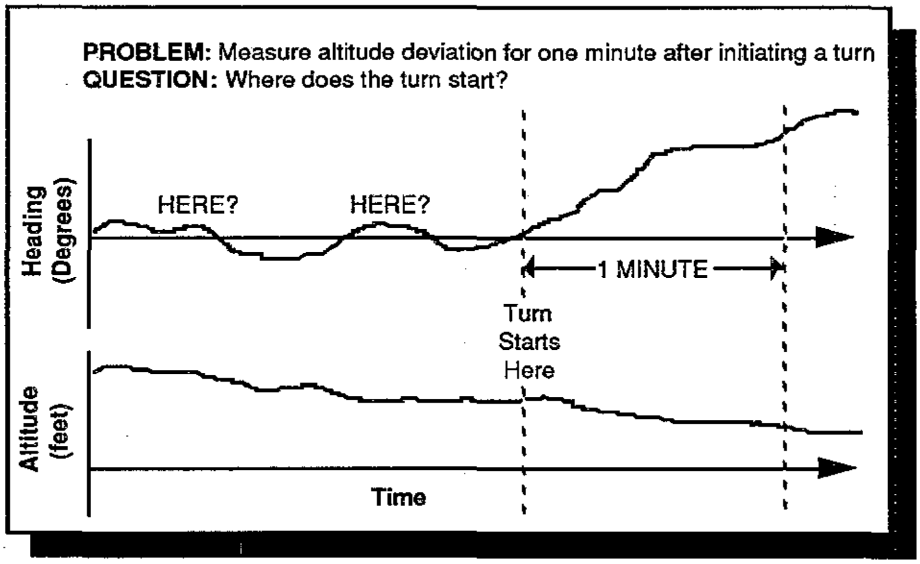

Figure 1.
Window Based Logic (adapted from Obermayer, 1991)

The inspiration for this kind of expert system is a medical diagnosis system called ATTENDING (Miller, 1984). ATTENDING instructs medical students in anesthesiology by critiquing students' plans for anesthetic management. It "presents the student with a hypothetical patient about to undergo surgery and analyzes the management plan devised by the student.... The critique produced by the system has the form of commentary, typically four or five paragraphs of English text. ATTENDING was developed at the Yale University School of Medicine and reached the stage of a research prototype." (Waterman, 1986 p. 274)

Obermayer (1991) holds that the following benefits can be obtained by such a system: (a) The approach casts the computer in the role of the user's helper, rather than a possible adversary, (b) the user must think through the problem, making him an active participant, rather than a casual observer, (c) because there is no "right way" to perform the task users can form their own idiosyncratic technique, (d) nuances, which can be difficult to anticipate and quantify, can be addressed, and (e) since the computer is playing a secondary role, it hands over the major instructional obligation to the student and instructor. This way the system merely helps the student and instructor evaluate and optimize the approach taken.

Performance Measurement

While it is desirable to cast the expert system as an ally to the student it is never-the-less a fact that the system must be able to critique a student's proficiency. To do so it must be able to measure and assess a student's performance. Consequently, the creation of OSCAR necessitated addressing several issues central to performance measurement, including automated performance measurement, near-real-time measurement, performance diagnosis, and measurement of expert performance (Vreuls and Obermayer, 1985; Obermayer, 1991). Part of OSCAR's performance measurement system was based upon a back-looking windowing approach with lagged logic. Figure 1 shows that this technique works by buffering a sufficient amount of information into a window. When sufficient data are collected a decision can be made as to what the student was doing. In actuality, this technique takes two passes at the recorded data. The first pass determines what the students actions were, the second pass assesses the students performance.

The actual student critique is created by running the performance data through a commercially available, rule-based, expert system shell. A major portion of developing OSCAR was determining the rules (termed knowledge engineering) for critiquing air intercepts. All of the information used in this knowledge engineering effort was obtained from unclassified sources such as the current air intercept tactics manual (Department of Defense, 1989). The rules were then reviewed by subject-matter experts to determine their accuracy and validity.

Performance Feedback

OSCAR uses a computer generated display to provide feedback to students. As seen in Figure 2 the computer display provided quite a bit of information. For example the top of Figure 2 shows that students were presented with a radar display and aircraft variables as they occurred during the intercept scenario. In addition students could view a plan-view and a horizontal-view of

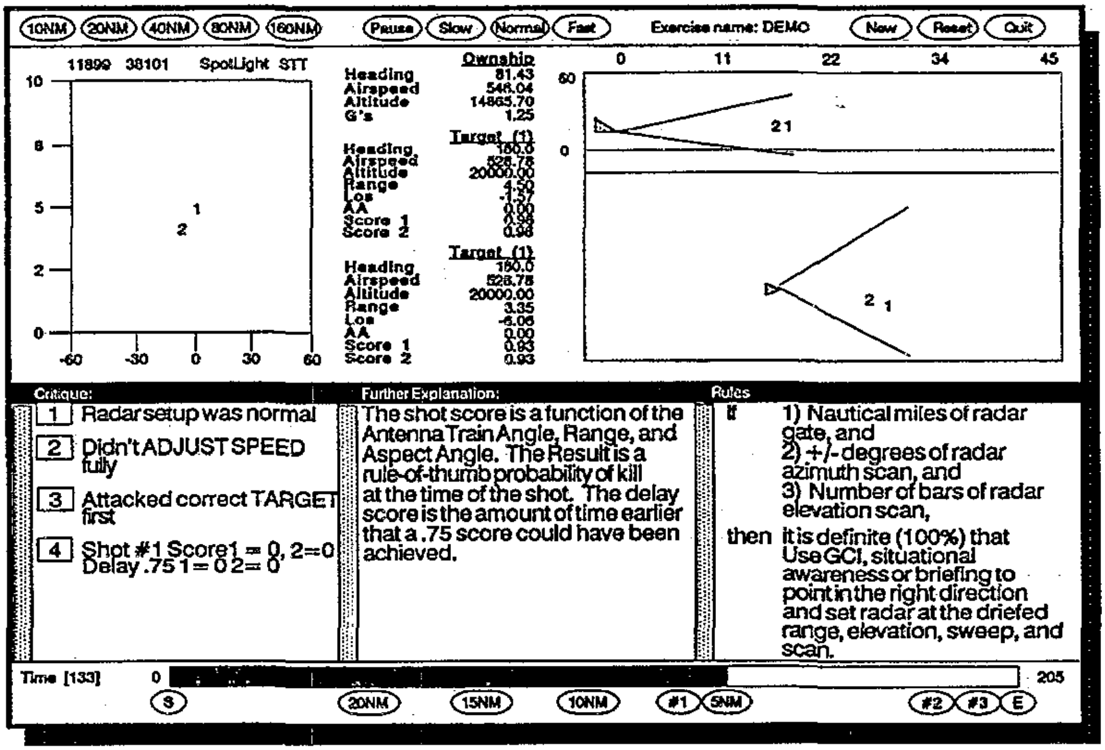

Figure 2.
OSCAR Screen Display (from Obermayer, 1991)

the intercept. These are familiar views for student pilots and they have relatively little problem in integrating the information form these three displays.

The bottom of the screen presents a time-line display. Students can replay any portion of the intercept scenario by pointing (with a mouse) at the timeline.

The middle of the screen presents the actual critique information. This is found in the left box. In the middle box students can see additional information such as unfamiliar terms, general rules of thumb, etc. The box on the right contains the actual rules which were used to determine the narrative found in the critique box.

Strengths/Weaknesses

A preliminary evaluation of the OSCAR system was conducted at the 58th Tactical Training Squadron, Luke Air Force Base, AZ. Data were collected from low time students (students with very little radar or intercept experience), high time students (students with a good foundation in air intercepts and are advancing into multiship tactical intercepts), and instructor pilots. All categories of OSCAR users rated the system very highly. In fact the more instructor pilots used the system, the higher they rated it (Obermayer, 1991). From this one can assume that the more instructor pilots used OSCAR the more they were able see how it would unburden them and allow students to learn on their own.

One serious drawback was the fact that OSCAR only does its critiquing after the intercept is performed. Students were first required to perform an intercept on an Air Intercept Part-Task Trainer (Gray & Edwards, 1991; Edwards & Hubbard, 1991). Then they had to wait for OSCAR to read the data and create the critique. This waiting period could be as long as 5 to 7 minutes. Some students felt that it was a real drawback that the system took as long as 7 minutes to come up with a critique.

Another related weakness of the system was the fact that they had to wait at all. Some students wanted to be able to have a critique in real time as they were flying. Others wanted to be able to start the intercept again at the point where an error occurred.

ACM EXPERT SYSTEM TRAINER

Merit Technology Inc. created an expert system which would give a student pilot feedback during a simulated ACM exercise. According to Bechtel (1992) the "ACM

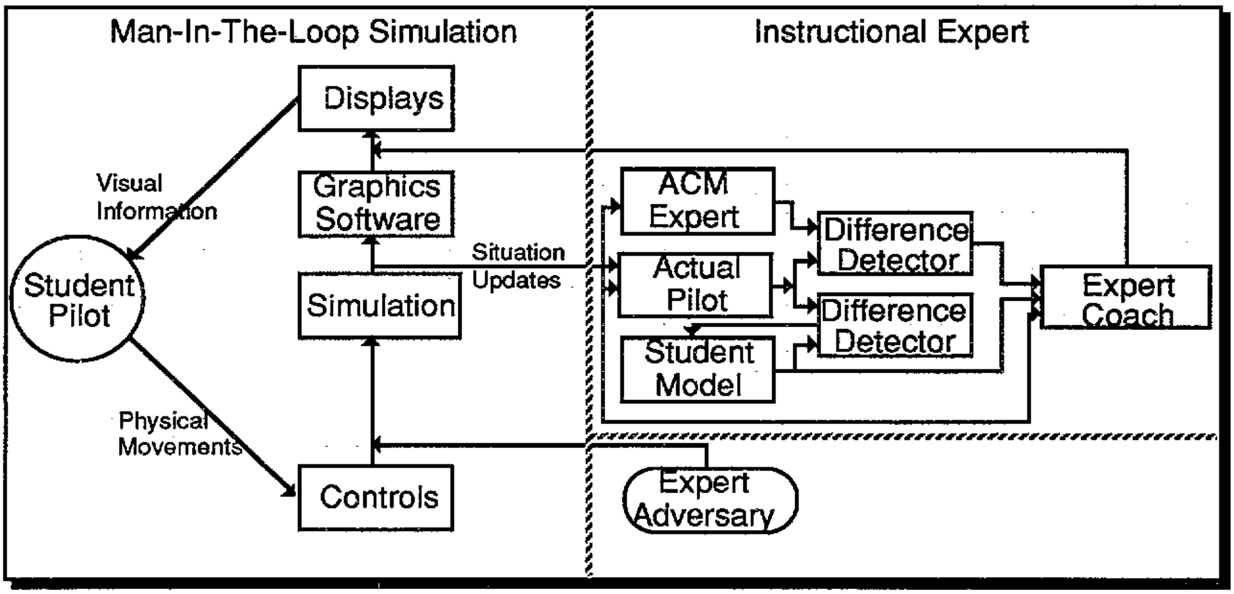

Figure 3.
ACMEST Configuration (adapted from Bechtel, 1992)

Expert System Trainer" (ACMEST) was designed to provide the following components:

- a dynamic, man-in-the-loop, real-time simulation with one pilot flight control over realistic fighter aircraft models and autonomous goal-directed (computer) control of adversary fighter aircraft models (models to include realistic flight equations, sensors and weapons);
- graphics simulations with displays capable of indicating the relative position and geometry of up to three aircraft simultaneously from any perspective (including cockpit out-the-window displays with realistic head-up display (HUD) symbology);
- a knowledge-based ACM tactics planner capable of integrating the current situation, training objectives and an expert model of offensive and defensive pilot decision making to determine the best course of action for the student pilot to follow at any given point in time during the simulated mission; and
- a computer-aided instructional (CAI) subsystem capable of comparing the pilot-selected and "school" courses of action, ascertaining the differences, and preparing a training plan of on-line, scenario replay and post-flight tutorial material. (p. 2)

The Coaching Approach

Instead of using a debrief approach like OSCAR (above), ACMEST is designed to provide an on-line, interactive coach to student pilots as they learn air combat maneuvers. It is designed to: a) coach fighter pilots to perform basic air combat maneuvers, b) recognize situations in which each maneuver is appropriately applied, and c) learn the critical decision process likely to be used during air combat. The system continuously evaluates student pilots to determine their level of proficiency and automatically informs them of critical departures from expected performance.

This coaching is done by displaying text in a message bar on the out-the-window view whenever student pilots' actions diverge from what the expert system considers to be the "correct" actions. All the expert coaching in ACMEST occurs during the simulated flight, as though an instructor pilot is looking over the student's shoulder.

Performance Measurement and Feedback

Figure 3 describes the basic process which by which ACMEST is "flown." The left side of the figure represents a fairly straightforward implementation of a moderate fidelity, man-in-the-loop flight simulation system. The right side shows however, that there are two additional components which enable ACMEST to provide expert system based ACM coaching. First, ACMEST employs an expert adversary. This is a "computerized behavioral model of of a skilled pilot" (Bechtel, 1992). In ACMEST this expert adversary a self-contained, production rule-based, expert system. This adversary model has the ability to control a second simulated aircraft and can fly as an intelligent opponent to the student pilot.

Second, ACMEST incorporates an instructional expert into the system. As Figure 3 shows, the instructional expert receives situation updates from the student pilot through the simulation loop. This information is fed directly to two expert subsystems, the ACM Expert and the Coach. The ACM Expert is a production rule based model of an ACM competence. This expert knows what to do and can product the correct actions in any air combat situation for which it has rules. As situation updates come to the ACM Expert, it produces a set of recommended "correct" actions. These are compared with the actual pilot actions in a difference detector.

The Coach subsystem is an expert system which can provide training feedback to student pilots. This is done by comparing the differences between the ACM expert, the current model of the student, and current pilot actions. Once actual pilot behavior has been compared with ideal behavior (taking into account the present ability and understanding of the pilot as contained in the student model) the expert coach produces instructional actions. These can "run the gamut form no current action through suspension of the simulation and presentation of tutorial text." Most frequently, the instructional action selected is the display of one or more "hints" to the student as to the proper course of action (Bechtel, 1992).

Strengths/Weaknesses

Perhaps the greatest strength of this effort was in finding that a real-time, rule-based expert system could be implemented in a simulation setting. It has generally become a rule-of-thumb that production system based inferencing can not keep up with an evolving situation such as found in real-time ACM. Conventional wisdom holds that the rule base inference engine would just fall further and further behind as the flight/fight progressed. ACMEST however, showed that a 386 PC was able to support multiple aircraft, rules of expertise, an expert coaching system, and at the same time port graphics to a dedicated graphics machine. Given the fact that a 386 class PC was "state-of-the-art" when this project was started, it is extremely encouraging to contemplate what could be accomplished in light of today's desktop computers.

A second positive result of this effort was the product. Students now have a real time simulation which can coach them through the rigors of ACM training. They need not be dependent on after action reports, debriefs, or post-mission critiques. Instead it has been shown that a simulation can incorporate expert coaching directly into its interface.

The above optimism needs to be tempered however, with a few caveats. First, it should be noted that the interface between student and expert coach does not seem optimal. As presently implemented, the coaching system prompts students through a text block on the simulator screen. This can be a major drawback because reading text on the screen requires considerable attention and can draw students' focus away from where they should be looking. At one point ACMEST developers actually disabled the text critique and instead presented students with synthesized speech instead. They found this to be even more distracting than text but they speculated that it was due to the poor quality of voice synthesis systems which were commercially available at the time.

Perhaps ACMEST's greatest weakness is the fact that it is a rule-based expert system. While the developers were able to successfully extract knowledge of ACM tactics from subject matter experts and from documentation such as Shaw (1985), their results are still a bit mixed.

First, it should be recognized that extracting ACM knowledge from fighter pilots and structuring it as a set of if-then rules can be a daunting task. It requires a thoroughness that can be nearly impossible to achieve. In terms of ACMEST, the knowledge engineering effort paid off with a usable coaching system. However, that coaching system only scratches the surface of a real ACM expert's knowledge base.

Second, as the knowledge base expanded into more and more specialized tidbits of information, it became necessary to stretch the limits of a rule-oriented representation technique. The developers noted that: "oftentimes we found it necessary to write rules which seem unnecessarily convoluted or strained to enable some necessary interface." (Bechtel, 1992)

Third, rule-based expert systems are notoriously brittle. That is, the system is only an expert in the limited domain for which rules have been developed. If an expert system is given conditions for which no rules exits, it will simply fail to function properly. While ACMEST has been well thought out, and the rule base is quite involved, it will also exhibit brittle tendencies when presented with air combat conditions it was not explicitly programed to address.

NEURAL NETWORK APPROACH TO ACM TRAINING

The above paragraphs point out that rule-based expert systems have several disadvantages which may make ACM training less than optimal. What is needed is an more robust technique for acquiring ACM knowledge and a way to "program" a system which alleviates the problem of expert system brittleness. The field of Artificial Neural Networks, (also known as Connectionism and Artificial Neural Systems) has shown quite a bit of promise in overcoming some of the more intractable elements of simulating intelligent behavior (Gallant, 1988; Crowe, 1991). First, neural networks make it easier for knowledge engineers to overcome the knowledge acquisition bottleneck since they don't need to generate precise sets of rules. Instead neural networks learn to copy the experts actions and to correct their own mistakes as they "learn.". Second, neural networks can be set up to be extremely fault tolerant. That is, they can trained to respond with reasonable behaviors despite having incomplete knowledge,

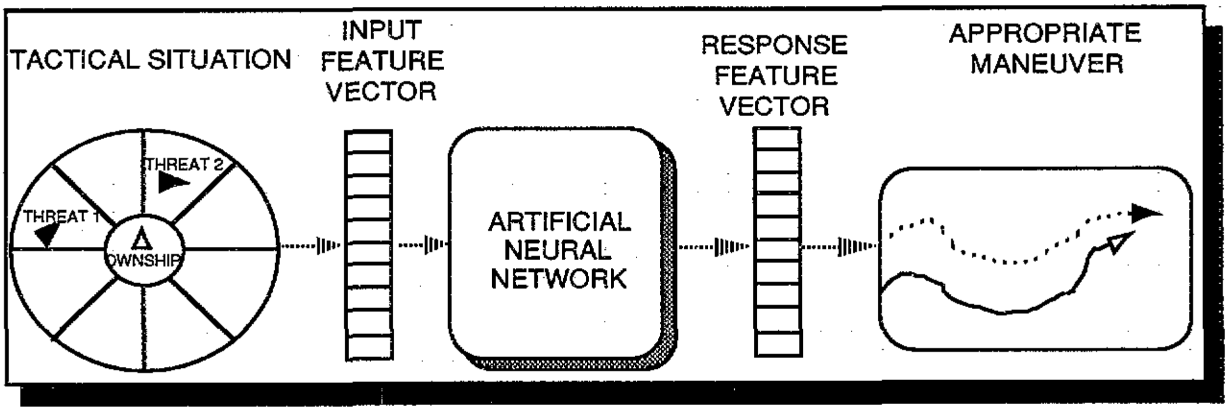

Figure 4.
ACMTES Approach to Neural Network Modeling *(adapted from Crowe, 1991)*

or noisy data. Unlike rule-based expert systems, neural networks are even able to respond gracefully when they are presented with situations which they have never seen before. Finally, neural networks seem to be more suitable for real-time simulation based systems. As Crowe (1991) points out; "This is because, unlike the training process, which may require many iterations of example data, a response to input requires only a single feed-forward pass through the network with a consistent time requirement. " (p. 305)

Following is a description of an ACM training system which is based upon neural network technology.

ACM TRAINER EXPERT SYSTEM

Ball Systems Engineering Division created an artificial neural system which would produce realistic one-versus-one air combat maneuvers under "within visual range" tactical situations. This system is called the ACM Trainer Expert System (ACMTES). According to Roorda and Crowe (1992):

> At the implementation level, the overriding questions is: How can neural network technology best be applied to bring about new solutions in the simulation of ACM decision making? The goal of this effort was to create a system which takes situational data as input and combines it in the proper way to produce a reasonable and realistic maneuver as output. Input data takes the form of relative geometry and specific aircraft parameters which the pilot might use and would have available during a real ACM engagement. The chosen form of output control is the amount of heading, pitch, and velocity control required throughout the flight envelope. The resulting system provides a working framework for the evaluation of neural networks which simulate air combat maneuvering and allows for the extension of these results to other simulations of expertise and training environments. (p. 3)

The Neural Network Approach

Figure 4 shows the basic approach by which ACMTES was created. As shown in the figure, the neural network "learns" to associate a tactical situation to a set of appropriate maneuvers. This learning process is accomplished by showing the neural network the tactical situation by means of an input database. The neural network then defines a set of outputs which it deems appropriate for the situation. The network is then given the correct solution to the problem and is instructed to modify its underlying structure so that its output will match the correct solution. This process of selective approximations continues until the neural network arrives at a solution which closely matches the optimal solution.

The underlying network is constructed as a hierarchy of layers. Each layer contains a number of simulated neurons, known as processing elements (PEs). Figure 5 shows the structure of a typical three-layer network. The network "learns" by changing the weights between connected PEs. From this one can see that the neural network is not programmed conventionally. Rather, expertise is arrived at by the weighted connections of the network's PEs.

In the case of ACMTES, the initial training data was obtained from a series of scenarios flown on the Simulator for Air-to-Air Combat (SAAC) at Luke AFB. All the scenarios were flown by active-duty Air Force Pilots against the same adversary in F-16 versus F-16 engagements. In order to obtain enough training data, long scenarios were selected over short ones. (This fact will become important when the systems strengths and weaknesses are discussed below.)

TRAINING WITH ACMTES

Once the ACMTES neural net is trained on F-16 ACM maneuvers it can be used as a training tool itself. Figure 6

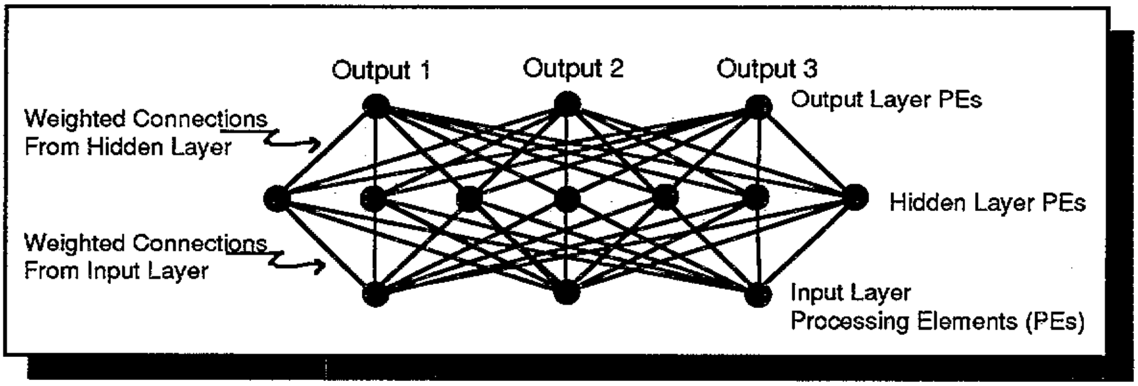

Figure 5.
Structure of a Typical Three Layer Network

shows a representation of the computer display of ACMTES when it is used in training. As can be seen, the computer displays important parameters of the own and adversary aircraft in both alphanumerical and graphic formats. The graphic portion of the display shows several views of an unfolding ACM engagement. In Figure 6 the computer is showing a situation in which two aircraft have been in a turning, descending fight. The engagement begins with each aircraft separated by two nautical miles with an altitude of 15,000 feet, flying straight toward each other at 500 knots. Upon passing each other both aircraft turn east and start to engage in what appears to be a diving-rolling scissors tactic. Both aircraft exchange an altitude advantage while descending toward the ground and terminate the engagement at about 500 feet in order to avoid hitting with the ground.

Students can use the computer keyboard or joystick to fly one of the aircraft against the neural network for practice in learning to predict what a "smart bogey" will do. They can also let both aircraft be controlled by neural nets and learn from the ensuing engagement what are (and what are not) appropriate maneuvers.

Strengths/Weaknesses

Evaluation of the ACMTES system showed not only that a neural net could learn to perform air combat maneuvers but that it could learn to generalize to the point where it could perform well even under novel tactical situations. ACMTES neural networks have shown some unique capabilities to overcome some of the more difficult aspects of knowledge engineering. It has shown that it can produce robust, generalized solutions even under novel circumstances.

By capturing and simulating the knowledge and skill of human pilots in a neural network, students are furnished with expert training devices which have the look and feel of real air-to-air combat (Crowe, 1991). In short, it seems that neural network based approaches show a great amount of promise in the area of ACM training.

Still, there are some weaknesses of the system which should be addressed. First, it should be noted that ACMTES is "flown" from the keyboard of a 386 class personal computer. In addition, as Figure 6 shows, the ACMTES screen displays are all from a perspective extrinsic from the ownship. This makes "flying" ACMTES extremely difficult. It needs to be incorporated into a flight simulation package which is more ergonomically designed.

Second and more importantly, ACMTES shows that it is crucial to give the correct training data to neural networks (both artificial and human). Recall that most of the SAAC training data upon which ACMTES is based came from long scenarios as opposed to short ones. The developers wanted to get the most data possible so that ACMTES could be trained in a wide variety of tactical situations. However, the developers may not have realized, (or perhaps realized too late) that long scenarios are qualitatively different from shorter ones. Long scenarios tend to turn into stalemates where neither side wins. Short engagements usually show a clear victor very quickly. Because ACMTES was trained with stalemate type engagements it exhibits the tendency to opt for stalemates itself! Under current training, it seems to be a very benign fighter.

Actually, ACMTES seems to exhibit another training problem. It seems that the SAAC data contained no engagements where the opponent is not maneuvering. In all scenarios, both aircraft engaged in very drastic, high speed, high - g, combat maneuvers. Because ACMTES never "saw" nonmaneuvering targets during training, it does not exhibit a capability to deal them in the simulation. Instead, if an opposing aircraft does not make an aggressive move against ACMTES first, then ACMTES will not respond. It seems that a major character trait of

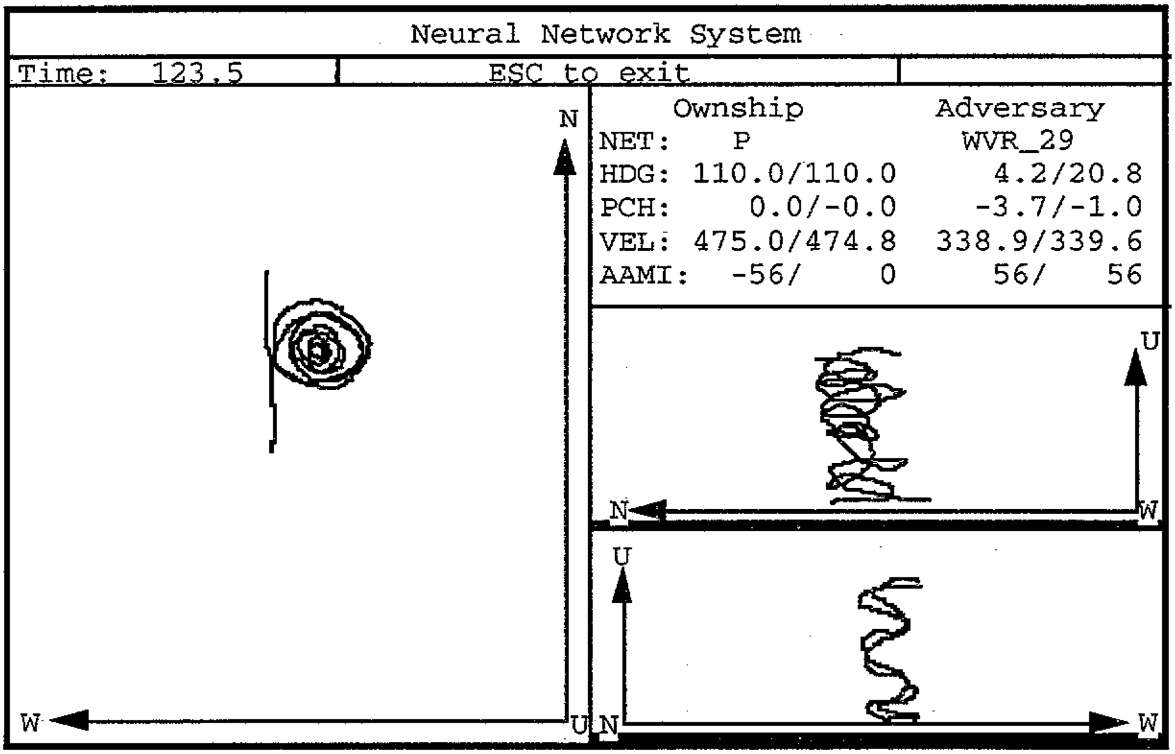

Figure 6.
ACMTES Display (adapted from Roorda & Crowe, 1992)

the most successful combat pilots, "aggressiveness," has been trained out of ACMTES.

A third weakness of ACMTES is that fact that neural networks are "black boxes." Unlike conventional expert systems, where one can examine the "rules" and determine what the expert system is "thinking," a neural network is not readily open to examination. If one were to examine the "insides" of a neural network all one would see are numbers representing weights to other numbers representing PEs. Thus ACMTES does not lend itself to examining the critical decision making going on just under the surface of an expert system. In certain training situations, it can be advantageous to see what an ACM expert is thinking or how it is arriving at a given determination. With ACMTES this is not easily achieved.

SUMMARY

While nothing in the foreseeable future appears capable of replacing the requirement of actual aircraft flight in developing ACM proficiency, the technology is now available to augment actual flight hours with meaningful training from computer-based simulations. At the Aircrew Training Research Division of the Armstrong Laboratory we have been developing several Artificial Intelligence (AI) based approaches to augmenting pilot training in simulation-based ACM. Because air-to-air combat is such a fast moving, complex task, automating (through AI) such tasks as performance measurement and assessment can provide a very important enhancement to a simulation. In addition, using AI techniques to create "smart bogeys" can provide a real boost in the training capabilities of a simulation.

REFERENCES

Bechtel, R. J. (1992). Air combat maneuvering expert system trainer (AL-TP-1992-0058). Brooks AFB, TX: Human Resources Directorate.

Crowe, M, X. (1991). The application of artificial neural systems to the training of air combat decision-making skills. Proceedings from the 13th Interservice/ Industry Training Systems Conference. pp. 302-312).

Department of Defense (1989). Mission Employment Tactics, Fighter Fundamentals, F-16 (TAC/PACAF/ USAMEM 3-3, VOL. V).

Edwards, B. J. & Hubbard, D. C. (1991). Transfer of training from a radar intercept part-task trainer to an F-16 flight simulator (AL-TR-1991-0038). Brooks AFB, TX: Human Resources Directorate.

Gallant, S. I. (1988). Connectionist expert systems. Communications of the ACM, 31, 152-169.

Gray, T. H. & Edwards, B. J. (1991). Aircrew part-task training research and development in the 1980's: Lessons learned (AL-TR-1991-0001). Brooks AFB, TX: Human Resources Directorate.

Kelly, M. J. (1988). Performance measurement during simulated air-to-air combat. Human Factors, 30, 495-606.

Obermayer, R. W. (1991). Observing system for critique, advice and review (AL-TR-1991-0059). Brooks AFB, TX: Human Resources Directorate.

Roorda, J. M. & Crowe, M. X. (1992). The application of artificial neural systems to the simulation of air combat decision-making (AL-TP-1992-00). Brooks AFB, TX: Human Resources Directorate.

Semple, C. A., Cotton, J. C. and Sullivan, D. J. (1981). Aircrew training devices: Instructional support features (AFHRL-TR-80-58). Brooks AFB, TX: Air Force Human Resources Laboratory.

Shaw, R. L. (1985). Fighter Combat: Tactics and Maneuvering. Annapolis: Naval Institute Press.

Vreuls, D. & Obermayer, R. W. (1985). Human-system performance measurement in training simulators. Human Factors, 27, 241-250.