

ARTIFICIAL INTELLIGENCE (AI) BASED TACTICAL GUIDANCE FOR FIGHTER AIRCRAFT

John W. McManus
Kenneth H. Goodrich
NASA Langley Research Center
Mail Stop 489
Hampton, Virginia 23665-5225
(804)864-4037/(804)864-4009

AIAA Guidance, Navigation, and Control Conference
August 20-22, 1990
Portland, Oregon

ABSTRACT

A research program investigating the use of Artificial Intelligence (AI) techniques to aid in the development of a Tactical Decision Generator (TDG) for Within Visual Range (WVR) air combat engagements is discussed. The application of AI programming and problem solving methods in the development and implementation of the Computerized Logic For Air-to-Air Warfare Simulations (CLAWS), a second generation TDG, is presented. The Knowledge-Based Systems used by CLAWS to aid in the tactical decision-making process are outlined in detail, and the results of tests to evaluate the performance of CLAWS versus a baseline TDG developed in FORTRAN to run in real-time in the Langley Differential Maneuvering Simulator (DMS), are presented. To date, these test results have shown significant performance gains with respect to the TDG baseline in one-versus-one air combat engagements, and the AI-based TDG software has proven to be much easier to modify and maintain than the baseline FORTRAN TDG programs. Alternate computing environments and programming approaches, including the use of parallel algorithms and heterogeneous computer networks are discussed, and the design and performance of a prototype concurrent TDG system are presented.

INTRODUCTION

The increased capabilities of modern weapons and sensor systems and the expanded capabilities and flight envelopes of high performance aircraft have changed the requirements of air combat simulation systems. A modern and realistic air combat simulation that can be used to evaluate the current and future air combat environments must have the ability to model

superagile aircraft as well as new weapons systems, aircraft subsystems such as sensors or propulsion systems, modifications to existing aircraft control systems, and changes to the aircraft's structural configuration. In support of the study of superagile aircraft at Langley Research Center (LaRC), a Tactical Guidance Research and Evaluation System (TGRES, pronounced "tigress") is being developed.^{1,2,3}

The TGRES system^{1,2,3}, shown in figure 1, is designed to allow researchers to develop and evaluate systems in a tactical environment. While TGRES is aimed specifically at the development and evaluation of maneuvering strategies and advanced guidance/control systems for superagile aircraft, the modular design of TGRES will make it easily adaptable to the analysis of other aircraft systems. The three main components of TGRES are a TDG, the Tactical Maneuver Simulator (TMS), and the Differential Maneuvering Simulator (DMS). The TMS and the DMS are described in greater detail in^{1,3}; the design and implementation of the current TDG system, CLAWS, and a prototype concurrent TDG will be detailed in this paper.

The TMS^{1,3} provides a high-fidelity batch air combat simulation environment for the development and testing of various guidance and control strategies. The researcher defines the initial conditions of the engagement and the TMS then executes the trajectories and attitudes of the aircraft using simple trajectory commands or through a tactical guidance system. The three main elements of the TMS are a high-fidelity, nonlinear six degrees of freedom (dof) rigid-body dynamic aircraft model, a TDG, and a user interface.³

TACTICAL GUIDANCE RESEARCH AND EVALUATION SYSTEM (TGRES)

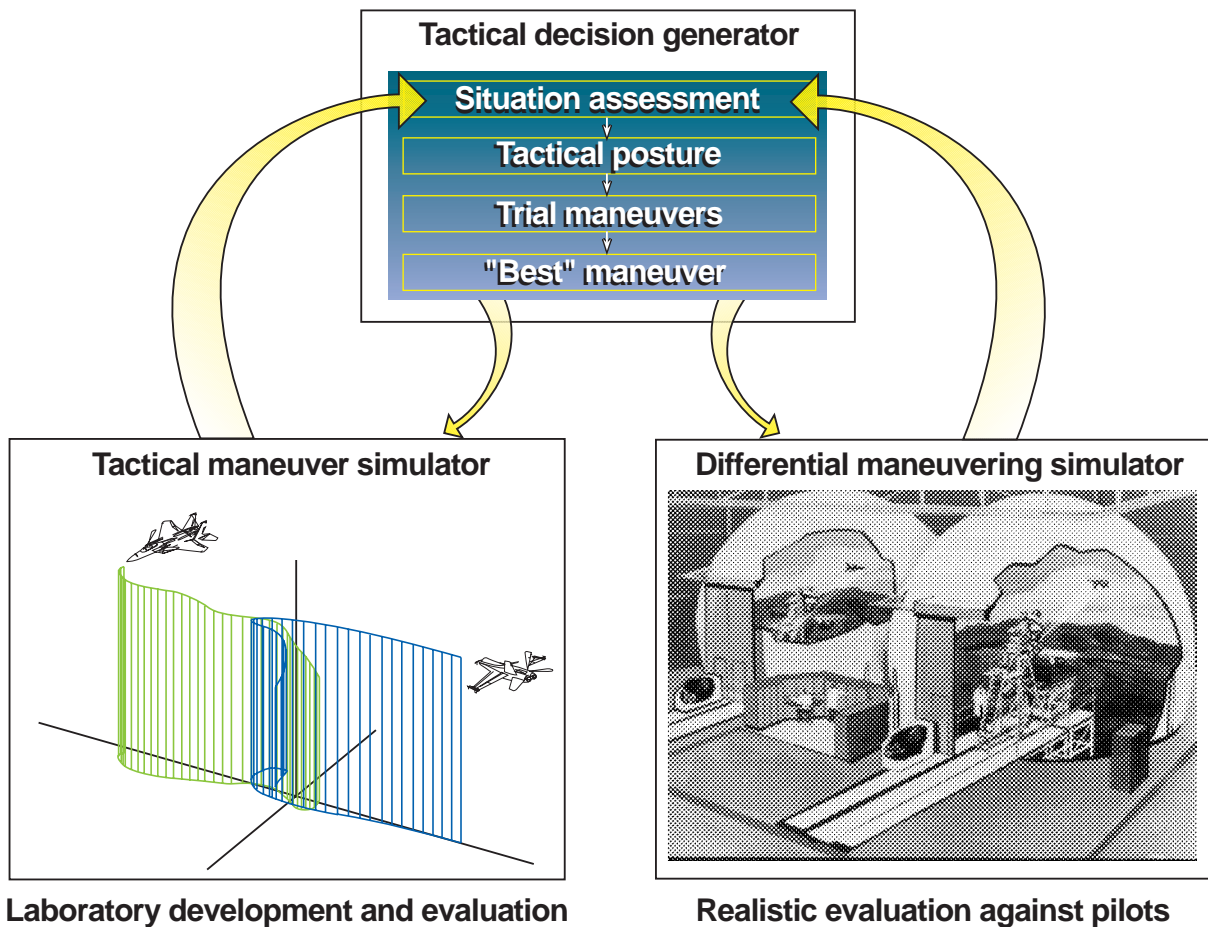


Figure 1. TGRES

The DMS consists of two 40' diameter domes and a 20' diameter dome located at Langley Research Center. The facility is intended for the real-time simulation of air combat engagements between piloted aircraft. By using a TDG to control one of the airplanes, it is possible to test the TDG against a human opponent. This feature allows the guidance logic to be evaluated against an unpredictable and adaptive human opponent. The recent addition of the third dome and the required target projectors and software allow the guidance logic to be evaluated in one-versus-two or two-versus-one scenarios, further enhancing the tactical capability of the DMS environment.

CLAWS is a knowledge-based TDG designed to provide researchers insight into both the tactical benefits and the costs of superagility^{1,2}. Knowledge-Based Systems use a large amount of information about a problem's domain to help understand the problem being

solved. The knowledge is stored within the program using some knowledge representation scheme like logic, procedural semantics, semantic networks, frames, or objects. CLAWS was developed as an Object Oriented Blackboard system in LISP using a Symbolics 3650[†] workstation.

A Blackboard system consists of a set of specialized Knowledge Sources, a centralized blackboard data structure, and a control strategy used to activate the knowledge sources. The blackboard model of problem solving is best described by H. Penny Nii.

" A Blackboard System can be viewed as a collection of intelligent agents who are gathered around a blackboard, looking at pieces of information written on it, thinking about

[†] Symbolics 3650 is a registered trademark of Symbolics Incorporated

the current state of the solution, and writing their conclusions on the blackboard as they generate them." 5

The blackboard is a global data structure, often partitioned in a hierarchical manner, used to represent the problem domain. The blackboard is also used to allow inter-knowledge source communication and acts as a global shared memory visible to all of the knowledge sources. This design allows for opportunistic problem solving and allows a knowledge source to contribute towards the solution of the current problem without knowing which of the other knowledge sources will use the information. It is important to note that although knowledge sources are often referred to as "experts", knowledge sources are not restricted to Expert Systems or other AI systems. Many knowledge sources are numeric or algorithmic in nature.

CLAWS has been designed with separate subroutines and specialized computer hardware for the aircraft simulation and the TDG knowledge sources. The separation of the aircraft simulation and decision logic components and the use of highly specialized knowledge sources allows each module / knowledge source to be designed and implemented using the hardware and programming techniques specifically suited for its function. The use of highly specialized and independent knowledge source also provides for "modular protection"⁶, confining the effect of an error occurring at run-time in one module to that module, or to a small set of neighboring modules in the program. The confining effect of the modular protection can be used to aid in the design and debugging process. Each knowledge source can be developed and tested independently before it is incorporated into CLAWS.

The independence of the knowledge sources also increases the efficiency of CLAWS by allowing knowledge sources to be distributed across a network of several heterogeneous processors. The network currently consists of a Symbolics 3650 workstation, a MacIvory[†] workstation, and several Vax 3200[†] class workstations. Communication between the blackboard and the knowledge sources is achieved using customized DecNet based Client/Server software developed in 1988 for TGRES at LaRC. This software allows for synchronization, communications, and data sharing between heterogeneous computers running DecNet. Since

[†] MacIvory is a registered trademark of Symbolics Incorporated

[†] Vax 3200 & DECNET are registered trademarks of Digital Electronics Company

the current CLAWS is a serial blackboard system no serialization or concurrency related software is required. Each knowledge source requests from the blackboard all of the data required to perform its computation at the start of its execution cycle, and posts its results to the blackboard at the end of its execution cycle.

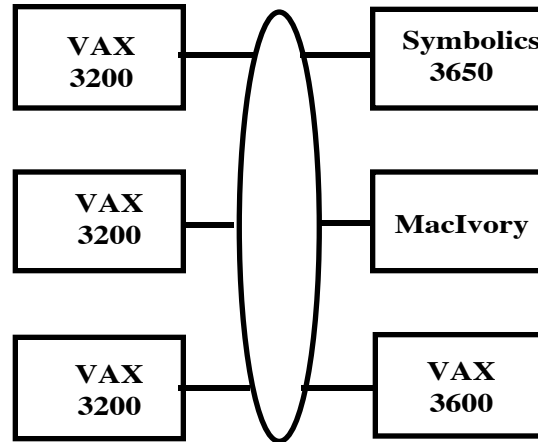


Figure 2. AGCB Computer Network.

KBS MODULES OF CLAWS

The development of the TDG has been a multi-stage process and two preliminary TDG's have been reported on^{1,2}. The COSMIC FORTRAN version of AML⁷ was used as an initial starting point. The current version of the TDG, CLAWS, is a blackboard based system written in LISP that uses an object-oriented programming approach to represent the aircraft states and subsystems. CLAWS uses the object-oriented programming approach to represent each aircraft in the simulation and the current state of the aircraft's offensive systems, defensive systems, and engines. This information is used by CLAWS's knowledge sources to help guide the reasoning process. CLAWS uses the trial maneuver concept outlined in the AML program with several extensions². CLAWS is being modified to incorporate the use of existing "optimal" guidance algorithms. These algorithms are used in conjunction with the existing trial maneuver approach. If the maneuver selection logic selects a guidance algorithm the logic will "switch" the algorithm on and monitor its effectiveness. If the performance of the guidance algorithm falls below expected performance, the system will revert to the trial maneuver selection mode. This additional feature will allow researchers to evaluate the performance of the existing "optimal" guidance algorithms in a high angle of attack combat scenario.

CLAWS has a knowledge-based Situation Assessment knowledge source that is executed at

each decision interval before the trial maneuvers are evaluated. The situation assessment knowledge source is used to determine the current mode of operation. The situation assessment knowledge source is executed at the start of each decision interval, before the maneuver scoring module, and determines the mode of operation. This determination is based on the aircraft's current mission, the current state of the aircraft's systems, the relative geometry between the aircraft and its opponent, and the opponent's instantaneous-intent. Each of the six mode of operations, table 1, has a unique vector of scoring weights and a unique decision interval associated with it. The scoring weights for each mode have been adjusted during the design and testing process to maximize CLAWS's performance in that mode of operation.²

Modes of Operation
Aggressive
Evasive
Missile Evasion
Ground / Stall Evasion
Evading opponent's "Lock-on"
Defensive
Neutral
Bugout

Table 1.

Both TMS and DMS test results^{1,2} have shown that a short decision interval, (0.5 sec.), improves the fine tracking performance in aggressive and evasive situations. In neutral or defensive situations the same short decision interval results in a "thrashing" motion degrading system performance. The thrashing is due to the system overcompensating for small changes in the motion of the opponent. These thrashing maneuvers bleed off excessive energy in the neutral and defensive situations. A longer decision interval, (1.0 sec.), is used in neutral and defensive situations to keep the system from thrashing and wasting energy. The situation assessment KBS also determines the opponents instantaneous-intent. The opponent's instantaneous-intent is defined to be an estimation of the opponent's intent at the current point in time based on CLAWS available sensor, positional, and geometric data. Currently, there is no attempt to use a history of instantaneous-intent to derive a long-term opponent intent. The flexibility provided by the use of mode of operation's and instantaneous-intent allows the system to more closely model the pilots changing strategies during the engagement.

The Scoring Module knowledge source is a KBS that uses a set of 17 fuzzy logic questions with responses ranging from [-1= NEGATIVE, ..0 = NEUTRAL, ..1.0 = POSITIVE] and the vector of mode-specific scoring weights selected by the situation assessment module to score each of the trial maneuvers. For each trial maneuver in the test set the updated positions for both the opponent and the CLAWS aircraft are computed. The position of the opponent is projected using a simple curve fit based on the past three seconds of the opponent aircraft's trajectory. The new position of the CLAWS aircraft is determined by executing the trial maneuvers control commands. The relative geometry between the two updated aircraft positions is determined and the score for the maneuver is then determined by evaluating the responses to the seventeen fuzzy logic questions, applying the selected scoring weight vector, and then summing the responses to generate a single numeric score. After all of the trial maneuvers have been evaluated the highest scoring maneuver is selected and the associated control commands executed.

A rule-based active Throttle Controller has been developed to actively control the throttle setting based on the current mode of operation. The throttle controller is called at the start of each decision interval and can set the throttle to any position from idle to full afterburner [0, ..., 2]. The throttle controller uses the current mode of operation and relative geometry information to select from a target acquisition mode, a fine tracking mode, or a target or missile avoidance mode. Each mode has a set of specific throttle control rules that are used to maximize system performance in that mode.

CLAWS TESTING PROCEDURES

CLAWS is currently being tested in the TMS and in the LaRC DMS using five dof aircraft dynamics. DMS test results are reported in Goodrich³. TMS testing is done in a non-real-time, batch mode environment against the TDG*² baseline. Each set of tests consists of 32 sets of initial aircraft conditions as shown in figure 3. The initial altitudes, airspeeds, and the separations between the two aircraft are adjusted for each set of test runs. All of the sets of initial conditions can be classified as Within Visual Range (WVR) engagements. The largest initial aircraft separation used places the aircraft at the transition point between Beyond Visual Range and WVR.

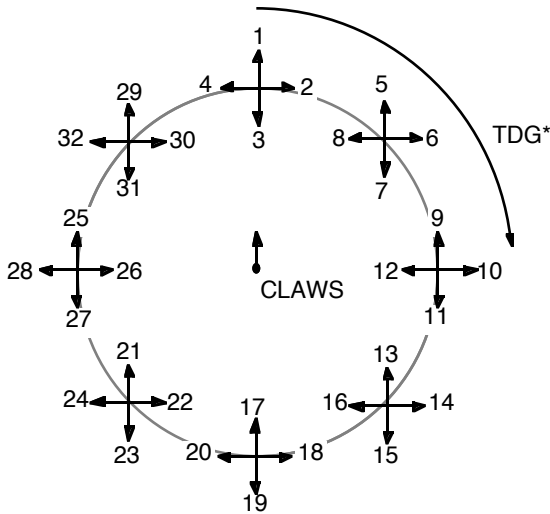


Figure 3. Initial Conditions.

A run evaluation module is used to calculate the amount of time that each aircraft has its weapons locked on its opponent; and the relative geometry between the aircraft. The Line-Of-Sight (LOS) vector is defined as the vector between ownship c.g. and opponent's c.g. The Line-Of-Sight (LOS) angle is defined as the angle between the LOS vector and ownship body x-axis; the deviation angle is defined as the angle between the LOS vector and ownship velocity vector; and the angle off is defined as the angle between the LOS vector and opponent's velocity vector.

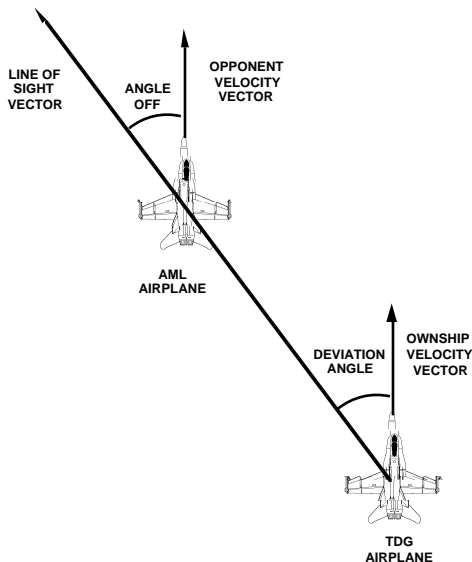


Figure 4. Angle Definitions

The weapons cones used represent a generic all-aspect missile, a generic tail-aspect missile, and a 20 mm cannon.

Four scoring metrics are currently used to evaluate each engagement. All metrics are computed at the aircraft simulation update rate of

32 times per second. The first metric computes the total time that each airplane has its weapons locked on the opponent, the probability that any weapons fired will hit the opponent, the distance between the opponents, the angle-off, and the deviation angle. The results are printed in a table format at the completion of each run.

The second scoring metric computes a Probability of Survival using the data computed by the first metric. The probability to hit for a missile and for the cannon are computed using the range and LOS angle to the opponent. The aircrafts' missiles are treated as a limited resource and a probability to hit of 0.65 is required to launch the first missile. The firing threshold increases by 0.05 for each missile launched, and all missiles are required to complete their flight to the target before the next missile is fired.

The third scoring metric was developed at LaRC and attempts to determine a Lethal Time (LT) advantage for each engagement. Lethal time advantage attempts to weigh the "lethality" of each distinct type of weapons lock.

$$LT = \frac{CLAWS\ GT - TDG* GT}{2} + (2 * (CLAWS\ TT - TDG* TT)) + (CLAWS\ AT - TDG* AT)$$

A positive lethal time value shows CLAWS with a lethal time advantage, and a negative lethal time shows TDG* with a advantage.

The fourth metric is Time on Offense (TOF).

$$TOF = \frac{CLAWS\ TOF - TDG* TOF}{2}$$

ΔTOF is computed as CLAWS TOF minus TDG* TOF. As for LT, a positive ΔTOF value shows CLAWS with an time on offense advantage, and a negative ΔTOF shows TDG* with a time on offense advantage.

These statistics are reviewed after each set of runs and the data are used to tune the mode specific scoring weights and test the completeness of the knowledge bases. Although the statistics are helpful no single statistic has been developed that can accurately measure the performance of an aircraft in the engagement. In some test cases an aircraft will score significant amounts of "dead" weapons lock time after it has been "killed." This "dead" weapons lock time can affect the lethal time and time on offense scores.

When a "stable" software configuration is reached, the set of initial conditions is expanded to 320 by modifying the initial separation between

the airplanes, the initial altitudes, and the initial mach numbers. This stepwise refinement process provides the large sets of results required to achieve "global" system improvements across the total air combat environment.

A baseline TDG, TDG*, is currently being tested in the LaRC DMS. TDG* contains a preliminary version of the situation assessment module, a modified version of the throttle control module, and the original set of five to nine trial maneuvers. This reduced set of trial maneuvers is used to insure real-time performance in the DMS. The situation assessment and throttle control modules were modified to increase the efficiency of the FORTRAN version but contain the same basic rules as the KBS versions used by CLAWS.

The development of TDG* has made it possible to evaluate a subset of CLAWS against human pilots in a realistic air combat environment. This capability has allowed experienced pilots to interact with the system and comment on its performance and suggest improvements. The pilots' comments and suggestions are then incorporated in the lab version for testing and refinement before being included in TDG*. To date, CLAWS has outperformed TDG* in the lab² and the TDG* has performed at the same level as the test pilots in the DMS³.

TEST RESULTS

A set of 32 engagements (figure 3) was used to compare the performance of CLAWS with the performance of the TDG* in the TMS. Airplanes

with identical performance characteristics were used for the lab simulation tests. The set of 32 initial conditions is neutral. Fourteen starting positions are neutral; nine sets of initial conditions favor the CLAWS aircraft; and nine favor the TDG* aircraft. There is a 2-nautical mile separation between the opponents and each airplane is at an initial altitude of 15,000 feet and an initial airspeed of 540 knots. All of the engagements were run for a full 90 seconds, even if one aircraft scored a kill in less than 90 seconds. The overall scoring metric used to evaluate the set of engagements was an Overall Exchange Ratio (OER).

$$OER = \frac{TDG^* \text{ Killed}}{CLAWS \text{ Killed}}$$

For the set of 32 test engagements CLAWS achieved nineteen "clean" kills, TDG* achieved one "clean" kill, and there were twelve mutual kills. The overall OER for the set of 32 engagements is 2.38. In the nine runs where CLAWS has a positional advantage the OER is 3.0; in the nine runs where TDG* has a position advantage the OER is 1.33; and in the fourteen neutral case the OER is 3.50. CLAWS earned a lethal time advantage in 29 of the 32 test engagements (shown in figure 4) and a ΔTOF advantage in 29 of the 32 test engagements (shown in figure 5). It is interesting to note that in the three cases where the scoring metrics are negative the final result is a mutual kill. Both lethal time and ΔTOF are positive (LT = 1.83, ΔTOF = 3.50) in the single engagement in which TDG* scores a clean kill (run 16).

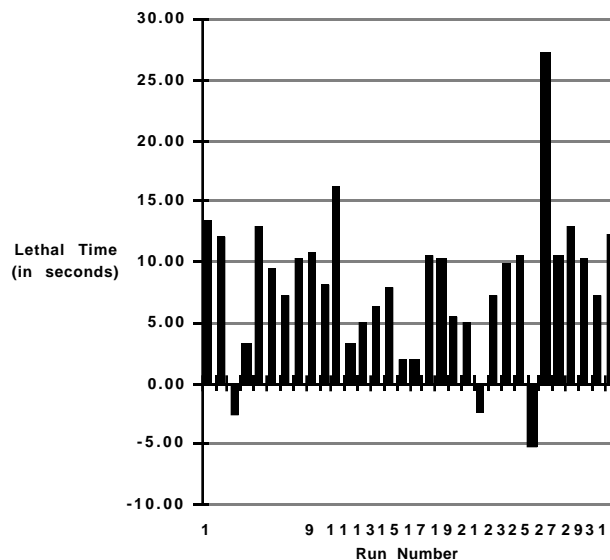


Figure 4. Lethal Time.

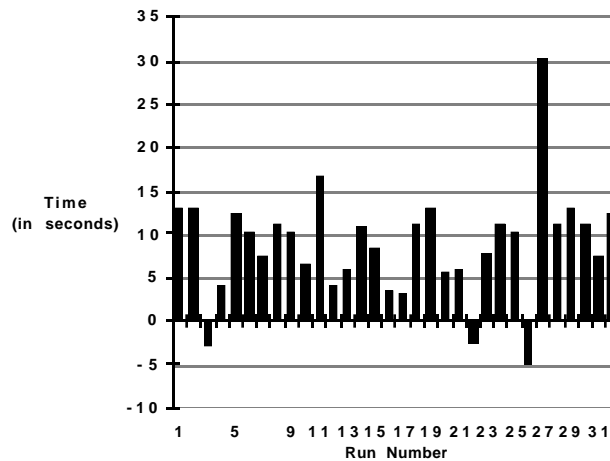


Figure 5. ΔTime on Offense.

Cube_CLAWS

Research is also being conducted to evaluate the use of concurrent programming techniques and specialized parallel processing hardware, such as an Intel Hyper Cube^{†4}, for the development of air combat simulations. The use of parallel processing techniques allows the development of larger and more complete simulations than is currently possible using serial hardware and programming techniques. The Cube_CLAWS is a Concurrent BlackBoard system designed to execute on a 16 processor Intel IPSC HyperCube. The Cube_CLAWS consists of a main Knowledge Source (Main) that contains the blackboard support software and aircraft model and three knowledge sources: a relative geometry knowledge source, a situation assessment knowledge source, and a knowledge source to evaluate prospective aircraft maneuvers (Move Evaluation). The control structure used for activating knowledge source's is message driven and is embedded in the knowledge source's. The blackboard data elements are passed as messages to and from the modules, and read/write synchronization is used to ensure blackboard consistency.

A detailed description of the Cube_CLAWS software and testing procedures can be found in⁴. The Cube_CLAWS software exploits parallelism in two ways. It exploits the natural parallelism of the engagement by creating separate parallel execution paths for each aircraft in the engagement. The main knowledge source for all aircraft in the simulation synchronize at the

start of each time step in the engagement to swap aircraft state data and then proceed down parallel execution paths. The evaluation of trial maneuvers is then performed in parallel. Multiple versions of the situation assessment and relative geometry knowledge sources are loaded onto nodes of the cube and are used to evaluate the candidate maneuvers. The maneuver evaluation knowledge source generates the prospective maneuvers and then sends one maneuver to each available relative geometry knowledge source. When all of the maneuvers have been distributed and processed the results are placed on the blackboard and the results are distributed to the available situation assessment / maneuver scoring knowledge sources. The resulting maneuver scores are then evaluated and the control commands of the highest scoring maneuver are placed on the blackboard for the main knowledge source to execute.

It is important to note that although multiple versions of the relative geometry and situation assessment / maneuver scoring knowledge sources are being executed in parallel there is still an inherent serialization between the two types of modules. The relative geometry must be computed for a maneuver before the situation assessment module can begin execution. Figure 6 is a schematic of the current Cube_CLAWS software configuration.

[†] HyperCube is a registered trademark of Intel Corporation

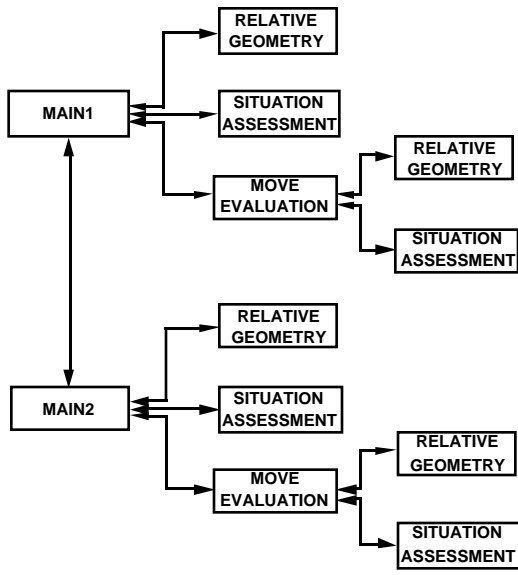


Figure 6. Schematic of Cube_CLAWS

Tests were conducted to measure processor process speedup and process efficiency as additional processors are added to run multiple versions of the maneuver evaluation knowledge source. The speedup and efficiency data for the evaluation knowledge sources is very promising and the overall speedup and efficiency data for the separate processes shows that there is a clear advantage to splitting the aircombat simulation problem in parallel execution paths for each aircraft⁴. Figure 7 shows the execution speedups achieved while testing the Cube_CLAWS software on a 16 processor HyperCube. PID 0 is the main knowledge source for the aggressor aircraft and PID1 is the main knowledge source for the target

aircraft. Execution speedup is almost linear for the one node and two node test cases, but drops off as additional processors are added. A more detailed discussion of the test results can be found in McManus⁴.

The Cube_CLAWS has provided a useful testbed to evaluate the development of a Concurrent Blackboard (CBB) systems. The project has shown that the complexity of developing specialized software on a distributed, message passing architecture such as the Hypercube is not overwhelming and that reasonable speedups and processor efficiency can be achieved by a CBB system. The project has also highlighted some of the costs of using a distributed approach to designing a BlackBoard system. Message passing costs, synchronization costs, and the cost of having multiple processes executing on a single node must be recognized during the system design phase so that their effect on the systems performance can be minimalize.

FUTURE WORK

Several enhancements to the existing CLAWS are planned. The maneuver selection logic will be expanded to replace the use of the trial maneuvers for modes of operation where conventional guidance algorithms provide better performance. This change to the logic and selection module will improve the CLAWS's ability to track its opponent. Initial lab results have shown that the development of mode-specific maneuver sets will increase system efficiency by reducing the number of maneuvers evaluated for some mode of operation's².

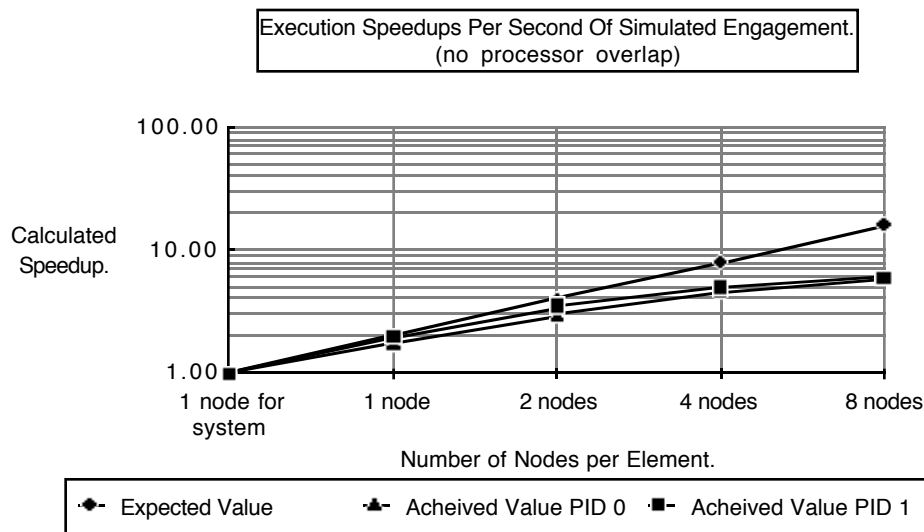


Figure 7. Cube_CLAWS Execution Speedup.

The development of logic for two-versus-one engagements is underway. The third aircraft will be dynamically allocated to either CLAWS or the opponent at the start of each run. This feature will allow researchers to evaluate CLAWS in both two-versus-one and one-versus-two engagements. A system for connecting the Symbolics workstation directly to the DMS real-time computing facilities using a high-speed data / communications network is currently being tested. This link will allow the full LISP based object oriented CLAWS to be tested in the DMS against human pilots.

CONCLUDING REMARKS

A KBS CLAWS is being developed to study WVR air combat engagements. The system incorporates modern airplane simulation techniques, sensors, and weapons systems. The system was developed using several concepts first outlined in the AML program originally developed for use in the LaRC DMS. An updated TDG* system is being used as a baseline to assess the functional and performance tradeoffs between a conventionally coded system and the AI-based system. Test results have shown that the AI-based CLAWS has performed better than TDG* in both the TMS and the DMS and the KBS CLAWS software has proved to be much easier to modify than the TDG* FORTRAN source code. Although software design and maintenance is not the major thrust of this research it is important to realize that approximately seventy percent⁶ of the cost of all software is devoted to maintaining the software after initial development. The TGRES system presents an excellent opportunity to evaluate the use of AI programming techniques and knowledge-based systems in a real-time environment. It also clearly shows that the existing maneuver selection and scoring techniques were not designed to perform in the modern tactical environment and are not suited for evaluating agile aircraft.

The use of KBS and AI programming techniques in developing CLAWS has allowed a complex tactical decision generation system to be developed that addresses the modern combat environment and agile aircraft in a clear and concise manner. The development of a concurrent system has highlighted both the performance gains that can be expected in a concurrent environment and the costs of developing a concurrent system. The ability to integrate CLAWS into the DMS offers a unique opportunity to evaluate the performance of the AI-based CLAWS software in a real-time tactical environment against human pilots.

REFERENCES

1. Goodrich, Kenneth H; McManus John W. : "*Development of A Tactical Guidance Research and Evaluation System (TGRES).*" AIAA Paper #89-3312, August 1989.
2. McManus John W.; Goodrich, Kenneth H. : "*Application of Artificial Intelligence (AI) Programming Techniques to Tactical Guidance for Fighter Aircraft.*" AIAA Paper #89-3525, August 1989.
3. Goodrich, Kenneth H; McManus John W. : "*An Integrated Environment For Tactical Guidance Research and Evaluation.*" AIAA Paper #90-1287 May 1990.
4. McManus, John W. : "*A Parallel Distributed System for Aircraft Tactical Decision Generation.*" Digital Avionics Systems Conference, October 1990.
5. Ni, H. Penny : "BlackBoard Systems: The Blackboard Model of Problem solving and the Evolution Of Blackboard Architectures." *AI Magazine*, 7 (3), 1986, pp. 38 - 53.
6. Meyer, Bertrand. *Object-oriented Software Construction*. Ed. C.A.R. Hoare. Prentice Hall International Ltd, 1988.
7. Burgin, G. H., et al.: *An Adaptive Maneuvering Logic Computer Program for the Simulation of One-on-One Air-to-Air Combat*. Vol I and Vol II. NASA CR-2582, CR-2583, 1975