NASA Technical Memorandum 107722

TRIAL MANEUVER GENERATION AND SELECTION IN THE PALADIN TACTICAL DECISION GENERATION SYSTEM

ALAN R. CHAPPELL
JOHN W. MCMNAUS
KENNETH H. GOODRICH

JANUARY 1993

National Aeronautics and Space Administration

Langley Research Center
Hampton, Virginia 23681-0001

ERRATA

NASA Technical Memorandum 107722

TRIAL MANEUVER GENERATION AND SELECTION IN THE PALADIN TACTICAL DECISION GENERATION SYSTEM

Alan R. Chappell
John W. McManus
Kenneth H. Goodrich

January 1993

The title should be changed from Trail Maneuver... to Trial Maneuver...

A revised cover and Report Documentation Page are attached.

Issued April 1993

TRIAL MANEUVER GENERATION AND SELECTION IN THE PALADIN TACTICAL DECISION GENERATION SYSTEM

Alan R. Chappell*
Lockheed Engineering and Sciences Company
Hampton, Virginia

John W. McManus** and Kenneth H. Goodrich†
NASA Langley Research Center
Hampton, Virginia

Abstract

To date, increased levels of maneuverability and controllability in aircraft have been postulated as tactically advantageous, but little research has studied maneuvers or tactics that make use of these capabilities. In order to help fill this void, a real–time tactical decision generation system for air combat engagements, Paladin, has been developed. Paladin models an air combat engagement as a series of discrete decisions. A detailed description of Paladin's decision making process is presented. This includes the sources of data used, methods of generating reasonable maneuvers for the Paladin aircraft, and selection criteria for choosing the "best" maneuver. Simulation results are presented that show Paladin to be relatively insensitive to errors introduced into the decision process by estimation of future positional and geometric data.

Introduction

Modern air combat simulations must perform in a greatly expanded and rapidly changing tactical environment. Such a simulation system must be able to model new aircraft and their advanced capabilities. The system should have a modular software structure so that new weapons systems or aircraft subsystems (e.g. sensors or propulsion systems), modifications to aircraft control systems, or changes to the aircraft configuration can be easily incorporated. In support of the study of superagile aircraft at the Langley Research Center (LaRC), a Tactical Guidance Research and Evaluation System (TiGRES) is being developed. The design and development of TiGRES as well as its relationship to past and current air combat simulation systems is described in detail in reference 1.

The TiGRES system is designed to allow researchers to develop and evaluate aircraft systems in a tactical environment. The three main components of TiGRES are a Tactical Decision Generator (TDG), the Tactical Maneuver Simulator (TMS), and the Differential Maneuvering Simulator (DMS).

A TDG is an intelligent system that selects the combat maneuvers to perform throughout an air combat engagement. Both the TMS and the DMS use a TDG as the automated opponent. Paladin is the TDG currently used in TiGRES research.

The TMS[2] 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 air combat engagement and the TMS then controls the aircraft using either simple trajectory commands or a tactical decision generation system. The main elements of the TMS are a high-fidelity, nonlinear six degree-of-freedom (d.o.f.) rigid-body aircraft dynamic model, including the control system, a TDG, and a user interface.

The DMS consists of two 40' diameter domes and one 20' diameter dome. 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 a TDG against a human opponent. This feature allows the guidance logic to be evaluated against one or more unpredictable and adaptive human opponents.

The Paladin System

Paladin is a knowledge-based TDG. Knowledge-based systems use a large amount of information about a problem's domain to understand and solve that problem. Paladin was implemented using a large amount of information about aircraft, flight control, and air combat so that the system could provide insight into both the tactical benefits and the costs of enhanced maneuverability.

Paladin uses an object-oriented programming approach[3] to represent each aircraft in the simulation. Each aircraft object includes information on the current state of the aircraft's offensive systems (e. g. guns, missile systems, fire control radars, etc.), defensive systems (e. g. electronic

* Senior Engineer, P.E., Member AIAA.
** Aerospace Engineer.
† Research Engineer, Member AIAA.

counter–measures, chaff, etc.), and propulsion system. This state information is used to help guide Paladin's reasoning process.

Paladin utilizes modular software subroutines and specialized computer hardware. The separation of the aircraft simulation and decision logic components allows each module or 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 sources also provides for modular protection[3], confining the effect of an error occurring in a module at run-time to that module, or to a small set of neighboring modules in the program. The confining effect of the modular protection was used to aid in the design and debugging process of Paladin. Each knowledge source was developed and tested independently before it was incorporated into the system.

The independence of the knowledge sources also increases the efficiency of Paladin by allowing knowledge sources to be distributed across a network of several heterogeneous processors. The network currently consists of a Symbolics 3650‡ workstation, a Symbolics MacIvory◊ workstation, and four Vax 3200¤ class workstations. Communication between the distributed knowledge sources is achieved using customized DECNet–based Client/Server software developed in–house for TiGRES. This software allows for synchronization, communications, and data sharing between heterogeneous computers running the DECNet communications protocol. Paladin is implemented as a serial blackboard system[4], so no serialization or concurrency related software is required. Each knowledge source requests all of the data required to perform its computation from the blackboard at the start of its execution cycle, and posts its results to the blackboard at the end of its execution cycle.

The development of Paladin has been a multi-stage process using a baseline version of the Adaptive Maneuvering Logic[5] (AML) program as the starting point. Figure 1 is a schematic of Paladin. As in AML, Paladin models a combat engagement as a series of discrete decisions. Hence, at temporally regular decision points, the system must choose the "best" tactical maneuver to follow until the next decision point. To make this choice, Paladin uses information about its own state and estimated data about the opponent to calculate the relative geometry between the two aircraft. This relative geometry is used to perform a situation assessment and to choose a new throttle position. After extrapolating the opponent's state a short time into the future, Paladin generates a situationally dependent set of trial maneuvers. A future engagement state is predicted for each of the trial maneuvers. These future engagement states are passed through a group of scoring functions that evaluate various aspects of the tactical situation. The results of the scoring functions are weighted, based on the mode of operation, to compute the current best maneuver. This maneuver is then used to direct the aircraft until the next decision interval. In the following sections, each of these steps will be described in detail.

‡ Symbolics 3650 is a registered trademark of Symbolics Incorporated.

◊ MacIvory is a registered trademark of Symbolics Incorporated.

¤ Vax 3200 & DECNet are registered trademarks of Digital Equipment Corporation.

Estimation of Opponent Information

To begin each decision cycle, Paladin must obtain certain information that will be needed throughout the decision making process. Data about the ownship (Paladin's aircraft) is readily available. However, information about the opponent would be less available outside the simulation environment (in a real aircraft). Paladin, thus, receives only the opponent's absolute X, Y, and Z coordinates and infers all other needed data. No noise is added to this position input. Previous research[6] has shown that realistic noise on the input has negligible impact on Paladin's capabilities, while it impairs repeatability and causal assessment.

Using the three most recent positions for the opponent, a quadratic curve fit is made for the flight path. From this flight path, a velocity and load factor are estimated. Assuming that the opponent's aircraft is aerodynamically similar to Paladin, all other needed data are estimated. Using these estimations and Paladin's known data, relative geometry parameters are calculated for use by the Situation Assessment Module and the Active Throttle Controller.

Situation Assessment Module

Six modes of operation, shown in table 1, have been incorporated in Paladin. The Situation Assessment knowledge source is used to model a pilot's situational awareness and changing problem–solving strategies. Just as a pilot will recognize the difference between an aggressive and an evasive situation and react accordingly, the Situation Assessment knowledge source provides information allowing Paladin to adapt its problem–solving strategy based on the current situation. The determination of the current mode of operation is based on the aircraft's current mission, the current state of the aircraft's systems, and the relative geometry between the aircraft and its opponent. Each of the six modes of operations has a unique vector of scoring weights and a unique decision interval (shown in table 1). The scoring weights[6] for each mode of operation

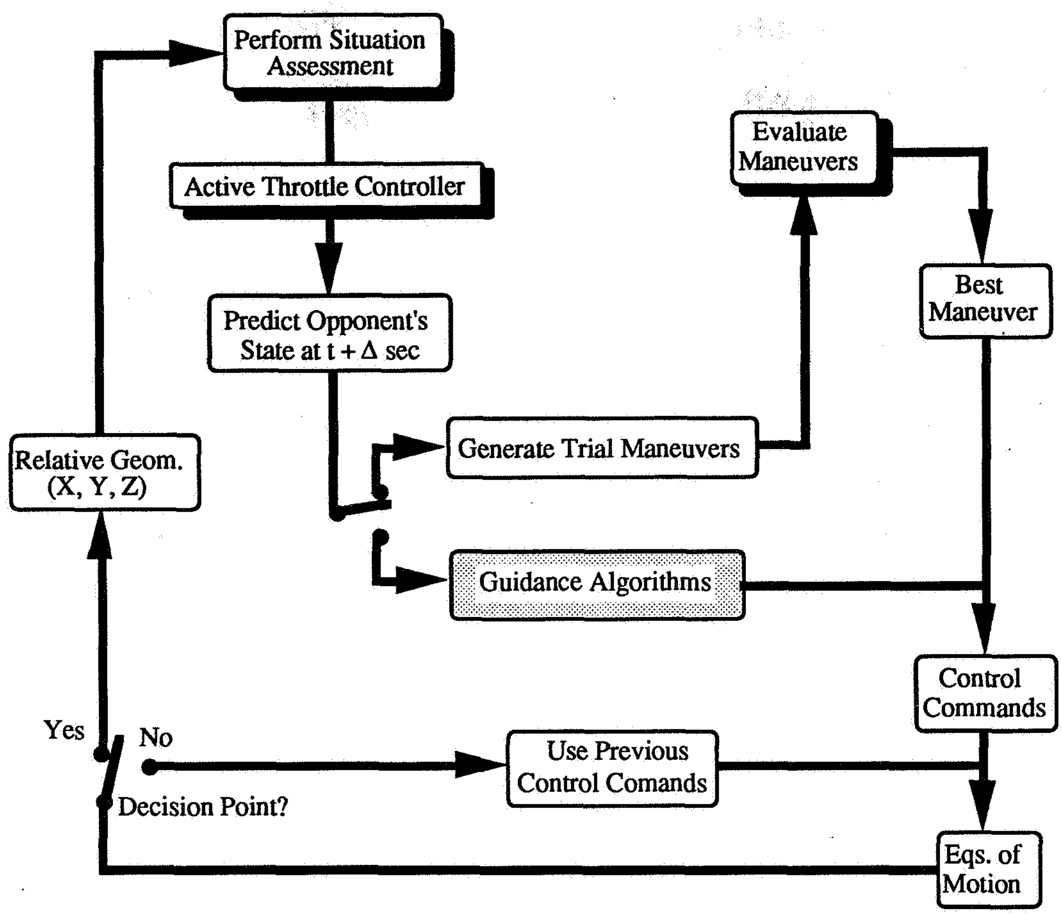

Figure 1. Schematic of The Paladin System

have been adjusted during the design and testing process to maximize Paladin's performance in that mode of operation. The testing procedures used to evaluate Paladin's performance are described in detail later in this paper. The Situation Assessment used by Paladin is covered more completely in reference 7.

Active Throttle Controller

A rule-based Active Throttle Controller was developed to determine throttle and speed brake settings based on the engagement situation. The throttle controller can set the throttle to any position between idle and full afterburner, and the speed brake to any position between fully retracted and fully extended. The throttle controller uses the current mode of operation and the relative geometry information to select one of four operational modes. These four modes are maintain best cornering speed mode, maintain/set range mode, separate quickly mode, and force overshoot mode. Each mode has a set of specific throttle control rules that are used to maximize system performance in that throttle mode. The throttle position produced is considered a suggested throttle position, as maneuver generation can accept this position or select another in order to coordinate the throttle with the maneuver. The Active Throttle Controller is covered in detail by reference 7.

Table 1. Modes of Operation

Mode	Decision Interval
Aggressive	0.25 sec
Defensive	0.5 sec
Evasive	0.25 sec
Ground Avoidance	0.125 sec
Neutral	1.0 sec
Disengage	0.5 sec

Extrapolation of Opponent's Near Future

Prior to deciding what maneuver to execute, Paladin estimates where the opponent will be at the end of the look-ahead period, assuming no change in the opponent's current maneuver. Currently, a look-ahead period of 4 times the decision cycle is used to enhance system performance[7]. This estimation of future state is necessary to guide the trial maneuver generation process and for selecting the best maneuver. Using the quadratic curve fit for the flight path of the opponent established by the estimation process (discussed earlier), a position, velocity, and load factor are extrapolated. Again assuming the opponent's aircraft is aerodynamically equivalent to Paladin's, all remaining data about the state of the opponent's aircraft is then calculated. This information will be used to calculate a future relative geometry between the aircraft for use in scoring Paladin's trial maneuvers.

However, the opponent can and does change his current maneuver. Extrapolation into the future, therefore, introduces a certain amount of error into the ensuing calculations. The magnitude of these errors and their effects on Paladin's capabilities are discussed later in this paper.

Trial Maneuver Generation Module

At the end of each decision interval Paladin must choose a maneuver to execute for the duration of the next decision interval. This is accomplished by generating a set of trial maneuvers, predicting the engagement state to some point in the future based on each of these trial maneuvers, scoring the future engagement states, and finally, selecting the maneuver with the highest score.

Trial maneuver generation uses several maneuvering concepts throughout the generation process that bear advanced definition. The maneuver plane is the plane containing the current velocity vector and the net force vector on the aircraft. Hence, this is the plane which will contain the flight path for some limited period of time. The bank to intercept[3] is the rotation angle around the velocity vector that places the opponent in the maneuver plane and above the maneuvering aircraft. The optimum cornering speed is the speed at which the aircraft has its fastest sustained turn rate.

Paladin generates a set of trial maneuvers completely dependent upon the current engagement situation. Using the situational information, Paladin selects one of five maneuver generation schemes. These schemes are fine tracking, over-the-top reversal, high-speed turning, low-speed turning, and target acquisition.

Fine Tracking

Fine tracking maneuvers are used when Paladin is close to, or has, a weapons solution. These maneuvers are clustered tightly around the bank to intercept and use a small fraction of the available maximum load factor. However, as range decreases, it may be necessary to turn sharply to track an active target. Therefore, one group of maximum load factor maneuvers is generated. The trial maneuvers generated in fine tracking are the 28 combinations of intercept bank angle and intercept bank angle ±2.5°, ±5.0°, ±7.5° at loads of 10%, 20%, 40%, and 100% of the maximum available load.

Over-the-Top Reversal

Over-the-top reversal maneuvers are used in a limited number of situations. If Paladin's velocity is above the current calculated corner velocity and either, the two aircraft are headed in nearly opposite directions, or the opponent is following nearly directly behind Paladin, then the over-the-top reversal maneuvers are generated. Only one maneuver is generated in these situations, a maximum load factor pull up with a bank angle of 180.0° minus the current azimuth angle. Given the situations in which this maneuver is generated, the result is a near vertical pull up maintained as long as the conditions stated above hold. Acquisition maneuvers will then take over to allow Paladin to roll out of the loop and pursue the opponent.

High-Speed Turning

High-speed turning maneuvers are used in cases similar to those of the over-the-top reversal, allowing larger directional differences between the aircraft. In these cases, where Paladin has a long way to turn and is above corner velocity, a high oblique turn is the best choice for several reasons. Primarily, the vertical component of the turn drops Paladin's velocity to near corner much faster than a flat turn (resulting in a speed through the turn which is closer to the optimum cornering speed). As an added benefit, the excess speed is traded for altitude, which can be used later to recover velocity lost in a turning contest. High-speed turning maneuvers are generated as maximum load factor turns between 0° and 45° off vertical (in 5° increments). Only turns in the direction of the opponent are generated, limiting these situations to ten trial maneuvers.

Low-Speed Turning

Low-speed turning maneuvers are used when Paladin has a long way to turn and is below corner velocity. These maneuvers force a diving turn where the vertical component of the turn raises/maintains Paladin's velocity nearer to corner than would a flat turn (resulting in a speed through the turn which is closer to the optimum cornering speed). Low-speed turning maneuvers are generated as maximum available load factor turns at bank angles between 135° and 180° (in 5° increments). Only turns in the direction of the opponent are generated, limiting these situations to ten trial maneuvers.

Target Acquisition

In the largest part of typical engagements none of the limited situations described for the generation schemes above holds. When this happens, target acquisition maneuvers are generated. Since target acquisition maneuvers are used in widely varying engagement states, some restrictions on the generation have been developed to ensure that the trial maneuvers do not violate the Paladin maneuvering assumptions and are physically realistic. The development of these limits will now be described.

In order to choose a maneuver for the next decision interval Paladin evaluates the position achieved by the trial maneuvers with respect to the extrapolated position of the opponent. This choice will be impaired by errors in the positions being evaluated. Moreover, maneuvers that consistently produce overly optimistic or pessimistic results will tend to bias the selection process away from the best results. It is, therefore, necessary to restrict maneuvers to those which produce flight paths that do not violate the basic assumptions of the prediction algorithm and are inside the aircraft's maneuvering capabilities.

Maintainable Limits

The prediction algorithm used by Paladin assumes that the flight path will be in a selected maneuver plane with a net positive load factor. These assumptions are met by only generating maneuvers which balance all out of the maneuver plane forces and yield a net positive vertical force in the maneuver plane. These limits on maneuvers are called the maintainability limits. (Since the limits represent assumptions made by the prediction algorithm and the prediction algorithm ignores the affect of thrust, the equations presented below do not include the thrust component.)

A balance of out of the maneuver plane forces can be stated as follows:

$$W \cos \bar{\theta} \sin \rho + L \sin \Phi^* = 0 , \quad (1a)$$

where W is the aircraft weight, L is the aircraft lift, $\bar{\theta}$ is the pitch angle of the maneuver plane, ρ is the rotation angle of the maneuver plane, and Φ^* is the offset between ρ and the wind axis bank angle (Φ). Figure 2 gives a graphic representation of these angles. Solving for Φ^* yields,

$$\Phi^* = \arcsin \left(- \frac{\cos \bar{\theta} \sin \rho}{n} \right) , \quad (1b)$$

where n is the load factor, or L / W. For this relation to hold, the magnitude of the argument to the arcsine function must be less than or equal to one. Hence,

$$-1 \leq - \frac{\cos \bar{\theta} \sin \rho}{n} \leq 1 . \quad (1c)$$

Noting that $\cos \bar{\theta} = V_H / V$, horizontal velocity over total velocity, and that this quantity is always positive,

$$- \frac{n V}{V_H} \leq \sin \rho \leq \frac{n V}{V_H} , \quad (1d)$$

which yields boundaries at angles of $\pm \arcsin (n V / V_H)$ and $180° - [\pm \arcsin (n V / V_H)]$. As would be expected, these limiting equations indicate that the assumption stated in equation 1a will always be met using load levels greater than one gravity (g). The limits establish two regions, symmetric around maneuver planes of ±90°, where this assumption will not be met with load levels below one g. These regions are shaded in medium grey in figure 3.

A positive net vertical force in the maneuver plane can be stated as follows:

$$L \cos \Phi^* - W \cos \bar{\theta} \cos \rho > 0 . \quad (2a)$$

This reduces to

$$\cos \rho < n V \cos \Phi^* / V_H \quad (2b)$$

which establishes a symmetric region around the 0° maneuver plane with limits at $\pm \arccos (n V \cos \Phi^* / V_H)$ where relation 2a will not be satisfied. This region is shaded in light grey in figure 3.

Achievable Limits

Likewise, a constraint is placed on the bank angles that can be physically achieved by the Paladin aircraft. Modeling the current wind axis roll rate as a limited, first-

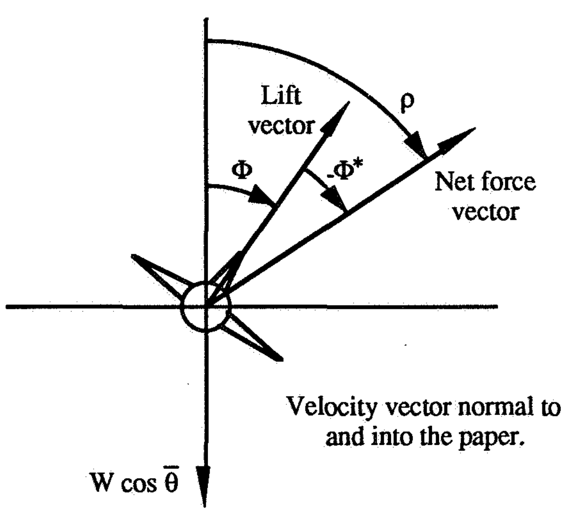

Figure 2. Definition of Angular Measurements

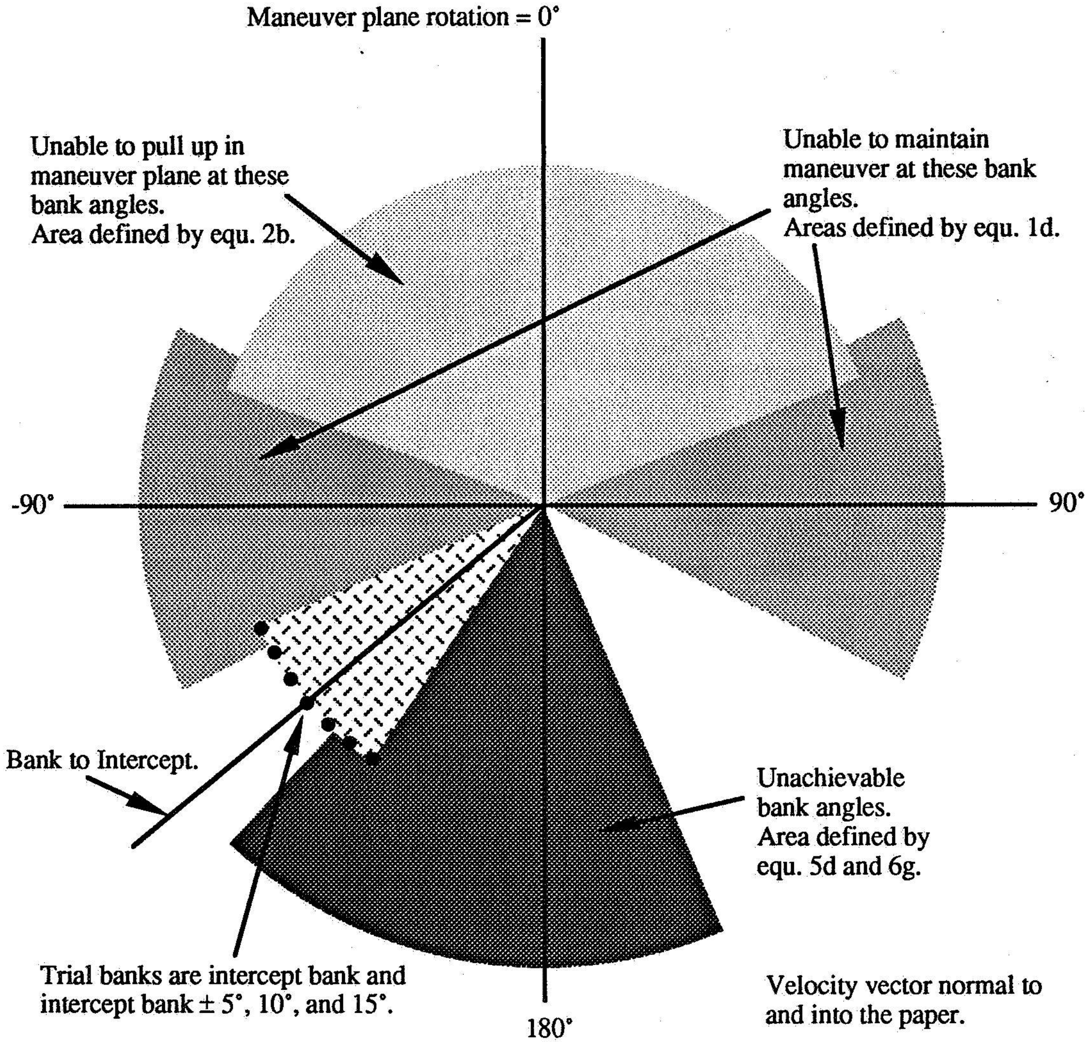

Figure 3. Trial Bank Elimination at a Given Load Level

order lag response, the maximum roll rate in one direction at any time in the future can be expressed as follows:

$$\dot{\Phi}(t) = (\dot{\Phi}_{max} - \dot{\Phi}_0)(1 - e^{\frac{-t}{\tau}}) + \dot{\Phi}_0 \qquad (3)$$

where $\dot{\Phi}_{max}$ is the maximum roll rate (+ or - depending on direction of roll to be limited), $\dot{\Phi}_0$ is the current roll rate, t is the elapsed time from now, and τ is the lag time constant. $\dot{\Phi}_{max}$ and τ are interpolated from data tables based on commanded angle of attack (α). These data tables represent the approximate response of the 6 d.o.f. aircraft model. Figures 4 (baseline aircraft) and 5 (thrust vectored aircraft) show the $\dot{\Phi}_{max}(\alpha)$ and $\tau(\alpha)$ relationships. $\dot{\Phi}_{max}$ is further limited based on airspeed such that,

$$\dot{\Phi}_{max} \leq 0.9375 \text{ V} \qquad (4)$$

for speeds less than 160 ft/sec.

Integrating $\dot{\Phi}(t)$ over the prediction interval yields the maximum possible change in roll during this time period. Hence,

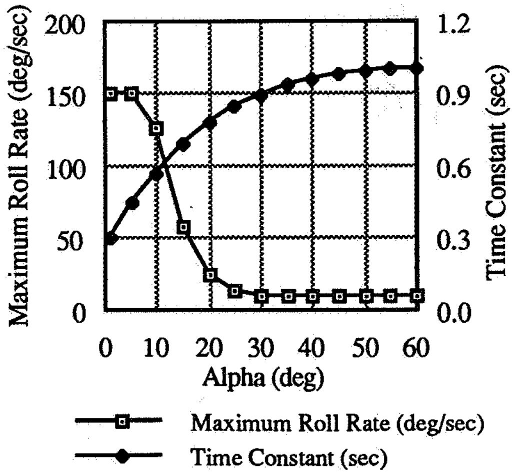

Figure 4. $\dot{\Phi}_{max}(\alpha)$ and $\tau(\alpha)$ for the Baseline Aircraft

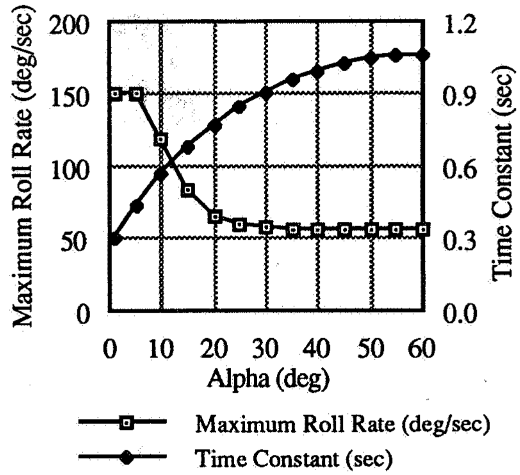

Figure 5. $\dot{\Phi}_{max}(\alpha)$ and $\tau(\alpha)$ for the Thrust Vectored Aircraft

$$\Delta \text{ roll} = \int_0^{\Delta t} \left[(\dot{\Phi}_{max} - \dot{\Phi}_0)(1 - e^{\frac{-t}{\tau}}) + \dot{\Phi}_0\right] dt \tag{5a}$$

$$\Delta \text{ roll} = \int_0^{\Delta t} (\dot{\Phi}_{max} - \dot{\Phi}_0)\, dt - \int_0^{\Delta t} (\dot{\Phi}_{max} - \dot{\Phi}_0)(e^{\frac{-t}{\tau}})\, dt + \int_0^{\Delta t} \dot{\Phi}_0\, dt \tag{5b}$$

$$\Delta \text{ roll} = (\dot{\Phi}_{max} - \dot{\Phi}_0)\Delta t - (\dot{\Phi}_{max} - \dot{\Phi}_0)(-\tau)(e^{\frac{-\Delta t}{\tau}} - e^0) + \dot{\Phi}_0 \Delta t \tag{5c}$$

$$\Delta \text{ roll} = \dot{\Phi}_{max}\, \Delta t + \tau(\dot{\Phi}_{max} - \dot{\Phi}_0)(e^{\frac{-\Delta t}{\tau}} - 1) \tag{5d}$$

where Δt is the length of the prediction interval.

Therefore, we have limits on how far the aircraft can physically roll in the wind axis system. In order to establish the limits on the achievable maneuver planes, a rotation angle ρ must be found that corresponds to each of these physical limits. Restating equation 1a, including a term for thrust (T),

$$W \cos \bar{\theta} \sin \rho + (L + T \sin \alpha) \sin \Phi^* = 0\,. \tag{6a}$$

Solving for $\sin \Phi^*$, and substituting $\Phi - \rho$ for Φ^*, yields:

$$\sin(\Phi - \rho) = \left(- \frac{W \cos \bar{\theta} \sin \rho}{L + T \sin \alpha}\right). \tag{6b}$$

A formula for ρ can then be derived as follows:

$$\sin \Phi \cos \rho - \cos \Phi \sin \rho = \left(- \frac{W \cos \bar{\theta} \sin \rho}{L + T \sin \alpha}\right) \tag{6c}$$

$$\frac{\sin \Phi \cos \rho}{\sin \rho} - \cos \Phi = \left(- \frac{W \cos \bar{\theta}}{L + T \sin \alpha}\right) \tag{6d}$$

$$\frac{\cos \rho}{\sin \rho} = \frac{\cos \Phi}{\sin \Phi} - \frac{W \cos \bar{\theta}}{(L + T \sin \alpha) \sin \Phi} \tag{6e}$$

$$\frac{\sin \rho}{\cos \rho} = \frac{(L + T \sin \alpha) \sin \Phi}{(L + T \sin \alpha) \cos \Phi - W \cos \bar{\theta}} \tag{6f}$$

$$\rho = \arctan\left[\frac{(L + T \sin \alpha) \sin \Phi}{(L + T \sin \alpha) \cos \Phi - W \cos \bar{\theta}}\right]. \tag{6g}$$

Using the maximum wind axis roll in both directions as calculated in the previous paragraph, maneuver plane rotations can be found which together define an area that is unachievable. This area is shaded in dark grey in figure 3 and is called the unachievable area.

In order to implement these restrictions on trial maneuver bank angle, a set of trial maneuvers that is appropriate to the situation is identified for testing. This set of trial maneuvers is defined by the intercept bank angle and intercept bank angle ±5°, ±10°, and ±15° at loads of 100%, 80%, 60%, 40%, 20%, and 1% of the current maximum load factor (42 possible combinations). A predefined maximum number of bins are available for storing trial maneuvers. A combination of load factor and bank angle is selected from the above lists as a candidate trial maneuver. If the trial maneuver falls in any of the unusable areas described above, it is discarded. Otherwise, the trial maneuver is placed in one of the bins to be sent to the scoring routine. Once all of the bins are filled, or all combinations are tested, trial maneuver generation is terminated.

At any particular decision point, acquisition maneuver generation proceeds as follows:

> Choose the maximum available load factor. Calculate the resulting alpha. Generate the unmaintainable and the unachievable areas for this load factor. Test the trial bank angles, starting at intercept bank (IB) and working outward (IB, IB+5°, IB-5°, IB+10°, ...) discarding those in the previously calculated areas. Save the remaining trial maneuvers.
>
> Choose 80% of the maximum available load factor. Calculate new areas of unmaintainable and unachievable bank angles. As load factor decreases, the unmaintainable areas will grow while the unachievable areas will diminish. Save only the valid trial maneuvers.
>
> Choose 60% of the maximum available load factor and proceed as before. Continue this cycle until all trial bins have been filled or all load factors have been exhausted.

Therefore, the highest load factor trial maneuvers for which outcomes can be accurately predicted will be generated, while physically impossible trial maneuvers are prevented from being chosen over realistic trial maneuvers which would yield lower angles of attack. These restrictions help Paladin make more informed decisions.

Ground Avoidance

Each of the maneuver generation schemes uses the same ground avoidance logic. Regardless of an engagement history, any aircraft that impacts the ground has lost the encounter. Considering that most lengthy engagements end up at low altitudes, as the participants trade potential energy to recover speed, ground avoidance is a significant concern throughout Paladin.

One of the most important aspects of ground avoidance is the calculation of the dive recovery angle. This value represents the maximum dive angle from which recovery is possible with an immediate pull-up given the current altitude, Mach number, and bank angle. Since the bank angle adds more complexity to the problem than one table look-up can reasonably represent, the calculation is broken into two parts (roll level then pull up). First, the time necessary to roll level is calculated, assuming roll rates corresponding to an alpha of the lesser of 10° or the current value. Given this time, an approximate altitude at the pull-up point can be predicted from the current altitude and the rate of change in the altitude. Then, a table look-up is performed based on the current Mach number and the predicted altitude, yielding the dive recovery angle. (This table was created from data gathered from the aerodynamics of the 6 d.o.f. simulation.) Using this dive recovery angle, ground avoidance trial maneuvers are generated.

In order to maintain maneuverability at low altitudes, trial maneuver generation must, in addition to providing for a pull up when necessary, also give the option of normal maneuvers as dictated by the engagement situation. The trial maneuver generation algorithm, therefore, is part of each of the five maneuver generation schemes discussed previously. In addition to normal trial maneuvers, when the altitude has fallen below 5000 feet, or the current relative $\dot{\theta}w$ (the rate of change in the wind axis pitch minus the rate of change in dive recovery angle) would put the dive angle above the dive recovery angle within 4 decision cycles, or the dive angle is currently above the dive recovery angle, three extra maneuvers are generated for a ground avoidance pull up. These maneuvers are in maneuver planes with rotation angles of 0° and ±10° and use a throttle setting of 0 (idle) if the Mach number is above 0.3 (speed for minimum turn radius) or the throttle setting provided by the throttle controller otherwise. If any of these maneuver planes require the aircraft to roll more than 15°, commanded alpha is limited to a maximum of 10° for that trial maneuver, thus avoiding the slow roll rates encountered at higher alpha. Otherwise, the alpha corresponding to maximum available lift becomes the trial alpha.

Together, the generation algorithms and the ground avoidance strategies produce a set of trial maneuvers dictated by the physical and tactical situation. Paladin must then choose one of these trial maneuvers to execute for the upcoming period between decisions.

Prediction of Paladin's Near Future

In order to evaluate a trial maneuver, the resulting aircraft state must be determined. Unfortunately, integrating the full 6 d.o.f. model forward for each maneuver is too time consuming for real-time application. It is, therefore, necessary to formulate a simplified prediction for this resulting state. To do this, Paladin makes certain assumptions about the attempted maneuver and its resulting flight path. For the look-ahead interval (4 times the decision cycle) the following are assumed:

1) the resulting flight path falls entirely in the commanded maneuver plane,

2) the flight path is a section of a circle (constant radius turn), and

3) the total velocity is constant through this segment of the turn.

With these simplifications, a final position, velocity direction, and attitude can be quickly calculated. Like extrapolation, prediction will introduce some error into the evaluation process. The magnitude of these errors and their effects on Paladin's capabilities are discussed later in this paper.

Maneuver Scoring Module

The Paladin Maneuver Scoring Module knowledge source uses a set of fuzzy logic questions[1] with responses ranging from [-1.0 = Negative, ..., 0.0 = Neutral, ..., 1.0 = Positive] and the mode-specific scoring weight vector selected by the Situation Assessment Module to score each of the trial maneuvers. Each scoring question is intended to encourage or discourage some quantifiable aspect of the tactical / geometric situation being evaluated.

Using the data extrapolated for the opponent's future state and the data predicted for Paladin's future state resulting from one of the trial maneuvers, the relative geometry between the future positions of the two aircraft is calculated. The score for the maneuver is determined by computing the responses to the seventeen fuzzy logic questions given this relative geometry, applying the selected scoring weight vector, and then summing the results to generate a single numeric score. After all of the trial maneuvers have been evaluated, the highest scoring maneuver is selected and the associated maneuver is commanded.

Data Error Evaluation

As discussed earlier, extrapolation and prediction introduce error into the maneuver selection process. Some amount of error is inevitable since Paladin does not know what the opponent will do (or is doing) and since the computation time is not available to fully integrate Paladin's future options using a high fidelity model. Table 2 gives the magnitude of these errors as they affect the inputs to trial maneuver generation and selection.

The data shown in table 2 represent a set of 32 batch simulations between Paladin and an equivalent opponent, containing 10773 decision points. Values give are the Mean and Root Mean Squared of the error (actual value minus estimated or predicted value) in each of the parameters listed. From these numbers, it is evident that Paladin has reliable information about both aircraft's future positions, velocities, and deviation angles (based only on position and velocity). However, those values that depend on orientation (line-of-sight, angle-off, azimuth, and elevation) are significantly less reliable. (As would be expected, Paladin can predict its own orientation better than that of the opponent.) Given only these results, it is difficult to assess the impact of the errors, as the relative importance of these data as well as the sensitivity of the selection process will heavily affect the results.

To see the full effect of the estimation and prediction errors, it is necessary to find the ways that these errors change the decision process. For the same set of 32 runs used for table 2, the following effects were accumulated:

Table 2. Errors Due to Extrapolation and Prediction

Error in:	Mean	RMS
Range (ft.)	0.028	1.792
Range Rate (ft./sec.)	0.298	15.279
Paladin's Z coordinate (ft.)	0.118	1.354
Paladin's Dive Angle (deg.)	0.016	1.110
Paladin's Dive Recovery Angle (deg.)	-0.055	1.341
Paladin's Line-of-Sight Angle (deg.)	0.999	7.259
Paladin's Deviation Angle (deg.)	0.167	1.100
Paladin's Angle-Off (deg.)	0.474	11.239
Paladin's Azimuth Angle (deg.)	-0.223	23.845
Paladin's Elevation Angle (deg.)	-0.205	14.597
Opponent's Deviation Angle (deg.)	-0.010	1.033
Opponent's Line-of-Sight Angle (deg.)	-0.474	11.239
Opponent's Angle-Off (deg.)	-0.999	7.259
Opponent's Azimuth Angle (deg.)	-0.473	26.330
Opponent's Elevation Angle (deg.)	3.832	18.400

- Paladin chose the wrong maneuver generation algorithm in 0.0362% of the decisions (where wrong means different from the choice that would have been made with perfect information),

- Paladin chose the wrong trial maneuver due to extrapolation error in 6.823% of the decisions, which lead to a projected scoring loss of $7.253*10^{-7}$ out of an average score of 6.248 on those decisions,

- over all the decisions, the mean scoring error due to extrapolation errors was 0.099 with an RMS value of 0.766 out of an actual score with mean 1.422 and RMS 11.375,

- over all the decisions, the mean total scoring error due to extrapolation and prediction errors was 0.593 with an RMS value of 1.960 out of an actual score with mean 1.422 and RMS 11.375.

It can be seen, therefore, that the extrapolation and prediction errors do have an effect on Paladin's decision making. Mistakes are made in choosing the maneuver generation algorithm, as well as the "best" trial maneuver. Nonetheless, these effects are both infrequent and relatively small in resulting loss of capability (as measured by maneuver score).

Paladin Testing Procedures

Paladin is currently being tested in the TMS using 6 d.o.f. aircraft dynamics, and in the DMS using 5 d.o.f. aircraft dynamics[1,6]. TMS testing is done in a non-real-time, batch mode environment against a baseline TDG. A group of test conditions consists of 32 sets of initial aircraft conditions. The initial altitudes, airspeeds, and the separations between the two aircraft are adjusted to provide representative coverage of the within–visual–range air combat arena.

A set of engagement scoring metrics[1] is reviewed after each group of test runs and the data are used to tune the mode specific scoring weights and test the completeness of the knowledge bases. Although the metrics are helpful, no single metric has been developed that can completely measure the performance of an aircraft in the engagement.

After initial adjustment of the scoring weights, the set of initial conditions is expanded to 320 initial conditions by modifying the initial separation between the airplanes, the initial altitudes, and the initial Mach numbers. This stepwise refinement process provides the large set of results required to achieve global system improvements across the total within–visual–range air combat environment.

A baseline version of Paladin is currently being tested in the DMS using a 5 d.o.f. aircraft model. The aircraft model lacks both the extra degree of freedom (lateral motion in body axes) as well as an accurate representation of the aircraft's rotational dynamics throughout the complete flight envelope. This predecessor to the Paladin system, the Computerized Logic for Air Warfare Simulation (CLAWS) contains the situation assessment module, the active throttle controller, and a similar set of situationally dependent trial maneuvers.

The development of CLAWS made possible the evaluation of the tactical decision generation software against human pilots in the DMS. This capability has allowed experienced pilots to interact with the system in a realistic air combat environment, comment on its performance, and suggest improvements. The pilots' comments and suggestions are then the basis for changing the TMS experimental version of Paladin. These changes are tested and refined before being included in the baseline system. In order to extend this valuable interaction, Paladin and the 6 d.o.f. model will shortly be implemented in the DMS.

Conclusions

Paladin, a computerized air combat tactical decision generator, has been developed to study air combat engagements. The system incorporates modern aircraft simulation techniques, sensors, and weapons systems. Paladin uses knowledge–based systems to address air–to–air combat and agile aircraft in a clear and concise manner. The Differential Maneuvering Simulator offers a unique opportunity to evaluate the performance of the Paladin software in a real-time tactical environment against human pilots.

Paladin models aspects of the decision–making processes used by human pilots through the generation of situationally dependent sets of trial maneuvers and the mode sensitive scoring of their predicted outcomes. The use of distinct modes of operation allows Paladin to perform complex air–to–air combat tasks and generate sound tactical decisions in real–time. Without the situationally dependent mode selection, Paladin would be forced to either sacrifice its real-time execution or assume an unrealistic tactical mind-set for the duration of the engagement.

Although predicting information about the outcome of a maneuver introduces errors into the decision making process, and these errors do interfere with the selection of the desired course of action, they do not greatly affect the overall capabilities of Paladin. Any errors that would greatly change the decision outcome are naturally discounted, as the tuning process will place less importance on unreliable data. However, as a result, Paladin also will be more sensitive to new errors in previously reliable data. Retuning is, hence, important for Paladin whenever a data item is changed or added.

References

1. Goodrich, Kenneth H.; John W. McManus: *"An Integrated Environment For Tactical Guidance Research and Evaluation."* AIAA Paper #90-1287 May 1990.

2. Goodrich, Kenneth H.; Dr. John W. McManus; Alan R. Chappell: *"A High-Fidelity Batch Simulation Environment for Integrated Batch and Piloted Air Combat Simulation Analysis"* AIAA Paper #92-4145, August 1992.

3. Meyer, Bertrand. *Object-oriented Software Construction.* Ed. C.A.R. Hoare. Prentice Hall International Ltd, 1988.

4. McManus, John W.: "A Parallel Distributed System for Aircraft Tactical Decision Generation" In *Proceedings of the 9th Digital Avionics Systems Conference*, 1990, pp. 505 – 512.

5. 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.

6. McManus, John W.; Kenneth H. Goodrich: *"Application of Artificial Intelligence (AI) Programming Techniques to Tactical Guidance for Fighter Aircraft."* AIAA Paper #89-3525, August 1989.

7. McManus, John W.; Alan R. Chappell; P. Douglas Arbuckle: "Situation Assessment in the Paladin Tactical Decision Generation System." *AGARD Conference Proceedings 504; Air Vehicle Mission Control and Management,* March 1992.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	January 1993	Technical Memorandum

4. TITLE AND SUBTITLE
Trail Maneuver Generation and Selection in the Paladin Tactical Decision Generation System

5. FUNDING NUMBERS
505-64-30-01

6. AUTHOR(S)
Alan R. Chappell, John W. McManus, and Kenneth H. Goodrich

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
NASA Langley Research Center
Hamtpon, VA 23681-0001

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)
National Aeronautics and Space Administration
Washington, DC 20546-0001

10. SPONSORING / MONITORING AGENCY REPORT NUMBER
NASA TM-107722

11. SUPPLEMENTARY NOTES
Chappell: Lockheed Engineering and Sciences Company, Hampton, Virginia
McManus and Goodrich: Langley Research Center, Hampton, Virginia

12a. DISTRIBUTION / AVAILABILITY STATEMENT
Unclassified-Unlimited

Subject Category 08

12b. DISTRIBUTION CODE

13. ABSTRACT ***(Maximum 200 words)***
To date, increased levels of maneuverability and controllability in aircraft have been postulated as tactically advantageous, but little research has studied maneuvers or tactics that make use of these capabilities. In order to help fill this void, a real-time tactical decision generation system for air combat engagements, Paladin, has been developed. Paladin models an air combat engagement as a series of discrete decisions. A detailed description of Paladin's decision making process is presented. This includes the sources of data used, methods of generating reasonable maneuvers for the Paladin aircraft, and selection criteria for choosing the "best" maneuver. Simulation testing results are presented that show Paladin to be relatively insensitive to errors introduced into the decision process by estimation of future positional and geometric data.

14. SUBJECT TERMS
Air Combat Simulation, Tactical Maneuver Generation, Estimation Error, Distributed Computing, Situation Assessment, and Aircraft Maneuver Limitations

15. NUMBER OF PAGES
12

16. PRICE CODE
A03

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. Z39-18
298-102