AIRCRAFT CONFLICT ANALYSIS AND REAL-TIME CONFLICT PROBING USING PROBABILISTIC TRAJECTORY MODELING

Lee C. Yang and James K. Kuchar

Report No. ICAT-2000-2
May 2000

MIT International Center for Air Transportation
Department of Aeronautics & Astronautics
Massachusetts Institute of Technology
Cambridge, MA 02139 USA

AIRCRAFT CONFLICT ANALYSIS AND REAL-TIME CONFLICT PROBING USING PROBABILISTIC TRAJECTORY MODELING

by

Lee C. Yang

James K. Kuchar

Department of Aeronautics and Astronautics

Massachusetts Institute of Technology

Cambridge, MA 02139 USA

May, 2000

ICAT-2000-2

AIRCRAFT CONFLICT ANALYSIS AND REAL-TIME CONFLICT PROBING USING PROBABILISTIC TRAJECTORY MODELING

by

Lee C. Yang

James K. Kuchar

ABSTRACT

Methods for maintaining separation between aircraft in the current airspace system have been built from a foundation of structured routes and evolved procedures. However, as the airspace becomes more congested and the chance of failures or operational error become more problematic, automated conflict alerting systems have been proposed to help provide decision support and to serve as traffic monitoring aids.

The problem of conflict detection and resolution has been tackled from a number of different ways, but in this thesis, it is recast as a problem of prediction in the presence of uncertainties. Much of the focus is concentrated on the errors and uncertainties from the working trajectory model used to estimate future aircraft positions. The more accurate the prediction, the more likely an ideal (no false alarms, no missed detections) alerting system can be designed.

Additional insights into the problem were brought forth by a review of current operational and developmental approaches found in the literature. An iterative, trial and error approach to threshold design was identified. When examined from a probabilistic perspective, the threshold parameters were found to be a surrogate to probabilistic performance measures. To overcome the limitations in the current iterative design method, a new direct approach is presented where the performance measures are directly computed and used to perform the alerting decisions.

The methodology is shown to handle complex encounter situations (3-D, multi-aircraft, multi-intent, with uncertainties) with relative ease. Utilizing a Monte Carlo approach, a method was devised to perform the probabilistic computations in near real-time. Not only does this greatly increase the method's potential as an analytical tool, but it also opens up the possibility for use as a real-time conflict alerting probe. A prototype alerting logic was developed and has been utilized in several NASA Ames Research Center experimental studies.

This document is based on the thesis of Lee C. Yang submitted to the Department of Aeronautics and Astronautics at the Massachusetts Institute of Technology in partial fulfillment of the requirements for the Degree of Doctor of Philosophy.

Acknowledgments

This research was supported by the NASA Ames Research Center under grant NAG-2-1111. The authors are especially appreciative of Paddy Cashion, Sandy Lozito, Walt Johnson, Diane Carpenter, Paul Soukup, Laura Colletti, Barry Sullivan, Joe King, Greg Pisanich, Dominic Wong, Tony Wang, and Kevin Corker at NASA Ames Research Center for their support and input during the development and implementation of the conflict alerting logic.

The research documented in this report would not have been possible without the help and input from Professor R. John Hansman, Professor Walt Hollister, and Professor Eric Feron at the Massachusetts Institute of Technology.

Table of Contents

Chapter 1

Introduction

As the sky above becomes more congested, new concepts of Air Traffic Management (ATM) are being proposed to handle the expected growth [RTCA, 1995; Wangermann, 1994; Phillips, 1996, Brudnicki and McFarland, 1997]. To enable more efficient ways and procedures of moving traffic about in the airspace, many methods will require the relaxation of the rigid airway structure and in-trail spacing currently being used to maintain traffic separation. The new concept is based on the idea of reducing restrictions on individual flight paths. Consequently, to handle the increased traffic volume or possible loss of airway structure, automated traffic conflict detection and resolution tools would be required to aid pilots and/or ground controllers in ensuring safe separation at all times.

To predict traffic conflicts, it is necessary to project the future positions of aircraft over time. However, uncertainty is inherent in the prediction of any future event and the same is true in conflict prediction. Due to random processes, there is variability and uncertainty in the aircraft trajectory that make it difficult to precisely determine the aircraft's location at future times. It is this deficiency that produces errors in the determination of conflict and causes problems in the design of an effective alerting system.

The use of a probabilistic approach can be helpful when uncertainty is expected or prevalent. It allows assessment of the likelihood of specific outcomes and provides the end-user with additional information that could be beneficial in the decision-making process. Probability methods can also be placed in the role of assessing the overall hazard level of the encounter situation and the difficulty of resolving the conflict at hand. By utilizing these properties, the effects of various uncertainty elements in the aircraft trajectory on alerting system performance may be analyzed. Of special interest is the importance of including additional intent information into the alerting scheme.

When aircraft intent is added into the aircraft trajectory prediction, it can significantly reduce the uncertainty in the estimated future path and hence lower errors in conflict determination. The effects can be analyzed using probability metrics such as the false alarm rate or successful alert probability to examine the benefits of the added intent information in specific situations. However, one should be cautioned that erroneous intent assumptions could also lead to missed detections of conflicts and inaccurate estimates of hazard. Mistakes such as these occur from incorrect modeling of the aircraft trajectories and can actually lead to additional sources of estimation errors.

1.1 Objectives

There are several major objectives contained within this thesis. One is to explain how uncertainties in trajectory estimation significantly impact and hinder conflict alerting systems. A unifying concept is proposed to explain various methods of improving prediction and alerting performance in the presence of aircraft trajectory uncertainties. The framework ties together such approaches as improved sensor accuracy, added intent information, path conformance checking, reduced false alerts, and also probabilistic trajectory estimation into a single underlying basis for improving conflict prediction and alerting. In addition, a foundation is laid to explain how probability concepts are already embodied within the general framework of alerting system design.

Finally, this work presents a method used to model, analyze, and compute the likelihood of conflicts in the presence of uncertainties. The approach uses probability density functions to model potential trajectories and utilizes Monte Carlo simulations to calculate the likelihood of violating separation minimums. A technique was developed and refined to perform the calculations in a manner efficient enough to be used as a possible real-time conflict detection and resolution probe.

1.2 Overview of Thesis

To begin, Chapter 2 introduces the relevant terms and definitions used to describe alerting system performance. The notion of uncertainty in the state trajectory estimation and its effects on prediction and alerting is given here in the context of state-space terminology. Also, the tradeoff between false alarms and missed detections is presented using System Operating Characteristic curves.

In Chapter 3, the problem of conflict detection and resolution is formally introduced. A brief overview of current operational and developmental conflict alerting approaches are discussed. Initial motivation will be provided for the use of the probabilistic approach as compared to other previous modeling efforts, though the foundation will be further substantiated in subsequent chapters.

Chapter 4 gives a general guideline for improving alerting performance in the presence of uncertainties. It ties together various approaches from two main goals of increasing prediction accuracy and reducing inherent uncertainty. The importance and concept of uncertainties and trajectory modeling errors are also explained in terms of their degrading effects on the alerting system performance.

A probabilistic connection is developed in Chapter 5 from looking at the typical alerting system design approach from a different perspective. The paradigm provides further motivation for utilizing probabilistic trajectory models for conflict alerting and prediction.

Chapter 6 discusses the methodology and tools used in this thesis for the calculations of the conflict probabilities (Monte Carlo simulations). The basic trajectory model is developed and accuracy in its modeling is discussed.

In Chapter 7, several example conflict encounters are studied utilizing the methodology developed in this thesis. The potential of the method to handle very complicated situations is shown by these examples.

Chapter 8 describes an effort of transforming probabilistic conflict analysis into a real-time alerting system. Issues related to the use of probability and the implementation of real-time conflict probing are discussed. Experience and lessons learned are brought forth.

Chapter 9 provides a final summary and major contributions introduced in this thesis.

Chapter 2

Review of Alerting System Performance

At its root, conflict detection and resolution is a process of determining the existence of a possible hazard and alerting the air traffic controller and/or flight crew. Accordingly, it is worthwhile to begin by examining the operation of the alerting system in general.

2.1 Generic Alerting System

A hazard alerting system is one of several safety components typically found in complex human-operated systems [Kuchar, 1995]. Its purpose is to monitor potential threats and issue warnings to human operators when undesirable events are predicted to occur. A simplified diagram of a generic alerting system within the context of the entire operating environment is shown below in Figure 2-1.

Information about the situation is measured by sensors and presented to the human operator via various types of displays. The same information or a subset of it can also be fed to an alerting system to help determine the possibility of a hazardous situation. In many cases, a hazard can be detected by the operator from the displays themselves; however, in other instances, the operator may not be fully aware of the situation or may need additional confirmation to aid in decision making. An alert is supposedly a prediction that an unsafe state may occur, but reliance is usually still on the human operator to make the final decision. The degree of automation can vary, with some alerting systems providing a simple warning, while others give additional resolution advisories. It is also possible to have an alert fully automated to initiate a resolution, and the operator is only informed of the action.

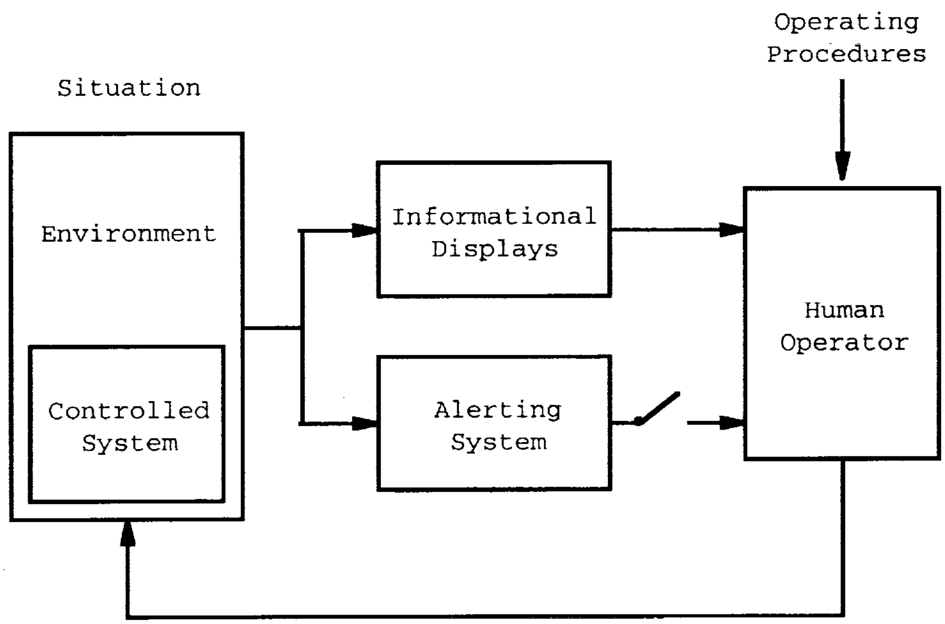


Figure 2-1: Generic Alerting System in Operation with Human Operator [Kuchar, 1995]

2.2 State-Space Representation of Alerting Systems

The use of state-space representation is a way of introducing the concepts and issues associated with alerting system design. The method was developed by Kuchar [1995, 1996] and is based on multivariable control system theory. The following is a brief review from that previous work.

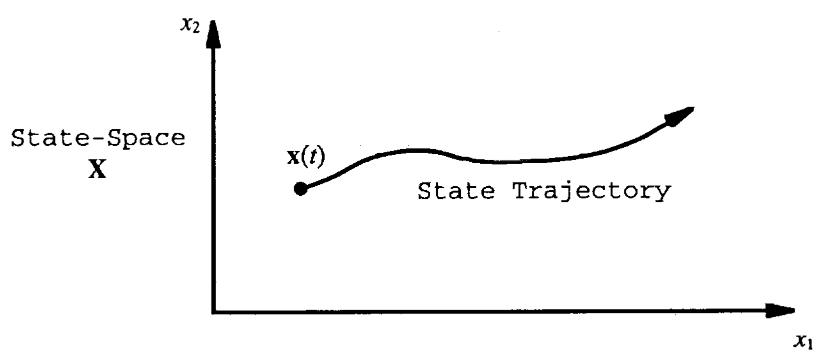
In the approach, the variables $x_1(t), x_2(t), \ldots, x_n(t)$ describe the *states* of the encounter situation at time t. These states can be thought of as the set of parameters utilized by the alerting system logic to characterize the dynamics of the threat condition. The state vector, $\mathbf{x}(t)$, is then defined as:

$$\mathbf{x}(t) = \begin{bmatrix} x_1(t) & x_2(t) & \cdots & x_n(t) \end{bmatrix}^{\mathrm{T}} \tag{2.1}$$

where n is the number of elements chosen to describe the situation. At any given time, t, the current state of the system, as known to the alerting logic, is at some particular point identified by $\mathbf{x}(t)$ in the n-dimensional state-space $\mathbf{X}$.

2.2.1 State Trajectory

The states of the system will typically change over time during the course of operation. These changes in the state vector occur in accordance with the system dynamics, the environment, and the inputs from the human controller. The set of values of $\mathbf{x}(t)$ over a given time interval is the *state trajectory*. Figure 2-2 shows an example of a state trajectory as viewed in a State-Space Diagram.



$$\mathbf{x}(t) = \begin{bmatrix} x_1(t) & x_2(t) \end{bmatrix}^{\mathrm{T}}$$

Figure 2-2: Example State-Space Diagram

2.2.2 Hazard Space

In certain regions of the state-space $\mathbf{X}$, there are domains where undesirable events can occur. These regions are termed *hazard space* (as denoted by $\mathbf{H}$). Whenever $\mathbf{x}(t)$ is allowed to enter a region of hazard space, a *missed detection* has occurred and the alerting system has failed to provide the necessary protection to prevent an unwanted event. An example of hazard space as depicted in a State-Space Diagram is shown in Figure 2-3.

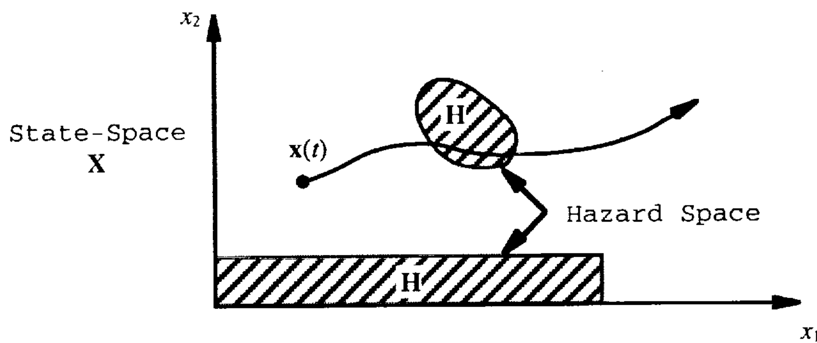


Figure 2-3: Example Hazard Space in State-Space Diagram

2.2.3 Alert Space

The *alert space* is defined as the set of all state-vectors, $\mathbf{x}(t)$, in which the alerting system will warn the operator in order to prevent a possible intrusion into hazard space. By definition, no alerts are generated when $\mathbf{x}(t)$ is outside the alert space. The boundaries of alert space are considered the alerting *thresholds* and basically define when alerts are given and when they are not. In a given state-space, $\mathbf{X}$, the regions of alert space will be denoted as $\mathbf{X}^A$.

In Figure 2-4, an example alert space is shown in a State-Space Diagram. When the state trajectory first enters alert space (point **1**), an alert will be given. At this point, the alerting logic has decided that an intrusion into hazard space is likely if nothing is done to warn the human operator (solid line). In other words, the alerting thresholds have been surpassed. By initiating the alert, it is expected that some action will be performed (depicted at point **2**) to alter the course of the state trajectory (dashed line) in order to prevent a hazard from taking place. There is usually some delay from point **1** to point **2** as the human operator decides on the appropriate response to take. Because an alert is a precursor warning to avoid a hazard, the alert space should encompass all regions of hazard space.

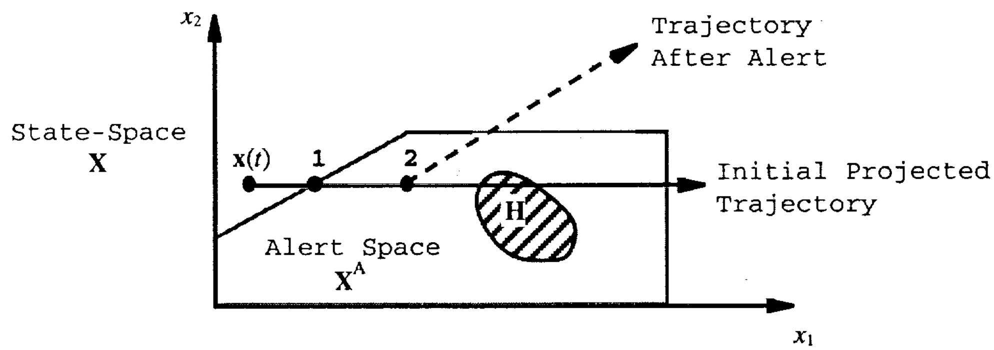


Figure 2-4: Example Alert Space in State-Space Diagram

2.3 Alerting Decision Outcomes

Ideally, an alert correctly notifies when a hazard will occur if nothing is done to alter the current situation of the system. To be more complete, the alert should also allow for absolute resolution of the threat if it is to be considered a safety feature. If both these elements are satisfied, then the alert is termed a *correct detection* (CD). If, however, the hazard is not prevented (whether or not an alert is given), the outcome would be considered a *missed detection* (MD) because the system has failed to provide the intended safe avoidance of the hazard. An alert that is given but was not necessary (because a hazard would not have occurred in the first place) is usually termed a *false alarm* (FA). For sake of completeness, normal operation with no threat and correctly indicated by the alerting system will be considered a correct rejection (CR). The complete decision outcome is diagrammed in Figure 2-5.

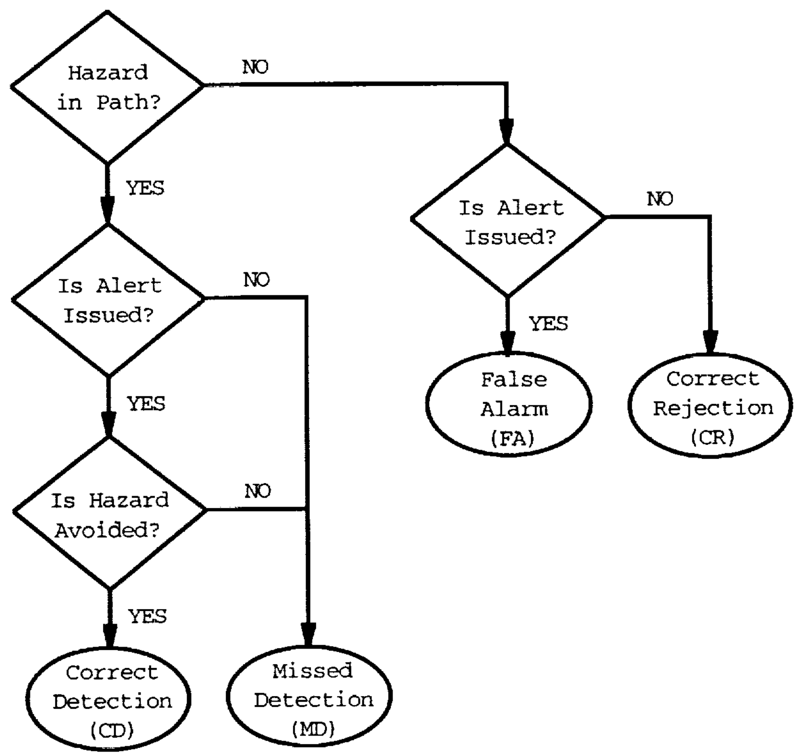
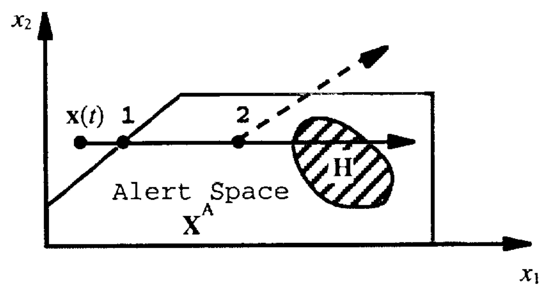
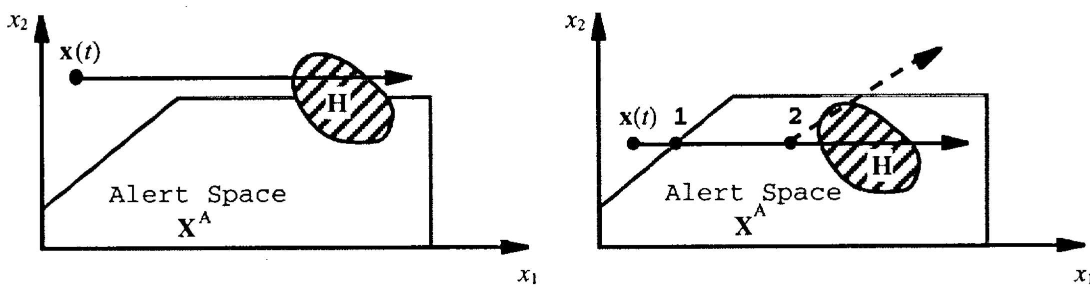


Figure 2-5: Alerting Decision Outcomes

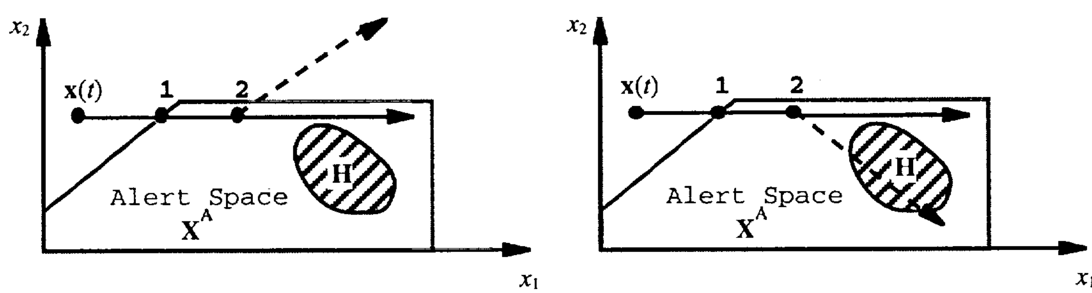
The corresponding outcomes can be graphically depicted using State-Space Diagrams as shown in Figure 2-6. The points **1** and **2** refer to when the alert is given and when the response action is initiated, respectively. The solid lines are used to indicate the state trajectory which would have occurred had the alert not been given, and the dotted lines refer to the new state trajectory from the response to an alert. Figure 2-6a (correct detection) has been discussed already, and along with Figure 2-6d (correct rejection), represent the two cases of an ideally operating alerting system.



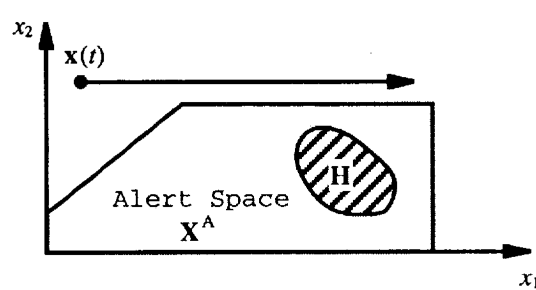
a) Correct Detection



b) Missed Detection (No Alert/Late Alert)



c) False Alarm (Nuisance Alert/Induced Hazard)



d) Correct Rejection

Figure 2-6: State-Space Diagrams of Alerting Outcomes

Trouble occurs when either a missed detection or a false alarm is experienced. In the case of a missed detection, an alert is needed (state trajectory will enter hazard space) but the alerting system fails in preventing the hazard from occurring. Sometimes, a missed detection is further sub-divided into two categories as shown in Figure 2-6b: missed detection due to no alert and missed detection due to late alert [Winder and Kuchar, 1999; Haissig, et al. 1999]. The former is most likely due to lack of information of the states of the system (both $\mathbf{x}(t)$ or $\mathbf{H}$) or from design errors in the alerting algorithm. In the late alert case, either the operator is not given enough time to decide and perform the appropriate action, or for some reason, the warning is not heeded or is just ignored.

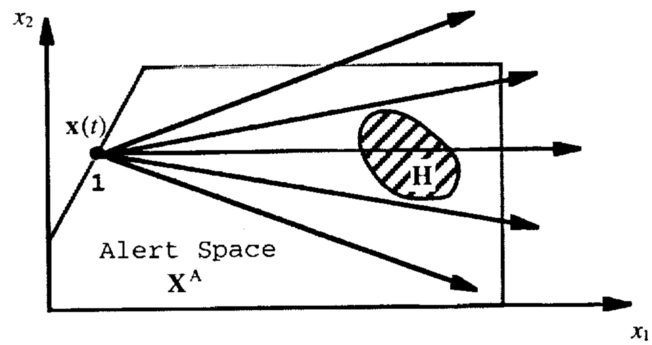
False alarms, as depicted in Figure 2-6c, occur whenever an alert is given but the state trajectory would not have entered into hazard space without it. A false alarm can also be parsed down further into two sub-categories: false alarm resulting in no hazard (nuisance alert) and false alarm causing induced hazard [Drumm, 1996; Winder and Kuchar, 1999; Haissig, et al. 1999]. Both cases decrease efficiency and increase workload for the human operators involved. At first glance, it might seem that the former would be of little concern to safety. However, the increased occurrence of such nuisance alerts can directly impact the response of the human operator in actual emergency situations. This is especially important when quick and decisive action is called for in order to prevent a catastrophic loss of the system (e.g. collision between two aircraft).

2.4 Tradeoff Between Missed Detections and False Alarms

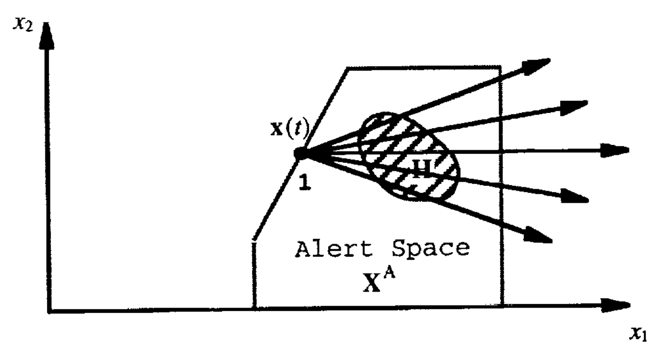
When a state trajectory first enters the alert space (assuming $\mathbf{x}(t)$ is coincidental with point **1** in Figure 2-4), an alert is initiated. As stated earlier, the boundaries of this alert space define the alerting threshold of the alerting logic. Since $\mathbf{x}(t)$ is only an estimate of the current state of the threat condition, an element of prediction is inherently involved in determining the path that $\mathbf{x}(t)$ will follow. The decision to alert is based on the logic's prediction that an intrusion into hazard space is likely given this $\mathbf{x}(t)$. If the prediction is wrong, then a false alarm (FA) has occurred. If the prediction is correct, then either the alert prevents the hazard from occurring (CD) or the alert is too late in avoiding a hazard space incursion (MD). Thus the event of an alert results in one of 3 mutually exclusive outcomes: CD, MD, or FA. The likelihood of any of these events occurring (given an alert) can be expressed in statistical properties such as the probability of correct detection, $P(CD)$, missed detection, $P(MD)$, and false alarm, $P(FA)$.

If the alert space is made relatively large, alerts will occur more often during the operation of the system. This is the conservative approach. It can reduce the number of missed detections but at the expense of an increased rate of false alarms. If the alert space is made relatively small, less alerts will occur (fewer false alarms), but at the expense of increased missed detections from late alerts. Thus, here lies the fundamental tradeoff between MD and FA in alerting system design: reducing $P(MD)$ will increase $P(FA)$ while reducing $P(FA)$ will increase $P(MD)$. The result is similar to the problem found in signal detection theory as demonstrated by Kuchar [1995, 1996] in his work on System Operating Characteristic (SOC) curves. An overview of the SOC technique is presented in a later section.

The reason a larger alert space will generally increase $P(FA)$ lies in the mere fact that more states are included in the alert space. This allows for a higher probability that an alert will be induced whether or not it is needed. There is much more room for error in the prediction that hazard space will be reached if nothing is done. Take, for example, Figure 2-7a where the alert space is relatively large compared to the hazard space. Depending upon the dynamics of the system, there can be a higher probability that the state trajectory will not enter into the region of hazard space as compared to the case in Figure 2-7b where the alert space is smaller. Thus, the larger alert space will usually result in a higher rate of false alarms.



a) Larger Alert Space (More Alerts/Higher Rate of FA/Lower Rate of MD)



b) Smaller Alert Space (Fewer Alerts/Lower Rate of FA/Higher Rate of MD)

Figure 2-7: Effect of Alert Space Size on False Alarms

However, the smaller alert space may end up sacrificing the ability to escape from the hazard (i.e. provide insufficient warning time) and thus lead to a higher rate of missed detections. This tradeoff is the fundamental design challenge which alerting system designers are often faced with.

2.5 Role of Uncertainty in the Alerting Outcome

Due to the nature of prediction, the estimate of future events is inherently uncertain to some extent. As alluded to in the previous section, the path of the state trajectory is usually not known exactly. This leads to a statistical description of the alerting outcomes (e.g. $P(CD)$, $P(MD)$, and $P(FA)$). Much of this can be attributed to uncertainties with predicting the state trajectory from only the current state, $\mathbf{x}(t)$.

Take, for example, Figure 2-8 where three different aircraft encounter scenarios are shown. In each case, the range (r) is 100 nautical miles and the range rate ($\dot{r}$) is 566 knots. If the alerting decision is to be based on only these two parameters, then the current state vector $\mathbf{x}(t) = [r \quad \dot{r}]^T$ would be identical for each of these three cases – the alerting algorithm would be unable to tell them apart. However, the outcome from each scenario is decidedly different. In the case of Figure 2-8a, a direct collision would occur, while in the other 2 cases, no real threat is encountered.

The comparison between Figure 2-8a and Figure 2-8b is especially important to note because it shows just how much the predictive path of the vehicle states can come into play even with the same apparent initial conditions. Though the current position and velocity of each aircraft is the same in these cases, the latter would result in a false alarm if an alert were to occur at the present time.

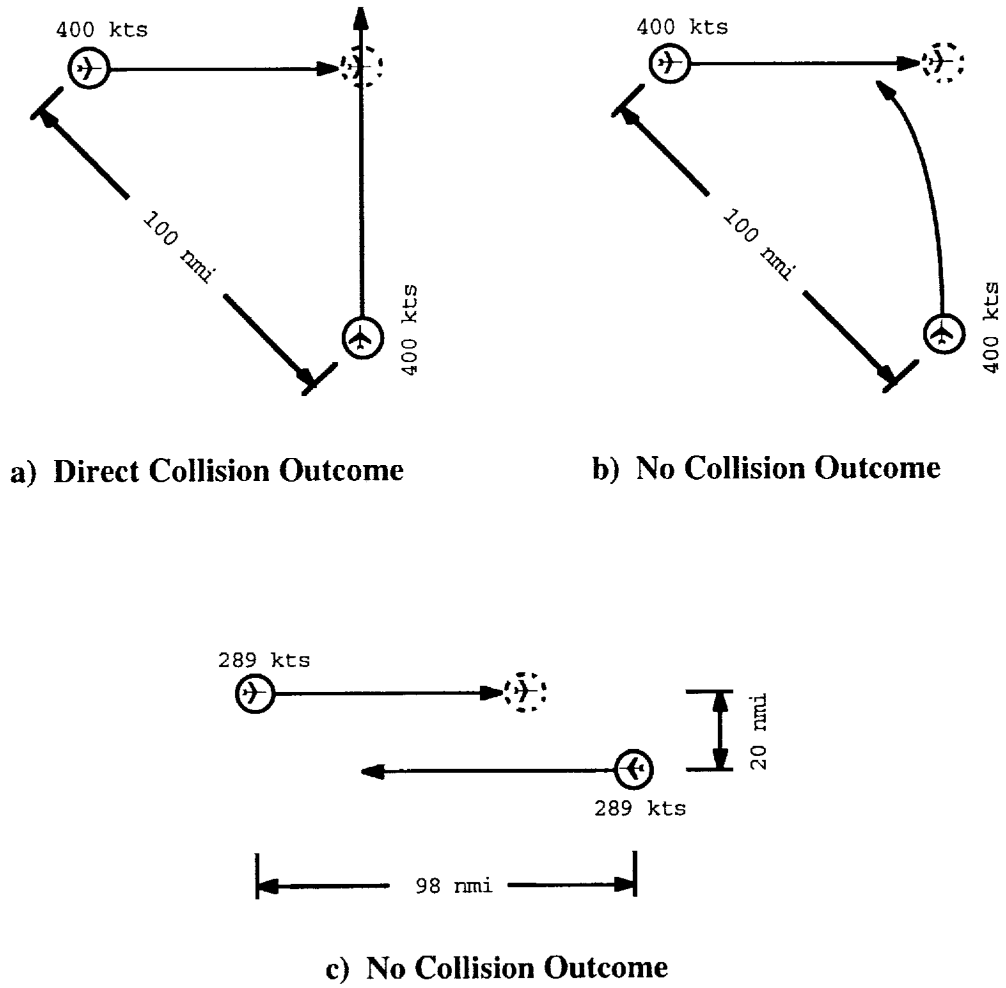


Figure 2-8: Example Encounters with the Same State Vector,
$\mathbf{x}(t) = \begin{bmatrix} r & \dot{r} \end{bmatrix}^T = \begin{bmatrix} 100\text{nmi} & 566\text{kts} \end{bmatrix}^T$

In the case of Figure 2-8c, the situation appears quite different to the previous two scenarios, but would actually be transparent to an alerting algorithm based on only relative range and range rate at the current time. Unless some other provision is included to differentiate the scenes (e.g. relative bearing), the alerting algorithm would likely treat all three scenario encounters the same at this particular instant in time. The consequence of this is a higher degree of uncertainty in the alerting decision outcome. This effect can be seen in the state-space representation shown in Figure 2-9 where the state trajectory of the three cases from the previous figure are plotted. In the figure, a hazard is assumed to be a loss of separation of some predefined distance, such as 5 nautical miles or less, between aircraft.

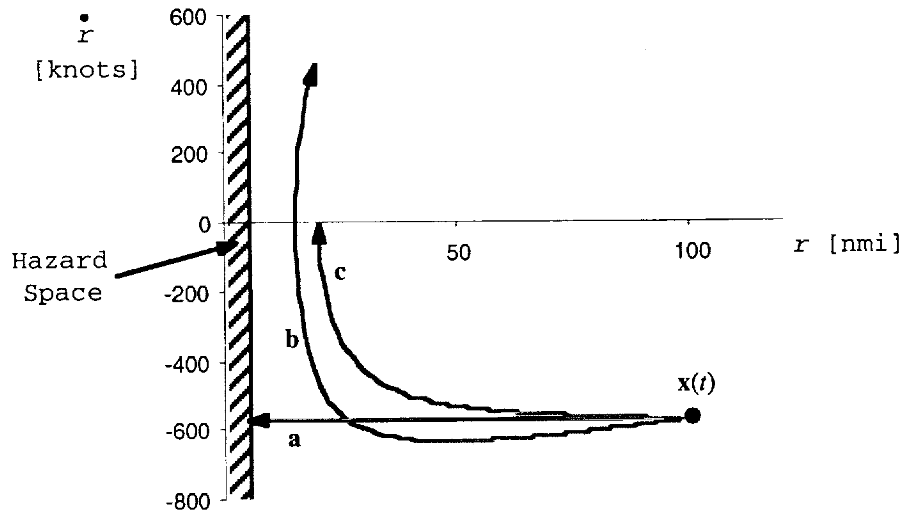


Figure 2-9: State-Space Representation of Figure 2-8

As shown in the figure, 2 of the 3 scenarios would have incurred a false alarm if the decision was made to alert at $\mathbf{x}(t)$. The result would be a high rate of false alarms due to the inability to predict the outcome of the action from current state vector.

Even if the decision to alert is justified, such as in Figure 2-8a, there can be further uncertainties that affect the new state trajectory in response to the alert. These uncertainties (e.g. response time and avoidance action) would inevitably influence the likelihood that the hazard could be avoided. The outcome would be a direct impact of the correct detection and missed detection (by late alert) rates, $P(CD)$ and $P(MD)$, respectively.

Further uncertainties, such as those due to stochastic randomness, can enter into the problem as well. In the case of aircraft, this could include fluctuations in speed, heading, or altitude of each vehicle involved. There may also be course changes not known at the present time that could by initiated by the flight crew to avoid weather or meet performance goals (e.g. time of arrival, fuel savings, flight comfort). The changes may also be inadvertent due to pilot blunders to maintain an expected course of flight. All this leads to uncertainties in the state trajectory which can affect the outcome of each alerting decision.

2.6 System Operating Characteristic Analysis

The approach of the System Operation Characteristic (SOC) method has its roots in signal detection theory. It was developed by Kuchar [1995, 1996] to help analyze and design alerting thresholds by examining the tradeoff between successful alerts and false alarms. Much of the method is based on the use of probabilistic trajectory analysis in the propagation of the state vector.

2.6.1 Probabilistic Trajectories and the Probability of Conflict

As stated before, the prediction of future events inevitably involves uncertainties, and the same is true of the state trajectory. In general, the path of the true state trajectory is not known exactly, and it can be assumed to be probabilistic and include uncertainties to some degree. The concept is detailed in the following Figure 2-10 where the shaded area represents possible state positions, or uncertainties, in the future of the system. Usually, but not always, the uncertainty in the trajectory will tend to grow with time as it naturally becomes more difficult to predict further into the future.

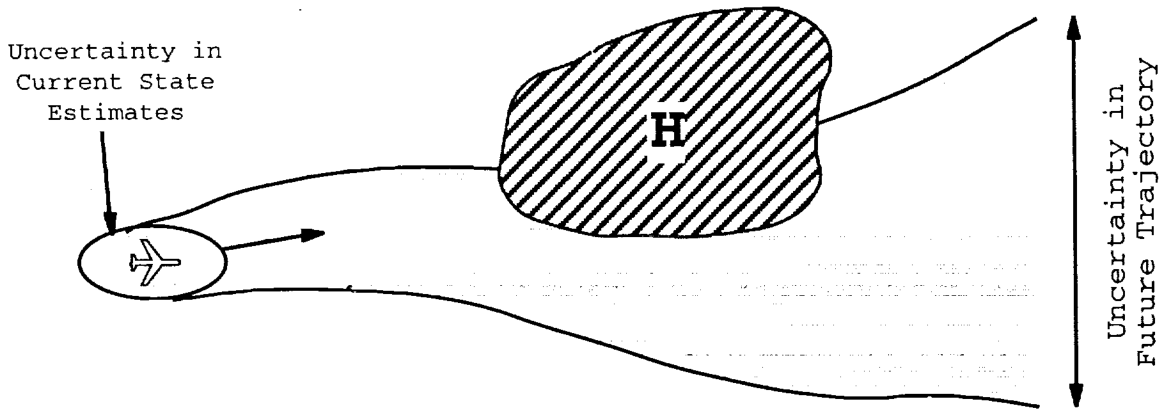


Figure 2-10: State Trajectory with Uncertainties

In state-space, the term *conflict* will refer to the occurrence of an undesirable event (i.e. hazard space incursion of the state vector). In estimating the likelihood of its occurrence, the term *Probability of a Conflict*, $P(C)$, will be used.

2.6.2 Nominal and Avoidance Trajectories

To determine if an alert is warranted in a given situation, it is necessary to examine the hypothetical outcomes of the alert / no alert decision. If no alert is issued, the state continues along what will be termed the projected *nominal trajectory*, denoted as **N**. Similarly, in response to an alert, there is a different projected path that is taken called the *avoidance trajectory*, denoted **A**. Both **N** and **A** are, in general, probabilistic due to the uncertainties in the current and projected future states. During an avoidance action, there are many variables which may make it difficult to predict the exact path taken, especially with human involvement (e.g. different response times and actions). Trajectory **A** may also include the possibility that no action is taken in response to the alert.

The probability of conflict along **N** and along **A** are denoted $P_N(C)$ and $P_A(C)$, respectively.. In Kuchar [1995], the Probability of a False Alarm, $P(FA)$, and the Probability of a Successful Alert, $P(SA)$, are defined as:

$$P(FA) = 1 - P_N(C) \tag{2.2}$$

$$P(SA) = 1 - P_A(C) \tag{2.3}$$

In the above two equations, both $P(FA)$ and $P(SA)$ are conditional upon an alert being given. Also, $P(SA)$ is specific to a particular avoidance trajectory, **A**.

2.6.3 SOC Curves

In previous work, Kuchar [1995, 1996] explored the performance tradeoff between false alarms and successful alerts. This technique is based on the System Operating Characteristic (SOC) Curve which facilitated the visualization of the exchange between the two parameters. In any conflict detection decision, there is usually some probability that the alert is not needed. Additionally, there is some probability that the alert is successful in prevent a conflict. As one varies the time at which the alert is generated, these probabilities trade off against one another as described by an SOC curve.

In order to determine if an alert is successful, it is necessary to consider what resolution action occurs when the alert is given. Some conflict resolution maneuver must be assumed so that it can be determined whether a conflict is ultimately averted by the alert. Thus, a SOC curve is specific to both the encounter geometry and the type of resolution action that is prescribed. In simple terms, a SOC curve is a plot of $P(SA)$ versus $P(FA)$ along a specific nominal path, **N**, and for a specific avoidance maneuver, **A**.

The choice of avoidance trajectories for conflict analysis will depend on the performance criteria to be met. The criteria can be safety-based, in which the trajectory is to reflect an aggressive maneuver, or it can be more cost driven, in which the trajectory represents a more strategic maneuver.

A sample SOC plot is shown in Figure 2-11 for a path on a direct collision course to a hazard. The points **1** and **2** correspond to different alerting times, with point **1** being earlier than point **2**. If the conflict decision is made while the hazard is far away, (upper right corner of the plot), the probability of a successful alert is likely to be very high ($P(SA) \rightarrow 1$); but because action is taken so early, the probability of a false alarm is also high ($P(FA) \rightarrow 1$). With the hazard far in the distance, there is typically too much uncertainty in the nominal **N** trajectory to alert without knowing if it was really necessary. As the conflict alert decision is delayed and the hazard continues with increasing threat, the probability of successful alerts and false alarms both decrease as shown by the curve. If alerts are delayed too long, the alerts will not be successful ($P(SA) \rightarrow 0$) and there will be no false alarms as well ($P(FA) \rightarrow 0$).

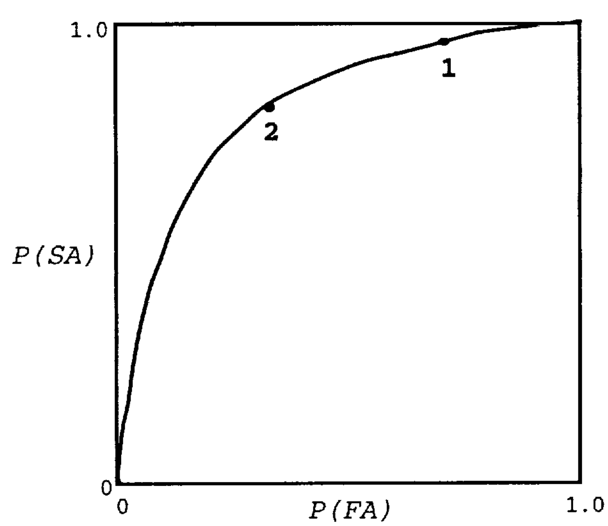


Figure 2-11: Typical SOC Plot

The SOC curve shows the tradeoff between $P(FA)$ and $P(SA)$ as a function of the alerting threshold location for a series of decision points along a chosen path. The curve in Figure 2-11 clearly shows the effect of delaying the alert on reducing the chance of a false alarm. The corresponding drop in a successful alert, $P(SA)$, is also evident. The location of the threshold can be examined relative to the desired level of nuisance alerts and safety margin. It is also possible to utilize the SOC curve as part of a preliminary design evaluation for setting the alert threshold.

In Figure 2-12, a pictorial perspective on the underlying principle behind the SOC is given. Far away, with the hazard at location **1**, there is usually sufficient uncertainty to warrant delaying an alert so that false alarms are not too predominant. However, waiting too long may result in an unavoidable hazard (missed detection). The state-space analogy was explained back in Figure 2-7 with the discussion on the alert space size.

As mentioned earlier, high rates of nuisance alerts can lead to mistrust of a system, and thus, also indirectly impact safety. Deciding when to alert (i.e. threshold placement) is one of the most crucial elements in alerting design. The choice becomes obvious in Figure 2-12 when the avoidance trajectory, **A**, is superimposed over **N**. To ensure the alert is successful, it must be issued prior to a high likelihood of conflict along the avoidance trajectory (location **2**). In other words, the alert must be in time for the aircraft to avoid the hazard. This appealing concept is captured nicely in SOC plots.

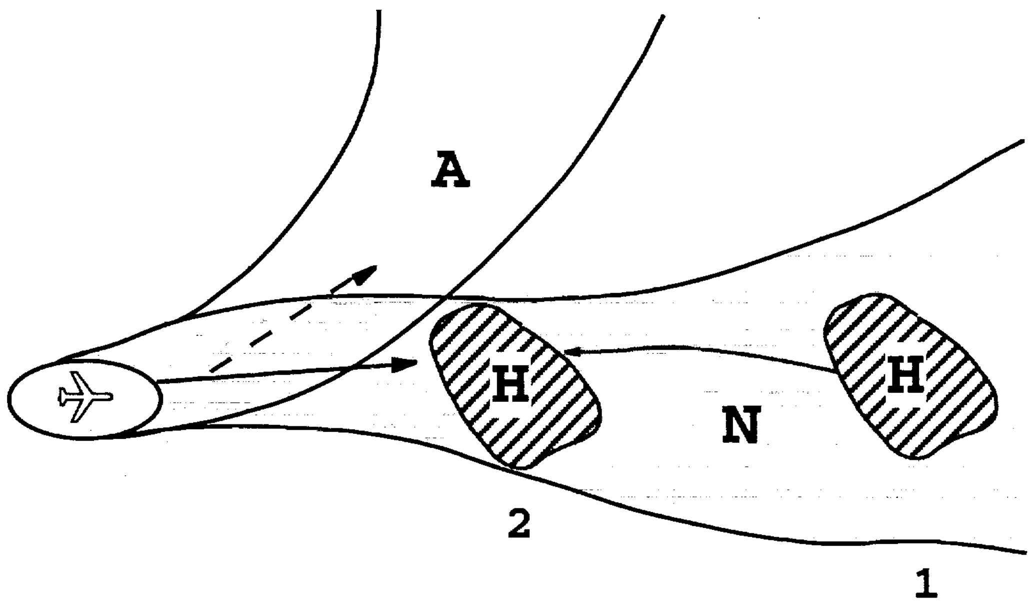


Figure 2-12: Delaying the Alert

The shape of the SOC plot can provide a lot of information regarding the possible performance of the alerting system. A curve that allows placement of the threshold at the upper left corner ($P(FA) = 0$, $P(SA) = 1$) is considered ideal since there would be no false alarms and only successful alerts (see Figure 2-13). Due to uncertainties in the conflict dynamics, however, the SOC curve will generally lie somewhere below this optimal point. The closer a system is able to operate near this optimal point, the more effective the system will be in terms of providing an acceptable level of safety while minimizing unnecessary alarms.

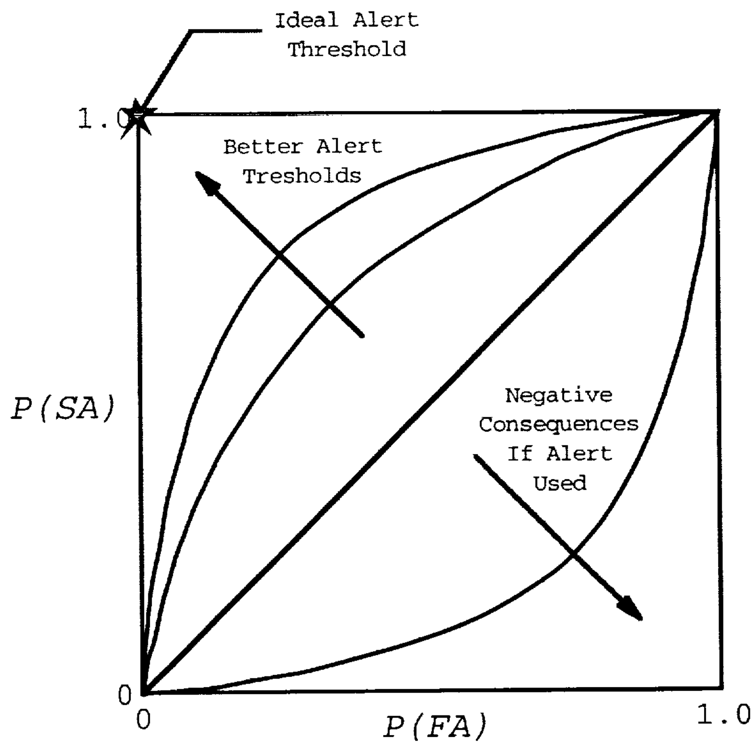


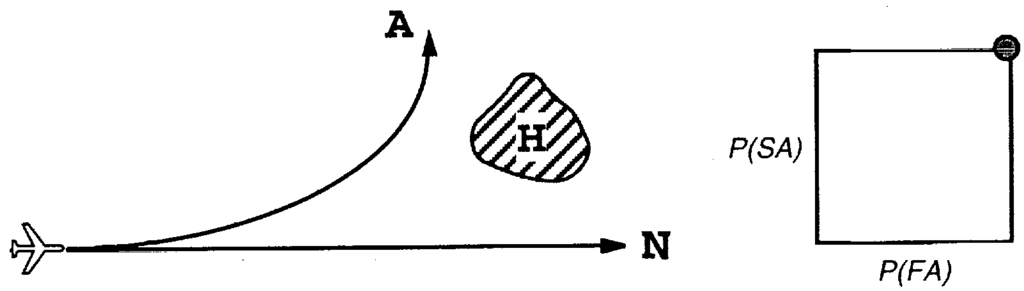
Figure 2-13: Various Shapes of the SOC Curve

A curve that lies diagonally from the origin ($P(FA) = 0$, $P(SA) = 0$) to the upper right corner ($P(FA) = 1$, $P(SA) = 1$) represents either a poorly chosen avoidance maneuver for the particular encounter or an inherently difficult situation due to the uncertainties involved. In such circumstances, the alert basically has no effect in altering the outcome of a conflict. The alert is just as likely to produce a conflict as if no alert was given. Thus the more the SOC curve deviates upward from the diagonal, the more likely a better alerting decision can be determined.

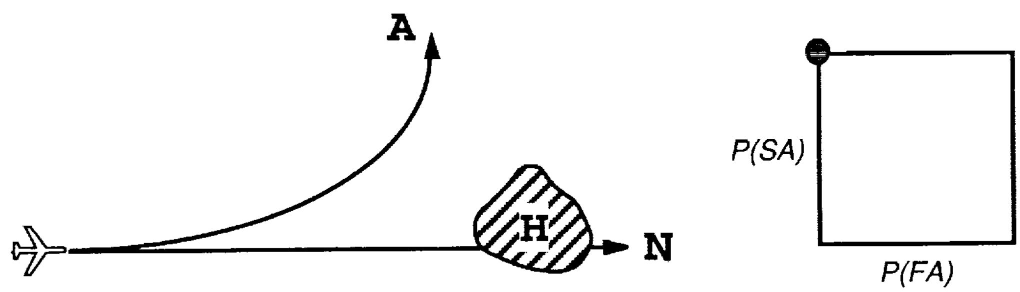
It is also possible that an alert can induce a negative consequence in the encounter situation. In this case, the avoidance trajectory incurs a higher likelihood of a conflict that along the nominal trajectory when no alert is given. The resultant SOC curve would deviate below the diagonal as shown in Figure 2-13.

The characteristic bend or drop in $P(SA)$ that is sometimes found in an SOC plot can be largely attributed to the specific avoidance maneuver being examined, but is also influenced by the underlying uncertainties in both the nominal and avoidance trajectories, **N** and **A**, as well as that from the hazard, **H**. In general, if no uncertainties were present and the future trajectories could be predicted with utmost precision, then an ideal alerting system would most likely be possible. A point at the upper left corner would exist provided the alert is given early enough. Different maneuvers would affect the required alert time, though. A 30 degree bank turn maneuver performed by an aircraft may provide an ideal system for a specific encounter if the alert is given prior to 15 seconds prior to predict conflict, while a 1000 feet per minute climb, in the same situation, might need the alert to be given 23 seconds ahead of time. But assuming there are no uncertainties involved, both maneuvers could provide perfectly ideal alerting thresholds ($P(FA) = 0$, $P(SA) = 1$).

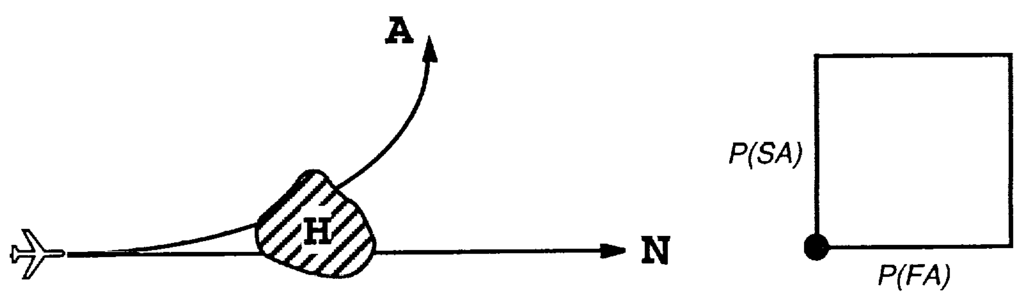
Each of the 4 extreme corners in the SOC diagram represents an absolute certainty condition. Thus, if no uncertainty is present in the trajectory or position of the aircraft, the state must lie at one of these corner positions. Either a conflict will exist along the nominal trajectory, **N**, or it does not. Either the conflict can be avoided with the avoidance trajectory, **A**, or it can not. These 4 extreme conditions are shown in Figure 2-14. Although it is assumed here that the trajectories are known perfectly, it is still possible for the SOC curve to be at any of the corner positions even with some uncertainties present in the trajectories. Most likely, however, the locus of points will lie somewhere within the boundaries of the 4 corners.



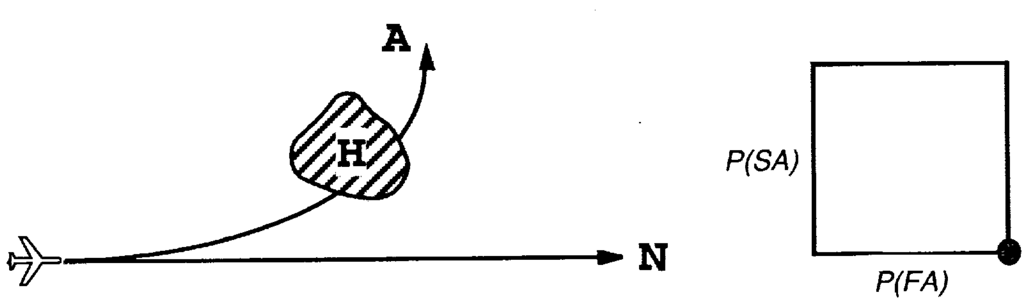
a) **No Alert Needed but Alert Causes No Conflict** ($P(FA) = 1$, $P(SA) = 1$)



b) **Alert Need and Alert Successful** ($P(FA) = 0$, $P(SA) = 1$)



c) **Alert Needed but Alert Unsuccessful** ($P(SA) = 0$, $P(SA) = 0$)



d) **No Alert Needed and Alert Induces Conflict** ($P(FA) = 1$, $P(SA) = 0$)

Figure 2-14: Four Corners of the SOC Diagram

Thus the shape of the SOC curve can also serve as a visualization tool to gauge the effects of uncertainties in the encounter scenario. If a high level of uncertainty existed in a trajectory (either **N** or **A**), then the curve would tend to diverge from the corner positions. The result could be used to determine if a more severe avoidance option is necessary, or even to consider a different type of maneuver altogether – one that is more robust to the uncertainties involved in the scenario.

2.7 Summary

The use of the state-space representation was shown as a way of presenting the concepts associated with alerting system design. It was used to explain the ideas behind alerting performance and the parameters associated with the different alerting decision outcomes: correction detection, missed detection, false alarm, and correct rejection. Also, the System Operating Characteristic technique was highlighted as a method for examining different performance tradeoffs and provides a framework for analyzing alerting performance in later chapters.

Chapter 3

Conflict Detection and Resolution Methods

3.1 Conflict Detection and Resolution

Methods for maintaining separation between aircraft in the current airspace system have been built from a foundation of structured routes and evolved procedures. In this framework, humans have been an essential element in this process due to their ability to integrate information and make judgements. However, because failures and operational errors can occur, automated systems have begun to appear both in the cockpit and on the ground to provide decision support and to serve as traffic conflict alerting systems. These systems use sensor data to predict conflict between aircraft, alert humans to the conflict, and may also provide commands and guidance to resolve the conflict. Together, these automated systems provide a safety net should normal procedures and controller and pilot actions fail to keep aircraft separated beyond established minimums.

Recently, interest has grown into developing more advanced automation tools to detect and resolve traffic conflicts. These tools could make use of more advanced technologies, such as datalink of current aircraft flight plan information, to enhance safety and enable new procedures to improve air traffic flow efficiency.

To begin, it is necessary to have a clear definition of what constitutes a conflict. For the majority of this thesis work, a conflict will refer to a situation in which an aircraft experiences a loss of minimum separation with another aircraft. In other words, the distance between them violates a preset criterion that is considered undesirable. One example might be a 5 nautical mile horizontal distance between aircraft and a 1000 feet vertical separation (current Air Traffic Control standards). The result is a *protected zone* or volume of airspace surrounding each aircraft that should not be infringed upon by another vehicle (see Figure 3-1). The protected zone could also be defined much smaller depending upon the goals of the alerting system (e.g. parallel runway incursions). It could also be specified in terms of parameters other than distance, such as time. In any case, the underlying conflict detection and resolution functions are similar although the specific models and alerting thresholds would likely be different.

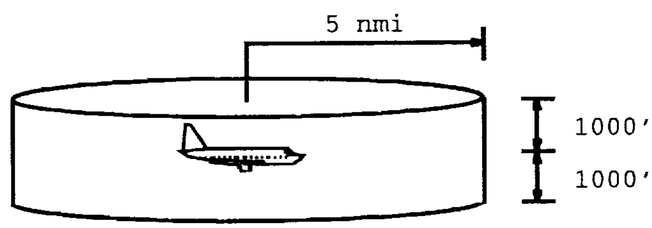


Figure 3-1: Example Protected Zone Around Aircraft

Any traffic management system in which vehicles are monitored and controlled to prevent collisions has certain basic functional requirements. The objective of a conflict avoidance system is to predict the occurrence of a conflict, communicate (alert) the detected conflict to the human operator, and then, in some cases, assist to resolve the conflict situation. These three fundamental processes can be organized into several phases or elements as shown in Figure 3-2.

To begin, the traffic environment must first be monitored and the appropriate aircraft state information must be collected and disseminated using sensors and communication equipment. These states provide an estimate of the current traffic situation (e.g. aircraft positions and velocities). However, due to sensor limitations, the information may not be complete enough to describe the actual situation. For example, a system may only have access to range and range rate information and unable to determine bearing (recall the state-space example of Figure 2-9). Additionally, there is generally some uncertainty within the values of the available states.

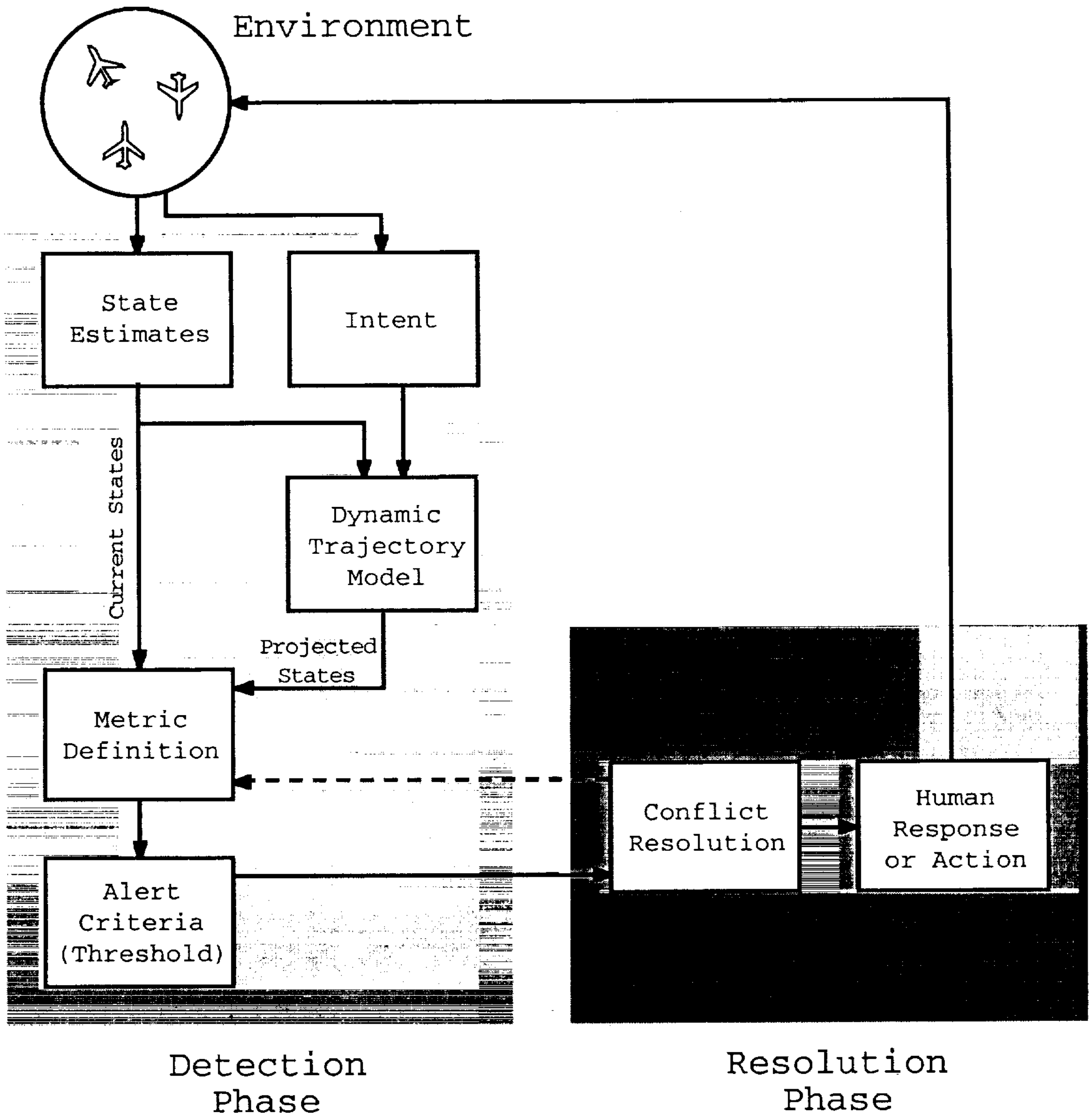


Figure 3-2: Conflict Detection and Resolution Framework

Information regarding the future intent of aircraft may also be available to the alerting algorithm. Such data might include the waypoints in flight plans, level-off altitudes in vertical maneuvers, or commanded heading during turns. The information can be used to provide additional prediction accuracy to the future trajectories of each aircraft. It is also possible that the intent of an action will not be followed so there is likely some potential uncertainties in the information as well.

Continuing on, a dynamic trajectory model is usually required to project the states into the future in order to predict whether a conflict will occur. This projection may be based solely on the current state information (e.g. a straight-line extrapolation of the current velocity vector) and may include additional intent information (e.g. the flight plan). As shown before in the previous chapter, the importance of this model cannot be understated as it has a direct impact on the overall performance of the system. Any prediction of future events inherently involves some uncertainties, and the choice of dynamic model to estimate future states is no exception.

The parameters used for the actual alerting decision will be referred to as the alerting *metrics*. Here, information from the current and predicted states are combined to provide an overall measure of threat to the occurrence of a conflict. Some example metrics include the relative range, closure rate, predicted miss distance, or the estimated time to closest point of approach. These metrics form the state space of the alerting algorithm as explained in Chapter 2.

Given the conflict metrics, a discrete decision (Conflict Detection) is then made regarding whether or not to inform the human operator of a threat. Often, this decision is based upon a simple check against specific *thesholds* (e.g. take action if predicted miss distance is less than 5 nautical miles), but could involve a more complex set of rules. The thresholds may include corrective adjustments or safety buffers to account for uncertainties as well.

Note, however, that the prediction of a conflict need not always require a notification. A conflict may be predicted, but its occurrence may be too far into the future or too uncertain to be considered a threat at the current time. The decision to alert could also hinge upon user preference, experience, or operational factors. For the purpose of this thesis work, a conflict is *detected* once it is both predicted to occur and it has been determined to be appropriate to alert the operator.

In some cases, notification of a conflict is all that is required of the alerting system (the human operator is expected to resolve the conflict independently). In other cases, a Conflict Resolution phase may be initiated. This involves determining an appropriate course of action and transmitting that information to the operator. For example, the system might present to the pilot of an aircraft the target rate of climb or descent necessary to avoid a potential collision with another aircraft. Although conflict resolution is shown as a single block in Figure 3-2, it requires its own set of current state estimates, a resolution maneuver trajectory model, and decision criteria which may be different from those used in the Conflict Detection phase. This simplification in the figure was intended to make the schematic less cluttered without diminishing the meaning of the concepts.

The response of the human to the alert is also critical to the design and efficacy of the alerting system as well. In many instances, the human's response can be included to some extent within the determination of the resolution maneuver, such as a 5 second delay. However, the human response is inevitably variable and needs to be considered as another source of uncertainty in the overall scheme of the conflict alerting process.

In the framework of Figure 3-2, conflict detection can be thought of as deciding *when* action should be taken while conflict resolution can be looked upon as determining *how* or *what* action should be performed. In practice, there may not always be a clear separation between alerting and resolution, however. Deciding when action is required may depend on the type of action to be performed; and similarly, the type of action that is required may depend on how early that action begins.

The multitude of various metrics and thresholds and also the interdependence between conflict detection and resolution are factors which make alerting system development challenging and interesting because there are many feasible design solutions. As will be shown in the next section, there are a number of ways to tackle the problem and develop a feasible solution. A more difficult task is determining the best one.

3.2 Survey of Algorithmic Designs

To provide better insight into different methods of conflict detection and resolution, a literature review of previous research approaches and current operational and developmental systems was performed. A total of over 60 different papers were used and a more detailed discussion can be found in Appendix A. These methods do not represent an exhaustive list by any means, but are believed to encompass a majority of the recent approaches to the conflict detection and resolution problems.

3.2.1 Current Operational Systems

3.2.1.1 Traffic Alert and Collision Avoidance System (TCAS)

The Traffic Alert and Collision Avoidance System (TCAS) has been the standard that many approaches to the conflict detection and resolution problem have been compared to. The system has been implemented on U.S. jet transports since the early 1990's as concern over the potential of future mid-air collisions grew. The algorithm is more complicated than will be explained here, so the reader is asked to refer to Ford [1986, 1987], Kuchar [1995], Miller et al. [1994], RTCA [1983], or Williamson [1989] for a more detailed description.

In abbreviated terms, the TCAS logic calculates threat in the horizontal and vertical dimensions separately and alerts if both criteria are met. The algorithm is based on the relative range (r) and range rate ($\dot{r}$), and also the relative altitude (h) and altitude rate ($\dot{h}$). TCAS uses a two-stage process with a cautionary alert called the Traffic Advisory (TA) and a warning alert called the Resolution Advisory (RA). RAs provide vertical avoidance commands, but TAs are merely cautions and lack any resolution. The following discussion will focus on RA alerts only.

The TCAS thresholds are actually more complex, but for the most part, can be summarized by what is commonly referred to as the Tau Criterion shown in Equation 3.1.

$$\frac{r - \mathrm{DMOD}}{-\dot{r}} < \tau \tag{3.1}$$

τ is a threshold parameter with units of time, and DMOD is a buffer distance used to account for slow closure rates, ensuring that aircraft will not drift closer than the DMOD distance without receiving an alert [Williamson, 1989]. Within the alerting logic, these two parameters are varied depending on the altitude and whether or not the aircraft are maneuvering vertically. These values are summarized in Figure 3-3.

In the notation used in the figure, the first value listed for τ is the alerting threshold for the TCAS equipped aircraft if it is level, or is climbing or descending in the same direction as the threat but at a lower rate; else the second value is used.

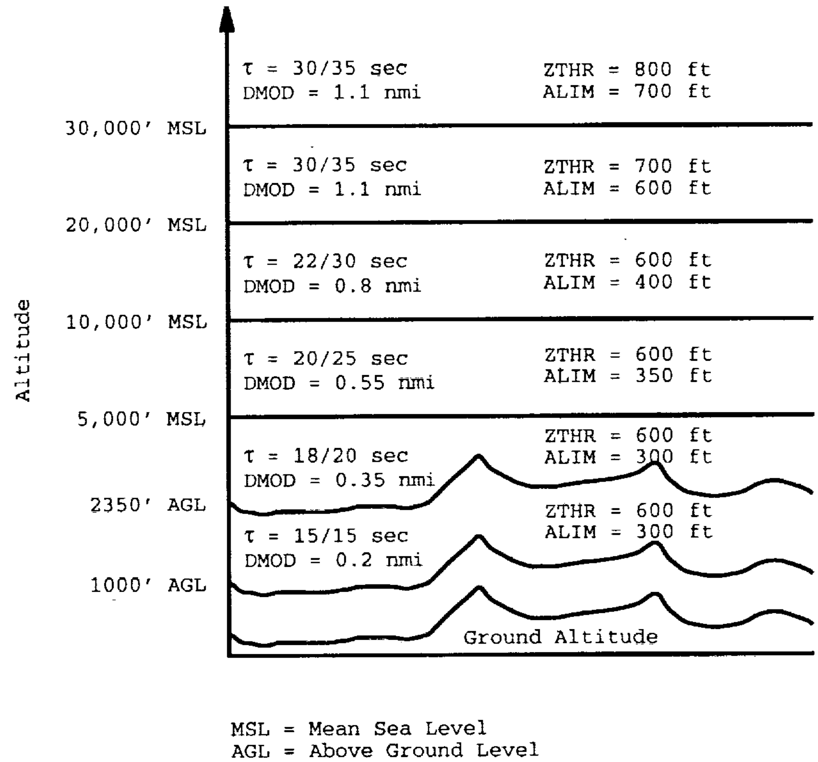


Figure 3-3: TCAS Version 6.04A RA Logic Parameters

The vertical criterion is a little more complicated, but in essence, also utilizes a Tau Criterion to estimate the time to co-altitude. It includes various buffers and parameters (ZTHR, ALIM) that are variable depending on the flight altitude and relative vertical separation.

The left side of the Equation 3.1 can also be thought of as an estimate of the time it will take for the range to decrease to a distance, DMOD, between aircraft [Miller et al., 1994]. From this point of view, the TCAS logic is assuming a straight line projection model and DMOD is acting as a buffer to account for possible deviations or sources of error.

A possible state-space representation of the alerting logic at work is shown in Figure 3-4. Here, the aircraft are assumed to be in level flight (30,000 feet) and traveling in opposite directions, each with a velocity of 400 knots. For case **A**, the opposing aircraft are on course for a direct collision; while in case **B**, the aircraft will miss by 5 nautical miles.

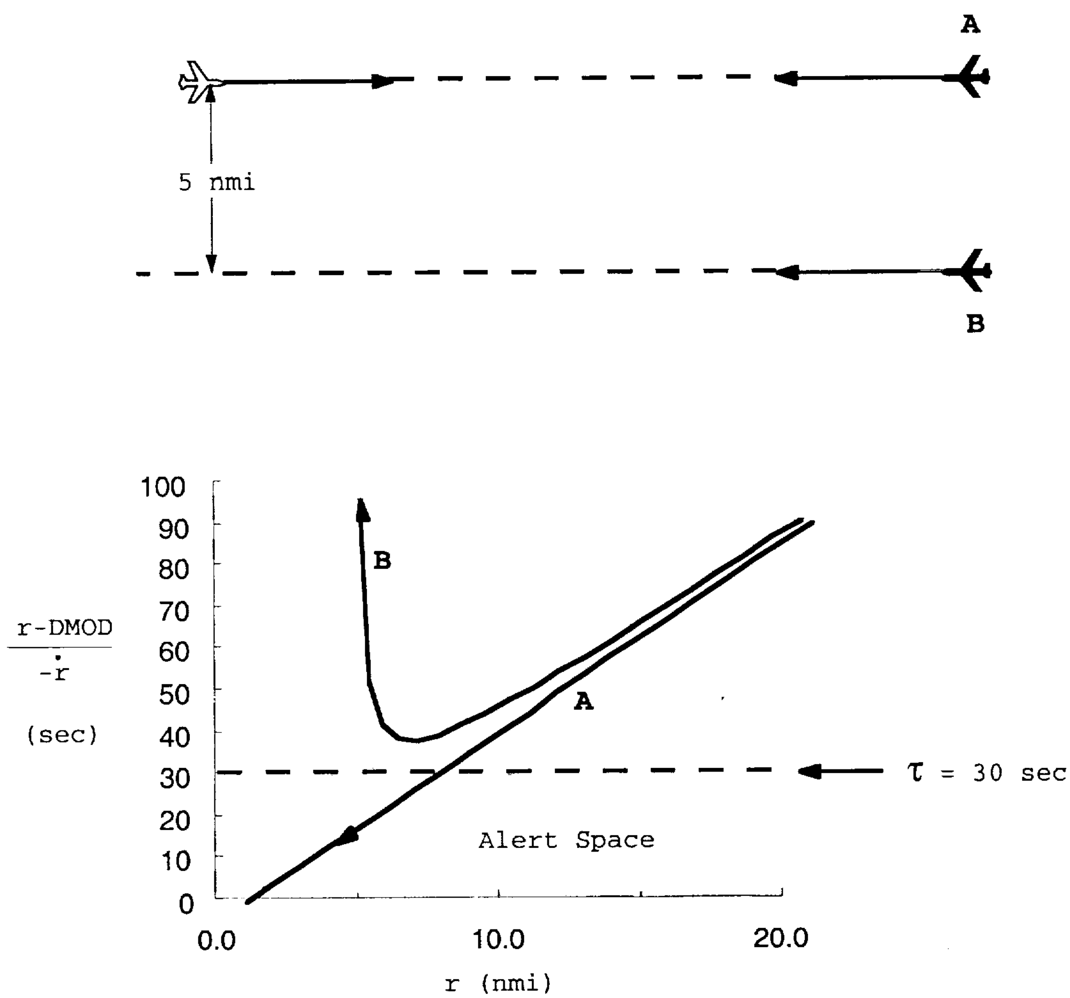


Figure 3-4: TCAS Example

Notice that for a range greater than about 10 nautical miles, it becomes increasingly more difficult for the TCAS logic to differentiate between the two cases. The τ threshold is 30 seconds for this particular scenario, which is just below the lowest point for which case **B** would trigger an alert. Trying to extend the warning time of TCAS in its present form would only introduce an increase in false alarms as shown from this simple example. For instance, raising τ beyond 40 seconds would inevitably cause an RA alert for case **B** even though the aircraft would be expected to miss each other by 5 nautical miles or more.

The parameters DMOD, ZTHR, ALIM, and τ effectively determine the frequency with which RAs are given. Reducing these values will reduce the alert rate and number of disruptions caused by false alarms [Miller et al., 1994]. However, the tradeoff is the risk of missed alerts due to insufficient warning time. The desire is a balance between false alarms and collision protection that TCAS is intended to provide.

To achieve this balance, the various design parameters (e.g. DMOD, τ) were set using an iterative, trial-and-error approach run through literally millions of simulation scenarios involving many hypothetical encounter geometries [Miller et al., 1994]. Modifications were also made from data and user comments during actual in-flight operations.

3.2.1.2 Traffic and Collision Alert Device (TCAD)

The Traffic and Collision Alert Device (TCAD) is a low cost, low complexity conflict alerting system directed at the general aviation industry [Ryan and Brodegard, 1997]. Its algorithm logic for conflict detection is based simply on range and altitude deviations alone, pilot selectable in one of three sensitivity levels: en route, standard, and terminal (see Figure 3-5). The basic function of TCAD is provided in an audible alarm and a numerical display of the range and relative altitude whenever another aircraft penetrates the alert space set by the pilot. If multiple aircraft are encountered, the data from the most prominent threat (based first on altitude, then on range) is shown.

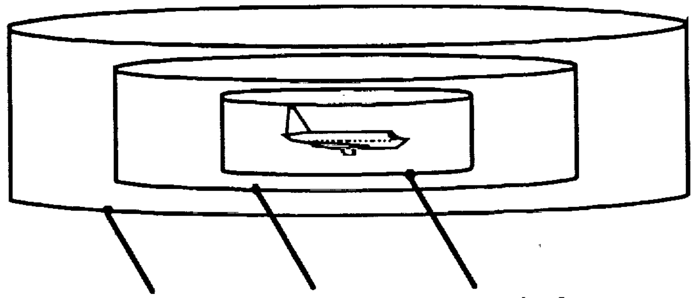


Figure 3-5: TCAD Alert Zones

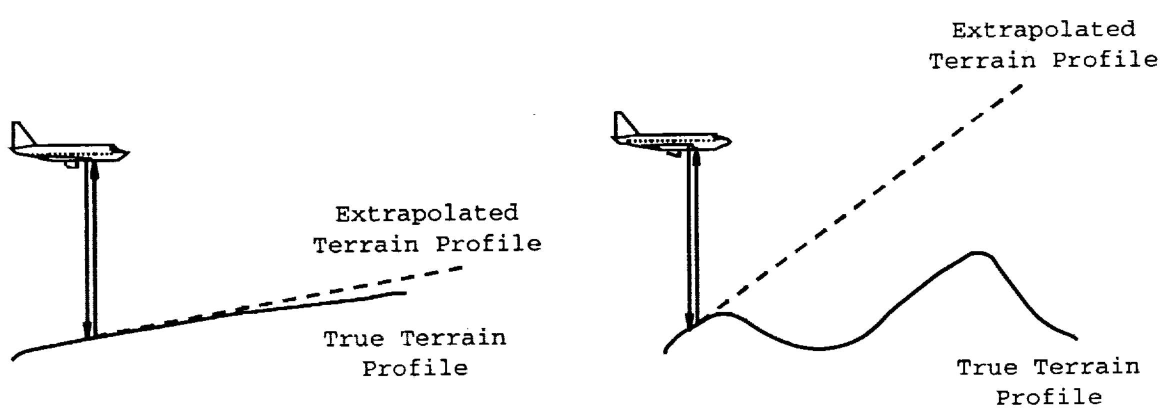
The threshold parameters (range and relative altitude) defining each alert zone are actually left to the discretion of the pilot. If the threshold is set too high, extraneous false alarms will result; if it is set too low, warning time is compromised. The result is a tradeoff between false alarms and missed detections which the pilot must optimize to his or her own preference.

3.2.1.3 Ground Proximity Warning System (GPWS)

Although the Ground Proximity Warning System (GPWS) is not designed to prevent traffic collisions, many of the problems encountered in alerting design can be seen its development (e.g. prediction of future hazards in presence of uncertainty, and tradeoff between false alarms and missed detections). In the case of GPWS, the hazard is terrain and the system is intended to prevent crashes while in controlled flight (no mechanical failures). The system has been mandated in jet transport aircraft in the U.S., and since its introduction in 1975, the number of Controlled Flight Into Terrain (CFIT) accidents has been reduced considerably [Kuchar, 1995]. However, CFIT accidents still occur worldwide and remain the leading cause of aircraft fatalities. Poor pilot response, either from delayed reaction or inadequate avoidance maneuver, was found to have contributed to many of these accidents. Previous experience with nuisance alerts was suspected to have played a role in a number of these poor responses [DeCelles, 1991].

At first glance, overly sensitive alerts appear to be conservative and on the safe side. The effects from false alarms seem to be primarily on efficiency. However, when alerts are judged to be erroneous or unnecessary, trust in the system validity becomes an issue. If the fallacy is excessive, then complacency may set in resulting in delayed pilot response times, inappropriate actions, or even no action at all. Thus safety becomes a direct fallout of false alarms when human operators are involved.

For GPWS, the dilemma comes mainly from the limited amount of terrain hazard information available to the system. Only the altitude from the current position, both Mean Sea Level (MSL) and Above Ground (AGL) is utilized in the calculations. It is based on only one dimension of the terrain – the altitude directly below the aircraft. No information is available to the GPWS regarding the terrain ahead or to the side. The GPWS must perform a derivative calculation from altitude separation alone and extrapolate the expected time to impact from this closure rate estimate. The prediction of the upcoming terrain hazard can be highly inaccurate from this information as shown below in Figure 3-6.



a) Accurate Extrapolation

b) Inaccurate Extrapolation

Figure 3-6: GPWS Measurement and Prediction of Terrain [Kuchar, 1995]

The lack of knowledge of the terrain features prominently shows the difficulty in the design and optimization of alerting systems when uncertainty is involved. In the case of the GPWS, the uncertainties come from the lack of information about the upcoming terrain features as well as the large variability that can occur in pilot response times and avoidance actions. These factors led to an iterative, evolutionary design, both from simulated scenarios and actual case studies, which proposes to balance unnecessary alerts and sufficient warning times [Bateman, 1994].

3.2.2 Additional Algorithmic Designs

There are many other methods which have been proposed to handle the conflict detection and resolution problem (see Appendix A). The problem is not limited to aerospace applications, but spans other areas such as automobile, naval, and robotic fields as well. Some are based on range, estimated miss distance, expected time to conflict, optimal escape paths, or force/potential fields. In all these cases, the element of prediction is inherently involved in determining the future behavior of the vehicles. The prediction occurs both in detecting a conflict and in resolving it. In many instances, the resolution requires the vehicles to comply exactly to the computed avoidance route in order to obtain full benefit from the calculated solution. However, as stated previously, there are uncertainties involved that will likely influence the final result.

Often, the solution is optimized and applied to only a few example scenarios. Most examples are given for 2-D horizontal conflicts, though it is noted sometimes that the methodology could be extended to the vertical dimension as well. Much of this is due to the relative ease of the solution in the planar case, especially for pairwise vehicle encounters where analytical solutions exist. Also, the system's overall performance can only be judged in real-life operation, or estimated in simulations over a large number of encounter situations. If done in simulations, the performance result is often stated in probabilistic terms such as false alarm and missed detection rates. Parameters in the algorithm are typically set to balance these performance measures. For example, the maximum look-ahead time of a conflict probe might be set to 20 minutes to minimize nuisance alerts (or in the case of TCAS, around 30 to 45 seconds), but still provide reasonable warning time.

3.3 Insights from Survey

Based on the review of the different methods of conflict detection and resolution, there were several important insights gained as described below.

3.3.1 Variety of Threshold Metrics

There did not appear to be a clear winner or single solution to the problem of conflict alerting. There were several different combinations of metrics that were mentioned, yet no analytical proof to determine the optimal set. Mostly, the variables utilized in the operational systems were the ones which were obtainable with the limited type of sensors available to the designers at the time of implementation. Some of the more commonly mentioned metrics used for conflict detection were range, range rate, altitude, altitude rate, expected time to closest point of approach, estimated miss distance, and probability of a conflict. Notice that all these variables have a natural correlation to the existence of a future conflict. Take range, for example. The likelihood that two vehicles would ever interfere with one another is intuitively higher if their separation is 10 miles, as opposed to being 100 miles apart (there is simply more volume of airspace possible for the future trajectory).

In general, the more information available for use in the alerting algorithm, the better the chances for improving prediction. For example, TCAS (which currently uses range and range rate in its horizontal threshold criterion) could benefit from additional data to predict expected miss distance [Burgess et al., 1994]. Also, one of the more talked about items in recent literature is the concept of intent information. This type of predictive information would allow for a better estimate of the future state trajectories, and thus, reduce the uncertainty in the entire conflict alerting process.

3.3.2 Prevalence of Ad Hoc Approach to Alerting Design

Though there were many possible metrics utilized in the different conflict alerting approaches, it was often mentioned that the appropriate setting of the thresholds was a tradeoff between overly sensitive nuisance alerts and adequate warning times. Where the process was actually described [Drumm, 1996; Bradley, 1992; Warren, 1997, Miller et al., 1994; Williamson, 1989], the settings were determined from an iterative, ad hoc approach using scenario simulations. The concept is shown in Figure 3-7.

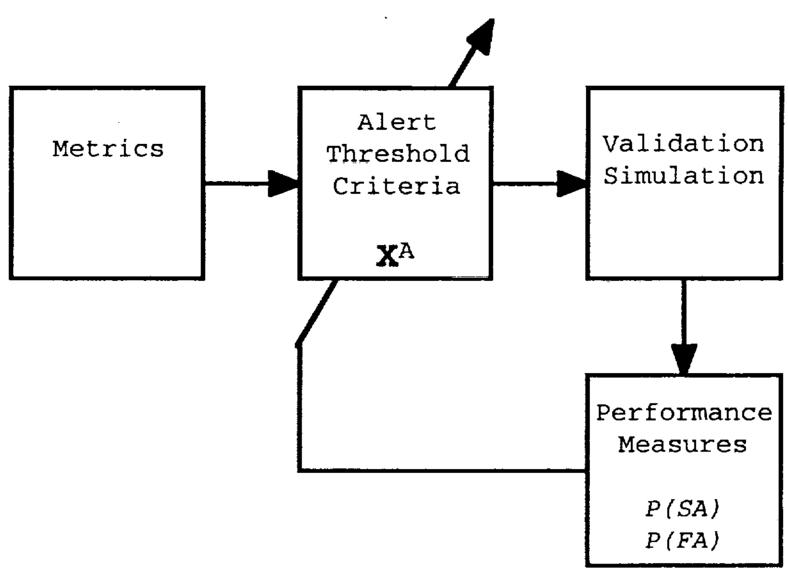


Figure 3-7: Ad Hoc Approach

Initially, the thresholds are set at predetermined values and put through a series of test scenarios. Often, these are done through Monte Carlo simulations and the thresholds are adjusted accordingly, depending on its performance. It is a cyclic process where the newly adjusted thresholds are repeatedly tested in simulation and the results tabulated with the alerting performance commonly judged by use of probabilistic means. The design must usually pass specified probabilistic parameters very similar to those already discussed in this work (e.g. false alarm rate, missed detection rate). If they are not met, the thresholds must be readjusted and the cycle repeats itself several times before it becomes satisfactory to the designers. In the progression of the alerting system through actual implementation, the thresholds will likely be altered again to compensate for problems encountered in the field. This type of threshold adjustment is considered an ad hoc process and can be expected to some degree from any design. Some references to this type of approach can be found in the TCAS design [Miller et al., 1994; Drumm, 1996; Bradley, 1992; Williamson, 1989] and the Airborne Information for Lateral Spacing (AILS) system [Winder and Kuchar, 1999].

3.3.3 Three Trajectory Projection Methods

The importance of the dynamic trajectory model in the conflict alerting system design cannot be understated, especially in the presence of uncertainty. As defined by Neelamkavil [1987],

> *"A model is a simplified representation of a system (or process or theory) intended to enhance our ability to understand, predict, and possibly control the behaviour of the system."*

The use of a trajectory model, either in the design of the thresholds or directly in the alerting logic itself, is simply to predict the occurrence of a possible conflict situation and the likelihood of avoiding it. In Chapter 4, the effects and consequences due to modeling errors will be discussed in detail, but for now, a quick look will be given to some of the more common dynamic modeling methods used in conflict detection systems.

Past approaches to conflict analysis have usually relied on one of two propagation methods in the modeling of future aircraft trajectories: 1) *single path* and 2) *worst case*. These two approaches can be considered *deterministic* models in that a definitive output or conclusion is produced for a given set of state inputs. A conflict either will occur, or it will not, resulting in a binomial output of a hit (1) or miss (0), respectively.

On the other hand, the outcome of a single event cannot usually be determined perfectly from a stochastic type model. For a specific set of state inputs, the result is not a precise outcome (the same inputs can produce a different output). There is some random variability involved and thus the output is usually expressed in terms of statistical properties or probabilities. Incidentally, a stochastic system will approach a deterministic one when the probability of the outcome is either 0 or 1 for each specific set of input states.

Take, for example, Figure 3-8, where two aircraft are in parallel, level flight with some fluctuations or variability in their flight paths. In one instance (Figure 3-8a), the aircraft never violate each other's protected zone. However, in Figure 3-8b, given the same set of initial aircraft state conditions, minimum separation is now violated at some future time. This is the concept of stochastic or probabilistic trajectories and is the main focus behind the methodology developed within this thesis.

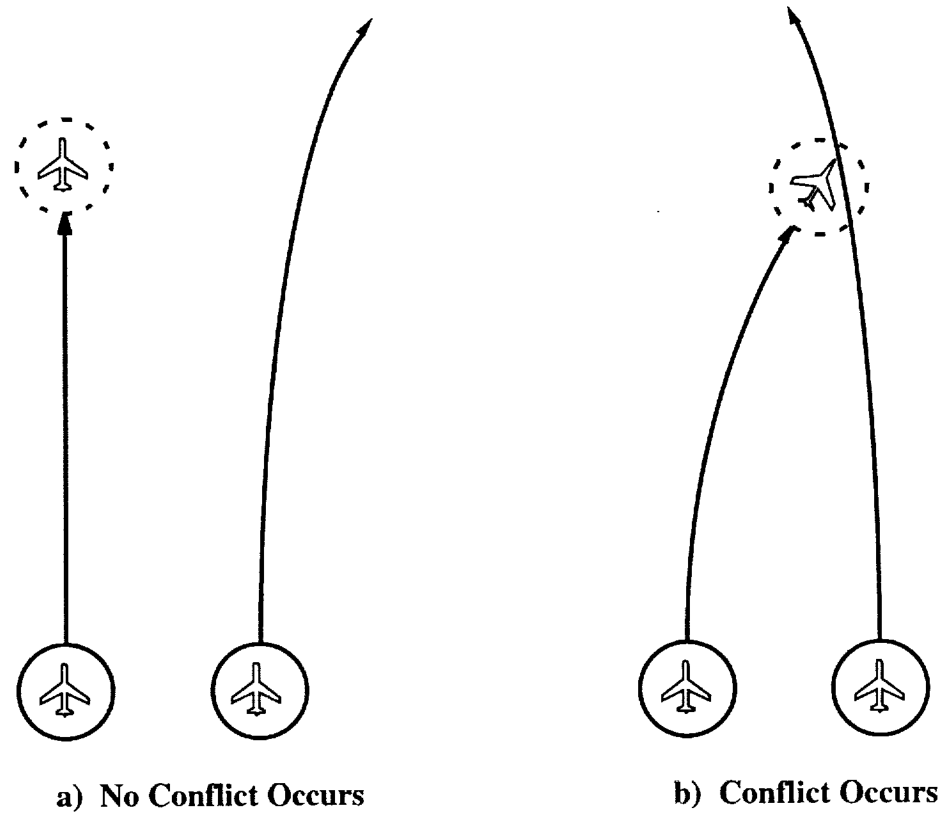


Figure 3-8: Example of Random, Stochastic Trajectory

3.3.3.1 Single Path Approach

In the single path approach, future aircraft positions are assumed to follow one specific trajectory (in many cases, usually along straight line projections from the current estimated velocity vector). This is the simplest approach and conflicts can easily be determined if the trajectory is not too complicated; for example, when heading and altitude changes are simply modeled as consecutive straight line segments. Figure 3-9 shows an example with two aircraft traveling in opposite directions. Assuming the current velocity vectors are held constant, the event of a conflict can be readily determined from analytical geometry. For the particular case shown in Figure 3-9, no conflict would be found. Many examples exist in the literature which utilize the single path trajectory method. A small sample set can be found in Andrews, 1978; Kosecka et al., 1997; Krozel et al., 1996; Sridhar and Chatterji, 1997; Bilimoria et al., 1996; Durand et al., 1995; Eby, 1994; Ford 1986, RTCA, 1983; Havel and Husarcik, 1989; Love, 1988; and Zeghal, 1994.

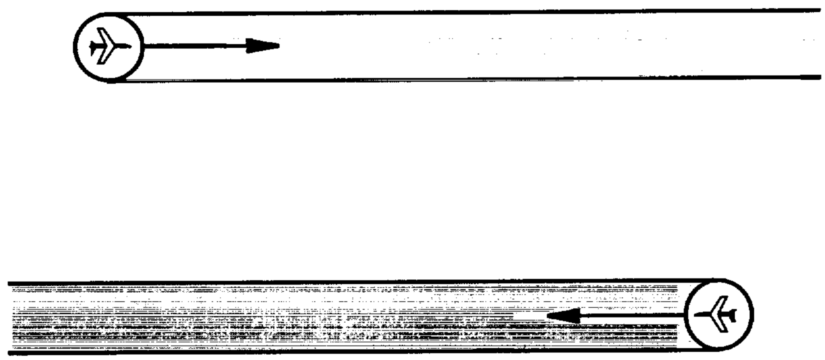


Figure 3-9: Example Single Path Approach for Conflict Prediction

It is obvious that the single path model does not compensate for uncertainties in the dynamic trajectory. The outcome is deterministic and results in either a hit (1) or a miss (0). In this type of modeling scheme, uncertainty and possible variability in flight trajectories are typically considered in the following stages of the metric definition or the threshold criteria using an iterative ad hoc approach to set the various threshold parameters, sometimes with appropriate safety buffers.

3.3.3.2 Worst Case Approach

In the worst case approach, every possible path (within expert reasoning) is considered and limited usually only by the aircraft's aerodynamic capabilities. The method is actually compensating for trajectory uncertainty within the dynamic model, but accuracy in estimating the occurrence of conflict is somewhat compromised (the concept of prediction accuracy is covered in the Chapter 4). Figure 3-10 is an example of this approach for the same situation as in Figure 3-9. Unlike the single path case, a conflict would be predicted here. The worst case assumption takes the conservative approach of underlining safety and it would seem most useful in short, time critical situations. In long term conflicts, it may be less practical since the swath of volume that could be encompassed by all the possible trajectories would be enormous. For instance, an aircraft that can be expected to climb or descend at 2000 ft/min would engulf an airspace 20,000 ft above and below it in after just 10 minutes. Previous work employing worst case analysis can be found in Ford and Powell, 1990; RTCA, 1995; Ratcliffe, 1988; Shepard et al., 1991; Shewchun and Feron, 1997.

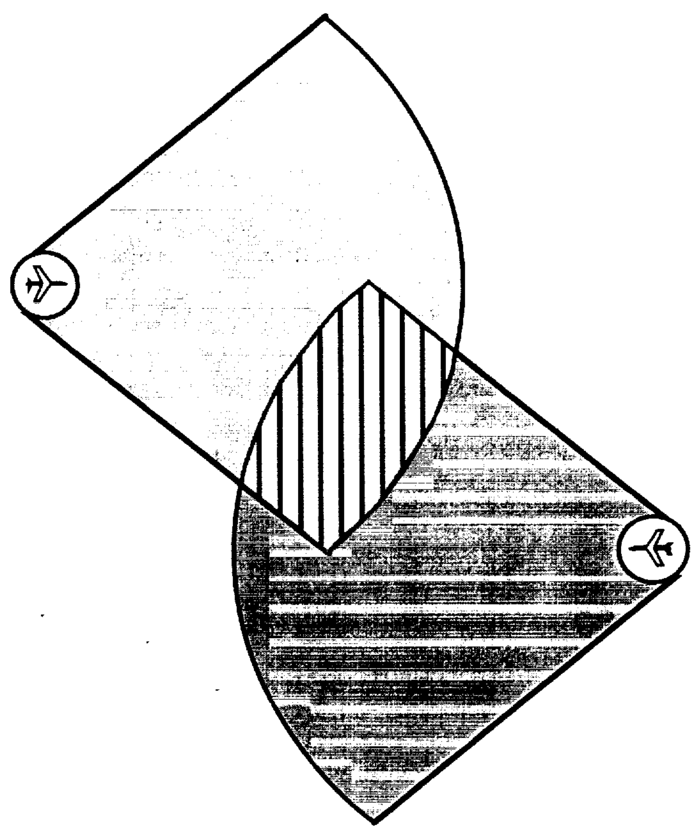


Figure 3-10: Example Worst Case Approach for Conflict Prediction

In order to confine airspace coverage, the prediction envelope is typically range or time limited at the metric or alerting threshold stages (e.g. look ahead time is bounded to less than 5 minutes or range to less than 50 miles). Again, this tends to lead to an ad hoc approach of maintaining a desirable coverage without being overly conservative. For example, would 5 minutes look ahead time still be too large, or would 3 minutes be more appropriate? Again, the answer may require simulations to establish a suitable medium.

3.3.3.3 Probabilistic Approach

In between the previous two modeling methods, however, is a middle ground where the various possible trajectories are weighted by their probability of occurrence. Uncertainties are modeled directly within the state estimates and the dynamic model. This approach has the distinct advantage of forcing an explicit, quantifiable measure of the uncertainty and accuracy affecting the conflict estimate.

Both the single path and worst case approaches provide either a hit (1) or a miss (0) in their evaluation of a conflict directly from the dynamic model. In the probabilistic approach, the prediction is in terms of the *likelihood* of a conflict, a value that can be explicitly attributed to the effects of uncertainties in the situation. A weighted value between 0 and 1 is determined referring to the estimated probability of a conflict, $P(C)$ in the future. It was shown in Chapter 2 that $P(C)$ is a direct indicator of (or link to) alerting performance. Thus this method provides a means for direct examination of the various levels and sources of uncertainty in the aircraft trajectory on the alerting system performance. Individual parameters can then be analyzed for their impact on the design.

To some degree, both the single path and worst case approaches can be considered subsets or special cases of the probabilistic approach (much the same way that a stochastic system can approach a deterministic one). In the single path method, it is assumed that the aircraft will follow a particular course with an absolute probability of 1.0 with no possibility of deviation, so the distribution is a single discrete point. For the worst case approach, any possibility of conflict in the desired region of interest is rounded conservatively upward to a $P(C)$ of 1.0. It does not really matter what trajectory distribution is inferred here, since any likelihood of intrusion would be considered off limits.

Though the probabilistic approach has been utilized sparingly in the past [Kuchar, 1996; Paielli and Erzberger, 1997; Heuvelink, 1988; Rome and Kalafus, 1988; Taylor, 1990; Bakker and Blom, 1993; Williams, 1993; Warren, 1997, Prandini et al., 1999; Innocenti et al., 1999], this thesis work expands the technique to a host of more complex encounter scenarios and adds a more theoretical basis underlying the purpose of the methodology. With the advent of high speed computing, the feasibility of the sometimes arduous or complex probability calculations is shown to be easily realized. This notion will be explained further in a later chapter.

3.3.4 Accounting for Uncertainties

In some approaches to the problem, the aircraft trajectories are assumed to be known exactly in 4-D. The determination of a conflict in this manner is relatively straightforward. Either a conflict will occur or it will not. Either a conflict can be avoided or it cannot. Approaches such as these are mainly concerned with obtaining a optimal solution based on some monetary or workload cost function.

However, in many cases, there is usually some leeway in the design approach to account for uncertainties that may occur in the prediction of a conflict and the ability to avoid it. Using TCAS as an example, the parameters such as DMOD, ZTHR, and ALIM act as buffers to account for uncertainties in the prediction process. This is a common approach with the single path projection method. The actual values of the buffers are set using an iterative, trial and error process (ad hoc) as discussed in Section 3.3.2. The final parameters are set from a balance between nuisance alerts (due to false alarms) and late alerts (due to missed detections).

In the worst case aproach, the uncertainties in the future path are accounted for somewhat in the path prediction, but not explicitly. It does not take into account the likelihood for which each of the paths would occur. All paths within bounds are considered equally likely. Also, a look ahead time limit is usually employed so as not to envelop too much open airspace. The actual boundaries are again commonly set using an iterative, trial and error process to set a balance between false alarms and missed detections.

In the probabilistic approach, the uncertainties are modeled directly within the trajectory estimation. This provides direct accountability of the sources of the errors in the prediction process. It is the most direct method of including uncertainties in the conflict estimation. As will be explained in a later chapter, the ad hoc approach is merely an indirect method of injecting uncertainties in the alerting system design

3.4 Summary

In this chapter, the problem of conflict detection and resolution was introduced. The resulting framework was the building block upon which subsequent discussions could be made. A survey of current operational and developmental approaches to the conflict alerting problem was also performed to provide insight into the design process and help determine underlying themes.

Chapter 4

A Unified Approach to Improving Alerting System Performance

As mentioned in the previous two chapters, the performance of an alerting system is often measured in terms of probabilities and prediction outcomes. For example, the false alarm and missed detection rates are indications of a system's ability to correctly determine the likelihood of an undesirable event (e.g. violating minimum separation). In this sense, the entire alerting problem can be perceived as a prediction problem in the presence of uncertainties. To improve performance is thus to increase the prediction accuracy of the alerting system. One way of achieving this is to reduce errors in the trajectory model. Another way is to make the future trajectory more easily predictable.

4.1 Errors in the Trajectory Model

4.1.1 Working Model (W) vs. "Truth" Model (T)

To be able to estimate $P(C)$ for either the nominal (**N**) or avoidance (**A**) trajectories, it is necessary for the alerting system logic to develop an approximate working trajectory model, **W**, for each aircraft as part of the estimation process. The detection of conflict is basically determined by where the modeling scheme predicts the position of the aircraft to be later in time. The importance of an appropriate model should not be understated since it is the defining source of the trajectory prediction. Errors in the model used by the alerting logic can increase the chances of missed detection of hazard and also add to false alarms. The choice of modeling schemes is already complicated by the existence of uncertainties in the aircraft trajectories as explained earlier. Common approaches for modeling were discussed in Section 3.3.3, but the following will explain in further detail the impact of the models on conflict prediction.

Let **T** be defined as the probabilistic state trajectory in the state-space of the alerting system. Thus, for a given state vector, $\mathbf{x}(t)$, the future state trajectory can be described by **T**. In this thesis, **T** will be referred to as the *"truth" model* and represents the best estimate of the uncertainties in the future states of the conflict situation. As shown in the example back in Figure 2-9, the state trajectory can have multiple outcomes from a single initial state, $\mathbf{x}(t)$. The unpredictability of a trajectory can be caused by many factors that are random such as wind, autopilot behavior, human actions, etc. and is thus a stochastic process. **T** can then be thought of as the trajectory defining the *true* probability of conflict, $P_T(C)$, at any given time.

Now let **W** represent the *working trajectory model* that is actually being utilized by the alerting system to estimate the future states of the system. The two models, **W** and **T**, are depicted in Figure 4.1. It goes without saying that ideally one would like **W** to be an exact copy of the true probabilistic trajectory, **T**, for all time t; thus, the probability of conflict predicted from the working model, $P_W(C)$, would be the same as the true $P_T(C)$. Unfortunately, uncertainties in the true trajectory make this all but unlikely except for a short time step into the future. In addition, as shown in Figure 2-8, multiple encounter situations with the same apparent state can lead to additional uncertainties in **W**.

Note that for now, the subscript **T** will be used to explicitly differentiate the true conflict probability $P_T(C)$ as opposed to that obtained from the working model **W**. However, in general and for subsequent chapters, it is impossible to actually utilize

$P_T(C)$ since the true trajectory distribution is unknown. The alerting decision is based on **W**, but the actual situation is really dependent on **T**. In the discussions on the SOC curves from the previous chapters, it was assumed the true stochastic trajectory, **T**, was known and being used in the plots.

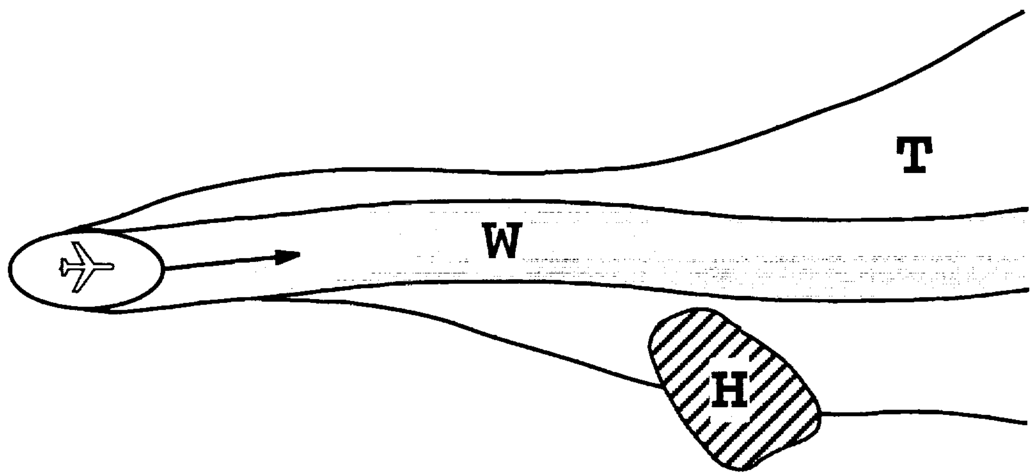


Figure 4-1: Working Trajectory Model (W) and "Truth" Model (T)

The error in the model, **W**, can be a result of many factors. It may be due to the limited amount of information available, oversimplification of the true trajectory, **T**, or even mistaken assumptions about the flight path. Sufficient and accurate state information about each aircraft is vital in the modeling process, but sometimes it is not available due to constraints on the equipment and technology currently available. For example, the Traffic Alert and Collision and Avoidance System (TCAS) is unable to obtain accurate relative bearing estimates of the surrounding aircraft because of limitations in the onboard equipment. It must rely on relative range and altitude data to estimate the conflict situation. Closure rates in the horizontal and vertical dimensions are estimated from derivative calculations and are used in the alerting decision. However, the relative bearing, which can be used to estimate the closest miss distance assuming a straight line projection, can not be obtained with sufficient accuracy using analytical means [Burgess et al., 1994].

As mentioned earlier, the single path and worst case propagation methods are common approaches taken in modeling the trajectories. Both can be considered simplifications of the true path of the aircraft. In the case of the single path trajectory method, **W** would be narrowed to a single track with probability 1.0. The aircraft would not be predicted to veer off this path. In the worst case method, **W** would encompass the realm of all likely paths with the addendum that any possibility of a conflict $\{ P_W(C) > 0.0 \}$ would be considered a violation. The single path and worst case approaches represent the two extremes of the modeling spectrum. Both have advantages and disadvantages which can depend on the situation. The single path approach may be preferred if the true trajectory is known with a high degree of confidence. The worst case approach lends itself to dealing with many of the uncertainties by setting a more sensitive alerting criteria.

However, whenever the working model **W** does not correctly correspond to the "truth" model, **T**, errors in the prediction of conflict will ultimately result. The more accurate the working model, the more likely $P_W(C)$ will approach $P_T(C)$, and the better the prediction of conflict. Ultimately, this will influence the potential performance of any conflict alerting system.

4.1.2 Errors and Uncertainties in the Trajectory Model

To obtain a better understanding of the effects from modeling deficiencies, an examination can be made by considering the position distributions of the aircraft from both the "truth" and working modeled trajectories at a given time t into the future. A representative depiction of these position distributions is shown in Figure 4-2. The area of **T** intersecting the hazard, **H**, is the true probability of conflict $P_T(C \mid t)$ at the particular time, t. The value predicted by the working model is $P_W(C \mid t)$. In general, however, the values need to be computed along the entire path of the probabilistic trajectory and not at just one particular instant in time.

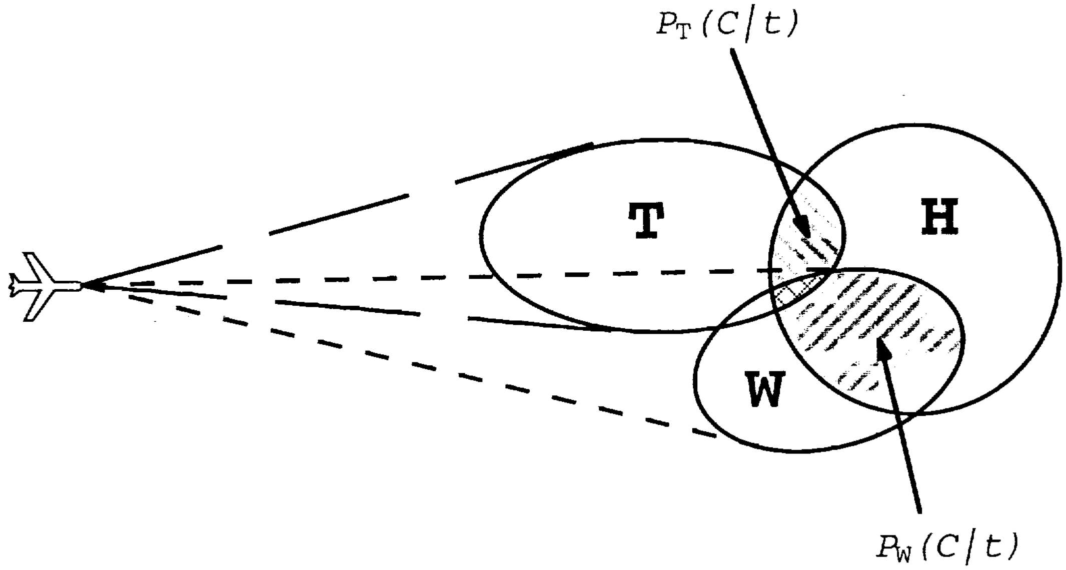


Figure 4-2: Prediction of Conflict at Time t

The effectiveness of the working model, **W**, depends on its ability to accurately represent the "truth" trajectory, **T**, and predict the value of the true conflict probability, $P_T(C \mid t)$. There are several ways in which the model will differ from the true distribution. In the same context as in the previous figure, Figure 4-3 shows the position predicted by the model to be in error by a *general displacement* from the true probabilistic distribution at time t. Because of uncertainties, it is natural to expect displacement error especially if the prediction time is long. Also, information such as a heading change, if not included in the model, will show up as a displacement. Most likely, the causes are a result of insufficient information either from sensor equipment or misunderstanding of the pilot's expected course of action.

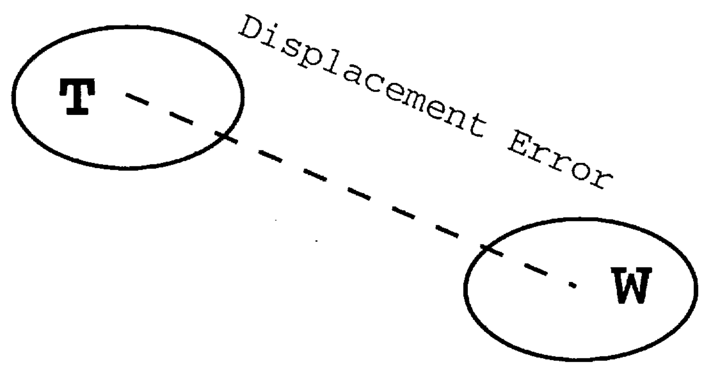
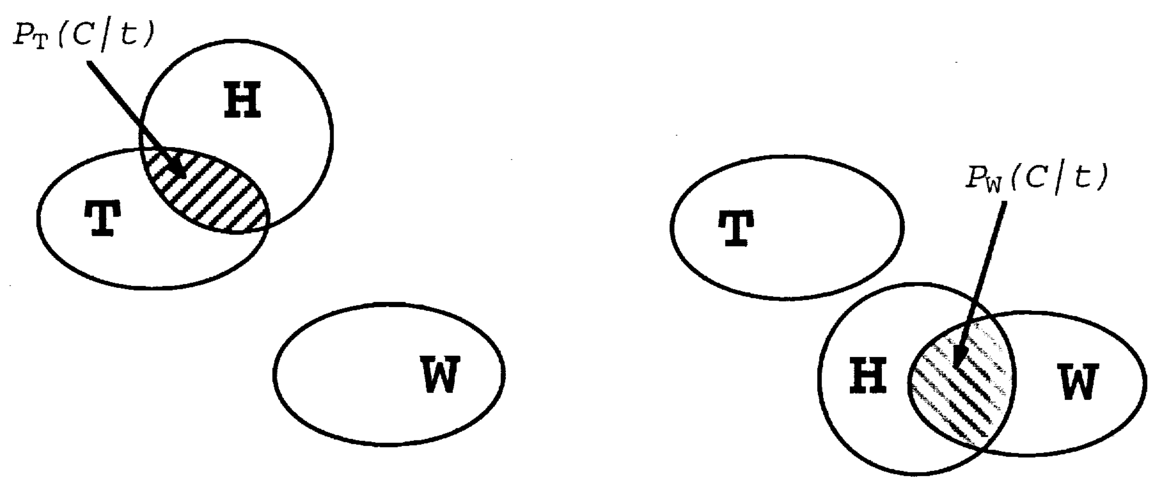


Figure 4-3: General Displacement Error in Working Model

Displacements can easily result in either a missed detection of a true hazard (Figure 4-4a) or a false indication of hazard (Figure 4-4b). Certainly, the ability to detect all forthcoming conflicts is of paramount importance. An alerting system would be basically useless otherwise. The effects of false alarms on the alerting paradigm was mentioned earlier in Chapters 2 and 3.



a) Missed Detection by Model
$P_W(C) = 0,\ P_T(C) > 0$

b) False Detection by Model
$P_W(C) > 0,\ P_T(C) = 0$

Figure 4-4: Incorrect Predictions from Displacement Error

General displacement errors are more likely to occur when a single path projection method is being used. The reason lies in the fact that the approach does not allow for variability in the predicted path. Even small inaccuracies in sensor readings (e.g. velocity, bearing) or wind variations can render significant displacement errors on the order of a mile in just 10 minutes. Generally, the approach used to compensate for the uncertainties and variations is to define a safety buffer zone about the predicted positions along the path. The net effect is much like a single tubular trajectory of constant width as shown in Figure 4-5. A conflict would then be declared if the hazard is predicted to pass within the specified distance from the modeled path. The main idea is contain some or most of **T** with the buffer region to compensate for the uncertainties of the fūture path.

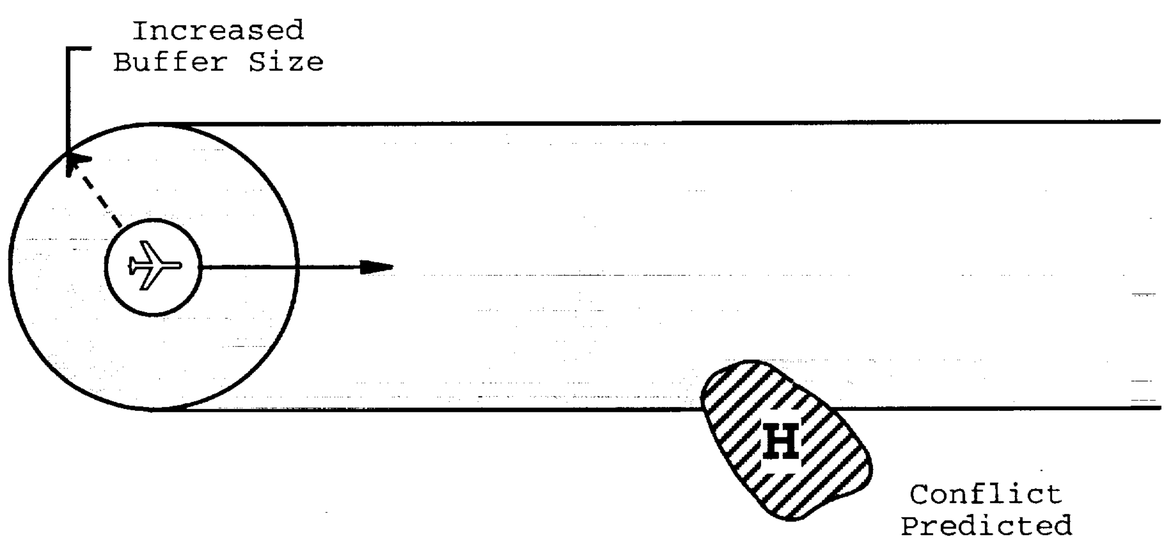


Figure 4-5: Safety Buffer Solution to Single Path Projection Approach

Differences in the shape and size of the distributions can also be expected due to uncertainties and incorrect modeling assumptions. In Figure 4-6, the working model's position distribution is shown much smaller than the true distribution, indicating not enough of the uncertainties were accounted for. Again, this situation would appear to occur often in the single path propagation approach. This type of occurrence can easily result in missed detection of the true hazard unless the conflict is very nearby and with high probability of conflict.

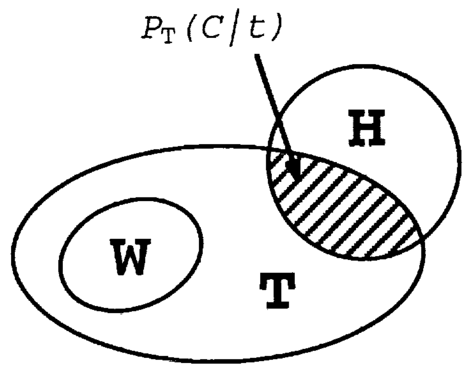


Figure 4-6: Modeled Distribution Too Small
Missed Detection by Model { $P_W(C) = 0$, $P_T(C) > 0$ }

In a similar fashion, the modeled distribution could end up being much larger than the true one (Figure 4-7) leading to excessive predictions of conflict when none exists - a condition leading to needless nuisance alarms. The worst case conservative approach discussed earlier is a possible example of this happening.

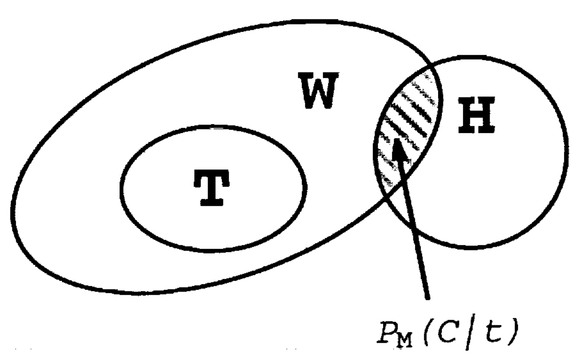


Figure 4-7: Modeled Distribution Too Large
False Alarm { $P_W(C) > 0$, $P_T(C) = 0$ }

Errors from the working model just add to the difficulty already present from the uncertainties within the "truth" model, **T**, and will only lead to higher rates of erroneous alerts or missed detection of conflict. The performance of the trajectory model can be defined simply as the difference between its predicted probability of conflict and the true probability of conflict.

$$\Delta P(C) = |P_{\mathrm{W}}(C) - P_{\mathrm{T}}(C)| \tag{4.1}$$

In actual application, the working trajectory model is what is used in the conflict analysis. Whenever $\Delta P(C) \neq 0$, additional errors in the hazard assessment are incurred and only increase the difficulty in deciding the appropriate action to take. Ideally, one would prefer the working model to match the uncertainties in **T**, but usually this will not be the case. Methods which only set $P_{\mathrm{W}}(C)$ to be either 0 or 1 will cause some important information to be lost in the analysis, especially effects of the inherent uncertainties in the future trajectory on alerting performance $\{P(FA)$ and $P(SA)\}$. The significance of calculating these parameters is not only to have the ability to estimate the performance of the alerting design, but also to examine the benefits from reducing the various levels of uncertainty in the trajectory prediction.

4.1.3 Effects of Modeling Errors on Performance Estimates

The effect of errors in the model can severely hamper and alter the design and analysis of the alerting system. A poor model does not represent the true situation at hand and can lead to uninformed decisions based on inaccurate data. It becomes much more difficult to fully assess the current conflict state and set an appropriate threshold. The SOC curve would be deceiving and may show the conflict situation looking better or worse than it really is. Without decent trajectory models, setting good alerting thresholds then becomes an achievement merely by chance.

In cases where uncertainty in **W** is modeled as smaller than in **T**, the faulty SOC curve will show up as being better (moving more toward the corners). Recall that when there is no uncertainty, the operating point must lie in one of the 4 corners. This gives a false sense of security that does not really exist. The single path modeling approach has basically this net effect. It assumes very little, if any, variability in its path prediction; thus it is very forthright in its estimate of conflicts - it either exists or it does not (binary). It is a mere simplification of the problem that if that situation (i.e. no uncertainty in the trajectory) was truly reflected in the conflict, then alerting would be greatly simplified.

In an alerting system, the threshold is based on the working model, **W**, being employed by the logic, but the actual performance is due to the probabilistic trajectory, **T**. Thus, two SOC curves really exist – one based on the designed tradeoff (due to **W**) and the other on the true tradeoff (actual **T**). A point on the **W** curve maps to some point on the **T** curve as shown by the dashed line in Figure 4-8 (the mapping need not be one-to-one, however). An inaccurate model may then induce an alert with results not expected by the analysis using the modeled **W** trajectory. In the particular case of Figure 4-8, the actual performance of the system will have a higher false alarm rate and lower successful alert rate than the intended design.

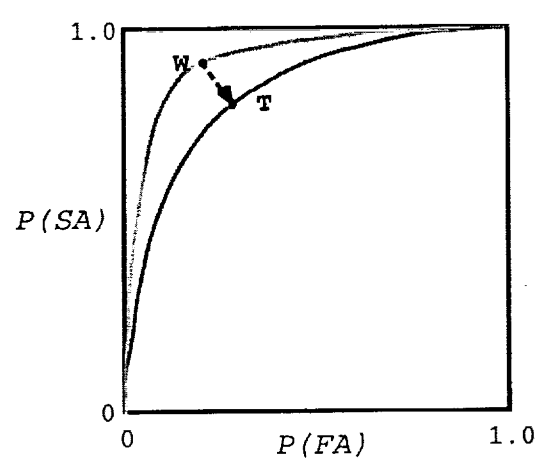


Figure 4-8: Effect of Underestimating Uncertainty

Overestimating the uncertainty of the "truth" trajectory, **T**, can also lead to problems. When the uncertainty in **W** is taken to be larger than in **T**, the SOC analysis will have the appearance of being in a more difficult conflict situation than it really is. When more uncertainty exists, the curve will tend more towards the diagonal line from the lower left corner to the upper right (Figure 4-9). The effect may cause one to alert earlier than necessary believing successful avoidance would be jeopardized otherwise; when in fact, it would be the false alert rate that is really compromised (alerting too early increases risk of false alarms).

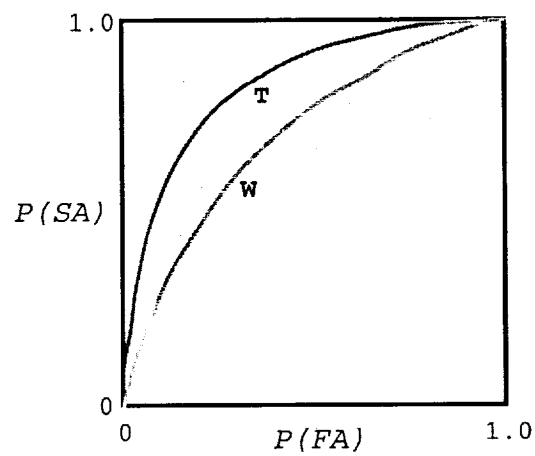


Figure 4-9: Effect of Overestimating Uncertainty

4.2 (W = T) Reducing Trajectory Modeling Errors (Increase Accuracy of Model)

The general effects of modeling errors (which were just explained) can lead to misrepresentation of the actual hazard at hand. <u>In order to effectively make well informed alerting decisions, one must have an appropriate assessment of the current threat situation</u>. The objective is to increase the prediction accuracy of the trajectory model, **W**, used by the alerting logic. This is achieved by increasing the degree to which **W** = **T**.

It is extremely important to keep in mind that the "truth" model of the trajectory, **T**, is still considered a stochastic process with a random outcome for a single initial realization. If this were not true, then **T** would not have uncertainties, and the above relation would imply that an ideal alerting system could almost always be designed (no false alarms, always safe avoidance). The statement **W** = **T** simply denotes that the two trajectory distributions should exhibit the same parametric characteristics of a random process such as the mean, standard deviation, and dispersion form. It indicates the desire to properly match the alerting logic's estimate of a hazard to the actual, and allows the accurate assessment of the true probability of conflict, $P_T(C)$, and thus $P(FA)$ and $P(SA)$ as well. The SOC method requires a curve to match as best it can to the true conflict situation, else the curve would be wrong and misguided as is shown in Figure 4-10. This is true of any performance analysis method.

The natural randomness in the problem still has to be dealt with, however. For any single event, such as an alert, the outcome is still probabilistic even if **W** = **T**. Take, for example, a flip of an unbiased coin. Even if the model of the outcome is perfect (50% chance of heads, 50% chance of tails), for any given toss, there is a 50% chance of being wrong - akin to a $P(FA) = 0.50$. For any given alert, the outcome of succeeding or giving a false alarm is still probabilistic (the trajectories are random processes with an associated variance). This is the notion of *inherent* uncertainties and will be dealt with in the next section.

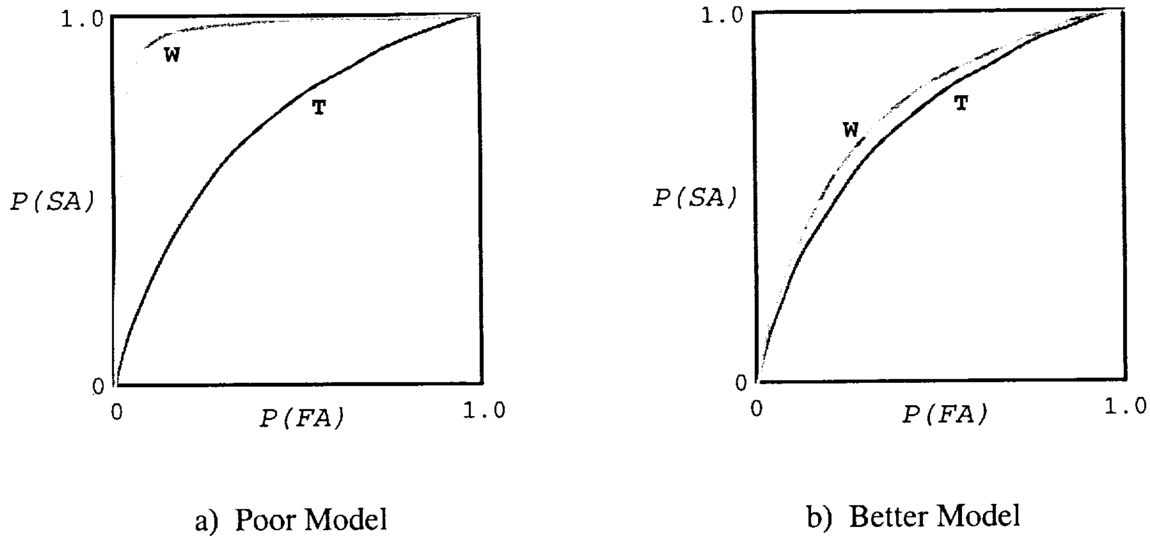


Figure 4-10: SOC Comparison

For now, the concentration will remain on increasing the trajectory modeling accuracy, **W** = **T**. In Figure 4-11, a schematic for two approaches to this theme is given: driving **W** toward **T**, and driving **T** toward **W**.

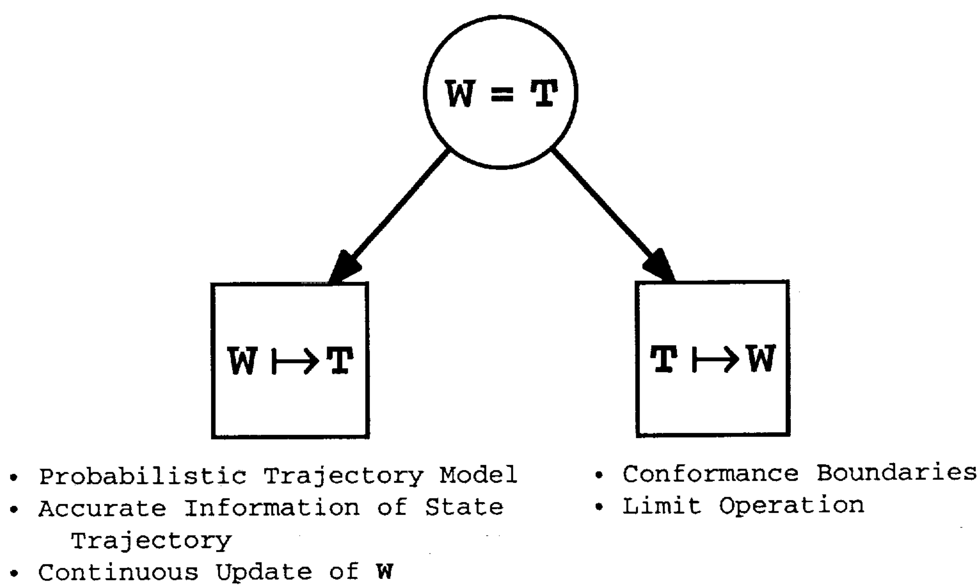


Figure 4-11: Approaches to Reducing Modeling Errors

4.2.1 (W $\rightarrow$ T) Drive W Toward T - Improve Trajectory Modeling

4.2.1.1 Utilize a Probabilistic Trajectory Model

The notion that **W** should equal **T** would imply that **W** should also exhibit characteristics of uncertainties, thus leading to a probabilistic modeling approach for the dynamic model of the alerting system. It is just one way of improving the accuracy of the conflict prediction process. This is the direct approach of dealing with the uncertainties - by independent modeling of the various sources and causes, quantitatively and explicitly. This includes uncertainties in flight path due to human factors and possible blunders.

The single path and worst case approaches, in general, are more likely to give inaccurate predictions of conflict (modeling errors), and must therefore account for the inaccuracies with other methods of instilling uncertainties into the analysis as explained in Chapter 3 (e.g. iterative modification of the threshold parameters and criteria). The goal of using a probabilistic trajectory for **W** is thus to directly reduce the chance of modeling errors in the dynamic modeling phase of Figure 3-2.

4.2.1.2 Utilize Sufficient and Accurate Information of the State Trajectory

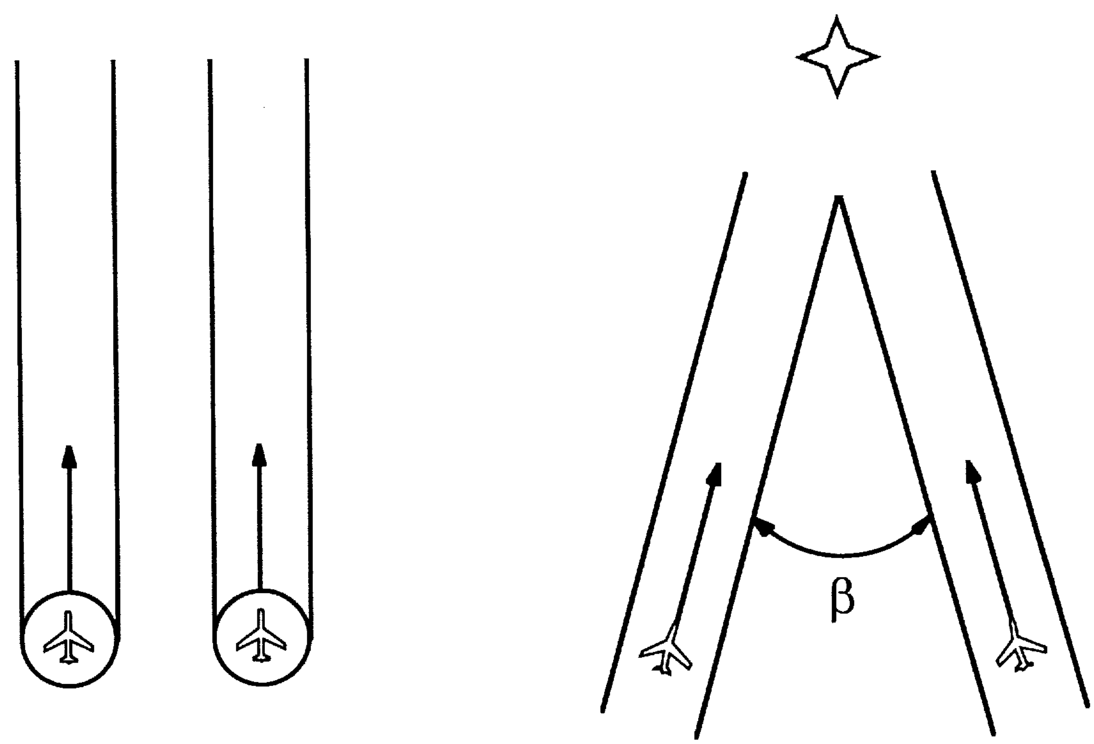
In developing a probabilistic trajectory model, **W**, one would prefer to have it coincide as much as possible with the random characteristics of the true trajectory, **T**. In order to do so and thus reduce the amount of error in the conflict prediction, sufficient and accurate information regarding **T** is necessary. In other words, it would be preferable to have more and better prediction information, assuming it is not in error. Else, the overreliance on a false assumption of the trajectory model can lead to problems.

Take for example, the current velocity vector of an aircraft. Using it to predict the position 5 minutes in advance would lead to a large displacement error if the aircraft was currently banked in a turn. Now adding the bank angle to the prediction would improve matters some, but only for a short extrapolation time because it is unlikely the plane would continue to bank in full circles. The addition of a commanded heading set by the pilot (intent) would be very helpful in this case.

Though it is highly unlikely that all the parametric estimates of **T** are available to an alerting system for modeling **W**, a sufficient amount of information should be utilized in order to give a reasonable estimate of the hazard situation. Sufficient is subjective and really depends on the goal of the system and the specific situation at hand - as it is with any type of modeling scheme.

The amount and type of aircraft state information necessary for the trajectory model **W** is somewhat arbitrary. Past conflict avoidance methods (see Kuchar and Yang, 1997) have shown the use of a variety of different combinations of state variables. The only state variable usually required is with regard to positional data since the conflict criteria are based on separation standards. All other state variables would certainly help improve the prediction of the future trajectory if utilized appropriately. However, sometimes the additional information may not be necessary as the impact on performance might only be minimal. It really depends on the situation in which the alerting system is operating.

Take the two examples shown in Figure 4-12. Figure 4-12a shows two aircraft in a simultaneous parallel approach, and Figure 4-12b has two aircraft converging on an intersecting waypoint along two separate airways angled at a certain β degrees apart. As long as the alerting system is <u>designed and limited</u> to a very specific type of encounter situation, it is possible to have a reduced number of variables to define the working model **W**. This is because of the implicit information already embedded within the specific scenario - the relative bearing in the case of Figure 4-12. It is conceivable that a dynamic model may not even be necessary as the relative position between aircraft may be sufficient enough to differentiate threat and non-threat situations. This is analogous to mapping or regression modeling where the number of variables required to sufficiently model a problem increases with the complexity and the uncertainty of the environment. If the alerting system is needed to handle many types of possible encounters (such as aircraft coming from all directions), then it becomes more necessary to be able to differentiate between the situations and more defining variables are required. A more thorough discussion on this topic will be given in Chapter 5.



a) Parallel Approach b) Intersecting Airways at Waypoint

Figure 4-12: Example Encounter Situations

In general, the more sensors and information available about the aircraft and hazard (such as from data-link), the better the ability of the system logic to match **W** to **T**. As previously discussed, the lack of certain information has been a recurrent problem in several aviation related alerting systems such as the GPWS and TCAS. However, caution should be used in assuring the information is utilized properly so as to truly match **W** = **T**. Else false reliance on the data will only lead to misguided decisions. The single path approach to modeling, for example, usually assumes a constant velocity even in vertical maneuvers, yet the velocities have been observed to fluctuate significantly especially in climbs and descents. Also, projecting a vertical maneuver over a long period of time is usually meaningless since the aircraft would be expected to level off sooner or later. This leads to the following sub-section of utilizing intent information to further improve prediction accuracy.

The use of additional *intent* information has been a recent topic in conflict avoidance lately, though an exact definition as to its meaning has not been thoroughly presented. <u>For this thesis, intent will be taken as any information that will support the prediction accuracy of future aircraft positional states</u>. Commanded heading, level-off altitude, or next waypoint could be considered intent information, though they must accurately depict the current situation in order to satisfy **W** → **T**.

Intent can also have uncertainties and must be adequately accounted for or else its limitations must somehow be expressed in the analysis. Take, for example, an aircraft (**B**) with the expected intention of leveling off 1000 ft above another aircraft (**A**) as shown in Figure 4-13. If aircraft **B** only has a 75% chance of following through with the level-off, the alerting system must acknowledge this information and decide when to alert to supply sufficient clearance to avoid the hazard and still hold false alarms at bay.

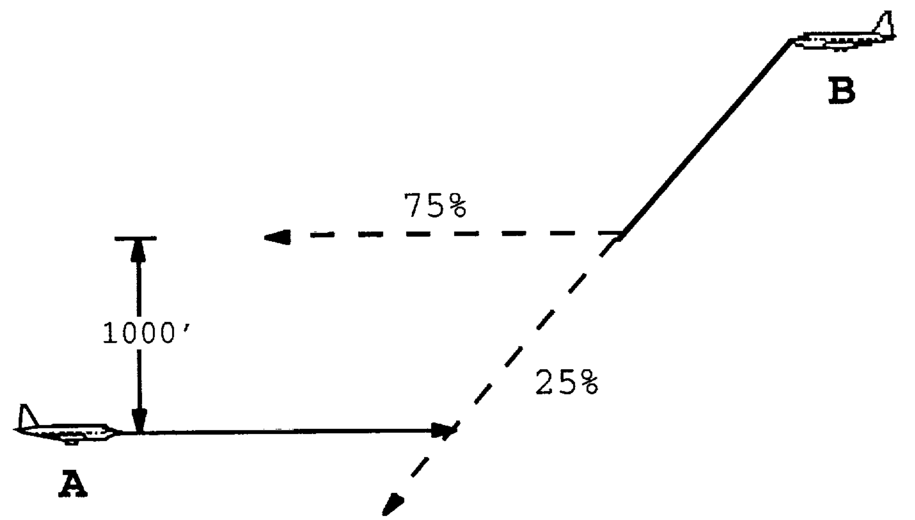


Figure 4-13: Uncertainty in Intent Information

Another example is the parallel approach scenario where aircraft are landing simultaneously on two different but closely spaced runways. The intent is obvious - fly straight, at a low rate of descent, down to the respective runway. However, if this intent were 100% foolproof, then aircraft could be allowed to be spaced extremely close laterally (subject to wake vortex constraints) if sufficient guidance and sensor systems were onboard (i.e. use of Differential GPS). The problem occurs when human pilots blunder or when weather, wind, or equipment failure become an issue. Spaced too close, the parallel runways may not provide the aircraft with adequate time to avoid a conflict if the intent is not followed. Overconfidence in the use of intent information can lead to trouble if it does not accurately depict the true uncertainty of the trajectory, **T**.

4.2.1.3 Update the Working Trajectory Model

In order to consistently obtain accurate predictions of the current threat situation, the dynamic model, **W**, should ideally be continuously adjusted to match changing conditions. It should really be adaptive to different state or intent information that becomes available. Recall that the TCAS Tau Criteria thresholds actually change depending upon altitude, climbing or descending, and direction of flight (see Figure 3-3). Though these values were designed using hand-picked simulation scenarios, they prove the necessity of adjusting for varying flight conditions.

Take an example where an aircraft begins to enter airspace where inclement weather is abundant. It becomes more likely now that the trajectory of the aircraft is more variable and may change course to avoid certain areas. The dynamic working model, **W**, should then allow for higher possibilities of aircraft varying off their present course. It should be updated accordingly so that **W** matches **T** as much as possible to reduce prediction errors. Assuming the aircraft will maintain a current heading, steady level flight when that behavior is unlikely will only lead to additional modeling errors. At the other extreme, over-modeling of uncertainties can also lead to incorrect predictions.

It is likely that the intent of an aircraft will change many times over the course of a flight. To assure proper conflict detection, the dynamic model should be coupled to the updated information and change accordingly. If possible, it would be best if a pre-check of the new intent trajectory was examined for immediate conflicts before allowing the intent to be carried out. However, this might be out of the scope of an alerting probe concept and more toward a centralized Air Traffic Management effort.

4.2.2 (T $\rightarrow$ W) Drive T Toward W

4.2.2.1 Utilize Conformance Boundaries

One way to improve algorithm prediction accuracy is to enforce *conformance* of the aircraft trajectory through the use of secondary alerts. <u>The principle is to alert not because of a conflict, per se, but because the aircraft is deviating from the model that</u>

would allow the conflict logic to more easily predict the outcome of a future conflict. This method can be especially useful when aircraft are closely spaced together and expected to fly a certain path or pattern. It makes it much easier for the conflict avoidance algorithm to work as expected with less error. A very simple working model, **W**, could be utilized (e.g. a single path projection) and the associated aircraft could be forced to conform to a specific route. However, it can be somewhat constraining to the flight of the aircraft since it is bound to restrictions on its path.

An example of conformance monitoring is shown in Figure 4-14 where an aircraft is expected to follow a specific path marked by conformance bounds. The boundaries usually provide some leeway for fluctuations from both environment, aircraft, and human variations. Both horizontal and vertical margins may be delineated so as to contain the path of the aircraft in 3-D. With future advancements, 4-D conformance monitoring (adding time) may be possible.

If the boundaries are exceeded by the aircraft, then an alert can be given to warn the pilot to return back on track. It is also possible to have a new path recomputed once the aircraft deviates from the original routing.

The usefulness and simplicity of the conformance approach is marked by the number of prototype conflict systems which utilize this method. Examples include the Center-TRACON Automation System (CTAS) [Isaacson and Erzberger, 1997], EUROCONTROL's Medium Term Conflict Detection (MTCD) system [Vink et al., 1997], the User Request Evaluation Tool (URET) [Wanke, 1997; Brudnicki et al., 1977], and the No Transgression Zone (NTZ) for parallel runway landings [Shank and Hollister, 1994; Carpenter, 1996]. Even motor vehicles on the inter-state highway system utilize a form of conformance check with the lane bumps indicating deviation outside of the current lane. Notice that a worst case approach to highway design would probably require individual lanes to be over a hundred feet in width, yet they are usually only about 12 feet wide.

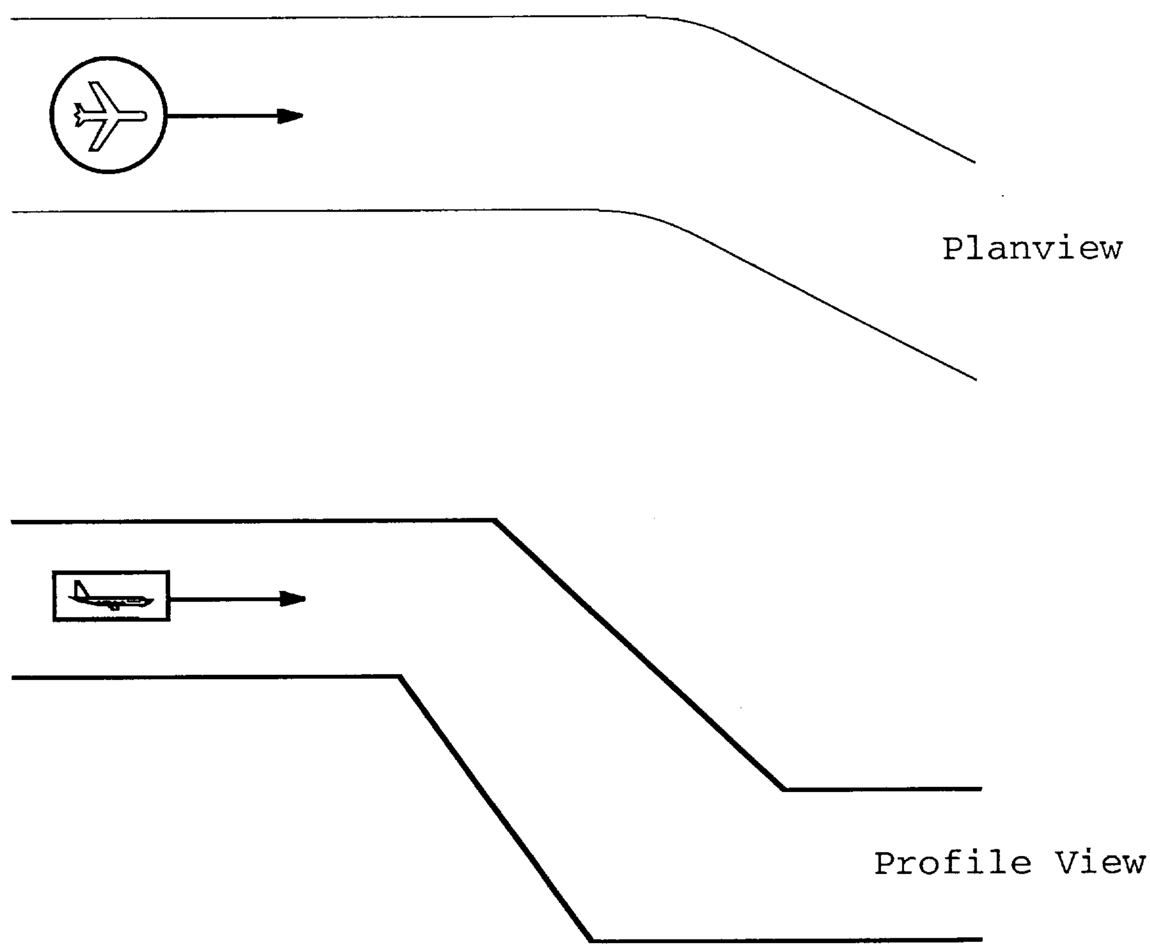


Figure 4-14: Example of Conformance Boundaries

To some extent, the current ATC system with its rigid airway structure can be thought of as conformance monitoring by the human controllers. Aircraft are maintained in their correct flight paths and place heavy trust in the controller to alert and clear them out of danger.

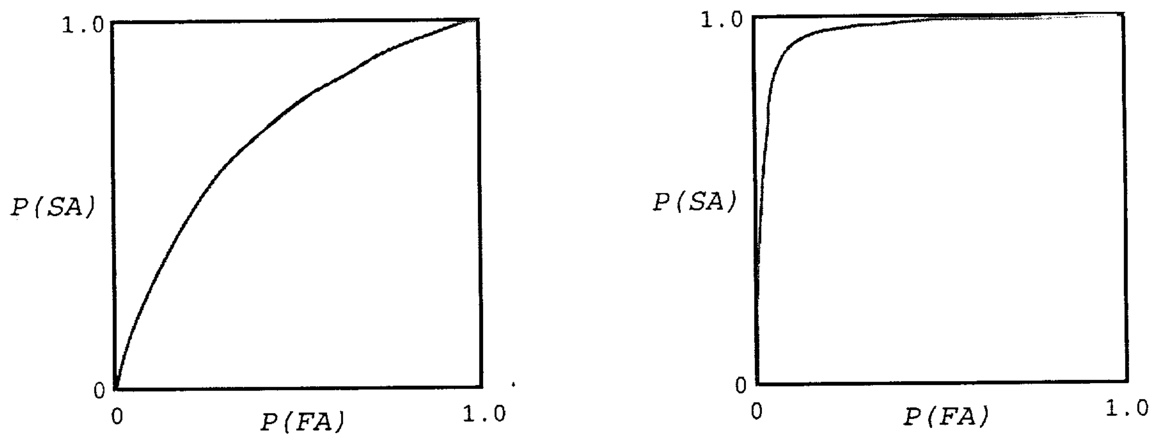
4.2.2.2 Limit Operation

Sometimes there are instances in which an alerting system may not work well and the thus the operation of the system will be limited to use for certain types of encounters only. For example, the TCAS algorithm is intended for near term conflicts on the order of less than 1 minute. Extending the original algorithm to long term encounters would likely incur deteriorating performance as shown back in the simple example of Figure 3-4. Without accurate bearing information, the TCAS logic simply cannot obtain satisfactory conflict prediction results in long term encounter situations [Burgess et al., 1994]. Thus the alerting system is limited to those operations in which it is capable of handling or predicting.

In another example, certain flight procedures are sometimes tailored to meet the limitations of the alerting system so that it may obtain adequate performance (e.g. minimize false alarms). For instance, in a situation often referred to as the "Seattle Encounter" [Drumm, 1996], an intruder aircraft is descending toward a TCAS equipped aircraft but levels off at an altitude above it. TCAS initially predicts a collision and issues an alert for the TCAS aircraft to climb. However, the intruder levels off above the TCAS aircraft resulting in a false alarm (the alert was not necessary). Not only has a false alarm occurred, but the situation can actually induce a hazard with TCAS aircraft climbing into the intruder. Because of the tendency to produce false alarms and a possible hazard situation, it has been recommended that aircraft slow their rates of descent when approaching their final altitude [Mellone and Frank, 1993]. The effect is a modification of **T** in order for the alerting logic to correctly anticipate (or predict) the leveling off of the intruding aircraft.

4.3 Reducing Inherent Uncertainties (Reduce Uncertainty in Future Trajectory)

Once the working dynamic model, **W**, is presumed to be a good representation of the "truth" trajectory, **T**, the decision to alert becomes a tradeoff between false alarms and successful avoidance. Assume the SOC plot of the current alerting design looks like that shown in Figure 4-15a. The curve designates the performance expected for various possible alerting points along a particular nominal path.



a) High Uncertainties

b) Reduced Uncertainties

Figure 4-15: Improved Performance Depicted in SOC Plot

The plot in Figure 4-15a shows a system that would not have great performance because of the difficulty in setting a threshold without sacrificing either $P(FA)$ or $P(SA)$. The effect is likely due to the amount of uncertainty in the true trajectories of the nominal (**N**) and/or avoidance (**A**) paths. It is an inherent characteristic of this alerting system given the random processes involved. This will be termed the *inherent uncertainty* of the system. In Figure 4-15b, there is less inherent uncertainty in the system.

To be able to improve the performance of this system, the uncertainties in the trajectories must be reduced so as to increase the prediction outcome for a single event in a random process environment. The situation is analogous to reducing the variance (σ^2) in a prediction. Take the coin flip example again. Inherent uncertainty is highest with an unbiased coin (the variance $\sigma^2 = p(1 - p)$ is maximum at $p = 0.5$). If the coin was biased toward 90% chance of heads and only 10% chance of tails, then the chances of predicting a head or tail with a single flip would be much better (variance is lower). The inherent uncertainty of this biased coin would be less than the unbiased one. <u>The effect is basically to make the outcome more deterministic</u>.

In terms of the SOC plot, reducing inherent uncertainty would drive the points to the outer perimeters making it easier to determine more suitable threshold locations. In the limit of no uncertainties, all possible threshold locations must exist at one of the corner positions as explained back in Chapter 2. For example, Figure 4-16 shows two aircraft on a nominal direct collision course in a 90° encounter situation. The SOC plot shows the deviation of the curve from the ideal position as the bearing accuracy of both aircraft is varied from $\sigma = 0°$ to $\sigma = 5°$ (normal distribution) in 1° increments. This simulation is based on a point-mass model of the aircraft flying along a straight path with the different random heading errors. The standard 5 nautical mile separation was used as the definition of conflict in this example, and the avoidance maneuver was a 20° turn after 10 seconds.

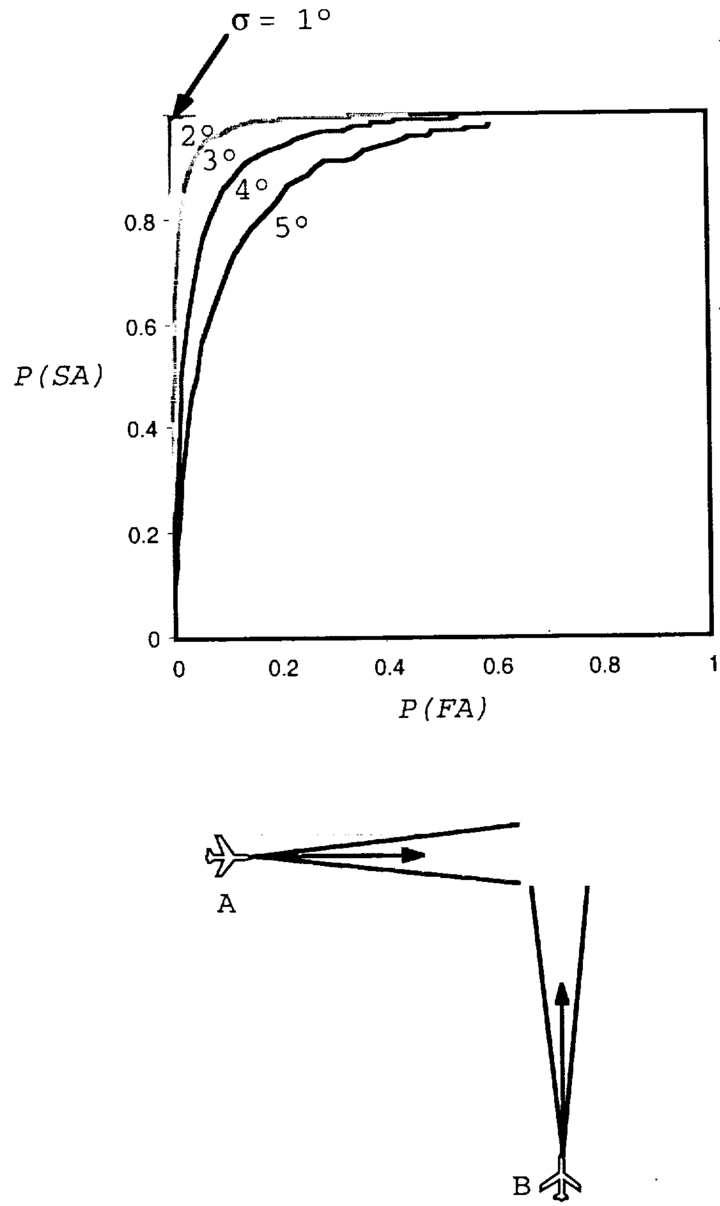


Figure 4-16: Example of Increasing Uncertainty (Heading) on SOC Curve

Inherent uncertainty is a function of the distribution and size of the "truth" trajectory relative to the hazard at the time of the prediction. The "truth" model can refer to either the nominal (**N**) or avoidance (**A**) paths. Analogous to the coin flip example, high uncertainty in the future path makes it more difficult to accurately predict the outcome for any single event. This can be seen in Figures 4-15 and 4-16 where higher levels of uncertainty make it more difficult to determine whether or not a conflict will occur. In the limit of no uncertainty, the conclusion is deterministic and binomial.

In the following sections, various ways of reducing the inherent uncertainty and thus improving alerting performance are given. It should be realized that these methods are not newly proposed ideas, but simply brought together to show that they all really fall under the category of reducing the inherent uncertainty of the underlying random process. The purpose, of course, is to increase the chances of correctly predicting the outcome of a conflict and the ability of avoiding it (i.e. to make results more deterministic).

Much of the effort is in reducing the uncertainty in the future track. Sensor inaccuracies, autopilot control behavior, weather changes, and variable winds are all contributing factors. However, much of the uncertainty involving the future path of an aircraft is a result of the human pilot in control and the decisions he/she makes; basically, not knowing what the pilot is going to do. Because of the high variability between humans, it is unlikely the course of action followed by each pilot would be the same in any given situation. Several methods are possible to help decrease these variabilities and are discussed below.

4.3.1 Restrict Flight Path

The position of an aircraft traveling at 450 knots with the ability to bank 30 degrees would, in just 6 minutes, encompass a nearly circular region of 45 nautical mile radius about its current position [Andrews and Hollister, 1997]. With a possible climb or descent rate of 2000 ft/min, the volume engulfed would reach 12,000 feet above and below the aircraft. Thus, in order to reduce the number of possible trajectories, some form of restrictions need to be placed on the flight path. Constraints on the trajectory would effectively narrow the region of uncertainty (see Figure 3-5b) and produce a more definitive outcome on whether or not a conflict would occur. In other words, $P(C)$, as well as $P(FA)$, would shift more toward 1 or 0.

This appears to be the simplest method and is currently implemented for enroute traffic today. Aircraft follow along in pre-assigned airways and at defined altitudes making it easier for pilots and controllers to predict potential conflicts. The system is not without shortcomings however. A tremendous amount of airspace is left under-utilized leaving many to believe a more efficient means of operation is necessary to handle the current congestion today and the increased air traffic demand in the future.

There is a movement under way to relax the current system of rigid airways and in-trail spacings to increase flexibility for more efficient operation. This notion of a *Free Flight* environment with less restrictions in course adjustments has been a source of much research and discussion lately [RTCA, 1995; Phillips, 1996]. Of course, the increase in flexibility will nonetheless increase the potential for more difficult conflict encounters and added uncertainty. The final report on Free Flight implementation by the Radio Technical Committee on Aeronautics [RTCA, 1995] suggests conformance to maneuver limits as an interim solution. For example, aircraft might be restricted to a 20 degree heading change within a 15 minute time period. Constraints placed on such maneuvers have been shown to significantly reduce the rate of conflict encounters [Andrews and Hollister, 1997].

4.3.2 Establish Protocol (Training, Rules of the Road, Convention)

The use of established rules-of-the-road with training can help increase the likelihood that pilots will follow specific patterns of flight behavior. The uncertainty in a pilot's action, and ultimately the aircraft's flight path, can thus be reduced. For example, rules-of-the-road regarding the right-of-way (depending on relative aircraft positioning) may help determine the expected trajectories each aircraft may fly. The importance of taking humans into account as a large contributor of uncertainties cannot be overlooked. A study of commercial jet accidents resulting in a complete hull loss has placed the flight crew as the primary cause in 70% of the accidents from 1988-1997 [Boeing, 1998].

Also, requiring communication between pilots or the controller prior to making any course correction would inevitably help reduce uncertainty in the expected flight trajectory. This has the notion of intent information but is explicitly forcing the pilots to decide and communicate immediate changes in intent before allowing the action to be taken. The idea certainly has merit and some have even proposed a new ATM environment where principled negotiation between pilots and controllers is the basis of an established protocol [Wangermann and Stengel, 1994]. The explicit communication of aircraft intent is a possibility for reducing the set of likely trajectories one can expect when deciding on potential conflict situations.

4.3.3 Introduce Better Equipment

Even with all the restrictions placed on the pilot to maintain a specific course, inevitably there will still be some random uncertainties involved. As mentioned previously, sensor inaccuracies, variations in wind, autopilot capabilities, and inherent aircraft dynamics all play a role in introducing variability in the aircraft trajectory. Empirical data from observing current aircraft maintaining a given track have shown deviation perpendicular to the nominal track (cross-track) that is approximately Gaussian on the order of 1 nautical mile standard deviation [Paielli and Erzberger, 1997]. Fluctuation in speed, due primarily to wind effects, was also observed to be upwards of 15 knots (one standard deviation) normally distributed [Paielli and Erzberger, 1997; Wanke, 1997]. Current technological advances in sensors (most notably the Global

Positioning System, GPS) and in computer hardware and software may be able to provide improved accuracy on future air transports. Future developments may also allow for 4-D path following capabilities.

The introduction of highly automated equipment onboard the aircraft is not without its critics, however, especially if it involves automation in the cockpit and new allocation of piloting tasks. The addition of new technology may likely alter the way aircraft are flown and may introduce new modes of human error. A more thorough examination of the underlying human-machine process would thus be necessary and is an area of continuing research.

4.3.4 Delay Alert (Minimize Size of T at Alert Time)

Though reducing the uncertainty in the trajectory would be ideal, sometimes it is not always possible. It may be impossible to alter the operating environment or add sophisticated instruments and sensors to reduce the variability in the flight path. To the alerting system designer, the only remaining alternative for decreasing the unnecessary alert rate is to delay the alert as long as possible. In effect, the delay is used to wait for a more definitive determination of a conflict before proceeding to warn the pilot and/or controller. <u>This has the effect of minimizing the uncertainty in the trajectory at the time of the alert</u> (the position distribution of **T** is smaller). Usually, the uncertainty in the position of an aircraft will grow with time (the exception might be 4-D trajectories). This effect is depicted in Figure 4-17. The shorter the prediction time, the smaller the uncertainty. Examination of Equation 2.2 shows that as $P(C)$ tends towards 1, $P(FA)$ will approach 0. Delaying the alert also has economic and reduced workload benefits as well, since nuisance alerts incur costs from maneuvering unnecessarily.

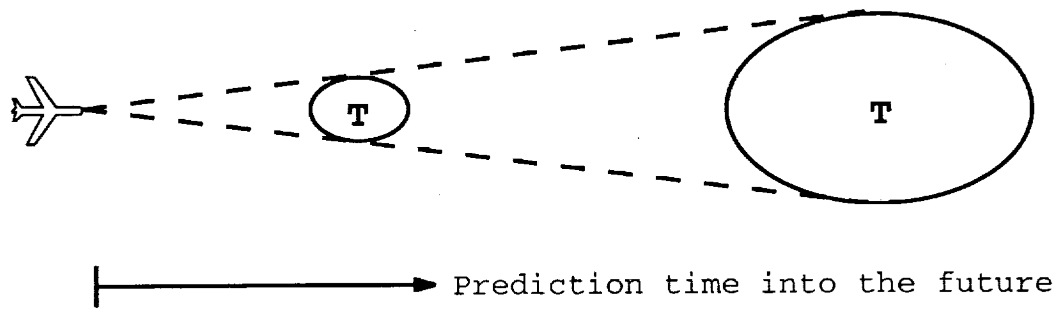
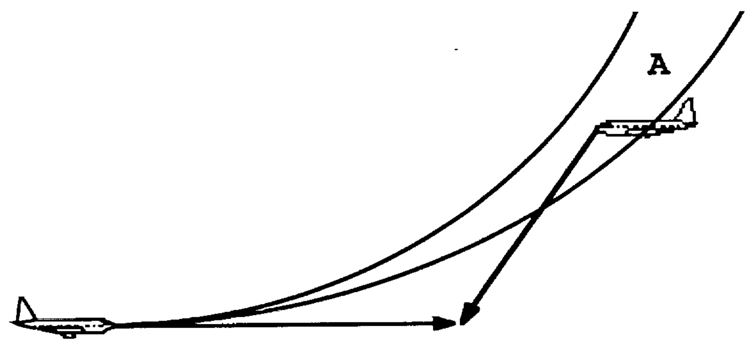


Figure 4-17: Increase in Uncertainty Due to Prediction Time

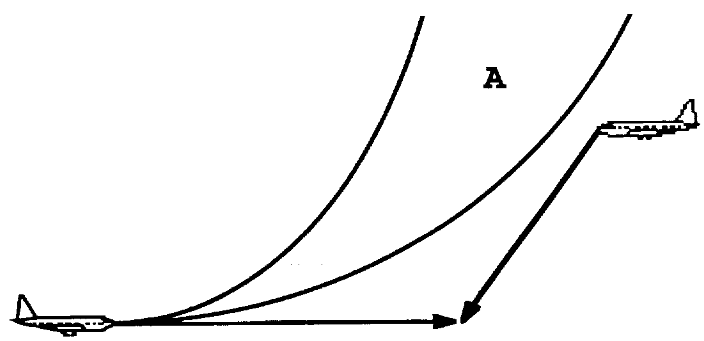
Of course there is a tradeoff to all this; namely that the delayed alert may be placing the aircraft at a higher risk of danger. The pilots would have less time and options to undertake avoidance action when the alert is given. It may also be possible that the human pilot will disagree with the delayed timing of the alert, thus resulting in mistrust of the system as well [Pritchett, 1999].

4.4 Investigate Other Avoidance Maneuvers

If the performance of an alerting system still proves insufficient to meet design goals given the methods discussed above, then another option is to utilize a different conflict resolution. The determination of $P(SA)$ discussed in the last chapter is specific to a given avoidance trajectory, **A**, which, as a reminder, still includes uncertainties and is a stochastic process. Various horizontal or vertical maneuvers could be tried including the addition of speed control or cooperative maneuvering between aircraft, or a more severe or drastic maneuver could be employed. Figure 4-18 shows how a larger climb rate could be used to increase the chances of avoiding another aircraft descending into a conflict situation.



a) **Original Climb Maneuver**



b) **Increase Climb Rate**

Figure 4-18: Utilizing a Different Maneuver to Avoid Hazard

However because of human involvement, there may be higher uncertainties associated with more complex maneuvers. The result may be worse performance than expected because of larger uncertainties in the avoidance trajectory, **A**. The pilot may also be less willing to perform complicated or severe maneuvers if for some reason he/she did not deem the threat to be real. In any case, it is always good to have viable, multiple avoidance options available to the pilot during emergency situations for added safety.

4.5 Design Issues

Reducing modeling errors and reducing inherent uncertainties results in one effect - it increases the ability to predict the outcome of a single stochastic event. The idea is, in effect, making the process more deterministic and thus increasing the performance of the alerting design. Any of the methods discussed above can be used individually or in combination.

To improve performance, one can either reduce the uncertainties in **T**, and then design an alerting system to match (reduce **T** and **W** $\rightarrow$ **T**), or one can reduce the uncertainties in **W** and then enforce trajectory conformance (reduce **W** and **T** $\rightarrow$ **W**). Either way, the optimal performance (in terms of conflict alerting) will occur when the true trajectory is deterministic (no uncertainty, 4-D path). The current ATM environment with its heavily structured airway system and ATC monitoring appears to allow for easy predictions of localized conflicts. There are restrictions and constraints in the system which make the trajectory more deterministic in many cases.

The notion of Free Flight seems to be contrary to this idea, however. It is probably unlikely that aircraft would be allowed to fly randomly about in the airspace. As explained in this chapter, the larger the uncertainties in the trajectories, the more difficult it is to determine and prevent possible conflicts. In Andrews and Hollister [1997], an analytical model was used to determine that a significant increase in conflict rate would result if aircraft maneuvering was left unconstrained. An increase in the number of conflicts coupled with a drop in alerting performance could only lead to problems. The end result would be a loss of efficiency and increase in workload rather than the more efficient system originally sought after.

One way around this is to have aircraft provide and confirm intent information *prior* to any changes in the current intended course. The information must be accurate, and some form of conformance monitoring would be helpful, else a modeling error would occur in the alerting system. Of course, the system should check for possible conflicts with the new path prior to allowing the changes to be made.

4.6 Summary

In this chapter, the problem of collision avoidance was recast as a problem of prediction in the presence of uncertainties. The importance of trajectory modeling was examined as a major source of errors in the outcome of conflict alerting. Without uncertainties, the problem would be greatly simplified. The issue of improving performance then becomes one of increasing prediction accuracy of the conflict situation and its resolution.

Chapter 5

A Probabilistic Perspective to the Alerting Design Process

In previous chapters, the use of iterative, ad hoc adjustments was discussed as a common method for setting threshold parameters. This approach to alerting design can be thought of as an implicit method of dealing with uncertainties. It is implicit because, as will be shown, the simulations are indirectly accounting for the uncertainties in the encounter situation.

In this chapter, a probabilistic perspective to the alerting design process will be examined and discussed in detail. It provides a different view to the current practice of locating suitable thresholds based on iterative searches using trial and error. Probabilistic elements will be shown to be embedded within the ad hoc approach, and thus the design is, in essence, influenced by probabilistic or stochastic concepts which may at first not appear to be present. A new, direct method will also be proposed to overcome some of the limitations in the ad hoc approach.

5.1 A Probabilistic Perspective to the Ad Hoc Approach

During the operation of an alerting system a discrete decision is made to either remain silent or issue an alert to warn the human operator. Typically, this decision is based on metrics and whether or not they exceed critical values defining the alerting thresholds. The manner in which these threshold parameters are set often follow an iterative, ad hoc approach as explained back in Section 3.3.2.

To reiterate, Figure 5-1 diagrams the general, iterative design process often seen in setting alerting threshold parameters. It usually begins with some working dynamic trajectory model (**W**) upon which the alert metrics are used to describe the encounter situation. The metrics can be thought of as forming the state-space of the alerting system. In some instances, the choice of metrics may be constrained by the type of information available, in which case the choice of working models may also be limited. For example, limited sensor information in the TCAS system allows the use of only range and range rate in the formulation of its horizontal alerting criterion, and a single path working model is used. Attempts to include information on the expected miss distance to improve prediction accuracy could not be achieved with the current equipment because of difficulties in estimating relative bearing angles between aircraft [Burgess et al., 1994].

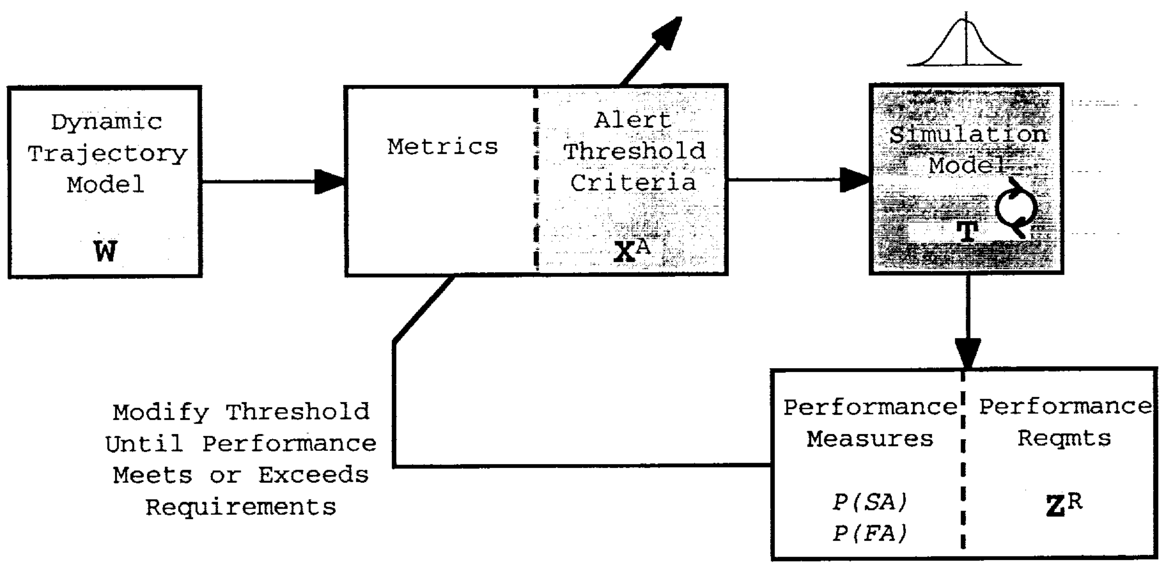


Figure 5-1: Current Design Process (Iterative Ad Hoc Approach)

In the common ad hoc approach, the thresholds would likely be initial settings (from some combination of analysis and user expertise), but usually require some fine tuning from test scenarios through simulations as shown in Figure 5-1. As a reminder,

$\mathbf{X}^A$ represents the alert space of the threshold criteria (Section 2.2.3). The number of test scenarios could run in the tens or hundreds of thousands , and would be used to evaluate the performance {e.g. $P(FA)$ and $P(SA)$} over the various simulation runs. For example, changes to the original TCAS design were tested using over 1 million hypothetical encounter geometries [Miller et al., 1994].

Adjustments and modifications to the thresholds or even the metric variables (the feedback path in Figure 5-1) would then proceed until a satisfactory setting is achieved. The values used to determine performance can themselves form a state-space which will be denoted as $\mathbf{Z}$. The symbology, $\mathbf{Z}^R$, in the figure is meant to represent the performance requirements that need to be satisfied by the alerting system. For example, $\mathbf{Z}^R$ could be the region of performance state-space where $P(FA) < 0.10$ and $P(SA) > 0.95$. If these requirements are not met, then the parameters in the threshold are iteratively adjusted until satisfactory performance is achieved.

There are some very important insights when the process is portrayed in the manner shown in Figure 5-1. The depiction looks quite similar to a neural network scheme where the simulations define a "truth" model from which the thresholds are adjusted to meet or optimize performance parameters.

From previous discussions, **W** is the working model being utilized by the alerting logic to predict conflicts; and **T** represents the "truth" model of the trajectory which determines the actual $P(C)$. **T** is still a random process and can be considered as an ensemble of individual trajectories. Thus, the simulations, being a collection of scenarios, can be interpreted as representing **T**. Taken together, the simulation scenarios are a probabilistic distribution (though discrete, it could be inferred to be a sample from a continuous distribution). The scenarios represent the variability or uncertainty in the true aircraft trajectories. If the distribution of the scenarios were changed, the threshold values would likely change also.

The performance measures, as it turns out, are commonly error and success rates (such as false alarms, successful alerts, or missed detections) as described earlier. Notice that even if the simulation scenarios are changed, the specification for the minimum level of performance will likely remain unaltered. If a design must achieve 99% success with less than 5% false alerts, then those requirements would not change with different sets of simulation runs. If they cannot be met, then changes in the metrics or threshold settings must be amended to obtain satisfactory results against the chosen scenarios. Thus the alerting thresholds can really only be considered to be indirect measures of the alerting performance. In essence, the design procedure is a mapping of the threshold metrics to the performance measures.

For example, if the thresholds were based on range (r), range rate ($\dot{r}$), and predicted miss distance (m), then the probability of a false alarm and a successful alert would be some function of these variables, $P(FA) = f(r, \dot{r}, m)$ and $P(SA) = g(r, \dot{r}, m)$, respectively. The functions, $f()$ and $g()$, would be specific to the scenarios used. In general, $P(FA)$ and $P(SA)$ can be expressed as a mapping from the threshold settings, $\mathbf{X}^A$, to the performance metrics as denoted in Equations 5.1 and 5.2.

$$P(FA) = f(\mathbf{X}^A) \tag{5.1}$$

$$P(SA) = g(\mathbf{X}^A) \tag{5.2}$$

where: $\mathbf{X}^A$ = threshold metric settings (alert space region)
$f()$ = false alarm mapping function (scenario specific)
$g()$ = successful alert mapping function (scenario specific)

The governing functions, $f()$ and $g()$, are typically not explicitly expressed or defined during the design process. Thus, it can become nearly impossible to predict the outcome or even make informed comparisons between different sets of simulation runs. This approach can lead to ambiguities and is really an indirect method of including uncertainties missed by **W** in the dynamic modeling stage.

5.2 A New Direct Approach

In Figure 5-2, a new approach to the alerting process is presented. As opposed to Figure 5-1, the intermediate block of metrics is removed in favor of directly estimating the probabilistic measures in which to make the alerting decision. The idea is to bypass the middle step since the results of the scenario simulations are being utilized to adjust the parameters in the threshold metrics in the first place. In Figure 5-1, the notion is that the probabilities { $P(FA)$, $P(SA)$ } are functions of the set threshold metrics; thus in terms of the cause-effect relationship, the set thresholds determine the probabilistic performance. The concept is somewhat reversed in Figure 5-2 where the probability values determine when and where to alert. Because of the feedback in Figure 5-1 to meet pre-determined probabilistic requirements, it is really the probabilistic parameters that drive the threshold placement. As mentioned before, in effect, the threshold placement is really just a function of the probabilistic performance measures and the probabilistic distribution of simulation scenarios.

In the concept of Figure 5-2, the working trajectory model is made to match as closely as possible to the "truth" model (**W** = **T**). In doing so, the alerting algorithm is obtaining a direct prediction of the likelihood of conflicts and the ability to avoid them. These values can then be utilized as the threshold metrics in the state-space of $\mathbf{Z}$ with the alerting criteria denoted by $\mathbf{Z}^R$ (performance requirements).

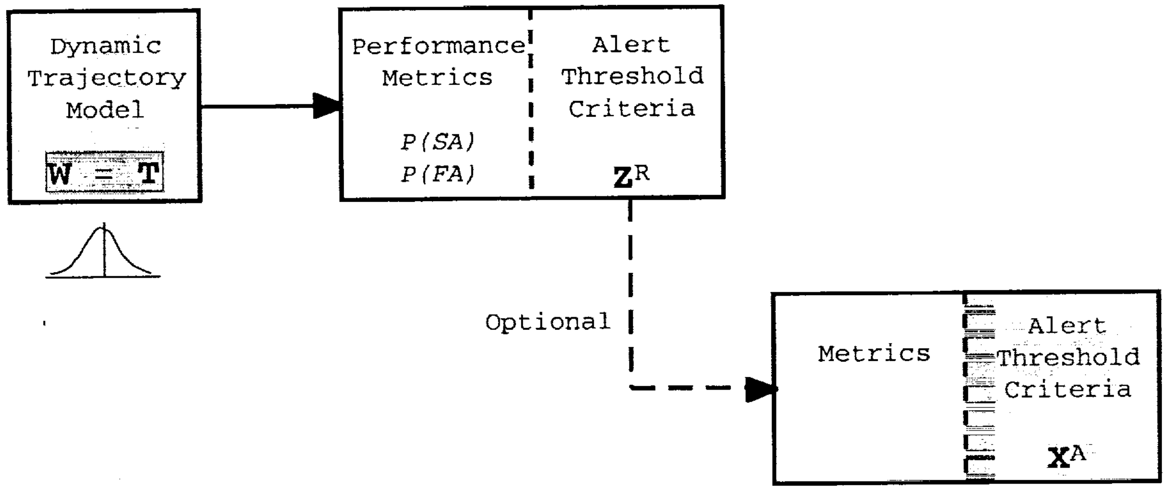


Figure 5-2: New Direct Approach

As indicated by the dashed line in Figure 5-2, it may be possible to map the probabilistic values to other metric variables in another state-space, $\mathbf{X}$, with alert space $\mathbf{X}^A$. In doing so, this may allow for easier interpretation of the threshold logic since probabilistic values may not always be clearly understood by the human operator. However, this may not always be possible unless a one-to-one mapping of variables exists. The problem is akin to the same type of dilemma associated with inverse kinematics.

5.3 Implications from a Probabilistic Perspective

The approach shown in Figure 5-2 requires a direct modeling of the uncertainties in the trajectories of the aircraft, which in turn can help determine the impact and influence of each source of uncertainty on the alerting performance. This direct link gives rise to some very important implications that can have a significant impact in the design and analysis of alerting systems. To fully understand the consequences requires a detailed explanation of the differences as well as the association between Figures 5-1 and

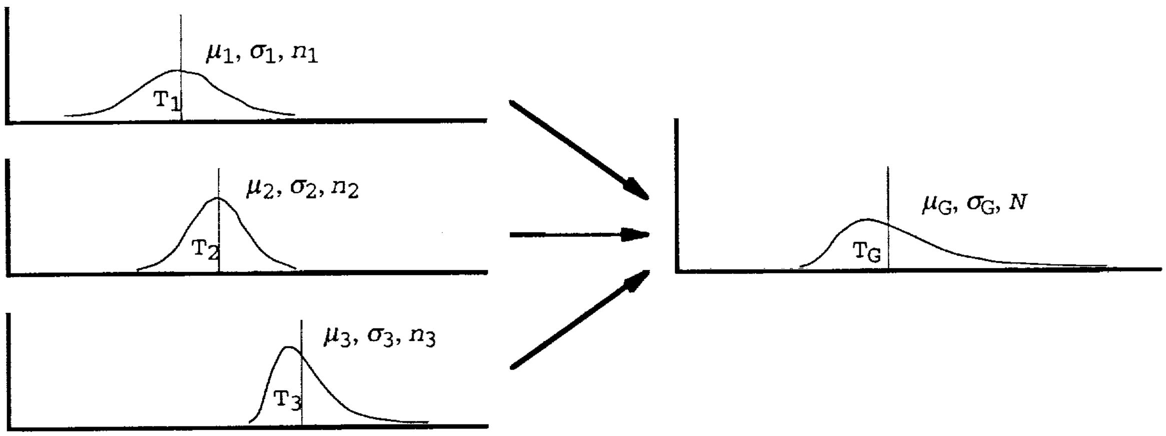
5-2. For the remainder of this work, the former method will be referred to as the ad hoc approach, and the latter as the direct approach.

There are some very important ramifications to notice here. First, the ad hoc method tends to develop a global threshold setting as opposed to a situation-specific threshold, one that is individually tailored to the current encounter situation. As will be shown, a global threshold tends to exhibit a higher level of uncertainty and reduced overall level of performance. Second, the approach is also heavily influenced by the distribution of test scenarios used for the simulations. Using an alternative sample of test cases could change the performance outcome resulting in a different set of threshold values. In effect, the thresholds could be highly biased toward certain conflict conditions while ignoring or discounting other possible encounters. Also, a complete rerun of the iterative process would be needed again to formulate a new set of threshold parameters. Finally, the ad hoc approach can be considered a functional mapping of the performance state-space, **Z**, to a different domain of state-space, **X**. Though this can be advantageous under certain conditions or applications, it can also be a disadvantage when the complexity of the mapping is considered.

5.3.1 Global Design vs. Situation-Specific Design

A global design refers to a process in which the simulations used to set the thresholds are based on an aggregate mixture of different encounter scenarios; while a situation-specific design only considers the current situation at hand. To illustrate this concept, consider the following example. Take an automobile company designing a "world" car to be sold globally under one baseline model. There are some obvious advantages to such a tactic, of course, since it may be minimizing labor, parts, design, and advertising costs. Suppose this company gathered the following data shown in

Figure 5-3a on the height of drivers in country A, B, and C. One of the design (or performance) requirements is to be able to seat 95% of the drivers comfortably.



a) Individual Distributions **b) Global Distribution**

Figure 5-3: Example of Global Design Distribution

Assume the data show that 95% of drivers sampled were between 5'0" and 5'6" in country A, 5'2" and 5'8" in country B, and 5'4 and 5'11" in country C. These distributions will be associated with the random variables T_1, T_2, and T_3, respectively; with corresponding means, variances, and sample sizes of μ_1, σ_1^2, n_1; μ_2, σ_2^2, n_2; and μ_3, σ_3^2, n_3. If the company were to design different cars for each of these markets, the size and dimensions of each car would more than likely be tailored to meet the requirements of each country separately. However, if restricted to a one car design in which a combined global distribution, T_G (see Figure 5-3b), is utilized as the test data, then some compromises and added difficulties would be encountered. The combined distribution would have the following mean, μ_G, and variance, σ_G^2 (see Appendix B for derivations):

$$\mu_G = \frac{n_1}{N}\mu_1 + \frac{n_2}{N}\mu_2 + \frac{n_3}{N}\mu_3 \tag{5.1}$$

$$\sigma_G^2 = \left(\frac{n_1}{N}\sigma_1^2 + \frac{n_2}{N}\sigma_2^2 + \frac{n_3}{N}\sigma_3^2\right) + \left(\frac{n_1}{N}\mu_1^2 + \frac{n_2}{N}\mu_2^2 + \frac{n_3}{N}\mu_3^2\right) - \overline{\mu}_G^2 \tag{5.2}$$

where:

$$N = n_1 + n_2 + n_3$$

It very important to note here the following characteristics:

$$\min(\mu_1, \mu_2, \mu_3) \le \mu_G \le \max(\mu_1, \mu_2, \mu_3) \tag{5.3}$$

$$\sigma_G^2 \ge \min(\sigma_1^2, \sigma_2^2, \sigma_3^2) \tag{5.4}$$

There are three significant consequences that come out of these equations and also graphically from Figure 5-3. First (from Equations 5.1 and 5.3), <u>the mean of the global distribution will not be the same as the mean of the individual distributions</u> (modeling error); unless, of course, $\mu_1 = \mu_2 = \mu_3$. In fact, it is probably unlikely that μ_G would be equal to even one of the individual means. Second (from Equations 5.2 and 5.4), <u>the variance, or spread, of the global distribution is larger than at least one of the other individual distributions</u> (increase in overall uncertainty). The increase is due to the additional dispersion caused by the conglomeration of the different distributions located at different positions (the effect of the individual means on the global variance, σ_G^2, can be seen in Equation 5.2). These additional terms will be called the *across-sample* variance as opposed to the *in-sample* variance, σ^2, of each separate distribution. And finally (from Equations 5.1 and 5.2), <u>the global distribution can be heavily biased toward individual distributions by having uneven sample sizes</u>. One consequence is that performance in certain situations may be compromised more than others. The global performance may appear to be adequate, but in individual situations, the outcome could be completely unsatisfactory even though it was included in the simulations.

Sometimes, there is not a clear line defining a global distribution. For instance, the different countries could be broken down further into male and female drivers, each with its own separate distribution. These could even be taken down more by, say, age group or household income. The appropriate amount to divide out depends on the problem itself. There may not be any justification to utilize the knowledge. For example, it may not make any sense to design a car specifically for women only. How specific the distributions need to be will likely depend on costs or what is the appropriate level of uncertainty that can be afforded. Also, the information or data may not exist to allow for more specific categorizations.

The graph in Figure 5-4 diagrams the level of uncertainty in a stochastic process. A deterministic system would fall on the far left of the bar while a completely chaotic and unpredictable system would be to the far right. A stochastic system would have some amount of inherent uncertainty built in (i.e. no matter how much information is available, the realization of a single event is still not completely predictable). This is indicated by the location of the vertical line in the figure and represents the uncertainty inherent in $\mathbf{T}$. Errors due to modeling will increase the overall uncertainty of the system by inducing additional components to the right of this line (e.g. Equation 5.2). Predictability and performance of the system is thus degraded if there are sufficient errors in the models used in the design process.

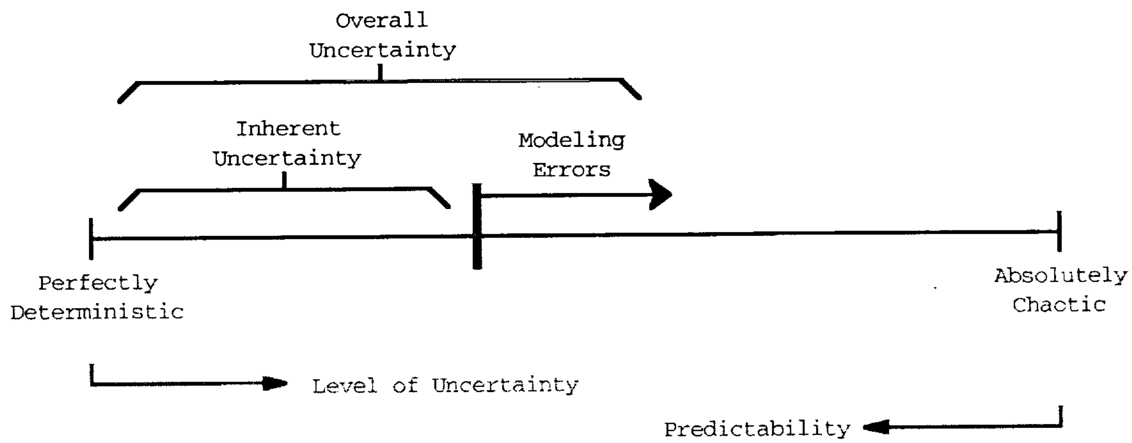


Figure 5-4: Level of Uncertainty

The examples above allow a better understanding of the importance of utilizing appropriate simulation models. Returning back to conflict avoidance and the ad hoc approach of Figure 5-1, the consequences can be re-worded in the following manner with reference to the terms defined in Chapter 4:

i) Use of a global distribution would likely result in <u>modeling errors</u> and therefore <u>increase the overall uncertainty</u> of the system since the design is not individually tailored to the current encounter situation. The threshold would instead be based on a weighted average of thousands of sample scenarios which may not even be applicable to the current situation at hand ($\mathbf{W} \neq \mathbf{T}$).

ii) Virtually any threshold setting can be "shown" to be optimal by merely weighting certain encounters to take place more often. This could occur inadvertently, of course, but may easily lead to inappropriate results and conclusions.

To further clarify these concepts, a simplified conflict simulation example will be given. Figure 5-5 shows a possible subset of sample simulation scenarios that might be used as test cases for the ad hoc approach. The aircraft in these six test cases each have stochastic trajectories with some random distribution so that repeated runs would result in different paths being taken. For the purpose of this discussion, these scenarios will be labeled $\mathbf{T}_1, \mathbf{T}_2, \ldots, \mathbf{T}_6$ with the indices to designate the different encounter conditions. Notice that the simulations represent the "truth" model of trajectories for which the alerting logic is to be tested against to determine the system's performance. For clarification, the six cases shown in Figure 5-5 are just a minute subset of thousands of simulation scenes to be used in the ad hoc process. For the sake of simplicity, assume there are a total of q different encounter situations in the simulations so that $\mathbf{T}_1, \mathbf{T}_2, \ldots, \mathbf{T}_q$ span the entire distribution of scenarios. In the testing of TCAS, there were literally hundreds of thousands of sample simulations used to help determine the appropriate threshold parameters [Drumm, 1996; Williamson and Spencer, 1989; Miller et al., 1994].

The performance of the threshold logic could be determined by use of the definitions of $P(FA)$ and $P(SA)$ given back in Sections 3.2, or more precisely in the following form,

$$P_G(FA) = \frac{\text{number of alerts that were unnecessary}}{\text{total number of alerts}} \tag{5.5}$$

$$P_G(SA) = \frac{\text{number of times an alert is successful in avoiding a conflict}}{\text{total number of alerts}} \tag{5.6}$$

In the ad hoc approach, these values are actually global or overall performance metrics which will be denoted with the subscript G. The reason for the differentiation is because the metrics are computed over the entire spectrum of simulation scenarios using the same threshold setting. They are a compilation, or weighted average, of the system threshold's performance over all situations, $\mathbf{T}_1, \mathbf{T}_2, \ldots, \mathbf{T}_q$.

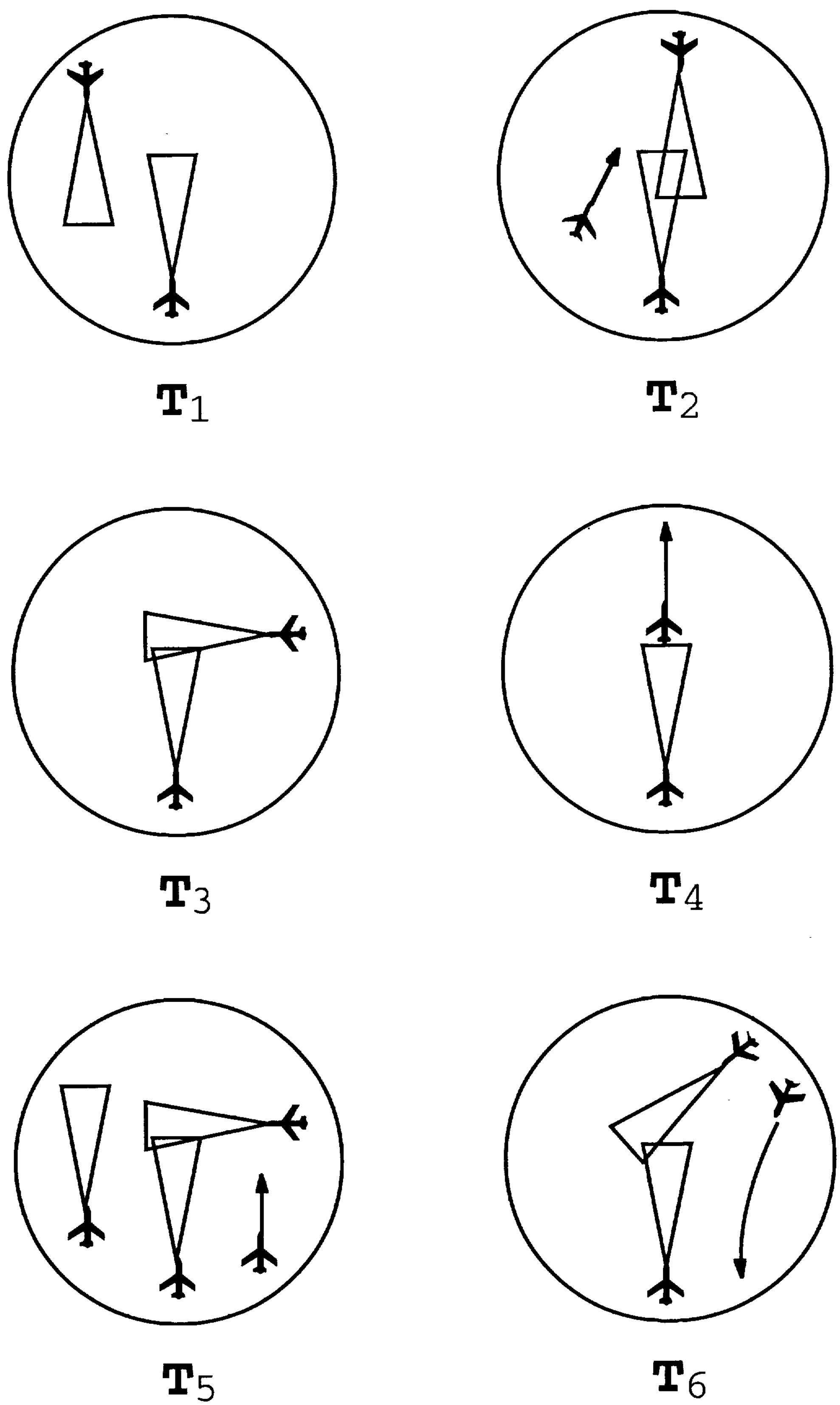


Figure 5-5: Sample Simulation Scenarios

A threshold designed in this manner would suffice if the particular threshold setting works well in each of the individual $\mathbf{T}_i$ cases (where the subscript, i, denotes some particular scenario from 1 to q). However, such a setting is really a compromise between the various test cases and is not optimized to deal with each of the individual situations separately. In fact, the particular threshold could be detrimental in certain cases, and yet, in the global metric, still appear to perform adequately. It has already been shown that this type of design approach, which is based on using a distribution of different scenes in the simulations, will lead to an overall drop in system performance because of the increase in modeling error and also an increase in the overall uncertainty of the process.

To elaborate a little more on the above statement, assume an example where the current situation is $\mathbf{T}_3$ of Figure 5-5 (two aircraft crossing at near right angles). Keeping in mind the situation is still stochastic, the most accurate way of determining an appropriate alert threshold would be to utilize a dynamic model $\mathbf{W} = \mathbf{T}_3$ to predict the likelihood of a conflict. This is the method of the direct approach. The threshold should not be determined with any influence, whatsoever, from any of the other test cases, $\mathbf{T}_1$, $\mathbf{T}_2, \ldots, \mathbf{T}_q$ (except for $\mathbf{T}_3$). They have no bearing on the current encounter and their influence could actually be detrimental to the decision to alert for the current situation $\mathbf{T}_3$.

Take the case of $\mathbf{T}_5$. The two middle aircraft in that scenario appear to be in a similar conflict encounter as $\mathbf{T}_3$; however, the additional surrounding aircraft would more than likely require a different set of alerting criteria to account for the loss of lateral maneuvering available to resolve the original conflict { $P(SA)$ will be affected}. A threshold optimally set to handle $\mathbf{T}_3$ may, on the other hand, be ineffective, or even hazardous, in the case of $\mathbf{T}_5$.

The ad hoc approach which uses an aggregate of the individual scenarios would simply be obtaining a weighted, global threshold. The alerting performance on average would be less than optimal because of modeling errors and increased uncertainties induced into the design. Utilizing only one threshold criterion to handle both $\mathbf{T}_3$ and $\mathbf{T}_5$ would result in a system that would not be best suited for either case individually, but instead would be a compromise between the two.

Take the case of the Ground Proximity Warning System. If the thresholds were to be designed based on an equal distribution of flat, medium, and high sloping terrain cases, then one might expect to find compromises in performance in each of the individual circumstances. The threshold would be averaged out to be optimized globally, but the overall uncertainties (from lack of information about the terrain features) would be large. The result is high rates of false alarms over relatively flat terrain, but inadequate warning time in mountainous terrain. Changing the distribution of the simulations would only result in biasing the thresholds toward certain types of terrain conditions and would not be a good solution to deal with the problem since jet transports are expected to be flown almost anywhere in the world.

It is, however, possible to break the thresholds down into multiple scenario-specific groupings using "if-then" statements or include additional metrics, provided, of course, the information is known. For example, if the aircraft is currently over flat terrain or the ocean, then one set of thresholds could be utilized; if it is in a region of high mountains, then another set of thresholds would be invoked.

In the case of the traffic example of Figure 5-5, the breakdown can then be expressed as Equation 5.7.

$$
P(FA) = \begin{cases} f_1(\mathbf{X}_1^A), & \text{if } \mathbf{T}_1 \\ f_2(\mathbf{X}_2^A), & \text{if } \mathbf{T}_2 \\ f_3(\mathbf{X}_3^A), & \text{if } \mathbf{T}_3 \\ \vdots & \end{cases}
\qquad
P(SA) = \begin{cases} g_1(\mathbf{X}_1^A), & \text{if } \mathbf{T}_1 \\ g_2(\mathbf{X}_2^A), & \text{if } \mathbf{T}_2 \\ g_3(\mathbf{X}_3^A), & \text{if } \mathbf{T}_3 \\ \vdots & \end{cases}
\tag{5.7}
$$

where:

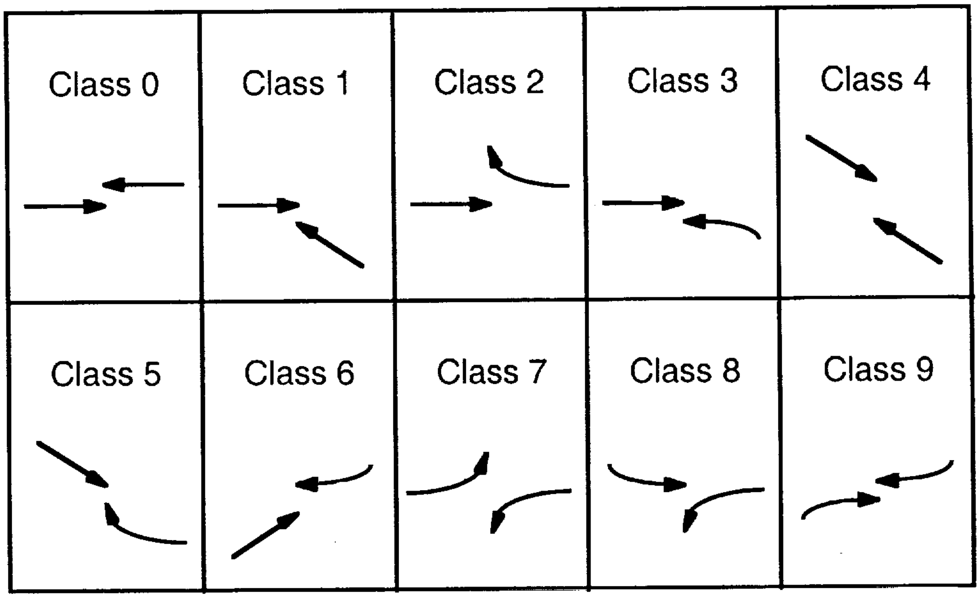
$\mathbf{X}^A$ = threshold metric settings (alert space)

$f()$ = false alarm mapping function (scenario specific)

$g()$ = successful alert mapping function (scenario specific)

This design process can be very time consuming and complex if many different situations are to be addressed separately and the iterative procedure of Figure 5-1 must be repeated for each one. This basically leads to the ad hoc approach of alerting design. As explained earlier, TCAS, for instance, utilizes different Tau Criteron values and DMOD buffers for different altitudes and encounter situations (see Figure 3-3). There are a fair amount of changes in threshold parameters just to account for climbing/descending aircraft and altitude, even for a seemingly simple design in which aircraft are assumed to fly in straight, constant velocity paths and only use vertical evasive maneuvers.

In the evaluation of TCAS, the MITRE Corporation generated a large database of pairwise aircraft encounters from actual recorded tracks in the United States airspace [McLaughlin and Zeitlin, 1992]. Using this database, MITRE defined 10 types of vertical encounter geometries (Figure 5-6) which were considered to encompass all aircraft maneuvers observed. In evaluating the performance of the system, a large number of simulations were used to cover each of these 10 encounter classes [Drumm, 1996]. Changes to the threshold parameters were then suggested due to the results of these simulations.



Arrows represent aircraft vertical profiles

Figure 5-6: TCAS Encounter Types Defined by MITRE
[Drumm, 1996]

In another example, Zeghal [1994] divides horizontal planar conflicts into 3 separate classes of encounters when using a force potential method. Three different sets of equations are defined to best express the threat of collision for: 1) head-on, 2) overtaking, and 3) tangent encounters. The need to derive a separate threshold metric for different encounter situations illustrates the desire to veer toward a more situation-specific design in order to maintain performance.

This method is one way of dealing with this problem, but it be can be a tedious process of breaking up and grouping the scenarios to cover all possible encounter geometries and flight conditions. In a more general conflict alerting environment (such as Free Flight) when waypoints, intent information, multiple aircraft, and 3-D encounters are all fused together, it becomes extremely difficult to utilize such a scheme to amalgamate all the individual situations and develop separate thresholds for each one. In the ad hoc approach shown in Figure 5-1, it would seem necessary to perform the task iteratively for each scenario $\mathbf{T}_1, \mathbf{T}_2, \ldots, \mathbf{T}_q$ in order to map out a different alert space, $\mathbf{X}_i^A$, for individual encounters. This would be true unless, of course, one could pick a set of threshold variables which would allow settings to be virtually invariant of the encounter situation. In fact, this is the approach shown in Figure 5-2 and the topic of discussion in the following sections.

5.3.2 Relating Performance Measures to Alerting Thresholds

In using the ad hoc approach, the choice of state-space variables for the threshold can be quite variable as was shown in the survey of alerting methods (Appendix A). If so, then what constitutes a viable or sufficient set of variables? Is the use of only the range variable (r) adequate? Or is the time to closest point of approach (t_{CPA}) or expected miss distance (m) also needed? The answer actually depends on two factors: 1) the type of situations to be encountered, and 2) the performance requirements.

It was mentioned earlier in this chapter that the ad hoc approach of Figure 5-1 can be thought of as a mapping of performance measures back to another set of threshold variables due to the iterative feedback adjustments of threshold parameters. Referring back to Equations 5.1 and 5.2, the mapping equations, $f()$ and $g()$, are governed by the scenarios used in the simulations; and the performance measures, $P(FA)$ and $P(SA)$, are used to judge the efficacy of the threshold setting, $\mathbf{X}^A$. Thus, the encounter scenarios and the performance requirements are the only defining factors which can determine whether the choice of threshold variables will be adequate.

Figure 5-7 shows a conceptual illustration of mapping thresholds in the state-space of $\mathbf{X}$ to the state-space of performance measures, $\mathbf{Z}$. An example of performance state-space, $\mathbf{Z}$, might be the variables of the SOC diagram, $P(FA)$ and $P(SA)$. The alert space in $\mathbf{X}$ is denoted by the region, $\mathbf{X}^A$, and the required performance region to be met in $\mathbf{Z}$ will designated $\mathbf{Z}^R$. When $\mathbf{X}^A$ is mapped into $\mathbf{Z}$, it actually becomes a single state vector $\mathbf{z}^A$. If $\mathbf{z}^A$ is outside the region of $\mathbf{Z}^R$, as depicted in the leftmost illustration of Figure 5-7, then the performance requirement is not met and the threshold parameters need to be adjusted until a suitable performance, $\mathbf{z}^A$, is obtained. This is shown in the series of drawings going from left to right, and represents the iterative search and fine tuning of the feedback loop back in Figure 5-1. Notice that $\mathbf{X}^A$ is changed in each step.

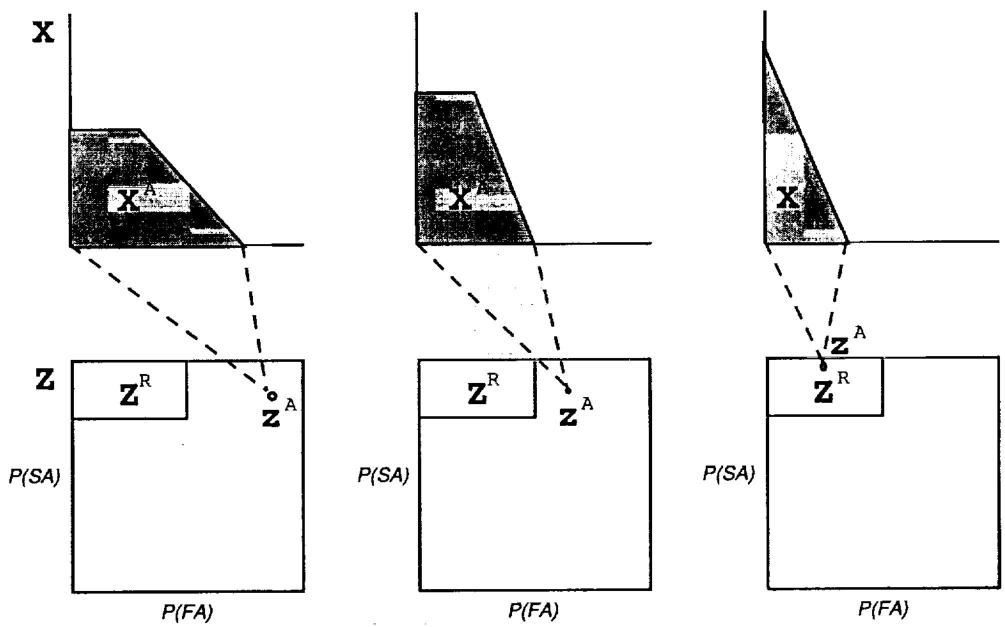
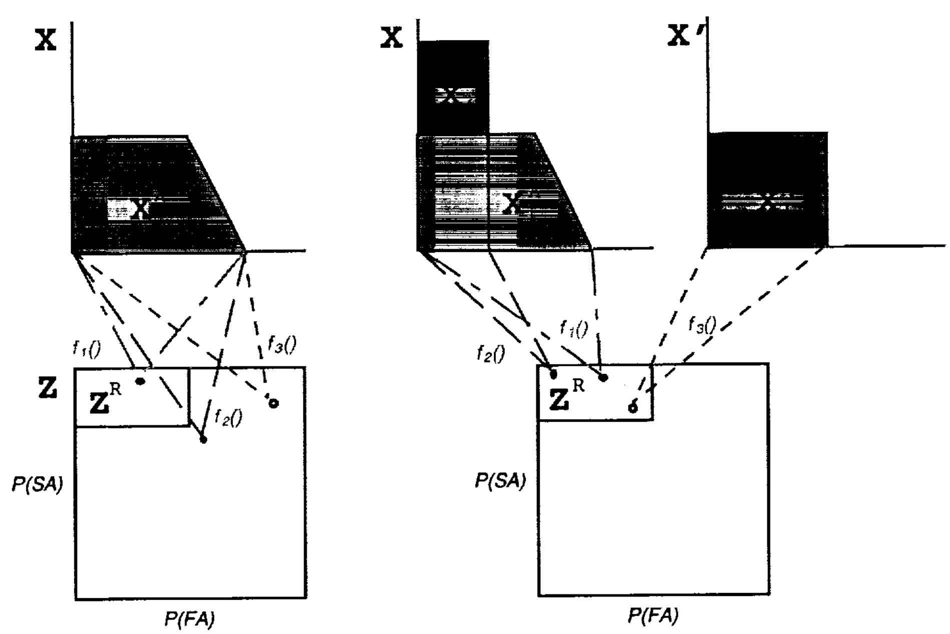


Figure 5-7: Mapping to Performance State-Space

If an acceptable $\mathbf{X}^A$ cannot be found, then there are four possible options. The first is to change to a different set of threshold variables (i.e. change the state-space, $\mathbf{X}$). The threshold metrics may not have been appropriate for the encounters, or else there may have been an insufficient number of variables to handle the complexity of situations.

The second option, which was explained in the last section, is to partition out the thresholds to handle more situation-specific groupings. Basically, different sets of threshold criteria are used for different encounter scenarios.

Take the example shown in Figure 5-8a where an alerting threshold, $\mathbf{X}^A$, is used for three specific types of encounters with the corresponding mapping functions $f_1()$, $f_2()$, and $f_3()$. In this case, only $f_1()$ maps adequately to the required performance specifications. If $\mathbf{X}^A$ were to be utilized for all three encounter situations, the overall performance of the system would be a weighted average of each of the individual outcomes. In Figure 5-8b, a second state-space, $\mathbf{X}'$, with different metric variables is used to derive adequate thresholds for $f_3()$; while the original state-space but different parameter settings sufficed for $f_2()$. The result is again an increased number of threshold metrics designed and tailored specifically for different types of encounters.



a) Global Threshold **b) Situation-Specific Thresholds**

Figure 5-8: Use of Situation-Specific Thresholds (State-Space Explanation)

The third option is to simply limit the use of the alerting logic to specific types of encounters, basically what is done with TCAS. It can only be used for near term, last minute conflicts due to the lack of accurate bearing information in the logic. In the case of GPWS, it might be conceivable to have two separate threshold designs, one for mountainous terrain and one for flat terrain. The switch could be made manually by the pilot or better yet, automatically with some onboard database coupled to navigational data.

The method of limiting the alerting logic to certain types of encounters is somewhat analogous to $\mathbf{T} \rightarrow \mathbf{W}$ as discussed back in Chapter 4. The idea is to maintain good results; albeit in restricted circumstances. Sometimes it is out of necessity to cope with the limitations of the design, such as with the lack of available information to the system. Other times, the functional requirements may not warrant or call for the additional capabilities (e.g. initial requirements for TCAS were for short term conflicts only).

The fourth and final option is to utilize a different resolution strategy. Given that the performance is partly based on the ability to avoid a conflict, it is natural to assume some metric such as $P(SA)$, which is based on a specific avoidance maneuver, is included in the performance state-space. Since this was already discussed in Chapter 4, not much more on this topic will be mentioned here other than to say a different or a more drastic avoidance maneuver might be examined.

5.3.3 Using Performance Measures as Alerting Thresholds

In the previous section, the relationship between the performance metrics and alerting thresholds was examined. Now, one might ask why go through all the trouble testing and re-testing, adjusting and re-adjusting all the threshold parameters, when the performance metrics, themselves, could be used as the alerting thresholds? It was already explained in Section 5.1 that the performance values were really driving the threshold settings in the ad hoc approach. If this is the case, then it appears that if the performance measures could be obtained directly in real-time, there would be no need to implement the additional iterative steps to map to what would essentially be a set of redundant metrics.

The mapping procedure shown in Figure 5-1 leaves open many different possible variables for use as metrics without real analytical computations of conflict in the presence of uncertainty. In essence, it is bypassing the dynamic modeling stage of Figure 3-2, either completely or partially while leaving the fine tuning to pattern matching. The reason for the required mapping is because of the disparity between the working trajectory model, **W**, and the "truth" model, **T**. Without the ability to obtain an accurate prediction of conflict directly from its own trajectory model (since $\mathbf{W} \neq \mathbf{T}$), the alerting logic is forced to trial and error methods.

The result is akin to obtaining a simplified model of a probabilistic model, such as through correlation or regression modeling to find a simplified set of metric parameters to best fit probabilistic data. However, there is really no need for this since <u>a prediction of alerting performance can be obtained directly by using probabilistic trajectory modeling assuming $\mathbf{W} = \mathbf{T}$</u> (the direct method). In the direct approach, there would be no modeling error provided **W** is a good depiction of **T**. The working trajectory model, **W**, is either a representation of the simulation scenarios in Figure 5-1 or a subset of them. In order to do so, <u>**W** must be allowed to exhibit any trait that would have been characterized in the simulations, including the likelihood of human errors and blunders</u>.

5.3.4 Continuous Update of Trajectory Model W

In the direct method as shown in Figure 5-2, there is a need to continuously update the dynamic model, **W**, utilized by the alerting system to keep up with the current situation. As long as the uncertainties in the trajectories can be modeled, the update process is a natural progression as new aircraft states and other data such as intent information is brought in to modify **W**. At any instant in time, the current aircraft states are projected into the future using **W** and the probabilistic values, $P(FA)$ and $P(SA)$, are computed. The decision to alert is then made directly from these performance estimates.

The direct approach which utilizes **W** = **T** is as situation-specific as one can get since the alert decision is based solely on any current information specific to the encounter. <u>All knowledge of the current situation, including the effects of uncertainties, is contained in **T**</u>. Take the example back in Figure 2-9 where only range and range rate were used to define an alerting threshold. The two variables are simply not sufficient to completely define a specific encounter situation. There is no information differentiating encounters at different bearings or predicted miss distances. Nor is there information regarding the effects of uncertainties or what type of intent information was involved. It is, however, conceivable to develop an infinite number of thresholds for every possible type of encounter scenario and store them in the alerting logic (much like the if-then statements of Equation 5.7). But this is not very practical if the alerting system were to be designed to handle multiple aircraft in 3-D flight and various types of flight conditions.

The idea behind the direct approach and **W** = **T** is to allow the computation of the threat condition on the run as the situation occurs. It is analogous to many current computer chess programs which wait for a move to be made; then based on the current configuration, propagate the probable moves of each chess piece (out to a finite number of moves ahead) and make a decision based on the results. This was the approach used by IBM's Deep Blue supercomputer in its highly touted and successful match against chess Grand Master Kasparov [Krauthammer, 1996]. Even in the limited confines of the chess board and the incredible processing power of today's supercomputers, it is nearly impossible to determine what all the moves should be prior to the start of the game (except for the first few moves the opening). There are just too many possible configurations even on the discrete space of a chess board. Instead, the simulations are performed by the computers on the run as the situation unfolds and the decisions are situation-specific based on the current configuration.

In Chapter 8, this similar tactic is used to develop a real-time conflict alerting probe. By keeping $\mathbf{W} = \mathbf{T}$, the alerting decision is tailored specifically to the current conditions of the encounter. Any changes to aircraft state or intent information are accounted for directly and done as the situation occurs. This resolves the problem of pre-determining separate threshold metrics for every possible encounter situation.

5.4 Summary

In this chapter, the common ad hoc approach to alerting system design was re-examined from a different perspective. It was shown that probabilistic concepts of performance and uncertainties were embedded within the design process. As discussed with reference to the iterative method of Figure 5-1, most threshold metrics can be thought of as a set of simplified variables mapped to satisfy probabilistic performance criteria. The notion that probabilistic analysis and uncertainty drive the alerting system design is clearly seen in the feedback loop of Figure 5-1.

A new direct approach to alerting design was presented and shown be a more compact method of estimating performance directly without the unnecessary step of mapping back to a redundant set of threshold metrics. Since all information with regard to the current situation is contained in the characteristics of the probabilistic aircraft trajectories of **T**, properly modeling these trajectories in the conflict logic (**W** = **T**) allows for the most accurate prediction of the current encounter in a stochastic environment. This ensures the alerting decision is based on situation-specific information rather than a global set of data which was shown to have a degrading effect on performance.

Chapter 6

Probabilistic Analysis of Conflict

As mentioned in Section 3.3.3.3, the use of probability estimation has been explored in conflict analysis before [Kuchar, 1996; Paielli and Erzberger, 1997; Heuvelink, 1988; Rome and Kalafus, 1988; Taylor, 1990; Bakker and Blom, 1993; Williams, 1993; Warren, 1997, Prandini et al., 1999; Innocenti et al., 1999]. In previous work, Paielli and Erzberger [1997], developed a viable analytical solution to determine the probability of a conflict for two aircraft maintaining a straight ahead course. Their approach used Gaussian uncertainties to model along- and cross-track error and can be rapidly solved and implemented in real-time. If more complex uncertainties (e.g. non-Gaussian, 3-D trajectories, aircraft changing course, pilot reaction times) are modeled, it becomes increasingly difficult to obtain an explicit analytical solution.

In this thesis, a Monte Carlo based methodology is employed. The approach can intake a large and complex assortment of probabilistic distributions without added difficulty. The complications of estimating the future trajectory were explained in the previous chapters and it was shown how modeling errors could adversely affect the conflict prediction and alerting process. Because the approach is based on Monte Carlo simulations, there is a great deal of flexibility built into handling difficult trajectory models. However, since the simulations are iterative, significantly more processing power is required for the computations than, say, the method employed by Paielli and Erzberger [1997]. Also, concerns arise on the stochastic nature of the process to achieve repeatability of the results. Nevertheless, a systematic approach can be devised to obtain fairly fast results with sufficient bounds on the accuracy of the values.

6.1 The Trajectory Model

To calculate the probability of a conflict, $P(C)$, the positions of the involved aircraft must be projected into the future using some form of trajectory model as was discussed in Chapter 2. Essentially, the subscript **W** has been dropped off $P(C)$ with the understanding that only a working model estimate of the actual truth trajectory can be used in simulation. The implications of this were discussed back in Chapter 4, and it has further consequences when intent information is included in the model as will be explained later in another chapter.

Figure 6-1 is a pictorial representation of an aircraft in flight showing some of the possible parameters which may affect the uncertainty in the future trajectory. The modeled parameters might include uncertainty in the current position estimate, future along- and cross-track position variability, and the potential for and magnitude of course changes.

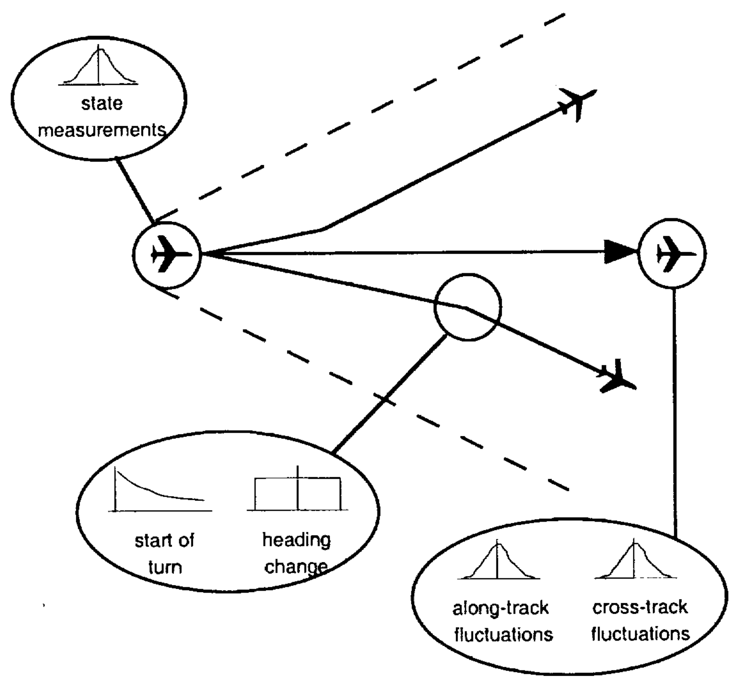


Figure 6-1: Potential Sources of Uncertainty in Trajectory

The approach of this thesis assumes that the uncertainty of the future path can be approximated by an ensemble of possible trajectories weighted by the likelihood of their occurrence. The work involves modeling each parameter that could influence the flight of the aircraft as a probabilistic distribution, and then using random sampling to generate variations in flight path during successive Monte Carlo iterative runs. Figure 6-2 shows the baseline model that was used. In this thesis, the aircraft with the alerting system will be termed the *host* aircraft while other vehicles involved in the encounter will be denoted as *intruder* aircraft.

Uncertainty in the current position is modeled after the accuracy of combined Global Positioning System (GPS) and Inertial Navigation System (INS) estimates, and is shown as a normally distributed random variable with standard deviation of 50 meters laterally and 30 meters vertically. For level flight, course drift in the future trajectory is modeled as a 15 knot standard deviation speed fluctuation (along-track error) and a 1 nautical mile standard deviation cross-track error. These tracking error values were based on data obtained empirically from observations of current traffic by Paielli and Erzberger [1997].

	Uncertainty Parameter	Modeled Distribution
Host Aircraft & Intruder Aircraft	Lateral Position Error	Gaussian σ = 50 meters
	Vertical Position Error	Gaussian σ = 30 meters
	Speed Fluctuation (Along-Track Variability)	Gaussian σ = 15 knots
	Cross-Track Variability	Gaussian σ = 1 nautical mile

Figure 6-2: Baseline Trajectory Model

Additional model parameters can easily be included into the Monte Carlo simulations without much added difficulty or loss in computational speed. The more common ones utilized within the scope of this thesis work are displayed in Figure 6-3. They include provisions for the likelihood of random course changes in heading and altitude; plus pilot response latency during avoidance maneuvers used in conflict resolution analysis. The specific distributions chosen serve only as one possible model and undoubtedly other distributions can be used. The modifications are relatively simple with the Monte Carlo approach, and usually only involve sampling from a different distribution and possibly making some appropriate changes in the program algorithm to reflect the nature of the adjustments in trajectory path. For example, in descending flight, fluctuation in speed has been observed to increase slightly to 20 knots standard deviation and the vertical rate can vary with a standard deviation of 300 ft/min for a 1500 ft/min descent, both normally distributed [Erzberger et al., 1998]. These additional parameters can be added easily to a Monte Carlo sampling algorithm without much effort and with little loss in processing speed.

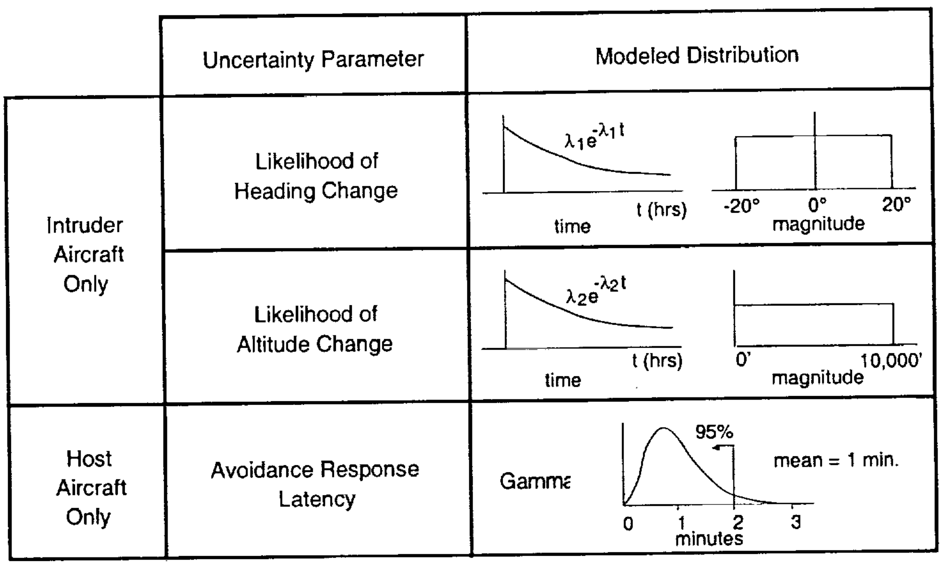


Figure 6-3: Additional Model Parameters

The task of modeling human behavior is extremely difficult to begin with, and any attempt to quantify the likelihood of the pilot in altering the aircraft's present course should be handled with caution. As Figure 6-3 shows, heading and altitude changes were modeled as Poisson processes with an average rate of occurrence defined by the parameters λ_1 and λ_2, respectively. The distributions are formally known as exponential distributions in probability theory [Drake, 1967] and show the likelihood of the first-order interarrival time (i.e. first occurrence of course change). The units of these parameters are arbitrary, but were taken as λ_1 turns per hour and λ_2 altitude changes per hour. Another possibility could have been a distance based unit such as altitude changes per mile. The magnitude of a random course maneuver was modeled to be uniformly distributed to a specified limit such as ± 20 degree in heading or within 10,000 ft in altitude.

The purpose of including random course changes into the trajectory is simply that they do occur when considering trajectories on a statistical basis and are most likely the largest source of uncertainty in the prediction process. When not included, the outcome becomes very similar to the single path model shown back in Figure 2-3, and conflict determination and resolution can often become overly simplistic. Simply choosing to ignore the possibility of pilot actions because of the complexity undermines the true difficulty involved in conflict prediction and analysis. The mere fact that researchers examine worst case methods indicates the concern over this problem. Also the situation is exasperated in light of the current push for less restrictions and more flexibility for rerouting in the newly termed Free Flight environment [RTCA, 1995; Phillips, 1996].

The modeling of the course adjustments into the trajectory serves to better understand the impact of their occurrence on the entire conflict prediction process. Again, the distributions used in the model are only estimates and cannot be expected to perfectly match the exact outcome. The purpose is to capture the essence of the uncertainties which may lead to possible conflict encounters in the desired time frame. Even if the values of the parameters are unknown, the impact of changes in the parameters can be evaluated to determine their relative importance in the conflict assessment or help determine trends. This in turn will help focus future efforts on improving trajectory estimation.

6.2 Calculating the Probability of Conflict

6.2.1 Monte Carlo Simulations

Once the distributions of the trajectory model are developed, the probability of conflict, $P(C)$, between aircraft can be obtained by extrapolating their positions out into the future. The goal is to determine the likelihood that one or more intruder aircraft will violate the protected zone of the host aircraft of interest, thus determining the level of threat to the host. For the discussions in this thesis, unless specifically stated otherwise, the protected zone is defined to be a cylinder 5 nautical miles in radius and extending 1000 feet above and below the host aircraft.

Figure 6-4 shows an example of the predicted position distributions for a single aircraft traveling with a nominal speed of 400 knots. Intent information of a 45° right turn at a waypoint 100 nautical miles ahead was assumed to be known. At each time shown in the figure, the aircraft is predicted to lie within the corresponding region with a probability of 0.9999.

Figure 6-4 was generated from Monte Carlo simulations using some of the baseline trajectory distributions shown back in Figure 6-2. It included along-track fluctuations (Gaussian with standard deviation σ = 15 knots) and cross-track variability (Gaussian with standard deviation σ = 1 nautical mile at steady-state). At t = 0 minutes, the position of the aircraft is known exactly since no sensor errors were included in this example for simplicity. As shown, the predicted position error grows both along-track and cross-track in time, but generally follows the intended path.

If for some reason there is uncertainty that the aircraft will make the intended turn at the waypoint, an additional confidence probability can be included. In such a case, the position distribution would split into two separate regions: one for the case in which the turn is followed, and one for the case in which the turn is not followed. A situation where this type of modeling might prove especially useful is in vertical conflict analysis where an intruding aircraft may not be entirely trusted to level off at the expected altitude.

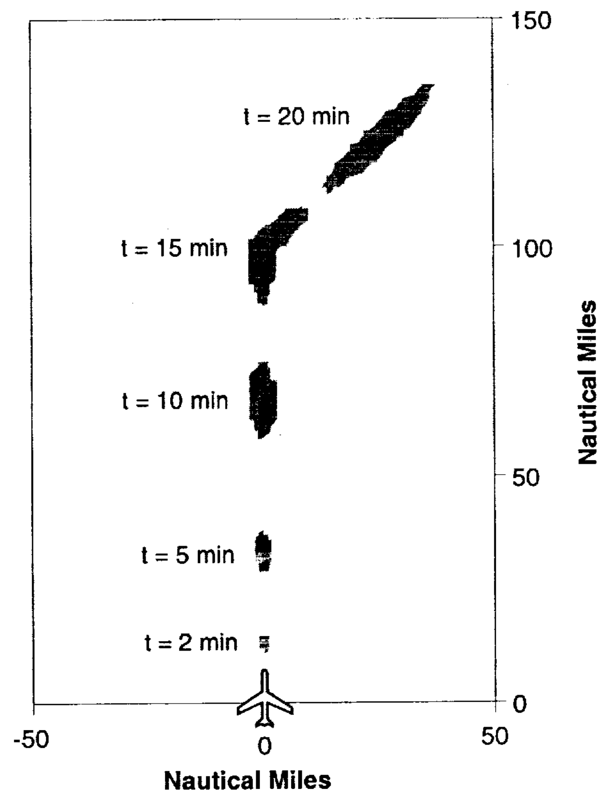


Figure 6-4: Example Projected Position Uncertainty

The probability of a conflict, $P(C)$, can be obtained by extrapolating each aircraft's position in a similar manner. Given the initial locations, speeds, and headings of the aircraft, the $P(C)$ can be estimated through Monte Carlo simulation. Each Monte Carlo run consists of propagating the trajectories over time (using point-mass dynamics) and determining whether separation minimums of the protected zone are violated. The trajectories vary randomly with each run according to the uncertainty distributions chosen to define the trajectory model (e.g. Figures 6-2 and 6-3). In each iteration, a random sampling from each distribution is chosen and used for the trajectory propagation. For instance, one run might have the intruder make a 14 degree heading change 1 minute into the flight; while another run may have the intruder follow a straight line path for over 30 minutes. After a certain number, N, of Monte Carlo runs, a count of the number of protected zone intrusions, x, is totaled. Dividing x by N is then an unbiased estimator of $P(C)$. A schematic of the Monte Carlo iterative process is shown in Figure 6-5.

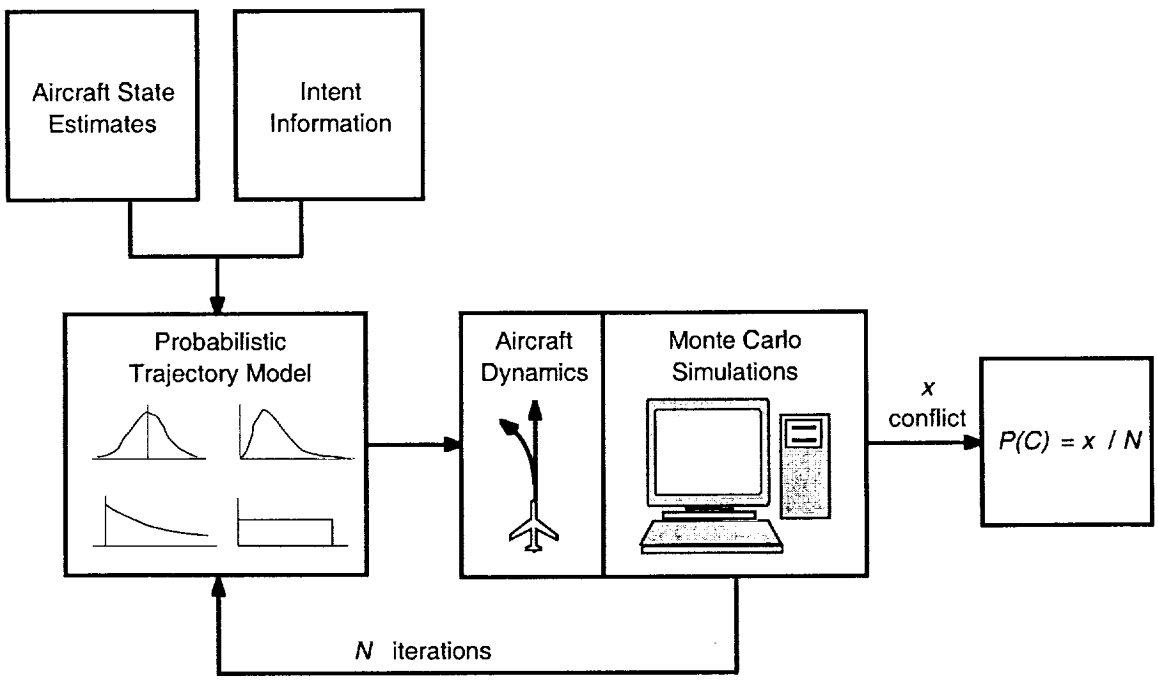


Figure 6-5: Monte Carlo Simulation

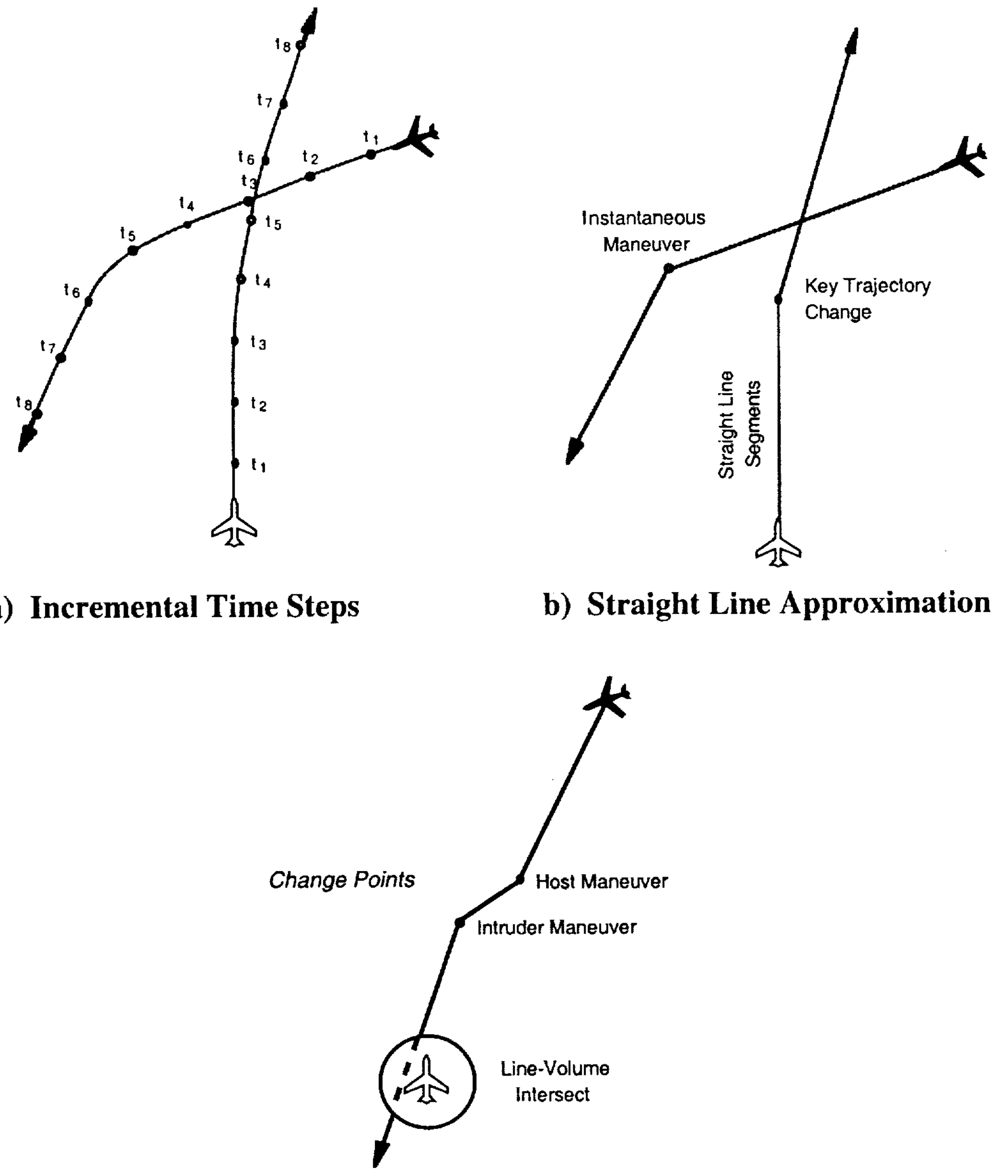
6.2.2 Propagation Method

When propagating the aircraft into the future, one possible approach is to check for a protected zone violation at the end of incremental time intervals as depicted in Figure 6-6a. For each time interval, Δt, the position of each aircraft is calculated and horizontal and vertical ranges are checked against minimum separation requirements. This method requires that the intervals be small enough so that intrusions which might occur in between each end point are not missed. However, reducing Δt can greatly increase the computational time. The problem becomes a tradeoff between the maximum projection time and the time required to calculate $P(C)$.

A more computationally efficient approach can be devised by assuming the trajectories to be comprised of a series of straight line segments with instantaneous trajectory modifications. This simplified assumption is represented in Figure 6-6b, where *change points* approximate key course changes in the trajectories previously depicted in Figure 6-6a. In between change points, the velocity vector of each aircraft is constant. Separate change points reflect a new heading, altitude rate, or speed change in the trajectory.

The simplification can lead to some inaccuracies from the trajectory model due to general displacement errors as discussed in Section 4.1.2. This is due mainly to the step changes around the transition regions induced by the simplified model. If deemed necessary, an added lag time can be included prior to the step changes to account for aircraft dynamics during the maneuver. Krozel et al. [1997] found the approximation of step maneuvers (bank angle and vertical rate) to adequately match simulated Boeing 737 dynamics provided a 2 to 5 second lag was included prior to initiation of turn and altitude changes. For speed maneuvers, they conceded to using an acceleration or deceleration component to better model the relatively slow dynamics of aircraft speed changes.

The approach is further simplified by transformation into a relative coordinate frame such as one with respect to the initial host aircraft position (shown in Figure 6-6c). The protected zone is placed around the origin representing the position of the host, and the relative trajectory of the intruder aircraft is propagated. Because of the assumptions made, the trajectory is comprised of straight line segments with each endpoint corresponding to a course or speed deviation by *either* the host or the intruder aircraft. The task is then to determine if any individual line segment passes through the protected zone around the host aircraft at the origin. Analytic geometry can be used to derive the solution for the intersection between the equation of lines (either finite or infinite) and a 3-D volume (the protected zone cylinder). The equations can be found in Appendix C.



a) Incremental Time Steps

b) Straight Line Approximation

c) Line-Volume Intersection in Relative Frame

Figure 6-6: Aircraft Trajectory Propagation

Not only does this method detect conflicts along the entire path, rather than at discrete points; but the computational time is decreased by orders of magnitude compared to the incremental time approach of Figure 6-6a. Also, the method is insensitive to the time scale of the projection (the equations for a line-volume intersection are applicable to an infinite line); it only depends on the number of course or speed changes that occur between both aircraft.

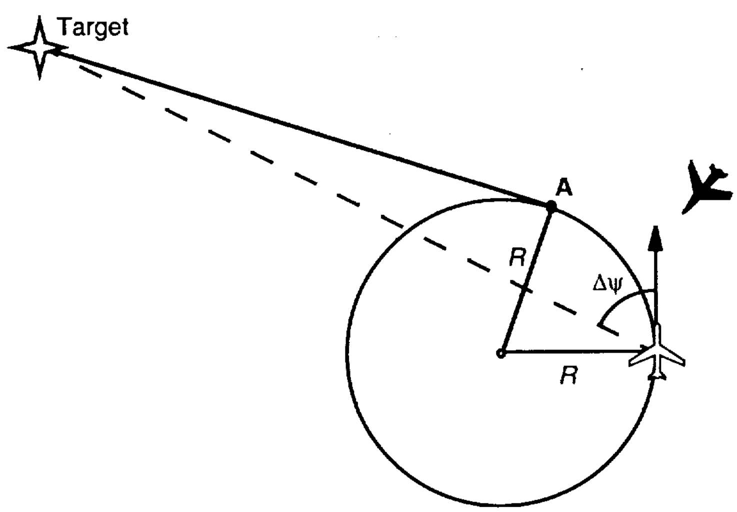
In some instances, it may be desirable to model the course transitions more accurately, as would be the case if an encounter is expected to be in the vicinity of the maneuver transition. If the maneuver is far ahead into the future, an instantaneous maneuver is less likely to be a factor as the uncertainty in the path grows over time. When the maneuver is expected in the near future, a more accurate representation of the maneuver may be in order. This might be the case if an intruding aircraft is relatively close and the crucial conflict point is somewhere near the region of the course change.

Take, for instance, the example shown in Figure 6-7a where the host aircraft (white) is currently in a turn toward a target waypoint. A trajectory modeled with an instantaneous turn (dashed line with sudden path change of $\Delta\psi$) may be overly simplistic since the actual turn radius can be on the order of 10 nautical miles or so depending on the speed and bank angle. This could lead to a missed detection of the conflict with the intruder aircraft (black) shown in the picture. Thus, it is more accurate to include additional line segments to better represent the actual change in the heading over time. Figure 6-7b shows one additional change point, **A**, added to better approximate the path of the host aircraft during the heading transition.

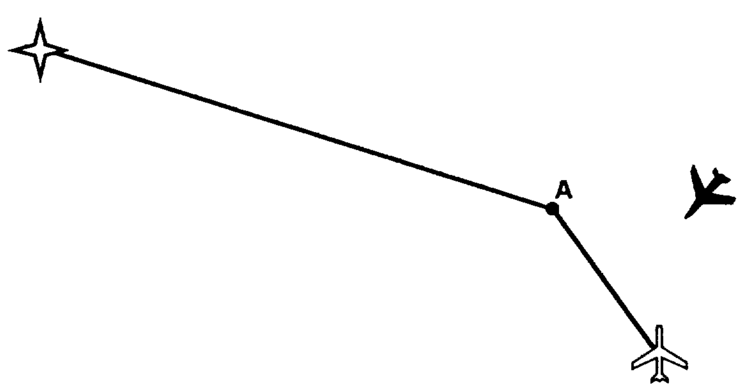
The turn radius (R) can be estimated from the intended bank angle (ϕ), speed (v), and gravitational acceleration (g) using Equation 6.1.

$$R = \frac{v^2}{g|\tan\phi|} \tag{6.1}$$

The center of the turn circle can be approximated to be in the direction perpendicular to the current aircraft heading and at a distance R away. Using geometry, the position of point A can then be determined as a tangent line from the turn circle to the target waypoint position. The result is a two-segment path (shown as a solid line in Figure 6-7b).



a) Instantaneous Heading Change



b) New Segmented Turn Maneuver

Figure 6-7: Heading Change Model with Bank Angle

If more accuracy is desired, the turn can be further sub-divided into additional straight line segments, though at a cost to computational time. The choice depends on the goals of the conflict analysis. For very critical, short scenario analysis such as the case for parallel approach studies, the need for a more accurate path model would be important. For long term conflict probing, the desire for further look-ahead time may take precedence over near term conflict prediction accuracy.

6.2.3 Computational Accuracy

Because Monte Carlo simulations are inherently stochastic, a discussion on the computational accuracy and performance is warranted. The problem posed in calculating $P(C)$ is basically that of estimating a value of proportion, p. Each iterative run is a binomial process in which a conflict (minimum separation criteria violated) occurs or it does not. The number of conflicts, x, divided by the total number of iterative runs, N, provides an unbiased estimator of p with variance σ^2 given by

$$p = \frac{x}{N} \tag{6.2}$$

$$\sigma^2 = \frac{p(1-p)}{N} \tag{6.3}$$

From the Central Limit Theory in probability, the sum of N independent, identically distributed random variables will approach that of a normal distribution as the number $N \to \infty$ [Drake, 1967]. Thus when the number of iterations, N, is sufficiently large (Johnson [1994] suggests $N \geq 200$ for the range of $0.75 \geq p \geq 0.925$), the normal approximation to the binomial distribution can be used to construct an approximate confidence interval for the binomial parameter, p [Johnson, 1994]. For 3σ standard deviation accuracy (99.7%), the error in using x / N (Equation 6.1) to estimate the true $P(C)$ can be computed from Equation 6.4. Noting that Equation 6.4 is a maximum at $p = 0.5$, the upper bound of the 99.9% confidence error can be found by using Equation 6.5.

$$3\sigma = 3\sqrt{\frac{p(1-p)}{N}} \qquad (99.7\%) \qquad (6.4)$$

$$3\sigma_{max} = \frac{3}{2}\sqrt{\frac{1}{N}} \qquad (99.7\%) \qquad (6.5)$$

The tradeoff between the number of iterative Monte Carlo runs, N, and accuracy in the estimate of $P(C)$ is evident from Equations 6.4 and 6.5. The effect can be visualized in Figure 6-8 where upper error bound (Equation 6.5) is plotted versus N.

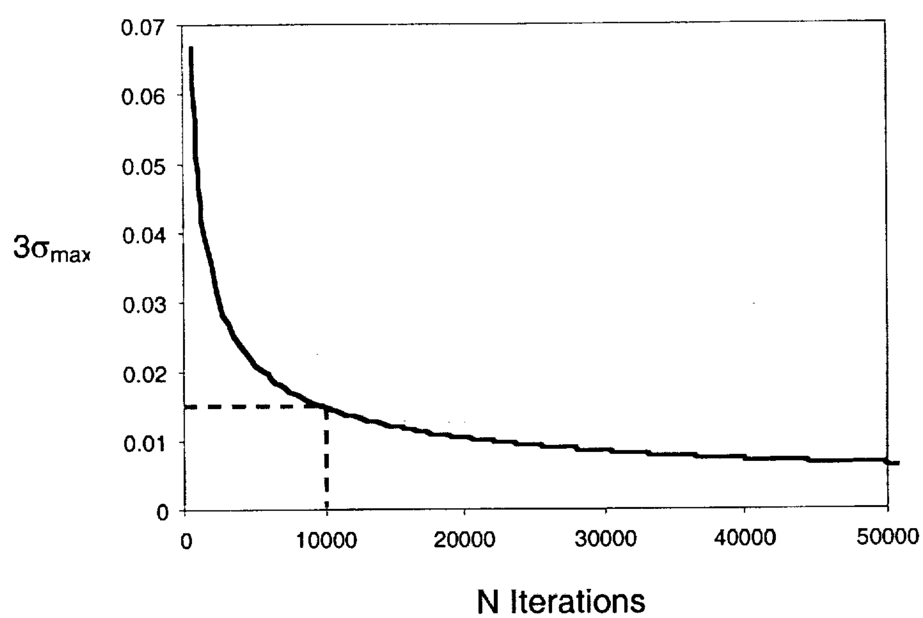


Figure 6-8: Monte Carlo Accuracy as a Function of Iterations

For the examples in this paper, $N = 10,000$ was used as a compromise between speed and accuracy, providing a 3σ error in $P(C)$ of at most ±0.015. There are diminishing returns on improved accuracy as the number of iterations increases beyond this point at the expense computational processing. Originally developed on a Silicon Graphics, Inc. Indigo Elan 4000 Workstation (purchased in 1993), computational time to obtain $P(C)$ from 10,000 iterations was on the order of 1 second for one pair of aircraft. Use of newer workstations (Silicon Graphics Indigo R10000 and Octane MXE) have shown a 2 to 3-fold increase in speed; bringing the up possibility of using the Monte Carlo simulations as part of a real-time conflict alerting probe. There are some advantages to such a system, and this concept will be brought up in the next chapter. A test bed system has already been incorporated into part-task simulators at Massachusetts Institute of Technology (MIT) and NASA Ames Research Center.

6.3 Conflict Probability Maps

Given the relative speed, heading, and altitude between a host and intruder aircraft, a conflict probability map can be constructed to display the locations where the intruder aircraft currently must be in order to result in a conflict at some later time. As an example, assume two aircraft are co-altitude and both flying with a velocity 400 knots with offsetting headings of 30 degrees. For simplicity, assume the host is flying directly North at a heading of 360° and the intruder's heading is 330°. Figure 6-9 shows a conflict map of the likelihood of conflict for this specific encounter scenario. The host aircraft is shown in white at the lower left.

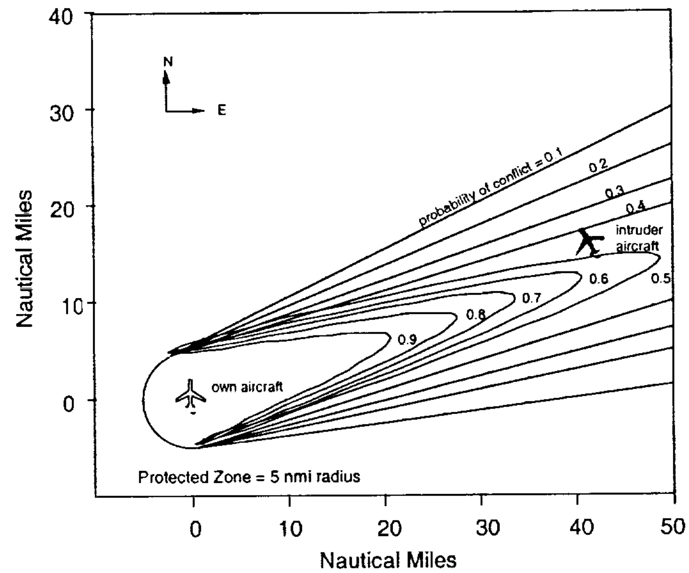


Figure 6-9: Example Conflict Probability Map

The plot shows the conflict probabilities for an intruder aircraft in the surrounding airspace relative to the host aircraft. For example, an intruder in the position shown in the figure (black aircraft) will cause a conflict in the future with probability $P(C) = 0.45$. If the intruder were farther north or east of the host aircraft, the probability of conflict would be lower. The plot shows actual data based on 10,000 Monte Carlo simulations at each 1 nautical mile spacing. The trajectory model was based on the uncertainties presented in Figure 6.2 and the random heading change of Figure 6.3 with λ_1 turns/hr. The magnitude of heading change was limited to 20° in either direction and uniformly distributed (see Figure 6.3).

6.4 Summary

In this chapter, a method was developed to calculated the probability of conflict, $P(C)$, based on Monte Carlo simulations. This approach allows a large number of complex variables to be handled easily and efficiently in what would otherwise be problems without tractable analytical solutions. Error bounds on the accuracy of the calculations can also be determined. In addition, a method to visualize the conflict situation through the use of probability contours was presented.

Chapter 7

Example Applications

In this chapter, several examples will be given to illustrate the utility of the probabilistic methodology in different encounter scenarios. The examples demonstrate the relative ease in which the methodology can be applied to both simple and complex situations. The method can handle encounters in 3-D, with or without intent information, and probabilistic trajectories. Since the approach is based on Monte Carlo simulations, the derivations are not hindered by the complexity of the encounters.

7.1 Horizontal Conflict Examples

As a simple example, assume two aircraft (host and intruder) are co-altitude and both flying with a velocity of 400 knots in opposite directions. If the intent of each aircraft is known, then potential conflict situations can be predicted. Assume that both aircraft have declared their intentions to maintain their current speed, heading, and altitude. This might be inferred, for example, through datalink of autopilot mode control settings.

The potential conflict map as obtained through Monte Carlo simulation is shown in Figure 7-1 (using the baseline trajectory uncertainty model from Figure 6.2). As a reminder, in this example the intruder aircraft is traveling in the opposite direction as the host. The chart is shown relative to the host aircraft located at the origin (0, 0) with its track pointing up. The top of the chart is 200 nautical miles ahead of the host aircraft and represents a 15 minute time frame. Contours of constant conflict probability are shown starting at $P(C) = 1.0$ around the host aircraft and decreasing in increments of 0.1. For example, the intruder aircraft shown in the figure 100 nautical miles ahead of the host aircraft will produce a probability of nearly 1.0. Variability and coarseness of the contours are a result of the accuracy of the Monte Carlo simulations. In this case, because the trajectory uncertainties are small, the corridor where aircraft must be located to generate conflicts is relatively narrow. Although the example shown is for a specific relative geometry and speed, similar maps can be generated for any situation.

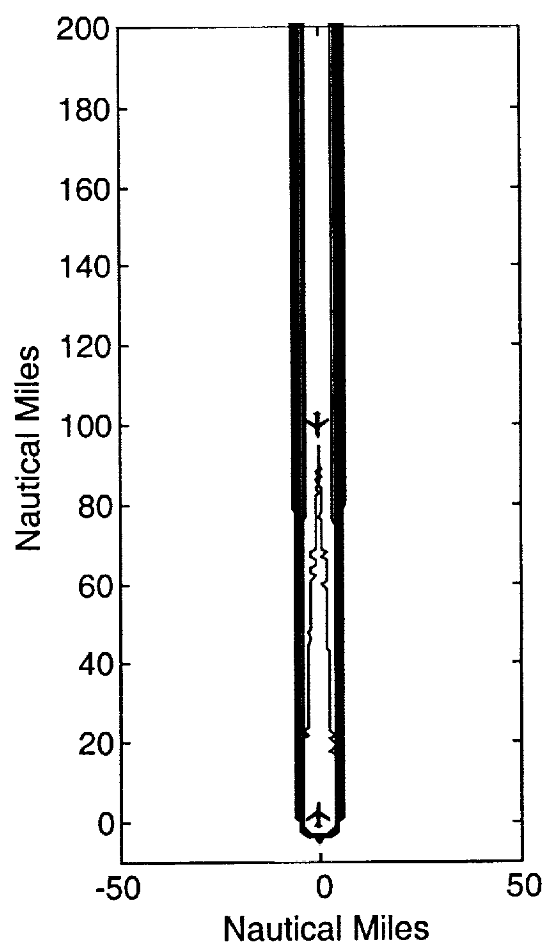


Figure 7-1: Intruder and Host Maintain Course

A more interesting case to observe is when aircraft may change course at some time within the foreseeable future. In many cases, the intentions of each aircraft are not known for certain, but information regarding rules-of-the-road, past experience, or flight restrictions can be helpful in establishing the likelihood of various trajectories. In Figure 7-2, the intruder aircraft is still headed in the opposite direction as the host, but now no explicit intention to maintain a straight course is assumed.

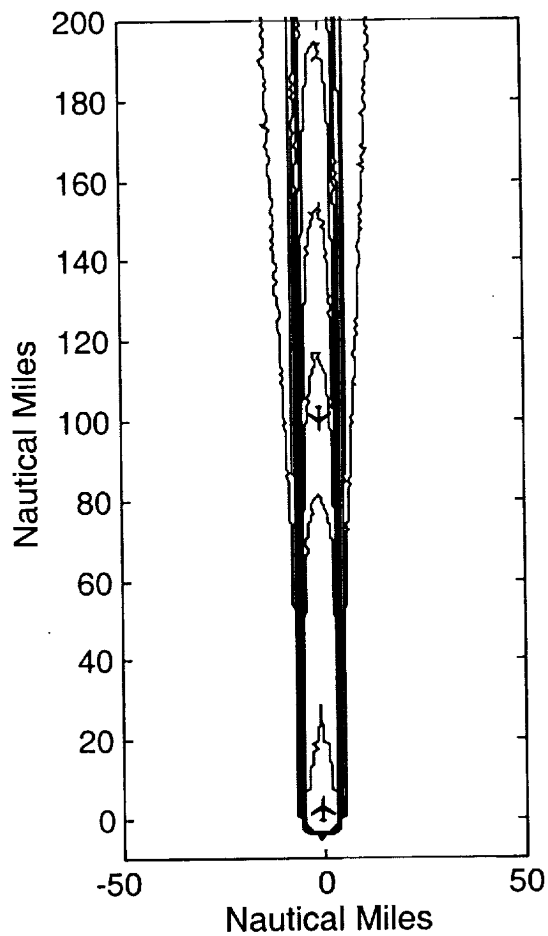


Figure 7-2: Potential for Intruder Course Change

For this particular case, the likelihood that the intruder would make a heading change is modeled as Poisson distribution with an average rate of $\lambda_1 = 4$ turns per hour (see Figure 6-3). Also, the hypothetical flight rules in the airspace are assumed to require aircraft to restrict heading changes to less than 20° within a 15 minute period. Thus, potential changes in heading were modeled with a uniform distribution between ±20°.

The resultant conflict map is shown in Figure 7-2, again using contour spacing of probabilities of 0.1. Note that the probability of conflict decreases more rapidly as one moves farther from the host aircraft The same intruder 100 nautical miles ahead of the host will now cause a conflict with a probability of approximately $P(C) = 0.83$ because there is some chance that the intruder will perform a turn.

In the next example shown in Figure 7-3, additional intent information regarding knowledge of waypoints is added. For this case, the intent is supplied by the host aircraft in terms of 3 future waypoint locations in which the host will shift its flight path laterally. Again, the conflict map is shown with contour spacings of 0.1. Here, the intruder aircraft 100 nautical miles ahead of the host will not create a conflict as long as the intended path is followed.

Comparing Figures 7-2 and 7-3 provides some insight into the potential benefit of intent information. Consider for example the flight path shown in Figure 7-3. If the host's waypoint information was not used in the conflict detection, the situation would likely be modeled as shown in Figure 7-2, resulting in a conflict alert. Such a conflict would be unnecessary, however, since as Figure 7-3 shows, there would not be a conflict with the intruder aircraft. The intent information has improved the modeling accuracy of the trajectory ($\mathbf{W} \rightarrow \mathbf{T}$) and thus increases the chance of making an better informed decision about alerting.

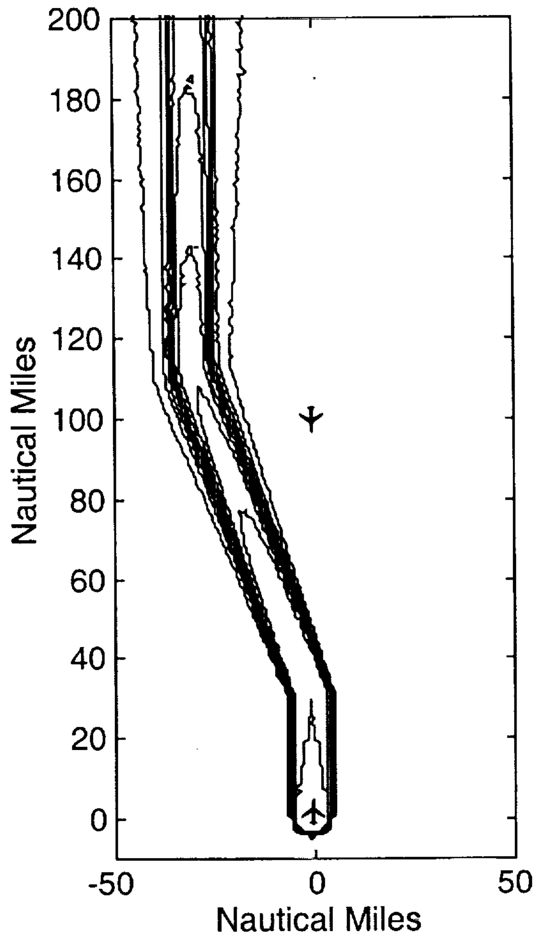


Figure 7-3: Host Aircraft Following Waypoints

Conflict maps can also be utilized in the examination of avoidance maneuver options for conflict resolution. Figure 7-4 shows an example 30° right turn avoidance maneuver made by the host aircraft in response to a conflict alert in the example from Figure 7-2. An additional uncertainty was included to represent variability in pilot response time in initiating the turn maneuver. The latency time was modeled as a Gamma distribution with an average of 1 minute and skewed with a 95% probability of the maneuver occurring within 2 minutes.

Comparison of Figure 7-2 and Figure 7-4 shows the effect of the avoidance maneuver on the probability on conflict. Similar analysis can be performed to determine what other avoidance options (e.g. heading, speed, or altitude changes) can be used for the resolution. For multiple aircraft in the airspace, the maneuver can be checked to see if it induces addition conflicts which would not have occurred without it.

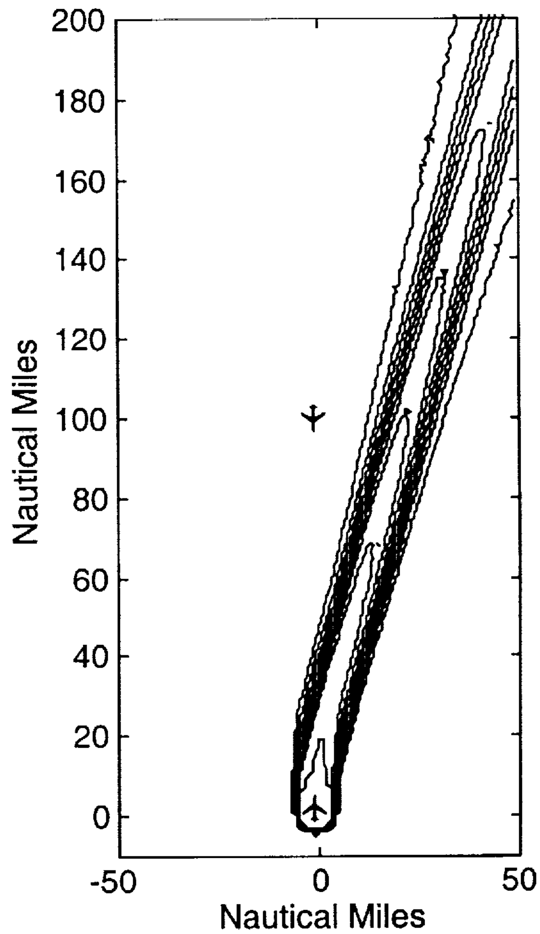


Figure 7-4: Host Aircraft Turns 30°

7.2 Vertical Conflict Examples

To more fully illustrate the utility of the probabilistic methodology (especially in more complex situations), several examples will be shown here to analyze the effects of intent information on conflicts in the vertical plane. Rather than depict conflict maps, however, the discussion will revolve around false alert and missed detection rates using the SOC methodology. In these examples, two aircraft are flying in opposing directions with the intruder currently above the host aircraft by 5000 feet. Suddenly the intruder descends directly toward the host aircraft at 1000 feet per minute. The uncertainties are modeled using the baseline model shown back in Figure 6-2, and a conflict is defined as a loss of minimum separation of 5 nautical miles in the horizontal plane and 1000 feet in the vertical plane.

Two cases will be considered here. In the *baseline* case, it is not known whether the intruder will level off at some point or continue its descent beyond the host aircraft's altitude. The vertical profile of the intruder is modeled such that it is equally likely that the intruder will level off at any altitude within 10,000 feet of its initial descent point. Thus, a conflict may exist (the intruder continues to descend into the host) or a conflict may not exist (the intruder levels off safely above the host). The situation is depicted in Figure 7-5a.

In the *intent* case, datalinked information from the intruder indicates that it will continue its descent at 1000 feet per minute through the host aircraft's altitude (Figure 7-5b). For simplicity, it is assumed here that the aircraft maintains this descent rate perfectly. However, there may likely be variability in the descent rate as pointed out by Paielli and Erzberger [1999]. Fortunately, the Monte Carlo method presented in this thesis could very easily handle this additional parameter without much additional complexity or loss of computational performance.

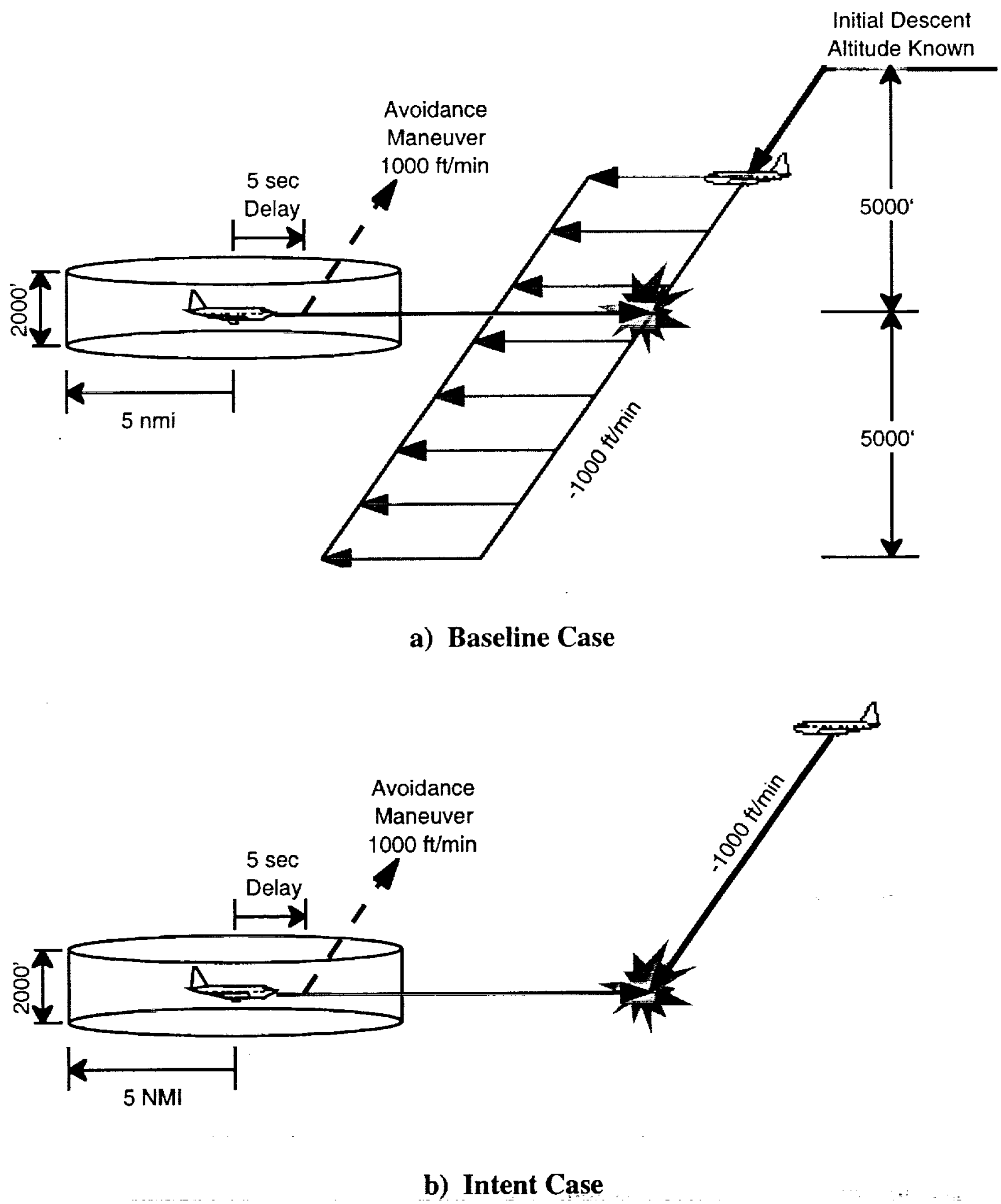


Figure 7-5: Vertical Conflict Examples

SOC curves for both cases are plotted in Figure 7-6. The assumed resolution maneuver is a 5 second delay when the conflict alert occurs, followed by a 1000 feet per minute climb. Variability in these response parameters could be added to the analysis as well, but are not included here for simplicity. The intent case SOC curve is shown by the solid line which happens to follow along the y-axis; the SOC curve for the baseline case is shown by the dashed line. Operating points for each case are shown in terms of the time at which the conflict alert occurs in increments of 10 seconds relative to the time to Closest Point of Approach (CPA) (assuming a straight line projection of the current velocity vector).

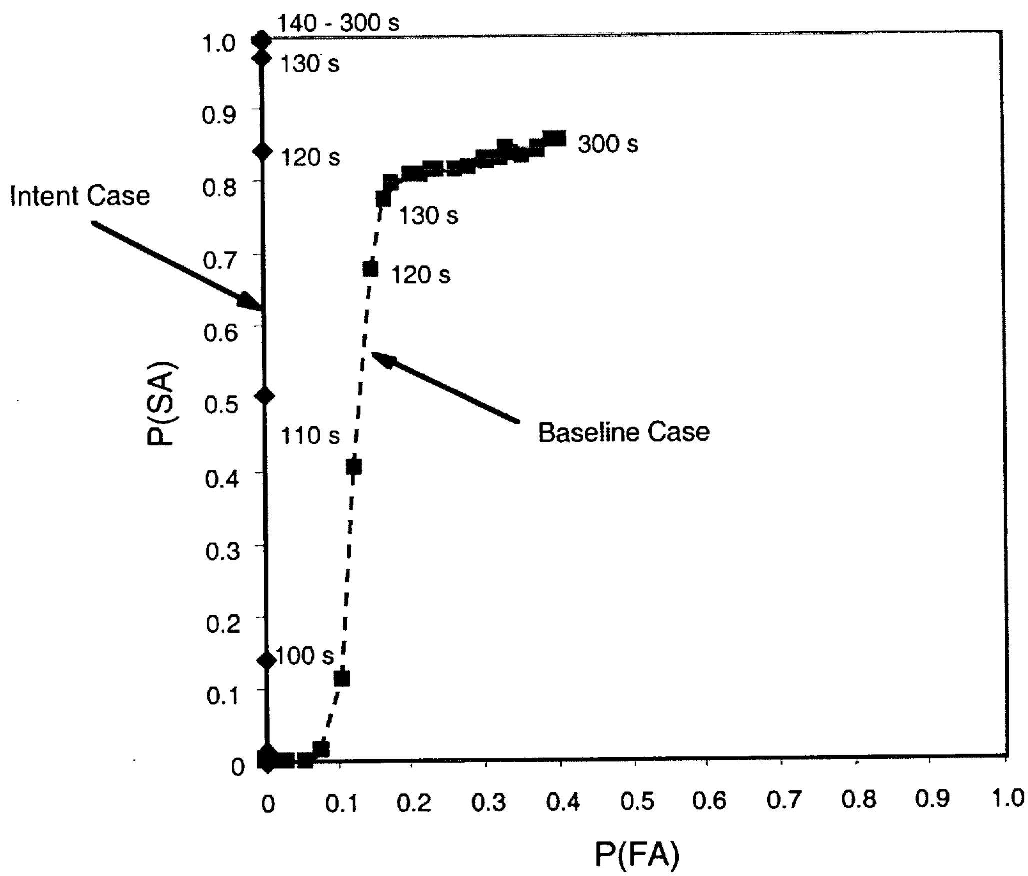


Figure 7-6: SOC Comparison of Baseline vs. Intent Cases

The SOC curve in Figure 7-6 shows that essentially an ideal alerting decision could be made in the intent case provided that the intended path was indeed followed by the intruder aircraft. By alerting at any time prior to 140 seconds before CPA, the host aircraft could avoid a protected zone violation with approximately 100% confidence

$\{ P(SA) \approx 1.0 \}$. Simultaneously, because the uncertainties in this case are relatively limited, the probability of false alarms is approximately 0. Alerting with less than 140 seconds to CPA reduces the probability of a successful alert as shown.

In the baseline case, there is a possibility of the intruder aircraft leveling off above the host aircraft. Thus, a climbing resolution maneuver may actually *induce* a conflict that would not otherwise have occurred. The net effect is an increase in the probability of a false alarm (a conflict would not have occurred) and a decrease in the probability of a successful alert (the avoidance maneuver induces a conflict). As a result, the performance of an alerting system in the baseline case would be lower than in the intent case.

Notice that the baseline SOC curve shows that the successful alert probability, $P(SA)$, cannot be increased beyond approximately 0.8 without greatly increasing the false alarm probability. Assuming that the intentions of the intruder aircraft were not known, this example shows the difficulty placed on the alerting system designer to develop a suitable threshold for this type of encounter. A viable option would be to examine a more aggressive climb maneuver, or else look into a different avoidance option all together. Figure 7-7 shows the SOC curves for a 1000 feet per minute descent and a 30° turn avoidance maneuvers. The climb maneuver from the previous figure is redrawn for comparison.

The advantage of utilizing a descent or turn maneuver in this particular encounter situation is obvious from the SOC diagram. Both these maneuvers allow for a threshold setting that is more ideal than the original climb maneuver (curves are closer to the ideal position of $P(FA) = 0$, $P(SA) = 1$). In addition, they also allow for the alert to be delayed longer before action is required. For example, the plots indicate that a descent maneuver can be delayed until 80 seconds to CPA before a large drop off in *P(SA)* is experienced. For the turn maneuver the delay can be as long as 100 seconds prior to CPA. In comparison, the climb option requires the alert to be given much earlier (140 seconds to CPA) if *P(SA)* is to be kept above 0.8. In this situation, the longer the delay, the better since the intruder aircraft may level off in altitude and an alert would never be needed.

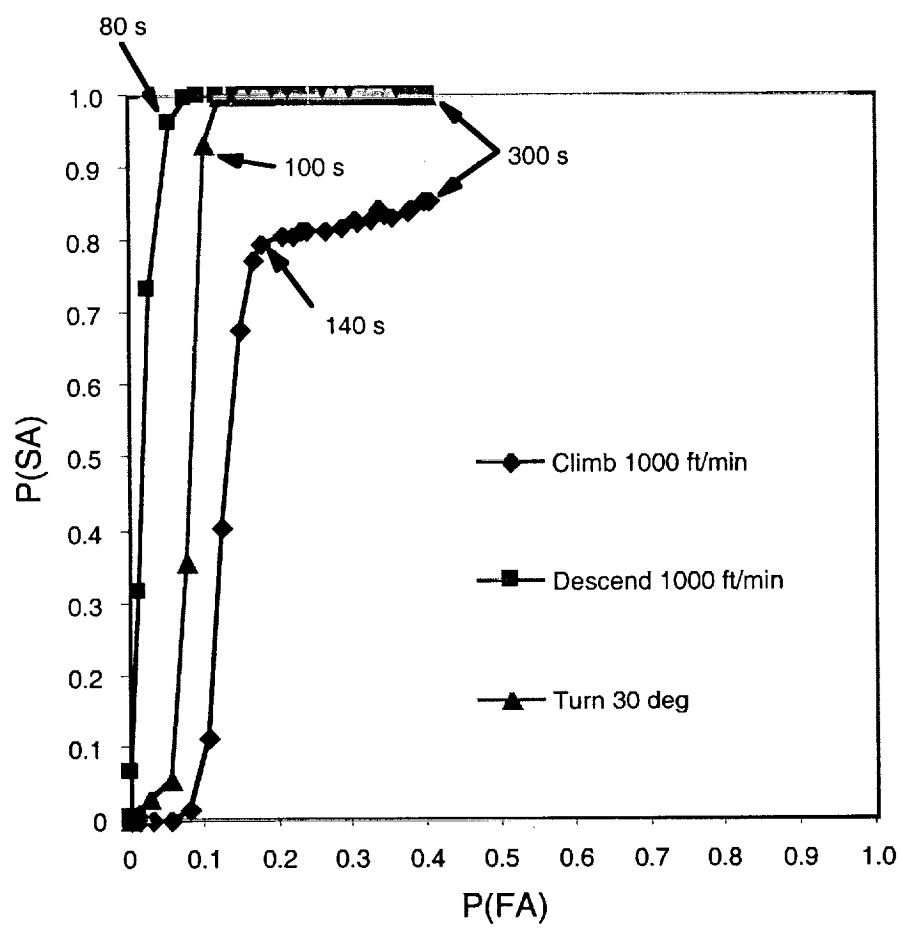


Figure 7-7: SOC Comparison of Climb, Descent, and Turn Maneuvers

The results shown in Figure 7-7 are very interesting when examined from the perspective of uncertainties. Maneuvers which deviate away from regions of high uncertainty will allow for increased alerting performance. In essence, the outcome of the conflict will become more certain. For example, the descending option allows the host aircraft to move away from the possible regions of airspace that might be occupied in the future by the intruder aircraft. As a result, the outcome of a possible conflict is more certain. The same is true with regard to the turning maneuver. By utilizing a horizontal escape maneuver, the host aircraft is removing itself from the major source of uncertainty (i.e. the intruder leveling off) that is involved in the encounter. The results help substantiate the desire to include horizontal resolution advisories in new versions of conflict avoidance systems.

7.3 Summary

The case studies presented in this chapter demonstrate the utility of the probabilistic methodology to handle both simple and complex encounter situations. The examples showed how intent information could be used to improve the quality of the conflict detection problem by increasing the prediction accuracy. Both probability contour maps and SOC curves were used in presenting the results.

Chapter 8

A Probabilistic Real-Time Alerting Probe

The previous chapters of this thesis have detailed most of the theoretical and computational issues of analyzing conflicts using the probabilistic approach. Its uses as an evaluative tool to investigate conflict scenarios have been shown with examples from the last chapter. Now this chapter will describe taking probability analysis one step further to the development of a real-time alerting probe. The first section will explain the rationale behind the endeavor and the next section will describe the most recent update of the logic to run Monte Carlo simulations concurrently with the real-time updates of aircraft state and intent information. Finally, the advantages and disadvantages behind this work will be discussed.

8.1 Alerting Probe Concept

In order to better understand the potential advantages and disadvantages of probabilistic threshold criteria, a prototype alerting logic was developed based on the concept of probabilistic conflict calculations discussed in the previous chapters. The basis for the logic follows very much in line with the concepts developed for the SOC curves (explained in Section 2.6) using $P(FA)$ and $P(SA)$ and the direct approach of Chapter 5.

The design of the prototype alerting system was guided in part by NASA requirements for their experiments in their Advanced 747-400 Full Motion Simulator as well as their less complicated part-task simulators. The logic was tailored to an airborne system where conflicts were expected to be resolved primarily on the flight deck, though the concept could be extended to a ground-based system to aid ATC. The first prototype assumed only that current state information (position, speed, and heading) was available through inter-aircraft datalink such as from Automatic Dependent Surveillance-Broadcast (ADS-B). The alerting system was specifically designed to provide ample warning time so that strategic maneuvers could be examined and coordination between flight crews could be carried out.

A multi-staged threshold approach was utilized to provide a series of alerts to indicate trends in conflict hazard. This approach allowed the means of implementing the alert to be tailored to the level of threat. Low probability threats resulted in relatively passive alerts such as changing the color of a traffic symbol. High probability, urgent threats produced aural warnings to actively inform the pilots or controllers of the conflict.

The multi-staged approach is shown in the schematic diagram of Figure 8-1. Three stages (marked **1**, **2**, and **3**) produced changes in the traffic display symbology in the cockpit of the host aircraft. As implemented, the outermost threshold provided an initial indication of potential threat more than 10 minutes into the future and up to 200 nautical miles away. In the NASA 747-400 simulator, a hollow traffic symbol on the map display would change color when the first threshold was exceeded and the flight crew was expected to initiate verbal communication with the intruding aircraft in an effort to coordinate a resolution. If the encounter continued, an additional stage would inform the crew of the heightening threat by filling in the traffic symbology on the map display. At stage **3**, an aural Alert Zone Transgression (AZT) was provided to the flight crew indicating that action must be taken to resolve the conflict. At this point, there was still ample time to coordinate resolution with other aircraft. If the conflict continued without resolution, a fourth level of alert called the Authority Transition (**AT**) would inform ATC to take control over the conflict situation.

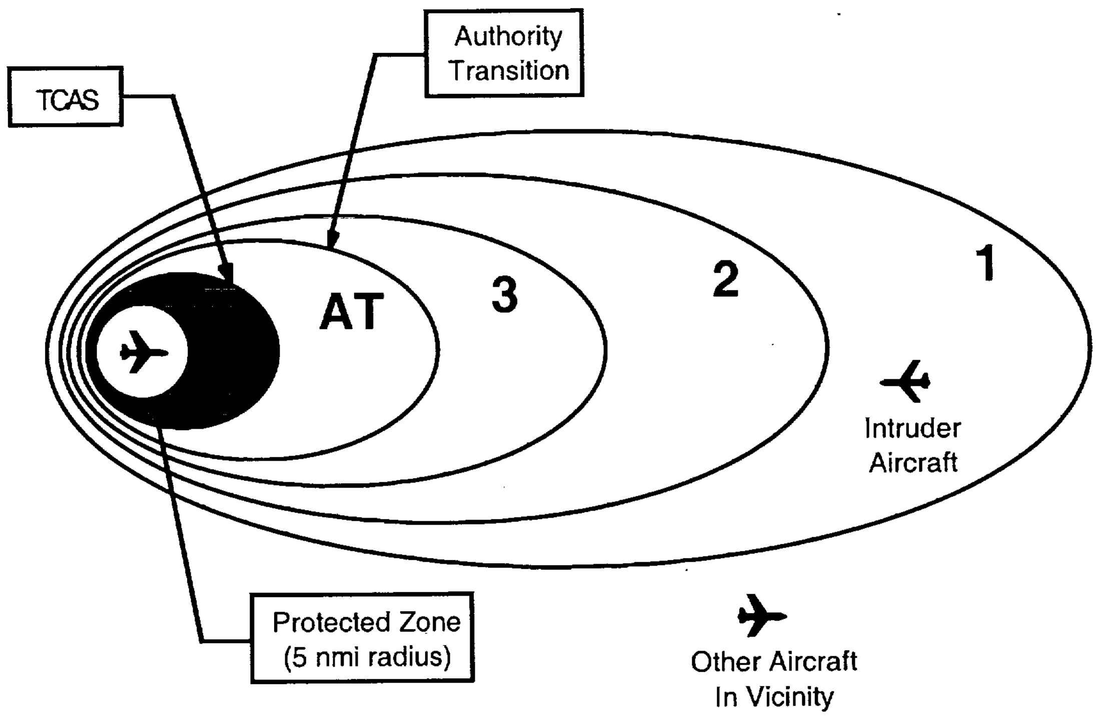


Figure 8-1: Multi-Stage Alerting Probe Concept

The prototype alerting logic was overlaid on top of the current TCAS logic which was not modified and kept in the simulation setup as an independent, final warning system. But because the alerting thresholds on the present TCAS 6.04A version are based on limited variables (range and closure rate), TCAS cannot accurately predict whether a conflict will occur beyond a few minutes. TCAS can track traffic within a range of 40 nautical miles and its earliest alert can be triggered approximately 1 minute prior to the projected closest point of approach [RTCA, 1983; Nordwall, 1997].

The newest version of TCAS (v7.0) became available in 1999, and increased the range to 100 nautical miles using ADS-B via mode-S to transmit additional position, heading, and vertical speed information [Klass, 1998]. The prototype alerting system could also be expected to utilize this same aircraft state data to estimate future trajectories.

Additional requests by NASA required the alerting probe be able to handle various types intent information (i.e. next waypoints, commanded headings, commanded altitudes) if made available. The difference in alerting thresholds with and without intent can be significant as was shown in the example applications of Chapter 7. The availability of intent allows for the more accurate prediction of future aircraft states and thus improve the outcome of the alerting process.

8.2 Prototype Alerting System

The ability to change the parameters of the trajectory model becomes important when intent information is to be considered in the probability computations. Depending on the type of intent information available, the parameters need to be reflected in the trajectory model to reduce errors in the estimates of conflict as explained in Chapters 4 and 5 (**W** = **T**). Also, if the intent of an aircraft changes, the trajectory model needs to adapt to the new information in order to determine possible conflicts along the new path.

In response to new directives within NASA to explore the use of intent information in conflict detection and resolution, a new alerting logic was necessary to accommodate their experimental requirements. A new setup was needed to handle the various levels of intent information, if available, and also adjust dynamically to changes in that information.

Utilizing the propagation method discussed in Section 6.2, specifically that shown in Figure 6-6c, a new alerting system was devised based on running Monte Carlo simulations in near real-time flight. Aircraft state and intent information are passed to the

Monte Carlo simulation engine to be processed, then the hazard level based on the conflict probabilities { $P(FA)$ and $P(SA)$ } is returned after computation is completed. Additional information regarding possible avoidance options and the location of the closest point of approach (assuming straight line extrapolation of the current velocity vectors) were also computed by the alerting logic at the request of NASA. The structure is shown in Figure 8-2.

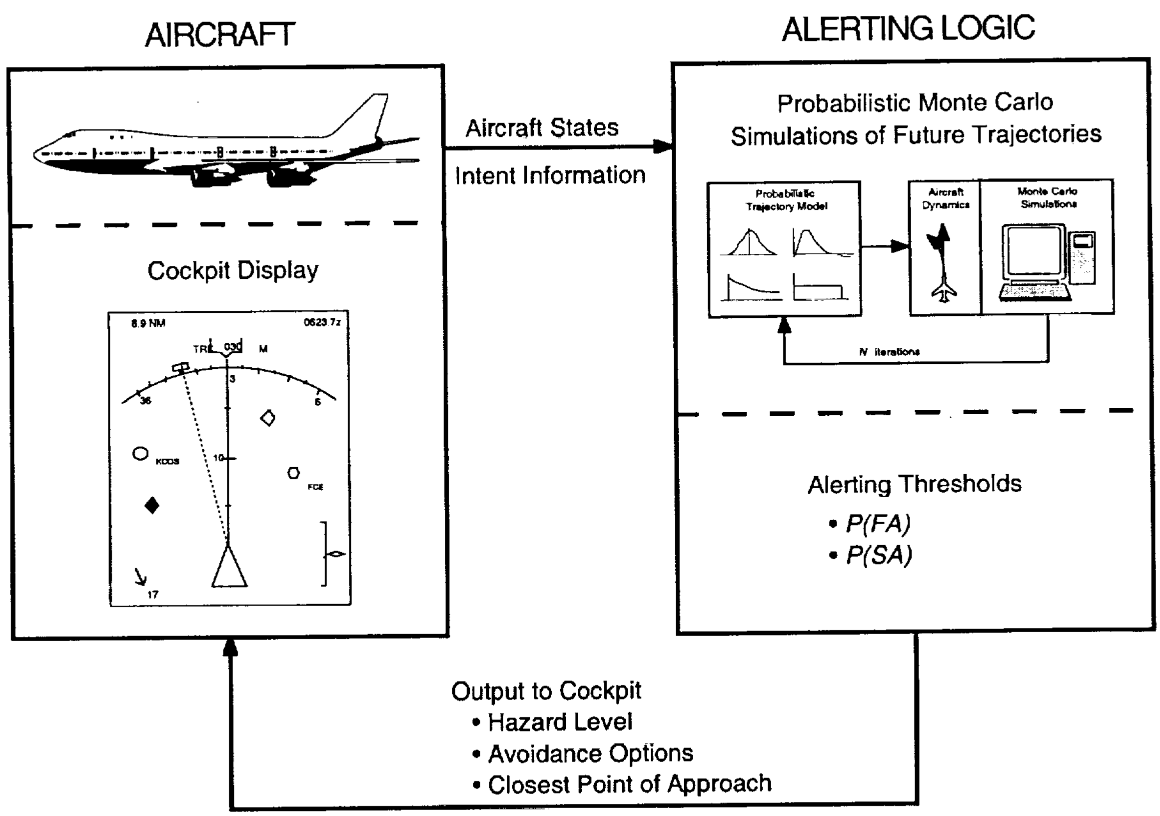


Figure 8-2: Prototype Alerting System Based on Real-Time Monte Carlo Calculations

Currently, the system handles intent information of 4 types: only current state and derivative information, future 3-D positional targets (waypoints, top and bottom of climb and descent), target headings, and target altitudes. These forms were chosen to satisfy requirements for experiments to be conducted by NASA.

This setup is flexible such that the parameters of the trajectory models may be adjusted according to the intent information provided (**W** = **T**). If intent is available, the alerting logic develops the state trajectory based on that information. If there is no information on intent, then the logic assumes a possibility of the surrounding aircraft deviating from their current track. The premise is similar to the results shown back in Figures 7-1 and 7-2. In the former case, the intent of intruding aircraft is to maintain its current heading resulting in a very long, narrow region of high conflict probability. In the latter case, the intruder aircraft is modeled with the possibility of deviating from its current track which results in a wider spread of the probability contours.

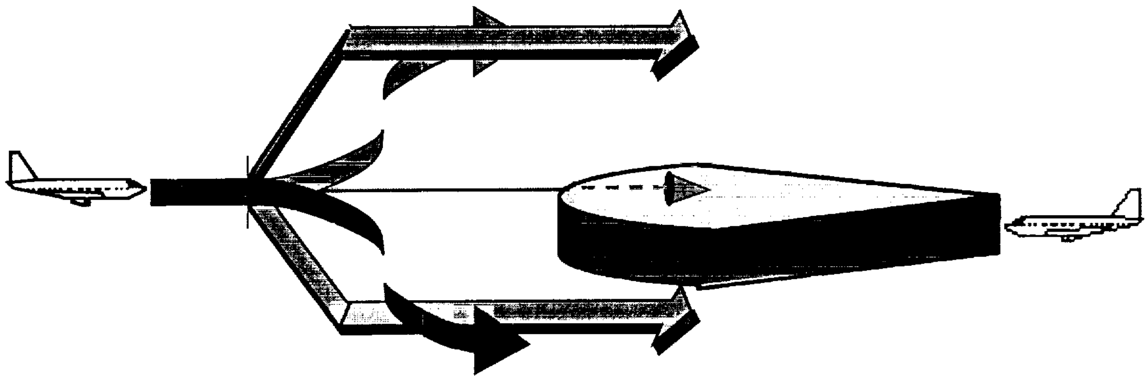
8.3 The Alerting Thresholds

In order to satisfy the requirements set forth by NASA, the prototype alerting system uses four stages of alert as shown in Figure 8-1. The first three stages produce alerts in the cockpit that are intended to aid the flight crew in resolving the conflict before tactical maneuvering is required. At the fourth stage, ATC is notified to issue commands to provide traffic separation. To set the conditions at which these stages are triggered, it is necessary to examine the tradeoffs between $P(FA)$ and $P(SA)$. This requires balancing the likelihood of a conflict against the ability of the host aircraft to avoid a conflict. Since $P(SA)$ is specific to different avoidance options, five standard conflict resolution maneuvers were considered:

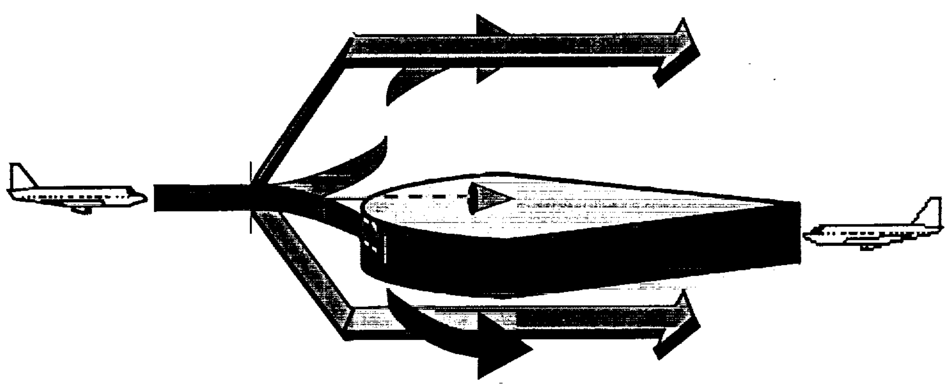
1) Left Heading Change of 30°
2) Right Heading Change of 30°
3) Climb or Descent of 2000 ft/min
4) Speed Increase of 50 kts
5) Speed Decrease of 50 kts

These maneuvers serve as benchmarks for estimating the ability of the host aircraft to avoid a conflict. When the intruder is far from the host aircraft, any of these five maneuvers could be used to resolve the conflict. As the intruder nears the host aircraft, some of these maneuvers may no longer provide the required separation between aircraft. The premise behind the alerting logic is that if a sufficient number of these maneuvers are still available to the pilot, the alert can be delayed. When the pilot's options begin to disappear, an alert should be issued. The concept is illustrated in Figure 8-3.

As shown in the figure, initially the hazard (e.g. intruder aircraft) is sufficiently far away that left and right turns and climb and descent maneuvers can easily avoid the hazard (Figure 8-3a). As the hazard closes in on the host aircraft (Figure 8-3b), the options to resolve the conflict will diminish as different avoidance maneuvers can no longer safely able to avoid the conflict. As shown in Figure 8-3b, the right turn maneuver is depict as an ineffective option to safely avoid the incoming intruder.



a) Avoidance Maneuvers Still Available



b) Avoidance Maneuvers Begin to Diminish

Figure 8-3: Loss of Available Avoidance Options

A maneuver was defined to be *available* to the host aircraft if, by performing the maneuver, the probability of a conflict was reduced to less than 0.05, i.e. $P(SA) > 0.95$. The five maneuver options listed above included the probabilistic response time depicted earlier in Figure 6.3 (with a mean latency of 1 minute). Thus, when a maneuver was deemed to be not available, safe separation could still be achieved if the pilot reacted more quickly or more aggressively than assumed in the model.

During actual simulator runs, the logic would determine the number of avoidance maneuvers remaining or available, N, to resolve a conflict with the intruder. The latest version of the alert logic computed these values in near real-time using Monte Carlo simulation during runtime. The logic would compute $P(FA)$ from $P(C)$ $\{P(FA) = 1 - P(C)\}$ and also the various values of $P(SA)$ for the five standard avoidance options. By comparing N with $P(FA)$, the appropriate alert stage was defined as shown in Table 8-1.

The leftmost column of Table 8-1 shows the probability of a conflict if the host aircraft continues along its current trajectory. This assumes that the intruder's trajectory can be represented by the model discussed earlier. The rightmost column shows $P(FA)$, which as discussed earlier is related to $P_N(C)$ by Equation 2-2. The other columns indicate the defined alert stages as a function of N. Generally, the more options available to the pilot, the lower the alert stage. For example, if $P(FA)$ is 0.35 and there are two avoidance maneuvers available, then the alert stage is **2**. If $P(FA)$ drops below 0.3 or if N is reduced to one, then the alert stage increases to **3**. If $P(FA)$ drops below 0.1, then the **AT** stage is triggered.

Table 8-1: Alert Level Classification

	Number of Avoidance Maneuvers Remaining, N				
$P_N(C)$	None	One	Two	Three +	$P(FA)$
0.0 - 0.1	-	-	-	-	0.9 – 1.0
0.1 – 0.2	**1**	**1**	-	-	0.8 – 0.9
0.2 – 0.3	**1**	**1**	-	-	0.7 – 0.8
0.3 – 0.4	**2**	**1**	**1**	-	0.6 – 0.7
0.4 – 0.5	**2**	**2**	**1**	**1**	0.5 – 0.6
0.5 – 0.6	**3**	**2**	**2**	**1**	0.4 – 0.5
0.6 – 0.7	**3**	**3**	**2**	**2**	0.3 – 0.4
0.7 – 0.8	**AT**	**3**	**3**	**2**	0.2 – 0.3
0.8 – 0.9	**AT**	**3**	**3**	**3**	0.1 – 0.2
0.9 – 1.0	**AT**	**AT**	**AT**	**AT**	0.0 – 0.1

8.4 Evaluation of Prototype System

To better understand the underlying design process, the thresholds from Table 8-1 can be mapped into SOC curves. Figure 8-4 shows SOC curves for two co-altitude aircraft on a collision course along flight paths at right angles to one another. SOC curves corresponding to each of the five resolution maneuver options are shown.

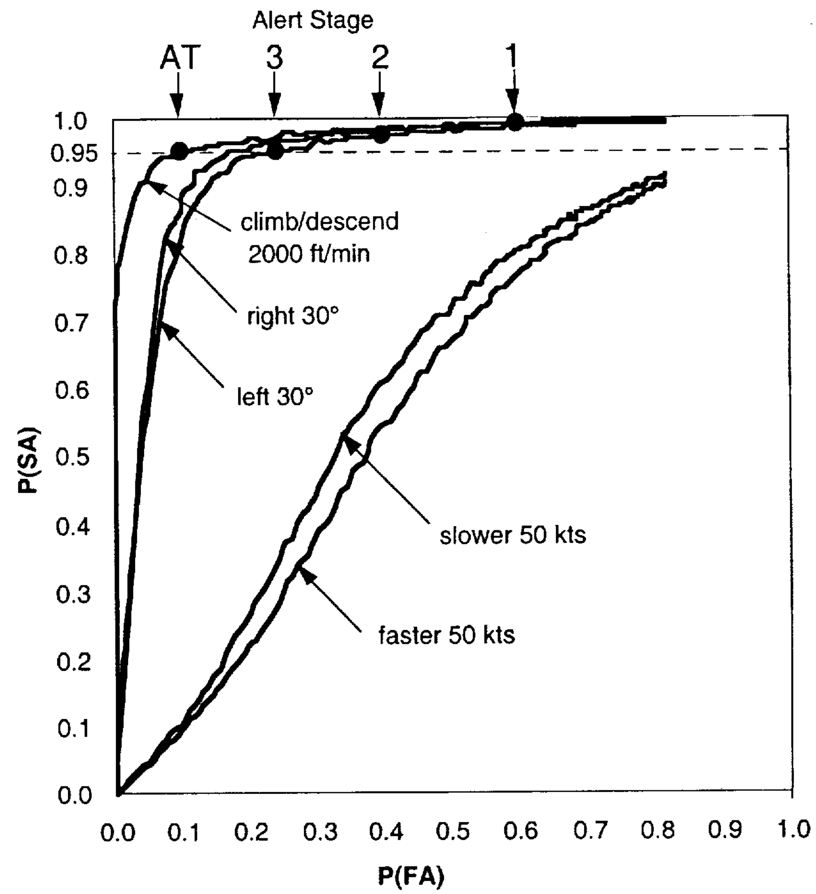


Figure 8-4: SOC Curve for Aircraft on Perpendicular Tracks

When the intruder is far from the host aircraft, the situation maps into the upper right of the plot: it is likely that a conflict will not actually occur { $P(FA) \rightarrow 1$} and it is likely that any avoidance action would resolve the situation { $P(SA)$ for each of the five avoidance maneuvers is 1}. Data for Figure 8-4 were not obtained beyond 200 nautical, and thus the SOC curves in the figure do not extend all the way to the upper right corner.

As the intruder continues on a collision course, it becomes more clear that a conflict will occur: $P(FA)$ decreases and the situation moves from right to left along the curves. Thus, $P(FA)$ is related to the distance between aircraft and to the time before closest point of approach. As $P(FA)$ decreases, $P(SA)$ also decreases in differing amounts according to the different SOC curves. The effectiveness of a given maneuver depends on how slowly its $P(SA)$ decreases. When a curve's value of $P(SA)$ drops below 0.95, the corresponding avoidance maneuver is no longer available. Thus, as the situation progresses to the left in Figure 8-4, the different avoidance maneuvers become unavailable, in order, from speed changes to turns and finally to climb or descent. Thus, the SOC curves show that for this case, vertical maneuvers are the most effective.

The first maneuvers to become unavailable are the speed change maneuvers, at $P(FA)$ of approximately 0.9. This is because large speed changes are generally required to resolve conflicts in the time scales under consideration.

Until $P(FA)$ drops below approximately 0.25, turns and climb/descent avoidance maneuvers will still provide the required separation. At approximately $P(FA) = 0.25$, however, a 30° left turn maneuver is no longer an option. At approximately $P(FA) = 0.2$, the 30° right turn is also no longer an option. When $P(FA)$ reaches approximately 0.1, the climb/descend options become unavailable.

At a given value of $P(FA)$, N corresponds to the number of SOC curves that have values above $P(SA) = 0.95$. Figure 8-4 also shows when the four alert stages are triggered as a function of $P(FA)$. Cross-referencing with Table 8-1, stage **1** is triggered when N is three or more and $P(FA)$ drops to 0.6. Stage **2** is triggered when $P(FA)$ drops to 0.4 and Stage **3** is triggered when N drops to two. Finally, the AT stage is triggered when N drops to zero. Although Figure 8-4 shows SOC curves for a direct collision between two aircraft on perpendicular flight paths, other geometries produced similar patterns.

The five avoidance maneuvers used here are intended to represent strategic maneuver limits. It should be reiterated that a large response time (mean = 1 min.) is modeled in the avoidance maneuvers (see Figure 6-3) and that when N is zero, the host aircraft can still maneuver out of the conflict. A more aggressive, tactical maneuver such as a 45° heading turn or a combined climbing turn may still be available when the five assumed strategic maneuvers are not.

Further examination of the SOC curves show that speed changes make only a limited contribution to the prototype logic. In many cases, a speed change of greater than 50 knots is required for adequate separation with 95% confidence. As can be seen from Figure 8-4, the SOC curves for the speed maneuvers deviate only slightly from the diagonal. Thus, it is difficult to provide successful, necessary alerts with speed control alone. Similar difficulties with relying on speed control are mentioned by Krozel, et al. [1996] using a much different conflict analysis method based on optimal control theory.

Figure 8-5 shows the observed times in which the alert stages were triggered for the perpendicular crossing case of two aircraft on a direct collision course. This is the same situation described by the SOC curves in Figure 8-4. Alert stage **1** is triggered 12.3 minutes prior to the time of Closest Point of Approach (CPA). Stages **2** and **3** are triggered at approximately 8. And .8 minutes to CPA, respectively. If the conflict is not resolved, ATC is notified to take over authority (at the **AT** stage) at 3.3 minutes to CPA. Finally, TCAS produces a traffic advisory (TA) at approximately 45 seconds and a resolution advisory (RA) at 35 seconds to CPA.

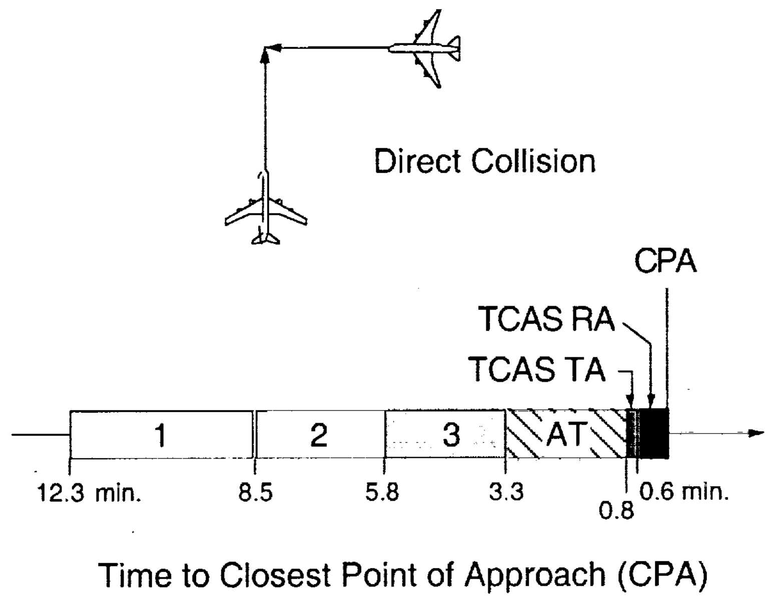


Figure 8-5: Alert Time Line: Direct Collision (90° Crossing Angle)

Figure 8-6 shows a slight different encounter in which two aircraft are not on a direct collision course but will pass within 6 nautical miles of one another. Stage **1** is triggered 6.5 minutes before CPA, and stage **2** is triggered 2.2 minutes before CPA. A TCAS TA is also generated at approximately 30 seconds before CPA. When the traffic passes the host aircraft, the alert stages gradually decrease. Thus, the logic increases the alert stage as the potential for a conflict rises and reduces the alert stage as it becomes less likely that the intruder could turn and cause a conflict.

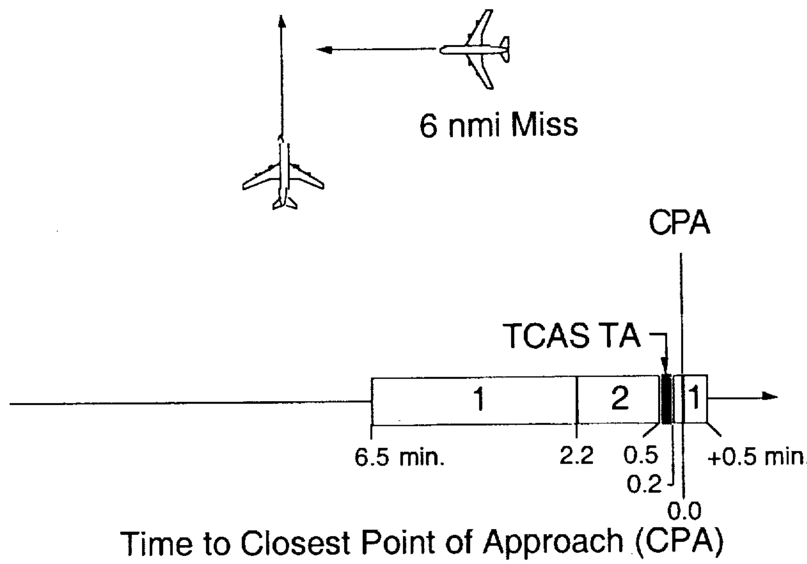


Figure 8-6: Alert Time Line: 6 nmi Minimum Separation (90° Crossing Angle)

8.5 Simulation Studies

At the NASA Ames Research Center, the prototype alerting logic was incorporated in several aircraft-ATC simulator experiments as part of a study on pilot decision-making aids for new Air Traffic Management environments [Johnson et al., 1997; Cashion et al. 1997; Battiste and Johnson, 1998; Cashion and Lozito, 1999; Dunbar et al., 1999; Johnson et al., 1999]. In these studies, enroute conflicts were scripted to examine pilot response and to exercise the alerting logic.

In operation, the alerting logic was used to trigger the four stages of alerts discussed earlier. Additionally, the probability data were used in one study to determine the magnitude of maneuvering required to resolve conflicts at a specified level of confidence. The pilots in the study were given an interactive tool to explore different maneuvering possibilities. These maneuvers were compared against the probability data to determine whether the conflict would be resolved with 95% confidence. The cockpit display then indicated to the pilots whether the proposed maneuver was likely to be successful.

Preliminary results from the NASA studies show that the pilots could successfully resolve conflict without ATC guidance in most cases. AT alert stages were only observed in scenarios where the intruding aircraft was purposely diverted toward the host aircraft at close proximity. However, a more complete analysis is required to more fully evaluate the alerting logic and to determine the potential impact of the airborne conflict resolution in air traffic management. Additional studies are now under way to exercise the logic utilizing the various intent information that might be available to aircraft in the future.

Early test runs from the intent studies showed some interesting results. With intent, aircraft could be allowed to be spaced in relatively close proximity of one other as shown in the example of Figure 8-7. Here the two aircraft are on parallel tracks and indicating an intent to maintain the same heading. Such intent allows the aircraft to be spaced a short distance apart since the alerting logic is expecting the intent to be followed. However, blunders or sudden course changes by one of the aircraft would result in an immediate hazard situation with little time for appropriate action on the part of the other aircraft. The original intent information has suddenly become a detriment to the overall safety of the system. Thus in order to utilize intent effectively and safely, some use of conformance check to maintain the intent is necessary so that $\mathbf{W} = \mathbf{T}$. If the intent is to be changed, it should also be cleared first with the alerting logic so that there is no disparity between $\mathbf{W}$ and $\mathbf{T}$, or a conflict in the new course.

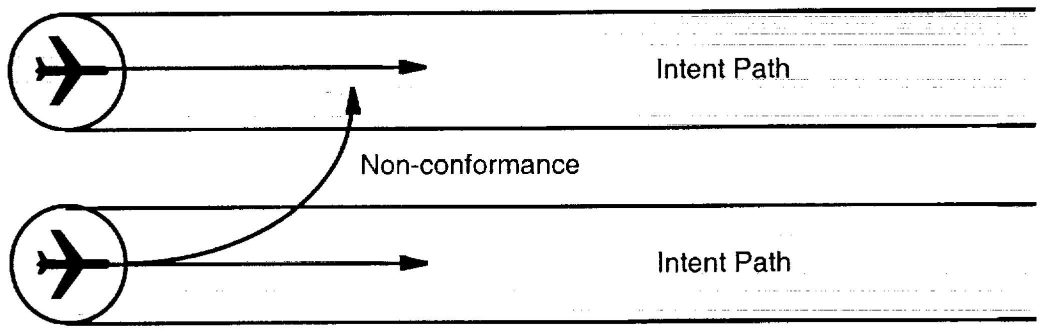


Figure 8-7: Error in Intent Information

8.6 Discussion

The Monte Carlo approach used in the alerting logic has the advantage that it can handle complex, 3-D encounters with complicated error distributions specific to each conflict situation. It uses whatever information is available to describe the conflict and makes a direct prediction of the alerting outcome {i.e. $P(FA)$, $P(SA)$ }based on that information. If a change in the working model (**W**) is required, such as a change in flight plans, the Monte Carlo simulations should reflect the new updates (**W** = **T**).

However, there are some limitations that should be considered when utilizing the Monte Carlo approach presented here. A sufficient amount of processing power is required to perform the vast number of Monte Carlo simulations at any instant in time. The methodology presented in Section 6.2 has allowed a relatively efficient way of computing these probabilities in near real-time (on the order of a quarter of second for each pair of aircraft). As the number of aircraft increases in the vicinity, the longer it will take to compute the desired probabilistic values.

Another consideration involves the scope of the conflict. The resolution maneuvers used to develop the alerting logic are based on the immediate problem of avoiding a conflict and do not consider the additional maneuvering required to return to the original flight path. Thus, the logic does not incorporate issues such as increased fuel burn or flight time in the decision to alert. It may be necessary to incorporate cost-based metrics into the alerting logic as efficiency becomes an increased priority. This might be achieved, for example, by weighting avoidance maneuver options by the additional cost of deviation each option would incur.

Finally, centralized traffic management issues have been ignored. Because, as assumed in the NASA experimental studies, pilots have initial responsibility for traffic separation, ground controllers could have difficulty when suddenly presented with a conflict that was not resolved by the flight crews. Additional conflict detection and resolution aids must be provided for ground controllers to enable them to return to the traffic management loop and handle traffic once they are alerted to the conflict. Alternatively, it may be more appropriate for all conflict detection and resolution activities to be performed on the ground. In either case, the design approach presented in this thesis could be applied in an air, ground, or mixed mode of operation to develop future alerting systems.

8.7 Summary

In this chapter, a conflict alerting probe was developed from the concepts explained in the previous chapters. It is a novel approach based on the utility of near real-time Monte Carlo simulations to predict conflicts and alerting performance. The thresholds are based on the concepts of the SOC methodology first presented by Kuchar [1995, 1996] and entails modeling uncertainties directly into the aircraft trajectory model. Some preliminary results for some of the NASA studies are discussed as well as some the advantages and disadvantages of the approach.

Chapter 9

Summary and Conclusions

9.1 Summary

9.1.1 Review of Alerting Systems and Alerting Performance

A brief overview of the state-space approach to describing alerting systems was given to provide a foundation for further discussion into the problems associated with alerting system design and performance. Terms such as false alarms, missed detections, and correct detections were presented in the realm of state-space. The problem of conflict detection and resolution was formally introduced, and the importance of the tradeoff between the different performance parameters (e.g. false alarms, missed detections) was discussed. A review of System Operating Characteristic (SOC) analysis was also given.

9.1.2 Survey of Alerting Approaches

A survey of different conflict alerting approaches was made to gain insight into the problem of designing a conflict probe to handle complex encounters (3-D, multi-aircraft, intent information, uncertainties). It was found that a large variety of methods existed in literature with no single, apparent underlying theme to drive designs. However, there was a prevalence of an iterative, ad hoc approach using test scenarios to set the threshold parameters. Also, three major trajectory propagation methods for predicting and resolving conflicts were identified: single path, worst case, and probabilistic.

9.1.3 A Unified Approach for Improving Alerting Performance

The alerting problem was recast as a prediction problem in the presence of uncertainties. The performance measures often used to gauge and set threshold parameters rely on the accuracy of the prediction. The importance of trajectory modeling in the prediction process was emphasized and errors in the model were shown to reduce prediction accuracy. A unified approach to improving alerting performance was stated which evolved around increasing prediction accuracy through better trajectory modeling ($\mathbf{W} \rightarrow \mathbf{T}$, $\mathbf{T} \rightarrow \mathbf{W}$) and reducing inherent uncertainties.

9.1.4 Probabilistic Influence in Alerting System Design

The iterative, ad hoc approach to alerting threshold design was revisited from a probabilistic standpoint. It was shown that probabilistic concepts of performance and uncertainties were embedded within this design process of setting threshold metric parameters. Also, a new direct approach to alerting design was presented and shown be a more compact method of estimating performance directly without the unnecessary step of mapping back to a redundant set of threshold metrics. This new approach is based on modeling all known information (including uncertainties and intent information) about the encounter directly into the working trajectory model used by the alerting logic to predict future aircraft positions. The result is a situation-specific design that is tailored to each individual encounter and not compromised from a globally averaged threshold.

9.1.5 Methodology for Computing Conflict Probabilities

A method of computing the probabilistic parameters was presented using a Monte Carlo simulation method. Uncertainties and intent information were modeled into the aircraft trajectories to predict the likelihood of conflicts and avoidance options. A relatively simple idea of utilizing a relative frame coordinate system allowed the use of the Monte Carlo simulations to be performed in near real-time. The accuracy of the computations was discussed and appears to be within the necessary scope for use in a real-time conflict alerting probe. In addition, a method to visualize the conflict situation through the use of probability contours was presented.

9.1.6 Application of Methodology

The methodology developed in this thesis was utilized both as an analytical tool and as the basis for a real-time conflict alerting probe.

9.1.6.1 Conflict Analysis Tool

Through the use of probability contour maps and SOC curves, the benefits of the method was shown to be able to handled both simple and complex types of encounters with relative ease. Examples were presented using intent information, various types of uncertainties, and different resolution options. In one example, the merits of utilizing horizontal maneuvering proved to be more effective than a climb maneuver in a certain type of vertical encounter situation.

9.1.6.2 Real-Time Conflict Probe

Several versions of a real-time conflict alerting probe based on probabilistic thresholds were developed for use in NASA Ames Research Center simulator facilities. The latest version employs direct Monte Carlo simulations run in real-time and can include information on certain types of intent if available.

9.2 Conclusions

The contributions from this thesis work are as follows:

1. The importance of uncertainties in the alerting design process was identified. Much of the work discussed in this thesis evolved from the notion that uncertainties must be dealt with at some point within the alerting system. Without uncertainties in a conflict situation, a perfect alerting threshold could be designed. It is because of uncertainties and errors in prediction that make the conflict detection and resolution problem difficult.

2. A unifying concept for improving alerting system performance was developed based on insight from recasting the problem as decision-making in the presence of uncertainties. The foundation is based on the realization that performance metrics such as false alarm rate and missed detections are measures of prediction accuracy. Thus to improve performance requires increasing prediction accuracy since the performance metrics are based on obtaining a correct prediction of a future hazard or the ability to avoid it. The result led to the concept of reducing modeling errors (**W** = **T**) and reducing inherent uncertainties (make future outcomes more deterministic). Common methods used to improve alerting performance (e.g. delaying alerts, including intent information, enforcing flight restrictions, conformance monitoring) were shown to belong in one of these two categories.

3. A probabilistic connection to alerting design was shown to exist in the common ad hoc approach of setting threshold parameters. Uncertainties in the aircraft trajectories are injected into the design by use of the test simulation scenarios. The outcome of the simulations are probabilistic measures of performance which end up driving the final threshold settings. The result is that the thresholds are really a simplified

mapping to probabilistic performance measures such as $P(FA)$ and $P(SA)$. The iterative feedback of the ad hoc approach explains the ability to utilize various combinations of metric parameters by different alerting logics.

4. Global versus situation-specific designs were identified. Using statistical computations, it was shown how global designs were a compromise between more situation-specific designs. The analysis provided an explanation to past problems encountered by current operational systems such as GPWS and TCAS. It is also the rationale behind tailoring thresholds to individual encounter situations especially as the complexity of the operating environment increases (multi-aircraft, 3-D encounters).

5. A new direct approach was extended from previous work by Kuchar [1995, 1996] where uncertainties and intent are modeled explicitly in the aircraft trajectory model of the alerting logic. The method allows for a direct prediction of the performance measures and is inherently situation-specific to each encounter scenario. Thresholds can then be designed in the state-space of the performance measures {e.g. $P(FA)$ and $P(SA)$} with the help of SOC plots.

6. A methodology was developed to compute the probabilistic values in near real-time setting up the possibility for rapid analysis and also for conflict probing.

7. Example problems were shown utilizing probability contour plots and SOC analysis. These examples showed the usefulness and capability of the method to handle both simple and complex encounter situations with relative ease. For instance, the benefit of intent information was shown in one example and the benefit of horizontal avoidance maneuvering in another.

8. The actual implementation of a prototype conflict alerting probe for flight simulation studies was developed for experimental use at MIT and NASA Ames Research Center facilities. The logic is based on near real-time Monte Carlo predictions of conflict and conflict avoidance. The logic can handle complex multi-aircraft, multi-intent, 3-D encounters.

References

Andrews, J. W. (1978). "A Relative Motion Analysis of Horizontal Collision Avoidance." *SAFE Journal*. Volume 8. No. 2.

Andrews, J. W. and Hollister, W. M. (1997). "Impact of Intent-based Maneuver Constraints on Alert Rates." Paper to be published in the *Air Traffic Control Quarterly*.

Bakker, G. J. and Blom, H. A. P. (1993). "Air Traffic Collision Risk Modeling." Proceedings of the 32nd IEEE Conference on Decision and Control. Vol. 2. San Antonio, TX. December, 1993.

Bateman, D. (1994). "Ground Proximity Warning Systems (GPWS) - Success and Further Progress." The International Civil and Military Avionics Conference. London. April 7, 1994.

Bateman, D. (1999). "The Introduction of Enhanced Ground-Proximity Warning Systems (EGPWS) into Civil Aviation Operations Around the World." Proceedings of the 11th Annual European Aviation Safety Seminar (EASS '99). Amsterdam, Netherlands. March 8-10, 1999.

Battiste, V. and Johnson, W. (1998). "Development of a Cockpit Situation Display for Free-Flight." AIAA and SAE, 1998 World Aviation Conference. Anaheim, CA. September 28-30, 1998.

Bilimoria, K. D., Sridhar, B., and Chatterji, G. B. (1996). "Effects of Conflict Resolution Maneuvers and Traffic Density of Free Flight." AIAA Guidance, Navigation, and Control Conference. San Diego, CA.

Boeing Commercial Airplane Group. (1998). "Statistical Summary of Commercial Jet Airplane Accidents." Presentation given by the Airplane Safety Engineering Division. Seattle, WA. June, 1998.

Bradley, S. (1992). Simulation Test and Evaluation of TCAS II Logic Version 6.04. U.S. Department of Transportation/Federal Aviation Administration Document DOT/FAA/RD-92/23.

Brudnicki, D. J., Lindsay, K. S., and McFarland, A. L. (1997). "Assessment of Field Trials, Algorithmic Performance, and Benefits of the User Request Evaluation Tool (URET) Conflict Probe." 16th AIAA/IEEE Digital Avionics Systems Conference Proceedings. pp. 9.3-35 - 9.3-44.

Brudnicki, D. J. and McFarland, A. L. (1997). "User Request Evaluation Tool (URET) Conflict Probe Performance and Benefits Assessment." Mitre Product MP 97W0000112. June, 1997.

Brudnicki, D., Lindsay, K., and McFarland, A. (1997). "Assessment of Field Trials, Algorithmic Performance, and Benefits of the User Request Evaluation Tool (URET) Conflict Probe." 16th Digital Avionics Systems Conference. Irvine, CA. October 26-30, 1997. pp. 9.3-35 - 9.3-44.

Burdun, I. and Parfentyev, O. (1999). "AI Knowledge Model for Self-Organizing Conflict Prevention / Resolution in Close Free-Flight Air Space." 1999 IEEE Aerospace Conference. Snowmass, CO. March 6-13, 1999. pp. 409-428.

Burgess, D. W., Altman, S. I., and Wood, M. L. (1994). "TCAS: Maneuvering Aircraft in the Horizontal Plane." *Lincoln Laboratory Journal*. Vol. 7. No. 2.

Carpenter, B. D. (1996). "A Probability-Based Alerting Logic for Aircraft on Parallel Approach." S. M. Thesis. Massachusetts Institute of Technology.

Carpenter, B. and Kuchar, J. (1997). "Probability-Based Collision Alerting Logic for Closely-Spaced Parallel Approach." Paper AIAA-97-0222. 35th AIAA Aerospace Sciences Meeting and Exhibit. Reno, NV. January 6-19, 1997.

Cashion, P., Macintosh, M., McGann, A., and Lozito, S. (1997). "A Study of Commercial Flight Crew Self-Separation." 16th AIAA/IEEE Digital Avionics Systems Conference. Irvine, CA. October 26-30, 1997.

Cashion, P. and Lozito, S. (1999). "Effects of Different Levels of Intent Information on Pilot Self Separation Performance." 10th International Symposium on Aviation Psychology. Columbus, OH. May 3-6, 1999.

Chakravarthy, A. and Ghose, D. (1998). "Obstacle Avoidance in a Dynamic Environment – A Collision Cone Approach." *IEEE Transactions on Systems, Man, and Cybernetics*. Vol. 28. No. 5. pp. 562-574.

Coenen, F. P., Smeaton, G. P., and Bole, A. G. (1989). "Knowledge-Based Collision Avoidance." *Journal of Navigation*. Vol. 42. No. 1.

DeCelles, J. L. (1991). "The Delayed GPWS Response Syndrome." Aviation Research and Education Foundation. Herndon, VA. July, 1991.

Drake, A. W. (1967). *Fundamentals of Applied Probability Theory*. McGraw-Hill, Inc. New York. Reissued 1988. p. 141.

Drumm, A. C. (1996). *Lincoln Laboratory Evaluation of TCAS II Logic Version 6.04a*. Volume I. Lincoln Laboratory, MIT. Lexington, MA.

Dunbar, M., Cashion, P., McGann, A., Macintosh, M., Dulchinos, V., Jara, D., Jordan, K., and Lozito, S. (1999). "Air-Ground Integration Issues in a Self-Separation Environment." 10th International Symposium on Aviation Psychology. Columbus, OH. May 3-6, 1999.

Duong, V. N. and Hoffman, E. G. (1997). "Conflict Resolution Advisory Service in Autonomous Aircraft Operations." 16th Digitial Avionics Systems Conference. Irvine, CA. October 26-30, 1997.

Durand, N., Alliot, J., and Chansou, O. (1995). "Optimal Resolution of En Route Conflicts." *Air Traffic Control Quarterly*. Vol. 3. No. 3. pp. 139-161.

Eby, M. S. (1994). "A Self-Organizational Approach for Resolving Air Traffic Conflicts." *Lincoln Laboratory Journal*. Vol. 7. No. 2.

Eby, M. and Kelly, W. (1999). "Free Flight Separation Assurance Using Distributed Algorithms." 1999 IEEE Aerospace Conference. Snowmass, CO. March 6-13, 1999. pp. 429-441.

Federal Aviation Administration (FAA). (1991). "Precision Runway Monitor Demonstration Report." Document DOT/FAA/RD-91/5. February, 1991.

Ford, R. L. (1986). "The Protected Volume of Airspace Generated by an Airborne Collision Avoidance System." *Journal of Navigation*. Vol. 39. No. 2. pp. 139-158

Ford, R. L. (1987). "The Conflict Resolution Process for TCAS II and Some Simulation Results." *Journal of Navigation*. Vol. 40. No. 3. pp. 283-303.

Ford, R. L. and Powell, D. L. (1990). "A New Threat Detection Criterion for Airborne Collision Avoidance Systems." *Journal of Navigation*. Vol. 43. No. 3. pp. 391-403.

Gazit, R. (1996). "Aircraft Surveillance and Collision Avoidance Using GPS." PhD Thesis. Stanford University. Stanford, CA. August, 1996.

Harper, K., Mulgund, S., Guarino, S., Mehta, A., and Zacharias, G. (1999). "Air Traffic Controller Agent Model for Free Flight." AIAA-99-3987. AIAA Guidance, Navigation, and Control Conference. Portland, OR. August 9-11, 1999. pp. 288-301.

Haissig, C. M., Corwin, B., and Jackson, M. (1999). "Designing an Airborne Alerting System for Closely Spaced Parallel Approaches." AIAA 99-3986. pp. 280-287.

Havel, K. and Husarcik, J. (1989). "A Theory of the Tactical Conflict Prediction of a Pair of Aircraft." *Journal of Navigation*. Vol. 42. No. 3.

Heuvelink, G. (1988). "An Alternative Method to Solve a Variational Inequality Applied to an Air Traffic Control Example." Analysis and Optimization of Systems: 8th International Conference.

Hoekstra, J., van Gent, R., and Ruigrok, R. (1998). "Conceptual Design of Free Flight with Airborne Separation Assurance." AIAA-98-4239. AIAA Guidance, Navigation, and Control Conference. Boston, MA. August 10-12, 1998. pp. 807-817.

Iijima, Y., Hagiwara, H., and Kasai, H. (1991). "Results of Collision Avoidance Maneuver Experiments Using a Knowledge-Based Autonomous Piloting System." *Journal of Navigation*. Vol. 44. No. 2.

Innocenti, M., Patrizio, G., and Pollini, L. (1999). "Air Traffic Management Using Probability Function Fields." AIAA-99-4149. pp. 1088-1097.

Irvine, R. (1998). "GEARS Conflict Resolution Algorithm." AIAA-98-4236. AIAA Guidance, Navigation, and Control Conference. Boston, MA, August 10-12, 1998. pp. 786-796.

Isaacson, D. R. and Erzberger, H. (1997). "Design of a Conflict Detection Algorithm for the Center/TRACON Automation System." 16th AIAA/IEEE Digital Avionics Systems Conference Proceedings. Irvine, CA. October 26-30. pp. 9.3-1 - 9.3-9.

Johnson, R. A. (1994). *Miller and Freund's Probability and Statistics for Engineers*. 5th ed. Prentice-Hall International, Inc. London. pp. 279-285.

Johnson, W., Battiste, V., Delzell, S., Holland, S., Belcher, S., and Jordan, K. (1997). "Development and Demonstration of a Prototype Free Flight Cockpit Display of Traffic Information." AIAA and SAE, 1997 World Aviation Congress. Anaheim, CA. October 13-16, 1997.

Johnson, W., Battiste, V., and Bochow, S. (1999). "A Cockpit Display Designed to Enable Limited Flight Deck Separation Responsibility." AIAA and SAE, 1999 World Aviation Conference. San Francisco, CA. October 19-21, 1999.

Kelly, W. E. (1999). "Conflict Detection and Alerting for Separation Assurance Systems." 18th Digital Avionics Systems Conference. St. Louis, MO. October, 27-29, 1999.

Klass, P. J. (1998). "New TCAS Software to Cut Unneeded Evasive Actions." *Aviation Week and Space Technology*. Vol. 146. No. 4. pp. 57-59.

Kosco, S. (1996). "Coordinated Parallel Runway Approaches." NASA Contractor Report under Contract NAS1-19704 - Task 11.

Kosecka, J. Tomlin, C., Papas, G., and Sastry, S. (1997). "Generation of Conflict Resolution Maneuvers for Air Traffic Management." International Conference on Robotics and Intelligent Systems. September, 1997.

Krauthammer, C. (1996). "Deep Blue Funk." TIME Mazazine. Vol. 147. No. 9. February 26, 1996.

Krozel, J., Mueller, T., and Hunter, G. (1996). "Free Flight Conflict Detection and Resolution Analysis." AIAA Guidance, Navigation, and Control Conference. San Diego, CA.

Krozel, J. and Peters, M. (1997). "Conflict Detection and Resolution for Free Flight." *Air Traffic Control Quarterly*. Vol. 5. No. 3. pp. 181-212.

Krozel, J., Peters, M.., and Hunter, G. (1997). "Conflict Detection and Resolution for Future Air Transportation Management." Final Report Prepared for NASA Ames Research Center. NASA CR-97-205944.

Kuchar, J. K. (1995). "A Unified Methodology for the Evaluation of Hazard Alerting Systems." Ph.D. Thesis. Massachusetts Institute of Technology.

Kuchar, J. K. (1996). "Methodology for Alerting-System Performance Evaluation." AIAA *Journal of Guidance, Navigation, and Dynamics*. Vol. 19. No. 2. March-April, 1996.

Kuchar, J. K. and Yang, L. C. (1997). "Survey of Conflict Detection and Resolution Modeling Methods." AIAA Guidance, Navigation, and Control Conference. New Orleans, LA. August 11-13, 1997.

Lachner, R. (1997). "Collision Avoidance as a Differential Game – Real-Time Approximation of Optimal Strategies Using Higher Derivatives of the Value Function." IEEE International Conference on Systems, Man, and Cybernetics. Orlando, FL. October 12-15. pp.2308-2313.

Love, D. (1988). "TCAS III: Bringing Operational Compatibility to Airborne Collision Avoidance." AIAA/IEEE 8th Digital Avionics Systems Conference.

McLaughlin, M. P. and Zeitlin, A. D. (1992). "Safety Study of TCAS II for Logic Version 6.04." Department of Transportation Report. DOT/FAA/RD-92/22. July, 1992.

McNally, B., Bach, R., and Chan, W. (1998). "Field Test Evaluation of the CTAS Conflict Prediction and Trial Planning Capability." AIAA-98-4480. AIAA Guidance, Navigation, and Control Conference. Boston, MA, August 10-12, 1998. pp. 1686-1697.

Mellone, V. J. and Frank, S. M. (1993). "Behavioral Impact of TCAS II on the National Air Traffic Control System." Seventh International Sysmposium of Aviation Psychology. The Ohio State University. April 27, 1993.

Miller, C. A., Williamson, T., Walsh, J. A., Nivert, L. J., Anderson, J. L. (1994). "Initiatives to Improve TCAS-ATC Compatibility." *Journal of ATC*. July-September.

NASA Conference Publication 10191. (1996). "Proceedings of the NASA Workshop on Flight Deck Centered Parallel Runway Approaches in Instrument Meteorological Conditions." Workshop Sponsored by NASA Langley Research Center. Hampton, Virginia.

Neelamkavil, F. (1987). *Computer Simulation and Modelling*. John Wiley & Sons. p. 30.

Nordwall, B. D. (1997). "New TCAS-2 Software Simplifies Operation." Aviation Week and Space Technology. Vol. 146. No. 10. March 10, 1997. p. 62.

Ota,T., Nagati, M., and Lee, D-C. (1998). "Aircraft Collision Avoidance Trajectory Generation." AIAA-98-4241. AIAA Guidance, Navigation, and Control Conference. Boston, MA, August 10-12, 1998. pp. 828-837.

Paielli, R. A. and Erzberger, H. (1997). "Conflict Probability Estimation for Free Flight." *Journal of Guidance, Control, and Dynamics.* Vol. 20. No. 3. May-June, 1997. pp. 588-596.

Paielli, R. A. and Erzberger, H. (1999). "Conflict Probability Estimation Generalized to Non-Level Flight." Air Traffic Control Quarterly. Vol. 7. No. 3. pp. 195-222.

Phillips, E. H. (1996). "Free Flight Poses Multiple Challenges." Aviation Week & Space Technology. Vol. 144. No. 13. March 25, 1996. p. 27.

Prandini, M., Lygeros, J., Nilim, A., and Sastry, S. (1999). "A Probabilistic Framework for Aircraft Conflict Detection." AIAA-99-4144. AIAA Guidance, Navigation, and Control Conference. Portland, OR. August 9-11, 1999. pp. 1047-1057.

Pritchett, A. R. (1996). "Pilot Non-Conformance to Alerting System Commands During Closely Spaced Parallel Approaches." Sc.D. Thesis. Department of Aeronautics and Astronautics. Massachusetts Institute of Technology. Cambridge, MA. December, 1996.

Pritchett, A. R. (1999). "Pilot Performance at Collision Avoidance During Closely Spaced Parallel Approaches." *Air Traffic Control Quarterly.* Vol. 7. No. 1.

Radio Technical Committee on Aeronautics (RTCA). (1976). "Minimum Performance Standards - Airborne Ground Proximity Warning Equipment." Document No. RTCA/DO-161A. Washington, D. C. May 27, 1976.

Radio Technical Committee on Aeronautics (RTCA). (1983). "Minimum Performance Specifications for TCAS Airborne Equipment." Document No. RTCA/DO-185. Washington, D. C. September, 1983.

Radio Technical Committee on Aeronautics (RTCA). (1995). "Final Report of RTCA Task Force 3: Free Flight Implementation." Washington, DC. October, 1995.

Ratcliffe, S. (1989). "Automatic Conflict Detection Logic for Future Air Traffic Control." *Journal of Navigation*. Vol. 42. No. 3.

Reitman, J. (1971). *Computer Simulation Applications.* Wiley-Interscience, a Division of John Wiley & Sons, Inc. New York. p. 39.

Rome, J. H. and Kalafus, R. (1988). "Impact of Automatic Dependent Surveillance and Navigation System Accuracy on Collision Risk on Intersecting Tracks." Proceedings from the National Technical Meeting of the Institute of Navigation. pp. 213-222.

Ryan, P. A. and Brodegard, W. C. (1997). "New Collision Avoidance Device is Based on Simple and Passive Design to Keep the Cost Low." *ICAO Journal*. Vol. 52. No. 4. May, 1997.

Schild, R. (1998). "Rule Optimization for Airborne Aircraft Separation." PhD Thesis. Technical University of Vienna, Austria. November 26, 1998.

Shank, E. M. and Hollister, K. M. (1994). "Precision Runway Monitor." The Lincoln Laboratory Journal. Vol. 7, No. 2. pp. 329-353.

Shepard, T, Dean, T., Powley, W, and Akl, Y. (1991). "A Conflict Prediction Algorithm Using Intent Information." 36th Annual Air Traffic Control Association Conference Proceedings, Fall, 1991.

Shewchun, M. and Feron, E. (1997). "Linear Matrix Inequalities for Analysis of Free Flight Conflict Problems." IEEE Conference on Decision and Control. San Diego, CA. December, 1997. pp. 2417-2422

Sridhar, B. and Chatterji, G. B. (1997). "Computationally Efficient Conflict Detection Methods for Air Traffic Management." Proceedings of the American Control Conference. Albuquerque, NM. June 3-5, 1997.

Taylor, D. H. (1990). "Uncertainty in Collision Avoidance Maneuvering." *Journal of Navigation*. Vol. 43. No. 2.

Thomas, G. B, Jr. and Finney, R. L. (1984). *Calculus and Analytic Geometry*. 6th ed. Addison-Wesley Publishing Company. Reading, MA. pp. 732-734.

Tomlin, C., Pappas, G. J., and Sastry, S. (1998). "Conflict Resolution for Air Traffic Management: A Study in Multi-Agent Hybrid Systems." *IEEE Transactions on Automatic Control*, Vol. 43. No. 4. April, 1998. pp. 509–521.

Vink, A., Kauppinen, S., Beers, J., and de Jong, K. (1997). "Medium Term Conflict Detection in EATCHIP Phase III." 16th AIAA/IEEE Digital Avionics Systems Conference Proceedings. pp. 9.3-45 - 9.3-52.

von Viebahn, H. and Schiefele, J. (1997). "A Method for Detecting and Avoiding Flight Hazards." Proceedings of SPIE Meeting on Enhanced and Synthetic Vision. Bellingham, WA. April 21-22, 1997. pp. 50-56.

Waller, M. and Scanlon, C. eds. (1996). "Proceedings of the NASA Workshop on Flight Deck Centered Parallel Runway Approaches in Instrument Meteorological Conditions." NASA Conference Publication 10191. Hampton, VA. December, 1996.

Wangermann, J. P. and Stengel, R. F. (1994). "Principled Negotiation Between Intelligent Agents: A Model for Air Traffic Management." ICAS-94-8.6.3. ICAS Proceedings. Vol. 3. pp. 2197-2207.

Wanke, C. (1997). "Using Air-Ground Data Link and Operator-Provided Planning Data to Improve ATM Decision Support System Performance." 16th AIAA/IEEE Digital Avionics Systems Conference Proceedings. pp. 9.3-18 - 9.3-26.

Warren, A. (1997) "Medium Term Conflict Detection for Free Routing: Operational Concepts and Requirements Analysis." 16th AIAA/IEEE Digital Avionics Systems Conference Proceedings. pp. 9.3-27 - 9.3-34.

Wiener, E. L. and Nagel, D. C. (1988). *Human Factors in Aviation*. Academic Press, Inc. San Diego, CA. p. 439.

Williams, P. R. (1992). "Aircraft Collision Avoidance using Statistical Decision Theory." Proceedings from Sensors and Sensor System for Guidance and Navigation II. Bellingham, WA. Vol. 1694. pp. 29-34.

Williamson, T. and Spencer, N. A. (1989). "Development and Operation of the Traffic Alert and Collision Avoidance System." Proceedings of the IEEE. Vol. 77. No. 11.

Winder, L. F. and Kuchar, J. K. (1999). "Evaluation of Collision Avoidance Maneuvers for Parallel Approach." *Journal of Guidance, Control, and Dynamics*. Vol. 22, No. 6. pp. 801-807.

Yang, L. C. and Kuchar, J. K. (1997). "Prototype Conflict Alerting Logic for Free Flight." *Journal of Guidance, Control, and Dynamics*. Vol. 20. No. 4. July-August, 1997. pp. 768-773.

Zeghal, K. (1994). "Towards the Logic of an Airborne Collision Avoidance System with Ensures Coordination with Multiple Cooperative Intruders." ICAS Proceedings. Vol. 3. pp. 2208-2218.

Zeghal, K. (1998). "A Review of Different Approaches Based on Force Fields for Airborne Conflict Resolution." AIAA-98-4240. AIAA Guidance, Navigation, and Control Conference. Boston, MA, August 10-12, 1998. pp. 818-827.

Zeghal, K, and Hoffman, E. (1999). "Design of Cockpit Displays for Limited Delegation of Separation Assurance." 18th Digital Avionics Systems Conference. St. Louis, MO. October 27-29, 1999.

Appendix A

A Review of Conflict Detection and Resolution Modeling Methods

A number of methods have been proposed to automate air traffic conflict detection and resolution, but there have been little cohesive discussion or comparative evaluation of differing approaches. This appendix presents a survey of 62 recent methods, several of which are currently in use or under operational evaluation. This is by no means an exhaustive list, but it is believed to encompass a majority of the recent approaches to the problem. The taxonomy includes: method of dynamic state propagation, dimensions of state information, conflict detection threshold, conflict resolution method, maneuvering dimensions, and management of multiple aircraft conflicts.

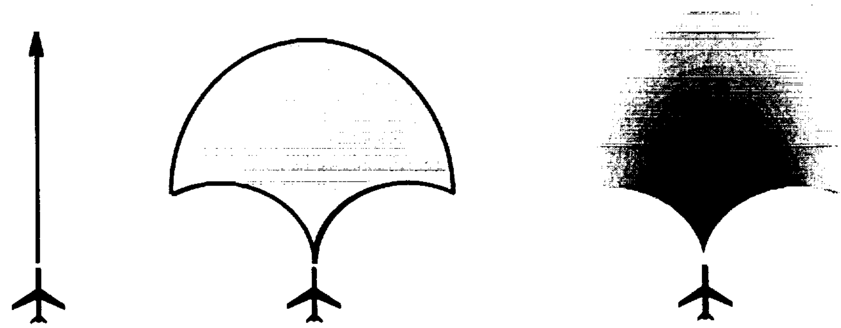
Nine of the models that were examined are existing operational systems in use or which have been evaluated in the field: Airborne Information for Lateral Spacing (AILS) [Waller and Scanlon, 1996], Center / TRACON Automation System (CTAS) [Isaacson and Erzberger, 1997], Ground Proximity Warning System (GPWS) [RTCA, 1976] and the recent Enhanced version (EGPWS) [Bateman 1999], Precision Runway Monitor (PRM) [FAA, 1991], Traffic Alert and Collision Avoidance System (TCAS) [RTCA, 1983], Traffic and Collision Alert Device (TCAD) [Ryan and Brodegard, 1997], User Request Evaluation Tool (URET) [Brudnicki et al., 1997], and a prototype conflict detection system for the Cargo Airline Association [Kelly, 1999]. The remaining approaches range from abstract concepts to prototype conflict warning systems being evaluated or used in laboratories. Five of the models were developed for robotic, automobile, or naval applications [Coenen et al., 1989; Iijima et al., 1991; Taylor, 1990; Chakravarthy and Ghose, 1998; Lachner, 1997], but are still applicable to aviation.

To provide a consistent basis upon which to describe the approaches, each one is classified by the manner in which it is explicitly described in its reference. An approach defined here to address only horizontal conflicts, for instance, could potentially be extended to work in 3-D (and the need for such an extension may be mentioned in the reference), but such an addition was not specifically described in the reference. As another example, if a model computes aircraft missed distance but does not define an explicit conflict detection threshold, the model is not classified as providing conflict detection even though the model could be adapted to perform such a task.

A.1 State Propagation

Because conflict detection and resolution can only be as reliable as the ability of the model to predict the future, the most important difference between modeling approaches involves the method by which the current states are projected into the future Three fundamental extrapolation methods have been identified: single path, worst case, and probabilistic.

In the single path approach, the current states are projected into the future along a single trajectory without direct consideration of uncertainties. An example would be extrapolating the aircraft's position based on its current velocity vector (Figure A-1a). The single path projection method is straightforward and provides a best estimate of where the aircraft will be based on the current state information. In situations where aircraft trajectories are very predictable (such as when projecting only a few seconds into the future), a single path model may be quite accurate. Single path projections, however, do not directly account for the possibility that an aircraft may not behave as expected – a factor that is especially important in longer term conflict prediction. Generally, this uncertainty is managed by introducing a safety buffer, minimum miss distance, or time to closest point of approach threshold at which point a conflict will be detected.



Single Path
(a)

Worst case
(b)

Probabilistic
(c)

Figure A-1: Propagation Methods

The other extreme of dynamic modeling is to examine a worst case projection. Here, it is assumed that an aircraft will perform any of a range of maneuvers. If any one of these maneuvers could result in a loss of minimum separation, then a conflict is declared. The result is a swath of potential trajectories which is monitored to detect conflicts with other aircraft (Figure A-1b). Worst case approaches are conservative in that they can trigger alerts whenever there is any possibility of a conflict within the definition of the worst case trajectory model. If such conflict inducing maneuvers are unlikely, protecting against them may severely reduce overall traffic capacity due to a high false alert rate. Accordingly, the worst case approach may be appropriate when it is desirable to determine if a conflict is possible, or for air traffic concepts in which aircraft are procedurally constrained to remain within a given maneuvering corridor. Each corridor then becomes the boundary of the worst-case aircraft trajectories, and conflicts can be predicted based simply on whether corridors intersect at the same point in time.

In the probabilistic method, uncertainties in the model are used to develop a set of possible future trajectories, each weighted by its probability of occurrence. Fore example, a distribution of future aircraft positions could be obtained by modeling uncertainty in along-track and cross-track guidance (Figure A-1c). A probabilistic approach provides an opportunity for a balance between relying too heavily on a aircraft adhering to a single trajectory versus relying too heavily that an aircraft performs worst case maneuvers. The advantage of a probabilistic approach is that decisions can be made on the fundamental likelihood of a conflict – safety and false alarm rates can be assessed and considered directly. The probabilistic method is also the most general – the single path and worst case models can be considered subsets of probabilistic trajectories. The single path trajectory corresponds to a case in which the aircraft will follow a given (e.g. maximum likelihood) trajectory with probability 1.0; the worst case model is one in which the aircraft will follow any trajectory with equal likelihood. However, the logic behind a probability-based system may be difficult to convey to operators, possibly reducing confidence in their usage [Pritchett, 1996]. There may also be difficulties in modeling the probabilities of the future trajectories with which aircraft may follow.

Tables A-1, A-2, and A-3 provide an organized listing of the 62 approach methods. To conserve space, only the first author is listed in cases where multiple authors are listed on a publication. The three tables are segregated by the propagation method taken: single path, worst case, and probabilistic. Five columns are used to organize the models: State Dimensions, Conflict Detection, Conflict Resolution, Resolution Maneuvers, and Multiple Conflicts.

A.2 State Dimensions

The *Dimensions* column shows whether the state information used in the approaches involves purely horizontal plan (H), vertical plane (V), or both (HV). The majority of approaches cover either 3-D or the horizontal plane; only GPWS focuses solely on the vertical plane. Some models may be easily extended to cover additional dimensions than are shown here, but such extension is not explicitly described in the reference.

A.3 Conflict Detection

The *Detection* column indicates (with a check mark) whether a modeling approach explicitly defines when a conflict alert should be issued. Approaches that do indicate an explicit threshold may provide valuable tools and metrics upon which conflict detection decisions can be made, but do not precisely draw the line between predicted conflict and non-conflict. Additionally, models shown to not provide conflict detection may be primarily concerned with the resolution of the conflict rather than the in determining when action should begin. Although developing conflict resolution methods are important, at some point it will be necessary to define conflict detection thresholds and examine the false alarm / missed detection tradeoff. Approaches that are shown to provide conflict detection may use an extremely simple criterion (e.g. current range between aircraft) to determine when a conflict exists or may use a more complex set of threshold logic.

A.4 Conflict Resolution

The *Resolution* column shows the method by which a solution to a conflict is generated. Five categories are included here: Prescribed (P), Optimized (O), Force Field (F), Manual (M), and no resolution (—).

Prescribed resolution maneuvers are fixed during system design based on a set of predefined procedures. For example, GPWS issues a standard "PULL UP" warning when a conflict with terrain exists. GPWS does not perform additional computation to determine an optimal escape maneuver. AILS [Waller and Scanlon, 1996] and Carpenter and Kuchar [1997] assume that a fixed climbing-turn maneuver is always performed to avoid traffic on a parallel approach. Prescribed maneuvers may have the benefit that operators can be trained to perform them reflexively. This may decrease response time when a conflict is issued. However, prescribed maneuvers are, in general, less effective than maneuvers that are computed in real-time since there is no opportunity to modify the resolution maneuver (the maneuver is performed open-loop to some extent). In many conflicts, it will be necessary to adapt the resolution maneuver to account for unexpected events in the environment, or to reduce the aggressiveness of the maneuver should the conflict be resolved more easily than first predicted.

Optimization approaches typically combine a kinematic model with a set of cost metrics. An optimal resolution strategy is then determined by solving for the trajectories with the lowest cost. TCAS, for example, searches through a set of potential climb or descent maneuvers and selects the least-aggressive maneuver that provides adequate protection [RTCA, 1983]. This requires the definition of appropriate cost functions – typically projected separation, or fuel or time, but costs could also cover workload. Developing cost functions may be fairly straightforward for economic values, but difficult when modeling human utilities. Because current interest in this field is generally centered on strategic resolution of conflicts before immediate tactical evasion is required, economic costs and operator workload will be important to the system design.

Some of the models denoted as using optimized conflict resolution apply techniques such as game theory, genetic algorithms, expert systems, or fuzzy control to the problem. Expert systems use rule-based methods to categorize conflicts and decide whether to alert and/or resolve a conflict. These models can be complex and require a large number of rules to completely cover all possible encounter situations. Additionally, it may be difficult to certify that the system will always operate as intended, and the "experts" used to develop or train the system may in fact not use the best strategy in resolving conflicts. However, the rule base, by design, may be easier for a human to understand or explain than an abstract mathematical algorithm.

Force field approaches treat each aircraft as a charged particle and use modified electrostatic equations to generate resolution maneuvers. The repulsive forces between aircraft are used to define the maneuver each performs to avoid a collision. A force field method, while attractive in the sense that a conflict resolution is continuously available using relatively simple equations, may have some pathologies that require additional consideration before they can be used in operation. For example, force field methods may assume that aircraft continuously maneuver in response to the changing force field, or that aircraft can vary their speed over a wide range. This requires a high level of guidance on the flight deck and increases complexity beyond issuing simple heading vectors, for example. Several human-in-the-loop implementations of the force field approach, however, have shown that the method can be effective if properly applied [Duong and Hoffman, 1997; Hoekstra et al., 1998; Zeghal and Hoffman, 1999].

Some models allow the user to generate potential conflict resolution solutions and obtain feedback as to whether the trial solution is acceptable. These models are denoted as handling a Manual solution in the table. The benefit of a manual solution is that it is generally more flexible in the sense that it is based on human intuition (using information that may not be available to the automation). For example, weather information that is not available to the conflict detection and resolution system may be important when considering a conflict resolution maneuver. Automated solutions that do not take relevant environmental information into account will likely produce nuisance solutions that the human finds unacceptable.

A "—" in the *Resolution* column indicates that the model does not provide an explicit output of an avoidance action or feedback on a user-defined trial solution. These models perform conflict detection but are not designed to explicitly consider conflict resolution. In some cases, successful conflict resolution is presumed (the focus of the approach is only on detecting or counting conflicts).

A.5 Resolution Maneuvers

The *Maneuvers* column indicates what dimensions of resolution maneuvers are allowed. Possible maneuver dimensions include Turns (T), Vertical maneuvers (V), and Speed changes (S). The notation TV, for example, means that either turns or vertical maneuvers may be performed (but not both simultaneously). In some cases, combined maneuvers may be commanded or performed, indicated by C(). Thus C(TV), for example, indicates that a simultaneous climbing or descending turn may be performed.

Generally, providing more maneuvering dimensions allows for a more efficient solution to a conflict. However, it does place additional responsibility on the operator in the sense that a more complex maneuver must be controlled and monitored, possibly increasing response time and workload.

A.6 Multiple Conflicts

Finally, the *Multiple* column describes how the model handles more than two traffic conflict simultaneously. This can take two forms: Pairwise (P), in which multiple conflicts are addressed sequentially in pairs; and Global (G) in which the entire situations is examined simultaneously.

In a realistic traffic environment, it will be necessary that a conflict detection and resolution system be able to manage more than one conflict at a time. In a pairwise approach, if one conflict solution induces a new conflict, the original solution may need to be modified until a conflict-free solution is found. This is the approach taken by TCAS, for example, and is effective but also could potentially fail in certain situations. A global solution, while potentially more complex, may be more robust. For example, consider the situation shown in Figure A-2. On the left, a pairwise solution is shown. The aircraft on the left detects a conflict with a co-altitude threat at a certain preset time before collision, and attempts to climb or descend. Neither solution is acceptable since it results in a conflict with another aircraft. On the right, a global solution considers all three threat aircraft simultaneously and determines that the climb or descent maneuver must begin earlier than the baseline threshold time in order to safely resolve the conflict. At the least, models should be examined in multi-aircraft situations to determine their robustness to this type of problem.

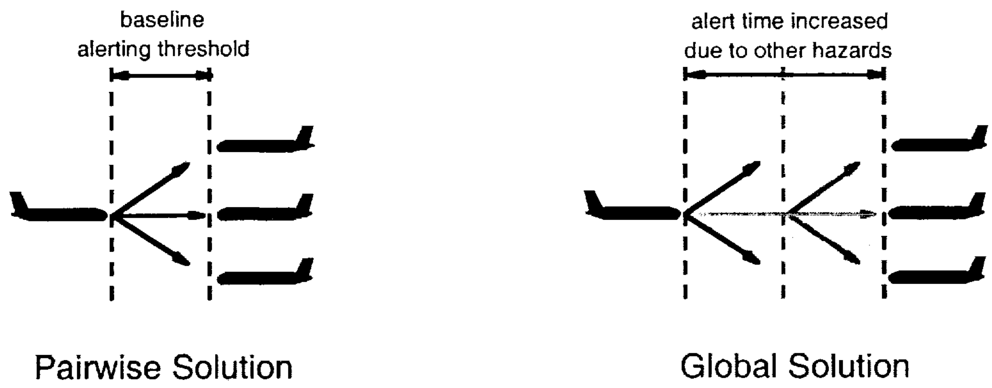


Figure A-2: Pairwise vs. Global Solution

A.7 Other Model Elements

In addition to the six factors used to distinguish between modeling approaches in Tables A-1, A-2, and A-3, there are several other issues to be considered but are not fully described here. These issues include specifically which current states and metrics are used to make conflict detection and resolution decisions, how uncertainty is managed in the model, and the degree to which the model assumes coordination between aircraft involved in a conflict.

Consideration of the states that are used in conflict detection and resolution is important because these states represent the means by which the system observes the environment. Some approaches use a simplified set of states which reduces sensor requirements, but increases the uncertainty in which the conflict detection and resolution decisions can be made. An additional set of data that will be valuable in strategic conflict detection is aircraft intent information such as a programmed flight plan. This information can be used to better model the future trajectory of the aircraft, and thereby be better able to make correct alerting decisions.

The manner in which uncertainties are managed in the design of a conflict detection and resolution system varies widely. Most approaches to the problem combine the uncertainties into a spatial safety buffer to reduce missed detection probability and also incorporate a look-ahead time boundary to limit false alarms. This provides for a reasonable accommodation of uncertainties, but it may not be as effective or accurate as more complete, probabilistic trajectory models.

Coordinate conflict resolution between aircraft has two primary benefits. First, the required magnitude of maneuvering can be reduced when both aircraft maneuver cooperatively as opposed to the case when only one aircraft maneuvers. Second, coordination helps ensure that aircraft do not maneuver in a direction that could prolong or intensify the conflict. However, a system designed assuming that coordination will occur should also be evaluated in cases in which coordination is not carried out as planned. This would provide some measure of the robustness of the system to a datalink failure or pilot error. For example, TCAS was found to perform poorly in situations in which one aircraft did not respond to the recommended advisory [Drumm, 1996]. In fact, it was deemed unproductive to analyze such encounters in depth since it was felt to completely overshadow any other factors contributing to poor performance. However, if such situations do occur in actual operation, the problem must be resolved else it can only lead to devastating consequences.

Table A-1: Single Path Trajectory Propagation

Model	Dimensions	Detection	Resolution	Maneuvers	Multiple
Andrews	H	—	O	T	P
Chakravarthy	H	—	O	C(ST)	P
Tomlin	H	—	O	T	G
Irvine	HV	—	O	C(STV)	P
Ota	HV	—	O	C(TV)	G
Kosecka	H	—	F	C(ST)	G
Zeghal (1998)	H	—	F	C(ST)	G
Eby	HV	—	F	C(STV)	G
Sridhar	H	√	—	—	P
Bateman (EGPWS)	HV	√	—	—	—
Havel	HV	√	—	—	P
Kelly	HV	√	—	—	P
Ryan (TCAD)	HV	√	—	—	P
RTCA (GPWS)	V	√	P	V	—
FAA (PRM)	H	√	P	C(TV)	P
Bilimoria	HV	√	P	STV	P
Burgess	H	√	O	TV	P
Coenen	H	√	O	ST	P
Gazit	H	√	O	VT	P
Harper	H	√	O	C(ST)	G
Iijima	H	√	O	ST	P
Burdun	HV	√	O	C(STV)	P
Durand	HV	√	O	T	G
Ford	HV	√	O	V	P
Krozel	HV	√	O	STV	P
Love	HV	√	O	TV	P
Schild	HV	√	O	C(TV)	P
RTCA (TCAS)	HV	√	O	V	P
Hoekstra	HV	√	F	C(STV)	P
Zeghal (1994)	HV	√	F	C(STV)	G
Duong	HV	√	M / F	C(STV)	P

Table A-2: Worst Case Trajectory Propagation Methods

Model	Dimensions	Detection	Resolution	Maneuvers	Multiple
Lachner	H	—	O	C(ST)	P
Ratcliffe	HV	√	—	—	P
Shepard	HV	√	—	—	P
Shewchun	HV	√	—	—	P
Waller (AILS)	HV	√	P	C(TV)	P
Vink	HV	√	M	C(STV)	P

Table A-3: Probabilistic Trajectory Propagation Methods

Model	Dimensions	Detection	Resolution	Maneuvers	Multiple
Heuvelink	H	—	—	—	P
Paielli	H	—	—	—	P
Taylor	H	—	—	—	P
Bakker	HV	—	—	—	P
Innocenti	H	—	F	C(ST)	G
Rome	H	√	—	—	P
Warren	H	√	—	—	P
Williams	HV	√	—	—	P
Carpenter	H	√	P	C(TV)	P
Prandini	H	√	O	T	P
Krozel	HV	√	O	STV	P
von Viebahn	HV	√	O	TV	P
Isaacson (CTAS)	HV	√	M	C(STV)	P
McNally (CTAS)	HV	√	M	C(STV)	P
Brudnicki (URET)	HV	√	M	C(STV)	P
Yang	HV	√	M	C(STV)	P

Appendix B

Statistical Analysis of Global Distributions

The following will show the computations used to determine the descriptive statistics of combining two probability distributions into one global distribution. The results can then be extended to a combination of more than two distributions.

B.1 Statistics of Combining 2 Distributions

Let $f_1(x)$ and $f_2(x)$ be two separate distributions from which to sample from. The mean and variance of each function will be μ_1, σ_1^2 and μ_2, σ_2^2, respectively. Also, the fraction of times each distribution is sampled will be denoted as a_1 and a_2, and thus the combined probability density function, $f(x)$, will have a distribution of:

$$f(x) = a_1 f_1(x) + a_2 f_2(x) \tag{B.1}$$

$$a_1 + a_2 = 1 \tag{B.2}$$

The expected value or mean, μ_G, of this global distribution can be computed as follows:

$$\begin{aligned} \mu_G &= E[x] \\ &= \int_{-\infty}^{\infty} x f(x) dx \\ &= \int_{-\infty}^{\infty} x\left[a_1 f_1(x) + a_2 f_2(x)\right] dx \\ &= a_1 \int_{-\infty}^{\infty} x f_1(x) dx + a_2 \int_{-\infty}^{\infty} x f_2(x) dx \\ &= a_1 \mu_1 + a_2 \mu_2 \end{aligned} \tag{B.3}$$

The mean of the combined, global distribution is just a weighted average of the individual distribution means. Thus $\min(\mu_1, \mu_2) \le \mu_G \le \max(\mu_1, \mu_2)$.

The variance can be derived from the second central moment:

$$
\begin{aligned}
\sigma_G^2 &= E\left[(x - E[x])^2\right] \\
&= \int_{-\infty}^{\infty} (x - E[x])^2 [a_1 f_1(x) + a_2 f_2(x)] dx \\
&= \int_{-\infty}^{\infty} (x^2 - 2x\mu_G + \mu_G^2)[a_1 f_1(x) + a_2 f_2(x)] dx \\
&= a_1 \int_{-\infty}^{\infty} x^2 f_1(x) dx + a_2 \int_{-\infty}^{\infty} x^2 f_2(x) dx - \\
&\quad 2\mu_G a_1 \int_{-\infty}^{\infty} x f_1(x) dx - 2\mu_G a_2 \int_{-\infty}^{\infty} x f_2(x) dx + \\
&\quad \mu_G^2 \int_{-\infty}^{\infty} [a_1 f_1(x) + a_2 f_2(x)] dx
\end{aligned}
\tag{B.4}
$$

$$
\begin{aligned}
&= a_1 E_1[x^2] + a_2 E_2[x^2] - 2\mu_G a_2 E_1[x] - 2\mu_G a_2 E_2[x] + \mu_G^2(1) \\
&= a_1 E_1[x^2] + a_2 E_2[x^2] - 2\mu_G(a_2 E_1[x] + a_2 E_2[x]) + \mu_G^2(1) \\
&= a_1 E_1[x^2] + a_2 E_2[x^2] - 2\mu_G(a_2 \mu_1 + a_2 \mu_2) + \mu_G^2(1) \\
&= a_1 E_1[x^2] + a_2 E_2[x^2] - 2\mu_G(\mu_G) + \mu_G^2(1) \\
&= a_1(\sigma_1^2 + \mu_1^2) + a_2(\sigma_2^2 + \mu_2^2) - \mu_G^2 \\
&= (a_1\sigma_1^2 + a_2\sigma_2^2) + (a_1\mu_1^2 + a_2\mu_2^2) - \mu_G^2
\end{aligned}
$$

The term in the left-most parenthesis is just the weighted average of the individual variances. It can be shown that the other terms in the equation must be greater than or equal to 0 so that the overall result will be larger than the weighted average of the variances. The proof is through the use of the Lagrange multiplier method with the constraint of Equation B.2. The idea is to show that $(a_1\mu_1^2 + a_2\mu_2^2) - \mu_G^2 \ge 0$ by proving the minimum value cannot be negative. Thus, find the minimum of:

$$
A = (a_1\mu_1^2 + a_2\mu_2^2) - \mu_G^2 \tag{B.5}
$$

with the constraint of $a_1 + a_2 = 1$.

$$
\begin{aligned}
L &= A + \lambda(a_1 + a_2 - 1) \\
&= \left(a_1\mu_1^2 + a_2\mu_2^2\right) - \mu_G^2 + \lambda(a_1 + a_2 - 1) \\
&= \left(a_1\mu_1^2 + a_2\mu_2^2\right) - \left(a_1\mu_1 + a_2\mu_2\right)^2 + \lambda(a_1 + a_2 - 1) \\
&= \left(a_1\mu_1^2 + a_2\mu_2^2\right) - \left(a_1^2\mu_1^2 + 2a_1a_2\mu_1\mu_2 + a_2^2\mu_2^2\right) + \lambda(a_1 + a_2 - 1) \\
&= \left(a_1 - a_1^2\right)\mu_1^2 + \left(a_2 - a_2^2\right)\mu_2^2 - 2a_1a_2\mu_1\mu_2 + \lambda(a_1 + a_2 - 1)
\end{aligned}
\tag{B.6}
$$

Now minimize L:

$$\frac{\partial L}{\partial a_1} = 0 = (1 - 2a_1)\mu_1^2 - 2a_2\mu_1\mu_2 + \lambda \tag{B.7}$$

$$\frac{\partial L}{\partial a_2} = 0 = (1 - 2a_2)\mu_2^2 - 2a_1\mu_1\mu_2 + \lambda \tag{B.8}$$

$$\frac{\partial L}{\partial \lambda} = 0 = a_1 + a_2 - 1 \tag{B.9}$$

Solving Equations B.7 – B.9 results in the following:

$$\lambda = \mu_1\mu_2 \tag{B.10}$$

$$a_1 = \frac{1}{2} \tag{B.11}$$

$$a_2 = \frac{1}{2} \tag{B.12}$$

$$
\begin{aligned}
A_{\min} &= \left(a_1 - a_1^2\right)\mu_1^2 + \left(a_2 - a_2^2\right)\mu_2^2 - 2a_1a_2\mu_1\mu_2 \\
&= \left(\tfrac{1}{2} - \left(\tfrac{1}{2}\right)^2\right)\mu_1^2 + \left(\tfrac{1}{2} - \left(\tfrac{1}{2}\right)^2\right)\mu_2^2 - 2\left(\tfrac{1}{2}\right)\left(\tfrac{1}{2}\right)\mu_1\mu_2 \\
&= \left(\tfrac{1}{2} - \tfrac{1}{4}\right)\mu_1^2 + \left(\tfrac{1}{2} - \tfrac{1}{4}\right)\mu_2^2 - \left(\tfrac{1}{2}\right)\mu_1\mu_2 \\
&= \tfrac{1}{4}\left(\mu_1^2 + \mu_2^2 - 2\mu_1\mu_2\right) \\
&= \tfrac{1}{4}\left(\mu_1 - \mu_2\right)^2 \\
&\geq 0
\end{aligned}
\tag{B.13}
$$

The contribution of A to the new, combined variance, σ_G^2, is thus to increase the spread of the distribution due to the separation of the individual distribution means, μ_1 and μ_2.

B.2 Statistics of Combining More Than 2 Distributions

The statistical parameters of combining more than 2 distributions can be obtained in a similar manner as in the previous section. For example, the probability density function, $f(x)$, for sampling from 3 distributions is:

$$f(x) = a_1 f_1(x) + a_2 f_2(x) + a_3 f_3(x) \tag{B.14}$$

$$a_1 + a_2 + a_3 = 1 \tag{B.15}$$

And the mean and variance can be computed as follows:

$$
\begin{aligned}
\mu_G &= E[x] \\
&= \int_{-\infty}^{\infty} x f(x)dx \\
&= \int_{-\infty}^{\infty} x\left[a_1 f_1(x) + a_2 f_2(x) + a_3 f_3(x)\right]dx \\
&= a_1\int_{-\infty}^{\infty} x f_1(x)dx + a_2\int_{-\infty}^{\infty} x f_2(x)dx + a_3\int_{-\infty}^{\infty} x f_3(x)dx \\
&= a_1\mu_1 + a_2\mu_2 + a_3\mu_3
\end{aligned}
\tag{B.16}
$$

$$
\begin{aligned}
\sigma_G^2 &= E\left[(x - E[x])^2\right] \\
&= \int_{-\infty}^{\infty} (x - E[x])^2 \left[a_1 f_1(x) + a_2 f_2(x) + a_3 f_3(x)\right] dx \\
&= \int_{-\infty}^{\infty} \left(x^2 - 2x\mu_G + \mu_G^2\right)\left[a_1 f_1(x) + a_2 f_2(x) + a_3 f_3(x)\right] dx \\
&= a_1 \int_{-\infty}^{\infty} x^2 f_1(x) dx + a_2 \int_{-\infty}^{\infty} x^2 f_2(x) dx + a_3 \int_{-\infty}^{\infty} x^2 f_3(x) dx - \\
&\quad 2\mu_G a_1 \int_{-\infty}^{\infty} x f_1(x) dx - 2\mu_G a_2 \int_{-\infty}^{\infty} x f_2(x) dx - 2\mu_G a_3 \int_{-\infty}^{\infty} x f_3(x) dx + \\
&\quad \mu_G^2 \int_{-\infty}^{\infty} \left[a_1 f_1(x) + a_2 f_2(x) + a_3 f_3(x)\right] dx \\
&= a_1 E_1\left[x^2\right] + a_2 E_2\left[x^2\right] + a_3 E_3\left[x^2\right] - \\
&\quad 2\mu_G a_2 E_1[x] - 2\mu_G a_2 E_2[x] - 2\mu_G a_3 E_3[x] + \mu_G^2(1) \\
&= a_1 E_1\left[x^2\right] + a_2 E_2\left[x^2\right] + a_3 E_3\left[x^2\right] - \\
&\quad 2\mu_G\left(a_2 E_1[x] + a_2 E_2[x] + a_3 E_3[x]\right) + \mu_G^2(1) \\
&= a_1 E_1\left[x^2\right] + a_2 E_2\left[x^2\right] + a_3 E_3\left[x^2\right] - \\
&\quad 2\mu_G\left(a_2 \mu_1 + a_2 \mu_2 + a_3 \mu_3\right) + \mu_G^2(1) \\
&= a_1 E_1\left[x^2\right] + a_2 E_2\left[x^2\right] + a_3 E_3\left[x^2\right] - 2\mu_G\left(\mu_G\right) + \mu_G^2(1) \\
&= a_1\left(\sigma_1^2 + \mu_1^2\right) + a_2\left(\sigma_2^2 + \mu_2^2\right) + a_3\left(\sigma_3^2 + \mu_3^2\right) - \mu_G^2 \\
&= \left(a_1\sigma_1^2 + a_2\sigma_2^2 + a_3\sigma_3^2\right) + \left(a_1\mu_1^2 + a_2\mu_2^2 + a_3\mu_3^2\right) - \mu_G^2
\end{aligned}
\tag{B.17}
$$

Again, the left-most terms in parenthesis are just the weighted average of the individual distribution variances; while the remaining terms are due to the difference in the means of the distributions and increase the overall variance of the combined probability density function.

Appendix C

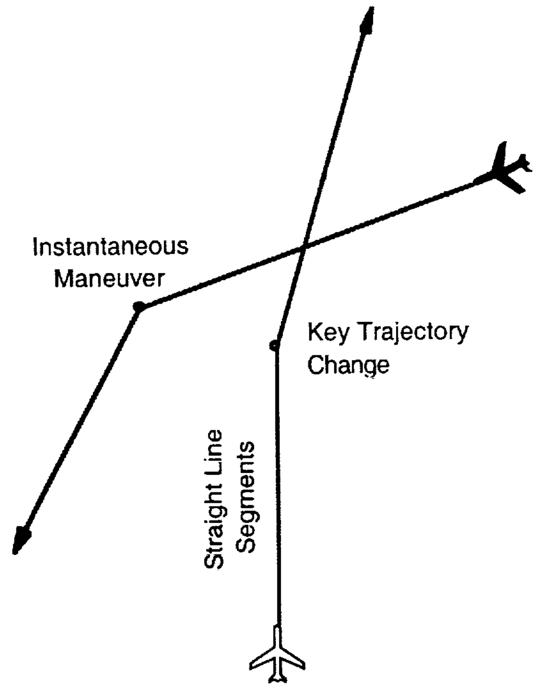
Conflict Detection Using Line-Volume Intersection

The determination of conflict between 2 moving objects is relatively simple if the path of the objects can be approximated as straight line segments. When placed in the relative frame of one of the objects, the solution becomes one of calculating the intersection of a line (relative velocity vector) with a protected volume surrounding the origin. If an intersection occurs, then minimum separation is violated.

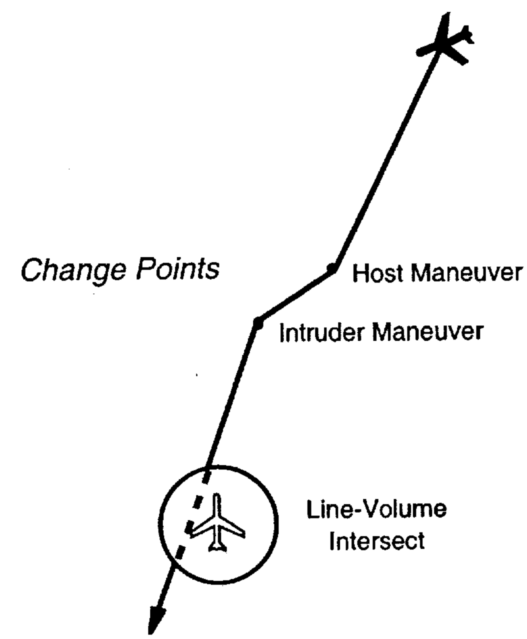
C.1 Relative Frame

Working in the relative frame of one of the aircraft can greatly simplify the computational complexity of the conflict determination problem. For this discussion, the aircraft at the origin will be termed the ownship, and the problem is to determine if the other aircraft (intruder) will pass through the protected volume surrounding the origin position.

Figure C-1a shows the encounter in absolute frame with individual trajectories depicted as straight line segments. A key change in the velocity vector constitutes a new segment. For Figure C-1b, the situation is shown in the relative frame of the ownship aircraft. Each *change point* is thus a change the relative velocity vector between the two aircraft. The task is then to determine if any of the line segments intersect the protected volume at the origin. If there is an intersection, the protected volume will be violated and a conflict is declared.



a) Absolute Frame Encounter



b) Relative Frame Encounter

Figure C-1: Absolute vs. Relative Frame

C.2 Line-Volume Intersection

The solution to solving the intersection between a line (relative velocity vector) and a cylinder (protected volume) can be split up into horizontal and vertical dimensions. The idea is to determine first if there is an intersection in the horizontal plane. If there is, then the line connecting the 2 points of horizontal intersection is cross-checked in the vertical domain to see if it lies within or passes through the cylinder.

For the discussion in this appendix, DMOD will refer to the horizontal radius of the protected zone and ZMOD will denote the minimum vertical separation (see Figure C-2).

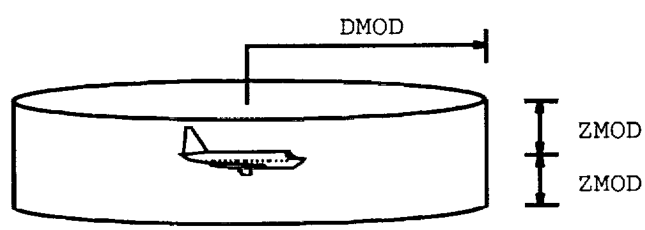


Figure C-2: Cylindrical Protected Zone Parameters

C.2.1 Horizontal Intersection

Let $(d_x, \quad d_y, \quad d_z)$ be the starting position of a line segment which begins at time t_o and ends at t_e. The direction of the line can just the relative velocity vector $\mathbf{v} = \begin{bmatrix} v_x & v_y & v_z \end{bmatrix}^T$. Any other point, p, on the line can be found from:

$$p = \left(d_x + v_x(t - t_o), \quad d_y + v_y(t - t_o), \quad d_z + v_z(t - t_o)\right) \tag{C.1}$$

In the horizontal plane, the time when the line intersects DMOD can be found from the equation of a circle:

$$\left(d_x + v_x(t - t_o)\right)^2 + \left(d_y + v_y(t - t_o)\right)^2 = \mathrm{DMOD}^2 \tag{C.2}$$

$$\left(v_x^2 + v_y^2\right)\left(t - t_o\right)^2 + 2\left(v_x d_x + v_y d_y\right)\left(t - t_o\right) + \left(d_x^2 + d_y^2 - \mathrm{DMOD}^2\right) = 0 \tag{C.3}$$

which has the form of quadratic equation:

$$a\left(t - t_o\right)^2 + b\left(t - t_o\right) + c = 0 \tag{C.4}$$

$$\left(t - t_o\right) = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a} \tag{C.5}$$

Note that is the solution assumes an *infinite* line.

There are 3 possible solutions depending on the value of the radicand, $b^2 - 4ac$. If $b^2 - 4ac < 0$, then no real solution exists and the line will never intersect the circle. If $b^2 - 4ac = 0$, then only one solution exists and the line intersects tangent to the circle of radius DMOD. If $b^2 - 4ac > 0$, then two solutions exist as the line passing through the circle.

C.2.2 Vertical Intersection

C.2.2.1 $b^2 - 4ac < 0$

A conflict does not exist so no there is no need to check the vertical intersection.

C.2.2.2 $b^2 - 4ac = 0$

The line intersects tangent to the circle edge. Thus, if a conflict exists, the intersection point must lie within ZMOD in the vertical dimension. Using the solution from Equation C.5, the position of the intersection point can be found from Equation C.1. The z-component must be checked to see if it lies within ±ZMOD. If it does, then one final check needs to be made to make sure $t_o \le t \le t_e$, If this is satisfied, then a conflict exists along this line segment.

C.2.2.3 $b^2 - 4ac > 0$

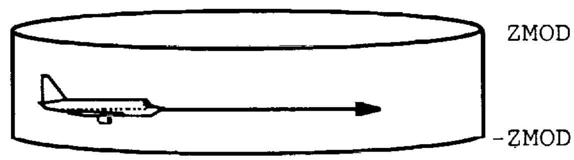
In this case, the line intersects the circle at 2 distinct points, p_1 and p_2, at times, t_1 and t_2, respectively. It will be assumed that $t_1 < t_2$. The following set of C code can be used to determine if the vertical domain intersection is also satisfied:

```
#define AND &&
#define OR  ||
#define MAX(a,b) (((a) > (b)) ? (a) : (b))
#define MIN(a,b) (((a) < (b)) ? (a) : (b))
#define HIT 1
#define MISS 0
#define ZMOD 1000.0      /* (ft) vertical threshold */

short conflict;     /* 0=miss, 1=hit                    */
double t0;          /* (sec) start of line segment      */
double te;          /* (sec) end of line segment te>t0 */
double t1;          /* (sec) 1st pt. of DMOD hit        */
double t2;          /* (sec) 2nd pt. of DMOD hit  t2>t1 */
double vz;          /* (fps) vertical velocity comp.    */
double dz;          /* (ft)  start of line segment      */
double z1;          /* (ft)  1st pt. of DMOD hit or     */
                    /*          start of line segment   */
double z2;          /* (ft)  2nd pt. of DMOD hit or     */
                    /*          end of line segment     */

if ((t2 < 0.0) OR (t < t1))
  conflict = MISS;
else {
  z1 = vz*(MAX(t0, t1) - t0) + dz;
  z2 = vz*(MIN(te, t2) - t0) + dz;

  if ((z1 <= ZMOD) AND (z2 >= -ZMOD))
    conflict = HIT;
  else if ((z1 > ZMOD) AND (z2 <= ZMOD))
```

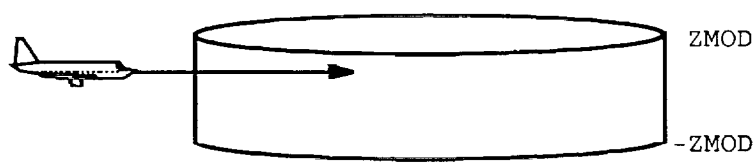
```
    conflict = HIT;
  else if ((z1 < -ZMOD) and (z2 >= -ZMOD))
    conflict = HIT;
  else
    conflict = MISS;
}
```

The first if-statement checks to see if the intersection occurs outside of the two endpoints of the line segment. If it does, then no conflict will exist within the time t_o to t_e.

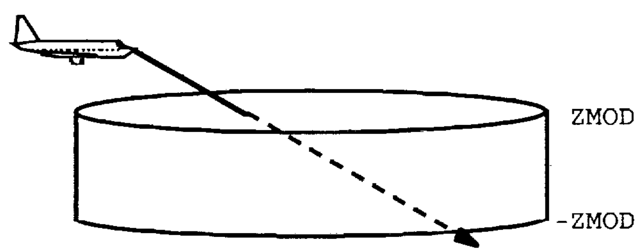
The first *nested* if-statement covers an intruder aircraft that is either currently within the ownship's protected volume (Figure C-3a) or enters it from the side (Figure C-3b). The next if-statement handles an intruder coming from above the protected zone (Figure C-3c), while the last if-statement takes into account the intruder entering from below (Figure C-3d).



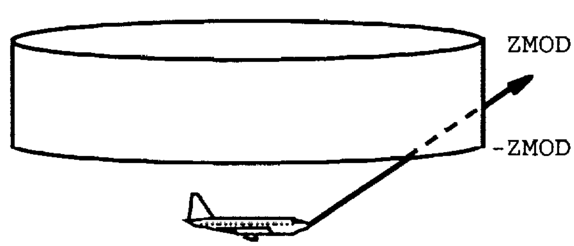
a) Intruder Inside Protected Zone



b) Intruder Entering from Side



c) Intruder Entering from Top



d) Intruder Entering from Bottom

Figure C-3: Vertical Conflict Dimension