6th CNS/ATM Conference, Taipei, Taiwan, March 27-29, 2001
(note: figure quality is reduced from the original in this PDF file)

Managing Uncertainty in Decision-Aiding and Alerting System Design

James K. Kuchar
Department of Aeronautics and Astronautics
Massachusetts Institute of Technology
Cambridge, MA 02139 USA

Abstract

Safety is often enhanced using automated alerting systems and other decision aids to support human decision-making. Examples in aerospace include aircraft system monitors, weather displays, traffic collision alerting, and ground proximity warning systems. Although such automation has been credited with enabling new procedures and preventing a number of accidents, there have also been mishaps induced directly or indirectly by alerting systems. Uncertainties in sensors, dynamics, and human performance reduce the quality of decision support provided by automation. By understanding the relationships between uncertainty, automation design methods, and the resulting system performance, it is possible to target the design of a system to best compensate for uncertainties and thereby provide higher performance. This paper outlines several fundamental design issues relating to decision-making under uncertainty, both from a conceptual standpoint and through quantitative models. The concepts are discussed in the context of applications including enroute traffic conflict detection and collision alerting during closely-spaced parallel approach.

Introduction

During the operation of many processes, threats may be encountered that require attention. Safety or robustness against these threats is often enhanced through the use of automated decision-aiding systems that independently monitor operations and warn the controller to intervene should it be necessary (Figure 1). Decision-aiding and alerting systems are becoming increasingly pervasive, and are used in applications including aerospace vehicles, automobiles, chemical or power control stations, air traffic control, and medical monitoring systems. In addition to providing a final safety net for many processes, some alerting systems also enable operating in regimes that would not possible without them. Closely-spaced parallel approaches at airports in poor visibility, for example, are only allowed when certain automated alerting systems are present to provide the necessary level of safety [1]. The additional traffic throughput, then, is contingent on the existence and performance of an alerting system.

The quality of information available to the alerting system has a direct impact on the quality of decisions that can be made by the system. It is therefore important to understand the relationship between uncertainties in a given problem and the types of alerting solutions that may be viable. To be effective, the alerting system must issue a warning early enough that corrective action can be taken, but not so early that nuisance alarms occur. This

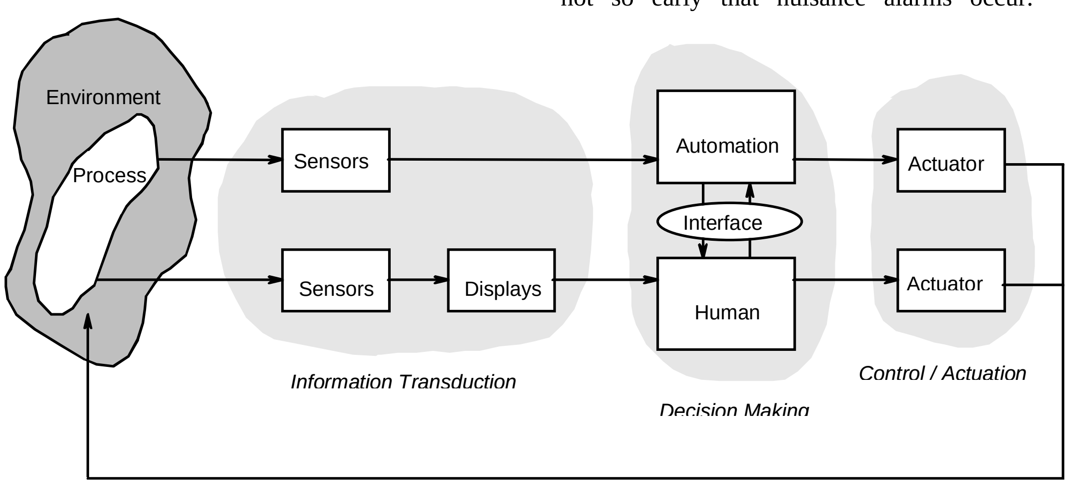


Figure 1: Decision-Aiding System Schematic

generates a tension in the design of these systems that is always present, regardless of the application. Failure to properly balance this tension leads to operator distrust of the system, inefficiencies, or accidents.

Decision-aiding systems have been developed for a number of application problems, and many different techniques have been proposed and investigated. Due to this diversity, it may not be clear what allows one method to achieve higher performance than another. A recent survey of alerting methods for air traffic conflict detection and resolution, for example, found that over 50 different modeling approaches have been proposed, tested, or implemented [2]. The design of these alerting systems may be greatly facilitated through the development of a cohesive modeling and design approach based on formal principles. The appropriate method to use in solving a problem may then be identified from first principles. This paper outlines fundamental considerations for decision-aiding system design, and uses several case studies to illustrate the concepts.

The general context for discussion in this paper are systems such as that shown in Figure 1. These systems all have the same core components including: a controlled process in an environment that includes hazards; sensors and displays that convert the physical state of the process and environment into information elements that can be used by the human and automation; decision-making components which involve a partnership between human and automation at some level, requiring additional displays and interfaces; and control actuation elements that convert decisions into actions in the physical process. The high-level design issue then is how to apply sensors, displays, automation, procedures, and controls to enable operating the process at a desired level of performance. In this paper, the focus is on how the information in the various elements of Figure 1 is molded and used to provide the most effective decision support from automation.

This paper begins by discussing the general modeling approaches that can be applied to decision-support automation design. These methods have been identified and categorized following a survey of a number of different approaches to decision-making in aerospace and other fields. Then, the system design process at large is examined to illustrate how these modeling methods can be applied in different ways to achieve desired system performance.

Modeling Methods

Through an examination of the structure of a number of proposed and implemented alerting systems, several key modeling methods have been identified [2,3]. A description of these methods serves to form a framework from which a given system can be placed in juxtaposition to others, and provides a basis for developing relationships between the type of hazard problem to solve and the modeling method that is most appropriate to use. First, three different overarching philosophies to alerting are described. Although these three philosophies are described in terms of example implementations for a specific aviation problem, they are sufficiently generic to be relevant to a wide range of applications.

Alerting Philosophies

Aircraft parallel approach collision alerting serves as a good case study to illustrate the three fundamental methods for alerting system design. The basic context is one in which it is desired to warn aircraft pilots when there is a collision risk with a nearby aircraft on a parallel approach to an airport. A review of proposed or operational collision avoidance systems shows that fundamentally there are three philosophies that drive the design of decision-making logic: termed here conformance, nominal trajectory, and escape trajectory. These three philosophies are diagrammed in Figure 2 and discussed below.

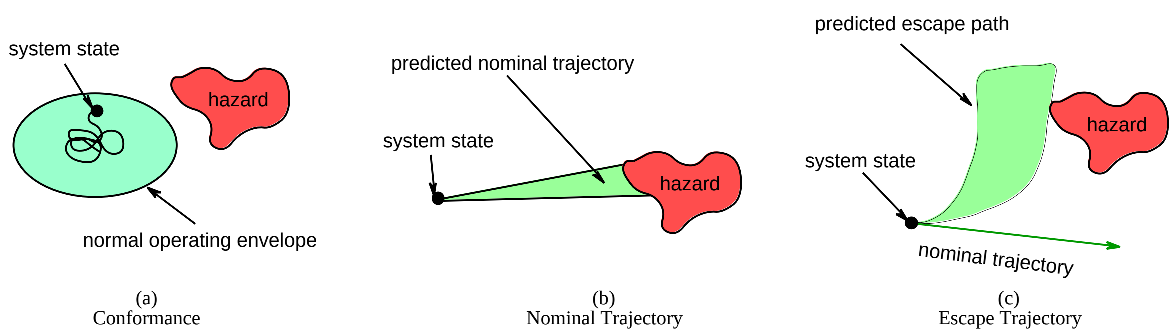


Figure 2: Fundamental Alerting Philosophies

The basic philosophy behind a conformance system (Figure 2a) is one in which alerts are considered to be justified when the aircraft does not follow expected behavior. More formally, a boundary of acceptable operating states is defined beforehand, and an alert is issued when the state of the aircraft exits this boundary. The boundary is placed around the normal approach corridor for the aircraft, for example, and should enclose a large enough region that false alarms during a normal approach (due to typical dynamic oscillations) are unlikely. The boundary should also be small enough that it does not lie too close to hazards — this ensures that enough time will be available to correct a problem if the state does exit the boundary. An example of a conformance system is the Precision Runway Monitor (PRM), which is active at a few airports in the United States [1]. PRM enables simultaneous independent approaches to runways as close as 3,400 ft apart in instrument conditions.

By constructing the region enclosed by the boundary to be free of hazards, it is possible to ensure safe operation as long as the state remains inside the boundary. This can greatly simplify the monitoring process, since it is then only necessary to ensure that hazards remain outside the boundaries and that the process states remain within the boundaries. Selecting the appropriate size of the boundary can be problematic, however, if the normal state is expected to vary greatly (in which case it may be difficult to prevent nuisance alarms) or when hazards lie close to the boundary (in which case there may not be enough time to prevent an accident after the boundary is exceeded). Still, this approach is relatively simple in that it relies only on the current state, so future trajectory predictions are not required. In fact, no explicit prediction of a hazard encounter is needed at all — simply knowing that the process is not following expected norms is enough to justify an alert. On the other hand, a normal range of state values must be determined, and so there generally must be some structure to the problem. Conformance methods would therefore be more appropriate for parallel approach problems in which normal aircraft positions can be readily identified, than for general free flight conflict detection systems in which aircraft could be located anywhere and be going in any direction. Adding aircraft flight plan information in free flight, however, may enable conformance methods to be used. In that case, flight plans can first be checked for conflicts, and then aircraft conformance to each plan can be monitored.

In the second philosophy (nominal trajectory, Figure 2b), the state of the process is projected into the future using some form of trajectory model. The projection is used to determine whether a hazard is explicitly expected to be encountered if the current control strategy continues. Should it become likely that a hazard will be encountered, an alert is then issued. This method is used in many collision alerting systems, including the Traffic Alert and Collision Avoidance System (TCAS), and in the proposed Airborne Information for Lateral Spacing (AILS) System for parallel approach [4,5].

The rationale behind the nominal trajectory method is that alerts are issued only when they are necessary to avoid a hazard. If the future trajectory does not encounter a hazard, an alert is not issued. The accuracy of trajectory prediction generally degrades into the future, so some cutoff or maximum lookahead time is typically required to avoid nuisance alarms. Additionally, alerting when the nominal trajectory is projected to encounter a hazard does not by itself guarantee that the alert will be successful in avoiding that hazard — it may already be too late to prevent a hazard encounter. As is discussed in more detail later in this paper, the appropriate alerting distance is typically determined through trial-and-error tests using fast-time simulations of aircraft encounters. The thresholds are then set to provide, on average, the best performance over the set of encounter conditions.

The third design approach for alerting systems (Figure 2c) is to issue an alert when the expected escape path is threatened by a hazard. This method extrapolates a trajectory from the current state into the future, but based on the assumption that an alert is issued and corrective action is taken. Conditions for a safe escape need to be defined, and the escape path is examined to determine whether those escape conditions are reachable. If the escape conditions are not reachable at some level of confidence, then an alert is issued. This philosophy, then, emphasizes the desire to ensure that a safe escape corridor exists.

Two examples of the escape trajectory method applied to the parallel approach alerting problem are given by Refs. 6 and 7. In these prototypes, the future position of the aircraft while flying an escape or breakout maneuver is examined to determine whether a collision with another aircraft is likely. Upon reaching a certain level of risk, an alert is issued. The rationale is that the pilot should always have recourse to the escape maneuver, and an alert should be issued as soon as that maneuver's safety is threatened.

As discussed above, with the nominal trajectory philosophy it is possible that an alert, though determined to be necessary, may be too late to prevent encountering the hazard. With the escape trajectory philosophy it may be the case that the alert, though it will be successful in avoiding the hazard, is not necessary. This is because there may be no hazard along the nominal trajectory, even though the escape path is threatened. Accordingly, there is a similar design problem to determine the appropriate lookahead distance or confidence level when alerting. Extrapolating too far into the future may lead to issuing alerts that are not needed, reducing the overall efficiency of the operation.

Combinations of these three philosophies are certainly possible, and in fact are probably desirable in many cases. A sequential combination has been proposed for AILS, for example. In that system, a conformance alert is issued when one plane deviates from its approach course, and a different nominal trajectory-based alert is issued if a collision is explicitly predicted between aircraft. TCAS also applies a sequential approach in that alerts are first issued based on a nominal trajectory model, and then the appropriate escape maneuver is determined based on examining various avoidance actions.

Simultaneous combinations of philosophies are also possible. For example, simultaneously examining both the nominal and escape trajectories allows one to first ensure that an alert is necessary (by examining the nominal trajectory) and second that the alert will be successful in avoiding the hazard (by examining the escape trajectory). Ideally, then, alerts would only be issued when they are known to be both necessary and successful. In practice, however, most alerting systems do not use simultaneous combinations of philosophies. The result is that the algorithms only consider one aspect of the problem explicitly (e.g., directly determining that a collision is likely using a nominal trajectory model). The other considerations (such as ensuring that a safe escape is possible following the nominal trajectory-based alert) are typically ensured only indirectly by running a series of simulations and observing the outcomes that result. These alerts then would be expected to be successful in avoiding the hazard based on previous simulation results, but not by explicitly checking the escape path. This issue and its effect on achievable system performance is discussed in more detail in a later section.

Quantitative Connection Between Trajectory Uncertainty and System Performance

One fundamental issue at this point relates to which philosophy or combinations of philosophies should be used in a given problem. Notionally, it would be expected that the quality of decisions made by a trajectory prediction system would decrease as uncertainty in the future trajectory increased. In the limit, a decision based solely on a completely inaccurate trajectory prediction would have no diagnostic benefit. A conformance-based approach might fare better, however, by alerting simply when the state deviated from desired bounds. Conversely, given perfect predictability, a trajectory prediction system would likely outperform a conformance based system because it uses that additional accurate information to better diagnose the need to alert the human.

As an illustration of this concept, the quality of decision-making for a conformance system was compared against a nominal trajectory system as a function of the predictability of the trajectory. To do this, a Monte Carlo simulation of random trajectories was performed. Each trajectory traced the path of the state, whose lateral velocity was specified by a Markov process. A Markov process has the characteristic that the next state in time depends only on the current state and not on previous states [8]. The predictability of the trajectory can be specified in terms of the autocorrelation of the Markov process. The more highly correlated the process, the more accurately that the future trajectory can be predicted. Two levels of correlation were used here, with characteristic correlation distances of 100,000 m and 10 m (Figure 3). The 100,000 m correlation distance resulted in essentially straight-line paths, while the 10 m correlation distance case resulted in more noisy paths as shown in Figure 3.

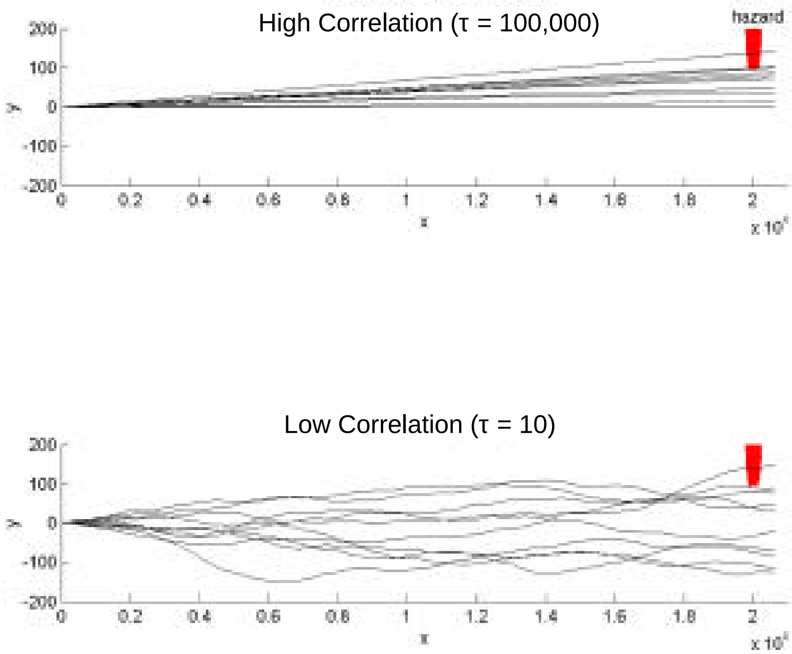
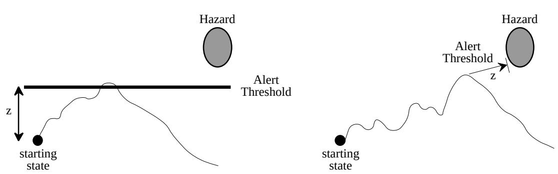


Figure 3: Example Trajectories

These trajectories were simulated in the presence of a hazardous region, shown in Figure 3. Alerting thresholds were then set using either a conformance method or using a trajectory prediction method (Figure 4). The conformance threshold was set at a parameter distance *z* laterally from the starting position as shown in Figure 4a. This lateral location was then systematically varied to trace out the performance of the system as a function of threshold position. In the trajectory prediction case, a projection from the current state was made using the instantaneous velocity vector. This projection continued for a parameter distance *z* as shown in Figure 4b; this parameter was also systematically varied to explore its effect on system performance.



(a) Conformance (b) Trajectory

Figure 4: Alerting Methods

Crossing an alert threshold altered the future trajectory of the process by adding a bias to the lateral velocity, simulating the corrective action taken in response to the alert. Depending on where the alert was issued, the state might still encounter the hazard even after this evasive maneuver had begun. At the moment an alert was issued, a second "ghost" trajectory was also simulated that followed the original Markov process statistics without the escape maneuver bias. This allowed for a check to see if the state would have encountered the hazard had no alerting system been present.

The outcome of each trajectory simulation was categorized as follows. Trajectories that produced an alert that was ultimately successful in avoiding the hazard were called successful alerts. The fraction of all alert cases that were successful was then denoted P(SA). Second, those alerts that were unnecessary were also counted. Unnecessary alerts were those in which the hazard would not have been encountered had the alert not been issued. In other words, after alert was issued the second ghost trajectory did not encounter the hazard. For that trajectory, then, the alert was not required according to this strict definition. The fraction of alerts that were unnecessary were denoted P(UA).

A given alerting threshold setting results in a single observed pair of P(SA) and P(UA) when averaged over a large number of simulations. The threshold setting for each method was then systematically varied, from extremely conservative (alerts were always generated) to extremely risky (alerts were never issued). This then traced out a so-called System Operating Characteristic (SOC) curve

[9]. A total of 5000 simulations were performed at each combination of threshold setting, alerting method, and trajectory correlation level.

The results are shown in Figure 5. In the high-correlation case, it can be seen that the trajectory prediction method performs very well. There is a threshold setting that provides nearly ideal performance, with almost no unnecessary alerts and with almost all alerts being successful (top left corner of the plot). The conformance system is not able to reach the same level of performance, regardless of threshold setting, and incurs a higher rate of unnecessary alerts.

In the low-correlation case, the trajectory prediction method performs poorly. Regardless of threshold setting, a high level of successful alert can only be attained while also incurring a high rate of unnecessary alert. The curve for the trajectory prediction case comes close to the diagonal line from (0,0) to (1,1) in the SOC plot, which indicates that the system is of little diagnostic benefit. The conformance system, however, is able to perform better than the nominal trajectory system in this case. Although both systems' performances are lower than in the high-correlation case, it is seen that a better decision can be made based on the current state (via the conformance boundary) than is possible when relying on inaccurate trajectory information.

Similar analyses can be performed to examine the relative quality of decision-making using each philosophy (or combinations of philosophies) under different conditions. This example serves to demonstrate, however, that a quantitative relationship can be obtained between the characteristics of a problem (e.g., uncertainties) and the performance that is achievable from a given philosophy. This quantitative relationship will be important in targeting design efforts toward the most effective modeling methods.

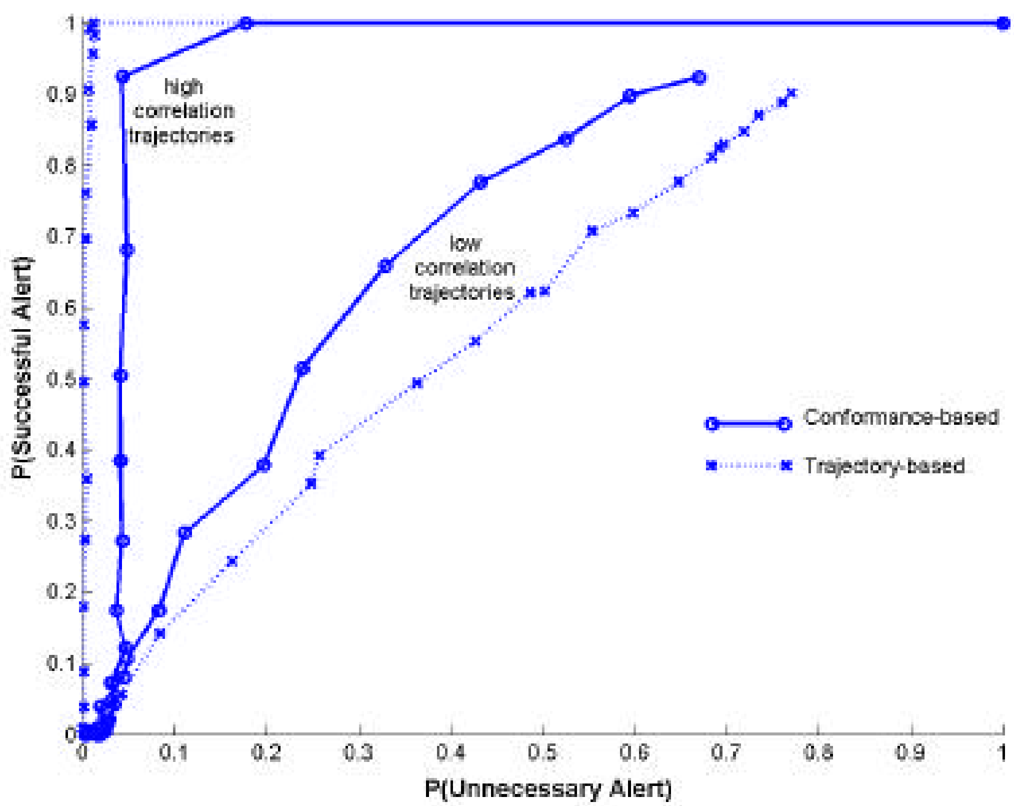


Figure 5: Alerting Performance Results

Trajectory Modeling

Even having determined that a trajectory-based philosophy may be most appropriate for a given problem, there is a rich design space to consider. A core consideration involves the type of trajectory model that is used to predict where aircraft will be in the future. Despite the wide variety of modeling approaches in conflict detection and resolution, for example, trajectory models can be reduced to three categories: maximum-likelihood, worst case, and probabilistic [2]. In the maximum-likelihood approach, the current states are projected into the future along a single trajectory without direct consideration of uncertainties. An example would be extrapolating a vehicle's position based on its current velocity vector, as is done with TCAS (Figure 6a). The maximum-likelihood projection method is straightforward and provides a best estimate of where the state will be, based on the current state information. In situations where state trajectories are very predictable (such as when projecting only a few seconds into the future), a maximum-likelihood model may be quite accurate. Maximum-likelihood projections, however, do not directly account for the possibility that the process or environment may not behave as expected — a factor that is especially important in longer-term decision making. Generally, this uncertainty is managed by introducing a safety buffer (e.g., minimum miss distance between vehicles) to reduce the likelihood of missed detections, and a maximum lookahead distance to reduce the rate of nuisance alerts.

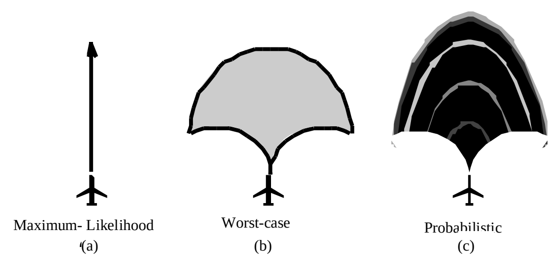


Figure 6: Trajectory modeling methods

The opposite extreme of dynamic modeling is to examine a worst case projection. Here, it is assumed that the state trajectory could follow any of a range of behaviors. If any one of these trajectories could encounter a hazard, then an alert is issued. The result is a swath of potential trajectories to be monitored (Figure 6b). Worst case approaches are conservative in that they can trigger alerts whenever there is any possibility of encountering a hazard within the definition of the worst case trajectory model. If such trajectories are unlikely, protecting against them may result in a high false alert rate. Accordingly, the worst-case approach may be appropriate when it is desirable to be conservative, or when dynamics are constrained within known bounds. AILS is an example system that uses a worst case trajectory model.

In the probabilistic modeling method, uncertainties are explicitly used to develop a set of possible future trajectories, each weighted by its probability of occurrence. For example, a distribution of future vehicle positions could be obtained by modeling uncertainties in winds or guidance (Figure 6c). A probabilistic approach provides an opportunity for a balance between relying too heavily on the state adhering to a single trajectory versus relying too heavily that the state exhibits a worst case behavior. The advantage of a probabilistic approach is that decisions can be made on the likelihood of encountering a hazard — safety and false alarm probabilities can be assessed and considered directly. The probabilistic method is also the most general, since maximum-likelihood and worst case models can be considered subsets of probabilistic trajectories. Three example probabilistic trajectory systems are the Center / TRACON Automation System (CTAS) developed by NASA [10], the User Request Evaluation Tool (URET) developed by MITRE [11], and a prototype system at MIT [12,13].

Decision Tradeoffs in Trajectory-Based Alerting

In the previous section, the notion was fielded that there may be benefit to combining a nominal trajectory philosophy with an escape trajectory philosophy. This would facilitate ensuring that alerts are issued when they are necessary and likely to be successful. A quantitative analysis of the benefits that can be gained from such an approach is discussed below.

Figure 7 shows a simplified situation involving an aircraft and some hazard to safe flight. At the current moment in time shown in the figure, the aircraft is located at an estimated position $\hat{\mathbf{x}}$. The aircraft is traveling along a nominal trajectory (labeled **N** in Figure 7) whose direction relative to the hazard can be estimated. This nominal trajectory is based on current knowledge of the intended paths of the aircraft and hazard, and may include intent information such as a flight plan entered into a Flight Management System.

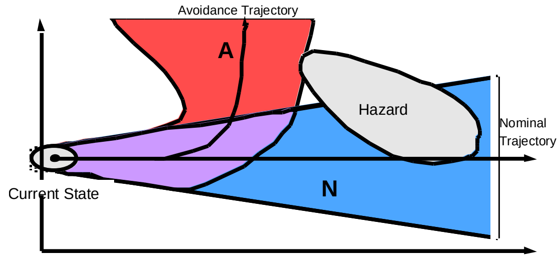


Figure 7: Nominal and Escape Trajectories

Regardless of whether a maximum-likelihood, worst case, or probabilistic model is used by the alerting system, at best the nominal trajectory is actually probabilistic, due to uncertainties. The probability of encountering the hazard in the future along the nominal trajectory is denoted $P_N(\hat{\mathbf{x}})$. As discussed previously, an alert is defined to be unnecessary if the hazard would not have been encountered assuming no alert had been issued. Therefore, the probability of an unnecessary alert, P(UA), is given by:

$$P(UA) = 1 - P_N(\hat{\mathbf{x}}) \qquad (1)$$

To minimize unnecessary alerts, the alert should be delayed until $P_N(\hat{\mathbf{x}})$ is close to 1. If the alert is delayed too long, however, there may be insufficient time and space to avoid an incident. It is therefore advantageous to also consider the escape trajectory (labeled **A** in Figure 7) that is followed after an alert is issued. The escape trajectory is also probabilistic in general, with a corresponding probability of encountering the hazard along the escape trajectory denoted $P_A(\hat{\mathbf{x}})$. If $P_A(\hat{\mathbf{x}})$ is close to 1, then it is likely that an alert is too late or is issued at an inappropriate time and the hazard may be encountered even with (or because of) the alert. In keeping with the earlier discussion on successful alerts, an alert is successful with probability P(SA):

$$P(SA) = 1 - P_A(\hat{\mathbf{x}}) \qquad (2)$$

The tradeoff between unnecessary alerts and successful alerts can be shown using a System Operating Characteristic (SOC) curve, shown in Figure 8 [9]. SOC curves are similar to Receiver Operating Characteristic (ROC) curves in Signal Detection Theory (SDT) and allow the alerting decision to be recast as a conventional signal detection problem. This enables the use of established SDT methods to determine an optimal alert threshold. An ideal alerting system would operate at the upper-left corner of the plot, but in general alerting systems are constrained to operate on SOC curves that do not pass through the ideal location. However, as uncertainties in the situation are reduced or as the avoidance maneuver becomes more aggressive, the SOC curve will approach the ideal operating point. It is also worth noting that the performance evaluation shown previously in Figure 5 is an SOC curve.

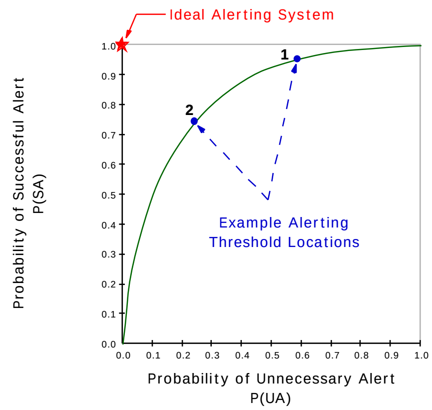


Figure 8: System Operating Characteristic Curve

The benefit, then, of simultaneously employing both the nominal and escape trajectory philosophies is that alerts can be controlled such that they are only issued when the probability of success is high and the probability that the alert is unnecessary is low. This would lead to a more effective system than one where, for example, only the nominal trajectory was considered. Essentially, in a nominal trajectory system only P(UA) is explicitly monitored, since the likelihood of encountering the hazard is only examined along N. Conversely, in an escape trajectory system only P(SA) is explicitly monitored by examining whether the hazard will be encountered along A. Acceptable performance in these two cases is achieved using the system design process discussed below.

System Design Process

Stepping back still one more level, there are also several issues to consider from the standpoint of the large-scale design process used when developing a decision-aiding system. These issues relate to the underlying metrics upon which decisions are based: that is, the specific parameters and variables that are estimated or computed and whose values determine whether an alert is issued or not.

The design process that is used in the majority of cases involves the process shown schematically in Figure 9. First, a set of decision metrics are developed for the system, based on physically-measurable parameters. Example metrics might be expected miss distance between aircraft, or the estimated time until minimum separation is reached. The decision-making model and its parameters (e.g., the threshold settings) are then exercised in a series of simulations (either through fast-time Monte Carlo simulation or human-in-the-loop studies). The decision logic is exposed to a wide range of encounter situations, and the resulting number of false alarms and loss-of-separation events (or other statistical performance metrics) are recorded [14,15]. Example situations typically include a variety of conflict geometries and aircraft dynamic behavior. This allows for uncertainties to be modeled and injected into the design of the system in order to explore system performance and robustness to uncertainty. If the observed system performance does not meet design specifications, then the model or the decision thresholds are modified. For example, time or range thresholds may be successively modified until there is an acceptable balance between loss of separation incidents and false alarms over the set of test scenarios. The result can be a complex, iteratively-evolved set of logic and threshold definitions. The TCAS alerting thresholds developed using this method, for instance, have numerous kinks and overhead associated with special cases using if-then logic [4].

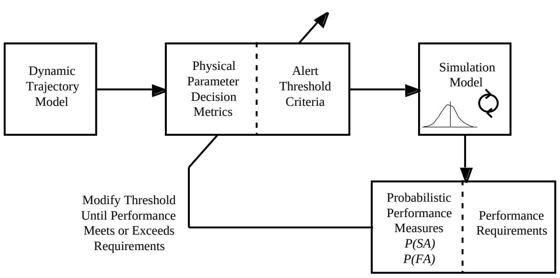


Figure 9: Physical Parameter Design Process

Ultimately, it is the observed performance in terms of false alarms and safety that determines whether a system design is acceptable. These performance parameters are typically based on the decisions that occur and on the outcomes that result, rather than on physically-measurable quantities like time or position. Closer examination of the design method in Figure 9 reveals that, at its core, what is happening is that the system's physically-based decision parameters are being tuned to the situations that are provided in the evaluation simulations. The process is somewhat analogous to designing a control system compensator, but in this case the decision-aid system is essentially mapping the given encounter situations into false alarm rate or separation performance. In this view, metrics such as range, miss distance, or time are simply surrogates for the real metrics of interest — the statistical measures of performance like safety and false alarm rate.

A more direct approach to system design would be to estimate performance parameters in real time during system operation (Figure 10). Then, rather than making a decision based on an indirect metric such as time to minimum separation, the alerting decision can be based on a direct comparison of the computed false alarm or loss of separation probabilities against the desired performance specifications.

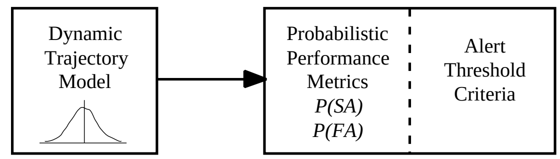


Figure 10: Performance-Based Design Process

Whereas physical parameters were once the only types of information available to an alerting system due to sensor and computational limitations, it is now becoming possible to estimate performance parameters in real time. One recent demonstration of a performance-based design approach is a conflict detection probe developed for simulation studies at NASA Ames Research Center [12,13]. This system uses a probabilistic trajectory model to estimate the probabilities of unnecessary alert and successful alert in real time. Whether an alert is issued then depends only on these probabilities, and not on physical parameters such as miss distance or time to closest point of approach.

Concluding Remarks

Ensuring effective decision-aiding system designs is going to become increasingly more important as these systems are employed in higher-criticality roles. Although a number of capable alerting systems have been developed to overcome challenging problems, this development has largely been performed in the absence of an underlying science behind system design. Some engineering tools have been created to address specific sub-problems, such as the application of Signal Detection Theory to quantitatively describe performance tradeoffs. Yet there are still many degrees of freedom with which alerting systems can be developed in terms of the models that are used and the metrics used to make decisions.

Based on observations of many currently-proposed or operational systems, a more formal structure or taxonomy of designs is being developed, as outlined in this paper. First, there are several options as to the overall decision-making philosophy to be used, be it conformance or trajectory based. There are also several types of trajectory models to be employed, each with certain characteristics. Finally, decision metrics can be based on physically-measurable quantities such as time or position, or based on performance quantities such as false alarm rate or safety level. Selecting from each of these elements (decision philosophy, trajectory model, and decision metrics) leads to a certain level of achievable performance for a given problem.

The next step in this research is to continue efforts to quantitatively link a given problem and its characteristics to each design approach and demonstrate the resulting system performance that results. This will enable designers to select the modeling method that is most appropriate to the type of problem under consideration so as to achieve effective and acceptable systems in an efficient manner. The goal is to develop a process that begins with a clear definition of the problem to be solved and which directly leads to the modeling approach that should be used.

References

1. Shank, E. M. and K. M. Hollister, "Precision Runway Monitor", *Lincoln Laboratory Journal*, Vol. 7, No. 2, MIT Lincoln Laboratory, Lexington, MA, 1994.
2. Kuchar, J. K. and L. C. Yang, "A Review of Conflict Detection and Resolution Modeling Methods", *IEEE Transactions on Intelligent Transportation Systems*, Vol. 1, No. 4, 2000.
3. Winder, L. F. and J. K. Kuchar, "Generalized Philosophy of Alerting with Applications for Parallel Approach Collision Prevention", MIT International Center for Air Transportation Report ICAT-2000-5, August, 2000.
4. Radio Technical Committee on Aeronautics (RTCA), *Minimum Performance Specifications for TCAS Airborne Equipment*., Document No. RTCA/DO-185, Washington, D.C., September, 1983.
5. Waller, M. C. and C. H. Scanlon (eds.), "Proceedings of the NASA Workshop on Flight Deck Centered Parallel Runway Approaches in Instrument Meteorological Conditions", NASA Conference Publication 10191, Hampton, VA, December, 1996.
6. Kuchar, J. K. and B. D. Carpenter, "Airborne Collision Alerting Logic for Closely-Spaced Parallel Approach", *Air Traffic Control Quarterly*, Vol. 5, No. 2, 1997.
7. Teo, R. and C. Tomlin, "Computing Provably Safe Aircraft to Aircraft Spacing for Closely Spaced Parallel Approach", 19th Digital Avionics Systems Conference, Philadelphia, PA, October, 2000.
8. Gelb, A. (ed.), *Applied Optimal Estimation*, The Analytic Sciences Corporation, MIT Press, Cambridge, MA, 1974.
9. Kuchar, J. K., "Methodology for Alerting-System Performance Evaluation", *AIAA Journal of Guidance, Control, and Dynamics*, Vol. 19, No. 2, March-April, 1996.
10. Isaacson, D. and H. Erzberger, "Design of a Conflict Detection Algorithm for the Center/TRACON Automation System", 16th Digital Avionics Systems Conference, 9.3-1 - 9.3-9, Irvine, CA, October 26-30, 1997.
11. Brudnicki, D., Lindsay, K., and A. McFarland, "Assessment of Field Trials, Algorithmic Performance, and Benefits of the User Request Evaluation Tool (URET) Conflict Probe", 16th Digital Avionics Systems Conference, 9.3-35 - 9.3-44, Irvine, CA, October 26-30, 1997.
12. Yang, L. C. and J. K. Kuchar, "Prototype Conflict Alerting Logic for Free Flight", *AIAA Journal of Guidance, Control, and Dynamics*, 20(4), July-August, 1997.
13. Yang, L. and J. K. Kuchar, 2000, "Aircraft Conflict Analysis and Real-Time Conflict Probing Using Probabilistic Trajectory Modeling", International Center for Air Transportation Report No. ICAT-2000-2, MIT, May, 2000.
14. Haissig, C., Corwin, B., and M. Jackson, "Designing an Airborne Alerting System for Closely Spaced Parallel Approaches", AIAA-99-3986, AIAA Guidance, Navigation, and Control Conference, 280-287, Portland, OR, August 9-11, 1999.
15. 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.