A KNOWLEDGE-BASED CONFLICT RESOLUTION ALGORITHM FOR TERMINAL AREA AIR TRAFFIC CONTROL ADVISORY GENERATION

Douglas R. Isaacson* and John E. Robinson III*
NASA Ames Research Center, Moffett Field, CA, USA

Abstract

NASA researchers, working under the Aviation System Capacity Program and in conjunction with the FAA Free Flight Program Office, have developed a set of decision support tools to assist terminal area air traffic controllers with control of arrival and departure traffic. Two of these tools, the Final Approach Spacing Tool (FAST) and the Expedite Departure Path (EDP) tool, provide air traffic controllers with heading, speed and altitude advisories to assist in spacing aircraft. This paper describes the conflict prediction and resolution algorithm shared by aFAST and EDP to produce conflict-free aircraft trajectories with realistic conflict resolution maneuvers. The process is accomplished in three stages: prediction, classification and resolution. A conflict prediction scheme is documented which incorporates all applicable FAA separation requirements, including automatic detection of separation during transition from staggered to simultaneous parallel approach operations. A method of classifying predicted conflicts with a limited set of criteria is detailed. Finally, a knowledge-based conflict resolution process is presented which allows for resolution of predicted conflicts in a manner consistent with controller practice: including prioritization of resolution tactics and mixture of multiple degrees of freedom to achieve separation. The scheme has been employed in both closed-loop simulations to determine solution stability and controller-in-the-loop simulations to begin development of the resolution tactics knowledge base.

Introduction

Large increases in air traffic over the past twenty years have led to substantial delays in the United States air traffic system. Traffic is expected to continue growing at the rate of 3-5% annually over the next decade.[1] Budget and environmental constraints make it difficult to significantly increase the capacity of the National Airspace System (NAS) through infrastructure improvements such as additional runways or new airports. Instead, much effort has been devoted to increasing the efficiency of current operational practices through the use of decision support systems for air traffic controllers (or controllers) and traffic management coordinators (TMCs). Tools have been developed to assist TMCs in developing an efficient plan to manage arrival traffic into the terminal area. Decision support systems have also been implemented to assist the terminal area controller in executing this plan. To date, assistance to radar controllers has been in the form of passive advisories; the efficient plan is presented to the controller in a concise format, but the execution of the plan is the responsibility of the controller. Current research is focused on providing the controller with the necessary information to execute an efficient plan with high precision. The precision is improved by providing the controller with active advisories to be issued to the pilot in the form of heading, speed and altitude clearances. However, relying on an advisory system to provide heading, speed and altitude advisories requires the solution be both conflict-free and easily executable by the controller and pilot.

Background

NASA Ames Research Center has developed a suite of air traffic decision support tools known as the Center/TRACON Automation System (CTAS). Two of these tools have been implemented, tested and included in the FAA Free Flight Phase One deployment plan: the Traffic Management Advisor (TMA) and the passive Final Approach Spacing Tool (pFAST). TMA provides an efficient schedule for arrival aircraft entering congested terminal areas. Within the Air Route Traffic Control Center (ARTCC or Center), the Traffic Management Unit (TMU) employs TMA to generate an efficient arrival schedule and to present this schedule to the radar controllers. TMA employs time-based scheduling to estimate delays required for each aircraft. The controller meets the schedule by vectoring or slowing each aircraft until the required amount of delay for an aircraft is absorbed. During field trials, and subsequent daily operation, TMA has been shown to reduce delay by an average of two minutes per aircraft.[2]

The Final Approach Spacing Tool was originally envisioned as an active advisory tool, but early human factors assessments of FAST determined heading and speed advisories unacceptably cluttered the monochrome display available to controllers at the time.[3] It was decided that FAST be implemented in two stages: passive FAST (pFAST) and active FAST (aFAST). Thus, pFAST was developed and tested as a

*Research Scientist, Terminal Area ATM Research Branch tool that provided controllers in the Terminal Radar Approach Control (TRACON) with advisories for runway assignment and landing sequence. Through field testing, pFAST was shown to increase airport throughput by 9-13%, while maintaining controller workload at existing levels.[4]

A portion of the success of pFAST and TMA is attributed to the method of design for all CTAS tools. By involving a team of controllers and TMCs in the design of each tool from the earliest stages, major changes in the design of the system are prevented in the later stages of development. Furthermore, the underlying algorithms reflect the preferences of the users of the system. This aspect of the CTAS development process will become even more important with active advisory systems: tools that have a close relationship to the fundamental responsibilities of the controller.

As with any technological advancement in human directed systems, there will be resistance to the automated control instructions proposed by active advisory systems. While it may be possible to fully automate an air traffic system, it is unrealistic to believe such a system could be implemented as a discrete event (or even in a short period of time). Economic realities and safety concerns prevent the immediate implementation of a fully automated system; a more likely scenario is a gradual transition towards an automated system. To take advantage of emerging technologies while providing equal access to all aircraft would likely require the controller to take an active role in a semi-automated system. It is believed that controller acceptance of such a system is unlikely without a close relationship between the system and preferred controller practices; the advisories presented to the controller must closely resemble what the controller would instruct in the absence of the system. While this seems contradictory to the required change in operations, it should be noted that only minor modifications to an aircraft trajectory are required to eliminate inefficiencies in inter-aircraft spacing. Active FAST and its departure counterpart, the Expedite Departure Path (EDP) tool, are being developed with this requirement in mind.[5, 6]

Both aFAST and EDP employ an algorithm that considers the separation implications of each sequence decision when building the schedule. Reference 6 describes a concurrent sequencing and deconfliction algorithm, and demonstrates the importance of considering resolution tactics in determining a sequence of aircraft. This paper presents a method for resolving conflicts in a manner consistent with controller practices. First, a brief discussion of previous work is included, with special emphasis on spatial oriented scheduling. Next, the knowledge-based deconfliction algorithm is detailed in its three components: prediction, classification and resolution. The status of the algorithm is briefly discussed, followed by some concluding remarks.

Previous Studies

Previous conflict resolution studies have been predominately focused in three areas: collision avoidance systems, optimal conflict resolution, and time-based conflict resolution. Collision avoidance systems, such as the Traffic Alert and Collision Avoidance System (TCAS) will not be considered here, as they are designed with a different purpose in mind: to serve as a safety net if the primary separation systems fail. Optimal conflict resolution is concerned with efficiency of resolution maneuvers with a specific goal (or goals) to achieve (i.e. minimum fuel burn). Many schemes have been developed to optimally resolve aircraft conflicts in unconstrained airspace, and could be easily adapted to constrained airspace. Durand developed a novel approach to optimally resolve en route conflicts while constraining the number of maneuvers, albeit not the path of the resolution.[7] While optimal resolution techniques could be implemented in an active advisory system, it is not necessary and likely very difficult to model controller practices with optimal cost functions. An attempt was made to optimize pFAST sequence decisions based on Brinton's implicit enumeration algorithm.[8] However, as the cost functions became more complex, it became increasingly difficult to introduce additional controller preferences without significantly impacting the system.

Time-based conflict resolution is the most widely used method, but has some significant limitations. Time-based conflict resolution usually considers separation only at the schedule location; it does not consider separation at other points along the trajectory. This method has been used with varying levels of success in many systems that do not actively advise control instructions (e.g. TMA, Aircraft Situation Display). However, failing to consider constraints (such as merge locations) in aircraft scheduling is akin to modeling a complex system of multiple merging and diverging streams with a single queue. Extensive simulation and operational assessment of pFAST has shown a high level of schedule acceptance can be achieved by considering all merge locations in the TRACON; an acceptable result was never achieved by only considering the threshold scheduling constraint.[9]

While pFAST demonstrated the importance of considering all TRACON merge locations in producing an acceptable schedule, the fidelity to which it modeled these merge locations does not fully support active advisories. Passive FAST modeled the arrival merge locations as static merge points, while they in fact can move within the defined airspace. For example, Figure 1a shows the final merge location for Aircraft B and Aircraft C if they are the only aircraft considered. Figure 1b shows the merge location if Aircraft A is already on the final approach. If a merge location moves a significant amount, the relative sequence in the schedule between aircraft can change. In Figure 1, as the merge location for Aircraft B and C shifts further from the runway, Aircraft B becomes closer to the predicted merge location until the point is reached that Aircraft B is sequenced ahead of Aircraft C. Conflict resolution is necessary to determine the location of a moving merge location (by determining the earliest allowed turn onto final in this example), and thus to determine the acceptability of a proposed sequence. Reference 6 describes this topic in great depth, and makes a strong case for concurrent sequencing and conflict resolution of arrival aircraft.

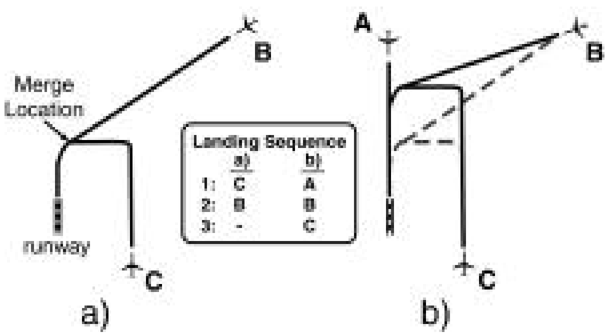


Figure 1: Effect of Conflict Resolution on Sequence

This paper describes the deconfliction algorithm used in the concurrent sequencing and deconfliction scheme described in Reference 6. As discussed, the acceptability of a proposed sequence depends on the conflict resolution of that sequence due, in part, to the variability of merge locations. It will be shown that the method of conflict resolution has a direct impact on this merge location. The conflict resolution algorithm described herein presents a method for modeling resolution tactics in a manner similar to those used by a skilled controller. By modeling controller resolution tactics, it is possible to more accurately predict the merge locations, and presumably to produce a schedule more acceptable to the air traffic controller.

Conflict Prediction Algorithm

The role of the conflict prediction algorithm is to predict if two aircraft will violate required separation: either spatial or temporal. For aFAST and EDP, it is required to adhere to FAA separation standards and to allow for controller or TMC specified separation requirements. Furthermore, to ensure schedule integrity, it is necessary to detect temporal, or schedule, conflicts at the runway threshold. This section will give a brief overview of the FAA separation standards, as well as a discussion of user-specified separation scenarios. Next, an overview of CTAS trajectory generation is given. Finally, the prediction algorithm and its possible outcomes are discussed.

Definition of a Conflict

The prediction algorithm recognizes two types of conflicts: spatial conflicts and schedule conflicts. A spatial conflict occurs when the distance and altitude between aircraft are both below the required minima. A schedule conflict occurs when aircraft arrive at the schedule constraint (e.g. runway threshold or metering fix) out of sequence relative to other aircraft in the schedule.

Spatial Conflicts: Spatial separation requirements are determined by the greater of the FAA prescribed minima or the user specified separation. This section provides an overview of the required separation minima, and a discussion of the circumstances under which user-specified separation standards apply. With the exception of user-specified separation, the requirements presented here are maintained in FAA Order 7110.65 (Air Traffic Control).[10]

Default Minima: The separation minima specify the default minimum separation required between two aircraft. It should be noted that an aircraft is only in conflict if neither lateral nor vertical separation is achieved. The vertical separation minima are 2,000 ft. above Flight Level 290 (29,000 ft. above sea level, standard sea-level conditions or FL290), and 1,000 below FL290. Lateral separation minima depend on the type of radar in use, and the proximity of the aircraft to the radar antenna. For terminal areas surrounding most major airports, the minimum lateral separation is 3nmi. There are many exceptions to these default minima, as will now be discussed.

Wake Turbulence Separation Application: For aircraft operating directly behind, or directly behind and within 1,000 ft. below, or following an aircraft conducting an instrument approach, wake turbulence separation standards override the default minima to avoid wake vortex interaction. The required wake turbulence separation depends on the weight class of the aircraft (refer to Reference 10 for weight class definitions and required separation), and the segment of flight. The required separation may be greater for aircraft on the same approach (or on approach to runways separated by less than 2500 ft.). The importance of the increased wake turbulence separation requirement to the conflict prediction and resolution algorithms will be further discussed in a later section.

Reduced Minima: Required separation may be reduced to 2.5 nmi for aircraft established within 10 nmi of the landing runway when the following conditions are met:

- Leading aircraft's weight class is the same or less than the trailing aircraft's weight class.
- Leading aircraft's weight class is not Boeing 757 or Heavy.
- An average runway occupancy time of 50 sec. or less is documented.
- Tower radar displays are in use.
- Turnoff points are visible from the control tower.

Dependent Parallel Instrument Approach Separation (Staggered Approaches): Required separation for parallel approaches is dependent on the distance between the runways. Parallel runways separated by less than 2500 ft. are to be treated as a single runway: wake turbulence and reduced minima rules apply. For aircraft established on parallel approaches separated by at least 2500 ft., but no more than 5300 ft., required lateral separation is1.5 nmi. For parallel approach courses separated by at least 5300 ft., but no more than 9,000 ft., 2 nmi of lateral separation is required. Furthermore, it is required to maintain a minimum of 1,000 ft. vertical separation or a minimum of 3 nmi lateral separation between aircraft during turn on to final approach.

Simultaneous Instrument Approach Separation: While separation is not required for aircraft established on independent approach courses, it is common for consecutive aircraft on parallel approach courses to be subjected to differing separation standards. The lead aircraft may be subjected to dependent approach separation standards due to final intercept location nearer the runway threshold, while the trailing aircraft may intercept the final approach course as required for independent approaches (further from the threshold). Such a transition is common for parallel approaches at busy airports as arrival demand and delay increase. Therefore, it is essential that the conflict prediction and resolution algorithm recognize when an aircraft is subject to dependent approach separation standards, and when an aircraft has met all the requirements for independent approach operations. Previous efforts have failed to adequately address this transition in real time, limiting the practical usability of these systems for active advisories.

User Specified: To accommodate user preferences and traffic management objectives, it is common for controllers to separate aircraft by more than the required separation. Aircraft declaring emergency may require additional separation, as well as aircraft with special security requirements. To coordinate arrival/departure operations on a single runway, the TMC may require gaps in the arrival stream for departures; this is accomplished through user-specified arrival separation. Some operational procedures may also be modeled in this manner. For example, inter-facility letters of agreement often specify the required separation when transitioning from terminal to en route airspace; TMC-specified spacing at the departure metering fix could vary depending on projected traffic load, and should be considered in the design of the conflict resolution algorithm.

Schedule Conflicts: Temporal, or schedule, conflicts refer to aircraft not adhering to a prescribed sequence or arrival time at the scheduling constraint. For arrivals, a schedule conflict occurs whenever an aircraft is predicted to land before an aircraft scheduled to land first. Enforcement of a sequence only applies at the scheduling constraint (i.e. the runway threshold or metering fix).

CTAS Trajectory Representation:
Accurate four-dimensional trajectories are the fundamental requirement for useful conflict prediction. All CTAS tools employ a common trajectory model that has been continually refined to accurately represent operational aircraft trajectories.[11, 12, 13] FAST and EDP model aircraft trajectories in the terminal area, including all degrees of freedom (DOFs) used by the controller during normal operations.

For each aircraft, a group of trajectories is generated every radar update cycle that span the degrees of freedom available to resolve conflicts and absorb delay. Figure 2 demonstrates the application of one such degree of freedom (base extension). Trajectories are computed for the minimum and maximum allowable application of each degree of freedom, and for every combination of degrees of freedom at minimum and maximum application. For each bound (degree of freedom limit), a corresponding delay value is associated: 0 sec. of delay is assigned to the minimum application of each degree of freedom, while the maximum amount of delay results from the maximum application of a degree of freedom. These so-called *fast* and *slow* limits for a degree of freedom determine the amount of delay that can be absorbed by that degree of freedom.

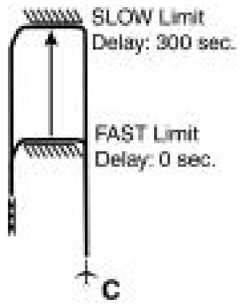


Figure 2: Base Extension Degree of Freedom Application

During the scheduling process, each four-dimensional trajectory is stored as a series of flight segments and turns connecting the segments. Each segment of flight along a trajectory is categorized as shown in Figure 3. These categories, referred to as spatial constraints (or constraints), are used to manage the scheduling process: all aircraft passing through a spatial constraint are sequenced and deconflicted on that constraint. For Aircraft C, the flight segments along the trajectory are classified as downwind, base, and final approach (or final). An aircraft is said to be a member of a spatial constraint if any of its flight segments are categorized as that constraint. The processes for determining constraint membership and sequence are described in references 9 and 6 respectively. Each trajectory is simultaneously represented as a series of equally spaced (temporally) state vectors, called time steps. The beginning and ending time steps for a given constraint are referenced for every aircraft belonging to that constraint (Figure 3). All time steps between the beginning and ending time steps (for a single constraint) are defined to belong to that spatial constraint.

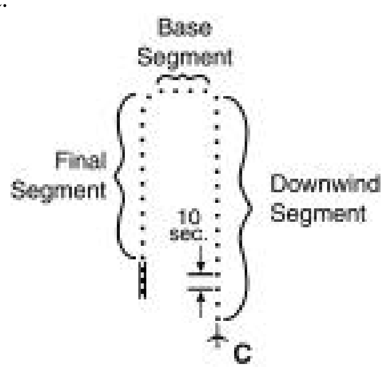


Figure 3: Time Step Constraint Membership

Conflict Prediction Method

Two methods are considered for conflict prediction given accurate four-dimensional trajectories: closed form solution and discrete location comparison. Software maintainability (algorithmic simplicity) and algorithmic performance requirements led to the decision to compare discrete trajectory time steps for each aircraft to predict conflicts for FAST and EDP. Reference 14 documents the method employed to compare time steps, along with several performance enhancing procedures to reduce the number of computations necessary to determine if a predicted conflict exists. The method presented allows for rapid comparison of trajectories with separation requirement changes along the flight path. The inclusion of variable separation requirements in the conflict prediction algorithm is essential for active advisory tools. While predicting conflicts based on maximum required separation along the trajectory may be sufficient to predict all potential conflicts, it has a direct impact on inter-arrival spacing when conflict resolution is considered. Furthermore, simply enforcing required separation at the scheduling location would fail to predict potential conflicts upstream of the scheduling location. An active advisory decision support system requires that the conflict prediction algorithm consider all separation criteria applicable in the airspace.

The conflict resolution procedure outlined in this paper places additional requirements on the conflict prediction algorithm. For reasons that will be discussed later, the conflict resolution algorithm employs a knowledge base that considers which spatial constraints each aircraft belongs to while in conflict, as well as previous and trailing constraints. These constraints are noted at the time of prediction. Additionally, the time steps at first loss of required separation, along with the time steps at minimum separation are logged. The first loss time steps are important, since they define the location along the dependent aircraft's trajectory after which modifications to the trajectory will not resolve the conflict. The minimum separation or closest time steps are used to determine the severity or magnitude of the conflict (i.e. how much delay must be absorbed to resolve the conflict).

Conflict Classification

The purpose of conflict classification is to group conflicts that share common resolution tactics. As with any knowledge-based approach, limiting the rule set is essential to maintaining an understandable association with user decision heuristics. Passive FAST demonstrated that it was possible to accomplish a complex controller task (arrival runway assignment) with a limited set of well-defined rules.[15] While it is not essential that every rule parallel a controller decision criterion, easily identifiable rules allow for rapid development of the knowledge base.

A controller can quickly evaluate and act on potential conflicts due to the repeatability of the scenario. In most cases, the scenario can be fully described by four factors: the conflict geometry, the conflict location, aircraft performance, and controller workload level. Aircraft performance prediction is inherently uncertain to the controller due to limited weather and aircraft state information. To ensure safety, the controller must act conservatively when resolving conflicts- often at the expense of efficiency. Decision support systems are more able than the controller to evaluate state and weather information in a timely manner. Within each radar update period, these systems produce accurate four-dimensional trajectory predictions. These trajectory predictions are an input to the conflict prediction algorithm; the remaining factors needed to define the conflict scenario must be included in the knowledge base. The conflict categorization step of this deconfliction scheme incorporates two of these factors: conflict geometry and conflict location. How controller workload impacts conflict resolution tactics depends on conflict geometry and location; thus, workload is considered in the conflict resolution portion of the knowledge base.

Conflict location is determined by inspection of the spatial constraints the involved aircraft occupy while in conflict. Conflict geometry can also be ascertained by comparing the constraints occupied while in conflict. These two rules will be discussed, followed by a conflict classification example.

Determination of Conflict Constraint

The location of the conflict, hereafter referred to as the conflict constraint, is usually determined by the location of the dependent aircraft at first loss of separation. The dependent aircraft is determined by the scheduling logic (as the aircraft sequenced behind in a pair). Assuming the aircraft are sequenced alphabetically in Figure 4, the trajectory segment corresponding to the conflict constraint is indicated in bold for each conflict type. Figure 4b shows a merging conflict where the conflict constraint is defined as the constraint the aircraft are merging on to (FINAL in this case) rather than the constraint occupied at time of first loss of separation.

Determination of Conflict Geometry

Conflict geometry is determined by inspection of the trajectory segments in the vicinity of the conflict. Terminal area procedures are generally designed to isolate arrival and departure operations, resulting in a largely two-dimensional treatment of aircraft separation. While altitude separation is used in TRACON airspace, operational procedures are in place to prevent routine conflicts between climbing and descending aircraft. Four conflict geometry classes fully describe two-dimensional conflict geometry: in-trail, merging, diverging and crossing. While conflicts between climbing and descending aircraft are predicted by the conflict prediction algorithm, they are classified as one of the two-dimensional conflict classes and resolved in a manner acceptable to TRACON controllers. Extension of the knowledge base to en route airspace is accomplished by including a rule to further delineate conflicts into level and non-level conflicts.

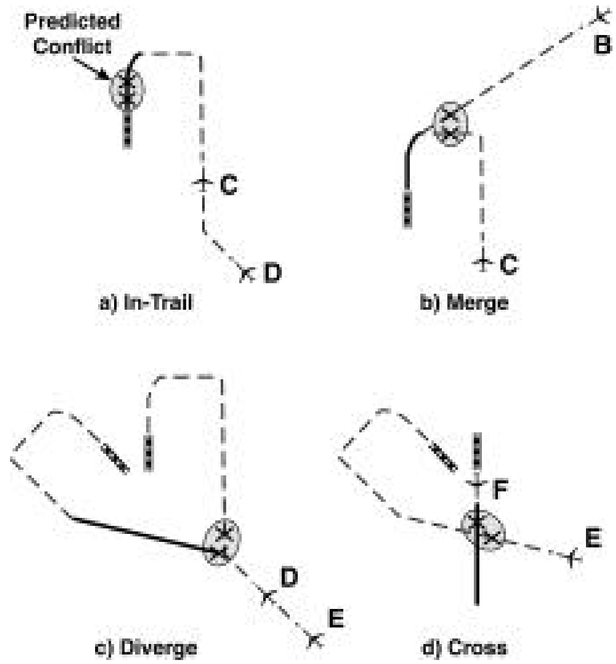


Figure 4: Conflict Types

In-trail Conflicts: Conflicts classified as in-trail occur when the involved aircraft share the conflict constraint and the previous constraint. Figure 4a depicts a sample in-trail conflict. In-trail conflicts can occur on a single constraint or across multiple constraints along a common flight path.

Merging Conflicts: Merging conflicts occur when aircraft share the conflict constraint, but not the previous constraint. Figure 4b gives an example of a merge conflict. Merging conflicts are common in the terminal area for arrivals merging toward final approach or through an arrival metering fix, and departures being merged through a departure fix.

Diverging Conflicts: Diverging conflicts occur when aircraft sharing the previous constraint, come into conflict on different constraints. The controller need not necessarily resolve diverging conflicts as long as they are not in conflict prior to path divergence. Allowances are made for aircraft otherwise in conflict if the courses are divergent by more than 15^{o} prior to loss of separation. Figure 4c shows two aircraft in conflict on diverging courses to land on different runways.

Crossing Conflicts: All conflicts not classified as in-trail, merging or diverging are classified as crossing. Crossing conflicts need not have crossing segments; only different constraints at time of conflict that are not merging or diverging. Crossing conflicts are common in en route airspace, but rare in the TRACON. However, prediction of crossing conflicts is essential for allowing direct climbs for departures thru arrival airspace (a central element of EDP). Departure aircraft frequently tunnel under arrival streams to procedurally prevent conflicts. Given the accurate trajectory prediction of FAST for arrivals, EDP will predict with a high degree of accuracy if direct climbs are possible (conflict free) for departure aircraft. Figure 4d depicts a crossing conflict between an arrival aircraft and a departure aircraft.

Conflict Classification Example

Conflict classification is accomplished through evaluation of the aforementioned rules in a decision tree format. If we assume we are to categorize the conflict of Figure 4b, following the conflict classification decision tree of Figure 5, we first must evaluate the conflict constraint (FINAL). Following the FINAL branch of the decision tree, the conflict class is evaluated and determined to be a MERGE type. The conflict is subsequently assigned the category FINAL_MERGE. The resolution tactics for FINAL_MERGE conflicts are defined in a separate knowledge base in the conflict resolution stage, as discussed in the next section.

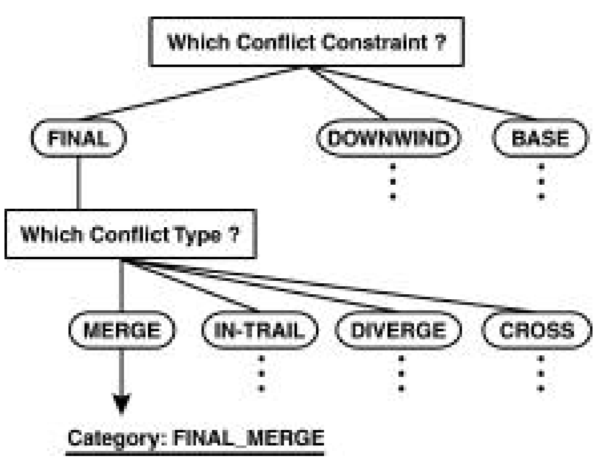


Figure 5: Conflict Classification Decision Tree

Conflict Resolution

Conflict resolution is accomplished in three steps: delay estimation, trajectory modification, and resolution verification. As previously described, the effects of the degrees of freedom are stored as delay perturbations about the nominal trajectory. To employ these delay perturbation estimates, it is first necessary to estimate the amount of delay required to resolve the conflict.

Resolution Delay Estimation

While the conflict prediction algorithm is spatially based, the conflict resolution algorithm is temporally based. The amount of additional separation required to achieve the required separation is translated into an equivalent amount of delay that needs to be absorbed by the trailing aircraft prior to first loss of required separation. The resolution delay (T_{RES}) is computed from the following equation where V_G is the dependent aircraft's ground speed at minimum separation, S_{REQ} is the required horizontal separation and S_{MIN} is the predicted minimum horizontal separation:

$$T_{RES} = (S_{REQ} - S_{MIN})/ V_G \tag{1}$$

Because V_G is not known for the post-resolution trajectory, it is assumed constant. An iterative approach to the resolution process accounts for the errors introduced due to this assumption.

Trajectory Modification

To resolve the conflict, it is necessary to modify the trajectory. Specifically, the degrees of freedom must be applied to absorb the delay required to avoid conflict. While estimating the aggregate delay required to resolve the conflict is straightforward, allocating this delay to the available degrees of freedom requires consideration of controller preferences and workload.

As mentioned previously, automated conflict resolution tactics must closely resemble those employed by skilled controllers. Furthermore, it is essential to the scheduling algorithm that the resolution tactics assumed in the deconfliction accurately predict the resulting merge locations of the proposed relative sequence. Just as neglecting to consider resolution maneuvers in determining a sequence wrongly results in a static merge location, employing degrees of freedom in proportions a controller would not use may result in inaccurate merge locations. For example, if Aircraft C (Figure 1) were to extend the base segment to resolve a conflict with aircraft B, the corresponding merge location would be further from the runway than if a combination of speed reductions were used. It is therefore essential that the method used to allocate delay to available degrees of freedom reflect controller preferences and practice. The consequences of this are two-fold: the scheme must allow prioritization of degrees of freedom and it must allow mixing of degrees of freedom to resolve a single conflict.

Controller workload is the final factor considered in the design of the conflict resolution algorithm. As workload increases, controllers become less concerned with efficiency, and more concerned with maintaining situational awareness and safety. The tactics controllers employ at high workload levels may not be the same as those employed at lower workload levels. To avoid conflict scenarios with limited options, controllers use limited amounts of multiple degrees of freedom rather than a large amount of a single degree of freedom. Furthermore, as delay and workload increase at a schedule point, delay is gradually distributed to aircraft upstream in the traffic flow. Early delay allocation implies that different degrees of freedom must be used to resolve a potential conflict when delay is high in the terminal area. This effect must be accounted for in the design of any active advisory system.

A knowledge-based conflict resolution procedure has been developed that includes DOF prioritization, DOF mixing and controller workload effects on resolution tactics. A group of tailored resolution tactics, or resolution strategy, is developed for each category of conflict enumerated in the conflict categorization knowledge base. Each resolution strategy defines degree of freedom prioritization and degree of freedom mixing strategies for the category of conflict being resolved. As discussed below, the design of the resolution strategy allows for consideration of resolution tactics that vary with increasing controller workload. First, the structure of the resolution category is discussed.

Table 1 shows a sample resolution strategy for the conflict category discussed previously (FINAL_MERGE). This resolution strategy is comprised of three degrees of freedom for resolving FINAL_MERGE conflicts: base extension (as shown in Figure 2) and airspeed reductions to 190 KIAS and 170 KIAS that vary along the aircraft path. The list is prioritized with preferred degrees of freedom higher on the list. The numbers included with each degree of freedom define how and when the degrees of freedom are used in combination.

Degree of Freedom	Mix Threshold	Mix Ratio
Base Extension	0.5	0.5
190 Speed Reduction	0.7	0.5
170 Speed Reduction	1.0	0.0

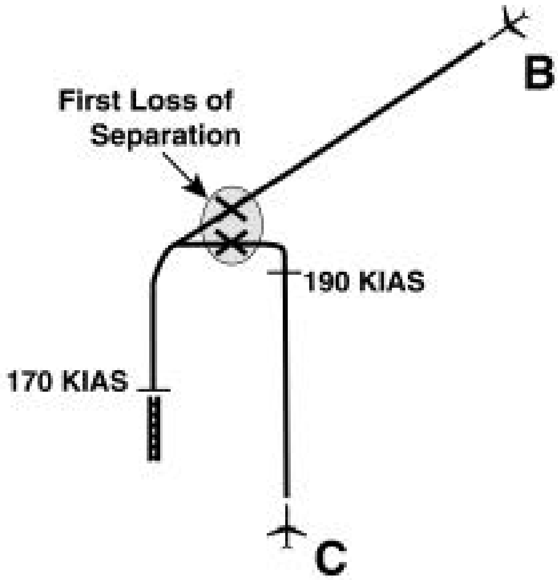
Table 1: Example Conflict Resolution Strategy

The first number for each degree of freedom is referred to as the mixture threshold and dictates the onset of degree of freedom mixing. The mixture threshold is a number between zero and one and is defined as the ratio of degree of freedom delay application to degree of freedom delay capability where mixing begins. For example, if the base extension degree of freedom for Aircraft C in Figure 2 provides up to 300 sec. of delay, the rule for base extension in Table 1 dictates that mixing of base extension with other degrees of freedom begins once 150 sec. of base extension delay has been applied. The second number after each degree of freedom defines the ratio that lower priority degrees of freedom are to be mixed with the employed degree of freedom once the mixture threshold has been reached. For the base extension degree of freedom for Aircraft C (Figure 2), a mixture ratio of 0.5 indicates equal amounts of base extension delay and cumulative delay of all following degrees of freedom once 150 sec. of base extension has been applied. By careful selection of DOF priority, mixture threshold and mixture ratio via simulation and controller feedback, it is hoped that the primary controller workload effects on resolution tactics will be captured. Prioritization and mixing with a delayed onset allow for use of the preferred degrees of freedom when workload is low, and for mixture when delay is high and workload is increasing. Assuming high delay indicates high workload, the mixture threshold provides a simple mechanism for employing differing tactics for higher workload levels. The following example illustrates the use of a resolution strategy to achieve a conflict-free solution trajectory.

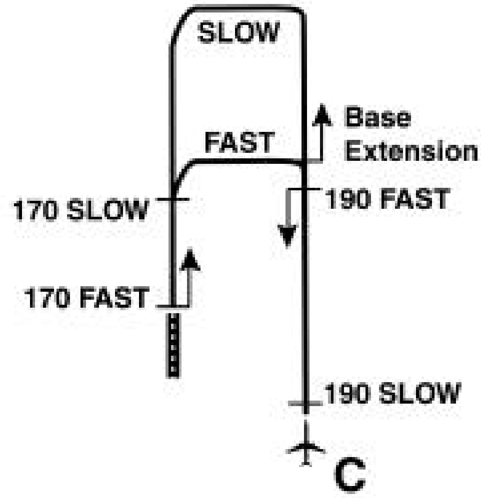
<u>Example of Conflict Resolution</u>

The conflict between Aircraft C and Aircraft B portrayed in Figure 6a has previously been categorized as FINAL_MERGE. First, we will assume Aircraft C has already absorbed 80 sec. of base extension delay due to the resolution of a previous conflict. Furthermore, we will assume the resolution time necessary to achieve required separation has been computed from Equation (1) to be 100 sec.. Aircraft C is on the downwind flight segment and is flying at 210 KIAS at an altitude of 9,000 ft. In our sample airspace, Aircraft C has three degrees of freedom remaining: base extension and speed reductions to 190 and 170 KIAS, with the application limits shown in Figure 6b and the corresponding delay values shown in Figure 6c.

The resolution is initiated with the first (highest priority) degree of freedom in the resolution strategy (from Table 1 for a FINAL_MERGE conflict). The resolution strategy indicates base extension is the preferred degree of freedom, and is to be used exclusively to resolve the conflict until 50% of its delay capability is used. Figure 6c indicates 300 sec. of delay potential for the base extension degree of freedom for Aircraft C. The mixture threshold for base extension is 150 sec. (50% of 300 sec.).



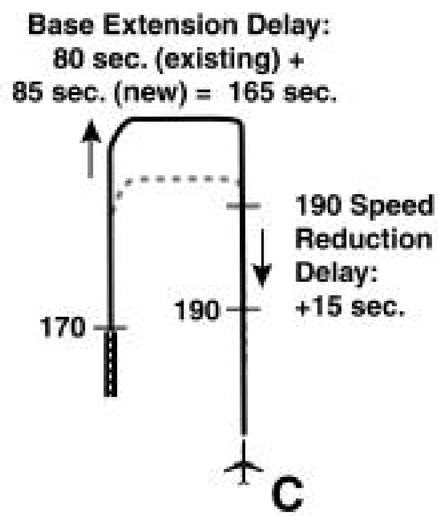
a) FINAL_MERGE Conflict



b) Degree of Freedom Limits

Degree of Freedom	Fast Limit	Slow Limit
Base Extension	80 sec.	300 sec.
190 Speed Reduction	0 sec.	30 sec.
170 Speed Reduction	0 sec.	10 sec.

c) Degree of Freedom Delay Values



d) Solution Trajectory

Figure 6: Conflict Resolution Example

Beyond 150 sec. of delay, base extension is mixed with equal parts of following degrees of freedom in the list. Figure 6b shows that the onset of the base extension degree of freedom (indicated by the arrow labeled 'base extension') is closer than the first loss of separation (indicated by trajectory penetration of the ellipse in Figure 6a). If the degree of freedom modifies the trajectory prior to the first loss of separation along the trajectory, the degree of freedom will assist in resolving the conflict. This check is repeated for every degree of freedom in the resolution strategy. The amount of delay to be absorbed by the base extension degree of freedom can then determined from the mixture threshold and mixture ratio (from Table 1), along with the previous base extension delay value. With an existing delay value of 80 sec., 70 sec. of base extension application can occur before mixing; this leaves 30 sec. of necessary delay to provide the required separation (100 sec.). According to the mixture ratio for base extension (50% from Table 1), 15 sec. of the remaining 30 sec. should be base extension, and 15 sec. should be absorbed by lower priority degrees of freedom. Using the same process, the remaining 15 sec. of delay are absorbed by the speed reduction to 190 KIAS. In this example, the resolution strategy for the FINAL_MERGE category results in 85 sec. of base extension delay and 15 sec. of delay from speed reduction to 190 KIAS earlier on the trajectory. Figure 6d shows the solution trajectory for the FINAL_MERGE conflict of Figure 6a adhering to the resolution strategy of Table 1. A new trajectory is computed and checked for conflicts to verify that the conflict has been resolved before moving on with the remainder of the scheduling process.

Resolution Verification

The application of each degree of freedom is accomplished by linearly interpolating between the minimum and maximum application values and the corresponding delay values. Because this relationship can be *non-linear*, the resulting trajectory may not fully resolve the conflict or may even add unnecessary delay to the trajectory. For this reason, a new four-dimensional trajectory must be computed. This new trajectory is again checked for conflicts, and the resolution process is repeated if a conflict is found. If no conflict is found for the new trajectory, further modifications may be necessary to remove any excess delay added to the trajectory. If the excess delay is greater than the prescribed resolution precision, the resolution tactics are adjusted according to the resolution strategy to eliminate the predicted excess. This entire process is repeated until no conflicts exist between the two aircraft concerned, and the delay absorbed is guaranteed to be within the prescribed tolerance of the minimum delay solution. Obviously, a tradeoff exists between the resolution precision and computational performance; initial trials have indicated acceptable performance with a five second resolution precision is easily attainable. Increasing computational capabilities, and yet-to-be-implemented performance-enhancing features will likely result in a one second resolution precision for aFAST and EDP.

Algorithm Status

The conflict prediction and resolution scheme presented in this work has undergone initial testing. As part of a stability demonstration of the scheduling algorithm employed by aFAST, the resolutions produced by this scheme were input to the software generating the aircraft targets in simulation. By doing so, basic stability of the system has been demonstrated: albeit without including uncertainties such as pilot reaction time and weather variance. Following this demonstration, controller-in-the-loop simulations have begun to refine the active FAST and EDP scheduling algorithm, including the conflict resolution knowledge base described herein. Future tests will attempt to demonstrate the closed loop stability of the system; this time including reasonable models of pilot and controller reaction time, weather uncertainties and state information uncertainty.

Concluding Remarks

A knowledge-based conflict prediction and resolution scheme has been developed that allows for realistic modeling of controller-preferred resolution tactics. A conflict categorization structure has been defined that uses only two rules: conflict location and conflict geometry. Four possibilities of two-dimensional conflict geometry have been identified: in-trail, merging, diverging and crossing. Conflict resolution is accomplished by defining tailored resolution tactics for each conflict category. The tactics defined in the resolution strategy incorporate degree of freedom prioritization, degree of freedom mixing and consideration of changing preferences at high controller workload levels. The knowledge-based approach presented here provides the structure and flexibility needed to model controller resolution procedures. It is hoped such an approach will lead to a high level of controller acceptance of proposed conflict resolutions and the resulting active advisories of the decision support system. Trials have demonstrated basic stability of the resolution scheme; real-time, controller-in-the-loop simulations are underway to refine the knowledge base. The simulations and resulting conflict resolution knowledge base will be presented in future publications.

References

1. Federal Aviation Administration, "Commercial Forecast Reports Eighth Consecutive Year of Aviation Growth," APA 16-00, U.S. Department of Transportation, Washington, D.C, March 2000.

2. Swenson, H. N., et al., "Design and Operational Evaluation of the Traffic Management Advisor at the Fort Worth Air Route Traffic Control Center," 1st USA/Europe Air Traffic Management Research and Development Seminar, Saclay, France, June 1997.

3. Lee, K.; Davis, T. J., "The Development of the Final Approach Spacing Tool (FAST): A Cooperative Controller-Engineer Design Approach", NASA TM 110359, August 1995.

4. Davis, T., Isaacson, D. R. and Robinson, J. E. III, "Operational Field Test Results of the Passive Final Approach Spacing Tool," Proceedings of the IFAC 8th Symposium on Transportation Systems, Chania, Greece, June 1997.

5. Johnson, C.W., Isaacson, D.R. and Lee, K.K., Expedite Departure Path (EDP) Operational Concept (MIT Lincoln Laboratory Report No. NASA/A-3), 1999.

6. Robinson, J. E. III and Isaacson, D. R., "A Concurrent Sequencing and Deconfliction Algorithm for Terminal Area Air Traffic Control," Proceedings of AIAA Guidance, Navigation, and Control Conference, Denver, CO, August 2000.

7. Durand, Nicolas, Optimisation de Trajectoires pour la Resolution de Conflits en Route, PhD Thesis, ENSEEIHT, Institut National Polytechnique de Toulouse, 1996.

8. Brinton, C., "An Implicit Enumeration Algorithm for Arrival Aircraft Scheduling", 11th Digital Avionics Conference, Seattle, WA, October 1992.

9. Krzeczowski, K. J., Davis, T. J., Erzberger, H., Lev-Ram, I. and Bergh, C., "Knowledge-Based Scheduling of Arrival Aircraft in the Terminal Area", AIAA Guidance, Navigation, and Control Conference, Baltimore, MD, August 1995.

10. Federal Aviation Administration, *Air Traffic Control*, Order 7110.65, U.S. Department of Transportation, Washington, D.C.

11. Erzberger, H.; Davis, T.J. and Green, S., "Design of the Center-TRACON Automation System," AGARD Guidance and Control Symposium on Machine Intelligence in Air Traffic Management, Berlin, Germany, May 1993.

12. Slattery, R. and Zhao, Y., "Trajectory Synthesis For Air Traffic Automation," AIAA J. Guidance, Control, and Dynamics, vol. 20, no. 2, pp. 232-238, March-April 1997.

13. Green, S. and Vivona, R., "Field Evaluation of Descent Advisor Trajectory Prediction Accuracy," AIAA Guidance, Navigation, and Control Conference, San Diego, CA, July 1996.

14. Isaacson, D. R. and Erzberger, H., "Design of a Conflict Detection Algorithm for the Center/TRACON Automation System," Proceedings of the 16th Digital Avionics Systems Conference, Irvine, CA, October 1997.

15. Isaacson, D. R., Davis, T. J. and Robinson, J. E. III, "Knowledge-Based Runway Assignment for Arrival Aircraft in the Terminal Area," Proceedings of AIAA Guidance, Navigation, and Control Conference, New Orleans, LA, August 1997.