To be published at the AIAA Guidance, Navigation and Control Conference, New Orleans, LA, August 1997.

FUZZY REASONING-BASED SEQUENCING OF ARRIVAL AIRCRAFT IN THE TERMINAL AREA

John E. Robinson III[*], Thomas J. Davis[*] and Douglas R. Isaacson[*]
NASA Ames Research Center, Moffett Field, California 94035

Abstract

A fuzzy reasoning-based method for scheduling air traffic in the terminal area has been designed at the NASA Ames Research Center and operationally tested at the Dallas/Fort Worth Terminal Radar Approach Control (TRACON). The scheduling system sequences and assigns landing times to arrival aircraft by utilizing continuous updates of aircraft radar data and controller inputs. The scheduling algorithm contains a knowledge base which was refined during several thousand hours of controller-in-the-loop real-time simulations. This knowledge base applies fuzzy reasoning to evaluate propositions that consider both performance criteria and workload criteria, such as delay reduction and conflict avoidance. Operational test results show that the algorithm determines an efficient arrival plan to land aircraft in a manner acceptable to the air traffic controllers. This paper details the scheduling algorithm and presents data regarding its effectiveness in predicting the landing order of arrival aircraft.

Introduction

The nation's air traffic system has experienced significant growth over the past fifteen years that has resulted in substantial increases in delays at nearly every major airport. This increase in traffic has not, however, been met with many deployed and operationally sound technologies for aiding the air traffic management and controller workforce. Many efforts to alleviate these problems have focused on developing decision support systems that schedule air traffic on a real-time basis in order that the traffic may be better managed and controlled. The main objective has been to provide an accurate prediction of the future traffic situation such that accurate traffic management and control decisions can made early, thus resulting in more efficient traffic flow.

NASA Ames Research Center has been conducting research on algorithms and decision support tools for the extended terminal area for nearly twenty years. This research has resulted in the development of a suite of tools known as the Center/TRACON Automation System (CTAS).[1] CTAS elements include the Traffic Management Advisor (TMA) to aid enroute traffic management, the Descent Advisor (DA) to supply arrival descent clearance advisories and management, and the Final Approach Spacing Tool (FAST) to provide sequence, runway, heading, and airspeed advisories to TRACON air traffic controllers.[2-5]

Previous Work

In order to build air traffic decision support systems such as CTAS, researchers at various labs worldwide have experimented with a wide range of algorithms for scheduling arrival aircraft. All of the methods explored to date build a schedule or plan based on estimated times of arrival at a fixed point, typically a runway threshold or meter fix. In Germany, Volckers successfully developed and deployed the COMPAS system which utilized a branch-and-bound algorithm to schedule arrival aircraft and provide controllers with broad advisories on whether to expedite, maintain, or slow aircraft in order to meet an overall strategic traffic plan.[6] The inputs to this scheduling algorithm included estimated times of arrival at a metering fix and runway threshold based on models of aircraft performance and the arrival route structure. In France, Garcia developed and deployed the MAESTRO system based on similar algorithms and modeling as the COMPAS system. MAESTRO, however, included expert systems logic for determining the most desirable runways in order to maintain proper airport balancing.[7] Brinton developed a more sophisticated branch-and-bound algorithm which again scheduled aircraft at a single point for an early version of a terminal controller tactical advisory tool.[8] While this scheduler produced near-optimal delay minimization, its schedules were found to be too difficult for controllers to implement in the busy terminal area air traffic environment.

Meanwhile, Erzberger designed the scheduling algorithm that forms the basis for the TMA functionality in CTAS.[9] The TMA scheduling algorithm utilizes accurate aircraft, winds aloft, and air

---

[*] Research Scientist, Terminal Area Air Traffic Management Research Branch

traffic control models as inputs to a 4D trajectory synthesis generator. The trajectory synthesis algorithm generates highly accurate 4D trajectories and estimated times of arrival (ETAs). These ETAs are the inputs that the scheduling algorithm uses to produce scheduled times of arrival (STAs) at the runway threshold and at the meter fix or entry point into a TRACON. Controllers use the STAs to "meter" arrival traffic into the TRACON and thus produce the desired flow rate of traffic. Operational tests of the TMA scheduling system produced highly successful results at Fort Worth Center in mid-1996.[10]

All of the methods for scheduling arrival aircraft described above were attempted during the development phase of FAST; however, all were ultimately unsuccessful. While these methodologies produce favorable results for decision support tools that provide strategic planning, none encompass the necessary factors to be acceptable in a more tactical decision support environment. In order to provide the detailed tactical advisories directly to the air traffic controllers, it was learned through real-time, controller-in-the-loop simulations, that the scheduling algorithm must account for a wide range of factors including efficiency, controller workload, and adaptability.

Based on the above, an initial spatial constraint satisfaction algorithm was designed to schedule aircraft at multiple points along their trajectories, not just at the runway threshold or meter fix.[11] Like the TMA scheduling algorithm, this early FAST scheduling algorithm utilized accurate 4D trajectories as inputs. All merges within the extended terminal area were considered by breaking the large global sequencing problem of merging at the runway threshold into a network of smaller local sequencing problems of merges at intersections of trajectory segments. A refined and enhanced spatial constraint satisfaction concept remains the foundation of the present FAST sequencing algorithm and will be presented as the topic of this paper.

Motivation

As a tactical decision support tool, FAST supplies air traffic controllers with assigned runway, sequence number, heading and airspeed advisories for aircraft in the terminal area. The underlying premise of its design has been that all increases in airport capacity must only be achieved in a manner that is acceptable to the air traffic controllers. A subset of the FAST functionality, encompassing only the sequence number and assigned runway advisories, was recognized as the minimum required functionality of a tactical decision support tool intended for the terminal area. This version of FAST, called "Passive" FAST, underwent a successful operational test at the Dallas-Fort Worth TRACON in mid-1996.[12]

Sequence advisories alone cannot significantly increase delay savings or airport throughput. Brinton has shown that the delay benefits of an optimal sequence versus a first-come-first-serve (FCFS) sequence in the terminal area are minimal.[8] Moreover, sequence advisories do not provide enough information to enable the reduction of mean excess spacing at the runway threshold. However, sequence advisories decrease a controller's workload by increasing his situational awareness. For example, a controller recognizes that extra separation is needed between two aircraft in order to merge another aircraft not yet under his control, when the sequence numbers of the two aircraft are not consecutive. In addition, accurate schedule information about traffic inside the terminal area, but under the control of another controller, reduces the controller-to-controller communication during hand-offs.

The remainder of this paper describes the design methodology and effectiveness of the sequencing algorithm used by Passive FAST. The other primary functionality of Passive FAST, runway assignment, is discussed in Reference 13.

Controller Techniques in the Terminal Area

In order to fully understand the design requirements for a sequencing algorithm in the terminal area, common techniques employed by air traffic controllers must be understood. These practices describe situations where the controller departs from a delay-optimal or FCFS landing order. These departures should not be viewed as controller ineffectiveness. Rather, they are based on a deeper understanding of operational procedures and their impact on controller workload.

Using Standard Trajectory Segments

Air traffic controllers are responsible for defined partitions of airspace, known as sectors. Sector boundaries prohibit a controller from arbitrarily vectoring an aircraft along paths that may interfere with other aircraft not under his control. Instead, controllers conduct aircraft along a series of standard trajectory segments towards the airport. Figure 1 illustrates the typical arrival routes[†] to runway 17C at Dallas-Fort Worth (DFW) International Airport. Routes for low side arrivals begin with "short" segments, followed by "base" segments and a "final" segment. Similarly, routes for high side arrivals begin with "long"

[†] These routes reflect airspace procedures that were changed in late 1997 due to the addition of the new runway 17L at DFW.

segments, followed by "downwind" segments, followed by "base" segments and a "final" segment.

Trajectory segments limit the areas where aircraft from different traffic streams intersect and merge, and, in fact, describe the standard procedures for controlling the air traffic. For a controller, standard segments reduce the workload required to sequence the arrival traffic, because they eliminate the need to compare every aircraft with every other aircraft.

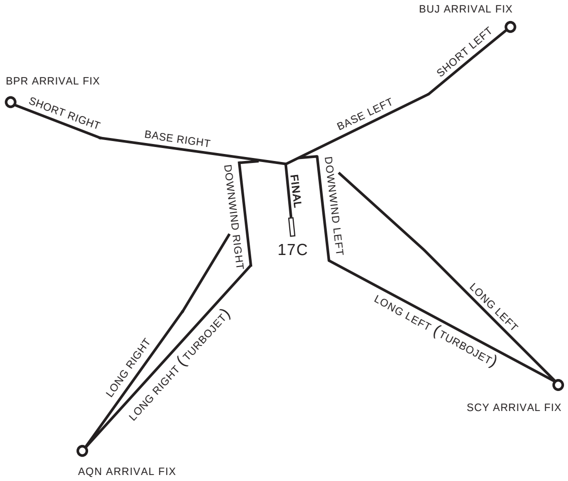

Fig. 1. Standard trajectory segments to Dallas/Fort Worth runway 17C

Allowing Overtakes

An overtake occurs when one aircraft passes alongside, over, or under another aircraft sharing the same trajectory segment. Overtakes in the terminal area are typically workload intensive for the controller and therefore undesirable in most cases. There are exceptions, however, and acceptable overtakes will generally need to meet three basic requirements.

First, an overtake must occur only in the airspace farthest from the airport. Nearly all arrival aircraft at a busy airport, such as Dallas/Fort Worth, are eligible for and capable of meeting the same set of standard clearances issued in the terminal area. Therefore, once inside the terminal area, all traffic is vectored to the same target speeds and altitudes, with little regard for aircraft type. Only near entry into the terminal area do traffic streams usually have speed or altitude differences large enough to warrant overtakes.

Second, an overtake must not demand excessive attention or controller-to-controller coordination. Rather, the overtake must be completed in a timely manner. If the overtake requires a long time to develop, the controller will need to focus extra attention on the situation in order to verify that the aircraft do not represent potential conflict.

Lastly, an overtake cannot require excessive aircraft vectoring. Regardless of delay savings, an overtake is avoided when it causes workload-intensive vectoring of either aircraft. For example, such vectoring would occur if no arrival slot exists ahead of the slower aircraft for the faster aircraft to occupy. In this situation, the controller would need to vector both aircraft - the faster aircraft to complete the overtake and the slower aircraft to achieve legal separation.

Merging In-trail Aircraft

Two aircraft are considered in-trail when they occupy the same trajectory segment with similar speeds and near-minimum legal separation. Similar to overtakes, splitting in-trail aircraft usually yields only minimal delay savings and substantial workload penalties.

For the controller, little effort is needed to keep such aircraft consecutive in the sequence all the way to the runway threshold. Therefore, the additional vectoring caused by merging another aircraft between them can significantly increase workload. For example, in order to merge an aircraft from a separate traffic stream between the in-trail aircraft, the controller needs to both vector the in-trail aircraft to gain an additional landing slot between them and vector the merging aircraft into that slot. Conversely, in order to place the aircraft behind the in-trail aircraft, the controller simply vectors it to that position.

Separating Stacked Aircraft

In the terminal airspace, near the final approach course, altitude often indicates an aircraft's place in the landing order. This is especially relevant for aircraft separated by altitude, or "stacked". While their current positions and ETAs to the runway threshold are nearly identical, the lower aircraft will be given preference. The controller's workload drives this decision. In order to sequence the higher aircraft ahead, the controller first needs to achieve minimum in-trail separation by vectoring the lower aircraft. Then, the controller needs to descend the higher aircraft and verify that no conflicts arise with the traffic below. Conversely, when the lower aircraft is sequenced ahead, the controller only needs to maintain minimum in-trail separation.

Equalizing Delay

The techniques previously described consist mainly of tactical methods used to sequence individual pairs of aircraft. It is also necessary to understand how controllers more generally sequence aircraft which are not in close proximity but who will be merging at some time in the future. Though the controller's cognitive process in making these longer term plans is not fully understood, a working model suggests that controllers attempt to ensure that aircraft occupying nearby slots in the landing sequence have similar levels of delay. A scheduler which is acceptable to controllers will try to equalize delay penalties to consecutive aircraft. In other words, a controller would be reluctant to implement an arrival plan which had base traffic intercepting the final approach course twenty miles from the runway threshold but downwind traffic intercepting only five miles from the runway threshold. Instead, controllers often identify a common "pivot" point along the final approach course where the different traffic streams will merge. This technique maintains comparable delay levels for all traffic streams, maximizes the controller's available options for vectoring aircraft, and thus increases the safety of the operation by decreasing his workload.

Criteria for Sequence Advisory Acceptability

It is also necessary to define qualitative measures of the acceptability of a system intended to provide sequence advisories to the air traffic controllers. Four criteria have been identified: controllability, speed of response, stability and flexibility.

Controllability of the Sequence Advisories

The arrival plan presented to the controllers must be achievable without the use of unnecessary or extreme vectors and the endurance of unacceptable workload levels. This is defined as the system's controllability. Implementation of a suggested arrival plan by the controller must not compromise the safety of the air traffic control system. Therefore, the arrival plan cannot, for the purpose of increasing delay savings, force aircraft into situations where their maneuverability or delay capacity are minimized. Moreover, the arrival plan cannot depart too drastically from the standard techniques used by the controller.[11]

Speed of Response of Sequence Advisories

The system must be quick to respond to controller decisions to change the relative order of aircraft in the arrival plan. No sequencing algorithm can be in agreement with all controllers 100% of the time, because there are many circumstances which the system cannot account for in its predictions. For example, a pilot may execute a clearance late or incorrectly, a pilot may vector around unknown weather, an aircraft may have unknown performance restrictions and individual controllers will have different styles. In light of these uncertainties, the system must be quick to recognize and respond when the controller is following a plan other than the recommended sequence.

Stability of Sequence Advisories

While a quick response time is desired, the system must also present a stable arrival plan to the controller. The relative orders of aircraft cannot be characterized by large or frequent jumps. Certainly, it is not possible to eliminate all sequence switches, especially for aircraft which are far from the airport or which are following non-standard routes. However, it is important that the number of these intrusive sequence number jumps be minimized. Controllers would like to be given a workable arrival plan, and then be allowed to execute the actions required to meet that plan.

Flexibility of Sequence Algorithm

Lastly, the system must be flexible. While the sequencing algorithm may be designed to be applicable to a generic terminal airspace, additional site-specific preferences may become necessary for the system to generate acceptable sequence advisories. The incorporation of these sequencing considerations into the knowledge base must be easy and straightforward. Similarly, the planning process of the automation system should be an extension of the controller's own planning process. An algorithm implemented in terms of criteria and decisions similar to those made by the human controller benefits from its ability to use the expert knowledge of those controllers in its design. Then, rather than optimizing a complex, non-intuitive cost function, the sequencing algorithm relates its decisions to the human controller's cognitive processes.

Passive FAST Sequencing Algorithm

This section describes the knowledge-based sequencing (KBS) algorithm which combines the trajectory based sequencing of the spatial constraint satisfaction method with fuzzy reasoning to produce acceptable sequence advisories for tactical control in the terminal area.[11]

Inputs to the Sequencing Algorithm

The KBS algorithm utilizes a 4D trajectory from each aircraft's current position to the runway threshold and an estimated time of arrival range for each aircraft to determine its relative order in the arrival plan.

Unlike time-based sequencing which schedules to a single point, the KBS algorithm considers each aircraft's complete trajectory to determine a satisfactory arrival plan. This requires an accurate 4D trajectory from each aircraft's most recently reported position to the runway threshold. These trajectories are produced by a trajectory generation engine which integrates point mass equations of motion along a 3D route in order to meet specific altitude, routing and airspeed constraints.[14] The horizontal route of the fastest trajectory represents the shortest path the aircraft is expected to take from its current position to the runway threshold, and the airspeed profile of the fastest trajectory represents the latest locations where required speed reductions will likely occur. The altitude profile approximates a standard terminal area profile by descending along fixed flight path angles to meet the altitude restrictions imposed at the final approach fix and runway threshold. The current aircraft state, winds aloft, atmospheric temperature and pressure profiles, aircraft flight characteristics and standard terminal area procedures are included in the integration of the equations of motion.

The time range over which the aircraft could arrive at the runway threshold is also used by the sequencing algorithm. This time range is computed by perturbing the fastest trajectory by its most extreme vector deviations and its earliest speed reductions. The estimated time of arrival range does not consider the influences of other traffic on the aircraft's route or airspeed and altitude profiles. Rather, the time range simply defines the upper and lower bounds on arrival time for the aircraft; this is the total delay capacity of the aircraft during normal operations.

Outputs from the Sequencing Algorithm

The essential output of the sequencing algorithm is the relative sequence of all aircraft in the arrival plan to their assigned runways. Sequence number advisories, indicating the relative order of aircraft destined to each runway, are displayed to the controller. As a result, it is possible to have aircraft assigned to different runways with the same sequence number advisory.

The sequence number advisories of aircraft throughout the terminal area are not decremented when an aircraft lands. Rather, after aircraft #1 lands, aircraft #2 becomes the next aircraft scheduled to land and so on until an adaptable maximum number is reached (typically 30). At this point, the sequence number advisories reset to a value of 1. By not decrementing the sequence numbers as aircraft land, a change in an aircraft's sequence number always indicates a change in that aircraft's relative position in the landing order.

The sequencing algorithm also computes a nominal scheduled time of arrival (STA) for each aircraft based upon the required separation behind the aircraft that it trails at the runway threshold. This nominal STA is defined as

$$STA_B^{AB} = \max(STA_A^{AB} + \bar{t}_{AB},\ TA_B^{earliest})$$

where, STA_A^{AB} and STA_B^{AB} are the STAs of Aircraft A and Aircraft B, respectively, given a relative sequence AB. $TA_B^{earliest}$ is the earliest time of arrival of Aircraft B and $\bar{t}_{AB}$ is the required separation time at the runway threshold.

The nominal STA does not account for additional delay used to resolve conflicts away from the runway threshold. Rather, it is used by the sequencing algorithm to compare estimates of the delay associated with different relative orders of aircraft.

Trajectory-Based Constraints

It was learned through extensive real-time simulations that to produce an acceptable arrival sequence it is necessary to consider all merges within the terminal area, not just the merge onto the final approach course. To accomplish this, the sequencing problem is broken into a network of common trajectory segments. These trajectory segments are first ordered and then merged to produce a consistent and acceptable sequence at the runway threshold.

Trajectory Segment Tree

A trajectory segment is the portion of a trajectory that falls within a defined segment of flight. Figure 2 shows several aircraft and their trajectories broken into trajectory segments, referred to as "long", "downwind", "base" and "final".

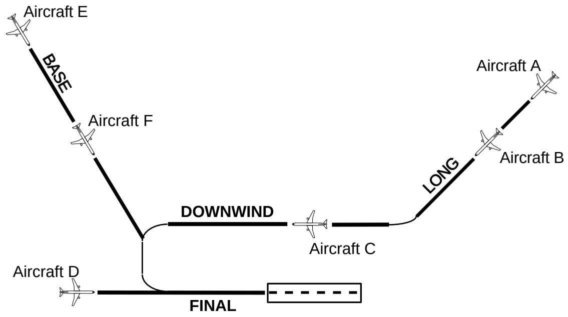

Fig. 2. Illustration of Merging Trajectory Segments

Determining the sequence of aircraft to a given runway consists of both ordering the aircraft on each trajectory segment and combining these relative orders into a consistent final sequence to that runway. Figure 2 depicts a situation where six aircraft merge together to land on the same runway. The trajectory segment tree shown in Figure 3 illustrates how this sequencing problem is broken into a series of smaller sequencing problems. The root of the tree represents the advised landing order. The branches of the tree symbolize merges that occur in the terminal airspace. These branches connect nodes which represent the relative sequence of aircraft on a particular trajectory segment.

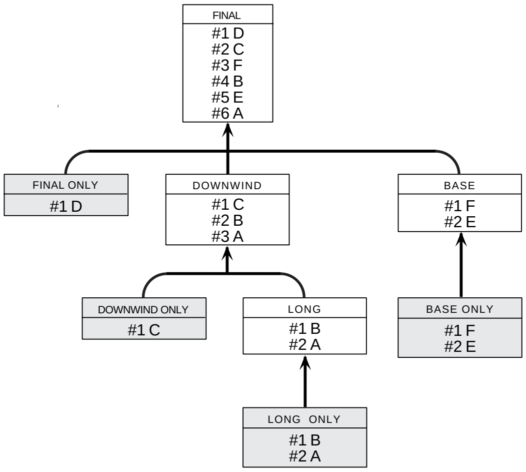

Fig. 3. Trajectory Segment Tree

The sequencing process begins at the leaves of the trajectory segment tree. A relative order is determined for the aircraft on each trajectory segment. This process of sequencing aircraft which share a common trajectory segment is referred to as "ordering." The ordering process considers many factors but its details are not needed to describe the trajectory-based sequencing methodology. The result is an ordered aircraft list for a particular leaf of the trajectory segment tree. This list represents the aircraft's relative order on its first trajectory segment.

Once the leaves are ordered, these leaves, and later branches, are repeatedly merged to create a relative order in each node. Thus, a node represents the relative order of aircraft on a particular trajectory segment which was formed by merging the earlier predictions of other leaves and branches. Like the ordering process, the merging process considers many criteria; the description of their details is deferred to a later section.

The merging process preserves the relative order of aircraft from preceding nodes and leaves. This constrains the number of possible sequences at the runway threshold by enforcing all sequence constraints encountered at earlier merge points of the trajectory.

Advantages of Trajectory-Based Constraints

This method of addressing the sequencing problem has several benefits. First, separate criteria can be specified for each trajectory segment which take into account specific methods controllers use to sequence aircraft on that segment, e.g. allowing or disallowing overtakes. Also, different criteria can be specified for the ordering and merging steps of the sequencing process. Therefore, aircraft already present on a trajectory segment can be sequenced differently from aircraft predicted to be on that trajectory segment at a future time.

Second, the relative order for each pair of aircraft is determined using information about when those aircraft are in close spatial proximity. The first set of relative orders is comprised of aircraft on a common trajectory segment - each aircraft's first trajectory segment. This set of decisions uses the current aircraft positions and short-term trajectory predictions. The next set of decisions combines the first set to produce the relative order on the next trajectory segment that the aircraft are approaching. These orders are determined from predictions made farther into the future. As the sequencing process continues, it is forced to use predictions that are farther into the future. Yet, at every merge, the sequencing process uses the earliest predicted information of that segment, hence the most accurate, that can be used to compare the aircraft.

Fuzzy Reasoning-Based Sequencing

This section details the knowledge base of the sequencing algorithm implemented by Passive FAST and successfully tested in the operational environment of Dallas-Fort Worth TRACON in mid-1996. The KBS algorithm combines trajectory-based sequencing with fuzzy reasoning to determine an acceptable arrival plan. The subsequent discussion assumes familiarity with fuzzy reasoning methods, specifically decision making based upon the fuzzification and defuzzification of linguistic propositions. These concepts and their inclusion in the KBS algorithm are discussed in detail in Reference 15. More general explanations of fuzzy set theory can also be found in References 16 and 17.

Spatial Sequence Constraints

Rather than sequence aircraft along each trajectory segment, the KBS algorithm sequences aircraft in larger groups called spatial constraint sets[‡]. A spatial constraint contains aircraft whose trajectories now or later will pass through the same section of airspace. There are three types of spatial constraints, GENERAL, FINAL and DEPENDENCY. For conciseness, only the specific sequencing criteria for GENERAL-type spatial constraints are presented in this paper. Refer to Reference 15 for an in-depth discussion of the sequencing criteria for FINAL-type and DEPENDENCY-type spatial constraints.

A GENERAL-type spatial constraint contains a set of aircraft from a particular trajectory segment, other than the FINAL trajectory segment. While a common trajectory segment is the primary attribute of these

---

[‡] Spatial Constraint Set is hereafter referred to as Spatial Constraint.

spatial constraints, other attributes are used to isolate a subset of aircraft from a single trajectory segment. For example, the spatial constraints, SHORT_LEFT_PROPS and SHORT_LEFT_JETS, subdivide the aircraft on the SHORT_LEFT trajectory segment.

Sequencing differences require FINAL trajectory segments to represent a separate type of spatial constraint. A FINAL-type spatial constraint is simply a specialized GENERAL-type spatial constraint that contains all aircraft from a single FINAL trajectory segment. Every arrival aircraft belongs to a single FINAL-type spatial constraint, and the aircraft's sequence number is its relative sequence on that spatial constraint.

Although two aircraft landing on different runways never physically merge, each can affect how the other is sequenced. Therefore, a DEPENDENCY-type spatial constraint links multiple FINAL-type spatial constraints in order to account for runway dependencies like staggered approaches to parallel runways or converging approaches to intersecting runways.

Fuzzy Set Antecedents

The KBS algorithm uses five fuzzy antecedents to build "if-then" propositions for its sequence decision-making logic of GENERAL-type spatial constraints. These antecedents evaluate numerous aircraft position and trajectory path distance quantities to determine an aircraft's levels of membership in the corresponding fuzzy sets. In order to be independent of airspace, distance values are normalized by the required separation distance. The normalized values of these quantities represent the number of landing slots. For example, a normalized distance value of 1.0 represents a savings of one full landing slot.

"<u>A is ahead of B at their FCTS:</u>" The primary criteria for ordering two aircraft on a GENERAL-type spatial constraint are the aircrafts' predicted positions at the first instance both belong to that spatial constraint. These positions are called First Common Time Steps (FCTS). The aircrafts' FCTS measure how far ahead or behind one aircraft is to another. Preference can be given to a FCFS ordering of the aircraft. The normalized separation at the FCTS, (NSD_{fcts}), is

$$NSD_{fcts} = \begin{Bmatrix} \dfrac{d^B_{fcts,ref} - d^A_{fcts,ref}}{\bar{d}_{AB}} & if\,(d^B_{fcts,ref} > d^A_{fcts,ref}) \\ 0 & if\,(d^B_{fcts,ref} \le d^A_{fcts,ref}) \end{Bmatrix}$$

where, $d_{fcts,ref}$ is the distance from an aircraft's FCTS to the spatial constraint's endpoint and $\bar{d}_{AB}$ is the required separation distance given the sequence AB.

A positive NSD_{fcts} indicates that Aircraft A is predicted to be closer to the runway threshold than Aircraft B at their FCTS. The membership function used to map the normalized separation at the FCTS to a proposition's firing strength is shown in Figure 4.

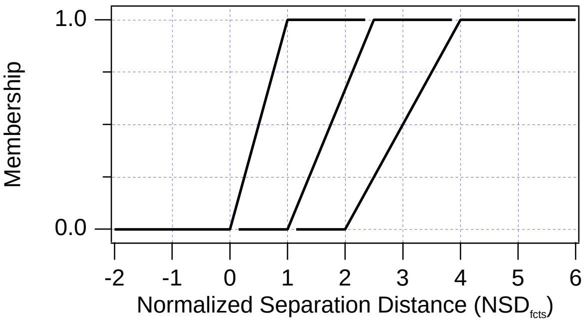

Fig. 4. Membership Functions for "Ahead at FCTS"

"<u>A is ahead of B at their current positions:</u>" The geometric order describes which aircraft is presently closest to the runway threshold along its predicted trajectory. This antecedent defines propositions which limit overtakes and give preference in the sequence to the aircrafts' current geometric order. The current normalized separation (NSD_{trk}) is

$$NSD_{trk} = \begin{Bmatrix} \dfrac{d^B_{trk,ref} - d^A_{trk,ref}}{\bar{d}_{AB}} & if\,(d^B_{trk,ref} > d^A_{trk,ref}) \\ 0 & if\,(d^B_{trk,ref} \le d^A_{trk,ref}) \end{Bmatrix}$$

where $d_{trk,ref}$ is the path distance from an aircraft's current position to its FCTS and $\bar{d}_{AB}$ is the required separation distance given the sequence of AB.

A positive NSD_{trk} indicates that Aircraft A is currently ahead of Aircraft B. The membership function used to map the current normalized separation to a proposition's firing strength is shown in Figure 5. For current normalized separation values less than 0.25, the aircraft are essentially considered tied, so the resulting membership is 0.0.

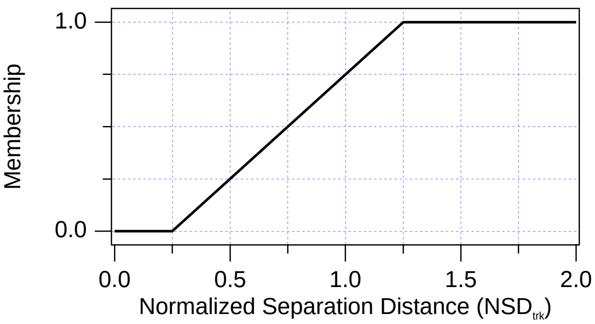

Fig. 5. Membership Function for "Currently Ahead"

"*A is faster than B at their FCTS:*" A ground speed difference at the aircrafts' FCTS is used to identify acceptable overtake situations. The membership function used to map a ground speed difference to a proposition's firing strength is shown in Figure 6. Since the error in computed ground speed is frequently more than 10 knots, ground speed differences less than 25 knots have a membership value of 0.

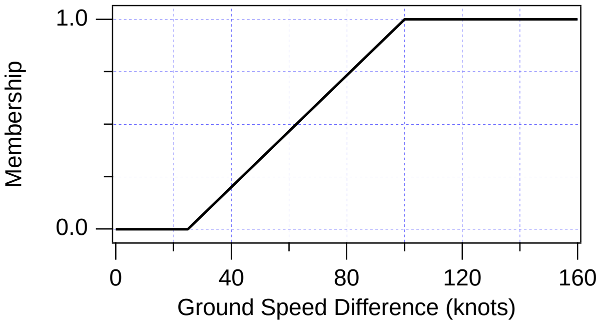

Fig. 6. Membership Function for "Faster at FCTS"

"*A is lower than B at their current positions:*" An altitude difference at the aircrafts' current radar track positions is used to define heuristics for ordering stacked aircraft. The membership function used to map the current altitude difference to a proposition's firing strength are shown in Figure 7.

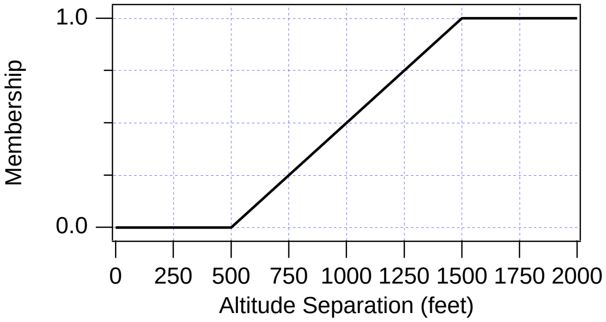

Fig. 7. Membership Function for "Currently Lower"

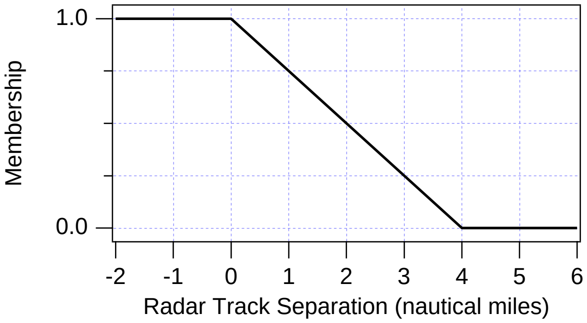

Fig. 8. Membership Function of "Close"

"*A and B are Close:*" The horizontal spatial separation of two aircraft is also used to determine if they are in close proximity. This antecedent is used in conjunction with the antecedent "Aircraft A is Faster than Aircraft B at their FCTS" to recognize a controller's decision to allow an overtake when none was initially being advised. The membership function used to map the current separation to a firing strength is shown in Figure 8.

Fuzzy Set Consequents

Fuzzy set consequents form the "then" part of an "if-then" relationship. The KBS algorithm defines a set of six standard consequents used in the expression of its fuzzy propositions. This set includes mathematical descriptions of marginally favored, slightly favored, somewhat favored, favored, quite favored and significantly favored. These linguistic labels and their associated fuzzy regions are illustrated in Figure 9.

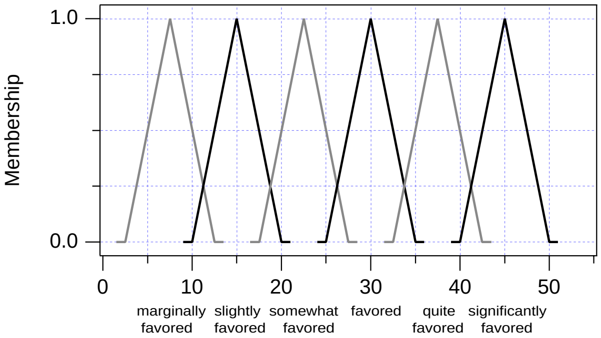

Fig. 9. Consequents Describing "FAVORED"

Early in the development of the fuzzy reasoning algorithm, each fuzzy proposition had its own consequent. Such complexity arose when the consequents were separately modified in each fuzzy proposition to fix specific deficiencies in the sequence advisories. Allowing a different consequent for each fuzzy proposition was later abandoned to improve maintainability. Instead, the consequent of each fuzzy proposition is chosen from the standard set. This bounds the number of permutations possible when the fuzzy propositions need modification.

By design, the consequents are symmetric. This attribute would not be appropriate for a fuzzy system whose output is a real-valued quantity. However, the output of this algorithm is simply an algebraic sign, positive for one relative sequence, negative for the other. Both the weight and the centroid of an asymmetric fuzzy region vary with membership value. Therefore, each rule's maximum strength relative to the other rules changes with membership value. This would needlessly increase the complexity of the KBS algorithm and decrease its maintainability.

Fuzzy Propositions for Ordering

Every aircraft belonging to the GENERAL-type spatial constraint is compared to every other aircraft belonging to that spatial constraint. Seven fuzzy propositions are evaluated in order to sequence each pair of aircraft. Since these seven propositions are independently evaluated for both aircraft of the pair, there are effectively fourteen different rules representing the decision making process. For two aircraft, "A" and "B," these fuzzy propositions are expressed as follows:

1. if (A is significantly ahead of B at their FCTS) then (A is significantly favored)
2. if (A is ahead of B at their FCTS) then (A is favored)
3. if (A is slightly ahead of B at their FCTS) then (A is slightly favored)
4. if (A is ahead of B at their current positions) then (A is slightly favored)
5. if (A is faster than B at their FCTS) then (A is marginally favored)
6. if (A is lower than B at their current positions) then (A is marginally favored)
7. if (A is close to B AND A is faster than B at their FCTS) then (A is slightly favored)

Aircraft occupy and thus are sequenced on GENERAL-type spatial constraints when they are farthest from the runway threshold. In this region, controllers, known as "feeder" controllers, are responsible for managing several traffic streams before transferring control of these aircraft to other controllers, known as "final" controllers. The final controllers are then responsible for merging this traffic onto the final approach courses. The feeder controllers must manage their traffic to supply multiple streams of aircraft which can be smoothly merged into a single stream. Therefore, feeder controllers execute a near FCFS sequence. Rules 1, 2 and 3 represent this preference by favoring the aircraft predicted to have the least path distance to fly to the runway threshold from the FCTS.

The few exceptions to a FCFS sequence occur when aircraft are in close proximity to each other. Therefore, several fuzzy propositions serve to constrain the problem by favoring a sequence based upon the aircrafts' current geometric relationships. This is accomplished by considering the aircraft's state at the current radar track. Rule 4 gives preference to the aircraft which has the least path distance to fly to the runway threshold from its current radar track location. Similarly, rule 5 gives preference to the lower aircraft, and rule 6 gives preference to the faster aircraft.

Finally, since ill-advised overtakes represent considerable workload for the controller, the system has been designed to be conservative in its prediction of overtakes. However, for the system's performance to remain acceptable, this approach was augmented to provide a quick response in situations where the initial prediction of no overtake was incorrect. Rule 7 represents logic intended to recognize and respond quickly to unadvised overtakes. If an overtake was not originally suggested by the system, but two aircraft are in close proximity at dissimilar ground speeds, rule 7 recognizes that an unadvised overtake is being executed by the controller and the aircraft are resequenced accordingly.

Fuzzy Propositions for Merging

Once the aircraft are ordered on their initial spatial constraints, the aircraft are repeatedly merged onto GENERAL-type spatial constraints closer to the runway threshold. This is analogous to traversing the network of spatial constraints by beginning at the leaves and ending at the root of the tree (Fig. 3).

The first aircraft of each spatial constraint immediately preceding the spatial constraint to be merged are compared to determine who will be next. Again, seven fuzzy propositions are evaluated for each aircraft in order to determine the relative order of each pair of aircraft. Like ordering, merging of GENERAL-type spatial constraints is well-described by a near-FCFS sequence. For this reason, the same decision making criteria were used in both situations. The aircraft chosen to be next is removed from the ordered list of its preceding spatial constraint, and appended to the ordered list of the merged spatial constraint. Moreover, the aircraft which immediately followed this chosen aircraft is now considered first from that preceding spatial constraint. The merging process continues by comparing this new set of aircraft first from the preceding spatial constraints. Again, that aircraft determined to be next is removed from its spatial constraint's ordered list, and appended to the merged spatial constraint's ordered list. The merging process is finished when the relative order of all aircraft on the merged spatial constraint has been determined.

Most importantly, the relative order of aircraft from each preceding spatial constraint is preserved in the relative order of aircraft on the merged spatial constraint. This approach is crucial for determining an acceptable final sequence of aircraft. It forces earlier decisions, made with near-term information, to constrain later decisions.

Operational Test Results

Following several thousand hours of real-time controller-in-the-loop simulations of the Passive FAST system, it was determined that an operational test was necessary to completely validate the results obtained in the laboratory.

A series of operational tests were conducted at the Dallas-Fort Worth TRACON during the period from January 1996 through July 1996. These tests included all major arrival rush periods, instrument flight rules (IFR) and visual flight rules (VFR), North and South Flow TRACON operations, two and three runway airport configurations and a variety of meteorological conditions.

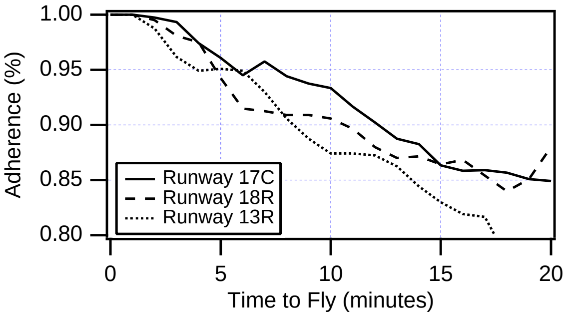

Fig. 10. Adherence to South Flow Sequence Advisories

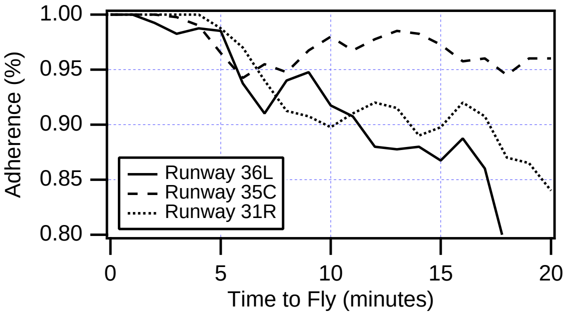

Fig. 11. Adherence to North Flow Sequence Advisories

A critical measure of the success of a sequencing algorithm is the degree to which the air traffic controllers adhere to the suggested arrival plan. Following the conclusion of the operational tests, several modifications were made to the sequencing algorithm. The algorithm documented in this paper represents this refinement of the algorithm used during the operational tests. These changes did not greatly increase the resultant adherence levels. Instead, they lessened the severity of unacceptable sequence advisories. Data collected during the operational tests were re-analyzed to quantify what fraction of the suggested arrival plans matched the actual aircraft landing orders. Figures 10 and 11 show the percentage of aircraft landing in the advised order as a function of time to fly for South Flow and North Flow airport configurations.

While these results seem to suggest a high level of adherence, these findings alone do not validate the algorithm. Controller comments during the operational test indicated that it is more important to guard against a poor sequence than to provide a perfect sequence. This is evident when a single ill-advised overtake during an hour-long, 100-aircraft arrival rush can make the difference between the system being judged acceptable or unacceptable. In the end, the controllers who evaluated Passive FAST during the operational test felt that any frustration caused by a rare ill-advised sequence advisory was greatly outweighed by the system’s benefits of decreased workload and increased throughput.

Conclusions

A fuzzy reasoning-based algorithm for sequencing arrival air traffic in the terminal area has been developed at the NASA Ames Research Center and operationally tested at the Dallas-Fort Worth TRACON. Although developed using several thousand hours of real-time controller-in-the-loop testing with DFW traffic scenarios and nearly thirty hours of operational testing at the DFW TRACON, the algorithm is designed to be generic, and thus, applicable to other terminal airspaces.

The sequencing algorithm performs trajectory-based planning by breaking the whole sequencing problem into sets of smaller short, medium and long term decisions. This design explicitly allows for special treatment of non-FCFS controller preferences which cannot be considered by time-based scheduling methods. It also permits separate sequencing preferences to be defined for trajectory segments throughout all phases of the flight from the extended terminal area to the runway threshold. Through the use of fuzzy-reasoning, the algorithm directly incorporates the experience of expert controllers in its design. Its knowledge base is appropriately defined in terms of fuzzy propositions since the problem of sequencing arrival aircraft in the terminal area is inherently difficult to define. Further, fuzzy propositions allow a more direct connection between the controller’s decision making process and the system’s.

Data and controller comments gathered during the operational test verify that the system generates an arrival plan that is acceptable to the air traffic controllers. These results show that the system’s advised landing order at the time the aircraft enter the terminal area was correct more than 90% of the time.

This advisory adherence steadily increases and converges to 100% as the aircraft gets closer to landing.

As a result of the successful operational test, Passive FAST will begin long-term operational testing at the DFW TRACON in July 1997.

References

[1] Erzberger H., Davis T.J., Green S.M., "Design of Center-TRACON Automation System,"ß Proceedings of the AGARD Guidance and Control Panel 56th Symposium on Machine Intelligence in Air Traffic Management, Berlin, Germany, 1993, pp. 11-2 – 11-12.

[2] Nedell W., Erzberger H., Neuman F., "The Traffic Management Advisor," Proceedings of the American Control Conference, San Diego, California, May 1990.

[3] Synnestvedt R.G., Swenson H., Erzberger H., "Scheduling Logic for Miles-In-Trail Traffic Management," NASA Technical Memorandum 4700, September 1995.

[4] Green S., Vivona R., "Field Evaluation of Descent Advisor Trajectory Prediction Accuracy," Proceedings of the AIAA Guidance, Navigation and Control Conference, San Diego, California, No. AIAA-96-3764-CP, July 1996.

[5] Davis T.J., Krzeczowski K.J., Bergh C., "The Final Approach Spacing Tool," Proceedings of the 13th IFAC Symposium on Automatic Control in Aerospace, Palo Alto, California, September 1994.

[6] Volckers U., "Arrival Planning and Sequencing with COMPAS-OP at the Frankfurt ATC-Center," Proceedings of the 1990 American Control Conference, San Diego, California, pp. 496-501.

[7] Garcia J., "MAESTRO - A Metering and Spacing Tool," Proceedings of the 1990 American Control Conference, San Diego, California, pp. 501-507.

[8] Brinton C.R., "An Implicit Enumeration Algorithm for Arrival Aircraft Scheduling," Presented at the 11th Digital Avionics Systems Conference, Seattle, Washington, October 1992.

[9] Erzberger H., "Design Principles and Algorithms for Automated Air Traffic Management," AGARD Lecture Series No. 200: Knowledge-Based Functions in Aerospace Systems, Madrid, Spain; Paris, France; San Francisco, USA, November 1995.

[10] Swenson H.N., Hoang T., Engelland S., Vincent D., Sanders T., Sanford B., Heere K., "Design and Operational Evaluations of the Traffic Management Advisor at the Fort Worth Air Route Traffic Control Center," To be published as a NASA Technical Memorandum.

[11] Bergh C.P., Krzeczowski K.J., Davis T.J., "TRACON Aircraft Arrival Planning and Optimization through Spatial Constraint Satisfaction," Air Traffic Control Quarterly, Volume 3, Number 2, pp. 117-138.

[12] Davis T.J., Isaacson D.R., Robinson J.E., den Braven W., Lee K.K., Sanford B., "Operational Test Results of the Passive Final Approach Spacing Tool," Presented at the 8th IFAC Symposium on Transportation Systems, Chania, Greece, June 1997.

[13] Isaacson D.R., "Knowledge Based Runway Assignment for Arrival Aircraft in the Terminal Area," To be presented at the AIAA Guidance, Navigation and Control Conference, New Orleans, Louisiana, August 1997.

[14] Erzberger H., Tobias L., "A Time-Based Concept for Terminal Area Traffic Management," Proceedings of the 1986 AGARD Conference, No. 410 on Efficient Conduct of Individual Flights and Air Traffic, Brussels, Belgium, 1986, pp. 52-1 – 52-14.

[15] Robinson J.E., Davis T.J., Isaacson D.R., "Fuzzy Reasoning-Based Sequencing of Arrival Aircraft in the Terminal Area," To be published as a NASA Technical Memorandum, October 1997.

[16] Yager R.E., Zadeh L.A. Eds., Introduction to Fuzzy Logic Applications in Intelligent Systems, Kluwer Academic Publishers, Norwell, MA, 1992.

[17] Cox E. Ed., The Fuzzy Systems Handbook, Academic Press Professional, New York, 1994.