Design and Development of the En Route Descent Advisor (EDA) for Conflict-Free Arrival Metering

Richard A. Coppenbarger*
NASA Ames Research Center, Moffett Field, CA, 94035

and

Richard Lanier†
Federal Aviation Administration, Moffett Field, CA, 94035

and

Doug Sweet‡ and Susan Dorsky§
Seagull Technology, Campbell, CA, 95008

This paper describes the fundamental design and development of a new decision-support tool for time-based arrival metering, referred to as the En Route Descent Advisor (EDA). This automation is being developed at NASA Ames Research Center as a component of the Center-TRACON Automation System (CTAS). EDA computes advisories for the air-traffic controller to help deliver aircraft to an arrival-metering fix in conformance with a scheduled time-of-arrival constraint, while preventing separation conflicts with other aircraft along the arrival trajectory. The CTAS Traffic Management Advisor (TMA), fielded at numerous facilities in the United States under the FAA Free Flight Phase 1 initiative, computes the scheduled arrival times at the metering fix that EDA targets for optimal throughput into the terminal airspace. By computing strategic maneuver advisories based on speed, altitude, and heading to meet arrival time with a high degree of precision, EDA has the potential to significantly extend the capacity, efficiency, and workload benefits already attributed to TMA operations. The paper first describes the underlying trajectory process, followed by a description of the core meet-time and conflict resolution algorithms employed by EDA. The key aspects of the user interface are then described, which allow controllers to receive, evaluate, and accept EDA advisories at their discretion. Finally, a closed-loop simulation system for evaluating EDA in the laboratory is described, along with key findings obtained during a series of preliminary controller-in-the-loop simulation experiments.

I. Introduction

THE benefits of Time-Based Metering (TBM) for achieving arrival capacity objectives have been well established in recent years. This is exemplified by the success of the Traffic Management Advisor (TMA),[1] developed as a component of the Center-TRACON Automation System (CTAS). TMA computes the optimal arrival schedule and sequence for aircraft in transition from en route to terminal airspace, and has been deployed at six Air-Route-Traffic-Control Centers (ARTCCs) in the United States under the FAA's Free Flight Phase 1 program. This deployment builds upon extensive prototype testing at the Ft. Worth ARTCC during the late 1990s. Recently, a significant effort has been undertaken to adapt TMA capabilities to the northeast corridor of the U.S, where arrival flows into the terminal area are managed from multiple feeder ARTCC facilities. This capability is referred to as

* Aerospace Engineer, Automation Concepts Research Branch, AIAA Senior Member
† Engineering Research Psychologist, FAA Liason Office
‡ Senior Systems Engineer, Seagull Technology Inc., AIAA Member
§ Senior Software Engineer, Seagull Technology Inc.

Multi-Center TMA (or MC-TMA).[2] TMA is designed for use by both Traffic Management Coordinators (TMCs) and sector controllers. TMCs use TMA information, presented in the form of timelines and load graphs, to strategically predict and plan arrival traffic flows. Sector controllers receive TMA arrival time information in the form of metering lists, presented directly on their plan-view radar display during TBM operations.

Although TMA provides controllers with arrival time targets for efficient metering, it does not provide the actual maneuver solutions required to achieve the prescribed, metering-conformant, traffic flows. In today's environment, controllers must rely entirely on their skill and judgment to bring aircraft into conformance with metering constraints while simultaneously guarding against separation conflicts. This is a difficult four-dimensional traffic management problem that is complicated by the convergent nature of arrival traffic. For this reason, a large number of corrective clearance instructions are often required to achieve safe and effective TBM, resulting in high levels of workload for the air-traffic controller. Under very heavy traffic conditions, the theoretical capacity benefits of TMA may even be compromised by the high workload required to "fill" each arrival slot in the TBM schedule. In addition, fuel efficiency for the airspace user is compromised by the frequent and aggressive maneuvering that commonly results from metering actions that are planned in a hurried, tactical manner due to a lack of supporting automation. For example, TBM actions in today's environment often require aircraft to execute inefficient maneuvers, such as step-down altitude descents, excessive lateral vectoring, and airborne holding. To address these problems and enable more effective and efficient TBM operations, a new CTAS tool, referred to as the En Route Descent Advisor (EDA), is being developed at NASA Ames Research Center.

EDA is intended for the ARTCC sector controller working at the Radar position (i.e. "R side"). EDA provides efficient maneuver advisories for bringing aircraft into conformance with their TMA-computed Scheduled Time-of-Arrival (STA) constraint at the meter fix, while simultaneously avoiding separation conflicts along the descent trajectory. Under typical TBM operations, solving this constrained "meet-time" problem requires efficiently delaying an aircraft in flight, using a combination of strategic speed, altitude, and heading maneuvers.[3]

Conceptually, EDA is deep rooted within the CTAS project at NASA Ames Research Center. Initial work on automated arrival metering began in the late 1980s.[4,5] This work led to the development of a pre-cursor to EDA known as the Descent Advisor (DA). A number of important simulation studies were conducted with DA in the early 1990s, followed by a series of field evaluations at the Denver ARTCC.[6,7] These early activities with DA focused on validating many of the core trajectory prediction and meet-time algorithms now incorporated into EDA. The current EDA effort, however, improves upon the original DA prototype in several critical areas: 1) additional, more powerful, meet-time modes involving speed and path-stretching have been introduced; 2) a realistic controller interface, absent in DA, has been implemented that allows controllers to interact with EDA advisories prior to acceptance; and 3) the current EDA prototype has been fully integrated into the modern CTAS software architecture, which includes an interface to TMA for receiving STA targets for metering.

The purpose of this paper is to describe the fundamental design and operation of EDA, associated with its recent development as a research prototype within CTAS. The underlying trajectory prediction process that supports EDA is first summarized. The core meet-time and conflict resolution algorithms are then described, along with the fundamentals of the user-interface for enabling controllers to receive, evaluate, and accept advisories. A closed-loop simulation system for evaluating EDA in the laboratory is then discussed, along with a number of important findings discerned from preliminary controller-in-the-loop simulation experiments.

II. General Solution Methodology

The overall process for computing conflict-free, meet-time solutions within EDA is illustrated in Fig. 1. This entire process is executed once over each 12-second computational cycle, defined to correspond with the receipt of updated Center radar track and flight-plan data from the FAA Host computer. Under a heavy arrival "rush", EDA might be required to perform this process for as many as 20 inbound aircraft at a time, within a given sector of en route airspace. As shown in Fig. 1, EDA first attempts to compute a solution that satisfies just the arrival time constraint at the meter fix, prior to any direct conflict resolution action. This is done in order to take advantage of the spacing between arrivals that occurs once aircraft are placed onto trajectories that meet their TMA-computed STAs at the meter fix. The temporal separation that TMA provides at the meter fix frequently prevents conflicts between arrivals that would otherwise develop along inbound trajectories.

Once the basic meet-time process is completed, conflict detection is performed for each aircraft along its predicted trajectory to the meter fix, in order to identify any remaining separation conflicts. In the event that conflicts remain, a separate conflict resolution process is invoked that attempts to resolve them on an aircraft-by-aircraft basis.

The following sections describe the basic sub-processes identified in Fig. 1, involved in the computation a complete EDA solution and subsequent advisory.

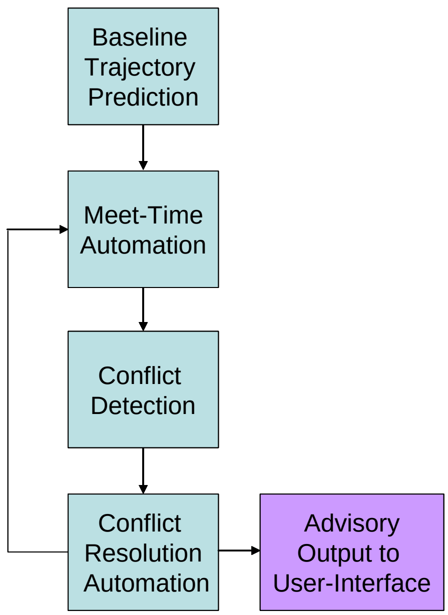

Figure 1. General EDA Solution Process

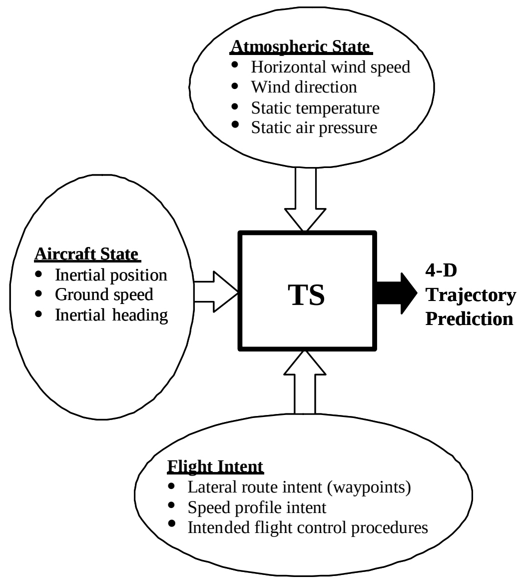

Figure 2. Trajectory Prediction Input/Output

III. Trajectory Prediction

A. Overview

Fundamental to EDA is the ability to accurately compute the predicted location (x, y, z) of an aircraft as a function of time (t), from its current position through to the arrival meter fix located at the TRACON boundary. This four-dimensional trajectory prediction capability is used not only to detect conflicts and compute the Estimated Time-of Arrival (ETA) at the meter fix, but it's also the key computational engine for iteratively computing EDA's conflict-free, meet-time solutions.

Trajectory prediction is performed by the CTAS Trajectory Synthesizer (TS) process[8]. For the purpose of EDA, the TS prediction is strategic in nature, involving look-ahead time horizons of up to 40 minutes, depending on the proximity of the aircraft to the arrival meter fix. The prediction is computed by integrating the basic point-mass aircraft equations of motion along the intended route of flight. As illustrated in Fig. 2, inputs to this process include: 1) accurate aerodynamic and propulsion models for each aircraft type, 2) current position and velocity state based on the latest radar track data, 3) intended route and speed profile obtained from filed flight plans and nominal airline preferences, 4) en route winds and atmospheric information obtained from sophisticated weather models, and 5) all known speed and altitude constraints that apply to the flight en route to the meter fix. The TS re-computes this prediction once every 12-seconds, upon the receipt of new radar track data from the FAA Center Host computer.

As described in detail in Ref. 6, the TS must generate a prediction that satisfies crossing conditions at the meter fix. These crossing conditions stipulate the altitude and maximum allowable airspeed for TRACON entry. Although meter-fix crossing conditions vary slightly between facilities, FAA regulations typically require aircraft to cross the fix at a barometric altitude of 10,000 ft above-sea-level, with a Calibrated Airspeed (CAS) of 250 kt or less.

To simplify the computational process, the TS decouples the lateral and vertical trajectory dynamics. An approximate lateral trajectory is first computed that turns "inside" the waypoints identified in the current flight plan; (the last of which is the meter fix itself). A vertical trajectory is then generated along this path based on the thrust, airspeed, and flight-path-angle parameters intended for each flight segment. For steady-state flight, knowing how any two of these parameters is varied as a function of time, while satisfying crossing conditions, is sufficient for the TS to define the vertical descent trajectory and associated Top-of-Descent (TOD) point.

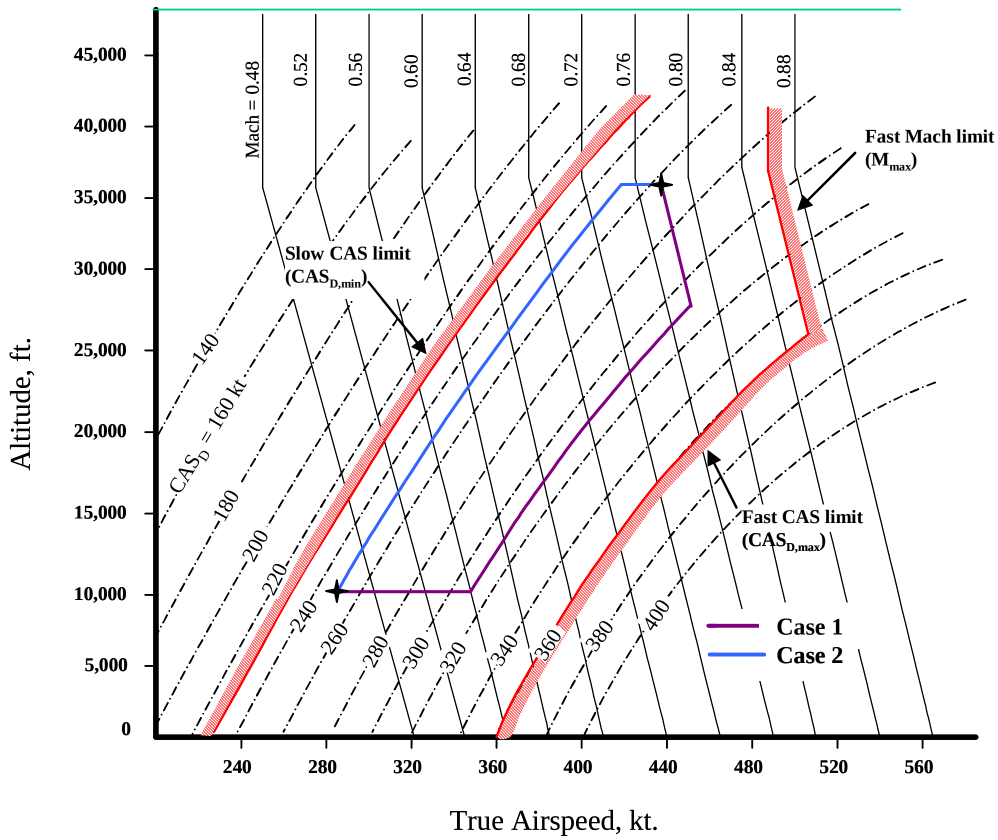

Figure 3. Typical Descent Speed Envelope for a Jet-Transport Aircraft with Examples of CAS Acceleration and Deceleration Maneuvers

B. Speed Transitions

For the jet transports that EDA is currently designed to accommodate, descents are assumed to be conducted at an idle-thrust power setting.[**] As described previously, the descent trajectory then becomes a sole function of the intended calibrated airspeed to be maintained during the descent to the meter fix (CAS_D). To compute an initial trajectory prediction prior to any action by EDA, the TS builds a trajectory by assuming a nominal, constant, descent airspeed ($CAS_{D,nom}$) that is stored within CTAS as a function of aircraft type and airline company preference[9]

Although the descent trajectory is defined by CAS_D, the correct procedure for transitioning to CAS_D from cruise conditions is dependent on the final cruise altitude, h_C, and whether an airspeed acceleration or deceleration is required. The reason for this is that airspeed accelerations that occur at high altitude (typically above 27,000 ft.) are potentially constrained by the maximum operational Mach number (M_{max}), in order to prevent undesirable sonic effects on the airframe/engine. Values for M_{max} are stored within CTAS as a function of aircraft type. This M_{max} constraint is illustrated in Fig. 3, which shows lines of constant Mach and CAS on a plot of altitude vs. true airspeed (TAS), under standard atmosphere conditions. As shown in Fig. 3, the operational speed envelope during a descent is constrained on the slow end by the minimum descent CAS ($CAS_{D,min}$) and by either maximum descent CAS ($CAS_{D,max}$) or M_{max} on the fast end, depending on altitude.

[**] The idle thrust descent represents the preferred, minimum-fuel, procedure for jet transports.

In building a descent trajectory, the TS must decide how to model the speed transition from cruise to descent. In the case of an airspeed acceleration where the descent CAS (CAS_D) is greater than the cruise CAS (CAS_C), the TS builds a trajectory with a final cruise segment conducted at the cruise Mach number, followed by a descent at that Mach number to an altitude where CAS_D is captured, followed by a final descent to the meter-fix altitude at constant CAS_D. This type of trajectory, represented by Case 1 in Fig. 3, is generated in order to safeguard against exceeding operational Mach limits during acceleration maneuvers at high altitude. In the case of an airspeed deceleration where CAS_D is less than CAS_C, the TS builds a trajectory that involves a level-flight deceleration to CAS_D at idle thrust, followed by a descent to the meter-fix altitude at constant CAS_D. This type of trajectory is exemplified by Case 2 in Fig. 3. Regardless of trajectory type, any final acceleration/deceleration required to satisfy the meter-fix crossing restrictions is assumed to occur in level flight at the meter-fix crossing altitude.

It's important to note that the method by which speed transitions are modeled by the TS is consistent with how similar transitions are modeled and flown by a modern aircraft Flight Management System (FMS). As described later in this paper, this congruency between EDA and FMS trajectory modeling is leveraged in order to simplify the required advisory instructions between the controller and flight-deck.

C. Active vs. Provisional Trajectories

To facilitate further discussion of the EDA automation and user interface, the two basic types of trajectory predictions – *active* and *provisional* - will first be defined. An *active* trajectory is defined as the best prediction of where an aircraft will be over the time horizon of interest, using everything that is currently known and accepted regarding that flight. Flight-intent information used to compute the active trajectory is based on the filed flight plan, stored speed preferences, and controller inputs that have already been entered into the FAA Host computer and/or CTAS. Updates to an aircraft's flight intent are based upon a controller's communicated clearance instructions regarding speed, altitude, and routing. In contrast, a *provisional* trajectory is one that has been generated by EDA for the purpose of solving a given meet-time and/or conflict resolution problem; i.e., it represents a trial-plan trajectory generated by the EDA automation, from which advisories are extracted. The active trajectory is replaced by the provisional trajectory *providing* that the controller accepts the corresponding advisories, as described later in this paper.

IV. Meet-Time Automation

A. Overview

A meet-time problem exists whenever an aircraft's current predicted ETA at the meter fix is out of conformance with its corresponding STA computed by TMA for efficient throughput. In order to prevent EDA from issuing pre-mature advisories based on a meter-fix schedule that hasn't yet stabilized, only STAs that are not subject to further change by the TMA scheduler are sent to EDA for advisory computation; such STAs are referred to as "frozen". The associated freeze horizon can be specified within TMA as either a time or distance threshold. For jet arrival traffic it is typically set to either 20 minutes or 130 n.mi from the targeted meter fix.[††]

The general meet-time problem is defined by: 1) the current position and velocity state of the aircraft, 2) the STA target at the meter fix, and 3) crossing conditions at the meter fix, required for TRACON entry. The meter fix targeted by TMA and EDA is the entry point into terminal airspace, located at the intersection of the Standard Terminal Arrival Route (STAR) and the TRACON boundary. The current EDA prototype attempts to solve this meet-time problem using speed and heading as maneuver Degrees-of-Freedom (DOFs). In order to conserve fuel and minimize operational complexity in the airspace, EDA attempts to first resolve a given meet-time problem with speed changes alone. In the event that the required delay is too large to be absorbed with just speed, lateral routing changes that stretch the path to the meter fix are prescribed.

B. Speed Modes

Speed is typically used to absorb delays of up to 4 minutes, depending on the available airspace over which the speed change will be effective. The solution is based on identifying a unique cruise speed, descent speed, or combination of the two that will result in a provisional trajectory with an ETA at the meter fix that conforms to the required STA. In all cases, EDA attempts to identify a specific airspeed profile that should be maintained over the remaining cruise and/or descent flight segments. The solution strategy depends on the one-to-one mapping of

[††] The magnitude of the TMA freeze horizon is subject to air-traffic facility preference. Although larger freeze horizons allow a wider range of EDA solutions, it results in a higher probability of the need to reschedule due to system uncertianties, such as aircraft that "pop up" upon departure from airports within the freeze horizon.

selected airspeed profile to meter-fix arrival time, described previously. EDA can be pre-configured by the controller to develop solutions based on three distinct speed modes, referred to as *Descent-Only*, *Cruise-Only*, and *Cruise-Equals-Descent*. These modes are described separately as follows.

1. Descent Only Mode

In this mode, EDA employs descent speed alone as the maneuver DOF for achieving the desired arrival time, and is typically effective for absorbing delays of up to 3 minutes in duration, depending on initial conditions the aircraft's operational speed envelope. In this mode, the aircraft is assumed to complete the cruise portion of its flight while maintaining its current cruise airspeed, and then transition to the advised descent airspeed, $CAS_{D,adv}$, for the descent to the meter fix. The available solution space for achieving arrival times with $CAS_{D,adv}$ in Descent-Only mode is therefore defined as lying between $CAS_{D,Max}$ and $CAS_{D,Min}$. This solution space is shown in Fig. 4, along with an illustration of a Descent-Only vertical profile. As discussed previously, the implied procedure associated with the transition regions in Fig. 4 is dependent on the relationship between initial cruise conditions, advised descent speed, and crossing restrictions at the meter fix.[‡‡] The correct speed transition procedure is selected automatically by the TS when building a trajectory for a given descent speed. For this reason, the iteration process within EDA takes place solely in the variable $CAS_{D,adv}$. Similarly, the output to the controller consists solely of this advised descent airspeed. Since the TOD is dependent on $CAS_{D,\ adv}$, it does not need to be given as a component of the EDA advisory as long as it is assumed that aircraft are operating with a FMS that is capable of computing a TOD based on descent speed that is consistent with that computed by EDA. This assumption, which depends on modeling similarities between CTAS and the FMS, not only simplifies the EDA advisory given to controllers, but also simplifies the resulting clearance instructions delivered to the flight deck.

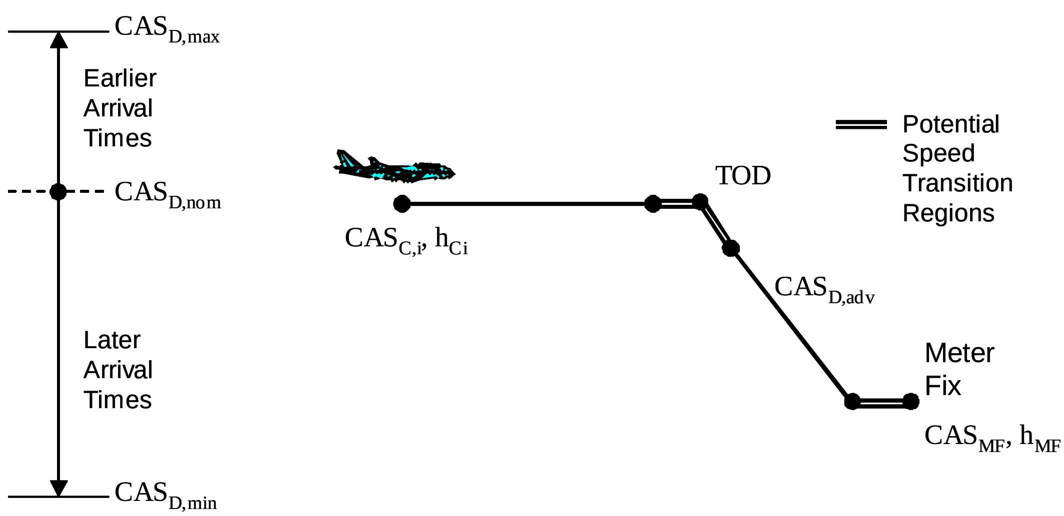

Figure 4. Descent-Only Solution Space and Illustrative Vertical Trajectory

2. Cruise-Only Mode

In this mode, EDA employs cruise speed alone as the maneuver DOF for achieving the desired arrival time, and is typically effective at absorbing delays of up to 2 minutes in duration, depending on initial conditions, speed envelope, and available airspace within the sector in which EDA is deployed. Here, the aircraft is assumed to transition to an EDA-advised cruise airspeed, $CAS_{C,adv}$, for completing the remaining cruise portion of the flight, prior to TOD. The CAS to be targeted during the descent, however, is considered unchanged from the nominal descent CAS determined by company/pilot preference, i.e., $CAS_{D,nom}$. The relevant solution space for the Cruise-Only mode is shown in Fig. 5, along with a representative vertical trajectory profile. As shown in Fig. 5, the solution space is bounded on the slow end by $CAS_{C,min}$, and on the fast end by either $CAS_{C,max}$ or the CAS equivalent to M_{max}

[‡‡] Note that airspeed accelerations from cruise conditions to descent conditions will still result in later arrival times as long as $CAS_{C,i} < CAS_{D,\ adv} < CAS_{D,\ nom}$.

at cruise altitude, whichever is lowest. Since the correct speed transition procedures and fast CAS limit are selected automatically by the TS when building a trajectory for a given cruise speed, the iteration process within EDA takes place solely in the variable $CAS_{C,adv}$. It should be noted that the ability of this mode to absorb delay is dependent on the distance remaining in cruise prior to TOD, at the time the advisory is computed. The output provided by EDA in this mode consists solely of $CAS_{C,adv}$. Alternatively, this can be expressed as an equivalent Mach number at higher altitudes, if desired by the controller.

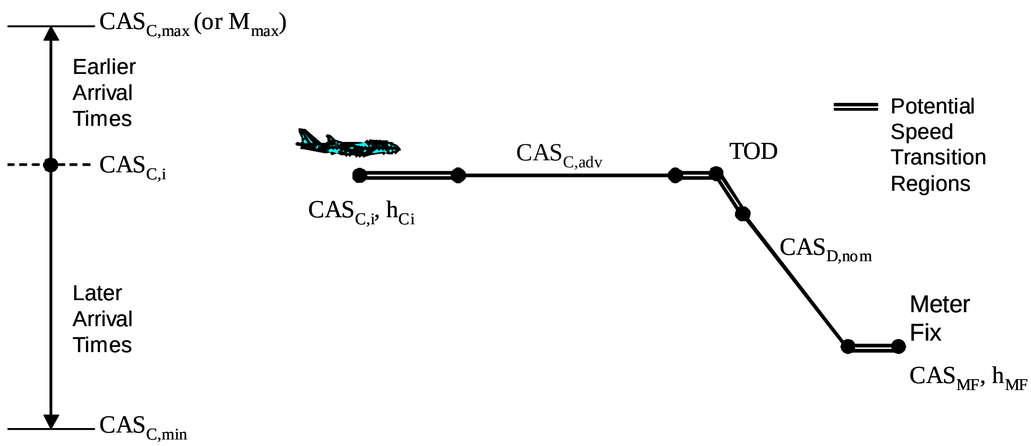

Figure 5. Cruise-Only Solution Space and Illustrative Vertical Trajectory

3. Cruise-Equals-Descent Mode

In this mode, EDA computes a combination of cruise and descent airspeed advisories, $CAS_{C,\ adv}$ and $CAS_{D,\ adv}$, which will satisfy the arrival time constraint at the meter fix. Depending on available airspace, initial conditions, and the aircraft's operational speed envelope, this mode is effective at absorbing delays of up to 4 minutes in duration. In order to minimize the number of speed change maneuvers required by the flight crew, the Cruise-Equals-Descent mode attempts to identify solutions where $CAS_{C,adv}$ and $CAS_{D,adv}$ can be made identical. This allows a single airspeed target to be maintained from cruise all the way to the bottom-of-descent at the meter fix altitude. An illustration of a vertical trajectory associated with the Cruise-Equals Descent mode is shown in Fig. 6.

The complete solution space for the Cruise-Equals-Descent mode is shown in Fig. 7, which shows the relationship between $CAS_{C,adv}$ and $CAS_{D,adv}$ for solving meet-time problems requiring both earlier and later arrival times. As represented in Fig. 7 by Case A and Case B, there are two variations of the $CAS_{C,adv}$ vs. $CAS_{D,adv}$ solution space, depending on whether the initial cruise speed, $CAS_{C,i}$, is lower or higher that the nominal descent speed, $CAS_{D,nom}$. The reason why $CAS_{C,adv}$ and $CAS_{D,adv}$ are not strictly equal over the entire solution space is to accommodate two important operational considerations: 1) the pilot-preferred procedure for absorbing small delays is to vary either cruise speed alone or descent speed alone;[10] and 2) the range of operationally feasible descent speeds is typically larger than the range operationally feasible cruise speeds, for a given aircraft type.

The former consideration is addressed by the vertical and horizontal regions at the center of the solution space in Fig. 7 where cruise and descent speed advisories are varied independently of one another for solving small meet-time problems. The decision of whether to use cruise or descent speed to solve a minor meet-time problem is determined by whether the solution space is governed by Case A or Case B, which is dependent on initial aircraft conditions as described above. In this manner, the Cruise-Equals-Descent mode exhibits behavior similar to that of the Descent-Only or Cruise-Only mode for minor meet-time problems.

To accommodate the second operational consideration mentioned above, it can be seen in Fig. 7 that $CAS_{D,adv}$ is varied over a wider range than $CAS_{C,adv}$. This results in large meet-time problems (i.e., those requiring earlier or later arrival times at the extreme limits of what can be achieved with the Cruise-Equals-Descent mode) being solved by advisories where $CAS_{D,adv}$ and $CAS_{C,adv}$ are not equal.

To facilitate iteration, the solution space in Fig. 7 is collapsed into a single variable upon which arrival time at the meter fix is a monotonic function. This is accomplished with the simple transformation: $CAS_{CD,adv} = CAS_{D,adv} + CAS_{C,adv}$. This transformation, together with the fact that the correct speed transition procedures are selected automatically by the TS in building trajectories, allows the iteration to proceed in the single, dummy, variable

$CAS_{CD,adv}$. Once a solution is found, $CAS_{CD,adv}$ is decomposed into its component cruise and descent speed advisories, $CAS_{C,adv}$ and $CAS_{D,adv}$, which are provided as output to the controller. This decomposition is accomplished by first computing the arrival times associated with the break-points in the $CAS_{CD,adv}$ solution space that correspond to changes in the relationship between $CAS_{C,adv}$ and $CAS_{D,adv}$, as previously described.

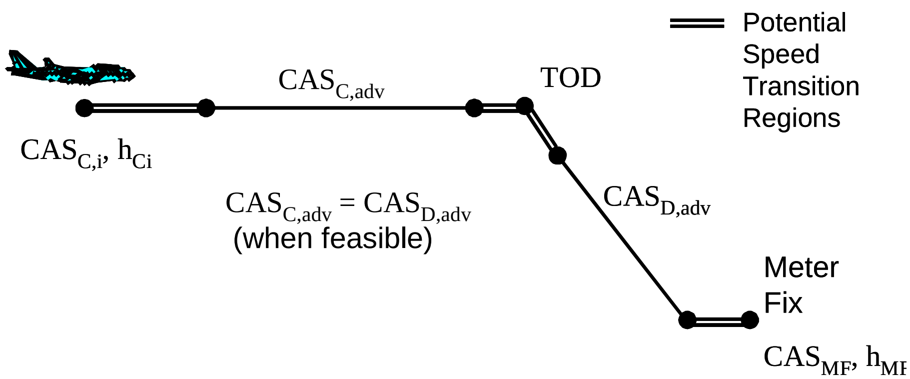

Figure 6. Illustrative Trajectory for the Cruise-Equals-Descent Mode

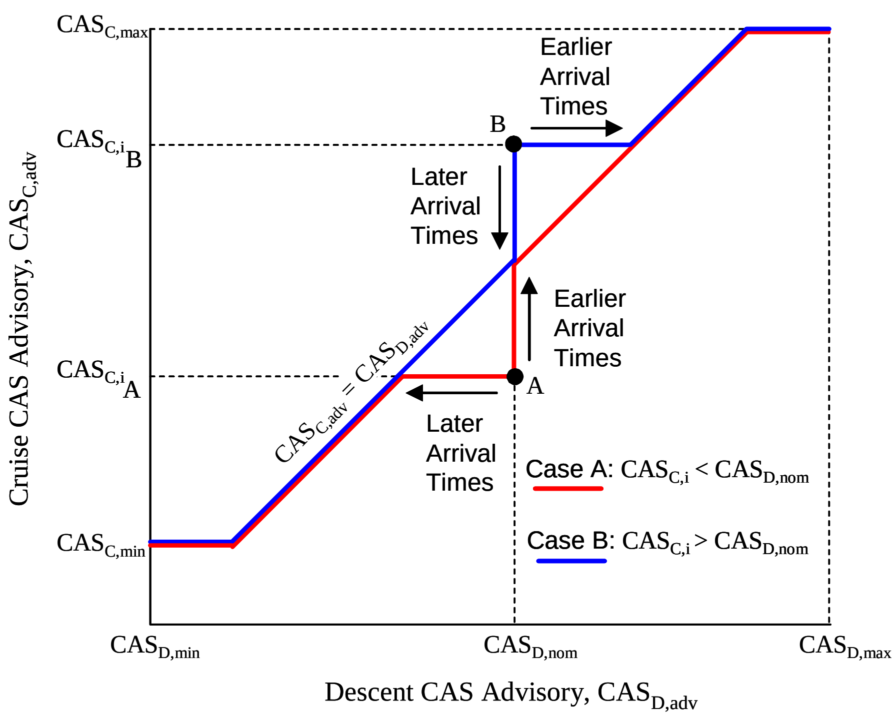

Figure 7. Solution Space for the Cruise-Equals-Descent Mode

C. Path-Stretch Mode

The EDA Path-Stretch mode was introduced to solve meet-time problems that cannot be effectively resolved with the speed-based maneuvers described above. In this mode, EDA advises a simple "dog-leg" maneuver that stretches the lateral flight path en route to the meter fix. In the current design, the Path-Stretch mode is only made available once the delay absorption capability of the pre-selected speed mode has been exhausted. For example, if the controller has chosen Cruise-Equals-Descent as the default meet-time mode, Path-Stretch is activated only at the slow end of the solution space in Fig. 7; i.e., where the speed profile is represented by $CAS_{C,adv} = CAS_{C,min}$, and $CAS_{D,\ adv} = CAS_{D,min}$. Depending on the underlying speed mode, initial aircraft conditions, and the amount of operationally-available airspace, Path-Stretch is typically applied to meet-time problems requiring 4 to 10 minutes of delay absorption. In the current EDA prototype, Path-Stretch solutions are presented as an option to the controller once the underlying speed mode has been exhausted, as described in the User Interface section of this paper.

The general Path-Stretch solution space and sample trajectory are illustrated in Fig. 8. The solution involves establishing the aircraft on an outbound heading that is off-set from the current route of flight by a fixed angle, and identifying the point at which the aircraft should be turned back towards the original flight plan to absorb the required delay at the meter fix. In the current design, the waypoint at which the stretched path rejoins the original flight plan is defined as the meter fix itself. Since the controller pre-selects the outbound heading off-set (as described later in this paper) the arrival time at the meter fix becomes a sole function of the time-of-flight along the stretched trajectory, prior to turn-back. As shown in Fig. 8, the solution space is bounded by trajectories corresponding to the minimum and maximum turn-back time, $t_{TB,min}$, and $t_{TB,max}$. The value for $t_{TB,min}$ is currently set to a constant value that allows the aircraft to be stabilized along the outbound heading prior to initiating the turn-back maneuver. The value for $t_{TB,max}$ is calculated dynamically as a function of speed and aircraft bank angle in order to prevent an excessive turn-back maneuver from occurring close to the meter fix. The solution trajectory, calculated between these minimum and maximum limits, is translated into an advisory output that conveys 1) the wind-corrected magnetic heading to be flown along the outbound leg of the stretched route, 2) the turn-back point (specified as either time or distance), and 3) the return heading (or waypoint name) associated with rejoining the original flight plan route.

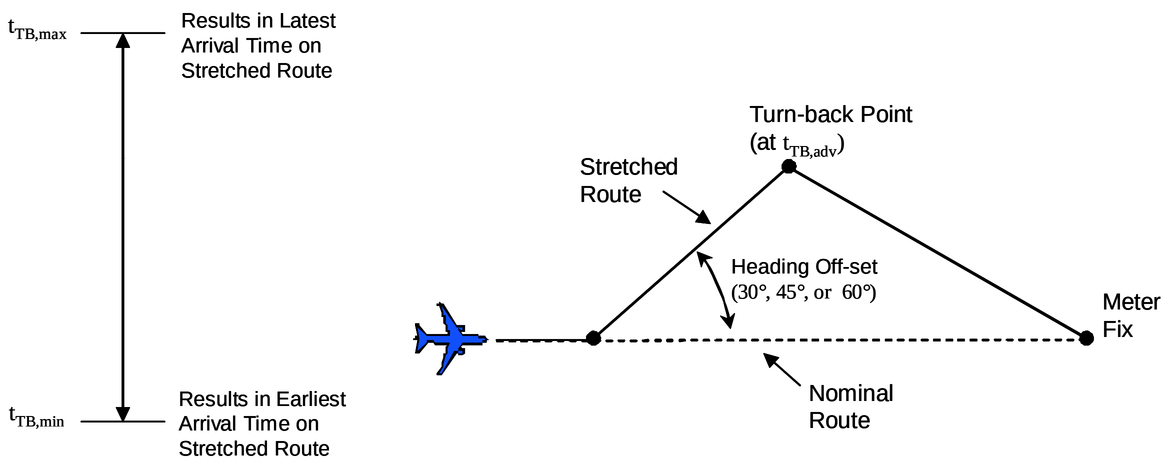

Figure 8. Path-Stretch Solution Space and Illustrative Lateral Trajectory

D. General Meet-Time Computation Method

Regardless of the meet-time mode, EDA employs a common approach for computing solutions. EDA begins by calculating three trajectories of interest to the problem. The first is the nominal trajectory (NOM), which is based on pilot/company intent preferences, as previously described. The second is referred to as the FAST trajectory, which represents the trajectory with the earliest achievable ETA using the iteration DOF pertaining to the meet-time mode in use. The third is referred to as the SLOW trajectory, which represents the trajectory with the latest achievable

ETA using the iteration DOF. The FAST and SLOW trajectories are used to establish the boundaries of the iteration space within which feasible meet-time solutions are sought.

Upon first checking NOM to determine if it satisfies the meet-time requirement, the FAST and SLOW trajectories are then tested. If one of these trajectories is found to satisfy the required STA, then it is returned as the appropriate solution. In addition, if the ETA associated with the FAST trajectory occurs later than the target STA, then the FAST trajectory is returned as the closest achievable solution. Similarly, if the ETA associated with the SLOW trajectory occurs earlier than the target STA, then the SLOW trajectory is returned as the closest achievable solution. In the more common case where the STA falls between the FAST and SLOW ETA limits, then a series of linear interpolations are performed until a solution is found. For each DOF interpolation, the TS process is called to compute the corresponding ETA at the meter fix, which is tested against the desired STA. In general, this linear interpolation process converges upon a solution within three iterations.

V. Conflict Detection

Upon computing a provisional meet-time solution for each metered arrival, EDA performs a Conflict Detection (CD) process on each aircraft in the airspace. This process incorporates the standard CTAS conflict probe used by other Center automation tools, such as Direct-To.[11] A conflict is defined by the predicted violation of an aircraft's Protected Air Zone (PAZ) by another aircraft over the CD time horizon, nominally set to 20 minutes for strategic conflict detection.

The CD process first checks for any active conflicts, defined as a predicted PAZ violation along any aircraft's active trajectory prediction over the CD time horizon. For active conflicts, the minimum size of the PAZ used by CD is defined by the FAA minimum separation requirements in Center airspace: 5 n.mi horizontal separation or 2,000 ft vertical separation, for aircraft flying above 18,000 ft (5 n.mi horizontal separation or 1,000 ft vertical separation, for flight below 18,000 ft).[§§] In the event that active conflicts are detected, data such as time-to-conflict and minimum predicted separation are presented to the controller for each conflict pair. For EDA, active conflict detection is performed in order to identify cases when the EDA solution results in both meet-time conformance and strategic conflict resolution. As described later, these cases can be pointed out to the controller as an aid to prioritizing advisory extraction and communication.

Once active conflicts have been detected, CD is repeated again in order to predict conflicts between each aircraft's provisional (meet-time only) trajectory and other aircraft active trajectories. The Conflict Resolution (CR) process is then invoked to resolve these provisional conflicts. In order to enable CR to generate conservative resolutions that account for conflict detection and maneuver execution uncertainty, the PAZ for conflicts between provisional and active trajectories is typically set larger than FAA minimum separation requirements.

VI. Conflict Resolution Automation

A. Overview

To provide strategic separation assurance, EDA performs an automated Conflict Resolution (CR) process once a provisional meet-time solution has been found. Although aircraft subject to metering are sometimes spaced to avoid conflicts well upstream of the meter fix, exceptions often occur. One typical example of this is the "blow by" situation, which results when a faster aircraft overtakes a slower aircraft to meet an STA at the meter fix that is earlier than that scheduled for the slower aircraft. Although the CR algorithm described here was originally designed to resolve only arrival vs. arrival conflicts,[12] it has recently been extended to resolve arbitrary conflicts between metered arrivals and other aircraft in the airspace. If a conflict is detected, however, the resolution maneuver is applied exclusively to the EDA arrival aircraft by adjusting its provisional meet-time solution. The current algorithm is limited to resolving conflicts with speed and altitude changes. Future research, however, will investigate the use of path adjustment as an additional method of conflict resolution.

The CR algorithm resolves conflicts by making speed and altitude adjustments about the provisional meet-time solution already developed. The general idea is to vary speed or altitude in the flight phase where the conflict occurs, and then make a compensating change in the remaining phase in order to change the trajectory geometry while preserving the meet-time solution. Two typical options are: 1) increase $CAS_{C,adv}$ while decreasing $CAS_{D,adv}$ or, 2) increase $CAS_{D,adv}$ while decreasing $CAS_{C,adv}$. In the event that speed changes alone are insufficient to satisfy both meet-time and separation constraints, then altitude is introduced as a secondary DOF. An example of applying this

[§§] Reduced Vertical Minimum Separation (RVSM), anticipated in January 2005, will require only 1,000 ft vertical separation throughout the Center.

resolution strategy to cruise and descent conflicts involving converging aircraft is illustrated in Fig. 9 and Fig. 10, respectively.

B. Resolution Algorithm

Prior to resolution, each aircraft in conflict is arranged in a list together with the aircraft that it's first predicted to come into conflict with. The speed and routing used to initialize CR is that associated with the underlying meet-time provisional trajectory. In the event that no DOF adjustments occurred during the meet-time process, then the initial speed and routing used for CR is identical to that upon which the current active trajectory is based.

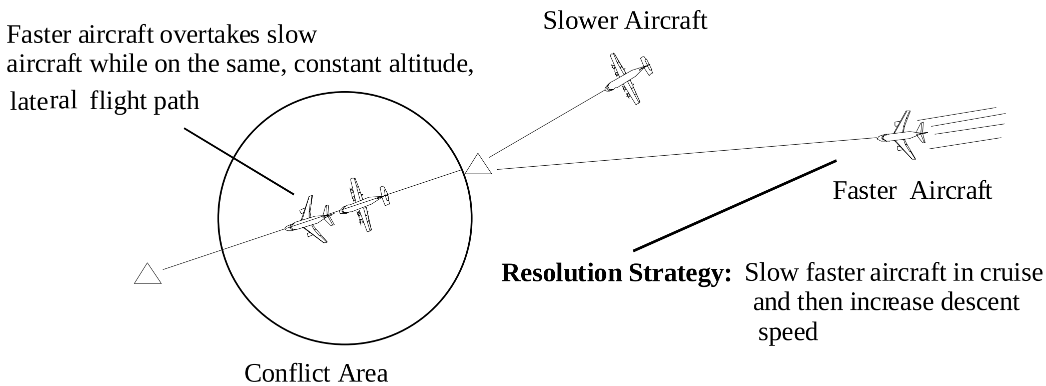

Figure 9. Cruise Conflict Example

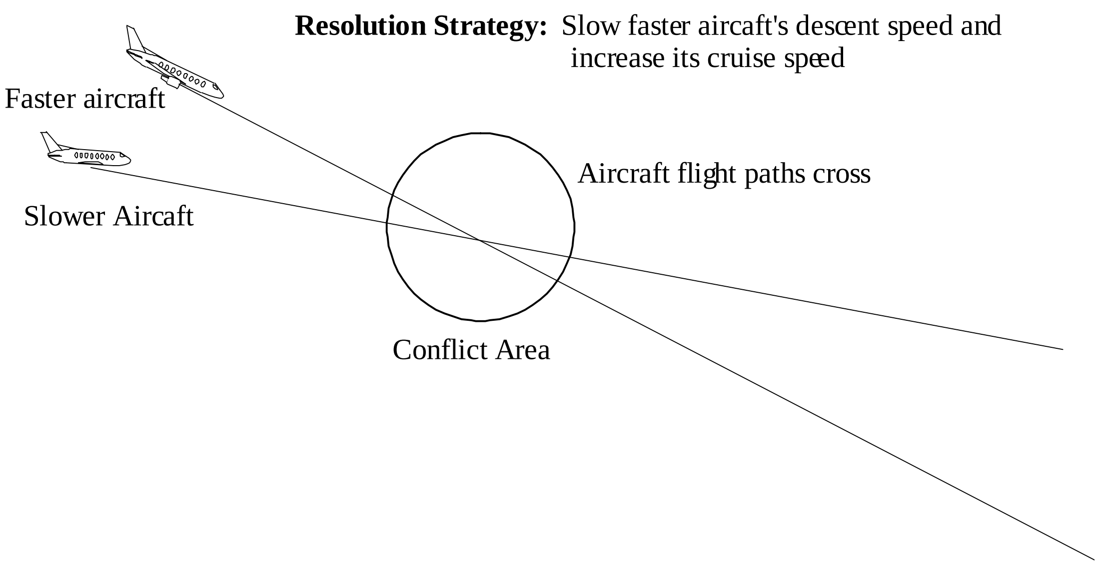

Figure 10. Descent Conflict Example

The basic CR algorithm is shown in Fig. 11. Since the algorithm is dependent on adjustments to both cruise and descent DOFs, it is only valid in cases where the aircraft has not yet reached its initial TOD. The search for a conflict-free, meet-time solution begins by incrementing $CAS_{D,adv}$ (by 5 kt). The TS then builds a new trajectory based on the adjusted descent speed, which is again tested for meet-time conformance at the meter fix. In the event that the trajectory no longer satisfies the meet-time constraint, as would typically be the case as a result of the faster descent speed, $CAS_{C,adv}$ is decremented in 5 kt intervals down to CAS_{Cmin}, until a meet-time solution is found. The trajectory is then re-tested for separation conflicts against all other active trajectories in the airspace. If no conflicts are detected, then the trajectory and associated speed advisories are returned by the CR process. In the event that conflicts remain, the algorithm resorts again to iterating on $CAS_{D,adv}$. For each $CAS_{D,adv}$ iteration, the algorithm alternately increments and decrements descent speed (in 5 kt intervals) until the min/max operational descent CAS limits are reached. For each descent speed adjustment, the algorithm repeats the process of adjusting cruise speed for meet-time conformance, and checking for conflicts until a complete solution is found.

If cruise speed limits are reached prior to finding a meet-time solution during the iteration process, then cruise altitude is invoked as an additional DOF[***] In this case, a lower cruise altitude, represented by $h_{C,adv}$, is tested to help identify a trajectory solution that satisfies both arrival time and separation constraints. For a given airspeed profile, lowering the altitude at which the aircraft flies over the remaining cruise segment has the effect of slowing ground-speed and delaying arrival time. Following the adjustment in $h_{C,adv}$, the provisional cruise and descent airspeed DOFs, $CAS_{C,adv}$ and $CAS_{D,adv}$, are reset to the initial values described previously. The iteration process then begins again with speed adjustments. In the event that the entire process continues to the point where $h_{C,adv}$ is decremented beyond the low-altitude limit (currently set to the meter-fix crossing altitude), the CR algorithm concludes that no satisfactory solution exists, given available DOFs.

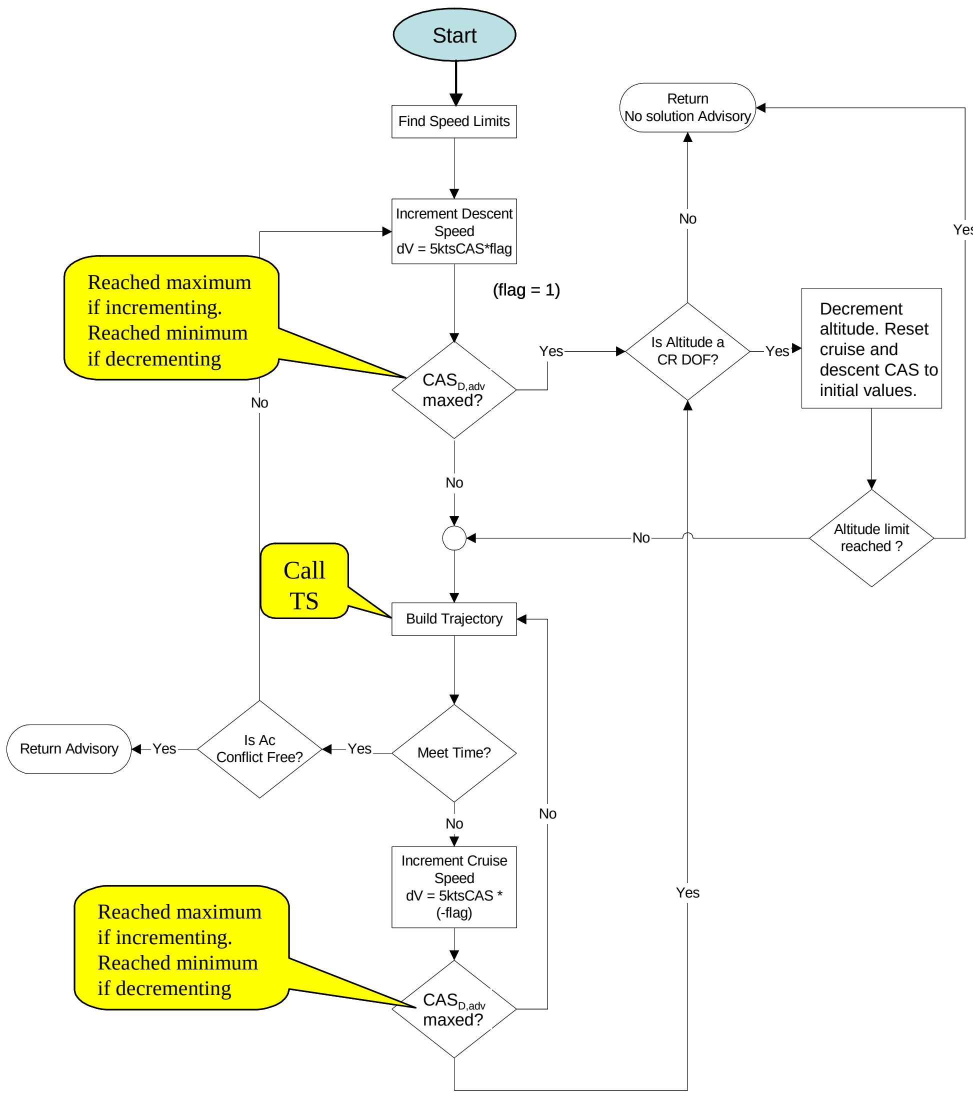

Figure 11. Basic EDA Conflict Resolution Algorithm

*** To meet current FAA altitude separation requirements and assignment rules based on heading, altitude is decremented by 4,000 ft for flight above 29,000 ft, and by 2,000 ft for flight below.

VII. User Interface

A. Overview

The EDA user interface design is based upon the premise of minimizing controller workload, while simultaneously providing sufficient control and flexibility over advisory content and timing. Past TBM field tests with CTAS have demonstrated the importance of automating advisory formulation and display to the greatest extent possible in order to mitigate workload in the arrival domain.[13] The EDA user interface therefore attempts to deliver a high degree of automation while still placing the controller at the center of decision-making process. This is accomplished by allowing controllers to view and evaluate advisories in a provisional manner before accepting them and delivering them as clearances to the flight-deck. Also, a controller can, for any reason, choose to ignore or cancel an EDA advisory and resort back to manual methods for solving the arrival problem.

B. Configuration

Prior to operation, controllers can configure EDA to provide advisories using any of the default meet-time speed modes previously described, i.e., *Descent-Only*, *Cruise-Only*, or *Cruise-Equals-Descent*. This is accomplished through the configuration panel shown in Fig. 12. Similarly, through this panel, the controller can choose to turn automated CR on or off. Other options, selected through this panel, pertain to the format in which advisories are displayed, and related trigger conditions. Conditions that trigger advisories to appear on the controller's Plan-view Graphical User Interface (PGUI) are based on the tolerance between an aircraft's current ETA - computed from the most recent active trajectory prediction - and it's required STA at the meter fix. The meet-time tolerance for initial EDA advisory annunciation is typically set to 30 seconds.

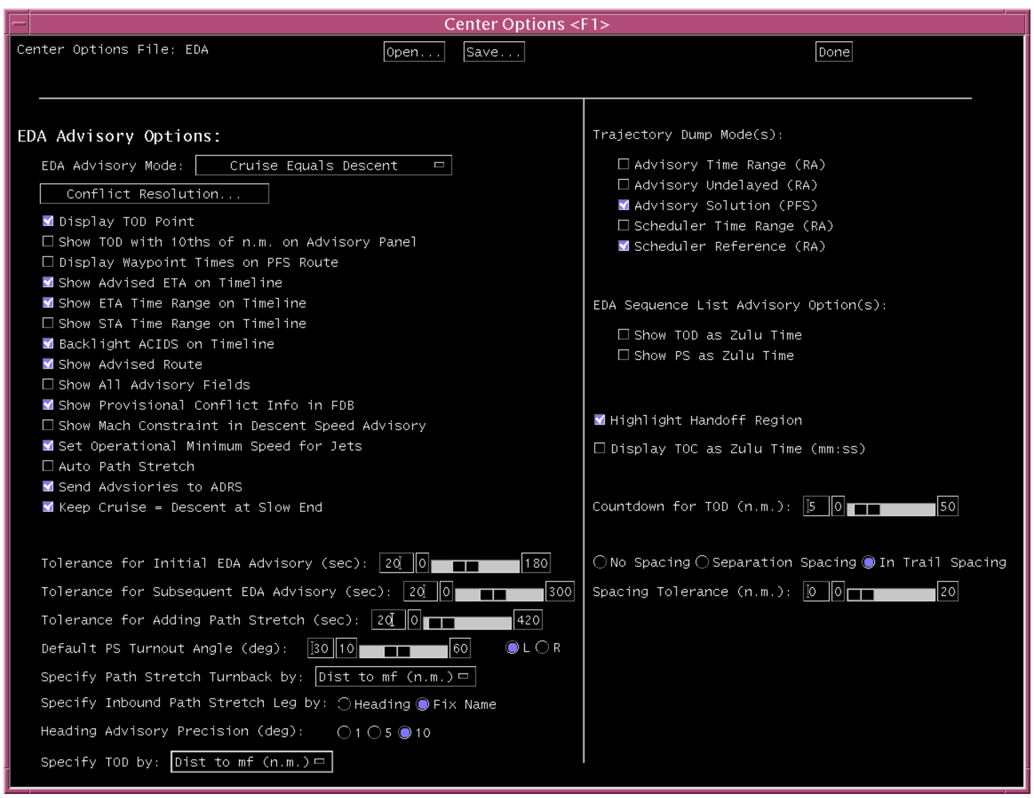

Figure 12. EDA Configuration Panel

C. EDA PGUI

A snapshot of the CTAS research PGUI, adapted for EDA, is shown in Fig. 13. The time line on the left of the display presents the metering schedule information. TMA-derived STAs are indicated by the position of the Aircraft Identifiers (ACIDS) on the right side of the timeline, while current ETAs are indicated by the corresponding ACIDS on the left. Whenever an aircraft is in conformance with the TMA schedule, its ACIDs to the left and right of the time line will be in alignment, indicating that its meter-fix ETA matches its TMA-computed STA. Similarly, any difference on the time line between ETA and STA ACIDS represents the amount of time that needs to be absorbed (or made up) through a subsequent EDA advisory. Fig.13 shows a Ft. Worth Center traffic scenario involving arrivals from the northeast being metered to the fix KARLA, en route to the Dallas/Ft. Worth airport. In this case, the controller has dwelled on the aircraft designated as EGF 764, which highlights a total required delay of 4.5 minutes for meet-time conformance at KARLA. The availability of a corresponding EDA meet-time advisory is indicated by the presence of the EDA portal, designated in yellow by the letters "EDA" in the last line of the aircraft's Flight Data Block (FDB). Following the standard format for Center automation, the first three lines of the FDB convey ACID, destination airport, altitude, aircraft type, and ground speed information. The 4th line of the FDB is reserved for any active conflicts that might be detected. The 5th line of the FDB is reserved exclusively for EDA advisory information.

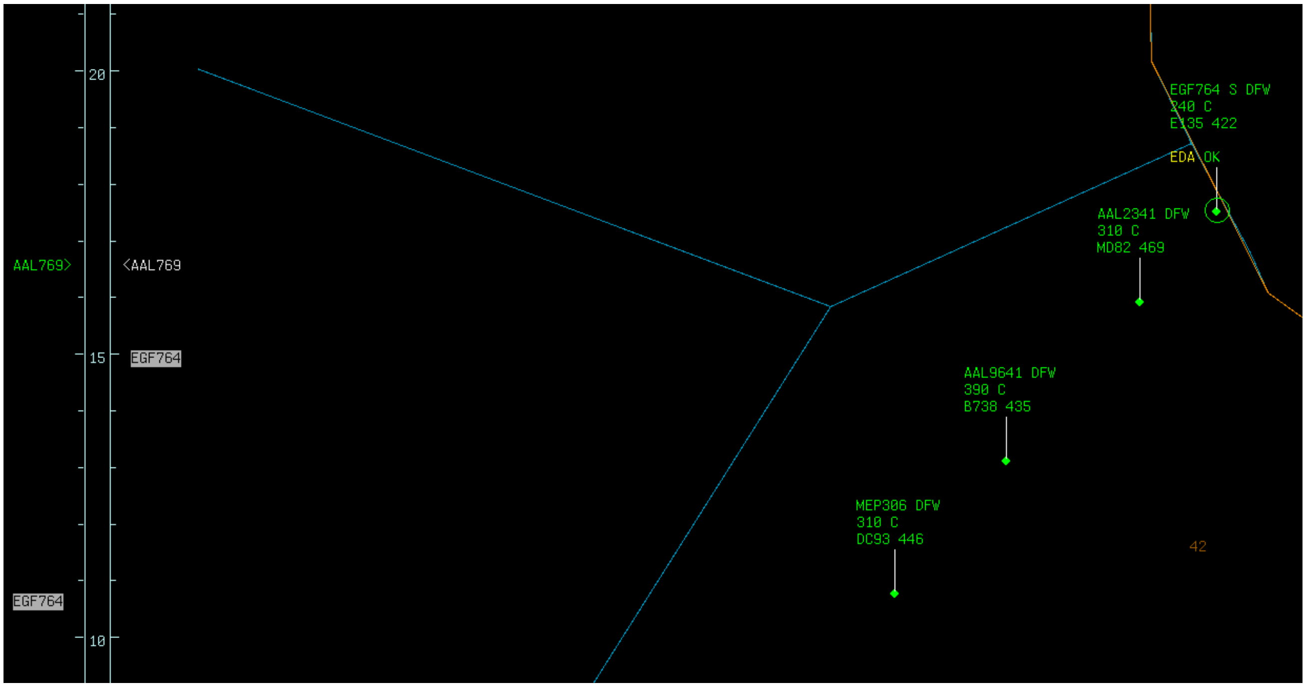

Figure 13. Scenario Showing Meet-Time Problem and Availability of EDA Advisory

D. Advisory Display

As shown in Fig. 14, once the controller is ready to receive an EDA advisory, he/she clicks on the EDA portal, which results in the portal being replaced by explicit advisory information. In this example, the default speed mode has been set to *Cruise-Equals-Descent*, and the initial advisory that results for EGF 764 is given as: "C/250 D/250 OK", which implies a 250 kt CAS advisory for both cruise and descent segments. The "OK" designation signifies that the provisional trajectory associated with this advisory is conflict free.[†††] Clicking on the portal also results in the display of a provisional TOD associated with speed advisory, along with provisional ETA information on the

[†††] If an original active separation conflict is resolved by the provisional meet-time solution, "OK*" is presented in this field. If the provisional meet-time solution is predicted to cause a conflict with another aircraft's active trajectory, then "C" is presented in this field in orange.

time-line. This provisional (i.e., advised) information is displayed in *yellow*, as opposed to active trajectory information that is displayed in *white*.

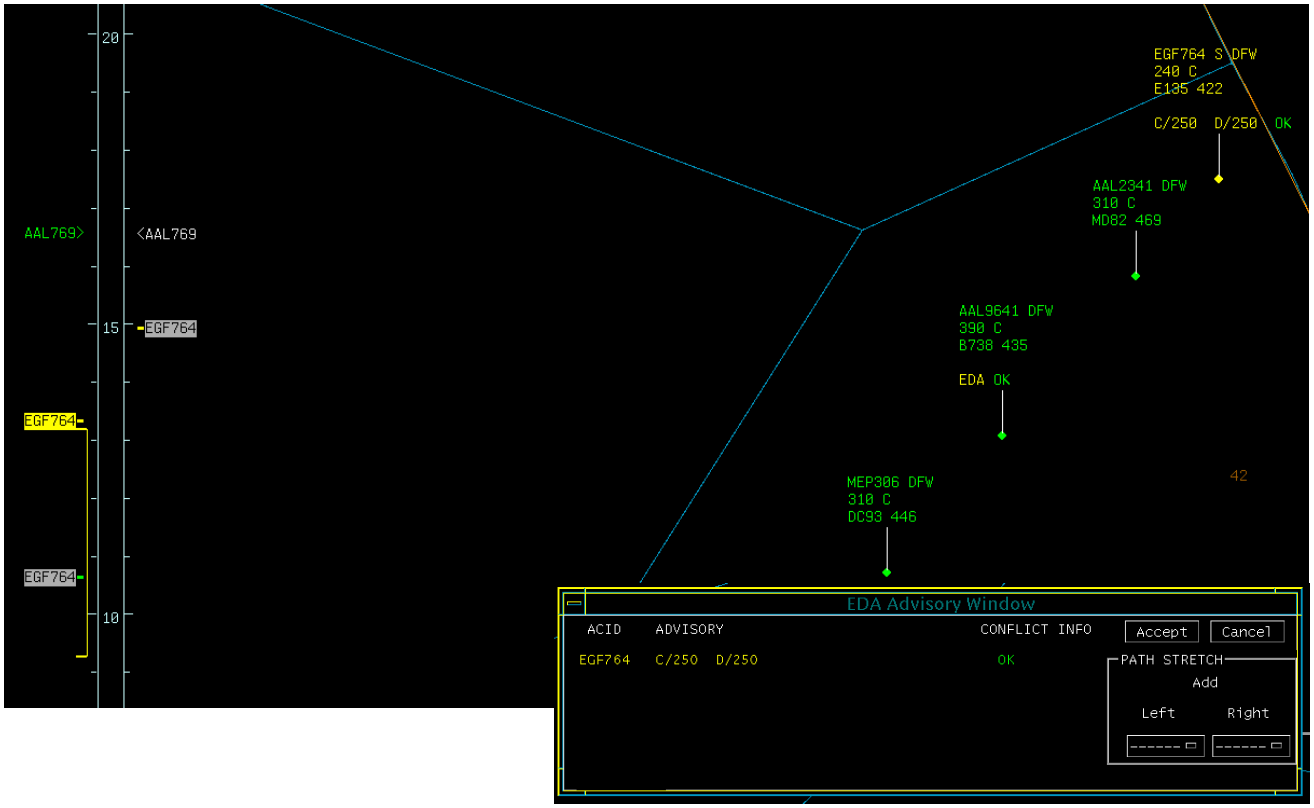

Figure 14. Partial Solution with Cruise-Equals-Descent Advisory

Due to the magnitude of the delay required, Fig. 14 shows that only a partial solution is achievable with the default speed mode. This is evident on the time-line by observing that the provisional ETA results in less than 3 minutes of total delay absorption. The fact that Cruise-Equals-Descent has been saturated is further evident by the yellow bar at the left of the timeline, which shows the range of arrival times achievable with this mode. Note; the ETA limits of this time-range bar are those associated with the FAST and SLOW trajectories, described previously.

As shown in the lower-right of Fig. 14, clicking the portal also results in the opening of the *EDA Advisory Window*. The advisory displayed in the FDB is repeated in this window along with any relevant conflict information. The window also contains options for accepting, canceling, or modifying the current advisory solution. In the current example, the controller is presented with the option of adding a *Path-Stretch* advisory component to resolve the remaining meet-time problem. Here, the controller can choose between turn-out angle options of 30˚, 45˚, and 60˚, and whether the path-stretch maneuver should occur to the left or right side of the nominal route. These options were developed in order to enable controllers to customize a conflict-free path-stretch solution that absorbs the required delay without violating airspace boundary constraints.

In the current example, the controller decides to introduce a left-side path stretch maneuver with an initial turn-out heading deviation of 30˚. This results in the display of a new provisional path and ETA reflecting this maneuver option, as shown in Fig. 15. From the time-line, it can be seen that adding Path-Stretch solves the remaining meet-time problem. The entire advisory associated with the complete Cruise Equals Descent plus Path Stretch solution is now given as “P/187 C/250 P/61/KARLA D/250”.[‡‡‡] In addition to the cruise and descent speed components

[‡‡‡] To facilitate clearance delivery by voice, the advisory components associated with immediate actions are presented first. Following the acceptance and communication of these initial instructions, the remaining components

already discussed, EDA advises stretching the route by flying an initial turn-out heading of 187˚, followed by a turn-back to KARLA that is initiated at 61 n.mi from the reference fix. Note that the portion of the advisory that requires immediate action by the pilot (i.e., “P/187 C/250”) is displayed in front of that requiring future execution (i.e., “P/61/KARLA D/250”). Upon receipt by the flight crew, it is expected that clearance instructions requiring future execution would be entered into the FMS.

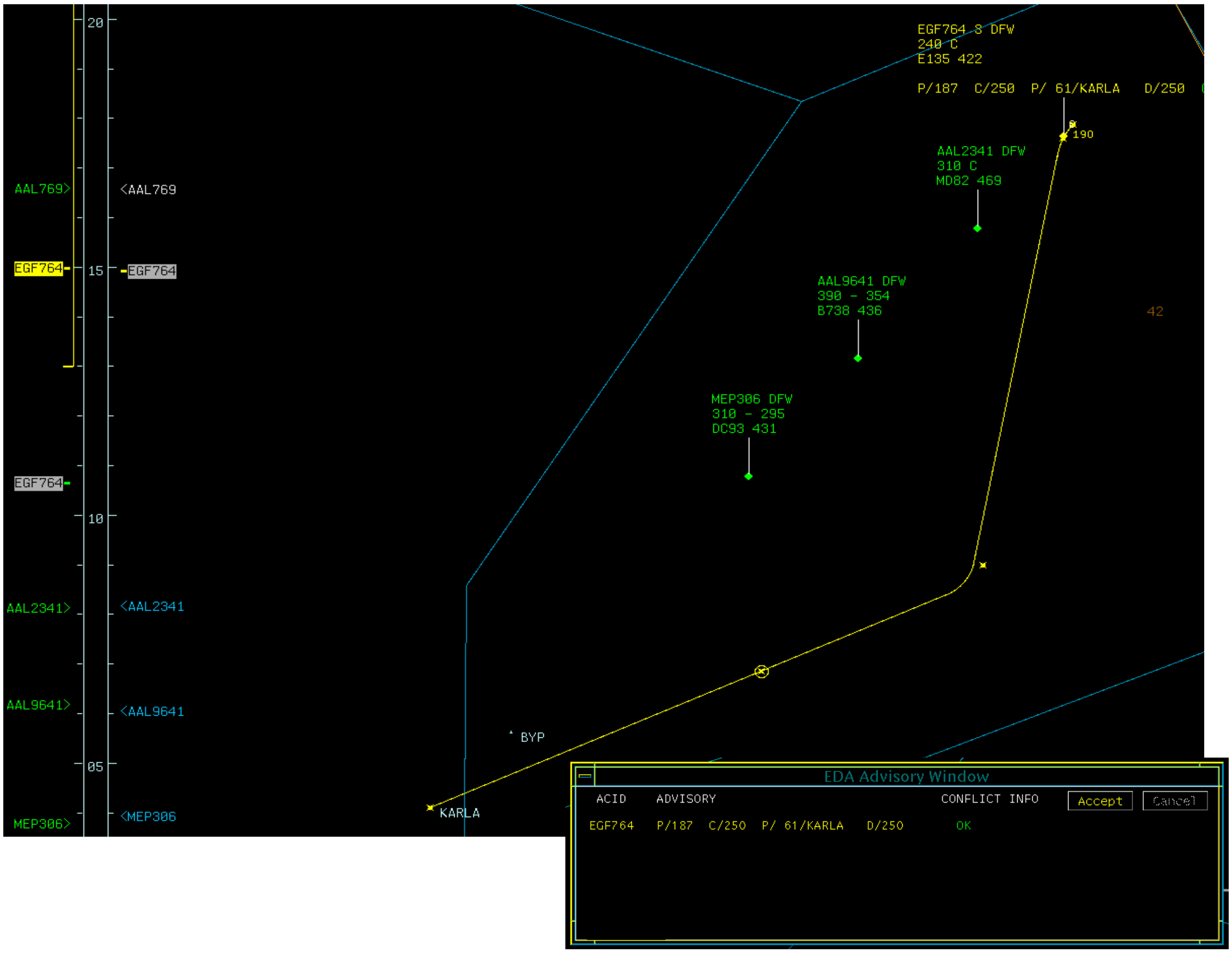

Figure 15. Complete Solution with Cruise-Equals-Descent Plus Path-Stretch Advisory

Once the entire advisory has been accepted, all provisional information on the PGUI disappears; the Advisory Window closes; and the portal changes from yellow to green, indicating that the meet-time problem has been solved. In addition, the TS updates the active trajectory prediction for the aircraft based upon intended actions associated with the accepted advisory. This results in the partial conformance of the active ETA on the timeline with the target STA. However, because active trajectory predictions are based on current cruise speed and heading, the ETA will only drift into full conformance with the STA once the aircraft complies with the initial advisory instructions. If for any reason the aircraft is again predicted to not meet its arrival STA, the EDA portal will reappear in yellow,

---

are appended. This two-step clearance delivery process could be simplified with the incorporation of Controller-Pilot Data-link Communication (CPDLC).

indicating that a corrective EDA solution is available. To allow the controller to track issued advisories and monitor conformance, the most recent advisory can be recalled in 5th line of the FDB, whenever the controller dwells on the EDA portal. In addition, an *Advisory History* list is available upon request in order to allow controllers to view all recent EDA advisories issued to aircraft.

VIII. Closed-loop Testing

A. Simulation System

The current EDA development process is characterized by incremental design and development followed by extensive human-in-the-loop testing by end-users (i.e., full-performance-level air-traffic controllers). The ability to conduct realistic simulation experiments that enable the control loop between EDA and the aircraft to be closed is essential to this process. To support this, the simulation system depicted in Fig. 16 has been implemented. In this system, TMA is operated by establishing an airport acceptance rate that translates into STA assignments at the meter fix. The controller, through the EDA PGUI, receives EDA advisories and communicates clearances to *pseudo pilots* through an audio link. Pseudo pilots receive these instructions, and input them into an FMS/autopilot emulator referred to as the Multi Aircraft Control System (MACS).[14] MACS then sends the appropriate control instructions to an aircraft target generator known as the Pseudo Aircraft System (PAS).[15] Data is then sent back to EDA from PAS in order to simulate the track and flight-plan information that would normally be received from the FAA Host computer. In order to facilitate engineering tests without requiring pseudo pilots, a two-way interface has been implemented that allows EDA maneuver instructions to be automatically up-linked and entered into MACS, following controller acceptance from the PGUI.

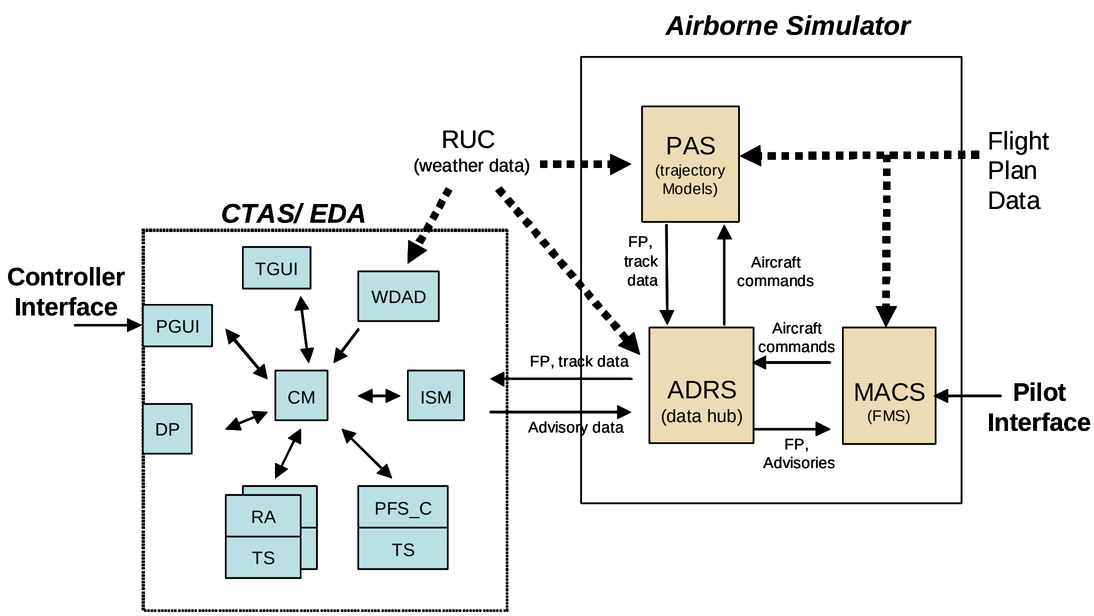

Figure 16. Closed-Loop Simulation System

B. Preliminary Findings

With the emphasis on concept definition and system development, most findings associated with recent experiments have been qualitative in nature. In general, controller subjects have commented on the enormous potential for EDA to reduce workload and increase efficiency during TBM operations. This results from the ability of EDA to generate a single, comprehensive, conflict-free, meet-time solution 20 to 40 minutes upstream of the meter fix. Simulation results, in the presense of flight execution error, indicate that aircraft can be routinely delivered to the meter fix within 20 seconds of their required STA, based on a single EDA clearance instruction given 25 minutes upstream of the meter fix. This equates to approximately a 70% improvement over operations using manual methods to target TMA arrival times. (Manual TBM techniques used in today's operation are generally associated with a meter-fix delivery accuracy of ±90-seconds, even with the use of multiple corrective clearances.)[16]

Although current operations do not support the delivery of strategic clearances that require maneuvers to be executed at future points in the flight, controllers believe that this can be accomplished with minimal changes to existing procedures. Similarly, controllers are optimistic that inter-sector coordination requirements associated with strategic EDA clearances can be successfully addressed with minor procedural changes. Although future air-ground data-link technology would certainly simplify the communication of EDA clearances, controllers feel strongly that EDA can be made to work in today's voice-based environment. To this end, the phraseology associated with EDA clearance delivery has been significantly streamlined during recent simulation experiments with controllers.

IX. Conclusion

A comprehensive description of the design, operation, and testing of EDA has been provided. The supporting trajectory prediction process has been explained in detail, along with the core meet-time and conflict resolution functions. In addition, the displays and mechanisms that allow controllers to receive, inspect, and accept EDA advisories have been explained through example. A system for conducting human-in-the-loop simulation experiments in order to obtain valuable, end-user design feedback has been described. Based on this system, preliminary experiments with controllers suggest that EDA has tremendous potential for improving the accuracy, efficiency, and workload associated with time-based metering operations.

Acknowledgments

The authors would like to acknowledge the following individuals for their crucial contribution to the design and development of the current EDA prototype: Dr. Husni Idris, Dr. Leo Javits, and Dr. David Chesler of Titan Corporation; Rey Salcido of the University of California, Santa Cruz; and Jeff Gateley and Liang Cheng of Seagull Technology.

References

[1]Swenson, H. N., Hoang, T., Engelland, S., Vincent, D., Sanders, T., et al., "Design and Operational Evaluation of the Traffic Management Adviser at the Ft. Worth Air Route Traffic Control Center," *Proceedings of the 1st USA/Europe Air Traffic Management R&D Seminar*, Saclay, France, 1997.

[2]Farley, T. C.,Foster, J., Hoang, and T., Lee, K., "A Time-Based Approach to Metering Arrival Traffic to Philadelphia," *Proceedings of the First AIAA Aircraft Technology, Integration, and Operations Forum*, Los Angeles, CA, 2001.

[3]Green, S. M., and Vivona, R. A., "En route Descent Advisor Concept for Arrival Metering," *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, Montreal, Canada, 2001.

[4]Erzberger, H., and Nedell, W., "Design of Automated System for Management of Arrival Traffic," NASA Technical Memorandum 102201, 1989.

[5]Davis, T.J., and S.M. Green, "Piloted Simulation of a Ground-Based Time Control Concept for Air Traffic Control," NASA Technical Memorandum 101086, 1989.

[6]Green, S., "Descent Advisor Preliminary Field Test," *Proceedings of the AIAA Guidance, Navigation, and Control Conference,* Baltimore, MD, 1995.

[7]Green, S., "Field Evaluation of Descent Advisor Trajectory Prediction Accuracy," *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, San Diego, 1996.

[8]Slattery, R.A., and Zhao, Y., "En Route Descent Trajectory Synthesis for Air Traffic Control Automation," *Proceedings of the American Control Conference*, Seattle, WA, 1995.

[9]Coppenbarger, R. A., Kanning, G., and Salcido, R., "Real-Time Data Link of Aircraft Parameters to the Center-TRACON Automation System (CTAS)", *Proceedings of the 4th USA/Europe ATM R&D Seminar*, Santa Fe, NM, 2001.

[10]Javits, L., "Design Document for Implementation of Cruise-Equals-Descent Meet-Time mode in RAPS,", Titan Systems Corporation, Billerica, MA, 2004 (unpublished).

[11] Erzberger, H., Paielli, R. A., Isaacson, D. R., and Eshow, M.M., "Conflict Detection and Resolution In the Presence of Prediction Error," *Proceedings of the 1st USA/Europe Air Traffic Management R&D Seminar*, Saclay, France, 1997.

[12]Slattery, R.A., and S.M. Green, *Conflict-Free Trajectory Planning for Air Traffic Control Automation,* NASA TM-108790, 1994.

[13]McNally, B. D., Bach, R.E., Chan, W., "Field Test Evaluation of the CTAS Conflict Prediction and Trial Planning Capability," *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, Boston, MA, 1998.

[14]Prevot, T., Palmer, E., Smith, N., and Callantine, T., "A Multi-Fidelity Simulation Environment for Human-in-the-Loop Studies of Distributed Air Ground Traffic Management", AIAA-2002-4679, Reston, VA, 2002.

[15]Weske, R.A., and Danek, G. L., “Pseudo Aircraft Systems: A Multi-Aircraft Simulation System for Air Traffic Control Research,” *Proceedings of the AIAA Flight Simulation Technologies Conference*, Monterey, CA, 1993