AIAA Guidance, Navigation, and Control Conference
08 - 11 August 2011, Portland, Oregon

AIAA 2011-6216

Impacts of Intermediate Cruise-Altitude Advisory for Conflict-Free Continuous-Descent Arrival

Miwa Hayashi[*] and Richard A. Coppenbarger[†]
NASA Ames Research Center, Moffett Field, CA 94035

Doug Sweet[‡] and Gaurav Nagle[§]
Sensis Corporation, Campbell, CA 95008

and

Gregory L. Dyer[**]
Federal Aviation Administration, Longmont, CO 80501

Efficient Descent Advisor (EDA) is a proposed ground-based decision-support tool for Air Route Traffic Control Center (ARTCC) sector controllers for managing arrival flows. It calculates the maneuver instructions for an arrival flight to fly a conflict-free (when able), fuel-efficient, continuous-descent trajectory and also meet a time-based metering requirement at the Terminal Radar Approach Control (TRACON) boundary. Currently, speed variations and path-stretch maneuvers are the only degrees of freedom used by EDA to find a solution. This study examined the feasibility of altitude change as an additional degree of freedom. Results of the human-in-the-loop simulation experiment showed that the altitude-advisory capability reduced the number of conflicting EDA advisories. It also reduced controller workload when the traffic situation was complex. Results also suggested that changes to EDA's user interface design and inter-sector coordination procedures are required for controllers to accept an altitude-advisory capability.

Nomenclature

3DPAM	=	Three-Dimensional Path Arrival Management
Alt	=	Runs where the altitude-advisory capability was provided
ARTCC	=	Air Route Traffic Control Center
ATC	=	Air Traffic Control
CA	=	Conflict Avoidance
CAS	=	Calibrated Airspeed
CDA	=	Continuous Descent Approach
CTAS	=	Center-TRACON Automation System
ETA	=	Estimated Time of Arrival
FAA	=	Federal Aviation Administration
FDB	=	Flight Data Block
FMS	=	Flight Management System
LNAV	=	Lateral Navigation
MACS	=	Multi Aircraft Control System
MCP	=	Mode Control Panel
NoAlt	=	Runs where the altitude-advisory capability was not provided

[*] Aerospace Engineer, Aviation Systems Division, MS210-8, Senior Member AIAA.
[†] Aerospace Engineer, Aviation Systems Division, MS210-10, Associate Fellow AIAA.
[‡] Director, Seagull Technology Center, 1700 Dell Avenue, Member AIAA.
[§] Research Engineer, Seagull Technology Center, 1700 Dell Avenue, Member AIAA.
[**] Support Manager, Airspace & Procedures, Denver ARTCC.

OPD	=	Optimized Profile Descent
STA	=	Scheduled Time of Arrival
TBM	=	Time-Based Metering
TMA	=	Traffic Management Advisor
TOD	=	Top of Descent
TRACON	=	Terminal Radar Approach Control
VNAV	=	Vertical Navigation
ZDV	=	Denver ARTCC

I. Introduction

The continuous descent approach (CDA), sometimes called optimized profile descent (OPD), has gained popularity recently as awareness of environmental concerns, as well as of airspace and airline operation efficiency, has grown. In the CDA/OPD, an aircraft descends continuously with an idle or a near-idle thrust setting. By eliminating level segments, the CDA/OPD reduces fuel consumption, noise, and greenhouse gas emissions. The CDA/OPD already has been tested in many airspaces (e.g., San Francisco,[1,2] Louisville,[3] Atlanta and Miami,[4] Schiphol,[5] Arlanda[6]). The CDA/OPD operations are typically conducted late at night or early in the morning, when the traffic density is low enough that the continuous-descent paths are not interrupted by other traffic.

At the same time, time-based metering (TBM) has been in effect during peak traffic times at several busy Terminal Radar Approach Control (TRACON) boundaries to ensure that the arrival demand does not exceed the available runway capacity as well as proper sequences and separations among incoming flights are maintained. TBM helps to reduce the frequency of low-altitude holdings and utilize maximum airport capacity.

Efficient Descent Advisor (EDA) is a proposed ground-based decision-support tool for Air Route Traffic Control Center (ARTCC) sector controllers. It aids CDA/OPD operation in busy en-route airspace where TBM is in effect. The goal is to provide an integrated solution to simultaneously optimize individual aircraft's flight paths and the entire arrival-flow scheduling. EDA's software calculates the cruise and descent speeds for individual arriving aircraft to meet the scheduled time of arrival (STA) at the TRACON boundary, assuming that the aircraft performs an idle or a near-idle thrust continuous descent. If speed changes alone are not sufficient to absorb the amount of delay needed to meet its STA, EDA adds a path stretch to either left or right of the course to allow for additional delay absorption (Fig. 1). EDA also tries to provide a conflict-free advisory whenever possible. This capability is a major enabler of CDA/OPD operation in high-density airspaces.

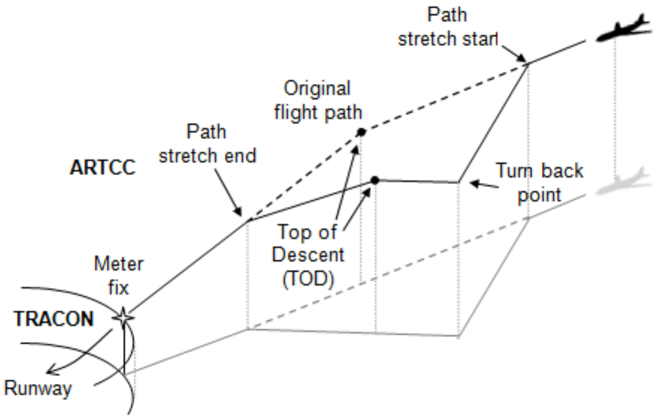


Figure 1. EDA path stretch.

It had been proposed to include an intermediate cruise-altitude change as the third degree of freedom to increase the chance of finding a conflict-free maneuver solution. Adding a vertical maneuvers, however, may also introduce new uncertainties. These tradeoffs are discussed further in a later section. The purpose of the present study is to investigate how the availability of an intermediate cruise-altitude advisory (hereafter *altitude advisory*) affects air traffic management performance, controller workload, and controller preference. A human-in-the-loop simulation experiment with current and retired controllers was conducted for this investigation.

The paper is organized as follows: Section II covers more details of EDA, including background, anticipated tradeoffs of the altitude advisory, EDA software computation process flows, and flight deck operations. Sections III and IV describe the human-in-the-loop simulation experiment method and results, respectively. Section V interrelates the results and interprets them in terms of impacts and design tradeoffs. Finally, Section VI presents conclusions.

II. Efficient Descent Advisor (EDA)

A. Background

EDA has been developed at NASA Ames Research Center as part of the ground-based air-traffic-control decision-support software suite called the Center-TRACON Automation System (CTAS).[7] EDA works in conjunction with Traffic Management Advisor (TMA), another element of CTAS. TMA calculates an individual arrival flight's STA used in the TBM operations.[8] EDA takes the STA as an input and computes the maneuver advisory for the flight to meet the STA.

The required onboard equipment for an aircraft to participate in EDA operation is a Flight Management System (FMS) with lateral-navigation (LNAV) and vertical-navigation (VNAV) modes and a radio communication device, both of which are already installed on most jet transport aircraft. Previous human-in-the-loop simulations[9] demonstrated that EDA operation works effectively with traditional voice communication. Not requiring a data link is an advantage, since it allows the possibility of deploying EDA technology into the field sooner. EDA recently has been a part of the 3D Path Arrival Management (3DPAM) program supported by the Federal Aviation Administration (FAA), NASA, and Boeing. The 3DPAM program focuses on a near-term (prior to 2018) potential deployment of EDA in a voice-based communication environment to support trajectory-based arrival operations in high-density airspace.

B. Anticipated Tradeoffs of Altitude Advisory

EDA searches for a conflict-free advisory by iterating through a series of speed and path-stretch angle combinations until it finds a solution that satisfies the metering-time requirement and is conflict-free, i.e., the projected trajectory does not violate the user-specified horizontal and vertical separation minima with other aircraft at any point in time up to reaching the meter fix. EDA may not be able to find such a conflict-free, meet-time trajectory solution in a high-density airspace. In previous EDA simulation studies,[9] the controllers expressed interest in including altitude as an additional degree of freedom in the search for a conflict-free solution. In real air-traffic management operations, cruise-altitude change, typically to a lower cruise altitude, is utilized regularly by controllers for separation assurance and flow management. Fig. 2 illustrates this maneuver.

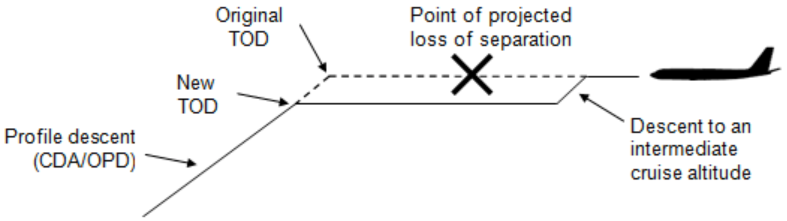

Original TOD
Point of projected loss of separation
New TOD
Profile descent (CDA/OPD)
Descent to an intermediate cruise altitude

Figure 2. Intermediate cruise altitude.

The cruise-altitude change can be a powerful tool in the EDA context as well. The obvious advantage is the ability to move the aircraft to a less-utilized altitude and, hence, increase the chance of finding a conflict-free solution. Another benefit that may be less obvious is that a larger delay can be absorbed at a lower altitude than at a higher altitude. This is due to the fact that an aircraft flies at a slower true airspeed at a lower altitude at the same indicated airspeed. For instance, in the advisory computation, EDA algorithms use 250 knots Calibrated Airspeed (CAS) or the equivalent Mach number as the minimum speed (the same minimum speed recommended for between FL280 and 10,000 ft., where the crossing altitude is typically located; Ref 10, ch. 5-7-3). However, 250 knots CAS is equivalent to Mach 0.76 at FL360, whereas it is equivalent to Mach 0.70 at FL320. The slight decrease in the true airspeed Mach number at the lower altitude is sometimes sufficient to absorb a large delay that could not be absorbed at the higher altitude.

There are two ways to determine a suitable cruise altitude for this purpose. One method is to have EDA provide an intermediate cruise-altitude advisory as part of its regular advisory package. An alternative method is to let the controller handle the altitude change and/or de-conflicting flights via legacy air traffic control (ATC) techniques before the EDA clearance is issued. The former is expected to reduce or at least maintain controllers' mental workload, since they would not have to search for a suitable altitude themselves. A downside of this approach is that the timing and the manner in which the aircraft descends to the intermediate cruise altitude may vary, which may introduce additional uncertainties. This may increase controllers' monitoring workload, negating the aforementioned workload benefit. In the latter method, controllers' workload may increase because there is no automation helping them to find a conflict-free altitude; however, they may retain better situation awareness because they will be determining an altitude that fits a traffic pattern they created in their mind rather than trying to discern why the tool suggested a certain altitude. The operation will be less affected by uncertainties because the altitude change is already completed before the EDA clearance is issued.

C. EDA Software Computation Process Flows

Fig. 3 illustrates the EDA computation process flows. For detailed descriptions of the algorithms used in each process, see Ref. 11. In the following descriptions, the corresponding block in Fig. 3 is referred to by the block number, and the text (*yes*, *no*, *accepted*, and *rejected*) following the number refers to the decision point that exits the block. The controller actions that took place in the simulation with the prototype EDA user interface are briefly described. More details of the prototype EDA user interface will be revisited in the User Interface and Phraseology section.

The process in Fig. 3 is invoked at each radar update (i.e., at 12-second intervals) for each arrival aircraft that has crossed the freeze horizon. The freeze horizon is a geographical threshold where the arrival aircraft's STA is frozen for schedule stability. The location of the freeze horizon is determined by the individual ARTCC facility and is typically situated around 130–200 nautical miles away from the meter fix. The radar update, consisting of the current position and speed of the aircraft (block 1) and its Flight Plan (block 2), is sent to the trajectory synthesis algorithms (block 3) to calculate the aircraft's Estimated Time of Arrival (ETA) at the meter fix. The TMA scheduler (block 4) computes the aircraft's STA based on its ETA[††] and freezes the STA when the aircraft's crossing of the freeze horizon is detected. Then, the required delay at the meter fix is calculated as Δ = STA – ETA. A negative delay means the aircraft needs to accelerate to make the scheduled slot. If the absolute value of the delay is equal to or less than the preset value, d (it was set at 20 seconds in the simulation) (block 5, *no*), the EDA computation process for this aircraft ends for this radar update (block 6) and will not be reactivated until the next radar update. If the absolute value exceeds d (block 5, *yes*), the controller is notified on the user interface (block 7) that the aircraft is not going to comply with the STA, and something needs to be adjusted. In the simulation, the *EDA Portal* appears in the Flight Data Block (FDB) to notify the controller. More than one aircraft's FDB may be indicating the EDA Portal at any given time.

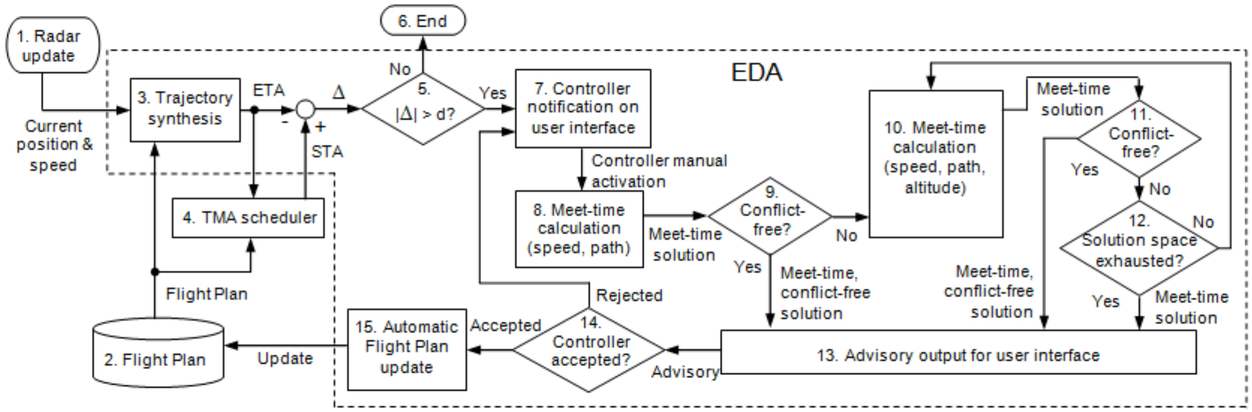

1. Radar update
Current position & speed
EDA
3. Trajectory synthesis
ETA
-
+
STA
Δ
4. TMA scheduler
Flight Plan
2. Flight Plan
6. End
No
5. |Δ| > d?
Yes
7. Controller notification on user interface
Controller manual activation
8. Meet-time calculation (speed, path)
Meet-time solution
9. Conflict-free?
No
Yes
Meet-time, conflict-free solution
10. Meet-time calculation (speed, path, altitude)
Meet-time solution
11. Conflict-free?
Yes
No
12. Solution space exhausted?
No
Yes
Meet-time, conflict-free solution
Meet-time solution
13. Advisory output for user interface
Advisory
14. Controller accepted?
Rejected
Accepted
15. Automatic Flight Plan update
Update

Figure 3. Computation process flows.

[††] The TMA scheduler also takes into account the airport arrival rate, separation at the meter fix, and the speed restriction at the meter fix.

The controller chooses one aircraft at a time to activate the meet-time calculation process (block 8). In the simulation, this action is accomplished via clicking on the EDA Portal of the selected aircraft. The meet-time calculation process computes the appropriate cruise and descent speeds and a path-stretch route (if needed) to meet the STA. The altitude degree of freedom is not considered in this initial meet-time calculation process, as it is used only for de-conflicting purposes. The solution obtained at this point always meets the meet-time requirement but may or may not be conflict free. If there is no conflict with other traffic detected (block 9, *yes*), the solution, which now meets the time and is projected to be free of conflict, becomes the EDA advisory and is presented to the controller (block 13). In the simulation, the EDA advisory is presented in a pop-up *EDA Window* on the controller's plan-view display.

If a conflict is predicted (block 9, *no*), then the conflict-avoidance (CA) algorithms, represented as a group of the blocks 10, 11, and 12 in Fig. 3, are activated. In this process, the meet-time calculation process is activated again (block 10) to iterate the calculations until either it finds a different combination of speeds and path-stretch route that is conflict free (block 11, *yes*) or the solution space is exhausted (block 12, *yes*).[‡‡] If the altitude-advisory capability is offered, the solution space extends to include different intermediate cruise altitudes. In such a case, first, the meet-time, conflict-free solution is sought at the current altitude only. Then, the same search process is repeated at 2000 feet below the current altitude and, then, again at 4000 feet lower. These two lower altitudes were selected to comply with the ATC's directional altitude assignment rules (Ref. 10, 4-5-2). The final solution—which is always a meet-time solution but may not be conflict free if the solution space has been exhausted—is then displayed on the user interface as an EDA advisory. If the advisory is not conflict free, the conflict information is presented alongside the EDA advisory.

If the controller agrees with the presented EDA advisory, he/she issues the EDA clearance to the pilot via voice communication. After issuing the EDA clearance, the controller makes an input via the user interface to inform the software that the clearance was issued (block 14, *accepted*). In the simulation, the controller presses the ACCEPT button in the EDA Window when the clearance is issued. This controller action automatically updates the aircraft's Flight Plan (block 15), relieving the controller from the need to type the updates manually every time. If the controller disagrees with the EDA advisory, he/she can reject it (block 14, *rejected*). In the simulation, the controller can do so by closing the EDA Window without accepting the advisory. For instance, if the advisory has a conflict, the controller may choose to close the EDA Window, de-conflict by changing the aircraft's cruise altitude or moving the conflicting aircraft, and open the EDA Window again to obtain a conflict-free EDA advisory. When the controller rejects the EDA advisory, the EDA Portal of the aircraft remains lit to indicate that the flight has a meet-time problem (block 7) until its $|\Delta|$ reduces below the threshold.

After an EDA clearance is issued to a flight, EDA continues to monitor the flight's $|\Delta|$ at each radar update, and if it exceeds the set threshold, *d* seconds, the same processes are started to provide a corrective EDA advisory. The EDA advisories are suppressed at 30 seconds prior to the projected Top of Descent (TOD) point.

D. Flight Deck Operations

Once the pilot receives the EDA clearance through the radio, he/she goes to the FMS's legs page and adds the waypoint to build the path-stretch route (if any). The turn-back point is specified as a Place-Bearing-Distance point from the path-stretch end waypoint. The cruise and descent speeds are input through the FMS's cruise and descent pages. The TOD for an idle-thrust CDA is calculated using the FMS's ECON-descent function, and then the ECON descent speed is overridden with the EDA descent speed.[1] To fly the FMS flight path properly, the LNAV and VNAV modes must be engaged in both the cruise and descent phases of flight.

III. Method

A. Airspace and Traffic

For the human-in-the-loop simulation experiment, the Northeast arrival traffic flows into the Denver International Airport were simulated. The flights went through high Sectors 09 and 16 and low Sector 15 of the Denver ARTCC (ZDV) and were eventually handed off to the Denver TRACON. See Fig. 4 for illustration. Two traffic flows, one via North Platte (LBF) and AMWAY, and the other via YANKI and Sidney (SNY), merged at the

[‡‡] Blocks 8, 9, 10, and 11 in Fig. 3 require calling the trajectory synthesis algorithms of block 3. These arrows are omitted in the diagram to reduce clutter.

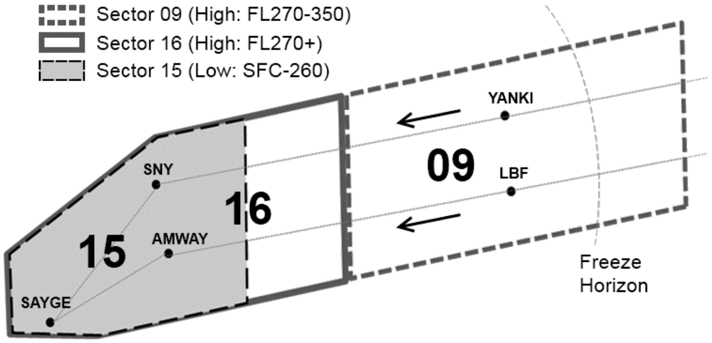


Figure 4. Simulated Northeast arrival flows in ZDV sectors.

SAYGE meter fix. The freeze horizon was located in Sector 09, approximately 190 nautical miles from SAYGE. A set of hypothetical published EDA profile-descent restrictions called TELLR 1 Profile was used in the simulation.

Two traffic scenarios were created by recording the live traffic data and modifying them, such as adding some over flights to increase conflicts. Scenario 1 included 36 arrivals and 31 over flights, whereas Scenario 2 consisted of 38 arrivals and 31 over flights. The arrival rate at SAYGE was 42 per hour in both scenarios to mimic a peak-time traffic load. The peak instantaneous aircraft counts during a run were in the range of 14~22 in Sectors 09 and 16, and 9~14 in Sector 15, depending on how the controller handled and handed off the traffic. The minimum horizontal separation threshold parameters set in the conflict identification computation were 6 nautical miles for conflict detection and 7 nautical miles for conflict resolution. The minimum vertical separation threshold parameters were 900 feet if both aircraft were in cruise or 2000 feet otherwise. The aircraft call signs were changed each time the scenario was reused in order to reduce learning effects.

B. Participants

Three ZDV controllers participated in the study, two of which were retired controllers. All three controllers had participated in the previous EDA simulation experiments[9] and were sufficiently familiar with the restrictions in the simulated ZDV airspaces. The controllers switched sectors after each run in the order specified in the experimental design (described later).

On the flight-deck side, three instrument-rated pilots were recruited from the local northern California area. Each pilot was assigned to a pseudo-pilot station for one of the three sectors, through which the pilot handled all the aircraft in the sector and any handoffs from/to a neighboring sector. The pilot participants' main tasks were to conduct radio communications with the controllers and perform handoffs via the pseudo-pilot station. They did not have to input the cleared EDA maneuvers into the FMS. The FMS was automatically uploaded, including intermediate altitude maneuvers, when the controller pressed the ACCEPT button. Occasionally, the controller issued non-EDA instructions, such as a heading, altitude, or speed change for separation. In that case, the pilot manually input the instructed maneuver via the Mode Control Panel (MCP) or the FMS.

C. Simulator

The simulation was conducted in the Crew-Vehicle Systems Research Facility at NASA Ames Research Center in April 2010. Eight computers running Windows XP Professional x64 and three Linux-based computers were networked for the simulation. Of the eight Windows machines, three were used for the sector controller seats, three for the pseudo-pilot stations, one for both traffic scenario control and ghost pseudo-pilot station operations that automatically managed all the traffic outside the three sectors, and one for audio data recording. CTAS was run on the two Linux-based experimenter stations to control and monitor the EDA computation processes. The other Linux machine was used for a ghost controller station, which, again, automatically managed all the traffic outside the three sectors. Multi Aircraft Control System (MACS) software developed at NASA Ames Research Center[12] was used to emulate the user interface of the sector controller's Display System Replacement radar-position (or the *R*-side) workstation. The data-position (or the *D*-side) workstation was not included in this simulation. The three controller seats were each equipped with a 2000 × 2000-pixel high-resolution BARCO monitor. Each controller was provided with the same Cortron keyboard and trackball used at the ARTCC facilities. MACS was also used as the pseudo-pilot stations to provide the pilot participants the interface to handle multiple aircraft in the sector. Each controller and the pilot participant wore a headset with a push-to-talk button to talk to each other on a set radio frequency.

D. User Interface and Phraseology

The following are the descriptions of the prototype EDA user interface designs implemented for the simulation. Fig. 5 shows the EDA Portal displayed on the fifth line of the FDB. (The fourth line is empty in this example. The first three lines indicate that the flight is Southwest 1300, whose mode-C-reported altitude is FL360 ± 200 feet, computer ID is 786, destination is Denver, and ground speed is 392 knots.) A cyan EDA Portal is presented while $|\Delta| = |\text{STA} - \text{ETA}|$ of the flight exceeds the set threshold, d. Clicking on the cyan EDA Portal opens an EDA Window (see Fig. 6, upper part) on the plan-view display.

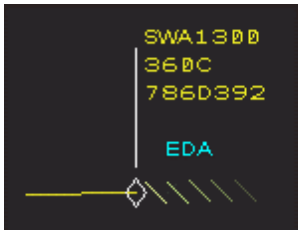
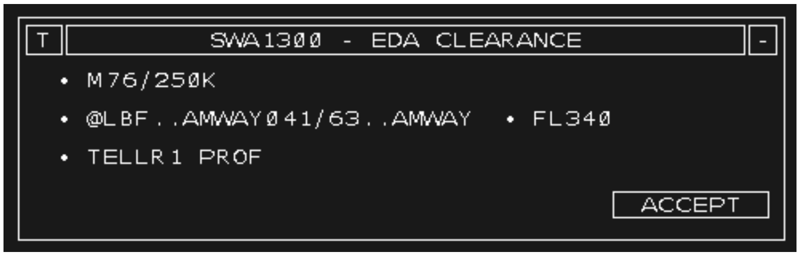


Figure 5. EDA portal on fifth line of FDB.

The EDA Window, shown in the upper part of Fig. 6, contains an example EDA advisory with cruise and descent speeds (Mach 0.76 and 250 knots CAS, respectively), a path stretch, and an altitude change. The lower part of the figure shows the phraseology corresponding to this example advisory. To facilitate clearance reading, each row represents a group of information to be transmitted to the pilot in a single radio communication.[13] The clearance that includes the altitude advisory required three radio communications to complete. Thus, the example EDA Window in Fig. 6 contains three rows. EDA clearances with a path stretch and speed changes (but no altitude change) required two radio transmissions, and those with only speed changes were completed in one radio transmission. The word "slash" was used to be consistent with the FMS input format. The EDA Window presents an ACCEPT button at the lower right corner. Clicking on the ACCEPT button updates the flight plan and the flight's FMS. The close button is provided at the upper right corner, and clicking on it closes the EDA Window without accepting the advisory. After the ACCEPT button is clicked, the EDA Portal color changes from cyan to gray to indicate the flight is now on an EDA clearance.

Fig. 7 depicts an example plan-view display where EDA could not find a conflict-free advisory. The path-stretch advisory will cause a conflict with Delta 2488 and FedEx 2577 (not shown). The lower part of the EDA Window presents the conflict information—from left to right, the flight call sign, the computer ID in parentheses, the conflict altitude in flight level, and the sector that currently owns the flight. The ACCEPT button is still enabled, but the label text is in red to warn the controller of the projected conflict. If the advisory is accepted with a conflict, a red box appears around the gray EDA Portal in the FDB as a reminder of the conflicts that existed at the time the EDA advisory was accepted.

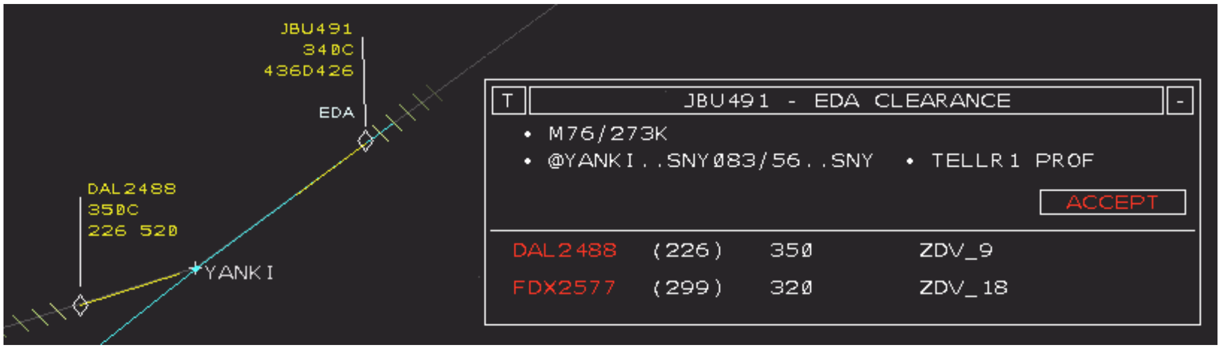


SWA13ØØ - EDA CLEARANCE

- M76/25ØK
- @LBF..AMWAYØ41/63..AMWAY • FL34Ø
- TELLR1 PROF

ATC: "Southwest 1300, EDA clearance, maintain Mach point 76 slash 250 knots, revised routing when ready to copy."
Pilot: "EDA clearance, maintain Mach point 76 slash 250 knots. Ready to copy revised routing. Southwest 1300."
ATC: "Southwest 1300, at North Platte, proceed direct the AMWAY 041 slash 63, then direct AMWAY. Descend and maintain flight level 340."
Pilot: "At North Platte, proceed direct the AMWAY 041 slash 63, then direct AMWAY. Descend and maintain flight level 340. Southwest 1300."
ATC: "Southwest 1300, descend via the TELLR 1 profile."
Pilot: "Descend via the TELLR 1 profile. Southwest 1300."

Figure 6. An example EDA window and its phraseology.



```
JBU491 - EDA CLEARANCE
• M76/273K
• @YANKI..SNY083/56..SNY    • TELLR1 PROF

DAL2488   (226)   35Ø     ZDV_9
FDX2577   (299)   32Ø     ZDV_18
```

Figure 7. An example of an EDA window with conflict information.

E. Experimental Variables and Measurements

The independent variables were the Participant (1, 2, and 3), the Conditions (with and without intermediate cruise-altitude advisory, or, hereafter, *Alt* and *NoAlt*), the Sector (09, 16, and 15), the Scenario (1 and 2), and the Run Group (1st four, 2nd four, and 3rd four runs).

There were two types of measured dependent variables, objective and subjective. One of the objective variables was the conflict-avoidance statistics obtained from the recorded EDA computation processes and screen video capture. The other was the meet-time accuracies calculated as STA minus the actual time of arrival (i.e., a positive value means the aircraft arrived early).

Subjective variables included real-time workload ratings, post-run workload ratings, and responses to the post-experiment questionnaire and debriefing. The real-time workload ratings were collected at seven-minute intervals, starting at 17 minutes after the scenario start. The scale for the real-time workload ratings was 1 through 6, where 1 was the lowest workload and 6 was the highest. A beep sounded at a seven-minute interval to prompt the experimenter to ask the participants to provide their current workload ratings. The post-run workload ratings were collected on a post-run questionnaire form. The Modified Bedford Workload Scale,[14] which ranges from 1 (the lowest workload) to 10 (the highest workload), was used for the post-run workload ratings, and the controller selected the rating that represented the maximum and average workload levels experienced during the run.

The main difference between the real-time and post-run workload ratings is that the former reflects the moment-to-moment traffic-management workload, while the latter assesses the controllers' aggregated impression of each run. The real-time workload scales also had fewer levels to facilitate quick assessment of the workload rating while the participants were working on traffic.

On the post-experiment questionnaire form, the participants marked their preference for the altitude-advisory capability on a Likert scale with five checkboxes (*disliked*, *somewhat disliked*, *neutral*, *somewhat liked*, and *liked*). Additional comments were also collected on the post-experiment questionnaire form and in the debriefing discussions.

F. Experimental Design

Data were collected from 12 ninety-minute runs and subjected to analyses. Each run was followed by a post-run questionnaire session, a break, and a debriefing. At the end of the 12 runs, participants filled out a post-experiment questionnaire, and the final briefing was held.

The 12 runs were originally designed to counterbalance the effects of the two conditions (Alt vs. NoAlt) within each participant, mainly to minimize possible learning and fatigue effects. However, technical problems forced five runs to be rerun at the end. This caused some deviation from the counterbalancing. The new run order may cause a slight advantage for the NoAlt condition over the Alt condition if learning effect is present, or a reversed bias if fatigue effect is dominant rather than the learning effect. This run order change also interfered with the Generalized Linear Model (GLM) analysis for the workload analyses. A brief explanation will be given in the Results section.

G. Scope and Limitations of the Study

The trajectory synthesis algorithms and the traffic scenarios used in the simulation were assumed to be reasonably realistic and accurate to the level required for the purposes of this laboratory study. It is not in the scope of this study to assess their fidelity. Evaluating the validity of the STAs calculated by the TMA scheduler was also out of the scope of this study, as the STAs were considered as inputs to EDA.

A limitation of the study that needs to be mentioned is that the pilots' behaviors simulated in this study were probably much better than the breadth of those normally observed in the real world. Our pilot participants were well versed in the EDA procedures, and the radio communication was free of static noise or chatter. In addition, the simulation environment that called for one pilot to control multiple aircraft forced the FMS input to be automatically and instantly uploaded. In other words, the simulation presented much fewer pilot-response uncertainties and errors than would normally be expected in real operations. This point is directly relevant to the test hypotheses and needs to be considered carefully. The study addressed this shortcoming by focusing more on the analyses of the subjective data (e.g., workload ratings and preference). The assumption was that the judgments the controllers made at each critical moment (such as "Should I give a vector?" or "Should I clear for the descent now?") were based on their own real-world operation experiences rather than the expected behaviors of the pilot participants in the simulation. The objective data (e.g., conflict-avoidance performance) were still collected to provide supportive information about the context for each run, which was valuable in understanding the subjective-data analysis results.

IV. Results

A. Objective Data

1. Conflict Avoidance

As expected, the altitude-advisory capability reduced the number of conflicting EDA clearances issued. In the NoAlt runs, the conflict-avoidance algorithms (i.e., blocks 10, 11, and 12 in Fig. 3) were activated 64 times, of which eight times ended up with conflicting EDA clearances being issued (12.5%). In the Alt runs, the conflict-avoidance algorithms were activated 58 times, of which only one resulted in a conflicting EDA clearance being issued (1.7%). This single case was inevitable because the two aircraft were already in conflict at the freeze horizon.

Table 1. Conflict avoidance statistics.

Condition	Traffic Scenario	# of Altitude Advisories Accepted	# of Conflicting Advisories Accepted	# of Times CA Algorithms Activated
Alt	1	7	1	31
	2	14	0	27
	Total	**21**	**1**	**58**
NoAlt	1	-	2	30
	2	-	6	34
	Total	**-**	**8**	**64**

Table 1 provides a breakdown of the above numbers by traffic scenario. The two traffic scenarios included about the same numbers of arrivals and over-flights, but the table suggests that their traffic complexities were different. Table 1 shows that Scenario 2 resulted in twice as many altitude advisories as Scenario 1 in the Alt runs (7 vs. 14) and three times as many conflicting EDA advisories as Scenario 1 in the NoAlt runs (2 vs. 6). The Scenario 2 traffic appears to have been more complex to manage than the Scenario 1 traffic.

2. Meet-Time Accuracy

No noticeable effect of the Condition, Scenario, and Run Group on the meet-time accuracy at the meter fix was observed. The average meet-time error was 3.5 seconds, and the standard deviation was 15.3 seconds (*N* = 409). During the testing, 93% of the flights managed to arrive at the meter fix within ±20 seconds of the STA.

Of the flights simulated, 29 fell outside the 20-second meet-time tolerance envelope. Close investigations of these flights revealed the following five reasons for the meet-time deviations: 1) the controller manually intervened near the meter fix to adjust separation at the meter fix; 2) the pilot used the MCP instead of the FMS to comply with a cruise-speed clearance after the EDA advisory speeds were input to the FMS, which caused the FMS-programmed descent speed not to be executed; 3) the arrival flights whose |*Δ*| was less than the set threshold did not receive an EDA advisory, and errors built up during the descent phase, when the EDA advisory was suppressed; 4) the EDA advisory included a sharp turnout for conflict avoidance, and errors in the turn trajectory caused a large meet-time error; and 5) occasionally, EDA was unable to come up with a corrective advisory because of trajectory-computation error, and the meet-time error was not corrected. Reasons 4) and 5) are under investigation. The remaining reasons are related to the EDA concept of operations and will be discussed in the Discussion section. When the meet-time-error data that fell outside the 20-second tolerance envelope were excluded, the average meet-time error was 3.6 seconds, and the standard deviation was reduced to 7.5 seconds.

B. Subjective Data

1. *Real-Time Workload Ratings*

Traffic demand varied during a simulation run. When a run started, there was no traffic in the sectors. Traffic slowly built up, maintained the peak flow rate for a while, and then slowly dissipated. The controllers' workload levels also changed as the traffic load changed. To capture the transitions of the workload levels, three *phase blocks* were defined: early traffic (P1), peak (P2), and late traffic (P3). Fig. 8 plots the raw workload-rating data to illustrate how these phase blocks were defined. In Fig. 8, each sector's plot shows the data from all 12 runs superimposed on each other. (Note that each run had a different participant, condition, and scenario combination, but Fig. 8 mixes them together.) The first workload rating data was always collected at 17 elapsed minutes into the scenario, and then the subsequent data were recorded at seven-minute intervals. The time intervals for the phase blocks, P1, P2, and P3, were defined as shown in Fig. 8. Every phase block contains two samples per run per sector. Each phase block starts at a different time in different sectors, as the downstream sector experienced the same traffic at a later time. The data that do not belong to any phase block were omitted from the analysis described below because 1) these workload ratings were mostly low due to the low traffic load, and 2) the analysis required the sample size in each phase block to be the same (the equal-variance assumption).

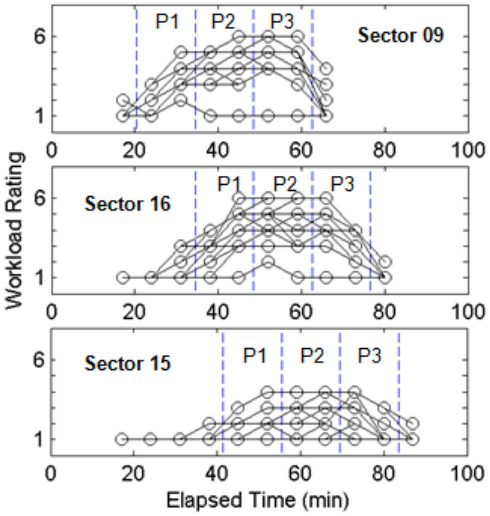


Figure 8. Phase blocks, early traffic (P1), peak (P2), and late traffic (P3) for real-time workload ratings.

A mixed-model Generalized Linear Model (GLM) analysis was applied to the real-time workload ratings. Five main effects—Participant, Sector, Condition, Scenario, and Phase Block—and interaction effects between the Condition and each of the remaining four main effects were included in the model. All other higher-order interaction effects were assumed to be pooled into the error term. The Run-Group effect could not be included in the GLM analysis model because of the run-order change. Participant-sector assignment distributions were uneven between the earlier and the later trials, which prevented computation of the mean square for the Run-Group × Participant interaction. For this reason, the effect of the Run Group will be examined separately later.

The participants and scenarios were considered as being randomly sampled from larger populations; hence, the Participant and Scenario effects were treated as *random effects* in the model, and all other main effects as *fixed effects*.[15] Since the model consisted of more than one random effect, the *quasi-F*-ratio (F^*), instead of the regular F-ratio (F), was used for deriving the p-values for the fixed effects.

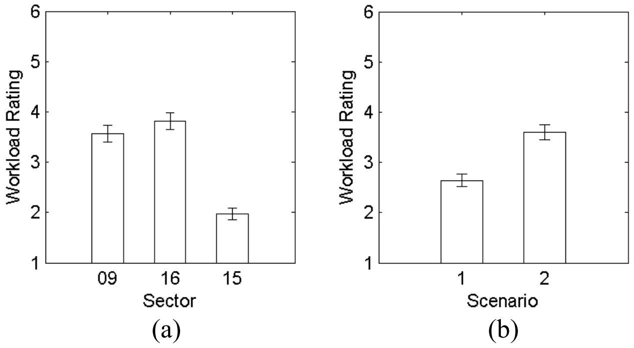


Figure 9. Means and standard errors of real-time workload ratings by (a) Sector and (b) Scenario.

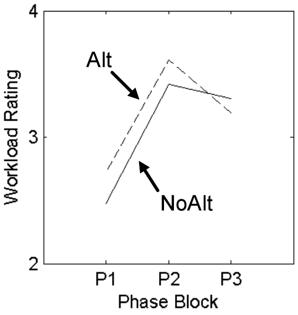


Figure 10. Means of real-time workload ratings by Phase × Condition.

The GLM analysis results revealed that the Sector and Phase-Block × Condition effects were statistically significant ($F^*_{2.64,\ 5.68} = 10.7$, $p = 0.01$; $F^*_{4.85,\ 4.91} = 9.67$, $p = 0.01$; respectively), and the Scenario effect was marginally significant ($F_{1,\ 2} = 11.40$, $p = 0.078$). The Condition main effect did not achieve statistical significance.

To visualize these effects, Fig. 9 plots the means and standard errors of the real-time workload ratings by each sector and scenario. Fig. 9(a) indicates that Sectors 09 and 16 resulted in significantly higher real-time workload ratings than Sector 15, which agrees with the experimenters' observations. Fig. 9(b) shows that Scenario 2 yielded significantly higher real-time workload ratings than Scenario 1, which also agrees with the conflict-avoidance statistics results.

Fig. 10 visualizes the Phase-Block × Condition interaction effect that was also found statistically significant. The plot exhibits an interesting crossing; in phase blocks P1 and P2, the workload ratings were higher in the Alt runs, whereas in P3, the trend reversed, and the workload ratings were slightly higher in the NoAlt runs.

To further investigate why such a crossing trend occurred, the Phase-Block × Condition interaction effect was examined with respect to possible learning or fatigue effect. Fig. 11 shows the same plots shown in Fig. 10 but divided into each Run Group. In Run Group 1 (the first four runs), the graph shows that the Alt workload ratings were higher than the NoAlt workload ratings in all three phases. In Run Group 2 (the second four runs), the trend was similar up to P2 (the peak traffic time in the scenario). Then, the trend reversed in P3 (the later time in the scenario). In Run Group 3 (the last four runs), the NoAlt workload ratings were slightly higher than the Alt workload ratings throughout all the phase blocks. There seem to have been some learning effects, especially for the Alt runs (the dashed lines), and that may be a contributing factor for the significant Phase-Block × Condition interaction effect.

The above results suggested the presence of learning effects in the real-time workload ratings. In fact, it agreed with the experimenters' observation that the workload ratings tended to decrease as the participants experienced more runs. Fig. 12 plots the means and standard errors of the real-time workload ratings by Run Groups. The graph confirms that the real-time workload ratings were significantly reduced in Run Group 3 (the last four runs) compared to those in the earlier run groups.

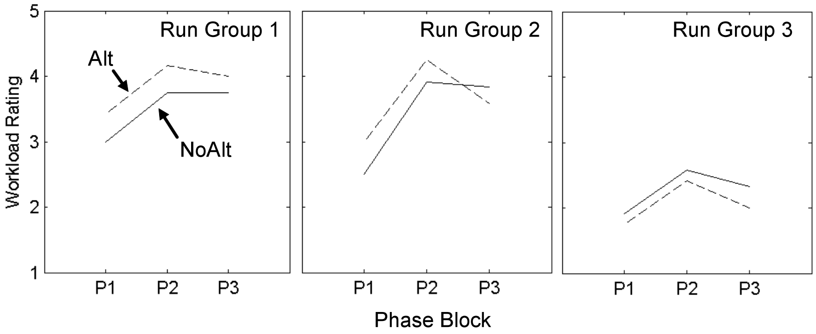


Figure 11. Means of real-time workload ratings by Phase-Block × Condition × Run-Group.

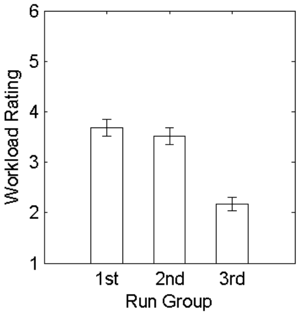


Figure 12. Means and standard errors of real-time workload ratings by Run Group.

2. *Post-Run Modified Bedford Workload Ratings*

The average and maximum workload ratings collected after each run were also subjected to similar mixed-model GLM analyses. The model included the four main effects—Participant, Sector, Condition, and Scenario—and interaction effects between the Condition and each of the remaining three main effects. Again, the Participant and Scenario effects were treated as random effects.

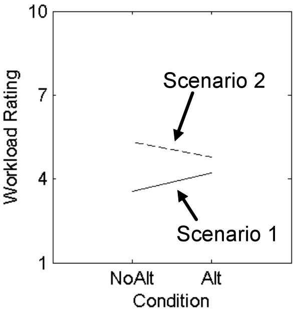
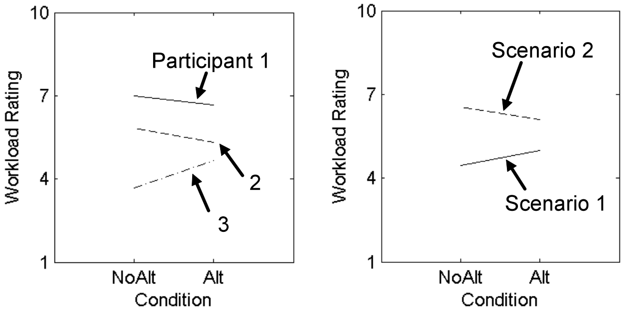


Figure 13. Means of post-run average workload ratings by Scenario × Condition.

In the *average* workload ratings, the Participant, Sector, Scenario, and Scenario × Condition effects were found statistically significant ($F_{2,2}$ = 29.2, p = 0.03; $F^*_{2.39,\,5.73}$ = 11.9, p < 0.01; $F_{1,2}$ = 21.0, p = 0.04; $F_{1,2}$ = 17.3, p = 0.05; respectively). The significant Participant effect was expected because of their different personal biases. The directions of the Sector and Scenario effects were the same as those found in the real-time workload rating analysis (see Fig. 9). To examine the Scenario × Condition effect, Fig. 13 plots the means of the average post-run workload ratings by Scenario × Condition. The plot shows that the Condition effect influenced the average workload ratings differently depending on which scenario was being used. In Scenario 1 (the easier scenario), the Alt workload ratings tended to be slightly higher than the NoAlt workload ratings, whereas in Scenario 2 (the more difficult scenario), the Alt workload ratings tended to be slightly lower than the NoAlt workload ratings.

In the *maximum* workload ratings, the Sector, Participant × Condition, and Scenario × Condition effects were statistically significant ($F^*_{2.38,\,4.85}$ = 9.22, p = 0.02; $F_{2,2}$ = 20.7, p = 0.05; $F_{1,2}$ = 22.9, p = 0.04; respectively). The directions of the Sector effects were the same as those found in the real-time workload rating analysis (see Fig. 9(a)). The plots in Fig. 14 visualize the two interaction effects that were found significant. The Participant × Condition plot in Fig. 14(a) shows that different participants displayed different workload trends; the workload ratings of Participants 1 and 2 decreased slightly in the Alt runs compared with the NoAlt runs, whereas the workload ratings of Participant 3 increased in the Alt runs. The Scenario × Condition plot in Fig. 14(b) indicates similar trends that were observed in the post-run average workload rating results (Fig. 13).

Like the real-time workload ratings, both maximum and average post-run workload ratings showed significant reductions in the last run group. Fig. 15 plots the means and standard errors of the post-run maximum workload ratings by the Run Group. (The plot for the post-run *average* workload ratings is similar to Fig. 15 and, thus, omitted here.)



(a)

(b)

Figure 14. Means of post-run maximum workload ratings by (a) Participant × Condition and (b) Scenario × Condition.

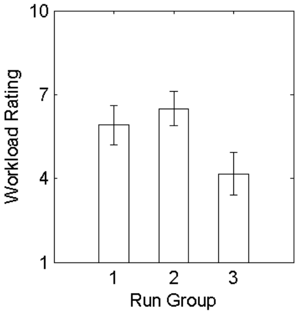


Figure 15. Means and standard errors of post-run maximum workload ratings by Run Group.

3. *Questionnaire and Briefing*

All three participants marked the *liked* checkbox on the Likert preference scale consisting of the five checkboxes: *disliked*, *somewhat disliked*, *neutral*, *somewhat liked*, and *liked*. When the question was asked about any difficulties or procedural concerns with the use of intermediate cruise-altitude advisories, one participant answered that the current user interface needs to be redesigned to match the procedure. The issue was further discussed in the final briefing, and some of the discussions will be described in the next section.

V. Discussion

As expected, the altitude-advisory capability drastically reduced the number of conflicting EDA altitude advisories from 8 in 64 (12.5%) to 1 in 58 (1.7%). Seeing an EDA advisory with a known conflict is always problematic for the controller. Thus, the ability to reduce this number is a great benefit of the altitude-advisory capability.

The counts of the altitude advisories (in the Alt runs) and the conflicting advisories issued (in the NoAlt runs) suggested that the traffic in Scenario 2 was more complex and more difficult to manage than that in Scenario 1. The workload analysis results, which showed that the participants perceived significantly higher workload in Scenario 2 than in Scenario 1, agreed with this insight.

The workload analyses did not find any statistically significant effects of the Condition (i.e., having vs. not having the altitude-advisory capability), which was the main interest of this study. However, the analyses did find statistically significant Scenario × Condition effects in both average and maximum post-run workload ratings. That means having or not having the altitude-advisory capability affected the workload differently in the two scenarios. In the easier scenario (Scenario 1), the workload became higher when the altitude-advisory capability was provided, whereas in the more difficult scenario (Scenario 2, where more conflicts were present), the workload was perceived as lower with the altitude-advisory capability. A possible explanation is that, when the traffic load was lower, the altitude advisory may have felt tedious, whereas when the traffic load was higher, the altitude-advisory capability may have been more helpful. It is interesting that the altitude advisories were issued twice as often in the more difficult scenario as in the easier scenario. Hence, these outcomes together may also imply that the more the controller is exposed to the altitude advisory during a high-workload phase, the more comfortable he/she feels utilizing the advisory to find a suitable altitude.

Another effect that was related to the Condition, and found statistically significant, was the Phase-Block × Condition effect on the real-time workload ratings. It was not clear why the workload suddenly dropped in the last phase of the scenario when the altitude-advisory capability was provided. The graphs of the workload ratings by Phase Block, Condition, and Run Group in Fig. 11 showed that the workload ratings were generally decreasing from the earlier runs toward the later runs, but especially the Alt-run workload ratings (the dashed lines). In Run Group 2, the Alt-run workload ratings suddenly dropped in the last phase of the scenario. One speculation is that this caused the statistically significant Phase-Block × Condition effect. In other words, the effect may have been a byproduct of the overall learning process.

All three types of workload rating data indicated significant reduction in the last four runs (Run Group 3), presumably due to the learning effect. Learning effects could be problematic in analysis when the experiment is not properly counterbalanced to guard against them. Because of the run-order change, the present experiment was not counterbalanced. As mentioned in Section III.F, if the learning effect were present, it was anticipated that the NoAlt runs would have a slight advantage over the Alt runs in the controllers' performance, workload ratings, and preference. However, the results showed no significant effect of the Condition on the workload ratings. That means either that the Condition effect was irrelevant to the workload ratings or that the Alt runs yielded lower workload ratings than the NoAlt runs, but the effect was canceled out by the learning effect. Plus, the Likert scale of preference showed all three controllers liked the altitude-advisory capability. Therefore, in this study, it appears the effect of the run-order change was not large enough to have altered the direction of the results.

It may seem contradictory that the Likert preference scale revealed a favorable rating for the altitude-advisory capability from all of the controllers, whereas the post-run maximum workload ratings showed a statistically significant Participant × Condition effect implying different bias among the participants. However, the reader is reminded that the former was related to the controllers' preference ratings, whereas the latter was based on their ratings on the difficulty of the traffic management tasks. The results imply that even if the controllers are favorable about the concept of the altitude-advisory capability, different controllers may find this added capability easier or more difficult to use. Consequently, if EDA with the altitude-advisory capability is deployed to the field, the amount of controller training required for sufficient adaptation may vary among different controllers.

One way to assist controllers' adaptation to the tool is to improve the user interface design and operation procedures. During the final briefing, the controllers expressed a desire for a capability to accept the profile-descent clearance near the TOD separately from the remainder of the EDA clearance typically accepted far upstream. Although this capability could be useful for any EDA clearances, it would be especially important when an intermediate altitude advisory is included in the EDA clearance. The pilot is generally expected to execute the descent upon receipt of the altitude-change clearance at the descent rate specified in Ref 16, 4-4-10, but his/her adherence is not guaranteed. Therefore, there is always uncertainty in the actual location of the cruise-altitude change and the descent rate. If they are substantially different from what EDA assumed, that may instigate a large meet-time error because of the true-airspeed difference between different altitudes and/or it may cause an unexpected conflict.

Furthermore, there is potential ambiguity when a single clearance contains two vertical clearances, i.e., the intermediate cruise-altitude change and the EDA profile descent to the meter fix. The guiding principle in ATC is that the last clearance has precedence over the previous clearance (Ref. 16, 4-4-10). There is currently no established phraseology rule to issue two vertical clearances at once.[10] For this reason, some pilots may mistakenly think that the intermediate cruise-altitude clearance has been superseded by the subsequent profile clearance and start the profile descent at any point. The simplest method to prevent this mistake is to issue the profile-descent clearance after the cruise-altitude change is completed. This also conforms to the long-held ATC principle of positive control of aircraft (controlling aircraft with clearly defined constraints that ensure separation from other aircraft). However, the current prototype user interface design does not support such a function. As a result, it was often observed that the Sector 09 controller issued an EDA clearance without the profile-descent clearance, pressed the ACCEPT button (as if all the components of the EDA clearance, including the profile-descent clearance, were issued, which was not true), and asked the Sector 16 controller to issue the profile-descent clearance. This was why one of the controllers noted that the user interface design did not match the procedure. The user interface should be redesigned so that such user workarounds do not occur regularly and the procedure adheres to current ATC practice.

Of the arrival flights simulated, 93% reached the meter fix within 20 seconds of their STA. One of the reasons why some flights missed this tolerance level was that the controller manually intervened to change the speeds and/or paths of the flights right before the meter fix to assure the proper separation at the TRACON boundary. Separation assurance is the controller's primary responsibility. Thus, this type of deviation can never be removed, and the system should be designed to tolerate it. Another reason was that the pilots sometimes used the MCP instead of the FMS to comply with a cruise-speed clearance, which resulted in failing to transition to the correct descent speed programmed in the FMS. It was an oversight on the experimenters' side not to emphasize this critical cockpit procedure to the pilot participants. This raises an important issue of pilot training. For EDA to be successful, it somehow needs to be assured that the pilots will not use the MCP for a cruise-speed adjustment after the EDA route and speeds are programmed into the FMS. This may be a challenge, since, as mentioned in the Scope and Limitation of the Study section, the pilots' behaviors in the real world vary far more than those in the laboratory simulation. Meet-time errors were also attributed to flights whose $|\Delta|$s were initially under the set threshold, d, but then built up additional errors during the descent phase, when the EDA advisory was suppressed. In such cases, there was no EDA advisory to fix the problem, and the flights missed the STA by more than 20 seconds. This problem potentially could be resolved by issuing all arrival flights an EDA clearance regardless of their $|\Delta|$.

VI. Conclusion

The intermediate cruise-altitude advisory capability always has been considered the most powerful degree of freedom for searching for a conflict-free EDA advisory. However, additional vertical maneuvers also introduce uncertainties, which may negatively impact controllers' workload and operational efficiency. In this study, the feasibility of adding the altitude-advisory capability to EDA was assessed by conducting a human-in-the-loop simulation experiment with experienced ARTCC sector controllers. A controlled experiment was designed and conducted to examine the effects of an altitude-advisory capability on system performance and the controllers' subjective data. The results showed, as expected, a drop in the number of conflicting EDA advisories when the altitude-advisory capability was available (from 12.5% to 1.7%). Analysis revealed that having the altitude-advisory capability affected the controllers' workload differently in the easier and more difficult traffic scenarios. It was speculated that the altitude-advisory capability may have felt tedious when the traffic situations were less complex, whereas the same capability was perceived as more helpful when the traffic complexity was high. All controllers favored use of the altitude-advisory capability. Yet, even if the controllers approve of the altitude-advisory capability, results suggest that different controllers may exhibit different levels of adaptation, at least initially, in the actual use of the altitude-advisory capability. Based on discussions with controllers, the acceptability of these EDA advisories may be improved with changes to the user interface and inter-sector operational procedures to allow the separation of intermediate cruise-altitude clearances from the primary profile clearance.

Acknowledgments

This study was jointly funded by NASA and the FAA. The authors express thanks to Dr. Charles M. Buntin of the FAA for his overall leadership of the 3DPAM effort. Special thanks go to our controller and pilot participants. Technical support for this experiment was provided by staff members at Sensis Corp., University Affiliated Research Center (UARC), Flight Research Associates, and NASA Ames. Without their hard work, the experiment would not have happened. The authors are also grateful for helpful insights provided by Richard Lanier (FAA) and Todd Farley (NASA).

References

[1] Coppenbarger, R. A., Mead, R. W., and Sweet, D. N., "Field Evaluation of the Tailored Arrivals Concept for Datalink-Enabled Continuous Descent Approach," *the 7th AIAA Aviation Technology, Integration, and Operations Conference (ATIO)*, AIAA, Washington, DC, 2007.

[2] Elmer, K. R., Mead, R., Bailey, L., Cornell, B., Follet, J., Lanier, R. C., et al., "Operational Implementation of Tailored Arrivals at SFO and Expected Environmental Benefits," *the 26th International Congress of the Aeronautical Sciences*, 2008.

[3] Clarke, J.-P. B., Ho, N. T., Ren, L., Brown, J. A., Elmer, K. R., Tong, K.-O., et al., "Continuous Descent Approach: Design and Flight Test for Louisville International Airport," *Journal of Aircraft*, Vol. 41, No. 5, 2004, pp. 1054-1066.

[4] Sprong, K. R., Klein, K. A., Shiotsuki, C., Arrighi, J., and Liu, S., "Analysis of AIRE Continuous Descent Arrival Operations at Atlanta and Miami," *the 27th Digital Avionics Systems Conference*, 2008.

[5] Wat, J., Follet, J., Mead, R., Brown, J., Kok, R., Dijkstra, F., et al., "In Service Demonstration of Advanced Arrival Techniques at Schiphol Airport," *the 6th AIAA Aviation Technology, Integration, and Operations Conference (ATIO)*, AIAA, Washington, DC, 2006.

[6] Manzi, P., "Wide Scale CTA Flight Trials at Stockholm Arlanda Airport," *the 28th Digital Avionics Systems Conference*, 2009.

[7] Erzberger, H., and Nedell, W., "Design of Automated System for Management of Arrival Traffic," NASA TM-102201, 1989.

[8] Swenson, H. N., Hoang, T., Engelland, S., Vincent, D., Sanders, T., Sanford, B., et al., "Design and Operational Evaluation of the Traffic Management Advisor at the Fort Worth Air Route Traffic Control Center," *the 1st USA/Europe Air Traffic Management Research & Development Seminar*, FAA, Washington, DC, 1997.

[9] Coppenbarger, R., Dyer, G., Hayashi, M., Lanier, R., Stell, L., and Sweet, D., "Development and Testing of Automation for Efficient Arrivals in Constrained Airspace," *the 27th International Congress of the Aeronautical Sciences*, 2010.

[10] FAA, "Air Traffic Control," No. 7110.65R, 2006.

[11] Coppenbarger, R. A., Lanier, R., Sweet, D., and Dorsky, S., "Design and Development of the En Route Descent Advisor (EDA) for Conflict-Free Arrival Metering," *the AIAA Guidance, Navigation, and Control Conference*, AIAA, Washington, DC, 2004.

[12] Prevot, T., "Exploring the Many Perspectives of Distributed Air Traffic Management: The Multi Aircraft Control System MACS," *the Human-Computer Interaction in Aeronautics (HCI-Aero)*, 2002.

[13] Hayashi, M., and Lanier, R., "Designing an Efficient Trajectory Clearance Information Format for Air Traffic Controllers' Displays," *the 3rd International Conference on Applied Human Factors and Ergonomics*, 2010.

[14] Huntley, S., Turner, J. W., and Palmer, R., "Flight Technical Error for Category B Non-Precision Approaches and Missed Approaches Using Non-Differential GPS for Course Guidance," U.S. DoT Volpe National Transportation Systems Center, DOT/FAA/RD-93/38, 1993.

[15] Lindman, H. R., *Analysis of Variance in Complex Experimental Designs*. W. H. Freeman and Company, San Francisco, CA, 1974.

[16] FAA, "Aeronautical Information Manual (AIM)," 2010.