

NASA/TM-2017-219662



# An Overview of a Trajectory-Based Solution for En Route and Terminal Area Self-Spacing: Eighth Revision

*Terence S. Abbott*  
*Science Applications International Corporation, Hampton, Virginia*

*Kurt S. Swieringa*  
*Langley Research Center, Hampton, Virginia*

## NASA STI Program . . . in Profile

Since its founding, NASA has been dedicated to the advancement of aeronautics and space science. The NASA scientific and technical information (STI) program plays a key part in helping NASA maintain this important role.

The NASA STI program operates under the auspices of the Agency Chief Information Officer. It collects, organizes, provides for archiving, and disseminates NASA's STI. The NASA STI program provides access to the NTRS Registered and its public interface, the NASA Technical Reports Server, thus providing one of the largest collections of aeronautical and space science STI in the world. Results are published in both non-NASA channels and by NASA in the NASA STI Report Series, which includes the following report types:

- **TECHNICAL PUBLICATION.** Reports of completed research or a major significant phase of research that present the results of NASA Programs and include extensive data or theoretical analysis. Includes compilations of significant scientific and technical data and information deemed to be of continuing reference value. NASA counter-part of peer-reviewed formal professional papers but has less stringent limitations on manuscript length and extent of graphic presentations.
- **TECHNICAL MEMORANDUM.** Scientific and technical findings that are preliminary or of specialized interest, e.g., quick release reports, working papers, and bibliographies that contain minimal annotation. Does not contain extensive analysis.
- **CONTRACTOR REPORT.** Scientific and technical findings by NASA-sponsored contractors and grantees.

- **CONFERENCE PUBLICATION.** Collected papers from scientific and technical conferences, symposia, seminars, or other meetings sponsored or co-sponsored by NASA.
- **SPECIAL PUBLICATION.** Scientific, technical, or historical information from NASA programs, projects, and missions, often concerned with subjects having substantial public interest.
- **TECHNICAL TRANSLATION.** English-language translations of foreign scientific and technical material pertinent to NASA's mission.

Specialized services also include organizing and publishing research results, distributing specialized research announcements and feeds, providing information desk and personal search support, and enabling data exchange services.

For more information about the NASA STI program, see the following:

- Access the NASA STI program home page at <http://www.sti.nasa.gov>
- E-mail your question to [help@sti.nasa.gov](mailto:help@sti.nasa.gov)
- Phone the NASA STI Information Desk at 757-864-9658
- Write to:  
NASA STI Information Desk  
Mail Stop 148  
NASA Langley Research Center  
Hampton, VA 23681-2199

NASA/TM-2017-219662



# An Overview of a Trajectory-Based Solution for En Route and Terminal Area Self-Spacing: Eighth Revision

*Terence A. Abbott*

*Science Applications International Corporation, Hampton, Virginia*

*Kurt A. Swieringa*

*Langley Research Center, Hampton, Virginia*

National Aeronautics and  
Space Administration

Langley Research Center  
Hampton, Virginia 23681-2199

September 2017

The use of trademarks or names of manufacturers in this report is for accurate reporting and does not constitute an official endorsement, either expressed or implied, of such products or manufacturers by the National Aeronautics and Space Administration.

Available from:

NASA STI Program / Mail Stop 148  
NASA Langley Research Center  
Hampton, VA 23681-2199  
Fax: 757-864-6500

## Table of Contents

Definitions.....	3
Acronyms and Nomenclature.....	4
Introduction.....	6
Overview.....	7
Trajectory-Based Operation Algorithm Implementation.....	10
Trajectory Computation.....	11
General.....	11
Final Approach Speed.....	12
Initial Cruise Altitude and Speed.....	13
Top of Descent Monitoring.....	13
Recalculation of the Vertical Path.....	13
Atmospheric Forecast Data.....	13
Trajectory State Data Computation.....	15
Calculation of the TBO Spacing Interval.....	17
TBO Time Interval.....	17
TBO Distance Interval.....	17
TBO Speed Control Law.....	17
TBO Speed Change Minimization and Lag Compensation.....	21
TBO End Speed Estimation.....	21
TBO Nominal Deceleration Roll-In Logic.....	22
TBO Speed Command Quantization.....	23
Look-Ahead Speed Change Inhibit.....	24
Stated-Based, Constant-Time Delay Algorithm.....	25
CTD Design Considerations and General Implementation Notes.....	27
Ownship State and TBO Data.....	29
FIM Clearance.....	29
All Traffic State Data.....	29
Process and Record Traffic Data.....	30
Compute the Nominal Speed and the CAS Chunk Size.....	30
Compute the Spacing Error.....	31
Compute and Limit the Speed Error.....	31
Determine if a Speed Change is Required.....	32
Calculate the Speed Change.....	34
Apply Profile and Configuration Speed Limits.....	38

Limits based on the nominal profile speeds .....	38
Airframe limits .....	39
Procedural limits .....	39
Linearize Intermediate Speed Command .....	39
Condition 1 .....	39
Condition 2.....	39
Default Condition.....	40
Linearization .....	40
Apply Procedural 10,000 ft / 250 kt Speed Limit .....	40
Switch to Final Approach Speed.....	40
Distance-Based, Ground Speed Matching .....	40
Control Law Selection .....	42
Operational Considerations.....	44
Common Speeds After Merging .....	44
Envelope Protection .....	44
Radius-to-Fix Speed Limiting.....	44
Off-Nominal Mach / CAS Transition .....	44
Future Design Considerations.....	44
Estimated Time of Arrival Data from the Traffic to Follow .....	45
Trajectory Data from the Ownship FMS .....	45
Speed Prediction, Speed Quantization, and Lag Compensation Functions .....	45
Summary .....	46
References.....	47

## Definitions

Achieve-by point:	A designated waypoint where the assigned spacing goal is expected to be achieved. It is also the waypoint where the spacing technique should transition from trajectory-based to state-based.
Assigned spacing goal:	The spacing interval, either in time or distance, assigned by ATC in the spacing clearance.
Constant time delay:	A state-based spacing algorithm that typically uses stored state data (time and position) from the designated traffic aircraft to determine the spacing interval. This algorithm also uses the designated traffic's previous or current speed as the nominal speed for the control law.
End speed:	The projected or expected speed command at the end of a change in the commanded speed. I.e., this is the expected, steady-state speed command at the end of a speed change.
End speed command:	See End speed.
Federated system:	An aircraft system that is designed as a stand-alone implementation, i.e., not integrated into the existing avionics.
FIM aircraft or ownship:	The aircraft that is using its FIM equipment, e.g., an ASTAR13 implementation, to conduct a self-spacing operation.
Measure spacing interval:	Using the stored state data (time and position) from the designated traffic aircraft, the measured spacing interval in time is the difference between the current time and the time when the designated traffic aircraft was at the FIM aircraft's proximate position. The measured spacing interval in distance is the along-path distance between the FIM aircraft and the designated traffic aircraft.
Ownship:	See FIM aircraft.
Planned termination point:	A designated waypoint where the FIM operation is to terminate. For distance-based clearances, the FIM operation terminates when the lead aircraft passes this point. For time-based clearances, the FIM operation terminates when the FIM aircraft passes this point.
Spacing error:	The error value that the control law is attempting to minimize. The trajectory-based and state-based algorithms use different methods for calculating the spacing error. Additionally, the spacing error may be defined in either time or distance, depending on the assigned spacing goal.
Speed command:	The continuous, instantaneous speed command provided by the algorithm.
State-based operations:	Based on the previous and current physical states (time, position, ground speed, and ground track angle) of the lead and FIM aircraft. These operations are only valid when the FIM and lead aircraft are on a common path.

Time-to-go:	The time to go from the aircraft's current position to a designated point using the TBO prediction calculation.
Traffic aircraft or Traffic to follow:	The designated traffic against which the spacing aircraft is performing a spacing operation. I.e., the lead aircraft.
Trajectory-based operations:	Operations based on a calculated 4-D path, typically from the aircraft's current position to the runway threshold. The calculated 4-D paths are used to calculate the time-to-go of the FIM and designated traffic aircraft. The difference in the estimated time-to-go of the FIM and designated traffic aircraft to a common point is used to define the spacing error.

## Acronyms and Nomenclature

2D:	2 dimensional; longitudinal and lateral
4D:	4 dimensional; longitudinal, lateral, vertical, and temporal
ABP:	Achieve-by point
ADS-B:	Automatic Dependence Surveillance Broadcast
ASG:	Assigned spacing goal
ASTAR:	Airborne Spacing for Terminal Arrival Routes
ATC:	Air traffic control
ATD-1:	Air Traffic Management Technology Demonstration-1
CAS:	Calibrated airspeed
CTD:	Constant time delay
DTG:	Distance-to-go
ETA:	Estimated time of arrival
FAF:	Final approach fix
FIM:	Flight-deck interval management.
FMS:	Flight management system
ft:	Foot/feet
gs:	Ground speed
IM:	Interval management



kt:	Knots
MSI:	Measure spacing interval
NA:	non-applicable
nmi:	Nautical miles
PTP:	Planned termination point
RTA:	Required time of arrival
SpcE:	Spacing error
STAR:	Standard Terminal Arrival Route
TBO:	Trajectory-based operations
TCP:	Trajectory change point
TOD:	Top-of-descent
TTF:	Traffic to follow
TTG:	Time-to-go

## Abstract

*This paper presents an overview of the eighth revision to an algorithm specifically designed to support NASA's Airborne Precision Spacing concept. This paper supersedes the previous documentation and presents a modification to the algorithm referred to as the Airborne Spacing for Terminal Arrival Routes version 13 (ASTAR13). This airborne self-spacing concept contains both trajectory-based and state-based mechanisms for calculating the speeds required to achieve or maintain a precise spacing interval with another aircraft. The trajectory-based capability allows for spacing operations prior to the aircraft being on a common path. This algorithm was also designed specifically to support a standalone, non-integrated implementation in the spacing aircraft. This current revision to the algorithm supports the evolving industry standards relating to airborne self-spacing.*

## Introduction

Concepts for self-spacing of aircraft operating in an airport terminal area have been under development by the National Aeronautics and Space Administration (NASA) since the 1970's (ref. 1). Interest in these concepts have recently been renewed due to a combination of emerging, enabling technology (Automatic Dependent Surveillance Broadcast data link, ADS-B) and the continued growth in air traffic with the ever increasing demand on airport and runway throughput. Terminal area self-spacing has the potential to provide an increase in runway capacity through an increase in the accuracy of over-the-threshold runway crossing times (ref. 2).

A follow-on to NASA's terminal area in-trail spacing development (refs. 3 and 4) and the initial development of a concept and implementation of a trajectory-based merging capability (ref. 5) was instantiated in an application called the Airborne Spacing for Terminal Arrival Routes, ASTAR. This concept extended the self-spacing capability beyond the terminal area to a point prior to the top of the en route descent. This implementation was a trajectory based concept for the entire arrival spacing operation. The second revised implementation of this algorithm (ref. 6) was designed to support dependent runway operations and was referred to as ASTAR10. This second revision provided the ability to manage spacing against two traffic aircraft, with one of these aircraft operating to a parallel runway. This support for parallel dependent runway operations also included the computation of offset threshold crossing times based on the longitudinal distance offset between the two parallel runways and the ability to use diagonal distance spacing once the aircraft are on parallel approaches (ref. 7). This revision of ASTAR also had a rewritten control law relative to the previous version that was based on the original Advanced Terminal Area Approach Spacing (ATAAS) algorithm (ref. 3). The third revision of ASTAR, referred to as ASTAR11 (ref. 8), was a "lite" version of ASTAR10. In this revision, the ability to space against a second aircraft that is operating to a dependent runway was removed. Additionally, several implementation improvements were made based on observations from a pilot-in-the-loop simulation using ASTAR10. The fourth revision of ASTAR (ref. 9) was another update to ASTAR11. In this revision, the primary focus was to modify ASTAR to better support the flight deck interval management portion of the Air Traffic Management Technology Demonstration-1 (ATD-1) (ref. 10). Part of this modification included a change to the basic control law and the ability to recalculate an aircraft's vertical path if that aircraft is off of its preplanned vertical path. The fifth revision of ASTAR (ref. 11), referred to as ASTAR12, was an update to ASTAR11. In this fifth revision, the focus was to again modify ASTAR to better support the flight deck

interval management portion of ATD-1, where the ATD-1 environment does not support the rich data exchange envisioned for the NextGen environment (ref. 12). Due to this data sharing limitation between the ground systems and the aircraft, a ground speed compensation term was added to the control law to account for some of the operational differences between the scheduled times of arrival used by the ATD-1 ground tools and this airborne tool, where these differences are not communicated to the aircraft due to the voice-only ATC environment in ATD-1. For example, the ground tools typically include a schedule delay during high demand periods at an airport. Without knowledge of this schedule delay, the airborne system may issue speed commands that seem inappropriate to ATC. Additionally, a ground speed feedback term could eliminate some of the spacing error at the final approach fix (FAF) during situations where the traffic aircraft is flying a constant speed offset from the planned, nominal speed.

In addition to the trajectory-based operation (TBO) concepts like ASTAR12, earlier work also examined the use of state-based concepts for Flight-deck Interval Management (FIM), i.e., airborne self-spacing. These state-based concepts do not require any knowledge of the leading aircraft's planned route (refs. 13-16) and typically use saved aircraft state data recorded from the leading aircraft. While these state-based concepts could easily support self-spacing operations, they could not provide the potential benefits in airport noise reduction and reduced fuel usage that are offered by TBO concepts. However, because these state-based concepts do not require knowledge of the leading aircraft's intended flight path, they are not susceptible to path prediction errors encountered in the TBO concepts.

To define FIM equipment requirements for near-term operational environments, the RTCA began developing a document that describes equipment requirements for FIM operations. Partly because the near-term ATC environment does not support the data exchange necessary for an optimized TBO-based FIM operation, the RTCA defined several FIM capabilities and requirements that are better suited for this near-term period. The primary FIM capabilities defined by RTCA require the use of an integrated TBO and state-based concept for most FIM operations. The sixth (ref. 17) revision of ASTAR integrated a new state-based spacing capability into ASTAR's existing TBO concept, with additional logic to determine which of the two control laws to employ. The primary focus of this revision was to better support the evolving industry requirements described in the draft RTCA document titled *Minimum Operational Performance Standards (MOPS) for Flight-deck Interval Management (FIM)* (ref. 18), otherwise known as the FIM MOPS. The seventh (ref. 19) of ASTAR was developed for the *Interval Management Alternative Clearances (IMAC)* experiment (ref. 20). The FIM system requirements for the IMAC experiment were based on the RTCA FIM MOPS, with the MOPS requirements themselves evolving during this development period (ref. 21). The current, eighth revision of ASTAR, described in this document, corrects some performance deficiencies noted in the IMAC experiment and further aligns the design and performance of ASTAR13 with the FIM requirements of the FIM MOPS of reference 21.

## Overview

The RTCA FIM MOPS describes five FIM clearance types: Maintain Current Spacing, Capture Then Maintain Spacing, Achieve-by Then Maintain, Final Approach Spacing, and IM Turn. ASTAR13 is designed to support all of these clearances except for IM Turn. Additionally, the spacing interval that is assigned by ATC, the assigned spacing goal, may be defined in either time or distance. Significant modifications were made in this revision of ASTAR to better support distance-based spacing. In all of the supported clearance types, ASTAR ceases providing speed commands once the ownship passes the planned termination point (PTP), which is a point on the ownship's route that is either designated by ATC or is part of the planned procedure. For all of the distance-based clearances, there is a special sub-mode of the state-based control law, referred to as ground speed matching, that is used after the target aircraft passes the planned termination point. A simple description of each of the supported clearances is as follows:

*Maintain Current Spacing Clearance:* The Maintain Current Spacing clearance, referred to as MAINTAIN, is intended to be used when ATC desires the FIM aircraft to maintain its current relative spacing, either in

time or distance, from its designated lead aircraft. For the Maintain Current Spacing clearance, only the state-based solution is used. The aircraft must be on a common path for this operation. At the initiation of this clearance, the measured spacing interval (MSI) at the time of initiation becomes the assigned spacing goal (ASG). In this operation, only the state-based algorithm is used as the control law.

*Capture then Maintain Spacing Clearance:* From an algorithm standpoint, the Capture then Maintain Spacing Clearance, referred to as CAPTURE, is similar to the Maintain Current Spacing clearance with the exceptions that the ASG is specifically assigned by ATC and that the spacing error is initially allowed to be outside of the acceptable spacing tolerance, i.e., the capture phase. Only the state-based solution is used and therefore the aircraft must be on a common path for this operation.

*Achieve-by Then Maintain Clearance:* A generic scenario for the Achieve-by Then Maintain Clearance, referred to as CROSS, would begin with the FIM and lead aircraft on different routes. ATC would issue the FIM clearance and spacing would begin using the TBO algorithm. For the time-based Achieve-by Then Maintain clearance, ASTAR uses a TBO control law until the Ownship arrives at the designated achieve-by point (ABP) and the state-based control law is used afterward. For the distance-based Achieve-by Then Maintain clearance, ASTAR uses a TBO control law until the Traffic aircraft arrives at the achieve-by point and the state-based control law is used afterward.

*Final Approach Spacing Clearance:* The Final Approach Spacing clearance type, referred to as SPACE, is similar to the Achieve-by Then Maintain clearance, with the achieve-by point and planned termination point co-located for the Final Approach Spacing clearance type. This clearance is only issued when one or both of the aircraft are on the final approach course. For this clearance, the achieve-by point is always the planned termination point, dictating that only the TBO algorithm within ASTAR is used for spacing.

From a design standpoint, ASTAR can conceptually be thought of as two distinct self-spacing, speed control laws with logic to determine which control law is to be invoked. The first control law is the TBO concept of ASTAR12. The second control law is a newer, state-based concept based on a design described in reference 22, which is a constant-time delay (CTD) implementation. The mode selection criteria is shown in table 1 and the fundamental design of ASTAR is shown in figure 1, noting that a special submode of the CTD control law is used for distance-based operations once the target passes the PTP.

Table 1. Control Mode Selection Criteria

Clearance	ASG type	TBO	CTD
MAINTAIN	Time	NA	Until ownship passes PTP
	Distance		Until target passes PTP
CAPTURE	Time	NA	Until ownship passes PTP
	Distance		Until target passes PTP
CROSS	Time	Until ownship passes ABP	Until ownship passes PTP
	Distance	Until target passes ABP	Until target passes PTP
FINAL	Time	Until ownship passes PTP	Until ownship passes PTP
	Distance	Until target passes PTP	Until target passes PTP

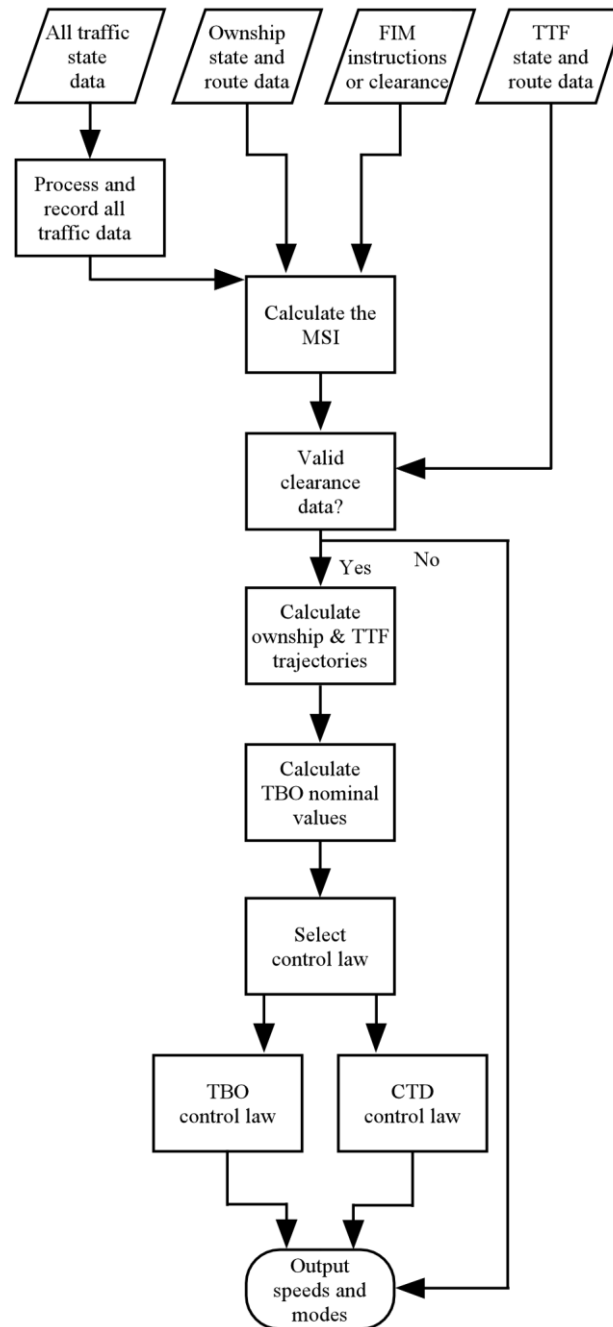


Figure 1. ASTAR13 basic design structure.

A brief summary of the elements shown in figure 1 is provided as follows, with detailed descriptions in the subsequent sections:

- All traffic state data: Input data that contains the ADS-B state data from all of the surrounding aircraft. This would obviously include the state data from the traffic to follow (TTF), if one was selected and its data were available.

- Ownship state and route data: Input data that contains the ownship's inertial and air references data, planned performance data, and augmented route information used in the generation of its 4-D path.
- FIM instruction or clearance data: Input data that contains ASTAR control options and the FIM clearance information. If only the calculation of the MSI is desired, then only the identification of the TTF, i.e., the designated traffic aircraft, is required.
- TTF state and route data: Input data that contains the designated traffic aircraft's inertial data, planned performance data, and augmented route information used in the generation of its 4-D path.
- Process and record all traffic data: Stores approximately 10 minutes of 1 Hz ADS-B state data from all of the surrounding aircraft.
- Calculate the MSI: If a TTF has been identified and both aircraft are on a common path, then calculate both the along-path distance and along-path time between the ownship and the TTF using the state-based data.
- Valid clearance data: As a minimum, valid clearance data would include the FIM clearance type, the identification of the designated traffic aircraft, i.e., the TTF, the identification of the expected routing for the TTF, and the assigned spacing goal. For an Achieve-by Then Maintain Clearance, the identification of the ABP would also be required. The PTP may be identified in the clearance or may be part of the procedure.
- Calculate ownship and TTF trajectories: From the ownship and TTF route information, calculate the planned 4D trajectory for each aircraft.
- Select control law: Depending on the spacing clearance type and the current positions of the aircraft on their respective routes, either the TBO or CTD control law is selected.
- TBO control law: Use the TBO control law to calculate the speed commands.
- CTD control law: Use the CTD control law to calculate the speed commands.
- Output speeds and modes: Output various control mode state data and if appropriate, the speed commands.

## Trajectory-Based Operation Algorithm Implementation

As with the previous versions of ASTAR, the overall concept for a trajectory-based solution for en route and terminal area self-spacing is fairly straightforward. If the 4D trajectory of an aircraft and its position are known, then the aircraft's position on its trajectory can be determined. By knowing the aircraft's position on its trajectory, the aircraft's estimated time-to-go (TTG) to a point is known. To apply this to a time-based, self-spacing concept, a TTG is calculated for both the TTF and for the ownship, noting that the trajectories do not need to be the same. The nominal spacing time,  $t_{nominal}$ , and the spacing time error,  $t_{error}$ , can then be calculated as:

$$t_{nominal} = TTG_{TTF} + \text{spacing goal} ,$$

$$t_{error} = TTG_{ownship} - t_{nominal} ,$$

where the identification of the TTF aircraft and the determination of the spacing goal are performed by ATC.

A required time of arrival (RTA) capability can also be implemented in a manner similar to the traffic spacing technique. In this case,

$$t_{nominal} = RTA - current\ time.$$

From  $t_{error}$ , a speed error value can then be calculated. A conceptual example for the determination of  $t_{error}$  for traffic spacing, i.e.,  $t_{error} = TTG_{ownship} - t_{nominal}$ , is shown in figure 2.

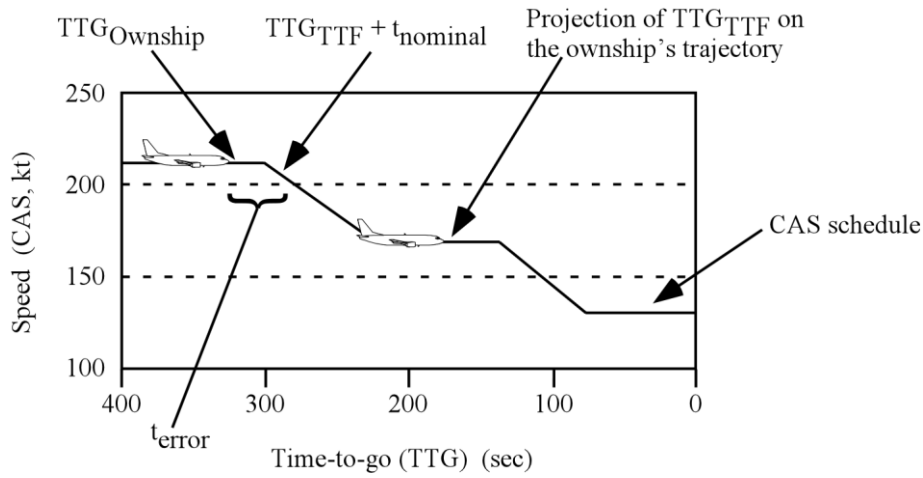


Figure 2. Time error example.

By design, the TBO portion of ASTAR13 is considered an achieve-by algorithm (ref. 23), i.e., it is designed to attain the spacing goal at the ABP. The algorithm does not exactly obtain and maintain the spacing goal until the ownship is near the ABP. Using this control method, the aircraft should be able to fly speeds that are closer to the nominal profile for a longer portion of the operation relative to a more stringent control method that would maintain a fixed spacing interval.

The implementation of the TBO portion of ASTAR13 is comprised of five major elements: trajectory computation, current trajectory state data computation, the calculation of the spacing interval, the speed control law, and speed change minimization; noting that the trajectory and current trajectory state data computations are also performed when the CTD is used for speed control. Details of these elements are provided in subsequent sections.

## Trajectory Computation

### General

The prototype system developed at the NASA Langley Research Center uses a standalone trajectory generator that calculates a full 4D trajectory from a 2D path specification. Reference 24 provides a description of the latest documented version of this algorithm. The trajectory definition begins with a simple 2D path definition along with relevant speed and altitude constraints. The 2D path definition is typically an augmented traditional Standard Terminal Arrival Route (STAR) with a continuous connection to an instrument approach procedure. The trajectory generator then computes a full 4D trajectory defined by a series of Trajectory Change Points (TCP's). This standalone approach was developed for two reasons. First, a near-term implementation separate from the flight management system (FMS) was considered to be more

practical from a development and implementation cost perspective. Second, since ASTAR needs to calculate the trajectory for the TTF, the additional complexity of calculating the trajectory for the ownship was minimal. Neither of these reasons, however, would preclude use of the FMS for providing the ownship trajectory into ASTAR nor the use of a data linked estimated time of arrival (ETA) or TTG from the TTF.

One of the major difficulties in computing a 4D trajectory involves the calculation of the length of the ground path during a turn. The turn radius can change due to the presence of wind or changes in the aircraft's specified speed, affecting the length of the ground path. This change in the path length can then affect the distance to a deceleration point, which then affects the turn radius calculation. To accommodate this interaction, the trajectory calculation uses a multi-pass technique when generating the 4D path with the first pass generating a close approximation to the TCP's based on the computed ground speeds. The following iterations then use the input from the previous pass as a starting point to refine the solution.

In conjunction with the basic 4D calculation, ASTAR preprocesses the trajectory input data. Depending on the situation, ASTAR may update the following generic trajectory parameters: ownship and TTF final approach speeds, initial cruise altitude and speed, differences between the predicted and actual top of descent point, and differences in wind forecast data.

### ***Final Approach Speed***

Two sets of independent options were developed for situations where the end of the flight path was the runway threshold. The first option is designed for the situation where the ABP is the runway threshold, noting that this situation is not valid in the current FIM MOPS. To support this first option, the use of an achieve-by algorithm coupled with the operational requirement to achieve a stabilized approach means that the algorithm must compensate for differences in the TTF's and the ownship's actual final approach speeds. ASTAR has the ability to modify each aircraft's trajectory data by substituting the individual aircraft's planned final approach speed for the trajectory's generic runway threshold crossing speed. By using the individual aircraft's planned final approach speed, the TTG calculations explicitly compensate for the final approach speed differences between the spacing aircraft.

The second option gives ASTAR the ability to support several different operational techniques used to determine where the final approach speed is achieved. In the generic case, the final deceleration starts at a point prior to the runway threshold, causing the aircraft to achieve its final approach speed as it crosses the runway threshold. This baseline technique does not enable stabilized approaches and is not typical for transport aircraft approaches; thus, ASTAR provides three other choices for final approach speed initiation. These choices are:

- Begin the final deceleration at the waypoint that is designated as the final approach fix (FAF). This is the default behavior for the majority of the research studies using ASTAR.
- Begin a deceleration such that the aircraft reaches its final approach speed at a specific altitude, e.g., to be at the final approach speed at 1000 feet (ft) above the runway's elevation, which is also the minimum requirement for instrument approaches in transport aircraft (ref. 25). To support this option, a special waypoint is included in the trajectory input data and is placed between the runway threshold and the FAF waypoint. Only the crossing altitude and crossing speed data are included in this special waypoint's data and the trajectory generator calculates its position on the horizontal portion of the path.
- Begin the final deceleration at a point prior to the FAF such that the aircraft just achieves its final approach speed as it crosses the FAF, noting that transport category aircraft do not normally operate in this manner.



### ***Initial Cruise Altitude and Speed***

A second change that ASTAR may make to the ownship's or TTF's trajectory input data is to substitute the individual aircraft's actual cruise altitude and Mach for the initial, generic altitude and Mach specified in the basic augmented 2D path. This change will only occur at the initiation of a new 3D path and with the aircraft's current altitude and Mach matching a relevant data set being provided to ASTAR. That is, for the ownship, the current altitude and Mach of the aircraft must match a special cruise altitude and Mach data set being sent to ASTAR. For the TTF, this special data set was estimated from the TTF's ADS-B state data.

### ***Top of Descent Monitoring***

The FMS may calculate and the aircraft may fly from a top-of-descent (TOD) point that is appreciably different than the generic TOD estimated by the trajectory generator. Since this difference in the TOD point can introduce a significant error into the estimation of the aircraft's ground speed during this descent and therefore, lead to a significant error in the aircraft's TTG, ASTAR monitors the conformance of the aircraft to its predicted TOD point. If the aircraft begins its descent from cruise before the point that ASTAR predicted, ASTAR will calculate the actual, current descent angle based on this actual TOD and the next altitude crossing restriction, replace the generic descent angle in the augmented 2D path data with this new value, and then recalculate the 4D path. A similar technique is used for a late descent except that ASTAR may recalculate the 4D path several times, depending on how far beyond the originally estimated TOD point that the actual TOD occurs.

### ***Recalculation of the Vertical Path***

Similar to the problem noted in the section *Top of Descent Monitoring*, if an aircraft is significantly far from its planned vertical path, the difference between the planned ground speed and the actual ground speed can be large. This difference in ground speed can then produce a significant error in the aircraft's TTG. Several previous versions of ASTAR allowed the vertical path error, i.e., the difference between the planned altitude along the path and the aircraft's actual altitude at that horizontal position on the path, to reach 6,000 ft, at which time it was assumed that the planned path was no longer valid.

In this version of ASTAR, once an aircraft reaches 4,000 ft of vertical path error, ASTAR will attempt to construct a new 4D path beginning at the aircraft's current estimated position on the original horizontal path. The altitude constraints are then deleted for any downstream waypoint that has an altitude constraint that is higher than the altitude of this new, initial waypoint. Lastly, ASTAR determines if the first altitude constraint following the initial waypoint can be met. If this constraint cannot be met using the original crossing angle, a new crossing angle is calculated based on a linear descent between the initial waypoint and the constraint. This new angle is used in subsequent 4D trajectory calculations.

### ***Atmospheric Forecast Data***

The last modification that ASTAR may apply to the trajectory input data is to modify the original atmospheric forecast data provided to the algorithm. Atmospheric data into and within ASTAR is based on waypoint locations instead of a typical wind grid. It was assumed in the design of ASTAR that a highly developed atmospheric forecast model could be used to provide vertical profile wind data at the waypoint locations. Of special importance to ASTAR would be the atmospheric estimation at the altitude that the trajectory would be crossing the waypoint's position. It was assumed then that the externally provided waypoint atmospheric data would provide reasonably accurate data that would bound the expected waypoint trajectory crossing altitude. Up to 10 altitude atmospheric data sets (altitude, wind direction, wind magnitude, and temperature deviation) per waypoint may be input into ASTAR, along with a barometric setting value for that waypoint. From this initial, external input ASTAR may provide both local and global modifications to the forecast atmospheric data provided to the trajectory generator.

While up to 10 altitude data sets per waypoint may be input into ASTAR, ASTAR itself maintains an internal atmospheric model that uses local aircraft-sensed atmospheric data in addition to the input waypoint atmospheric data. This internal atmospheric model maintains a 1000 ft incremental vertical profile, from 0 ft to 60,000 ft, for every waypoint on all of the paths. This incremental vertical profile contains a "gain" value, the original input wind and temperature forecast for this altitude, a measured wind and temperature for this altitude, and the current estimated wind and temperature forecast for this altitude. Initially, the gain values are all set to zero and the external wind and temperature forecast is used to populate the input wind and temperature forecast for each altitude in its profile. An altitude-based linear interpolation is used to populate the altitudes that do not directly have any input value.

Measured atmospheric values may be adjusted using local or global data. For the local data case, the ownship's wind derivation is used to update the estimated wind forecast. In this case, wind profiles for every waypoint within 50 nautical miles (nmi) of the ownship's horizontal position may be modified. For these waypoints, if the ownship is at or above 12,000 ft, then each of the 1000 ft incremental vertical profile data sets may be modified for altitudes within  $\pm 5000$  ft of the ownship's altitude. For the situation where the ownship is below 12,000 ft, then each of the 1000 ft incremental vertical profile data sets may be modified for altitudes within  $\pm 3000$  ft of the ownship's altitude. Whether a specific 1000 ft incremental vertical profile data set is modified depends on the current gain value for that data set and the gain value computed for the current ownship's position relative to this data set. The ownship's current gain is calculated as follows:

$x_{ownship}$  = relative horizontal position of ownship (in nmi) to the wind profile point and

$z_{ownship}$  = relative vertical position of ownship (in ft) to the wind profile point.

if ( $x_{ownship} > 50$  nmi) then  $gain_{horizontal} = 0$ ,

otherwise  $gain_{horizontal} = 1 - (x_{ownship} / 50$  nmi).

if ( $altitude_{ownship} \geq 12,000$  ft) then

if ( $z_{ownship} > \pm 5000$  ft) then,  $gain_{vertical} = 0$ ,

otherwise  $gain_{vertical} = 1 -$  absolute value of ( $z_{ownship} / 5000$  ft).

otherwise

if ( $z_{ownship} > \pm 3000$  ft) then,  $gain_{vertical} = 0$ ,

otherwise  $gain_{vertical} = 1 -$  absolute value of ( $z_{ownship} / 3000$  ft).

ownship's current gain =  $gain_{horizontal} * gain_{vertical}$ .

If the ownship's computed gain is greater than the gain value for the data set, then the estimated atmospheric data are updated with the new gain value and measured atmospheric data. The new estimated atmospheric data are computed based on a double linear interpolation between the original forecast data and the measured data. The double linear interpolation uses the relative horizontal position, the relative vertical position, and the previously calculated, associated gain values.

ASTAR has the option to include a global wind updating capability in its wind forecast update. In this case, ASTAR uses time correlated ADS-B state vector and air referenced velocity reports from all surrounding aircraft to generate a local wind estimate at each aircraft's position. The estimated wind forecast

is then updated in the manner previously described. This option is not used in the ATD-1 implementation due to communication constraints in the near-term national airspace system.

To exclude erroneous measured wind values which can typically occur when an aircraft is turning, a simple track-file for each aircraft is maintained for each aircraft's true ground track. If this ground track value is changing, based on the aircraft's current and previous track angle values, the aircraft's wind data are excluded from the wind forecast update. In other words, if an aircraft is turning, its wind estimation is not used in the internal forecast.

Once a new internal forecast has been generated, ASTAR selects the best altitudes for each waypoint, based on bounding the trajectory crossing altitude, to update the atmospheric data profile in the trajectory input data. To determine if a new trajectory calculation is required due to a change in the forecast atmospheric data, a waypoint-by-waypoint comparison between the wind data profiles of the current trajectory wind data and the internal forecast data is performed. If there is a significant difference between these data sets, then the trajectory profile atmospheric data are updated using the internal wind forecast data and a trajectory recalculation is performed. The determination of a significant difference between these data sets is based on the following calculation for each wind-altitude point in the trajectory:

*if the difference in wind speed is greater than the variable “s”, then the difference is significant  
or*

*if the wind speed of the trajectory data is greater than s and the difference in wind angle is  
greater than the variable “a”, then the difference is significant, where*

*if the distance to go for the ownship,  $DTG_{ownship}$ , is greater than 200 nmi,  $s = 5$  kts and  $a = 10^\circ$ ,*

*otherwise if  $DTG_{ownship}$  is greater than 20 nmi,  $s = 3$  kts and  $a = 5^\circ$ ,*

*otherwise  $s = 1.5$  kts and  $a = 5^\circ$ .*

Additionally, if the barometric altimeter setting value from the internal forecast data and the trajectory atmospheric data are not equal, a trajectory recalculation is performed. Lastly, if the internal forecast temperature value at the ownship's current position differs from the trajectory temperature data by more than 5 degrees, then the trajectory profile atmospheric data is updated and a trajectory recalculation is performed.

## **Trajectory State Data Computation**

The trajectory state data are the trajectory data, e.g., altitude, CAS, ground speed, and ground track, at a point on the trajectory. By design, speed and altitude changes occur linearly between TCP's as defined by the trajectory generator. Because of this, the determination of a trajectory state based on an aircraft's position is reasonably easy to calculate. First, the determination of the relative segment, i.e., between which two TCP's does the aircraft's position lie, must be calculated. For the example shown in figure 3,  $TCP_1$  is the first TCP on the trajectory, which is typically a high-altitude, cruise waypoint, and  $TCP_n$  is the last TCP, which is typically the runway threshold. Beginning with the first TCP segment, i.e., the segment defined by the TCP pair  $TCP_1$  and  $TCP_2$ , a determination is made if the aircraft's position lies angularly between the two TCP's and if so, is the orthogonal distance (fig. 4) between the aircraft's position and that segment a minimum for the trajectory? In this example, the aircraft is forward of  $TCP_1$  (fig. 5), in the direction of the trajectory's ground path, and behind  $TCP_{i+1}$  (fig. 6).

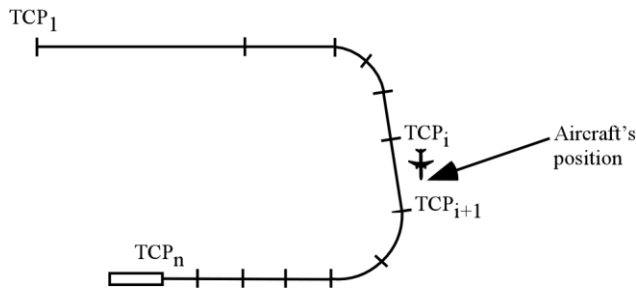


Figure 3. Trajectory state estimation.

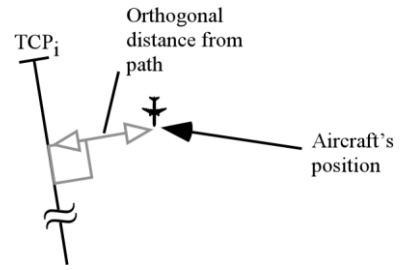


Figure 4. Orthogonal distance measurement.

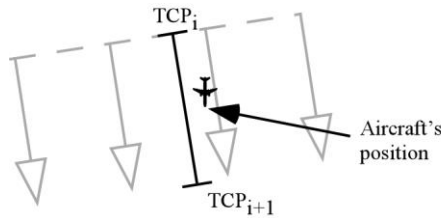


Figure 5. Aircraft's position forward of  $TCP_{i+1}$ .

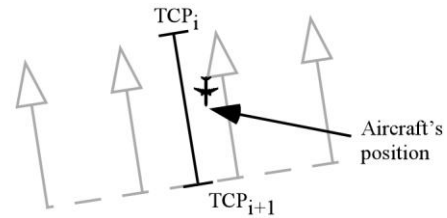


Figure 6. Aircraft's position behind  $TCP_{i+1}$ .

The trajectory state distance, i.e., the distance-to-go (DTG), is then simply calculated from the distance to  $TCP_{i+1}$  ( $DTG_{i+1}$ ) plus the relative distance between  $TCP_{i+1}$  and the projection of the position unto the segment (this relative distance is shown as  $d$  in figure 7). The trajectory altitude is then computed using a simple linear interpolation between the distance between the trajectory state point ( $d$  in figure 7) and  $TCP_{i+1}$  and the distance between  $TCP_i$  and  $TCP_{i+1}$ , i.e.,  $DTG_i - DTG_{i+1}$ . For example, the altitude,  $alt$ , at a position  $p_d$  on the trajectory can be calculated as:

$$x = d / (DTG_i - DTG_{i+1})$$

$$alt_d = alt_{i+1} + x * (alt_i - alt_{i+1})$$

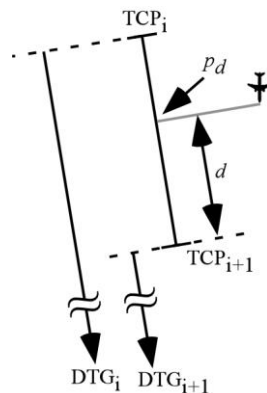


Figure 7. Distance-to-go measurement.

Since speed changes are constant between TCP's, the trajectory state speeds and time at  $p_d$  may be calculated using the linear equations of motion. For example, the CAS at  $p_d$ ,  $CAS_d$ , may be calculated as follows:

$$CAS_d = \sqrt{CAS_{i+1}^2 + x * (CAS_i^2 - CAS_{i+1}^2)}$$

The determination of the trajectory state from the TTG can be computed using a similar technique.

### Calculation of the TBO Spacing Interval

The assigned spacing interval provided by ATC, the assigned spacing goal (ASG), may be given to ASTAR in either time or distance. An explanation of these two spacing interval types is provided in the following three sections.

#### *TBO Time Interval*

The ASG is the interval that ATC would assign for the spacing aircraft to obtain at the ABP against the assigned TTF. The spacing interval for ASTAR is a time-reference interval against a TTF that is typically landing on the same runway as the ownship, although the ATC assigned ABP could be almost any point on the route. The operational goal in this situation is for the ownship to cross the ABP at the ASG time interval after the TTF crossed the ABP.

Internal to the algorithm, once the TTF crosses the ABP, the adjusted spacing time interval for figure 8 is then calculated as:

$$\text{adjusted spacing time interval} = \text{adjusted spacing time interval}_{original} - (\text{current time} - \text{ABP crossing time}_{TTF}),$$

where the *adjusted spacing time interval<sub>original</sub>* is the ASG along with any other adjustments, e.g., runway offset, and *ABP crossing time<sub>TTF</sub>* is the time that the TTF crossed the ABP.

#### *TBO Distance Interval*

In the distance spacing interval case, the operational goal is for the ownship to be at the ASG distance behind the TTF just as the TTF crosses the ABP. For this case, the distance spacing interval is converted to a time-based interval using the speeds associated with the 4D trajectory. The applicable spacing time that is used by the control law is calculated from the 4D trajectory by determining the difference between the ownship's trajectory TTG at the ASG distance from the ABP and the time-to-go on the ownship's trajectory at the ABP. The conversion of this distance-based ASG to a time interval is calculated as:

$$\text{adjusted spacing time interval} = TTG_{ownship \text{ at } (ABP \text{ DTG} + ASG \text{ distance})} - TTG_{ownship \text{ at } ABP}.$$

Once the TTF has crossed the ABP, TBO-based distance spacing is not valid and therefore the calculation of an adjusted spacing interval after the TTF has crossed the ABP is not required.

### TBO Speed Control Law

The use of the trajectory calculations in the speed control law is relatively straightforward. The time error term calculation described previously,

$$t_{error} = TTG_{ownship} - t_{nominal},$$

is used in the speed control law (fig. 8). The overall design concept for this control law was to command speeds within  $\pm 15\%$  of the nominal speeds, providing some level of speed predictability to the flight crew and to ATC. This technique also eliminates the unbounded speed command problem noted in reference 14.

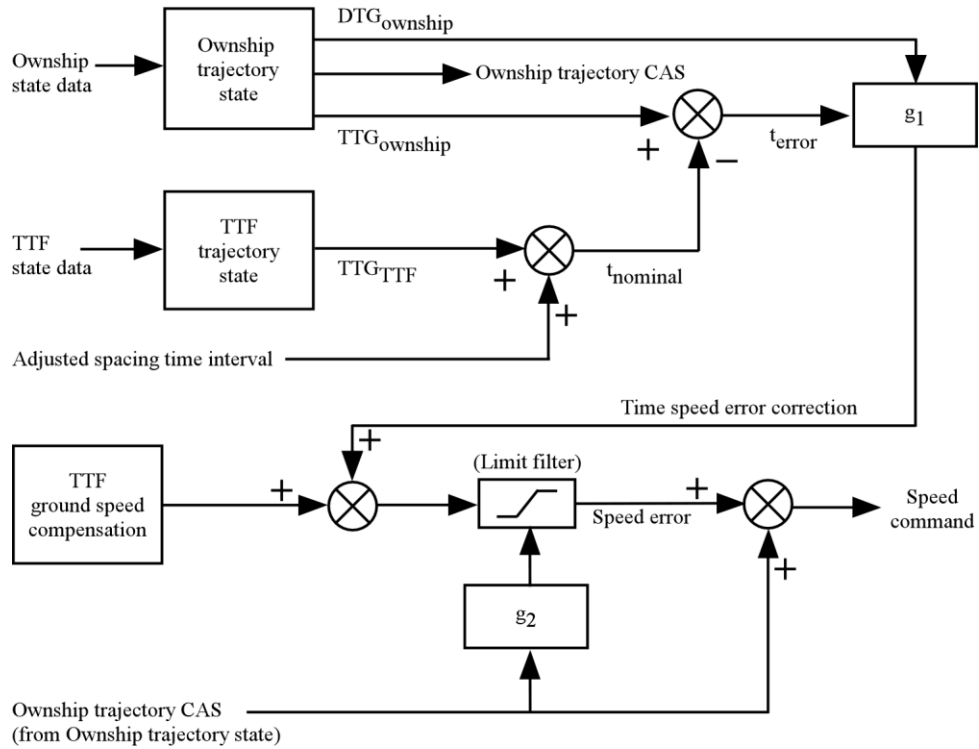


Figure 8. Speed control law.

The value of  $g_1$  (fig. 9), which is used to gain schedule the speed error term, is:

$$\text{if the ASG is time-based, } DTG_{gain} = DTG_{ownship} - DTG_{ABP \text{ to end-of-path}}$$

$$\text{otherwise, } DTG_{gain} = DTG_{ownship} - ASG - DTG_{ABP \text{ to end-of-path}}$$

$$\text{if } (DTG_{gain} > 100 \text{ nmi}) \text{ then, } g_1 = 0.375,$$

$$\text{otherwise if } (DTG_{gain} > 40 \text{ nmi}) \text{ then } g_1 = 0.375 + 0.125 * (100 - DTG_{gain}) / 60,$$

$$\text{otherwise if } (DTG_{gain} > 20 \text{ nmi}) \text{ then } g_1 = 0.5 + 0.5 * (40 - DTG_{gain}) / 20,$$

$$\text{otherwise if } (DTG_{gain} > 5 \text{ nmi}) \text{ then } g_1 = 1.0 + 0.5 * (20 - DTG_{gain}) / 15,$$

$$\text{otherwise } g_1 = 1.5.$$

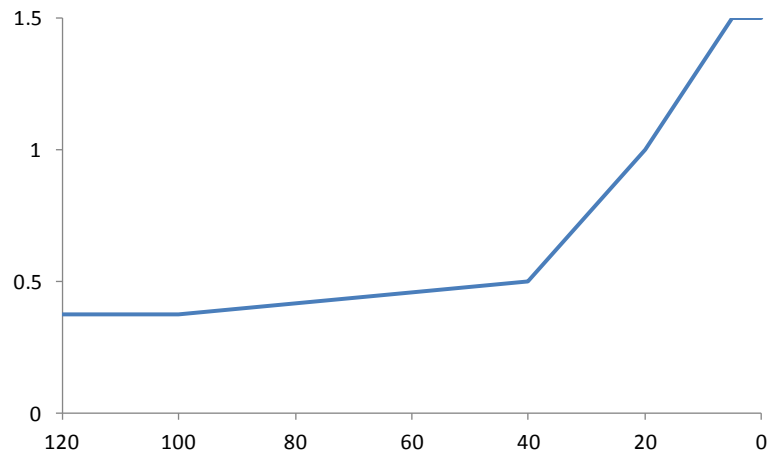


Figure 9. Gain schedule,  $g_1$ .

The value of  $g_2$  is 0.15. This limit filter limits the speed-error value to  $\pm 15\%$  of the ownship's nominal trajectory speed. At 10 nmi, these values produce 1.5 kt of speed correction (fig. 9) for one second of time error.

For the case of the RTA, the nominal spacing time is simply:

$$t_{nominal} = RTA - current\ time$$

where this value is substituted for the nominal spacing time from the TTF data and the TTF ground speed compensation term is set to zero in figure 8.

In ASTAR, ground speed compensation (fig. 10) is used in the TBO control law (fig. 8) as a way to limit the ownship closing too quickly on the TTF when both are relatively far from the airport. Ground speed compensation in this implementation is only used when the TTF is flying a speed that is slower than the nominal ground speed and was meant to compensate for the accepted ATC tendency to slow aircraft below the nominal arrival speeds when they are farther from the airport.

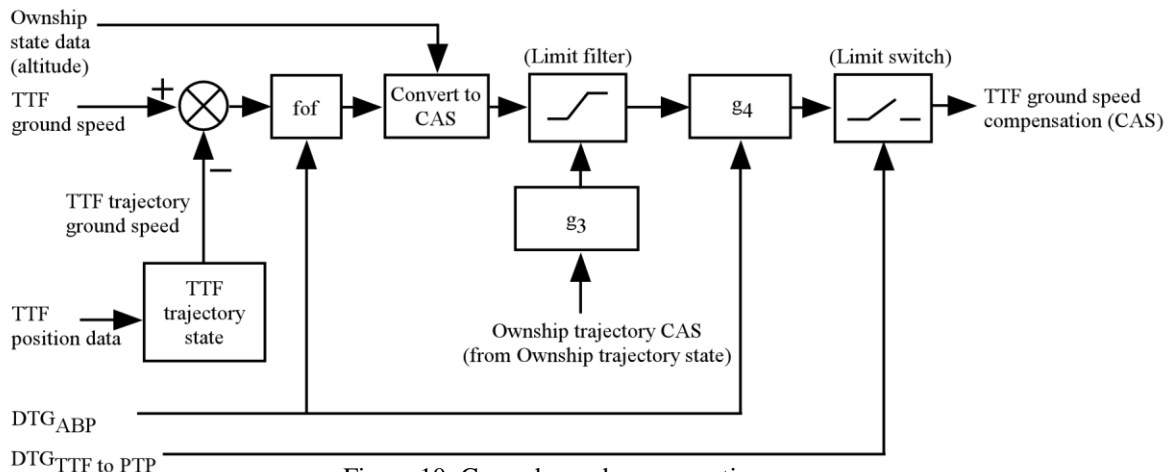


Figure 10. Ground speed compensation.

In this ground speed compensation, the difference between the TTF's actual and planned ground speeds is input into a first order filter, with this first order filter,  $f_{of}$ , being used to dampen out short term variability

and noise within the TTF's ground speed deviation value. The time constant,  $t$ , used in this filter is as follows:

$$\begin{aligned} & \text{if } (DTG_{gain} > 30 \text{ nmi}) \text{ then, } t = 60 \text{ sec,} \\ & \text{otherwise if } (DTG_{gain} \leq 0) \text{ then } t = 0, \\ & \text{otherwise } t = 30 \text{ sec} + DTG_{gain} (\text{sec} / \text{nmi}), \end{aligned}$$

where  $DTG_{gain}$  is the same value as used in the previous speed error term calculation. The output of this first order filter value is then converted to a CAS term using the ownship's current altitude. This value is then gain limited, with a  $g_3$  limit term of +0 to -0.15. This limit filter limits the compensation value to a range between +0% to -15% of the ownship's nominal trajectory speed.

To eliminate the ground speed compensation input into the basic control law as the ownship approaches the ABP, a gain schedule term,  $g_4$ , based on the TTF's distance to the ABP,  $DTG_{TTF \text{ to } ABP}$ , is used. The  $g_4$  term is defined as follows:

$$\begin{aligned} & \text{if } (DTG_{gain} > 40 \text{ nmi}) \text{ then } g_4 = 1.0, \\ & \text{otherwise if } (DTG_{gain} < 20 \text{ nmi}) \text{ then, } g_4 = 0, \\ & \text{otherwise } g_4 = (DTG_{gain} - 20 \text{ nmi}) / (20 \text{ nmi}) \end{aligned}$$

Because the operational envelope for this algorithm can include high altitude Mach portions, both the trajectory calculations and the TBO control law accommodate Mach. If the aircraft is operating in a Mach regime, then the Mach value from the trajectory speed, converted to CAS, is used in the control law. The commanded CAS from the control law is then converted to a Mach command for output. Using this technique, with no spacing error and no ground speed compensation input, the control law provides a single speed command during a constant Mach descent.

Finally, if the planned path terminates at the runway, there comes a point on final approach when the ownship needs to decelerate to its final approach speed and speed changes to correct spacing errors are no longer appropriate. The earlier subsection titled 'Final Approach Speed' describes the operational techniques for terminating active spacing and transitioning to the aircraft's planned final approach speed. An example of how this capability is supported in ASTAR is shown in figure 11. In a purely nominal situation, i.e., where there was no spacing error, the speed command would simply follow the nominal trajectory speed profile with the deceleration to the aircraft's final approach speed beginning at the nominal point on the trajectory. If the commanded speed were faster than the nominal speed (fig. 11), then the deceleration to the final approach speed would need to occur earlier. To accommodate this situation, ASTAR projects the final approach speed deceleration backwards from the nominal beginning of the final deceleration segment. Once the commanded speed point intercepts this deceleration line, ASTAR transitions into a final speed mode and provides a speed command that equals the appropriate speed along this deceleration line. An analogous technique is used for the situation where the commanded speed prior to the final deceleration is slower than the nominal speed (fig. 12). In this case, ASTAR would again maintain the original commanded speed until the commanded speed point intercepts this deceleration line, with the intercept point being after the nominal beginning of the deceleration segment.



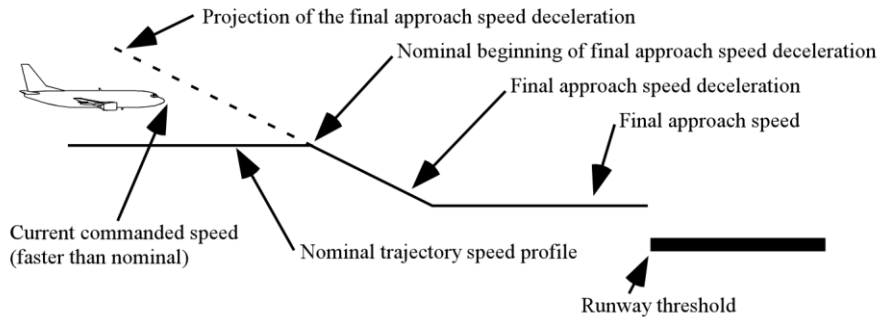


Figure 11. Final approach speed deceleration from an initially faster commanded speed.

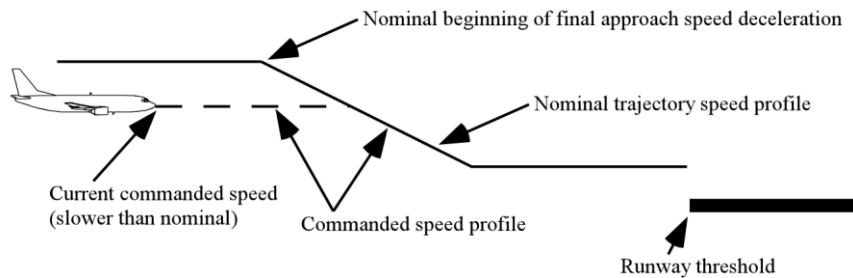


Figure 12. Commanded speed profile from an initially slower commanded speed.

### TBO Speed Change Minimization and Lag Compensation

One of the most significant design requirements for the last three versions of ASTAR was the ability to support a low cost, aircraft retrofit option with very minimal integration with other aircraft systems. In this option, it was assumed that the speed command value would be presented to the pilot and the pilot would then change the speed target of the autothrust system to match the commanded speed from ASTAR or, less likely, directly track the speed command through manipulation of the thrust levers. While this option is probably less than ideal from both a human factors and speed tracking performance perspective, there has been interest from the aviation community in providing a relatively low cost option (ref. 26) for airborne self-spacing. To support this option, from a pilot workload standpoint it was deemed beneficial to minimize the number of speed command changes presented to the pilot. Several capabilities are provided within the algorithm that attempt to balance the number of speed changes against the spacing performance. The implementation of these capabilities has led to a considerable increase in complexity of the ASTAR algorithm.

### TBO End Speed Estimation

In this implementation of ASTAR, the pilot is expected to implement the algorithm's speed command by matching the aircraft's autothrust command to the ASTAR speed command. During a programmed deceleration segment without any spacing error, e.g., a change in the nominal speed profile from 210 kt to 170 kt (fig. 13), the ASTAR speed command would change continuously during the deceleration segment, with the command speed following the nominal speed profile. To reduce pilot workload so that the pilot did not need to continuously monitor the speed command and continuously change the input to the autothrust system, a secondary speed command is output by ASTAR for display to the pilot. This secondary speed command, termed the end speed command, is an estimate of the speed command at the end of the speed change. In the example of figure 13, the end speed command would change from 210 kt to 170 kt as the aircraft reaches the start of the 210 to 170 kt deceleration segment. For long deceleration segments, the end speed command could be used first by the pilot to set the autothrust speed target and then the basic,

instantaneous speed command could be used to modulate the thrust or aircraft's drag devices to better follow the decelerating speed command profile.

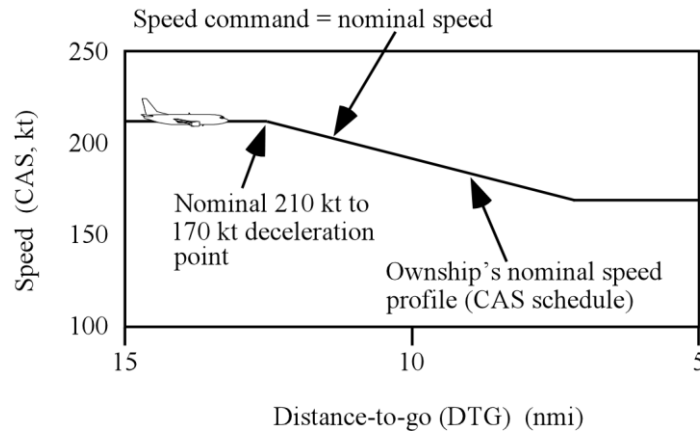


Figure 13. Example speed change with no spacing correction.

A similar situation would occur in the presence of a required speed correction due to a spacing error or RTA adjustment. In figure 14, the nominal speed profile is the same as in figure 13, but there is now a positive 10 kt spacing correction. Prior to the start of the nominal 210 to 170 kt deceleration segment, both the speed command and the end speed command would be 220 kt. At the start of the deceleration segment, the speed command would be 220 kt while the end speed command would change to 180 kt.

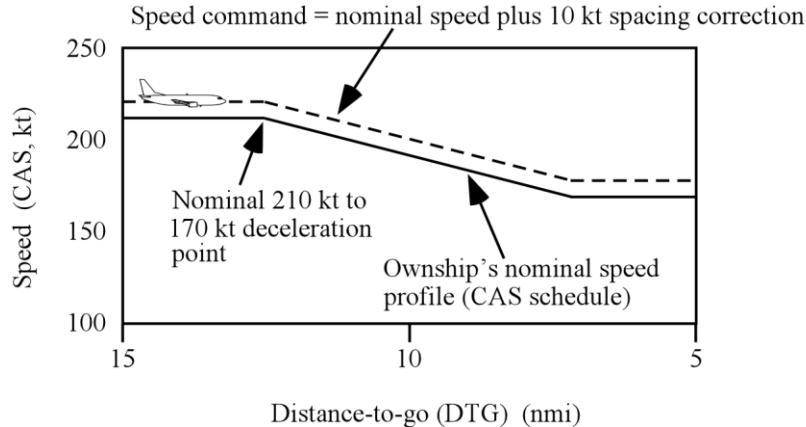


Figure 14. Example speed change with a positive 10kt spacing correction.

### ***TBO Nominal Deceleration Roll-In Logic***

During the initial evaluation of ASTAR10 (ref. 27), it was determined that the lag in response to a speed command change by the simulated aircraft was problematic and contributed to undesirable spacing performance, especially during situations where several aircraft were spacing one after another, i.e., a spacing string. To reduce this problem at the start of a planned deceleration segment in the nominal profile, where this response lag was most apparent, predictive, nominal speed roll-in logic was added to the speed command. An example of a deceleration in the nominal profile without this roll-in logic is shown in figure 15. In this example, there is no speed error, so the instantaneous speed command would match the nominal speed profile. Additionally, the change in the end speed command would occur at the deceleration point on the nominal speed profile. Therefore, at 300 seconds TTG in the example of figure 15, the end speed command would change from 210 kt to 170 kt and the instantaneous speed command would begin to

decrease at a rate equal to the change in the nominal speed profile. Using a 3 second initial look-ahead and a 12 second roll-in half period for the end speed command, the equivalent situation is shown in figure 16 with the roll-in logic. In this figure, the end speed command would change from 210 kt to 170 kt 15 seconds earlier relative to the basic profile. The instantaneous speed command would begin changing speed 3 seconds after the end speed change and would change in a manner such that the nominal deceleration rate and speed would just match the nominal command speed and deceleration rate 24 seconds later.

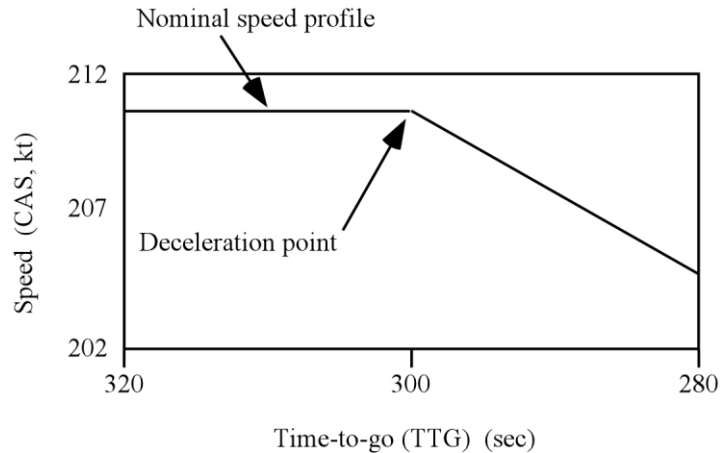


Figure 15. Nominal speed change without roll-in logic.

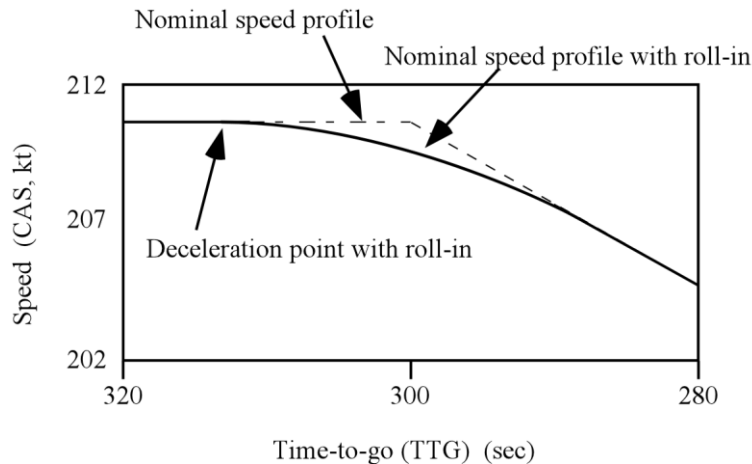


Figure 16. Nominal speed change with roll-in logic.

### ***TBO Speed Command Quantization***

Another means for reducing the number of speed changes in ASTAR was to use a quantization technique on the end speed command and, except during speed changes, on the instantaneous speed command. By applying a quantization to the speed command prior to its output, the end speed command changes only occur in discrete intervals, thus reducing the number of commanded speed changes. For example, if the speed command (fig. 8) was to change from 210 kt to 172 kt and a 5 kt quantization value was used, then the following would occur:

- Immediately prior to the speed change, the output values for both the speed command and the end speed command would be 210 kt.

- At the start of the speed change, the output value for the speed command would slowly begin to decrease, e.g., 209, 208, 207 kt. The output value for the end speed command, because it is being discretized in 5 kt increments by the quantization process, would change to 170 kt.
- At the end of the speed change, the output values for both the speed command and the end speed command would be 170 kt.

Hysteresis was included in the quantization logic to reduce dithering of the end speed command when the command speed is near the breakpoint for the quantization value. The quantization value used in the ATD-1 implementation is either 5 kt or 10 kt, with the value determined as follows:

- The default value is 10 kt.
- If the FIM aircraft is within 5 nmi of the planned termination point and the absolute value of the time error is less than 9 sec, then the quantization value is 5 kt.
- Once the quantization value is 5 kt, it is locked at that value.

### ***Look-Ahead Speed Change Inhibit***

To eliminate a speed command increase prior to a programmed deceleration segment, i.e., where the planned trajectory specifies a deceleration, a look-ahead speed change inhibit option was used. For example, if a 10 second look-ahead interval was used, the algorithm would look ahead by 10 seconds in the nominal speed profile (fig. 17) to determine if a change onto a deceleration segment would occur. Within this 10 second interval, any speed command increase would be inhibited. If the nominal deceleration roll-in logic, described in the previous section, is used, its 12 second roll-in interval would be added to the 10 second look-ahead interval. Thus, the speed change inhibit logic would be applied 22 seconds prior to the deceleration point on the basic, nominal speed profile. The look-ahead inhibit interval was calculated as follows:

*if* ( $DTG_{ownship} > 120 \text{ nmi}$ ) *then*  $t = 60 \text{ sec}$ ,

*otherwise if* ( $DTG_{ownship} > 40 \text{ nmi}$ ) *then*  $t = 30 \text{ sec} + 30 \text{ sec} ((DTG_{ownship} - 40 \text{ nmi}) / 80 \text{ nmi})$ ,

*otherwise if* ( $DTG_{ownship} > 10 \text{ nmi}$ ) *then*  $t = 15 \text{ sec} + 15 \text{ sec} ((DTG_{ownship} - 10 \text{ nmi}) / 30 \text{ nmi})$ ,

*otherwise*  $t = 10 \text{ sec}$ ,

*then the look-ahead interval* =  $t + \text{roll-in interval}$ , with *roll-in interval* = 12 sec for CAS commands, 0 sec for Mach commands.

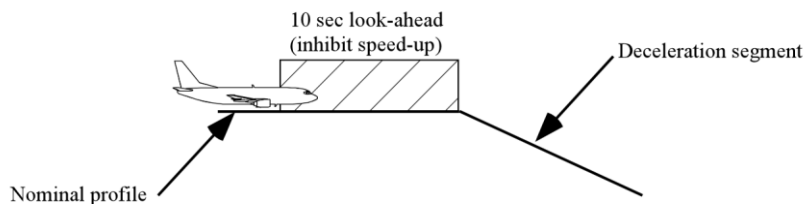


Figure 17. Look-ahead speed-up inhibit.

## Stated-Based, Constant-Time Delay Algorithm

The fundamental concept used for the state-based algorithm in ASTAR13 was derived from previous work at NASA Langley (refs. 13, 15, and 16). The state-based algorithm in ASTAR13 is a type of time-history algorithm referred to as a constant-time delay (CTD) algorithm. This CTD algorithm was originally designed to minimize the measured time spacing interval, which is defined as the difference in time between when the TTF crossed the ownship's current along-path position and the current time. This fundamental concept is shown in figures 18 and 19. In these figures, the TTF is ahead of the FIM or ownship aircraft. The TTF is continuing to reduce its speed and the assigned spacing goal is such that the FIM aircraft is behind its nominal position. In figure 19, the state-based, CTD algorithm would, for this situation, calculate the spacing error, calculate an appropriate speed correction value (i.e., the speed correction due to spacing error), and add this correction value to the traffic's speed history value that occurred at the ownship's current position. This new value would be the resulting intermediate speed command. With no spacing error, the intermediate speed command would obviously be the same as the traffic's speed at that point on the path.

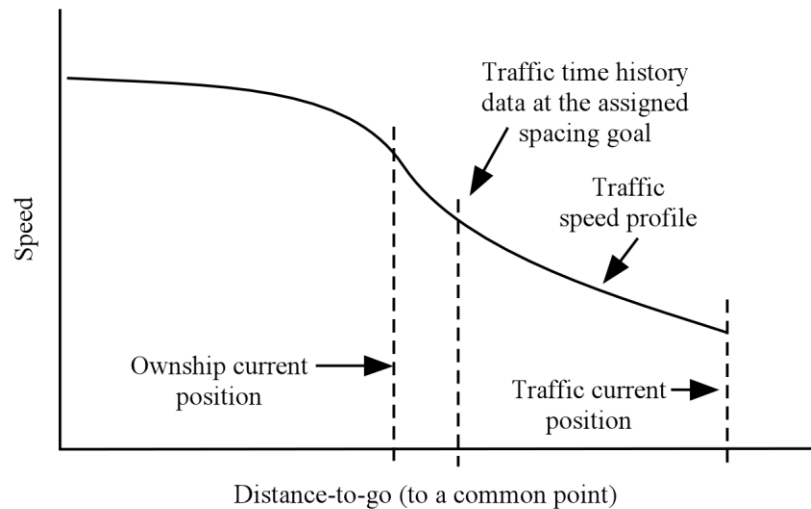


Figure 18. CTD nominal profile data.

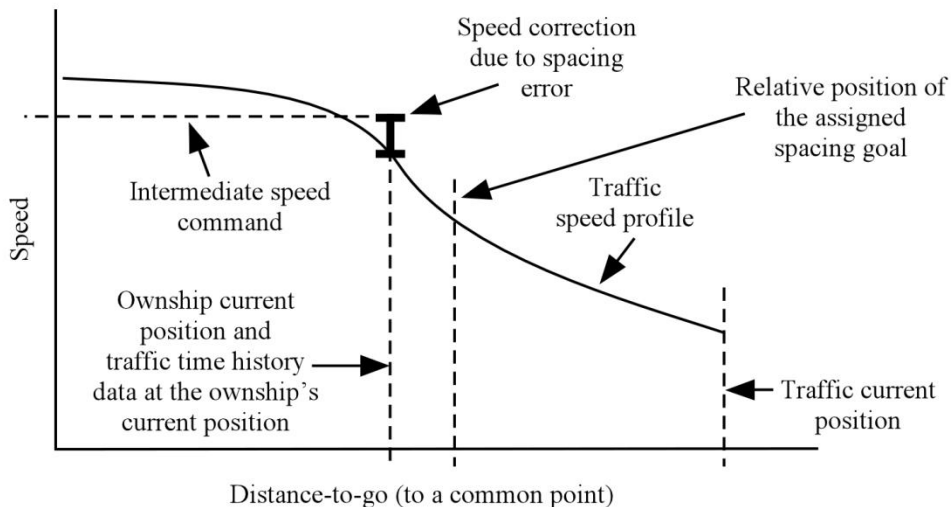


Figure 19. Time-based, CTD spacing error and speed correction.

This latest adaptation of the CTD algorithm includes several new innovations that better support the near-term operational environment and the RTCA FIM MOPS. As part of this MOPS adaptation, the previous portion of the CTD software that supported a distance-based ASG was extensively modified. When the ASG is distance-based, the CTD control is basically a station-keeping algorithm, with the speed correction added to the traffic's current speed to calculate a speed command (fig. 20). It can be thought of as two aircraft connected by a string, with the string length being equal to the desired spacing distance. In this situation, when the TTF changes speed, the FIM aircraft should make the exact corresponding speed change. Unlike the TBO control law or the CTD control law with a time-based ASG, all of which use a nominal or reference speed that is relative to the ownship's position, the nominal speed is simply the traffic's current speed. With no spacing error, the intermediate speed command would be the same as the traffic's current speed.

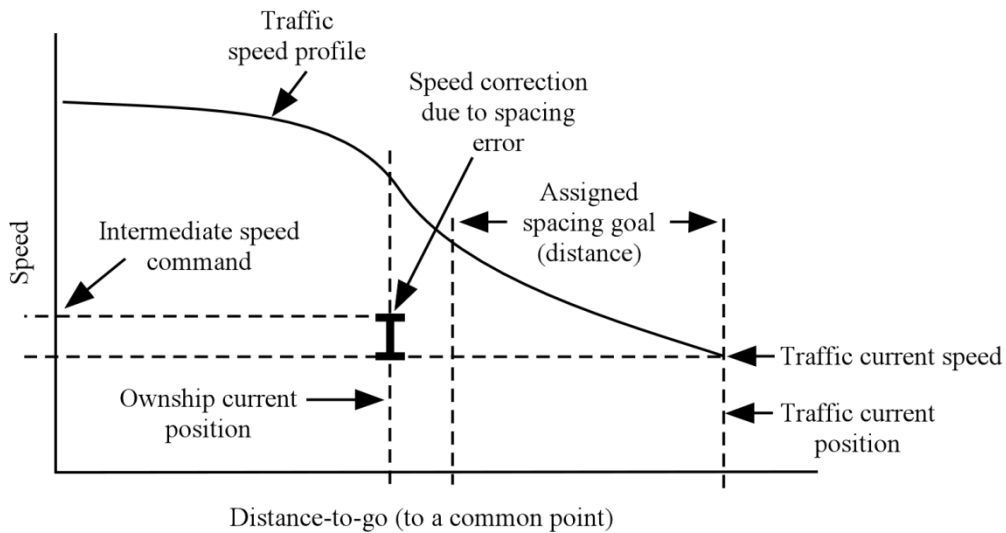


Figure 20. Distance-based, CTD spacing error and speed correction.

A flow diagram of this CTD algorithm is shown in figure 21. The subsequent text will briefly describe several general design considerations for each block in this diagram.

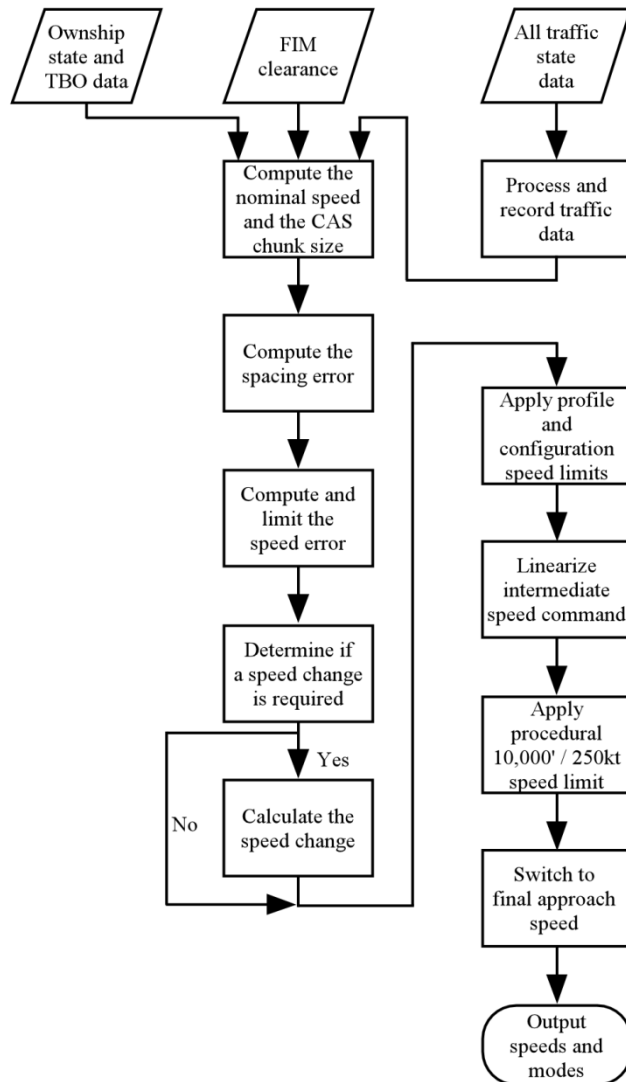


Figure 21. CTD block diagram.

### CTD Design Considerations and General Implementation Notes

One of the more significant differences between the original design of ASTAR and the CTD portion of ASTAR13 is the derivation of when speed changes occur. In the original design, which was only TBO, it was assumed that the actual speed control would be conducted in a manner such that ASTAR would be directly driving the aircraft's autothrust system, i.e., that the ASTAR speed command would be directly integrated in some manner with the aircraft's speed control system. Because of this, what is considered the intermediate or instantaneous speed command is continually calculated, where this speed command could then continuously adjust the autothrust speed. Since the initial development of ASTAR, however, the industry focus and operational testing has moved toward a low cost solution with the FIM equipment no longer expected to be integrated with the aircraft systems. Because of this move from an integrated solution toward a federated, stand-alone implementation, an additional speed command was added to the last two versions of ASTAR (see the *End Speed Estimation* section in this document). The additional speed command was the end commanded speed, which is the projected or expected speed command at the end of a change in the commanded speed. The end commanded speed is expected to reduce pilot workload in the stand-alone implementation if it is changed in discrete increments (see the *Speed Command Quantization* section in this document). It is the determination of the point at which the end command speed quantization

(end command speeds in discrete increments) occurs and the calculation of the CTD algorithm's end commanded speed that is innovative in this solution. These capabilities will be further described in the speed change description sections. A representative plot of the end commanded speed is given in figure 22. This may be compared with the plot of intermediate speed command given in figure 19. Similar to the TBO control law, the CTD control law can accommodate Mach commands. If the aircraft is operating in a Mach regime, then Mach values, converted from their respective CAS values, are used in the control law. For simplicity, these Mach conversions are not shown in this document.

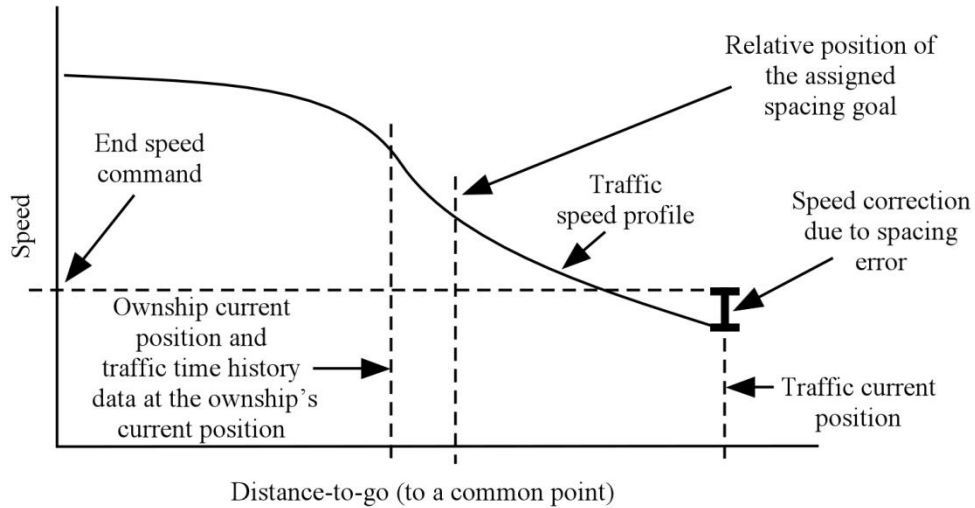


Figure 22. CTD end speed command.

A conversion process that was found to be beneficial during early testing of the CTD algorithm is in regard to how the traffic's ground speed is both averaged and converted to airspeed (CAS). Because the resulting speed commands are required to be CAS values, attention must be taken in the conversion of the ground speeds derived from the traffic's ADS-B data into a CAS values for use by the control law. All ground speed to CAS conversions used by the CTD algorithm use the following method, where the ground speed is from a position  $p$  in the traffic's time history data:

$$CAS = CAS\ conversion( averaged\ ground\ speed_p + headwind_p, altitude, temperature),$$

where  $CAS\ conversion$  is a standard true airspeed to CAS conversion;  $averaged\ ground\ speed$  is a derived and averaged ground speed value from the traffic's time history data at position  $p$  (see section *Process and Record Traffic Data*); the  $headwind_p$  is the current wind speed for the ownship, adjusted for the difference between the current wind angle and the traffic's ground track angle at position  $p$ , such that:

$$headwind_p = ownship's\ current\ wind\ speed * \cosine(ownship's\ current\ wind\ angle - traffic's\ ground\ track\ at\ p);$$

$$altitude = altitude_p + altitude\ bias,$$

where  $altitude_p$  is the altitude value from the traffic's time history data at position  $p$  and the  $altitude\ bias$  is the ownship's altitude minus the traffic's altitude (ownship's altitude - traffic's altitude) at the ownship's position, and  $temperature$  is the air temperature for the ownship.



Data values for variables are retained between iterations of the CTD algorithm. If the initial values for these data values are not explicitly described, then it should be assumed that they are initialized to some reasonable value. Because of the complexity in describing several sections of this CTD algorithm, pseudo-code is included to enhance understandability. Nesting levels in the pseudo-code description are denoted by the level of indentation of the document formatting. Additionally, long sections of logic may end with *end of statements* to enhance the legibility of the text. Key data variables for the CTD in the pseudo-code descriptions are:

*CAS<sub>1</sub>*: the intermediate speed command prior to any limiting or quantization.

*CAS<sub>2</sub>*: a quantized value of *CAS<sub>1</sub>*.

*CAS Step Size*: the quantization size or grain size in the quantization of the command speed.

*Deceleration Start Speed*: the speed at the start of a deceleration segment.

*End Speed Command*: the output end speed command that includes speed limiting.

*Interim Speed Command*: the interim, intermediate speed command prior to speed limiting.

*Interim End Speed Command*: the interim, end speed command prior to speed limiting.

*In Deceleration*: a status variable indicating that the changes to the intermediate speed command are occurring along a decelerating portion of the traffic's speed history profile.

*Nominal History Speed*: the CAS value used as the current nominal speed in the calculation of the *Interim Speed Command*.

*Speed Command*: the output intermediate speed command that includes speed limiting.

*Speed Error*: the CAS correction that is applied to the nominal or profile CAS to calculate the intermediate speed command.

A discussion for using a distance-based clearance is presented at the end of this section. Unless otherwise stated, the subsequent text addresses time-based clearance.

## **Ownship State and TBO Data**

The ownship state data and trajectory data are inputs from the imbedded TBO algorithm. Valid TBO data are required for variables such as the DTG to the PTP and the nominal TBO profile CAS to the ownship's current position.

## **FIM Clearance**

The FIM clearance information defines the information that uniquely specifies the clearance and specifically includes the assigned spacing goal (ASG), which is defined in either time or distance.

## **All Traffic State Data**

The RTCA FIM MOPS allows the traffic aircraft's time history information to be extrapolated when the IM clearance is initiated. This technique, however, will initially introduce large speed command errors if the traffic has not been in constant steady-state flight. One of the innovations in ASTAR13 is that it continuously monitors and records the traffic records for all of the surrounding aircraft. At the initiation of

the FIM clearance, the algorithm locates the basic ADS-B data for the aircraft that is now specified as the TTF in the FIM clearance and populates the CTD traffic history data records with these time-based data sets.

### Process and Record Traffic Data

The CTD algorithm requires time history data from the TTF in order to calculate the measured spacing interval and calculate the speed that the TTF had when it was at the ownship's current position. To ensure that the TTF's time-history information is available when a new IM clearance initiates, ASTAR13 stores time history information for all aircraft that are broadcasting ADS-B data. Traffic data are normally stored at the reception of new ADS-B data, which occurs at approximately 1 Hz. In addition to the traffic's basic state data (data time, latitude, longitude, altitude, ground speed, and ground track), a pseudo distance-flown and an averaged ground speed variable are added to each record. The pseudo distance-flown variable is initialized to 0, with the distance correlated to the first, oldest data point for the traffic. The averaged ground speed variable is an averaging of the past 4 seconds of ground speed data. In lieu of the basic ground speed value, this value is used for all of the ground speed computations in ASTAR13 requiring the traffic's ground speed.

### Compute the Nominal Speed and the CAS Chunk Size

Previous CTD concepts computed the speed command as:

$$\text{Intermediate Speed Command} = \text{Nominal History Speed} + \text{speed correction due to the spacing error},$$

where the *Nominal History Speed* is the traffic's speed history at the ownship's current position. What was found in previous ASTAR experiments, however, was that some speed command anticipation was required to overcome the time lag in the ownship's response to a change in the speed command. Based on previous experimental data, a 15 second lead compensation was found to be adequate. The computation of the *Nominal CAS History Speed* for a time-based ASG is then:

$$\text{Nominal CAS History Speed} = \text{CAS conversion}(p),$$

where  $p$  is traffic's time history data from 15 seconds in front of the ownship's current position. For a distance-based ASG, the *Nominal History Speed* is simply the traffic's current speed, with no lead compensation (fig. 20).

The quantization value, used in the majority of the subsequent CAS calculations, is:

$$\text{CAS Step Size} = 10kt, \text{ which is the default value,}$$

Determine if the FIM aircraft is within 60 sec of the PTP,

*if* ( $TTG_{\text{ownship}} < (TTG_{\text{PTP}} + 60 \text{ sec})$ ) *then*

*if* ( $CtdChunkFlag = true$ )  $CAS \text{ Step Size} = 5kt$

*otherwise if* ( $|Spce| < 3 \text{ sec}$ ) *then*

$CtdChunkFlag = true$

$CAS \text{ Step Size} = 5kt,$

where *CtdChunkFlag* is initialized to false.

### Compute the Spacing Error

If the ASG is time-based, the time spacing error is simply the difference between the MSI and the spacing goal:

$$SpcE = MSI_{time} - ASG_{time},$$

where the MSI is computed as

$$MSI_{time} = \text{current time} - \text{history time when the TTF was at the FIM aircraft's proximate position.}$$

If the ASG is distance-based, a conversion for distance to time is used in the error calculation. The derived time spacing error is calculated as:

$$\text{pseudo-ASG time} = \text{current time} - \text{history time when the TTF was at a history distance equal to the ASG distance behind its current position,}$$

and the spacing error, using the TTG data from the TBO trajectory, is then:

$$SpcE = TTG_{ownship} - (TTG_{TTF} + \text{pseudo-ASG time})$$

The MSI is computed as

$$MSI_{distance} = DTG_{ownship \text{ to PTP}} - DTG_{TTF \text{ to PTP}}.$$

### Compute and Limit the Speed Error

The calculation of the speed error value is based on the time spacing error, *SpcE*, and the gain schedule is based on the FIM aircraft's distance to the planned termination point (PTP) and the magnitude of the spacing error. This gain scheduling, as in the TBO portion of ASTAR13, is designed to reduce the sensitivity of the control law when the FIM aircraft was far from the PTP but provide a highly increased sensitivity when close to the PTP. This sensitivity adjustment is designed to reduce the number of speed command changes while providing high spacing accuracy at the PTP and ensuring that the spacing error is maintained within the required tolerance once captured. The actual gain scheduling logic contains heuristics to eliminate gain value flip-flopping at the schedule break points. The basic gain values are:

*if* ( $|SpcE| > 30 \text{ sec}$ ) *then* gain = 0.5

*otherwise if* ( $|SpcE| > 10 \text{ sec}$ ) *then*

*if* (the FIM aircraft's distance to the PTP > 7.5 nm) gain = 0.5

*otherwise* gain = 1.0

*otherwise*

*if* (the FIM aircraft's distance to the PTP > 7.5 nm) gain = 1.0

*otherwise* gain = 1.5

The speed error value is then:

*if the ASG is time-based, Speed Error = gain(kt / sec) \* SpcE ,*

*otherwise Speed Error = 2 \* gain(kt / sec) \* SpcE ,*

To ensure that the speed correction meets the FIM MOPS requirement to capture the spacing error at a minimum rate 3 sec per minute, the following test is performed:

*if ((Ctd3SecFlag = true) and (|SpcE| < 15 sec)) then Ctd3SecFlag = false*

*otherwise if ((Ctd3SecFlag ≠ true) and (|SpcE| > 20 sec) then Ctd3SecFlag = true,*

where *Ctd3SecFlag* is initialized to false. Then,

*MopsSpd = 0.05 \* Nominal TBO Profile Speed<sub>ownship's position</sub> + 0.5 \* CAS Step Size*

*if ((Ctd3SecFlag = true) and (|Speed Error| < MopsSpd)) then*

*if (Speed Error ≥ 0) Speed Error = MopsSpd*

*otherwise Speed Error = -MopsSpd*

To prevent saturating the calculated speed command, this value for *Speed Error* is then limited to ±33% of the *Nominal CAS History Speed*.

### **Determine if a Speed Change is Required**

In the ASTAR13 CTD algorithm, as in the TBO portion, a significant part of the design was focused on minimizing the number of speed command changes while maintaining an accurate spacing interval. A large part of this design feature occurs in the determination of the need for a speed change, and if a speed change is not required, either provides a smooth transition to or maintains the previously calculated command speed. This process of only changing the speed command if that change is larger than some threshold value, relative to the previous command, is an innovative feature of ASTAR13. The following process is used to determine the speed change requirement:

Initialize the variable for the time that a planned deceleration would end.

*Decel End Time = -1*

If the *In Deceleration* flag is false, determine if a speed change is needed.

*if (In Deceleration ≠ true) then*

*CAS<sub>1</sub> = Nominal CAS History Speed + Speed Error*

Determine the direction of the difference.

*if (CAS<sub>1</sub> > Saved CAS<sub>1</sub>) then Up = true*

*otherwise Up = false,*

Quantization test to determine if there is a change using a 5 kt directional bias if the direction of change and the error are in opposite directions.

*if ((Up = true) and (Speed Error < -5 kt)) then*

$$CAS_2 = \text{round}((CAS_1 - 5 \text{ kt}) / \text{CAS Step Size}) * \text{CAS Step Size}$$

*otherwise if ((Up = false) and (Speed Error > 5 kt))*

$$CAS_2 = \text{round}((CAS_1 + 5 \text{ kt}) / \text{CAS Step Size}) * \text{CAS Step Size}$$

*otherwise*

Use a 2 kt offset to prevent flip-flopping.

$$\text{if } (Up = \text{true}) \text{ then } CAS_2 = \text{round}((CAS_1 - 2 \text{ kt}) / \text{CAS Step Size}) * \text{CAS Step Size}$$

$$\text{otherwise } CAS_2 = \text{round}((CAS_1 + 2 \text{ kt}) / \text{CAS Step Size}) * \text{CAS Step Size}$$

Now determine if a speed change should occur, with logic to reduce the number of speed changes.

*Speed Change = false*

*if (((Up = true) and (CAS<sub>2</sub> > Interim Speed Command)) or  
((Up ≠ true) and (CAS<sub>2</sub> < Interim Speed Command))) Speed Change = true*

Determine if it is desirable to inhibit a normal speed change.

*if (Speed Change) then*

$$\text{limit test} = 0.15 * \text{Nominal CAS History Speed}$$

Do not allow the speed change test value to exceed 22 kt.

$$\text{if } (\text{speed change test} > 22 \text{ kt}) \text{ speed change test} = 22 \text{ kt}$$

Do not change the speed command to a slower value if:

- the proposed speed change is to decrease speed and
- the calculated, proposed speed is slower than the saved commanded speed and
- the spacing error is greater than 10 seconds and
- the absolute value of the difference between the profile CAS value at the time of the saved commanded speed and the current profile speed is less than or equal to *speed change test*.

*if ((Up ≠ true) and  
(CAS<sub>2</sub> < Interim Speed Command) and  
(SpCE > 10 sec) and  
(|Ctd Last Profile CAS - Nominal CAS History Speed| ≤ speed change test) ) then  
Speed Change = false*

Do not change the speed command to a faster value if:

- the proposed speed change is to increase and
- the calculated, proposed speed is faster than the saved commanded speed and

- the spacing error is less than -10 seconds and
- the absolute value of the difference between the profile CAS value at the time of the saved commanded speed and the current profile speed is less than or equal to *speed change test*.

*if ((Up = true) and  
 (CAS<sub>2</sub> > Interim Speed Command) and  
 (SpcE < -10 sec) and  
 (|Ctd Last Profile CAS - Nominal CAS History Speed| <= speed change test) ) then  
 Speed Change = false*

where *Ctd Last Profile CAS* is initialized to the value of *Nominal CAS History Speed*.

*end of statement if (In Deceleration ≠ true)*

To enhance the spacing precision near the PTP, a speed change update may be forced if the spacing error is relatively large. There are two condition sets that will force a speed change update:

- if the control law has been in a deceleration for more than 60 sec and the absolute value of the spacing error is greater than 10 sec or
- if the control law has been in a deceleration for more than 30 sec, the absolute value of the spacing error is greater than 5 sec, and the ownship is within 10 nmi of the PTP.

*if ((In Deceleration = true) and  
 ( ( (current clock time - Deceleration Start Time<sub>1</sub>) > 60 sec) and  
 (|SpcE| > 10 sec) ) or  
 ( (current clock time - Deceleration Start Time<sub>1</sub>) > 30 sec) and  
 (|SpcE| > 5 sec) and  
 (DTGownship ≤ (DTGPTP + 10 nmi)) ) ) then  
 In Deceleration = false*

### **Calculate the Speed Change**

If a speed change is required, based on the state of the *Speed Change* variable determined in the previous subsection, then values for both the interim intermediate speed command, *Interim Speed Command*, and the interim end speed command (see the section *End Speed Estimation*), *Interim End Speed Command*, are calculated. If a speed change is not required, then the previously computed values are retained. The speed command values are computed as follows:

*if ((In Deceleration ≠ true) and (Speed Change)) then*

*Saved CAS<sub>1</sub> = CAS<sub>1</sub>*

*Ctd Last Profile CAS = Nominal CAS History Speed*

Save the speed value before it is changed.

*Deceleration Start Speed = Interim Speed Command*

*Interim Speed Command = CAS<sub>2</sub>*

*Interim End Speed Command = CAS<sub>2</sub>*

Determine if a large deceleration is occurring and if so, estimate the end speed command at the end of the deceleration. This deceleration evaluation process is an innovation designed to minimize the number of speed commands presented to the flight crew.

*if (Up = false) then*

Beginning at the same point used in *Compute the Nominal Speed* for determining the *Nominal History Speed* value, i.e., 15 seconds in front of the ownship's position, determine if the changes in the traffic's history average ground speed values indicate that a deceleration by the TTF is occurring. If so, determine the point in the traffic's history data when the deceleration ends (fig. 23).

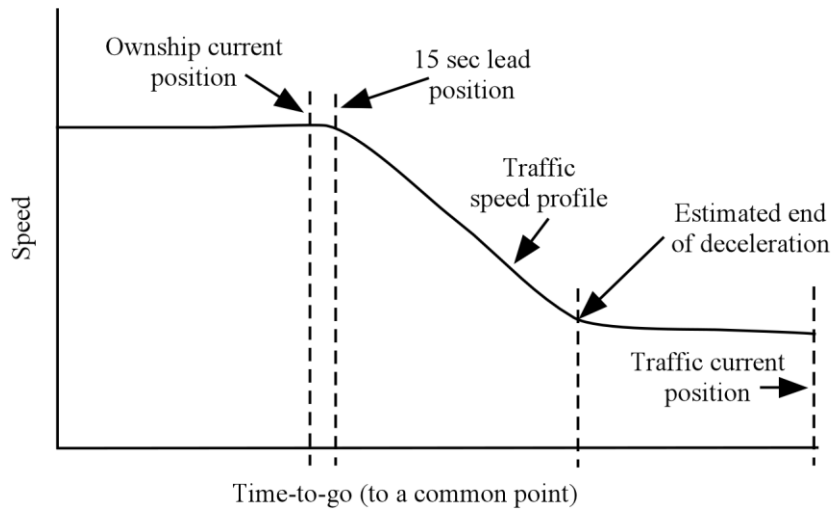


Figure 23. CTD end of deceleration estimation.

This deceleration estimation uses the following process:

The test conditions are initialized such that the data record position in the traffic's history record,  $p$ , is set to the 15 second lead position and the end of the deceleration CAS value, *Deceleration End CAS*, is calculated using the previously described CAS conversion at position *Deceleration Position*.

*Valid Deceleration = false*

$i = \text{Deceleration Position}$

$ibegin = \text{Deceleration Position}$

$last = \text{Deceleration Position}$

*Finished = false*

If the final approach speed option requires crossing the FAF at the FAF crossing speed, break the deceleration estimation at the FAF. This situation would be a result of either of the first two options from the *Final Approach Speed* section. If this is the case, then the variable *LimitAtFaf* is set true, otherwise it is set to false.

*iterate until (Finished = true)*

*i = i + the position 4 history records closer to the traffic's current position*

*if (i > traffic's current position) then Finished = true*

*otherwise*

*Test CAS = CAS conversion(i)*

If the difference between the current test CAS value and the previous minimum CAS value is greater than an average of 0.8 kt / sec, i.e., the traffic's speed has increased, then the deceleration has ended.

*if (Test CAS > (Deceleration End CAS + 0.8 kt/sec \* (traffic history record time at position i - traffic history record time at position last))) then Finished = true*

If the current test CAS value is 0.333 kt / sec slower than the previous minimum CAS value, then the deceleration has started or is continuing.

*if (Test CAS < (Deceleration End CAS - 0.333 kt/sec \* (traffic history record time at position i - traffic history record time at position last))) then*

*Valid Deceleration = true*

*Deceleration Position = i*

*ibegin = i*

*Deceleration End CAS = Test CAS*

*last = i; this is the end of the deceleration estimation process.*

If *LimitAtFaf* is true, test to keep from looking past the FAF during a deceleration.

*if (LimitAtFaf and  
((traffic history record distance at position i -  
traffic history record distance at the Ownship's position) ≥  
(DTG<sub>ownship</sub> - DTG<sub>FAF to end-of-path</sub>))) Finished = true*

If no changes after 10 tests, i.e., ~40 sec., terminate.

*if (i > (ibegin + 10)) Finished = true*

If a profile deceleration exists, then initialize the variables used to calculate the interim speed command during the deceleration.

*if (Valid Deceleration = true) then*

*In Deceleration = true*



*Deceleration Start Time<sub>1</sub> = current clock time*

*Deceleration Start Time<sub>2</sub> = traffic's history data time at ownship's proximate position*

*Saved CAS<sub>1</sub> = Deceleration End CAS + Speed Error*

*end of statement if (Up = false)*

*end of statement if ((In Deceleration ≠ true) and (Speed Change = true))*

If *In Deceleration*, calculate the interim speed values.

*if (In Deceleration = true) then*

Calculate the deceleration initial values at the start of the deceleration.

*if (Deceleration Start Time<sub>1</sub> = current clock time) then*

*s<sub>1</sub> = Deceleration End CAS + Speed Error*

*s<sub>2</sub> = round((s<sub>1</sub> + 2 kt) / CAS Step Size) \* CAS Step Size*

*Interim End Speed Command = s<sub>2</sub>*

Calculate the deceleration time interval.

*t = traffic history time at Deceleration Position - Deceleration Start Time<sub>2</sub>*

*Decel End Time = Deceleration Start Time<sub>1</sub> + t*

Calculate the required deceleration rate, *r*.

*if (t > 0) then*

*r = (Deceleration Start Speed - Interim End Speed Command) / t*

Low limit to at least 0.5 kt / sec.

*if (r < 0.5 kt/sec) then r = 0.5 kt/sec.*

*d = (current clock time - Deceleration Start Time<sub>1</sub>) \* r*

*Interim Speed Command = Deceleration Start Speed - d*

Limit the calculated speed so that it is no slower than the end speed command.

*if (Interim Speed Command < Interim End Speed Command) then*

*Interim Speed Command = Interim End Speed Command*

Has the end of the deceleration time been reached?

*if (current clock time  $\geq$  Decel End Time) then*

*In Deceleration = false*

*Saved CAS<sub>1</sub> = Interim End Speed Command*

*Ctd Last Profile CAS = Nominal CAS History Speed*

*otherwise, the case where  $t \leq 0$*

*Interim Speed Command = Interim End Speed Command*

*In Deceleration = false*

*end of statement if (In Deceleration = true)*

Determine if the speed command values that are output from the algorithm should be updated.

*if ((Speed Change  $\neq$  true) or (In Deceleration = true) or  
(End Speed Command  $\neq$  Interim End Speed Command)) then*

*Speed Command = Interim Speed Command*

*End Speed Command = Interim End Speed Command*

### **Apply Profile and Configuration Speed Limits**

The CTD speed commands, *Speed Command* and *End Speed Command*, may be limited by various speed restrictions, including procedural and airframe limitations. These limiting functions are similar to the comparable functions in the TBO algorithm. The limiting factors in this algorithm include:

#### ***Limits based on the nominal profile speeds***

The CTD speed commands are limited by the TBO nominal profile speeds such that the CTD command speeds remain within  $\pm 15\%$  of the related TBO nominal profile speeds. For the *Speed Command*, its value is limited to  $\pm 15\%$  of the ownship's current trajectory CAS. For the *End Speed Command*, its value is usually limited to  $\pm 15\%$  of the ownship's projected trajectory CAS at the end of a planned deceleration, if one exists, or the ownship's current trajectory CAS otherwise. If this latter limitation is applied, then the resulting *End Speed Command* value will be quantized using *CAS Step Size*. The determination of the limit value, *s*, used in the *End Speed Command* calculation is determined as follows:

*if (In Deceleration  $\neq$  true) then  $s = \text{Nominal TBO Profile Speed}_{\text{ownship's position}}$*

*otherwise if (Nominal TBO Profile InDecel = true) then*

Both profiles have decelerations. If the CTD deceleration ends before the TBO deceleration, calculate the TBO profile speed at the CDT end point and use that CAS value as the limit.

*$d = \text{DTG}_{\text{ownship}} - \text{Distance from ownship to CTD deceleration end}$ ,*

where *d* is the DTG to the CTD deceleration point.

*if ( $d > \text{DTG}_{\text{end of TBO deceleration}}$ ) then  $s = \text{Nominal TBO Profile Speed}_{\text{at DTG} = d}$ ,*

otherwise  $s = \text{Nominal TBO Profile Speed}_{CTD \text{ deceleration end}}$ .

otherwise  $s = \text{Nominal TBO Profile Speed}_{CTD \text{ deceleration end}}$ .

### **Airframe limits**

Airframe limits such as the maximum operating limit,  $V_{mo}$ , or maximum flaps limits may be input in ASTAR13. If airframe limits are used, then both the *Speed Command* and the *End Speed Command* values may be limited by these limit values. If these limitations are applied, then the resulting *End Speed Command* value will be quantized using *CAS Step Size*.

### **Procedural limits**

Procedural limits are imposed for RNAV segments that include maximum speeds. If either the *Speed Command* or the *End Speed Command* is projected to be above the RNAV speed limit for a waypoint, based on the maximum of the current deceleration rate or a 0.75 kt / sec deceleration, then a speed reduction is implemented so that the speed at the waypoint does not exceed the speed limit. In this situation, the *End Speed Command* is set to the RNAV speed limit and the *Speed Command* is linearly reduced from the speed at the initiation of this deceleration to the limit speed at the waypoint.

### **Linearize Intermediate Speed Command**

To aid the pilot in transitioning to a new commanded speed, the intermediate speed command, *Speed Command*, is designed to provide a smooth and continuous series of speed commands as the algorithm transitions between different values of *End Speed Command*. As such, the *Speed Command* may be modified under the following two conditions:

#### **Condition 1**

If the *Speed Command* is not equal to the *End Speed Command* and the value of the *Speed Command* is not changing, e.g., *In Deceleration*  $\neq$  *true*, then apply a change at a constant rate. This condition is applied for the situation where the changed in the *Speed Command* has "stalled." This process would be performed as follows:

Assume that the value *NeedLinearization* is initialized to *false*,  $time_{previous}$  is the time at the previous iteration of the algorithm,  $time$  is the time at the current iteration of the algorithm,  $Speed\ Command_{previous}$  is the value of the *Speed Command* at the previous iteration of the algorithm, and *Speed Command* is the current value of the intermediate speed command. Then,

*if* (*(In Deceleration*  $\neq$  *true)* and  
*(Speed Command*  $\neq$  *End Speed Command)* and  
*(Speed Command*<sub>previous</sub> = *Speed Command*)*) then* *NeedLinearization* = *true*

#### **Condition 2**

If the *Speed Command* is not equal to the *End Speed Command* and the value of the *Speed Command* has changed by more than 2 kt/sec, then apply a change at a smaller rate. This process is to eliminate impulse changes in the speed and would be performed as follows:

The same variables as noted in *Condition 1* are used here. Then,

$rate = (Speed\ Command - Speed\ Command_{previous}) / (time - time_{previous})$

*if* ( $rate > 2 \text{ kt/sec}$ ) *then* *NeedLinearization* = *true*

### **Default Condition**

If *NeedLinearization* is not set to *true* in either of the two previous test conditions, then a test to determine if linearization is no longer required is performed as follows:

*if (Speed Command = Speed Command<sub>previous</sub>) then NeedLinearization = false*

### **Linearization**

If *NeedLinearization* is *true*, then the linearization calculations are performed as follows:

*if (NeedLinearization = true) then*

change the Speed Command by 1 kt / sec starting with the slowing condition.

*if (Speed Command<sub>previous</sub> > End Speed Command) then*

*Speed Command = Speed Command<sub>previous</sub> - 1.0 kt/sec \* (time - time<sub>previous</sub>)*

Limit any undershoot.

*if (Speed Command < End Speed Command) then*

*Speed Command = End Speed Command*

*otherwise*

*Speed Command = Speed Command<sub>previous</sub> + 1.0 kt/sec \* (time - time<sub>previous</sub>)*

Limit any overshoot.

*if (Speed Command > End Speed Command) then*

*Speed Command = End Speed Command*

### **Apply Procedural 10,000 ft / 250 kt Speed Limit**

The 250 kt speed limit below 10,000 ft is implemented such that once the ownship is projected to descent below 10,000 ft within the next 60 seconds, a limiting process is applied if the command speeds crossing 10,000 ft are projected to be above 250 kt. If this condition does exist, the *Speed Command* is linearly ramped such that it reaches 250 kt at the 10,000 ft projection point. The *End Speed Command* value is immediately set to 250 kt.

### **Switch to Final Approach Speed**

This uses the same logic that is in the previously described TBO algorithm to determine when to terminate the spacing control law and switch into the final deceleration speed logic. If the switch is made to the final approach speed logic, then the *Speed Command* and *End Speed Command* values will be modified to satisfy the final approach speed requirements.

## **Distance-Based, Ground Speed Matching**

As a requirement of the RTCA FIM MOPS, there is a special speed control submode for distance-based operations that is employed between the times that the TTF reaches the PTP and the ownship reaches the PTP. In this situation, the command speed is no longer affected by the spacing error and is based solely on

the TTF's ground speed such that the command speed when the ownship reaches the PTP is within 10 kts of the TTF's ground speed at this point. The speed command for this submode is conceptually calculated as follows:

*Initial conditions:*

*Speed Command<sub>previous</sub> =  
Speed Command from the appropriate control law prior to the TTF reaching the PTP*

*Last Time = the last clock time value prior to the TTF reaching the PTP*

Calculate a CAS value  $s$  that is based on the TTF's ground speed at the time when the TTF reached the PTP, using the ownship's current wind speed and direction and at an altitude equal to the TTF's altitude at the time when the TTF reached the PTP plus the ownship's current vertical error.

$s = CAS\ conversion(Ground\ Speed_{TTF\ at\ DTG = PTP} + headwind,$   
 $Altitude_{TTF\ at\ DTG = PTP} + Vertical\ Error_{ownship},$   
 $temperature),$

where *headwind* is the current wind speed for the ownship, adjusted for the difference between the current wind angle and the traffic's ground track angle at the PTP.

Calculate the speed command such that the command is equal to the TTF's ground speed at a time no later than 30 seconds prior to the ownship crossing the PTP.

$t = TTG_{ownship} - TTG_{TTF\ at\ PTP} - 30\ sec$

*if* ( $t > 0$ ) *then*

$rate = (Speed\ Command_{previous} - s) / t$

*if* ( $Speed\ Command_{previous} > s$ ) *then*

This is a deceleration, decelerate at no less than 0.75 kt / sec.

*if* ( $rate < 0.75$ )  $rate = 0.75$

$s = Speed\ Command_{previous} - r * (current\ time - Last\ Time)$

$Speed\ Command = s$

$Speed\ Command_{previous} = Speed\ Command$

$Last\ Time = current\ time$

The value of the speed command is then limited and quantized by the means described in the section titled *Stated-Based, Constant-Time Delay Algorithm*.

## Control Law Selection

Figure 24 describes the control law selection logic that is used to determine whether the TBO or the CTD control algorithm is used to calculate the speed guidance. A description of the input variables and logic for the control law selection process are as follows:

Ownship valid: The ownship's state and trajectory data are valid and the aircraft is on its planned path.

FIM mode: If the overall control is one of the various FIM clearance options or an RTA.

TTF valid: The TTF's state and trajectory data are valid and the aircraft is on its planned path.

Achieve-by and Maintain clearance: If the FIM clearance is the Achieve-by and Maintain option.

Time-based clearance: The assign spacing goal is in time versus distance.

Ownship past ABP: The ownship is past the achieve-by point.

TTF past ABP: The TTF is past the achieve-by point.

TTF past PTP: The TTF is past the planned termination point.

Common path: The history data from the TTF's state data and the ownship's current position relative to that data are used to determine if the ownship is on a common path with the TTF. The processing for the value of this variable is performed external to the control law selection process

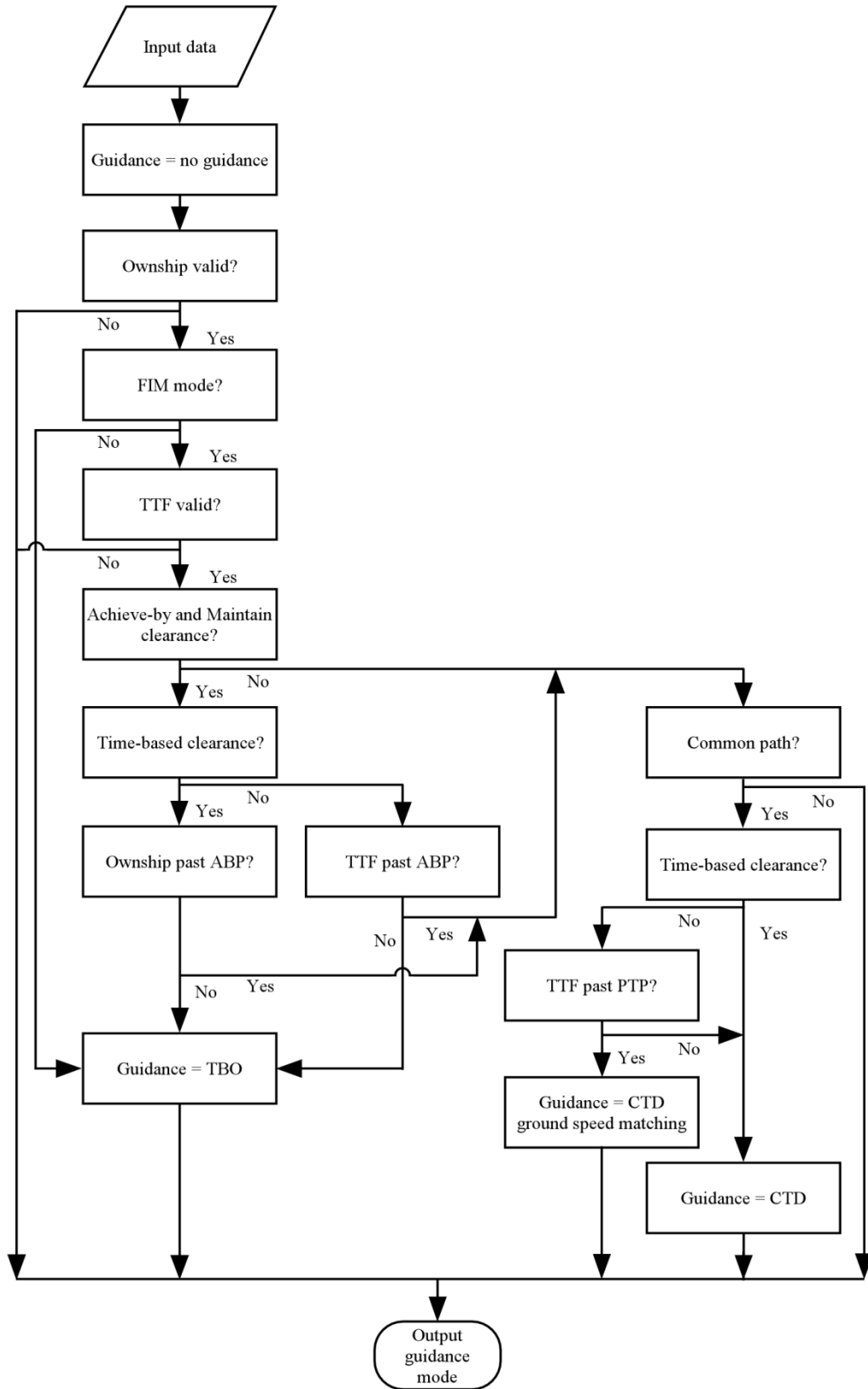


Figure 24. Control law selection diagram

## Operational Considerations

### Common Speeds After Merging

The potential for the loss of separation or less than operationally desirable separation distances between the ownship and the TTF can be minimized by the design of the speed profiles on the respective 2D paths. The speeds specified in the path definitions at and after the point where the paths join in the horizontal plane must be the same speeds (fig. 25). That is, common path points must have common speeds.

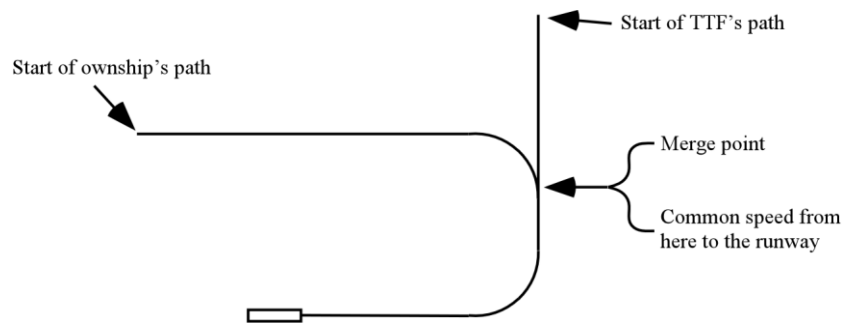


Figure 25. Example of common speeds after the merge point.

### Envelope Protection

Since the speed command value from ASTAR could be used to directly drive an autothrust system, speed envelope limiting can optionally be provided by the algorithm. To invoke this feature, the maximum and minimum desired speed values, both Mach and CAS, are input into ASTAR. These input limit speeds are usually based on the design limiting speed, the maximum gust penetration speed, the maximum flap extended speed for the current flap setting, and minimum maneuvering speed. The algorithm then limits the command speed to remain within these values. When the command speed is limited, the algorithm sets an output flag indicating this limiting condition.

### Radius-to-Fix Speed Limiting

ASTAR supports maximum speed limiting for radius-to-fix (RF), constant radius turn segments. For path segments that include this type of speed constraint, ASTAR would project ahead from the ownship's current position and determine if a deceleration to the speed constraint value must occur prior to reaching this path segment. If a deceleration was required, the speed command would be modified to meet this crossing speed. Once the aircraft was within the path segment, the command speeds would be limited to this speed constraint value.

### Off-Nominal Mach / CAS Transition

The algorithm provides both Mach and CAS speed command values and a Mach/CAS flag indicating which of these values, Mach or CAS, is appropriate for use relative to the aircraft's current flight conditions. While the 4D trajectory data provides the nominal altitude value for the Mach to CAS transition, this altitude value is only valid if the aircraft is exactly on the planned vertical path from the 4D trajectory and is at the nominal Mach. Because these conditions are not generally true, e.g., the Mach speed command is slower than the nominal value to correct a spacing error, ASTAR computes the Mach to CAS transition altitude for the current commanded speeds. Once the aircraft descends below this altitude, the algorithm transitions to a CAS command for the remainder of the operation.

### Future Design Considerations

Even with the design requirement to support a low cost, aircraft retrofit option with very minimal integration with other aircraft systems, several modifications could be made to ASTAR to reduce its design



and implementation complexity along with supporting greater operational viability. Two of these modifications involve the calculations for the trajectory data and the third modification involves the speed command limiting, lag compensation, and output quantization.

### **Estimated Time of Arrival Data from the Traffic to Follow**

In the TBO portion of ASTAR, the assumption is that the TTF may only need minimal equipment, or possibly no extra equipment beyond ADS-B Out, to support airborne self-spacing. It was assumed that a normal self-spacing operation would start prior to the top of descent and continue through the start of the ownship's deceleration to its final approach speed. To compute the trajectory data for the TTF, the ownship would need the following data for the TTF: the names describing its full path, i.e., arrival, transition, and approach names; its cruise Mach and altitude; and its planned final approach speed. It is also assumed that a database that is either local to or part of the ASTAR equipment would interpret the information describing the full path into data that are appropriate for the ASTAR trajectory computation. If these data required to calculate the TTF trajectory are not available via data link, other means of obtaining these data, or eliminating the need for these data, must be found.

Regardless of how these path data are obtained, the trajectory data calculated for the TTF is only a generic "guess" on how the TTF will actually fly the route. Discrepancies between the ASTAR trajectory calculation for the TTF and how the TTF actually flies the route, typically dictated by the FMS, would be propagated in ASTAR as a spacing error. One obvious option for eliminating a significant portion of the trajectory prediction error would be for the TTF to broadcast its ETA, or TTG, via data link. This broadcast could be done at a low frequency, e.g., once every 30 seconds, since the ETA value would typically not change very rapidly. This option would also eliminate the data requirements to the TTF's trajectory generation that were described in the previous paragraph. However, this option would require that the TTF have the ability to generate an accurate ETA and, obviously, the ability to data link this information.

### **Trajectory Data from the Ownship FMS**

Similar to the issues and benefits for using the TTF's ETA data from a data linked FMS source, the use of the ownship's FMS data could significantly reduce the complexity of the spacing algorithm and increase its operational viability. The first obvious option in this regard would be to use the ownship's FMS ETA data in place of the ASTAR trajectory TTG. While this option may provide a more accurate TTG value, it still requires ASTAR to calculate a trajectory for the nominal speed values. As a second option, if the FMS could provide the ETA and the current, non-RTA nominal speed values, then ASTAR would not need to calculate a trajectory for the ownship. The most extensive option for reduction in ASTAR's complexity would come if the speed error value in figure 8 could either be used as an input to the FMS speed requirements, e.g., to adjust the next waypoint crossing speed, or superimposed on the FMS speed output command. Alternatively, the spacing time error value could be used to calculate a pseudo RTA value and this RTA value input into the FMS.

### **Speed Prediction, Speed Quantization, and Lag Compensation Functions**

Many of the ancillary functions in the TBO portion of ASTAR, e.g., the aircraft speed response lag compensation, were added to the implementation ad hoc to overcome operational or performance issues that were observed prior to and during simulation evaluations. Several of these functions were designed to compensate for the lack of integration between the FIM algorithm and aircraft avionics in a retrofit implementation. As such, the overall design of these numerous functions were not designed "in the large." It would be beneficial for both the simplification of the algorithm and potentially better operational performance if these functions could be consolidated into one or two coherent functions.

## Summary

This paper provides an overview of the eighth revision to the Airborne Spacing for Terminal Arrival Routes (ASTAR) algorithm. This algorithm is an airborne self-spacing tool that uses ADS-B data from a leading aircraft assigned by ATC to either achieve or maintain a precise spacing interval behind another aircraft. This algorithm places significant emphasis on providing a low cost, retrofit avionics option that requires minimal integration with other aircraft systems. In that regard, it was assumed that the speed command value would be presented to the pilot and that the pilot would change the speed target of the autothrust system to match the commanded speed from ASTAR. Several capabilities are provided within the algorithm that attempt to balance the number of speed changes against the spacing performance. The current ASTAR design was tailored toward supporting the Air Traffic Management Technology Demonstration-1 (ATD-1) and to conform to the industry Flight-deck Interval Management (FIM) standards document, with the majority of the modifications made in this revision of the algorithm to align with the still evolving FIM standard. The two most significant modifications made in this revision of ASTAR were to add a Mach command capability to the state-based control law and considerable changes to both control laws in support of distance-based spacing operations. In addition to describing the trajectory computations, spacing interval calculations, and both the trajectory-based and state-based speed control laws, this paper discusses operational issues that were addressed in the development of the ASTAR algorithm.

## References

1. Sorensen, J. A., et al: Traffic Alert and Collision Avoidance System (TCAS) - Cockpit Display of Traffic Information (CDTI) Investigation, DOT/FAA/RD-91/8, April 1991.
2. Swedish, W. J.: *Evaluation of the Potential for Reduced Longitudinal Spacing on Final Approach*, FAA-EM-79-7, August 1979.
3. Abbott, Terence S.: Speed Control Law for Precision Terminal Area In-Trail Self-Spacing. NASA/TM-2002-211742, 2002.
4. Oseguera-Lohr, Rosa M.; Lohr, Gary W.; Abbott, Terence S.; Nadler, Eric D.; and Eischeid, Todd M.: Evaluation of a Tool for Airborne-Managed In-Trail Approach Spacing. NASA/TP-2005-213773, August 2005.
5. Barmore, Bryan E.; Abbott, Terence S.; Capron, William R.: Evaluation of Airborne Precision Spacing in a Human-in-the-Loop Experiment. NASA Center for AeroSpace Information (CASI), AIAA Paper 2005-7402; AIAA 5th Aviation Technology, Integration, and Operations Conference (ATIO), Arlington, VA, United States, 26-28 Sep. 2005.
6. Abbott, Terence S.: An Overview of a Trajectory-Based Solution for En Route and Terminal Area Self-Spacing to Include Parallel Runway Operations. NASA/CR-2011-217194, 2011.
7. *Air Traffic Control*. DOT/FAA Order JO 7110.65T, Feb. 11, 2010.
8. Abbott, Terence S.: An Overview of a Trajectory-Based Solution for En Route and Terminal Area Self-Spacing: Third Revision. NASA/CR-2012-217786, 2012.
9. Abbott, Terence S.: An Overview of a Trajectory-Based Solution for En Route and Terminal Area Self-Spacing: Fourth Revision. NASA/CR- 2013-218044, 2013.
10. Baxley, B.T.; Johnson, . C.; Swenson, H. N.; Robinson, J. E.; Prevot, T.; Callantine, T. J.; Scardina, J.; and Greene, M.: Air Traffic Management Technology Demonstration-1 Concept of Operations (ATD-1 ConOps). NASA/TM-2012-217585, 2012.
11. Abbott, Terence S: An Overview of a Trajectory-Based Solution for En Route and Terminal Area Self-Spacing: Fifth Revision. NASA/ CR-2015-218680, February 2015.
12. Federal Aviation Administration, NextGen Mid-Term Concept of Operations for the National Airspace System, FAA ATO, Version 2.0, April 30, 2010.
13. Abbott, Terence S: A Brief History of Airborne Self-Spacing Concepts, NASA/CR-2009-215695, 2009.
14. CoSpace 2003 Aircraft Guidance Model Based Experiments Evaluation of Airborne Spacing For Traffic Sequencing, EEC Report No. 391, 2004.
15. Kelly, James R. and Abbott, Terence S.: In-Trail Spacing Dynamics of Multiple CDTI-Equipped Aircraft Queues, NASA TM-85699, 1984.
16. Williams, David H. and Well, Douglas C.: Jet Transport Flight Operations Using Cockpit Display of Traffic Information During Instruments Meteorological Conditions, Simulation Evaluation, NASA TP-2567, 1986.
17. Abbott, Terence S: An Overview of a Trajectory-Based Solution for En Route and Terminal Area Self-Spacing: Sixth Revision. NASA/ CR-2015-218774, July 2015.
18. RTCA: Minimum Operational Performance Standards (MOPS) for Flight-deck Interval Management (FIM), pre-FRAC draft, 2014.
19. Abbott, Terence S: An Overview of a Trajectory-Based Solution for En Route and Terminal Area Self-Spacing: Seventh Revision. NASA/ CR-2015-218794, August 2015.
20. Baxley, Brian T., et al: Human-in-the-Loop Assessment of Alternative Clearances in Interval Management Arrival Operations. NASA/ TP-2016-219362, December 2016.
21. RTCA: Minimum Operational Performance Standards (MOPS) for Flight-deck Interval Management (FIM), post-FRAC draft, 2015.

22. Williams, David H.: Time-Based Self-Spacing Techniques Using Cockpit Display of Traffic Information During Approach to Landing in a Terminal Area Vectoring Environment, NASA TM 84601, 1983.
23. *Safety, Performance and Interoperability Requirements Document for Airborne Spacing – Flight Deck Interval Management (ASPA-FIM)*. RTCA DO-328, 2011.
24. Abbott, Terence S.: A Trajectory Algorithm to Support En Route and Terminal Area Self-Spacing Concepts: Third Revision. NASA/CR- 2014-2182884, 2014.
25. *Instrument Procedures Handbook*. DOT/FAA-H-8261-1A, 2007.
26. Norris, Guy: *UPS to Test 'SafeRoute' Next Year*, Flight International Magazine, p. 12, 9-15 Aug. 2005.
27. Swieringa, Kurt; Murdoch, Jennifer L.; Baxley, Brian; and Hubbs, Clay: Evaluation of an Airborne Spacing Concept, On-Board Spacing Tool, and Pilot Interface. 11th AIAA Aviation Technology, Integration, and Operations (ATIO) Conference, Virginia Beach, VA, 20-22 Sep. 2011.

**REPORT DOCUMENTATION PAGE**

Form Approved  
OMB No. 0704-0188

The public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing the burden, to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.  
**PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.**

<b>1. REPORT DATE (DD-MM-YYYY)</b> 01-09-2017		<b>2. REPORT TYPE</b> Technical Memorandum		<b>3. DATES COVERED (From - To)</b>	
<b>4. TITLE AND SUBTITLE</b>  An Overview of a Trajectory-Based Solution for En Route and Terminal Area Self-Spacing: Eighth Revision				<b>5a. CONTRACT NUMBER</b>	
				<b>5b. GRANT NUMBER</b>	
				<b>5c. PROGRAM ELEMENT NUMBER</b>	
<b>6. AUTHOR(S)</b>  Abbott, Terence S.; Swieringa, Kurt A.				<b>5d. PROJECT NUMBER</b>	
				<b>5e. TASK NUMBER</b>	
				<b>5f. WORK UNIT NUMBER</b>  330693.04.10.07.09	
<b>7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)</b>  NASA Langley Research Center Hampton, VA 23681-2199				<b>8. PERFORMING ORGANIZATION REPORT NUMBER</b>  L-20856	
<b>9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)</b>  National Aeronautics and Space Administration Washington, DC 20546-0001				<b>10. SPONSOR/MONITOR'S ACRONYM(S)</b>  NASA	
				<b>11. SPONSOR/MONITOR'S REPORT NUMBER(S)</b> NASA-TM-2017-219662	
<b>12. DISTRIBUTION/AVAILABILITY STATEMENT</b>  Unclassified - Unlimited Subject Category 03 Availability: NASA STI Program (757) 864-9658					
<b>13. SUPPLEMENTARY NOTES</b>					
<b>14. ABSTRACT</b>  This paper presents an overview of the eighth revision to an algorithm specifically designed to support NASA's Airborne Precision Spacing concept. This paper supersedes the previous documentation and presents a modification to the algorithm referred to as the Airborne Spacing for Terminal Arrival Routes version 13 (ASTAR13). This airborne self-spacing concept contains both trajectory-based and state-based mechanisms for calculating the speeds required to achieve or maintain a precise spacing interval with another aircraft. The trajectory-based capability allows for spacing operations prior to the aircraft being on a common path. This algorithm was also designed specifically to support a standalone, non-integrated implementation in the spacing aircraft. This current revision to the algorithm supports the evolving industry standards relating to airborne self-spacing.					
<b>15. SUBJECT TERMS</b>  Interval Management; Spacing					
<b>16. SECURITY CLASSIFICATION OF:</b>			<b>17. LIMITATION OF ABSTRACT</b>	<b>18. NUMBER OF PAGES</b>	<b>19a. NAME OF RESPONSIBLE PERSON</b>
<b>a. REPORT</b>	<b>b. ABSTRACT</b>	<b>c. THIS PAGE</b>			STI Help Desk (email: help@sti.nasa.gov)
U	U	U	UU	53	<b>19b. TELEPHONE NUMBER (Include area code)</b> (757) 864-9658