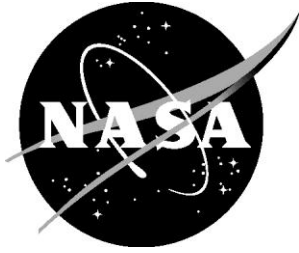


NASA/CR-2012-217786



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

*Terence S. Abbott*  
*Booz Allen Hamilton, McLean, Virginia*

---

November 2012

## 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 NASA Aeronautics and Space Database and its public interface, the NASA Technical Report 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 counterpart of peer-reviewed formal professional papers, but having 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)
- Fax your question to the NASA STI Information Desk at 443-757-5803
- Phone the NASA STI Information Desk at 443-757-5802
- Write to:  
STI Information Desk  
NASA Center for AeroSpace Information  
7115 Standard Drive  
Hanover, MD 21076-1320

NASA/CR-2012-217786



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

*Terence S. Abbott*  
*Booz Allen Hamilton, McLean, Virginia*

National Aeronautics and  
Space Administration

Langley Research Center  
Hampton, Virginia 23681-2199

Prepared for Langley Research Center  
under Contract NNL10AA14B

November 2012

Available from:

NASA Center for AeroSpace Information  
7115 Standard Drive  
Hanover, MD 21076-1320  
443-757-5802

# Table of Contents

Acronyms and Nomenclature.....	iv
Introduction.....	1
ASTAR11 Algorithm Implementation.....	2
Trajectory Computation .....	3
General .....	3
Final Approach Speed .....	3
Initial Cruise Altitude and Speed .....	4
Top of Descent Monitoring.....	4
Wind Forecast Data.....	4
Trajectory State Data Computation.....	6
Calculation of the Spacing Interval.....	7
Basic Time Interval .....	7
Basic Distance Interval.....	7
Achieve-By at the Final Approach Fix .....	8
Speed Control Law.....	8
Speed Change Minimization and Lag Compensation .....	11
Error Gain Scheduling.....	11
End Speed Estimation .....	12
Speed Command Quantization.....	13
Nominal Deceleration Roll-In Logic.....	13
Look Ahead Speed Change Inhibit .....	14
Operational Considerations.....	15
Common Speeds After Merging .....	15
Envelope Protection .....	15
Off-Nominal Mach / CAS Transition .....	15
Landing of the Traffic to Follow Aircraft .....	16
Future Design Considerations.....	16
Estimated Time of Arrival Data from the Traffic to Follow .....	16
Trajectory Data from the Ownship FMS .....	17
Speed Prediction, Speed Quantization, and Lag Compensation Functions .....	17
Summary .....	17
References.....	18

## Acronyms and Nomenclature

2D:	2 dimensional; longitudinal and lateral
4D:	4 dimensional; longitudinal, lateral, vertical, and temporal
ADS-B:	Automatic Dependence Surveillance Broadcast
ASTAR:	Airborne Spacing for Terminal Arrival Routes
CAS:	Calibrated airspeed
DTG:	Distance-to-go
End speed command:	Estimated speed command at the end of a speed change
ETA:	Estimated time of arrival
FAF:	Final approach fix
FMS:	Flight Management System
gs:	Ground speed
IAS:	Indicated airspeed
kt:	Knots
nmi:	Nautical miles
Ownship:	In this document, ownship refers to the aircraft that is performing the spacing operation
RTA:	Required time of arrival
Speed command:	The continuous, instantaneous speed command provided by the algorithm
STAR:	Standard Terminal Arrival Route
TCP:	Trajectory change point
TOD:	Top-of-descent
TTF:	Traffic to follow; the aircraft against which the spacing aircraft is performing a spacing operation
TTG:	Time-to-go

## Abstract

*This paper presents an overview of the third major revision to an algorithm specifically designed to support NASA's Airborne Precision Spacing concept. This algorithm is referred to as the Airborne Spacing for Terminal Arrival Routes version 11 (ASTAR11). This airborne self-spacing concept is trajectory-based, allowing for spacing operations prior to the aircraft being on a common path. Because this algorithm is trajectory-based, it also has the inherent ability to support required-time-of-arrival (RTA) operations. This algorithm was also designed specifically to support a standalone, non-integrated implementation in the spacing aircraft.*

## 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 for 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 totally 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 implementation 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).

This current, third revision of ASTAR, referred to as ASTAR11, is a "lite" version of ASTAR10. In this revision, the ability to space against a second aircraft that is operating to a dependent runway has been removed. Additionally, several implementation improvements have been made based on observations from a pilot-in-the-loop simulation using ASTAR10.

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, where in the case of ASTAR10 is the runway threshold, is known. To apply this to a self-spacing concept, a TTG is calculated for both the traffic to follow aircraft (TTF) and for the ownship, noting that the trajectories do not need to be the same. The nominal spacing time,  $t_{\text{nominal}}$ , and the spacing time error,  $t_{\text{error}}$ , can then be calculated as:

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

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

where the identification of the TTF aircraft and the determination of the planned spacing time is 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,

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

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 1.

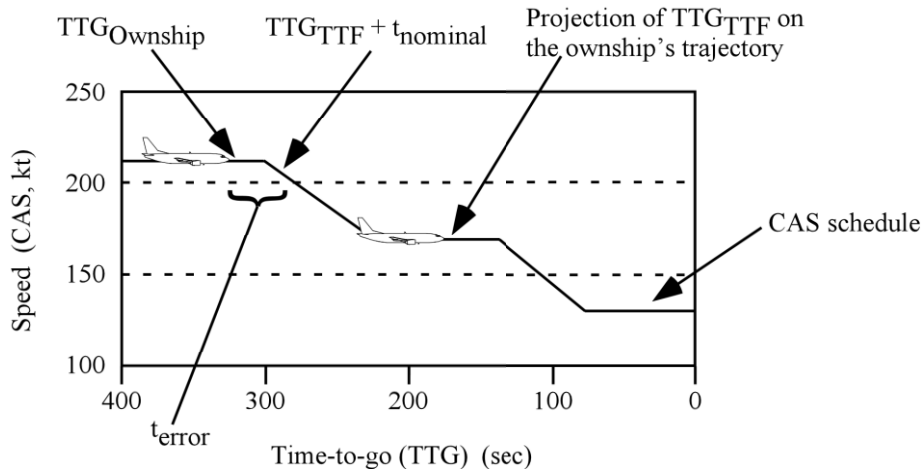


Figure 1. Time error example.

By design, ASTAR11 is considered an achieve-by algorithm (ref. 8), i.e., it is designed to attain the spacing goal at the achieve-by point, which in the NASA Airborne Precision Spacing concept is normally the runway threshold. The algorithm does not exactly obtain and maintain the spacing goal until the ownship is near the achieve-by point. 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.

## ASTAR11 Algorithm Implementation

The implementation of the ASTAR11 algorithm is comprised primarily 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. Details of these elements are provided in subsequent sections.



## Trajectory Computation

### *General*

For the prototype system developed at the NASA Langley Research Center, a standalone trajectory generator was developed to calculate a full 4D trajectory from a 2D path specification. Reference 9 provides a description of this algorithm to include its input and output parameters. In ASTAR11, the trajectory definition begins with a simple, augmented 2D path definition, e.g., a traditional Standard Terminal Arrival Route (STAR) with a continuous connection to an instrument approach procedure, along with relevant speed and altitude constraints. 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 ASTAR11 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 ASTAR11 nor the use of a data linked estimated time of arrival (ETA) or TTG from the traffic aircraft.

One of the major difficulties in computing a 4D trajectory involves the calculation of the length of the ground path during a turn. During turns in either the presence of winds or with a change in the specified speed, the turn radius is changed, which then affects 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 in 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, ASTAR11 preprocesses the trajectory input data depending on the situation. ASTAR11 may change the generic trajectory parameters relative to aircraft 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*

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. Because of this requirement, ASTAR11 modifies 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. In addition, there are several different operational techniques used in determining where the final approach speed is achieved. In the generic case, the final deceleration starts at a point prior to the runway threshold such that the aircraft just achieves its final approach speed as it crosses the runway threshold. Since this baseline technique is not typical for transport aircraft approaches in that it does not provide for a stabilized approach, ASTAR11 provides two other options. These options are:

- Begin the final deceleration at the waypoint just prior to the runway. This is typical for operations where the final deceleration starts at the final approach fix (FAF).
- 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 ft above the runway's elevation, which is also the minimum requirement for instrument approaches in transport aircraft (ref. 10). 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.

A fourth option was considered, where the final deceleration starts at a point prior to the FAF such that the aircraft just achieves its final approach speed as it crosses the FAF. This option, however, is not implemented.

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

A second change that ASTAR11 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 2D path and with the aircraft's current altitude and Mach matching a relevant data set being provided to ASTAR11. That is, the current altitude and Mach of the aircraft must match a special cruise altitude and Mach data set being sent to ASTAR11. For the TTF, this special data set could be made available to the ownship via data link.

### ***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 in the estimation of the aircraft's ground speed during this descent and therefore lead to a significant error in the aircraft's TTG, ASTAR11 monitors for the conformance to its TOD point. If the aircraft begins its descent from cruise before the point that ASTAR11 predicted, ASTAR11 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 ASTAR11 may recalculate the 4D path several times, depending on how far beyond the originally estimated TOD point that the actual TOD occurs.

### ***Wind Forecast Data***

The last modification that ASTAR11 may apply to the trajectory input data is to modify the original wind forecast data provided to the algorithm. Wind data into and within ASTAR11 is based on waypoint locations instead of a typical wind grid. It was assumed in the design of ASTAR11 that a highly developed wind forecast model would be used to provide vertical profile wind data at the waypoint locations. Of special importance to ASTAR11 would be the wind estimation at the altitude that the trajectory would be crossing the waypoint's position. It was assumed then that the externally provided waypoint wind data would provide reasonably accurate wind data that would bound the expected waypoint trajectory crossing altitude. Up to 10 altitude-wind speed data sets (altitude, direction, and magnitude) per waypoint may be input into ASTAR11. From this initial, external input ASTAR11 may then provide both local and global modifications to the forecast wind data provided to the trajectory generator.

While up to 10 altitude-wind data sets per waypoint may be input into ASTAR11, ASTAR11's internal wind model maintains a 100 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 forecast for this altitude, a measured wind for this altitude, and the current estimated wind forecast for this altitude. Initially, the gain values are all set to 0. At the receipt of a new, external wind forecast, the input

wind forecast for each altitude is populated, with an altitude-based linear interpolation used to populate the altitudes that do not directly have any input value.

Measured wind 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 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 100 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 100 ft incremental vertical profile data sets may be modified for altitudes within  $\pm 3000$  ft of the ownship's altitude. Whether a specific 100 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, with the ownship's current gain being 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  $x_{ownship}$  is greater than 50 nmi,  $gain_{horizontal} = 0$ ,

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

if  $z_{ownship}$  is greater or equal to 12,000 ft, then

if  $z_{ownship}$  is greater than  $\pm 5000$  ft,  $gain_{vertical} = 0$ ,

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

otherwise

if  $z_{ownship}$  is greater than  $\pm 3000$  ft,  $gain_{vertical} = 0$ ,

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

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 wind data are updated with the new gain value and measured wind data. The new estimated wind data are computed based on a double linear interpolation between the original forecast winds and the measured winds. The double linear interpolation uses the relative horizontal position, the relative vertical position, and the previously calculated, associated gain values.

ASTAR11 has the option to include a global wind updating capability in its wind forecast update. In this case, ASTAR11 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.

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 would not be used in the internal forecast.

Once a new internal forecast has been generated, ASTAR11 selects the best altitudes for each waypoint, based on bounding the trajectory crossing altitude, to update the wind data profile in the trajectory input data.

### 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 of figure 2,  $TCP_1$  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. 3) between the aircraft's position and that segment a minimum for the trajectory? In this example, the aircraft is forward of  $TCP_i$  (fig. 4), in the direction of the trajectory's ground path, and behind  $TCP_{i+1}$  (fig. 5).

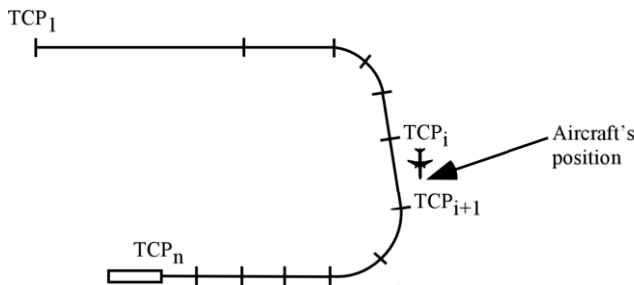


Figure 2. Trajectory state estimation.

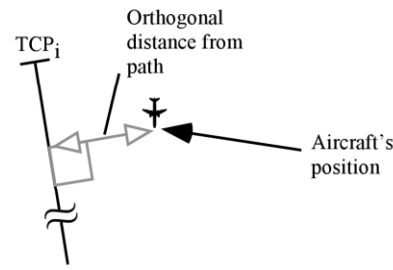


Figure 3. Orthogonal distance measurement.

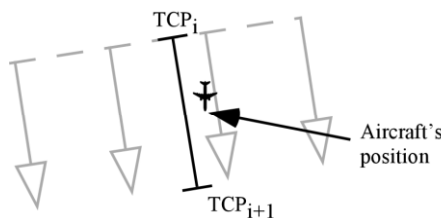


Figure 4. Aircraft's position forward of  $TCP_i$ .

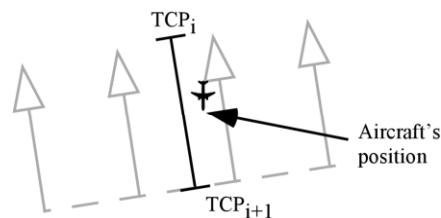


Figure 5. Aircraft's position behind  $TCP_{i+1}$ .

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 6). The trajectory altitude is then computed using a simple linear interpolation between the distance between the trajectory state point ( $d$  in figure 6) and

TCP<sub>i+1</sub> and the distance between TCP<sub>i</sub> and TCP<sub>i+1</sub>, i.e., DTG<sub>i</sub> - DTG<sub>i+1</sub>. For example, the altitude (alt) at a position  $p_d$  on the trajectory can be calculated as:

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

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

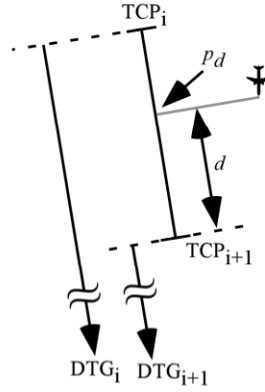


Figure 6. Distance-to-go measurement.

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

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

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

### Calculation of the Spacing Interval

The spacing interval provided by ATC may be given to ASTAR11 in either time or distance. An explanation of these two spacing interval types is provided in the following two sections.

#### **Basic Time Interval**

The basic time spacing interval is the interval that ATC would assign for the spacing aircraft to obtain at the runway threshold against the assigned TTF. The basic spacing interval for ASTAR11 is a time-reference interval against a TTF that is landing on the same runway as the ownship. The operational goal in this situation is for the ownship to cross the runway threshold at the assigned interval after the TTF crossed the same threshold. For this basic time interval case, there is no additional calculation required for the spacing interval; it is simply the time assigned by ATC.

#### **Basic Distance Interval**

In the basic distance spacing interval case, the operational goal is for the ownship to be at the ATC assigned distance behind the TTF just as the TTF crosses the runway threshold. As in the basic time interval case, the same runway is used by both the TTF and the ownship. For this case, the applicable spacing time that is used by the control law can be calculated from the 4D trajectory by determining the ownship's trajectory state at the assigned spacing interval distance-to-go from the threshold. The spacing

time goal is then the time-to-go to the threshold at this distance. That is, the relevant spacing time is the time-to-go on the ownship's trajectory at a distance-to-go equal to the assigned spacing distance.

### ***Achieve-By at the Final Approach Fix***

For the normal situation where the achieve-by point is the runway threshold, the use of an achieve-by algorithm coupled with the operational requirement to attain a stabilized approach means that the algorithm must compensate for differences in the TTF's and the ownship's actual final approach speed prior to the stabilized approach point. This capability is implicitly provided by the use of the respective aircrafts' final approach speeds in the calculation of their trajectory times to the runway threshold. For the situation where the achieve-by point is the final approach fix, then the algorithm simply offsets the aircrafts' TTG value by the time difference between the time to the runway threshold and the time to the FAF. A similar technique is used in the spacing distance calculations.

### **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 then used in the speed control law (fig. 7). The overall design concept for this control law was to command the nominal trajectory speed with any resulting spacing error only modifying this nominal speed by a maximum of  $\pm 10\%$ , thus 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 11.

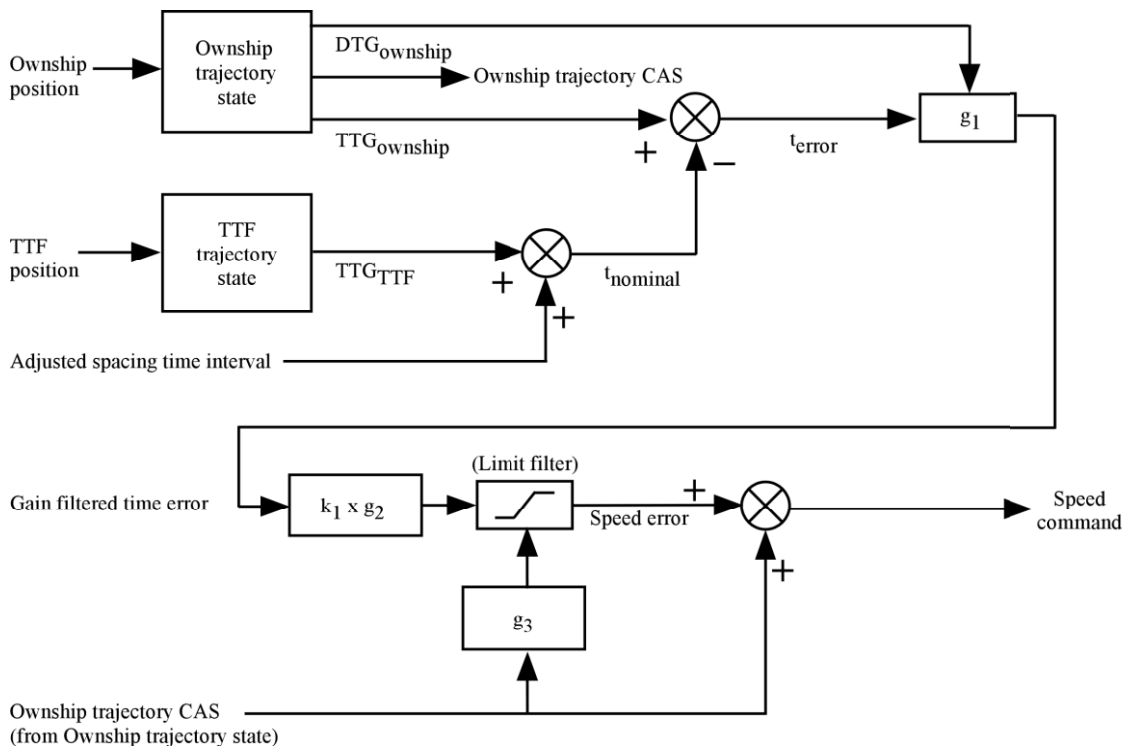


Figure 7. Speed control law.

For this control law, the units are nautical miles, knots, and seconds. The speed values are all in CAS. The  $g_1$  term is a time error gain schedule that is used to both reduce the effect of data noise and the number of changes of the speed command value while the ownship is still at some distance from the runway (fig. 8). For example, with the ownship at 80 nmi from the runway and a 24 sec spacing error, the output of the gain schedule would be an 18 second spacing error. The value of  $g_1$  is:

*if  $DTG_{ownship}$  is greater than 80 nmi,  $g_1 = 0.75$ ,*

*otherwise if  $DTG_{ownship}$  is less than 20 nmi,  $g_1 = 1.0$ ,*

*otherwise  $g_1 = 1.0 - 0.25 * (DTG_{ownship} - 20 \text{ nmi}) / 60 \text{ nmi}$ .*

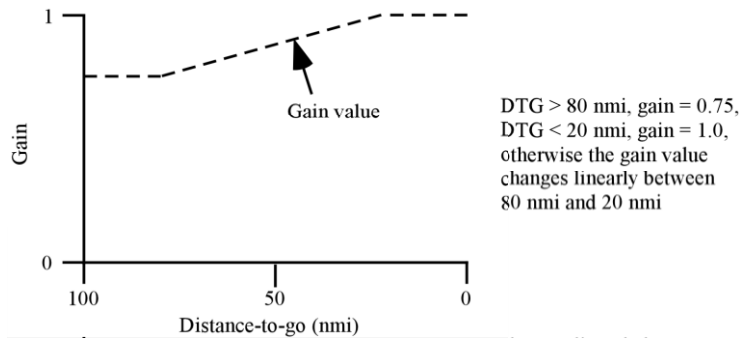


Figure 8. Gain schedule,  $g_1$ .

The value of  $g_2$  (fig. 9), which is used to gain schedule the speed error term, is:

*if  $DTG_{ownship}$  is greater than 40 nmi,  $g_2 = 0.66$ ,*

*otherwise if  $DTG_{ownship}$  is equal to 40 nmi,  $g_2 = 0.75$ ,*

*otherwise if  $DTG_{ownship}$  is less than 5 nmi,  $g_2 = 1.0$ ,*

*otherwise  $g_2 = 0.75 + 0.25 * (1 - (DTG_{ownship} - 5 \text{ nmi}) / 35 \text{ nmi})$ .*

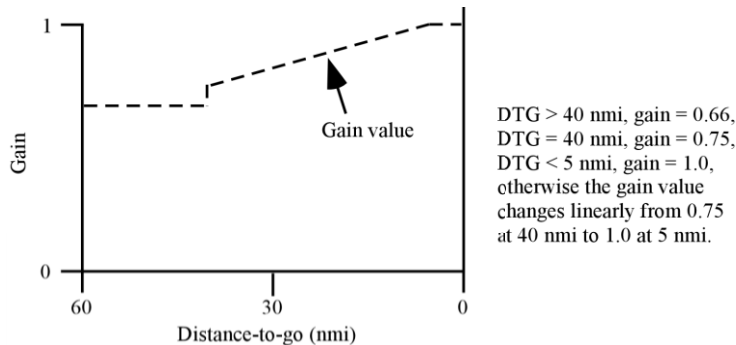


Figure 9. Gain schedule,  $g_2$ .

The value of  $g_3$  is 0.1 and the value of  $k_1$  is 6076 (nmi/hr) / 3600 sec. From  $g_3$ , the limit filter limits the speed-error value to  $\pm 10\%$  of the ownship's nominal trajectory speed. At 5 nmi, these values produce approximately 1.7 kt of speed correction (fig. 7) for 1 sec 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 in figure 7.

Because the operational envelop for this algorithm includes high altitude Mach portions, both the trajectory calculations and the control law accommodate Mach. If the aircraft is operating in a Mach regime, then the Mach value from the trajectory data, 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.

Finally, 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 two typical operational techniques for terminating active spacing and transiting to the aircraft's planned final approach speed. An example of how this capability is supported in ASTAR11 is shown in figure 10. 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. 10), then the deceleration to the final approach speed would need to occur earlier. To accommodate this situation, ASTAR11 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, ASTAR11 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. 11). In this case, ASTAR11 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.

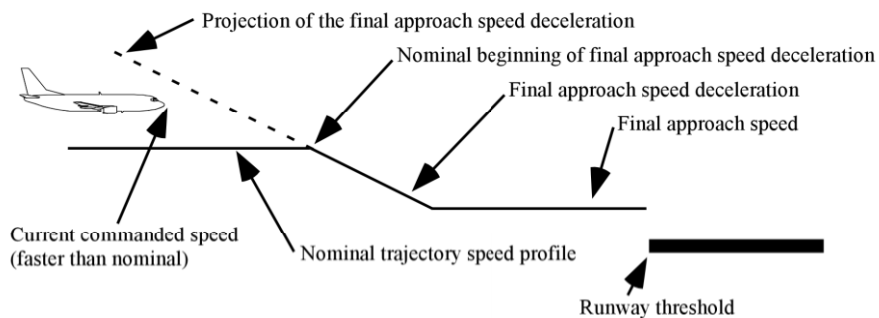


Figure 10. Final approach speed deceleration from an initially faster commanded speed.



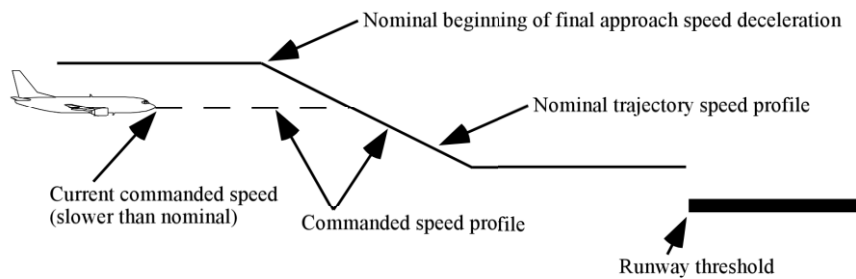


Figure 11. Commanded speed profile from an initially slower commanded speed.

### Speed Change Minimization and Lag Compensation

One of the most significant design requirements for the last two versions of ASTAR is 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. 12) for airborne self-spacing. To support this option, from a pilot workload standpoint it was deemed beneficial to attempt 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 ASTAR11 algorithm.

### Error Gain Scheduling

One means for reducing the number of speed changes in a spacing algorithm is to notch filter the time error value prior to its use in the speed command calculation. By notch filtering, fairly large errors are allowed when far from the runway threshold without inducing a speed correction. This technique was used in ASTAR10. One significant performance issue with using a notch filter is that by allowing large spacing errors when far from the runway, the algorithm may not be able to recover from what may have been a recoverable error. For example, if the aircraft were initially situated without any spacing error and the TTF then flew the approach as fast as possible, i.e., the profile speed plus 10 percent, then the following would occur:

- The ownship would continue to move farther behind the nominal spacing interval until the notch filter limit was reached.
- Once the notch filter's limit was reached, ownship's speed command would increase until the speed command reached the profile speed plus 10 percent.
- The ownship would maintain the spacing error that was present when the ownship's speed command reached the profile speed plus 10 percent.

Based on some unsatisfactory performance issues when using this notch filter technique (ref. 13), a simple time error gain schedule,  $g_1$  in figures 7 and 8, is now used in the speed control law for ASTAR11. This time error gain schedule in the speed control law (fig. 8) simply reduces the perceived spacing time error in the speed command calculation when the ownship is far from the runway threshold. Also note

that the speed error gain schedule,  $g_2$  in figures 7 and 9, is fundamentally a second, cascading time error gain scheduling value.

### ***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. 12), 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 12, 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.

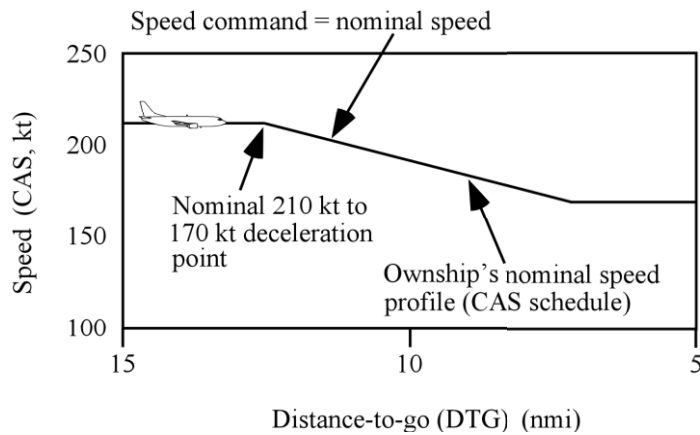


Figure 12. Example speed change with no spacing correction.

A similar situation would occur in the presence of a required speed correction due to a spacing error or RTA adjustment. In figure 13, the nominal speed profile is the same as in figure 12, 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.

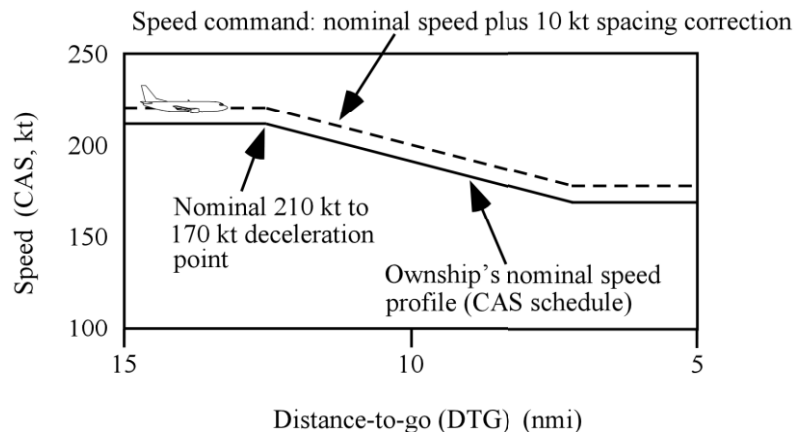


Figure 13. Example speed change with a positive 10kt spacing correction.

### ***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 would only occur in discrete intervals, thus reducing the number of commanded speed changes. For example, if the speed command (fig. 7) 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 "chunked" 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.

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

During the initial evaluation of ASTAR10 (ref. 13), 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 under 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 14. 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 sec TTG in the example of figure 14, 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 12 sec look ahead, the equivalent situation is shown in figure 15 with the roll-in logic. In figure 15, the end speed command would change from 210 kt to 170 kt 12 sec earlier relative to the basic profile. In this situation (fig. 15), the instantaneous speed command would change in a manner such that the nominal deceleration rate and speed would just match the nominal command speed and deceleration rate 24 sec after the start of the roll-in period.

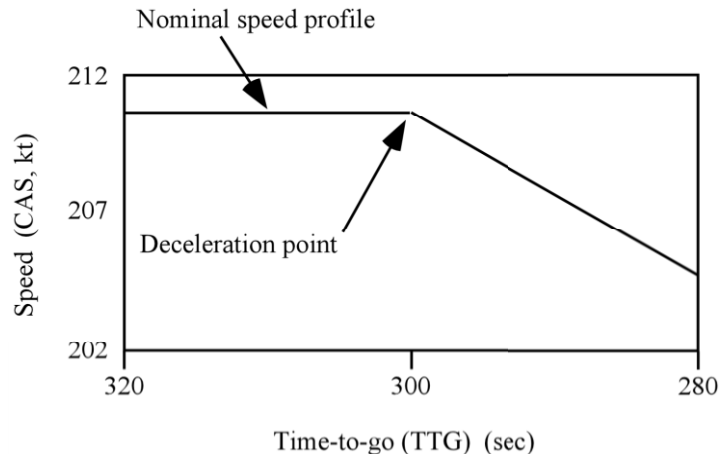


Figure 14. Nominal speed change without roll-in logic.

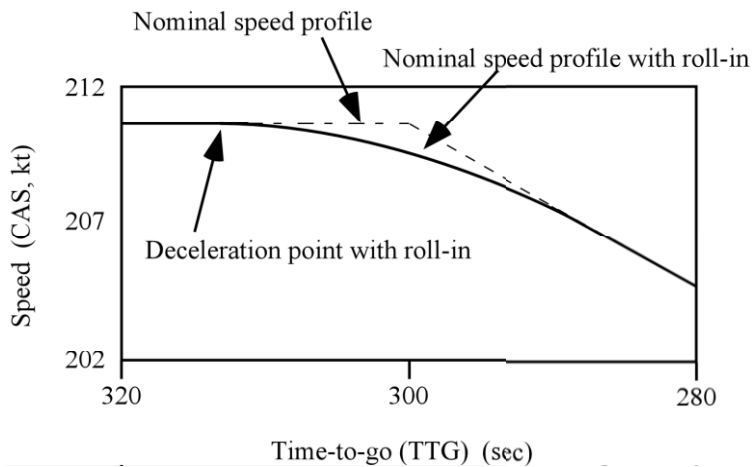


Figure 15. Nominal speed change with roll-in logic.

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

To minimize the number of speed changes prior to a programmed deceleration segment, i.e., where the planned trajectory specifies a deceleration, a look-ahead speed change inhibit option was used. In this regard, the algorithm would look ahead by 10 seconds in the nominal speed profile (fig. 16) 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 sec roll-in interval would be added to the 10 sec look-ahead interval. Thus the speed change inhibit logic would be applied 22 sec prior to the deceleration point on the basic, nominal speed profile.

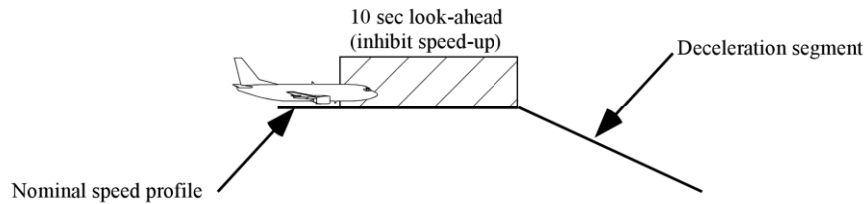


Figure 16. Look-ahead speed-up inhibit.

## 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. In this regard, 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. 17). That is, common path points must have common speeds.

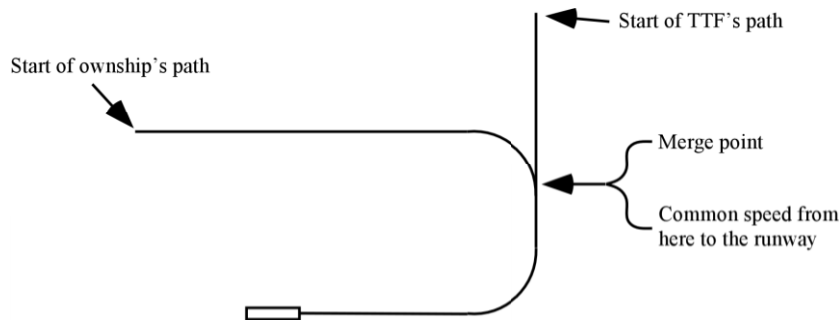


Figure 17. Example of common speeds after the merge point.

### Envelope Protection

Since the speed command value from ASTAR11 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 ASTAR11. These input limit speeds are usually based on the design limiting speed, the maximum gust penetration speed, the maximum flap extended speed, 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.

### 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, ASTAR11 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.

### **Landing of the Traffic to Follow Aircraft**

If the ownship has not reached its final deceleration point prior to the TTF crossing the runway threshold, ASTAR will continue to correct for spacing errors even after the TTF has crossed the runway threshold. To continue actively spacing after the TTF crosses the threshold, the algorithm “freezes” the TTF's state data, i.e., time and position, just as the TTF crosses the threshold. The algorithm then offsets the spacing time interval by an amount equal to the current time minus the time that the TTF crossed the threshold. The adjusted spacing time interval for figure 7 is then:

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

where the adjusted spacing time interval<sub>original</sub> is the original ATC assigned spacing interval along with any other adjustments, e.g., runway offset, and crossing time<sub>TTF</sub> is the time that the TTF crossed the runway threshold. Note that in the subsequent calculation of the nominal spacing time that the value for the TTG<sub>TTF</sub> (fig. 7) is 0.

### **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 ASTAR11 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 ASTAR11, 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 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 data base that is either local to or part of the ASTAR equipment would interpret the names 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, then other means for 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 much of this particular error would be for the TTF to broadcast via data link its ETA, i.e., its TTG. This broadcast could be done at a low frequency, e.g., once every 30 sec., 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 7 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 ASTAR11, e.g., the autothrust lag compensation, were added to the implementation ad hoc to overcome operational or performance issues that were observed prior to and during simulation evaluations. As such, the overall design for the numerous functions that were added for a non-integrated, aircraft retrofit application along with functions to compensate for the aircraft speed response to speed command changes, e.g., the autothrust lag compensation, was not a design "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 Airborne Spacing for Terminal Arrival Routes (ASTAR) algorithm. This algorithm is a trajectory-based, self-spacing tool using ADS-B data from a leading aircraft assigned by ATC to the following, self-spacing aircraft. This third major revision of this algorithm, referred to as ASTAR11, places significant emphasis on supporting 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. Several capabilities are provided within the algorithm that attempt to balance the number of speed changes against the spacing performance. In addition to describing the trajectory computations, spacing interval calculations, and the speed control law, this paper also discusses operational issues that were addressed in the development of this tool.

## 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. *Safety, Performance and Interoperability Requirements Document for Airborne Spacing – Flight Deck Interval Management (ASPA-FIM)*. RTCA DO-328, 2011.
9. Abbott, Terence S.: A Revised Trajectory Algorithm to Support En Route and Terminal Area Self-Spacing Concepts. NASA/CR-2010- 216204, 2010.
10. *Instrument Procedures Handbook*. DOT/FAA-H-8261-1A, 2007.
11. Abbott, Terence S: A Brief History of Airborne Self-Spacing Concepts. NASA/CR-2009-215695, 2009.
12. Norris, Guy: *UPS to Test 'SafeRoute' Next Year*, Flight International Magazine, p. 12, 9-15 Aug. 2005.
13. Swieringa, Kurt; Murdoch, Jennifer L.; Baxley, Brian; 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 this 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> 01-11-2012		<b>2. REPORT TYPE</b> Contractor Report		<b>3. DATES COVERED (From - To)</b> 01/10/2011 - 01/08/2012	
<b>4. TITLE AND SUBTITLE</b>  An Overview of a Trajectory-Based Solution for En Route and Terminal Area Self-Spacing: Third Revision				<b>5a. CONTRACT NUMBER</b> NNL10AA14B	
				<b>5b. GRANT NUMBER</b>	
				<b>5c. PROGRAM ELEMENT NUMBER</b>	
				<b>5d. PROJECT NUMBER</b>	
<b>6. AUTHOR(S)</b>  Abbott, Terence S.				<b>5e. TASK NUMBER</b>	
				<b>5f. WORK UNIT NUMBER</b> 411931.02.61.07.01	
				<b>8. PERFORMING ORGANIZATION REPORT NUMBER</b>	
<b>7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)</b> NASA Langley Research Center Hampton, Virginia 23681-2199				<b>8. PERFORMING ORGANIZATION REPORT NUMBER</b>	
<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/CR-2012-217786	
<b>12. DISTRIBUTION/AVAILABILITY STATEMENT</b> Unclassified - Unlimited Subject Category 03 Availability: NASA CASI (443) 757-5802					
<b>13. SUPPLEMENTARY NOTES</b> This report was prepared by Booz Allen Hamilton under NASA contract NNL10AA14B with SGT, Inc., Greenbelt, MD. Langley Technical Monitor: Bryan E. Barmore					
<b>14. ABSTRACT</b>  This paper presents an overview of the third major revision to an algorithm specifically designed to support NASA's Airborne Precision Spacing concept. This algorithm is referred to as the Airborne Spacing for Terminal Arrival Routes version 11 (ASTAR11). This airborne self-spacing concept is trajectory-based, allowing for spacing operations prior to the aircraft being on a common path. Because this algorithm is trajectory-based, it also has the inherent ability to support required time-of-arrival (RTA) operations. This algorithm was also designed specifically to support a standalone, non-integrated implementation in the spacing aircraft.					
<b>15. SUBJECT TERMS</b>  Air traffic control; Aircraft spacing; Algorithms; Runways					
<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	25	<b>19b. TELEPHONE NUMBER (Include area code)</b> (443) 757-5802