NASA/CR–2011-217194

# An Overview of a Trajectory-Based Solution for En Route and Terminal Area Self-Spacing to Include Parallel Runway Operations

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

November 2011

NASA/CR–2011-217194

# An Overview of a Trajectory-Based Solution for En Route and Terminal Area Self-Spacing to Include Parallel Runway Operations

*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  2011

Available from:

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

# Table of Contents

## 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 |
| FMS: | Flight Management System |
| gs: | Ground speed |
| IAS: | Indicated airspeed |
| kt: | Knots |
| nmi: | Nautical miles |
| Ownship: | From a flight crew's perspective, the aircraft that they are flying. In this document, it also refers to the aircraft that is performing the spacing operation. |
| RTA: | Required time of arrival |
| STAR: | Standard Terminal Arrival Route |
| t: | Time |
| 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 an algorithm specifically designed to support NASA's Airborne Precision Spacing concept. This airborne self-spacing concept is trajectory-based, allowing for spacing operations prior to the aircraft being on a common path. This implementation provides the ability to manage spacing against two traffic aircraft, with one of these aircraft operating to a parallel dependent runway. Because this algorithm is trajectory-based, it also has the inherent ability to support required-time-of-arrival (RTA) operations.*

## 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 the 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.

A specific implementation of this algorithm to support dependent runway operations, referred to as ASTAR10, provides 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 includes 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. 6). This latest implementation of ASTAR also has a rewritten control law relative to the previous versions that were based on the original Advanced Terminal Area Approach Spacing (ATAAS) algorithm (ref. 3).

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 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, $t_{\mathrm{nominal}}$, and the spacing time error, $t_{\mathrm{error}}$, can then be calculated as:

$$t_{nominal} = TTG_{TTF} + \mathit{planned\ spacing\ interval}\,,$$

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

where the determination of the TTF aircraft and 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,

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

From $t_{error}$, a speed error value can then be calculated. A conceptual example for the determination of $t_{error}$ for traffic spacing is shown in figure 1.

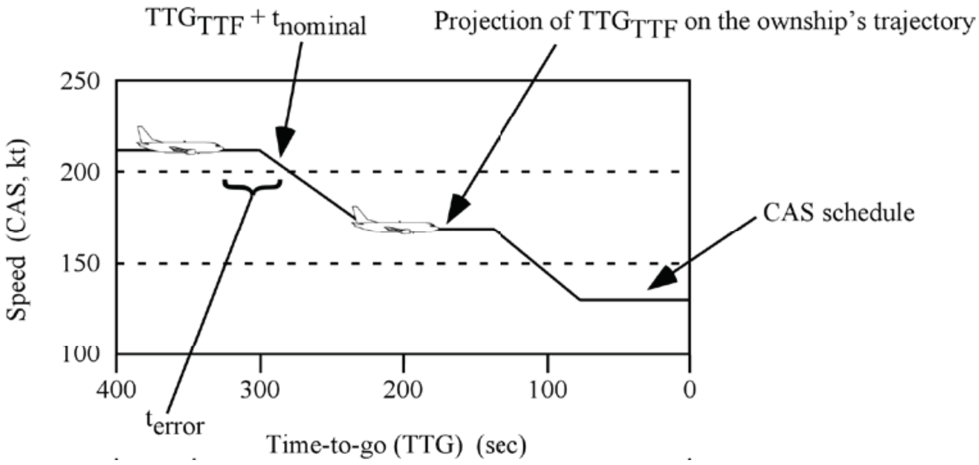


Figure 1. Time error example.

By design, ASTAR10 is considered an achieve-by algorithm (ref. 7), 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 runway threshold. 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.

# ASTAR10 Algorithm Implementation

The implementation of the ASTAR10 algorithm is comprised primarily of six major elements: trajectory computation, current trajectory state data computation, the calculation of the spacing interval, the selection of the spacing target, 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 8 provides a complete description of this algorithm to include its input and output parameters. In ASTAR10, 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 ASTAR10 needs to calculate the trajectories 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 ASTAR10.

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 passes then use the input from the previous pass as a starting point to refine the solution.

In conjunction with the basic 4D calculation, ASTAR10 preprocesses the trajectory input data depending on the situation. ASTAR10 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 TTFs' and the ownship's actual final approach speeds. Because of this requirement, ASTAR10 modifies each aircraft's trajectory data by substituting the individual aircrafts' 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 landing 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, ASTAR10 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.

- 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. 9).

### *Initial Cruise Altitude and Speed*

A second change that ASTAR10 may make to the ownship's or TTFs' 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 being provided to ASTAR10 and with the aircraft's current altitude and Mach matching a relevant data set being provided to ASTAR10. That is, the current altitude and Mach of the aircraft must match a special cruise altitude and Mach data set being sent to ASTAR10.

### *Top of Descent Monitoring*

The FMS may calculate and the aircraft may fly from a top-of-descent (TOD) 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, ASTAR10 monitors for the conformance to its TOD point. If the aircraft begins its descent from cruise before the point that ASTAR10 predicted, ASTAR10 will calculate the actual, current descent angle based on the 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 ASTAR10 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 ASTAR10 may apply to the trajectory input data is to modify the original wind forecast data provided to the algorithm. Wind data into and within ASTAR10 is based on waypoint locations instead of a typical wind grid. It was assumed in the design of ASTAR10 that a highly developed wind forecast model would be used to provide vertical profile wind data at the waypoint locations. Of special importance to ASTAR10 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 ASTAR10. From this initial, external input ASTAR10 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 ASTAR10, ASTAR10'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 ±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 ±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 nmi).
if $z_{ownship}$ is greater or equal to 12,000 ft, then
    if $z_{ownship}$ is greater than ±5000 ft,  $gain_{vertical}$ = 0,
    otherwise  $gain_{vertical}$ = 1 - absolute value of ($z_{ownship}$ / 5000 ft).
otherwise
    if $z_{ownship}$ is greater than ±3000 ft,  $gain_{vertical}$ = 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 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.

ASTAR10 has the option to include a global wind updating capability in its wind forecast update. In this case, ASTAR10 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 its 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, ASTAR10 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).

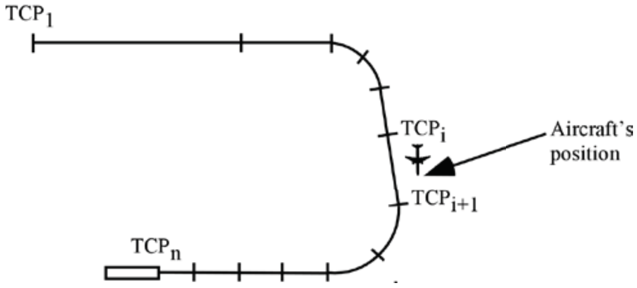


Figure 2. Trajectory state estimation.

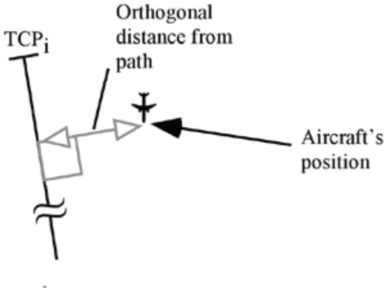


Figure 3. Orthogonal distance measurement.

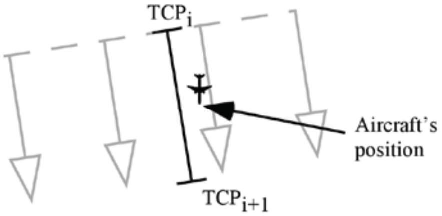


Figure 4. Aircraft's position forward of TCP$_i$.

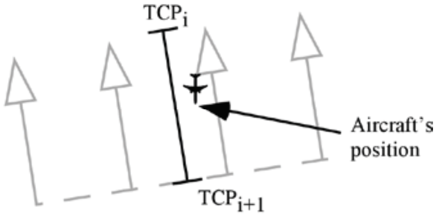


Figure 5. Aircraft's position behind TCP$_{i+1}$

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$_{i+1}$ 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:

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

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

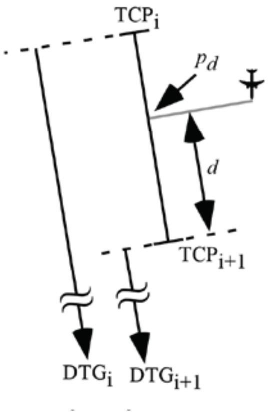


Figure 6. Distance-to-go measurement.

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:

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

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 ASTAR10 in either time or distance. For parallel dependent runway operations, ASTAR10 must also compensate for any offset of the parallel runway thresholds, and meet or exceed the different spacing requirements for the two TTF aircraft.

### *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 ASTAR10 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.

### *Offset Runway Compensation*

To accommodate parallel dependent approach spacing where the runway thresholds are typically not aligned, ASTAR10 internally calculates the approach time difference due to this offset and adjusts the spacing interval to account for this difference. In the example of figure 7, the runway threshold for the TTF is beyond the threshold of the ownship. Using the center point of the runway threshold positions, the distance and angle between these positions can be computed. Since the inbound approach course is known, the distance *d* can then be calculated using right-triangle geometry. From *d*, which is the distance-to-go value for the TTF when it is abeam with the ownship's runway threshold, the time-to-go at that point for the TTF can be determined. Defining this time as $threshold_{offset}$, the adjusted spacing time interval is then calculated as:

$$adjusted\ spacing\ time\ interval = planned\ spacing\ interval - threshold_{offset}.$$

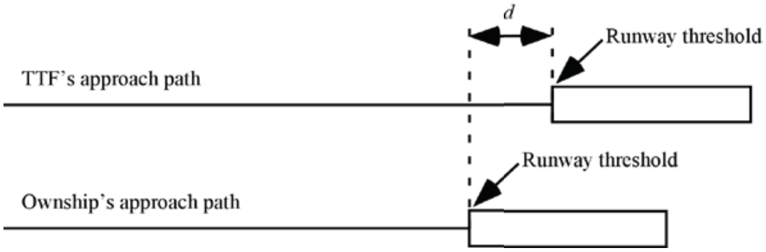


Figure 7. Example of non-aligned runway thresholds.

A similar calculation can be made for the case where the runway threshold for the ownship is beyond the threshold of the TTF. In this case the adjusted spacing time interval is calculated as:

$$adjusted\ spacing\ time\ interval = planned\ spacing\ interval + threshold_{offset}.$$

### *Diagonal Distance Interval*

To support parallel dependent approaches as described in reference 6, ATC uses a spacing interval based on a diagonal distance interval between successive aircraft on adjacent approaches (fig. 8). Given that the diagonal distance interval and the distance between the runway centerlines are known, the effective in-trail distance, i.e., the in-trail spacing interval that would provide the diagonal distance interval, can be calculated, again using a right-triangle geometry. From this effective in-trail distance, the adjusted time spacing interval can be calculated using the methods previously described in *Basic Distance Interval* and *Offset Runway Compensation*.

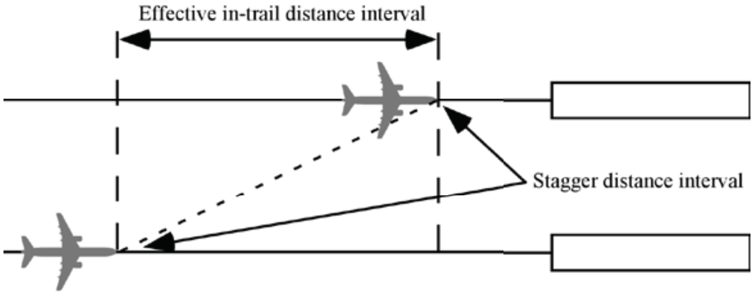

Effective in-trail distance interval
Stagger distance interval

Figure 8. Example of a diagonal distance interval.

## Selection of the Spacing Target

ASTAR10 can simultaneously calculate the time error relative to all of the operational targets, i.e., an RTA, spacing against a TTF going to a common runway, and spacing against a TTF going to a parallel runway. In a typical arrival operation, both of the TTF may be initially outside of ADS-B range. In this situation, ASTAR10 calculates the speed command against the RTA time error. Once ADS-B data from either of the TTF is received and a trajectory is calculated for that aircraft, spacing against the RTA is inhibited and pair-wise spacing, i.e., spacing against a TTF, is initiated. The algorithm does not revert into an RTA mode if the TTF data are subsequently lost or become invalid.

If the algorithm has valid data for both TTF, then data from the TTF that has the largest nominal spacing time is used to compute the speed command, where for each TTF

$$t_{nominal} = TTG_{TTF} + \textit{adjusted spacing time interval.}$$

Using this technique, there are no step changes in the nominal TTG value being used by the speed control law when the selection switches between the TTF.

## 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. 9). 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 ±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 10.

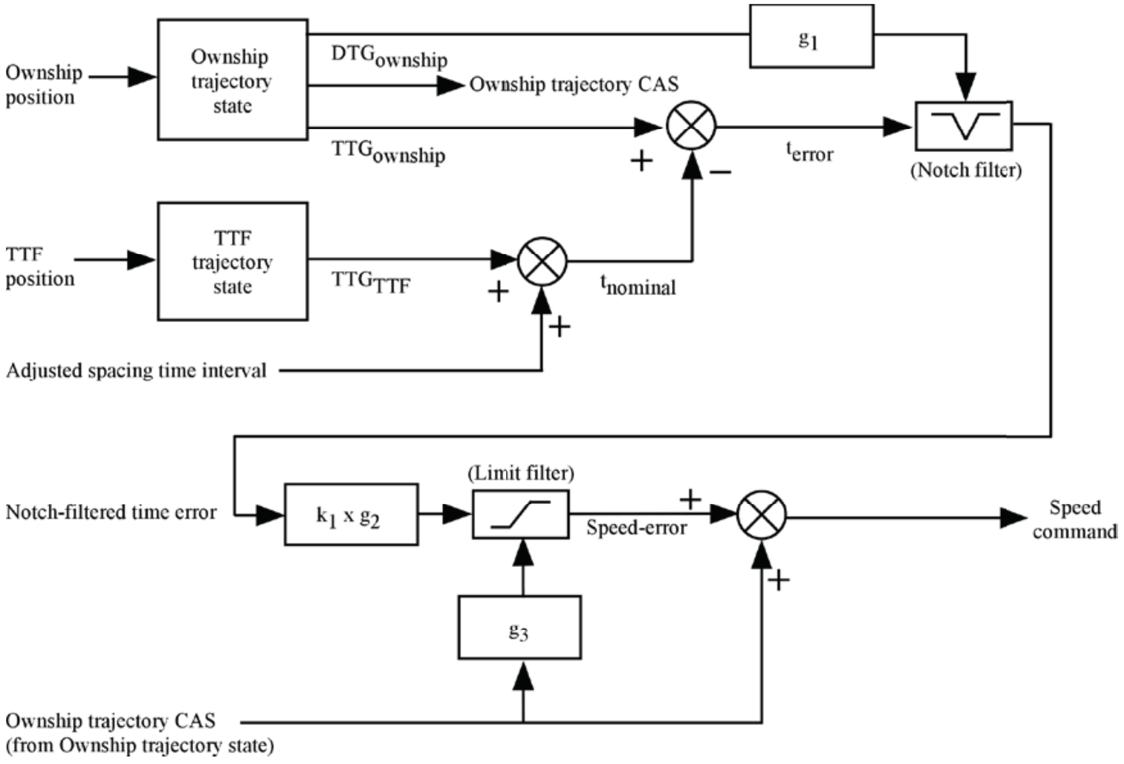


Figure 9. Speed control law.

For this control law, the units are nautical miles, knots, and seconds. The speed values are all in CAS. The value of $g_1$ is:

*if* $DTG_{ownship}$ *is greater than 80 nmi,* $g_1 = 15$ *sec,*

*otherwise if* $DTG_{ownship}$ *is less than 20 nmi,* $g_1 = 0$ *sec,*

*otherwise* $g_1 = (DTG_{ownship} - 20 \text{ nmi}) / 60 \text{ nmi} * 15 \text{ sec}.$

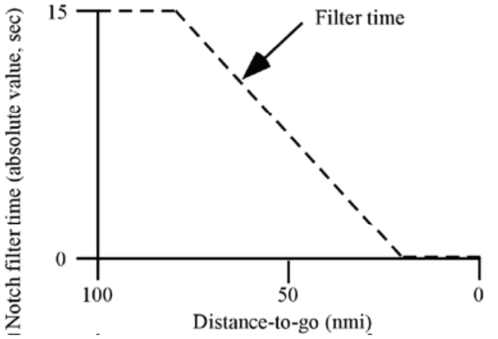
The value of $g_2$ is:

*if* $DTG_{ownship}$ *is greater than 40 nmi,* $g_2 = 0.5,$

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

*otherwise* $g_2 = 1.0 - 0.5 * (DTG_{ownship} - 5 \text{ nmi}) / 35 \text{ nmi}.$

The value of $g_3$ is 0.1 and the value of $k_1$ is 6076 (nmi/hr) / 3600 sec. The notch filter was used to both reduce the effect of data noise and the number of changes of the speed command value while at some distance from the runway (fig. 10). For example, with the ownship at 80 nmi from the runway and a 27 sec spacing error, the output of the notch filter would be a 12 second spacing error. A more aggressive notch filter value was an option, with the maximum value of the filter set to ±5 sec at 80 nmi. The limit filter limited the speed-error value to ±10% of the ownship's nominal trajectory speed.



DTG < 20 nmi, notch = 0 sec,
DTG > 80 nmi, notch = ±15 sec,
otherwise the notch values change linearly between 80 nmi and 20 nmi

Figure 10. Notch filter schedule.

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

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

where this value is substituted for the nominal spacing time from the TTF data in figure 9.

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 time 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 ASTAR10 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, ASTAR10 projects the final approach speed deceleration backwards from the nominal beginning of the deceleration segment. Once the commanded speed point intercepts this deceleration line, ASTAR10 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, ASTAR10 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.

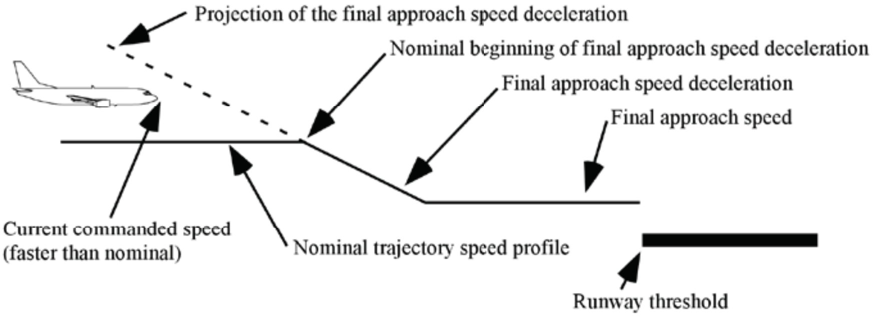

Projection of the final approach speed deceleration
Nominal beginning of final approach speed deceleration
Final approach speed deceleration
Final approach speed
Current commanded speed
(faster than nominal)
Nominal trajectory speed profile
Runway threshold

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

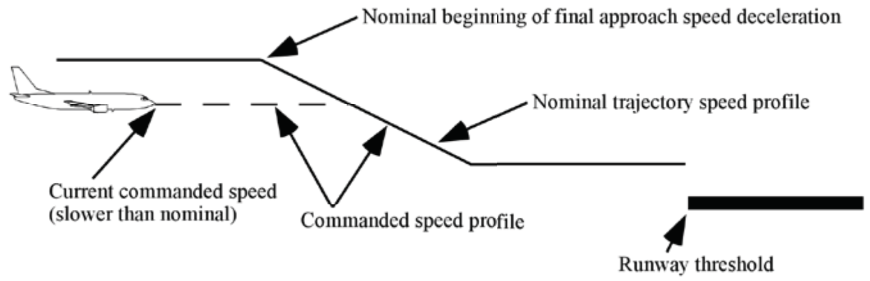

Nominal beginning of final approach speed deceleration
Nominal trajectory speed profile
Current commanded speed
(slower than nominal)
Commanded speed profile
Runway threshold

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

## Speed Change Minimization

As part of the ASTAR10 design, a low cost, aircraft retrofit option was considered. In this option, it was assumed that the speed command value would be presented to the pilot and the pilot would then either change the speed target of the autothrust system to match the commanded speed from ASTAR10 or would 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. 11) 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.

One means for reducing the number of speed changes would be to filter the spacing error value in the speed control law. This was done using a notch filter in the speed control law (fig. 10) that, when far from the runway threshold, allowed fairly large errors without inducing a speed correction. 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.

Another means for reducing the number of speed changes would be to use a quantization technique. By applying a quantization to the speed command prior to its output, the speed command changes would only occur in discrete intervals, thus reducing the number of commanded speed changes. For example, if the speed error (fig. 9) was to slowly change due to a change of the spacing error from 0 kt to 10 kt and the quantization value was 5 kt, then the output speed commands, instead of continually increasing to the 10 kt change, would be 0 kt, then 5 kt, and then 10 kt.

Lastly, a look-ahead option was also used to minimize the number of speed changes prior to a programmed deceleration segment, i.e., where the planned trajectory specifies a deceleration. In this regard, the algorithm would look ahead by 10 seconds in the nominal speed profile (fig. 13) to determine if a change onto a deceleration segment would occur. Within this 10 second interval, any speed command increase would be inhibited.

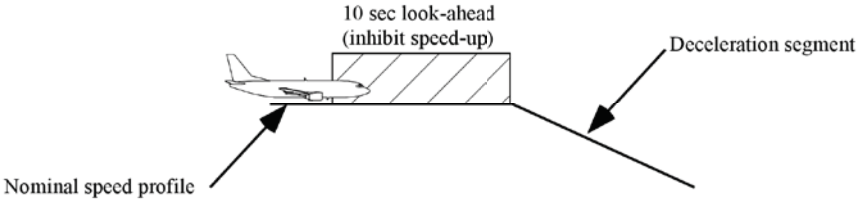


Figure 13. Look-ahead speed-up inhibit.

# Operational Considerations

## Spacing Interval for Parallel Dependent Approaches

From an operational standpoint, conducting parallel dependent approach operations can provide an increase in airport arrival rate relative to a single runway operation. To conduct this type of approach in the U.S., the minimum diagonal distance intervals used for parallel dependent approaches, i.e., staggered approaches, are 1.5 nmi and 2 nmi., with the interval choice dictated by the spacing between the runways. From a practical standpoint, however, successive approaches at these intervals usually do not occur due to other operational considerations (ref. 6). As with any instrument procedure, prior to initiating a staggered approach operation, the adjacent participating aircraft must either be vertically or longitudinally separated. If longitudinal separation is used, as would be dictated if the aircraft were performing Continuous Descent Approach (CDA) operations, then a 3 nmi minimum longitudinal separation would typically be required prior to the parallel dependent approach operation. In normal practice, a 3 nmi separation distance would result in an initial diagonal distance that is greater than 2 nmi. While the spacing aircraft could accelerate to obtain the 1.5 or 2 nmi diagonal distance interval, this catch-up technique would provide little system-wide operational benefit. For this situation where a diagonal distance interval is used following the use of a standard longitudinal separation technique, informal studies have suggested that a diagonal distance interval that corresponds to the longitudinal separation interval provides the greatest operational benefit.

## 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. 14). That is, common path points must have common speeds.

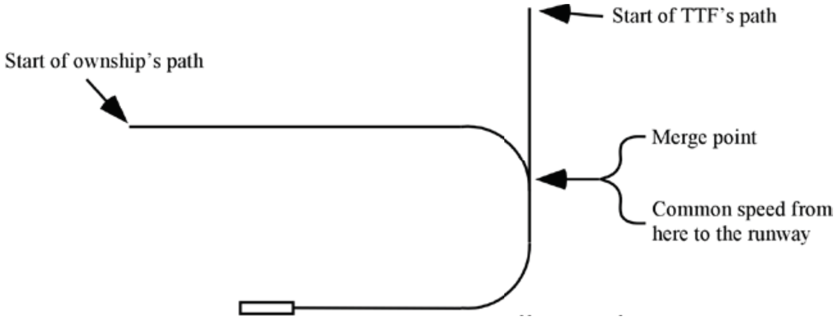


Figure 14. Example of common speeds after the merge point.

## Envelop Protection

Since the speed command value from ASTAR10 could be used to directly drive an autothrust system, speed envelop 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 ASTAR10. 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, ASTAR10 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, ASTAR10 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 9 is then:

$$\begin{aligned}&\textit{adjusted spacing time interval} =\\&\qquad \textit{adjusted spacing time interval}_{\textit{original}} - (\textit{current time} - \textit{crossing time}_{\textit{TTF}}),\end{aligned}$$

where the adjusted spacing time $\text{interval}_{\text{original}}$ is the original ATC assigned spacing interval along with any other adjustments, e.g., runway offset, and crossing $\text{time}_{\text{TTF}}$ 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 $\text{TTG}_{\text{TTF}}$ (fig. 9) is 0.

## 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 the aircraft assigned by ATC to follow. This latest version of this algorithm, referred to as ASTAR10, provides the ability to manage spacing against two traffic aircraft, with one of these aircraft operating to a parallel runway using dependent operations. Included in the support for parallel dependent runway operations is the capability to compute 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. In addition to describing the trajectory computations, spacing interval calculations, and the speed control law, this paper discusses several 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. *Air Traffic Control*. DOT/FAA Order JO 7110.65T, Feb. 11, 2010.

7. *Safety, Performance and Interoperability Requirements Document for Airborne Spacing – Flight Deck Interval Management (ASPA-FIM)*. RTCA DO-328, 2011.

8. Abbott, Terence S.: A Revised Trajectory Algorithm to Support En Route and Terminal Area Self-Spacing Concepts. NASA/CR-2010- 216204, 2010.

9. *Instrument Procedures Handbook*. DOT/FAA-H-8261-1A, 2007.

10. Abbott, Terence S: A Brief History of Airborne Self-Spacing Concepts. NASA/CR-2009-215695, 2009.

11. Norris, Guy: *UPS to Test 'SafeRoute' Next Year*, Flight International Magazine, p. 12, 9-15 Aug. 2005.

|  |  |
|---|---|
| **REPORT DOCUMENTATION PAGE** | *Form Approved*<br>*OMB No. 0704-0188* |

|  |  |  |
|---|---|---|
| **1. REPORT DATE** *(DD-MM-YYYY)*<br>01- 11 - 2011 | **2. REPORT TYPE**<br>Contractor Report | **3. DATES COVERED** *(From - To)* |

**4. TITLE AND SUBTITLE**

An Overview of a Trajectory-Based Solution for En Route and Terminal Area Self-Spacing to Include Parallel Runway Operations

**5a. CONTRACT NUMBER**
NNL10AA14B

**5b. GRANT NUMBER**

**5c. PROGRAM ELEMENT NUMBER**

**6. AUTHOR(S)**

Abbott, Terence S.

**5d. PROJECT NUMBER**

**5e. TASK NUMBER**

**5f. WORK UNIT NUMBER**
411931.02.61.07.01

**7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)**
NASA Langley Research Center
Hampton, VA 23681-2199

**8. PERFORMING ORGANIZATION REPORT NUMBER**

**9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)**
National Aeronautics and Space Administration
Washington, DC 20546-0001

**10. SPONSOR/MONITOR'S ACRONYM(S)**
NASA

**11. SPONSOR/MONITOR'S REPORT NUMBER(S)**
NASA/CR-2011-217194

**12. DISTRIBUTION/AVAILABILITY STATEMENT**
Unclassified - Unlimited
Subject Category 03
Availability: NASA CASI (443) 757-5802

**13. SUPPLEMENTARY NOTES**

This report was prepared by Booz Allen Hamilton under NASA contract NNL10AA14B with SGT, Inc., Greenbelt, MD.
Langley Technical Monitor: Bryan Barmore

**14. ABSTRACT**

This paper presents an overview of an algorithm specifically designed to support NASA's Airborne Precision Spacing concept. This airborne self-spacing concept is trajectory-based, allowing for spacing operations prior to the aircraft being on a common path. This implementation provides the ability to manage spacing against two traffic aircraft, with one of these aircraft operating to a parallel dependent runway. Because this algorithm is trajectory-based, it also has the inherent ability to support required-time-of-arrival (RTA) operations.

**15. SUBJECT TERMS**

spacing; traffic; runway; algorithm

| **16. SECURITY CLASSIFICATION OF:** |  |  | **17. LIMITATION OF ABSTRACT** | **18. NUMBER OF PAGES** | **19a. NAME OF RESPONSIBLE PERSON** |
|---|---|---|---|---|---|
| **a. REPORT** | **b. ABSTRACT** | **c. THIS PAGE** |  |  | STI Help Desk (email: help@sti.nasa.gov) |
|  |  |  |  |  | **19b. TELEPHONE NUMBER** ***(Include area code)*** |
| U | U | U | UU | 22 | (443) 757-5802 |