

# Adaptive Trajectory Prediction Algorithm for Climbing Flights

Charles A. Schultz<sup>1</sup>

*University-Affiliated Research Center, Moffett Field, CA, 94035-1000*

David Thipphavong<sup>2</sup>

*NASA Ames Research Center, Moffett Field, CA, 94035-1000*

Heinz Erzberger<sup>3</sup>

*NASA Ames Research Center, Moffett Field, CA, 94035-1000*

Aircraft climb trajectories are difficult to predict, and large errors in these predictions reduce the potential operational benefits of some advanced features for NextGen. The algorithm described in this paper improves climb trajectory prediction accuracy by adjusting trajectory predictions based on observed track data. It utilizes rate-of-climb and airspeed measurements derived from position data to dynamically adjust the aircraft weight modeled for trajectory predictions. In simulations with weight uncertainty, the algorithm is able to adapt to within 3 percent of the actual gross weight within two minutes of the initial adaptation. The root-mean-square of altitude errors for five-minute predictions was reduced by 73 percent. Conflict detection performance also improved, with a 15 percent reduction in missed alerts and a 10 percent reduction in false alerts. In a simulation with climb speed capture intent and weight uncertainty, the algorithm improved climb trajectory prediction accuracy by up to 30 percent and conflict detection performance, reducing missed and false alerts by up to 10 percent.

## Nomenclature

$a_0$	=	speed of sound at sea level (kt)
$a$	=	speed of sound (kt)
$D$	=	drag (lbf)
$\dot{E}_{obs}$	=	observed energy rate
$\dot{E}_{model}$	=	modeled energy rate
$\Delta \dot{E}$	=	energy rate difference, $\dot{E}_{obs} - \dot{E}_{model}$
$g$	=	gravitational acceleration (ft/sec <sup>2</sup> )
$\dot{h}$	=	rate of climb (ft/min)
$h_{pred}(t)$	=	predicted altitude at time $t$ (ft)
$h_{track}(t)$	=	actual altitude at time $t$ (ft)
$h_{error}(t)$	=	look-ahead altitude error at time $t$ (ft), $h_{pred}(t) - h_{track}(t)$
$L$	=	lift (lbf)
$m$	=	aircraft mass (lbm)
$p_0$	=	static pressure at sea level (kPa)
$p$	=	static pressure (kPa)
$T$	=	thrust (lbf)
$V_T$	=	true airspeed (kt)
$\dot{V}_T$	=	true airspeed acceleration (ft/sec <sup>2</sup> )
$V_{CAS}$	=	calibrated airspeed (kt)

<sup>1</sup> Software Engineer, Mail Stop 210-8, Charles.A.Schultz@nasa.gov, AIAA Member.

<sup>2</sup> Aerospace Engineer, Mail Stop 210-10, David.P.Thipphavong@nasa.gov, AIAA Member.

<sup>3</sup> Senior Advisor, Mail Stop 210-8, Heinz.Erzberger@nasa.gov, AIAA Fellow.

$W$	= aircraft weight (lbf)
$W_l$	= horizontal wind speed (kt)
$\gamma$	= ratio of specific heats at constant pressure and constant volume
$\gamma_a, \gamma_i$	= air-relative and inertial flight path angle (deg)
$\psi_i$	= inertial heading (deg)
$\psi_w$	= wind direction (deg)
$\psi_{rel}$	= relative wind angle, $\psi_i - \psi_w$

## I. Introduction

AIR traffic demand is expected to increase substantially over the next 20 years, but controller workload limits airspace capacity.<sup>1</sup> It is expected that higher levels of automation for separation assurance (SA) will be required to accommodate future demand growth. Trajectory prediction accuracy is essential to ensure both safety and efficiency in trajectory-based operations and especially automated conflict detection. For instance, previous research found an unacceptable number of late or missed conflict detections due to climb trajectory prediction uncertainty when using an automated conflict detection tool.<sup>2-3</sup> In addition, analysis of actual traffic in Fort Worth Center also suggested that more robust conflict detection could reduce the number of temporary altitude restrictions issued to climbing flights by around 70 percent.<sup>4</sup> An estimated 188 million gallons of fuel per year can be saved by allowing uninterrupted climbs at the 27 busiest airports in the U.S.<sup>5</sup>

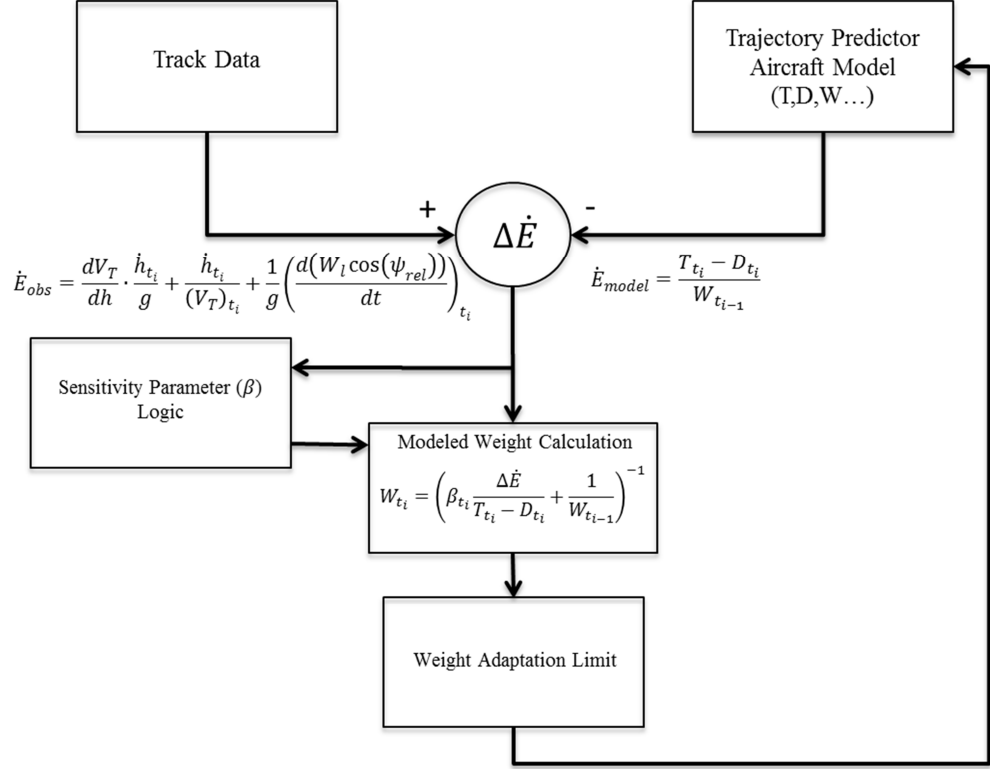
Researchers have studied various methods of improving climb trajectory prediction accuracy including the use of flight-planning data from airlines,<sup>6</sup> establishing real-time air-to-ground data link of flight parameters such as aircraft weight,<sup>7</sup> and improving aircraft performance models.<sup>8</sup> The first two methods have shown that a significant improvement in trajectory prediction accuracy is possible through the use of real-time data link of aircraft state. Of the aircraft states that were examined in these studies, aircraft weight and speed intent had the greatest influence on trajectory prediction accuracy, reducing the mean altitude error by 53 percent. Unfortunately, aircraft weight and speed intent are considered competitive parameters by airlines and are not available for use. This has led to the development of adaptive algorithms that compensate for uncertainty in aircraft state.

An approach to adapt climb predictions in real-time to more closely match observed track data has shown promise in past research,<sup>9-10</sup> but these algorithms were only evaluated with a few flights. As a result, it is not clear if similar reductions in climb trajectory prediction error could be achieved with these methods for all aircraft types, equipment, etc. The results demonstrated improved accuracy in terms of predicted top-of-climb time and range error. However, this does not necessarily imply that the entire trajectory prediction more closely matched actual track data because they did not report on altitude trajectory prediction errors for different look-ahead times. While these studies adjusted the thrust parameter used for trajectory predictions, a different approach is to adapt the more static weight parameter instead.<sup>11</sup> This method reduced the magnitude of climb prediction errors in half, but only for a limited set of 18 flights. By contrast, the purpose of the current study is to develop and evaluate an adaptive climb algorithm in terms of both trajectory prediction accuracy and conflict detection performance for a statistically significant number of flights consisting of many aircraft types.

## II. Algorithm Description

### A. General Concept

The adaptive climb algorithm uses observed aircraft track data to improve climb trajectory predictions by dynamically adjusting the modeled aircraft weight. It operates each time a track data update is received. At each update, the algorithm computes an observed energy rate from the track data and a modeled energy rate from the aircraft performance models used to generate trajectory predictions. The modeled weight is adjusted to bring the value of the modeled energy rate closer to the observed energy rate. From a physics perspective, the energy rate represents the rate of change of the kinetic and potential energy of the aircraft. A high-level overview is illustrated in Figure 1. The subscripts in the high-level illustration show how the energy rate is computed at observation time  $t_i$ .



**Figure 1: High-level overview of adaptive climb algorithm.**

## B. Energy Rate Calculation

The observed and modeled energy rates ( $\dot{E}_{obs}$  and  $\dot{E}_{model}$ ) for this algorithm are derived from the simplified form of the point-mass equations of motion of an aircraft in climb:<sup>12</sup>

$$\dot{V}_T = \frac{T - D}{m} - g\gamma_a - \frac{d(W_l \cos(\psi_{rel}))}{dt} \quad (1)$$

$$\dot{h} = V_T \sin(\gamma_a) \quad (2)$$

Dividing both sides of equation (1) by the gravitational constant  $g$ , substituting the aircraft weight  $W$  for  $mg$ , and rearranging the terms such that the left-hand side consists only of observable states and the right-hand side only of modeled aircraft parameters, results in a dimensionless form of equation (1):

$$\frac{1}{g} \dot{V}_T + \gamma_a + \frac{1}{g} \frac{d(W_l \cos(\psi_{rel}))}{dt} = \frac{T - D}{W} \quad (3)$$

For actual traffic, the true airspeed acceleration measurement is not sufficiently precise to estimate the energy rate, as it is derived from position data that have a 12-second update cycle. The  $\dot{V}_T$  term in equation (3) can be rewritten using the chain rule:

$$\dot{V}_T = \frac{dV_T}{dh} \frac{dh}{dt} = \frac{dV_T}{dh} \dot{h} \quad (4)$$

If the aircraft is flying a constant calibrated airspeed (CAS) to constant Mach climb profile,  $dV_T/dh$  can be assumed to be constant in the constant-CAS portion. A climb profile represents the flight procedure the pilot

executes from takeoff to the top-of-climb. A constant CAS to constant Mach climb profile is a flight procedure that is defined by the pilot setting the throttle to the maximum climb thrust and accelerating to a target CAS value. The pilot then maintains this CAS until a target Mach value is captured. When Mach capture occurs the pilot maintains this Mach value until the top-of-climb. Constant CAS to constant Mach climb profiles are common for commercial flights. In the constant-CAS portion of the climb,  $dV_T/dh$  can be assumed to be constant (see Appendix). Enabling the algorithm only during the constant-CAS portion of the climb (generally between 15,000 and 25,000 feet) permits the use of this assumption. Substituting equation (4) into equation (3) yields:

$$\frac{dV_T}{dh} \frac{\dot{h}}{g} + \gamma_a + \frac{1}{g} \frac{d(W_l \cos(\psi_{rel}))}{dt} = \frac{T - D}{W} \quad (5)$$

Equation (5) can be further simplified if the flight path angle  $\gamma_a$  is assumed to be small. An aircraft with a nominal climb rate of 2000 feet/min and ground speed of 450 knots has a flight path angle of approximately three degrees. Given,  $\gamma_a = \sin(\gamma_a)$  by the small angle approximation, equation (2) can be rewritten as:

$$\gamma_a = \frac{\dot{h}}{V_T} \quad (6)$$

Substituting equation (6) into equation (5) produces the dimensionless form of the energy rate equation:

$$\frac{dV_T}{dh} \frac{\dot{h}}{g} + \frac{\dot{h}}{V_T} + \frac{1}{g} \frac{d(W_l \cos(\psi_{rel}))}{dt} = \frac{T - D}{W}. \quad (7)$$

The left-hand side of the equation is composed entirely of state variables that can be derived from position data and wind state that is provided by the Rapid Update Cycle (RUC),<sup>13</sup> a numerical weather forecast model that updates hourly. Note the simulations run in this study did not incorporate wind. The left-hand side of equation (7) is the observed energy rate:

$$\dot{E}_{obs} = \frac{dV_T}{dh} \frac{\dot{h}}{g} + \frac{\dot{h}}{V_T} + \frac{1}{g} \frac{d(W_l \cos(\psi_{rel}))}{dt} \quad (8)$$

and the right-hand side is the modeled energy rate:

$$\dot{E}_{model} = \frac{T - D}{W} \quad (9)$$

Generally the observed energy at the current time is not equal to the modeled energy rate computed using the current modeled thrust  $T_{t_i}$  and drag  $D_{t_i}$  and the previously modeled aircraft weight  $W_{t_{i-1}}$ . The temporal subscripts are added to illustrate the actual implementation of the algorithm. The previously modeled weight is used because fuel burn is not computed between time steps when using the algorithm. The energy rate difference is obtained by taking the difference of equation (8) from equation (9):

$$\Delta \dot{E} = \left( \frac{dV_T}{dh} \right)_{t_i} \frac{\dot{h}_{t_i}}{g} + \frac{\dot{h}_{t_i}}{(V_T)_{t_i}} + \frac{1}{g} \left( \frac{d(W_l \cos(\psi_{rel}))}{dt} \right)_{t_i} - \frac{T_{t_i} - D_{t_i}}{W_{t_{i-1}}} \quad (10)$$

### C. Weight Adaptation Method

In equation (10) the modeled thrust, drag, or weight parameters can be adjusted to minimize the energy rate difference. The modeled weight was selected for this implementation of the algorithm because the error in weight is relatively static compared to the thrust and drag. The magnitude of the thrust and drag are largely dependent on the

altitude and the thrust can change rapidly depending on the throttle setting. To minimize the energy rate difference a new modeled weight  $W_{t_i}$  is selected using the current modeled thrust  $T_{t_i}$  and drag  $D_{t_i}$  such that the modeled energy rate equals the current observed energy rate:

$$W_{t_i} \cdot \frac{T_{t_i} - D_{t_i}}{W_{t_i}} = \dot{E}_{obs t_i} \quad (11)$$

Using equation (11) it is possible to match the observed energy rate in a single iteration. However, in previous work using real track data the measured state was often volatile and successive trajectory predictions were inaccurate in nature.<sup>11</sup> To resolve this, a sensitivity parameter  $\beta_{t_i}$  was introduced that regulates the adaptation speed as a way to improve the robustness to noise. This is important because a single uncharacteristic adaptation can completely negate the cumulative improvement. Also, due to the limited period in which the algorithm operates, there may not be enough time to recover. The temporal subscript is used because the value of the sensitivity parameter can change between iterations. To incorporate  $\beta_{t_i}$  into the algorithm, the energy rate difference defined in equation (10) was rewritten by substituting in equation (11) for the observed energy rate, and re-arranged to isolate the updated model weight:

$$W_{t_i} = \left( \frac{1}{W_{t_{i-1}}} + \frac{\Delta \dot{E}}{T_{t_i} - D_{t_i}} \right)^{-1} \quad (12)$$

The sensitivity parameter is applied to the fraction with the  $\Delta \dot{E}$  term since the energy rate difference is computed using imprecise position data:

$$W_{t_i} = \left( \frac{1}{W_{t_{i-1}}} + \beta_{t_i} \frac{\Delta \dot{E}}{T_{t_i} - D_{t_i}} \right)^{-1} \quad (13)$$

#### D. Sensitivity Parameter Logic

A simplistic approach would be to set  $\beta_{t_i}$  to a constant value. However, such an approach does not perform well when the track data are volatile or when there is an atypically large energy rate difference. In the case of volatile track data, a slower, more conservative approach is effective. On the other hand, an atypically large energy rate difference requires a more aggressive approach with a higher adaptation speed. To accommodate for these scenarios, a robust approach was used. The sensitivity parameter was designed as a flexible parameter

$$\beta_{t_i} = \begin{cases} \max(0.205, \beta_{t_{i-1}} + 0.05) & \text{if } i > 0, \Delta \dot{E}_{t_i} > 0.0001, \left| \frac{\Delta \dot{E}_{t_i} - \Delta \dot{E}_{avg}}{\Delta \dot{E}_{avg}} \right| < 3 \\ 0.005 & \text{otherwise} \end{cases} \quad (14)$$

$$\Delta \dot{E}_{avg} = \frac{\sum_{j=1}^5 \Delta \dot{E}_{t_{i-j}}}{5}$$

The logic that sets the magnitude of the sensitivity parameter  $\beta_{t_i}$ , depicted in equation (14), was designed to evaluate the integrity of the observed energy rate calculation and adjust the sensitivity parameter accordingly. To do this the current energy rate difference  $\Delta \dot{E}_{t_i}$  is compared to the mean of the five previously computed energy rate differences. A threshold was set that is used to determine if the current energy rate difference is an outlier. For this study, the threshold was set to three times the average energy rate difference  $\Delta \dot{E}_{avg}$ . If the current energy rate calculation is determined to be an outlier, the sensitivity parameter is reset to a conservative value to minimize the magnitude of the adaptation for that iteration. If it is not classified as an outlier, the algorithm increments the sensitivity parameter by a fixed amount up to a maximum value. The maximum value was added to limit the aggressiveness of the algorithm to ensure smooth incremental progress. Note the numerical values in equation (14) were selected based on performance using an iterative method and are specific to the simulation environment that was used in this study.

Additionally, constraints were placed on the magnitude of the change in weight between iterations and the overall cumulative adapted weight. The magnitude of the change in weight between iterations was restricted to one percent of the aircraft's nominal modeled weight as an absolute limit on the effect an outlier energy rate difference can have on the adaptation. The constraint placed on the cumulative adapted weight limits the adapted weight value to be within 80 and 120 percent of the nominal modeled aircraft weight. This range of acceptable weight values was selected based on observed weight variations from Coppenbarger's 2001 study<sup>7</sup>.

### III. Evaluation of Algorithm in Fast-Time Simulations

#### A. Simulation Environment

For this study, the Airspace Concepts Evaluation System (ACES)<sup>13</sup> was used to simulate the National Airspace System (NAS). ACES is a fast-time, gate-to-gate simulation of the NAS that generates four-degree-of-freedom trajectories of aircraft based on aircraft performance models from the Base of Aircraft Data (BADA)<sup>14</sup> from departure fix to arrival fix. The routes and departure times for the aircraft in the simulation are derived from actual routes flown on a specific day in the NAS. This platform was used for the capability to evaluate a large number of flights consisting of many aircraft types, the ability to precisely define the amount and types of uncertainty applied and the ability to evaluate conflict detection performance. The uncertainty model and conflict detection process in ACES are described in detail in [15].

#### B. Limitations of ACES

ACES is an effective platform for evaluating this algorithm, however it does have limitations. There is a discrepancy between track data generated using ACES and Host radar track data. The track data generated in ACES are the result of continuous integration of the equations of motion and as such do not contain noise. This is not the case for Host radar track data, and poses a significant problem for tools that require state feedback. This is one of the reasons that the adaptive algorithm was first developed and evaluated using ACES instead of a platform like the Center-TRACON Automation System (CTAS) that uses recorded track data. Future work is planned to incorporate feedback uncertainty in ACES.

Another discrepancy that is unique to ACES is related to the horizontal conformance of the aircraft. All flights in ACES are in-conformance horizontally throughout the duration of flight. However, from the point of view of an automated conflict detection system, this would not hold true for actual Host track data. Among other reasons, this is because it would not be aware of the majority of route clearances that are issued by controllers. A comparison of controller clearances by voice to flight plan amendments entered found that while 95 percent of altitude clearances were entered, less than one-third of route clearances were entered.<sup>16</sup> Thus, the improvements in conflict detection performance achieved in this study with the adaptive algorithm may not be realized in practice without horizontal conformance.

#### C. Experiment Setup

To evaluate the adaptive algorithm, two simulations were run using a traffic file with approximately 4800 departure flights. In these simulations the algorithm was applied to climbing flights starting at the first track at or above 15,000 ft. up to the first track above 25,000 ft. to ensure the aircraft was in the constant-CAS portion of climb. The algorithm's time step was set to 12 seconds to match the radar track update rate in Center airspace.

The first simulation served as a proof-of-concept and was designed to evaluate how well the algorithm reduced trajectory prediction error in the sole presence of weight uncertainty. Trajectory prediction error was defined as the altitude error of the trajectory prediction with respect to actual track of the aircraft computed as a function of look-ahead time  $t$ :

$$h_{error}(t) = h_{pred}(t) - h_{track}(t) \quad (15)$$

In ACES, weight uncertainty is applied to the fuel weight parameter of the aircraft. Fuel-weight uncertainty was uniformly distributed between -50 and +50 percent to the modeled nominal gross fuel weight.

The second simulation was designed to test the algorithm in a more realistic environment with uncertainty in climb intent in addition to fuel weight. For this experiment, uncertainty was uniformly distributed between -10 and

+10 percent to the modeled capture CAS and Mach values associated with the constant CAS to constant Mach climb profile. The fuel-weight uncertainty was uniformly distributed between -50 and +50 percent to the nominal modeled gross fuel weight. Since there is climb intent and fuel weight uncertainty the adapted weight is not expected to converge with the actual weight of the aircraft. The algorithms ability to reduce trajectory prediction error and the resulting impact on conflict detection performance were examined. The configuration details for both experiments are described in Table 1.

**Table 1: Configuration Details of Simulations**

<b>Simulation</b>	<b>Fuel Weight Uncertainty</b>	<b>Capture CAS Uncertainty</b>	<b>Capture Mach Uncertainty</b>
1	uniform(-50%,+50%)	N/A	N/A
2	uniform(-50%,+50%)	uniform(-10%,+10%)	uniform(-10%,+10%)

Conflict detection performance was measured by identifying discrepancies between conflict detections made using the adapted and non-adapted trajectory predictions and the true conflict detections identified using the flown trajectories of the aircraft. A conflict is defined as a point in time when two aircraft are separated by less than 5 nautical miles horizontally and 1,000 feet vertically. Two cases called missed alerts and false alerts were quantified and compared. A missed alert is an instance where a conflict detection is made at point in time using the flown trajectory of the aircraft but at the same point in time is not detected using the adapted or non-adapted trajectory predictions. A false alert is an instance at a point in time where a conflict detection is made for a non-existent conflict using the adapted or non-adapted trajectory predictions.

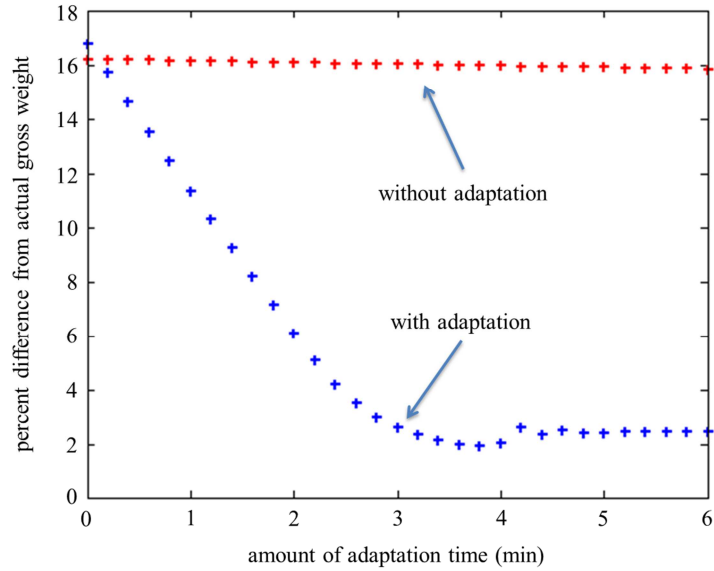
The terminology in Table 2 below is used to describe the results of these experiments.

**Table 2: Terminology used to describe the results of these experiments**

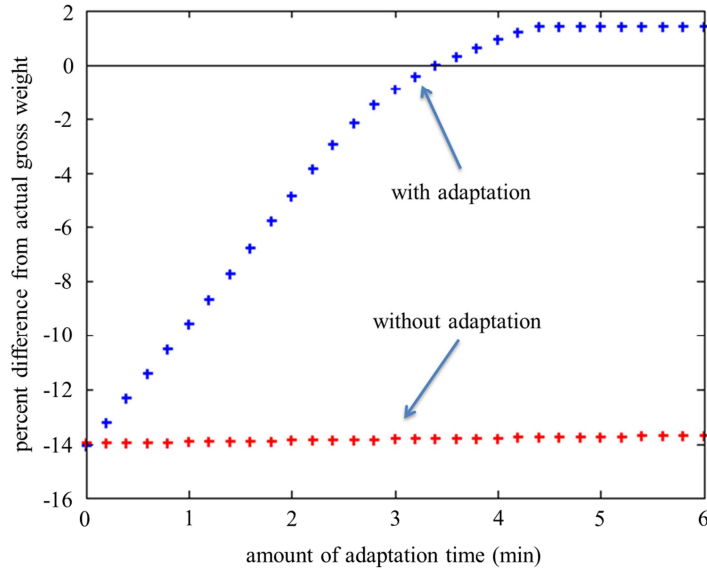
<b>Term</b>	<b>Definition</b>
Actual Weight	True gross aircraft weight of the simulated aircraft
Adapted Weight	Gross aircraft weight modeled using adaptive algorithm
Non-Adapted Weight	Gross aircraft weight with uncertainty included
Adapted Trajectory	4-D trajectory prediction generated using adapted weight
Non-Adapted Trajectory	4-D trajectory prediction generated using non-adapted weight

#### **D. Simulation with Fuel Weight Uncertainty**

To evaluate the adaptive algorithm's performance, the adapted weight was compared to the actual weight of the aircraft at successive iterations. The same equations of motion and atmospheric data are used to generate the trajectory predictions and simulate the modeled aircraft; therefore with just weight uncertainty, the adapted weight is expected to converge to the actual weight. A case where the algorithm adapts from a weight that is higher than the actual weight of the aircraft is illustrated in Figure 2. A case where the opposite is true is illustrated in Figure 3. Note in these figures the red line that represents the non-adapted weight has a non-zero slope due to fuel burn.



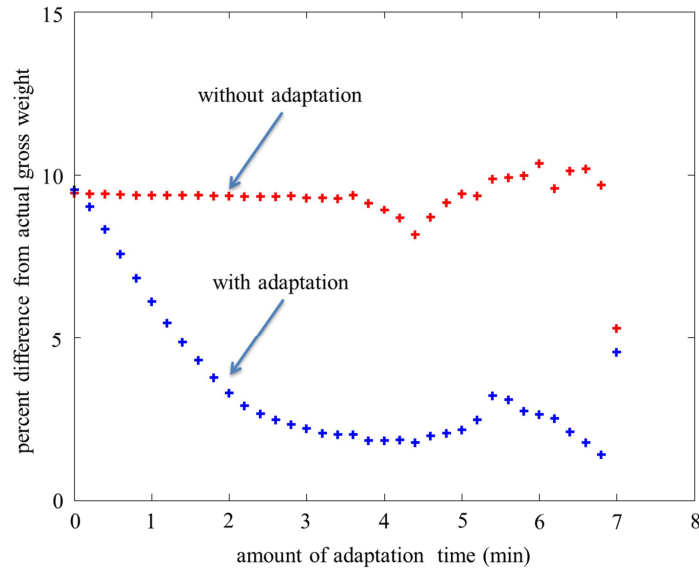
**Figure 2: Example of adaptation with heavier non-adapted weight.**



**Figure 3: Example of adaptation with lighter non-adapted weight.**

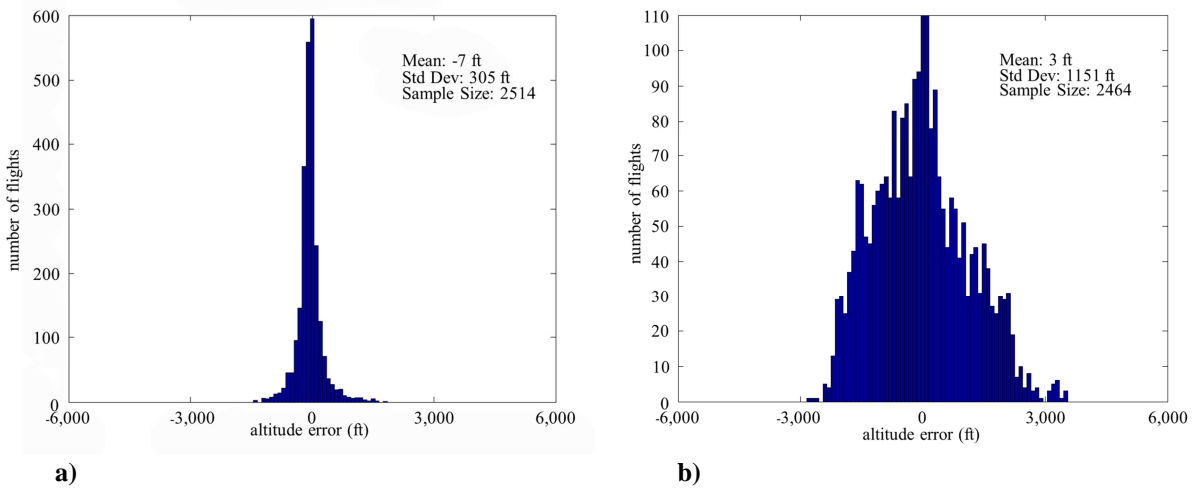
The algorithm adjusts the aircraft weight gradually and smoothly in these two cases (blue line). Furthermore, once the adapted weight converges to within 3 percent of the actual weight, the adapted weight continues to track the actual weight throughout the rest of climb. Note the algorithm does not completely converge with the actual weight of the aircraft because of the assumption that  $dVT/dh$  is constant and the use of the sensitivity parameter  $\beta_{t_i}$ . These features are especially relevant for separation assurance because inaccurate trajectory predictions can lead to erroneous conflict detections and unnecessary resolution maneuvers. This same general behavior was observed at an aggregate level (see Figure 4).



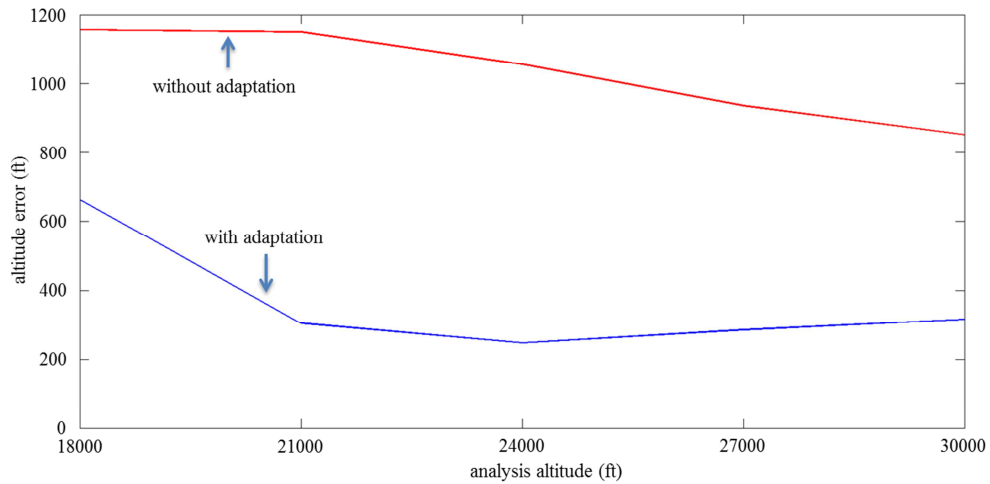


**Figure 4: Root-mean-square of gross weight error as a function of time.**

The convergence of the adapted weight to the actual weight results in major improvements in trajectory prediction accuracy. Consider the histograms below of altitude prediction errors for a five-minute look-ahead time at the first track above 21,000 feet (see Figure 5) with and without adaptation. Recall that the algorithm was enabled at the first track above 15,000 feet, thus, was in operation for two to three minutes by this point. The algorithm effectively reduced the standard deviation of the altitude error from 1,151 feet to 305 feet, a 73 percent improvement. Improvement occurs across a range of analysis altitudes starting at 18,000 feet up to 30,000 feet (see Figure 6). Note the improvement in trajectory prediction accuracy decreases with altitude because the aircraft is approaching top-of-climb.



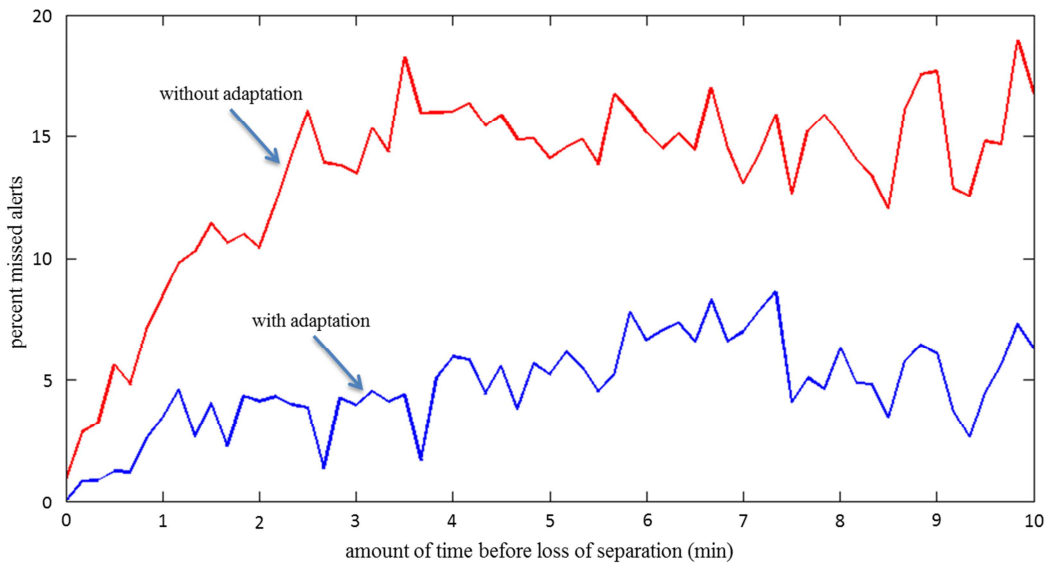
**Figure 5: Five minute look-ahead altitude error improvement at 21,000 feet with adaptation (a) compared to without adaptation (b).**



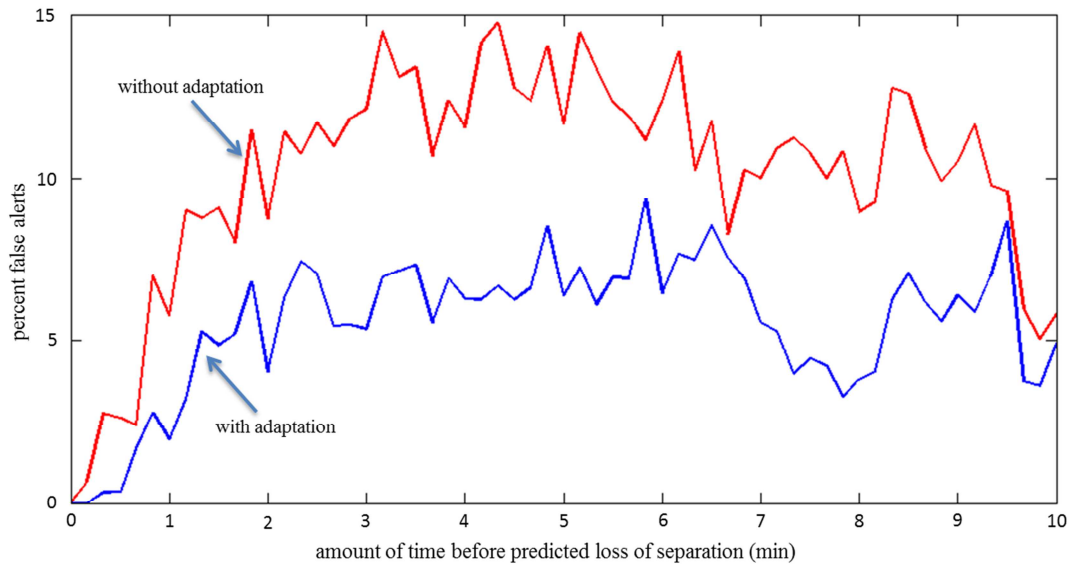
**Figure 6: Root-mean-square of five minute look-ahead altitude error during climb.**

Given the overall improvement in trajectory prediction accuracy the next step was to examine the impact on conflict detection performance. Missed alerts and false alerts were evaluated for conflicts that involved flights at a minimum altitude of 18,000 feet where at least one of the aircraft was in climb. The minimum altitude ensures that adaptation occurs for at least one of the flights. The results are plotted in Figure 7 for missed alerts and Figure 8 for false alerts.

The algorithm reduced the number of missed alerts for all look-ahead times by up to 15 percent. It also reduced the number of false alerts by up to 10 percent for all look-ahead times. Note for missed alerts and false alerts the results for times less than one minute prior to loss of separation are similar because the difference between an adapted and non-adapted trajectory are small for such a short prediction time. Overall, these results indicate a significant improvement in conflict detection performance.



**Figure 7: Missed alert rate.**

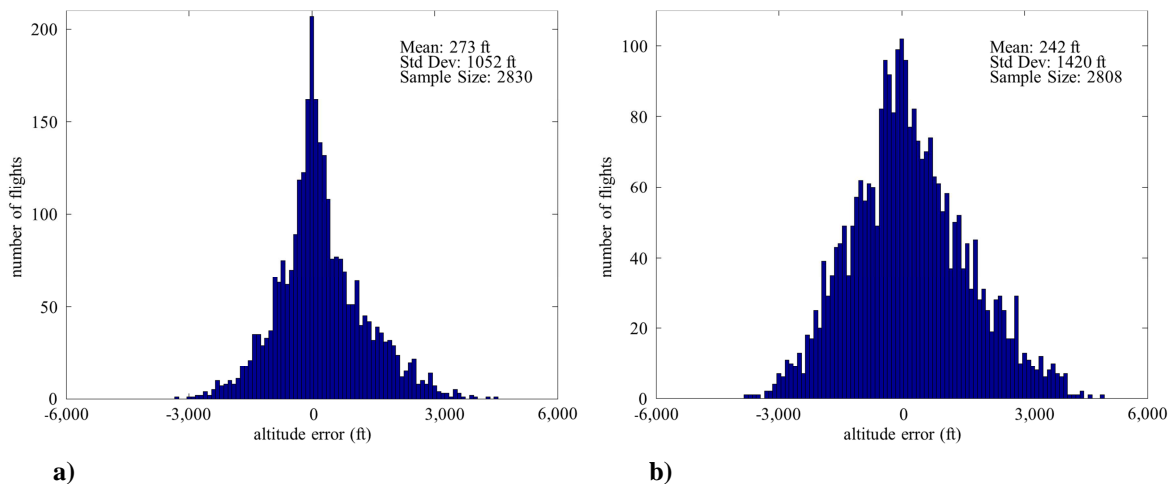


**Figure 8: False alert rate.**

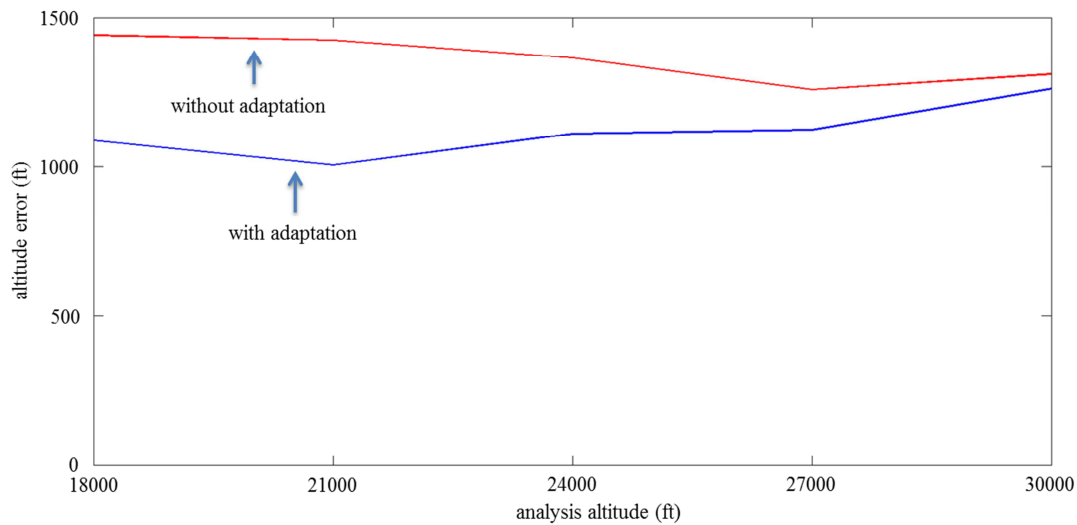
The algorithm has proven to be successful in compensating for weight uncertainty alone. However, in actual operations there are many sources of uncertainty. One major source of uncertainty is in the aircraft's climb intent. If the aircraft flies a flight procedure that is different from what is modeled, large trajectory prediction errors can occur.

#### **E. Simulation with Climb Intent and Fuel Weight Uncertainty**

Now that the algorithm's ability to improve trajectory prediction accuracy and conflict detection performance has been demonstrated the next step was to evaluate in a more realistic scenario. The five-minute look-ahead altitude error was evaluated at 21,000 feet (see Figure 9) with and without adaptation. The standard deviation of the altitude error was reduced by 26 percent with respect to the non-adapted trajectory prediction. A more comprehensive examination indicates that a reduction in five minute look-ahead altitude error occurs for analysis altitudes starting at 18,000 feet up to 30,000 feet (see Figure 10). Note the improvement in trajectory prediction accuracy decreases with altitude because the aircraft is approaching top-of-climb.



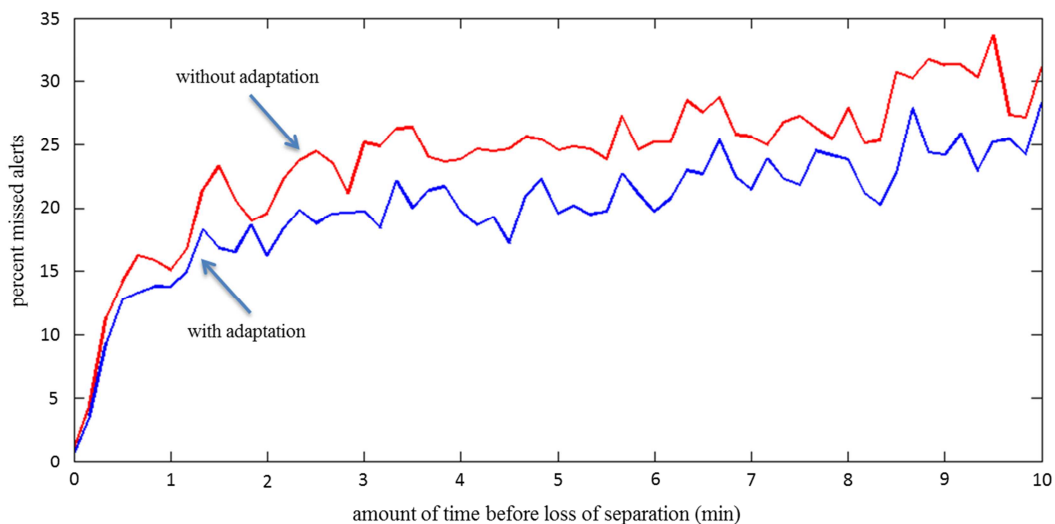
**Figure 9: Five minute look-ahead altitude error at 21,000 feet with weight and climb profile uncertainty for the adapted (a) and non-adapted (b) cases.**



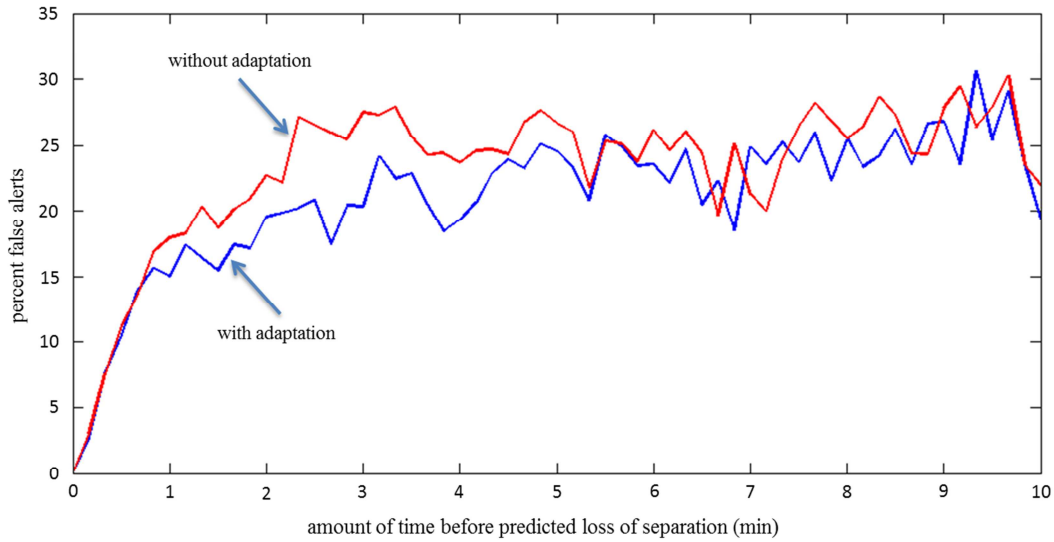
**Figure 10: Root-mean-square of five minute look-ahead altitude error during climb**

Given the improvement in trajectory prediction accuracy, conflict detection performance was examined. Missed alerts and false alerts were evaluated for conflicts that involved flights at a minimum altitude of 18,000 feet where at least one of the aircraft was in climb. The minimum altitude ensures that adaptation occurs for at least one of the flights. The results are plotted in Figure 11 for missed alerts and Figure 12 for false alerts.

The use of the algorithm reduced the number of missed alerts at all times prior to the conflict by up to 10 percent. For false alerts there is up to a 10 percent reduction that occurs between one minute and five minutes prior to the conflict. For times greater than five minutes prior to loss of separation, the algorithm shows little improvement with respect to the non-adapted trajectory predictions.



**Figure 11: Percentage of missed alerts made before loss of separation**



**Figure 12: Percentage of false alerts made before loss of separation**

The algorithm improved trajectory prediction accuracy by 20 to 30 percent and improved conflict detection performance by reducing the number of missed alerts and false alerts made up to five minutes prior to the conflict by up to 10 percent. The addition of climb profile uncertainty significantly reduced the improvement in trajectory prediction accuracy.

This reduction in trajectory prediction accuracy emphasizes the importance of accurate climb profile information. Climb profile and associated capture speeds (e.g., CAS and Mach) will remain unknown to the automation unless it is published in the flight plan or made available via data-link. These missing pieces of information are detrimental to the algorithm because it only adapts based on the most recent track data. It implicitly assumes that the actual flight will climb according to the profile in the aircraft model. If this does not hold, then the climb predictions made using the adaptive algorithm will be accurate for short look-ahead times of up to a couple minutes, but less accurate for longer look-ahead times.

## IV. Conclusion

The algorithm described in this paper can improve trajectory prediction accuracy for climbing flights significantly and as a result improve conflict detection performance. The adaptive climb algorithm dynamically adjusts the aircraft weight modeled for trajectory predictions based on observed rate-of-climb  $\dot{h}$  and true airspeed  $V_T$  feedback. In simulations with just weight uncertainty, the standard deviation of altitude errors for five-minute look-ahead predictions was reduced by 73 percent. When climb speed capture intent uncertainty was also modeled, the algorithm reduced the standard deviation of five-minute look-ahead altitude errors by 20 to 30 percent. The improvement in trajectory prediction accuracy correlated to better conflict detection performance. For the case with climb speed capture intent uncertainty, the number of missed and false alerts made up to five minutes prior to loss of separation was reduced by as much as 10 percent when using the algorithm.

Overall this study has demonstrated that the adaptive weight algorithm can be used as a foundation for improving trajectory prediction accuracy to the extent necessary for higher-levels of separation assurance automation needed to increase the capacity of the Next Generation Air Transport System.

## V. Future Work

### A. Rate-Of-Climb Feedback Uncertainty

In this study, the observed state feedback used by the algorithm to compute the observed energy rate did not contain noise. Thus, the observed energy rate was representative of the actual energy rate of the aircraft. In actual operations, however, there is significant noise in the observed state because it is derived from imprecise track data that is updated on a 12-second cycle. It is important for the algorithm to be capable of computing a representative observed energy rate using noisy track data otherwise the improvements in trajectory prediction accuracy and

conflict detection performance may not be realized. A future study using ACES is planned to evaluate the adaptive weight algorithm with rate-of-climb feedback uncertainty.

## B. Evaluation using CTAS

The next step in the validation process of the adaptive weight algorithm is to evaluate its performance in the presence of real-world uncertainties. This will be accomplished by implementing the algorithm in CTAS, which is a real-time research prototype system developed at NASA that includes mature capabilities for 4-D trajectory predictions, conflict detection, conflict resolution, and other functions [17-18]. Multiple days of actual Fort Worth Center traffic will be evaluated. In this study trajectory prediction accuracy will be the only metric since the CTAS input files include controller actions.

## C. Terminal Airspace Application

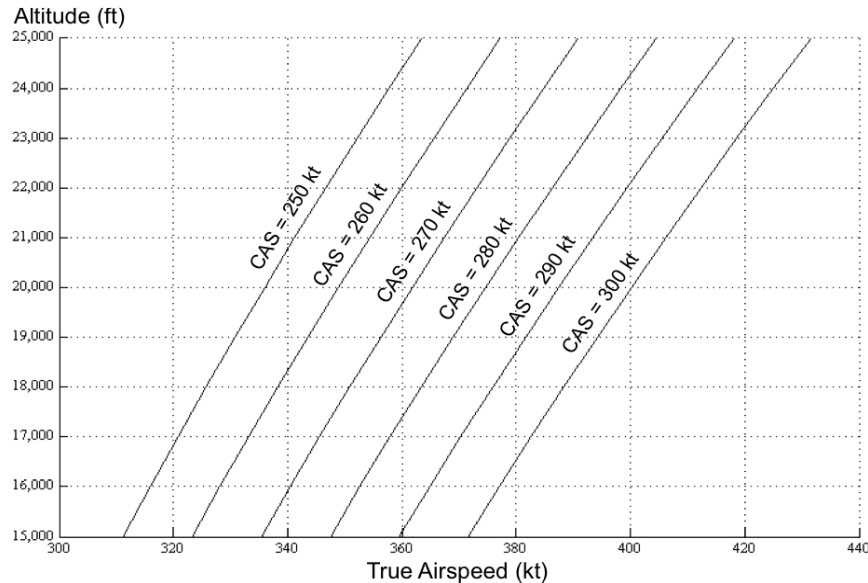
The adaptive climb algorithm developed in this paper is currently only enabled when climbing flights are above 15,000 feet. By the time an aircraft has reached 15,000 feet in most cases the acceleration segment has ended and the desired CAS has been captured. Recall that this is because the derivation of the airspeed-altitude gradient ( $dV_T/dh$ ) assumes that the flight is in the constant-CAS portion of climb.

In terminal airspace a CAS speed restriction is placed on aircraft at altitudes less than 10,000 feet. For jet aircraft, this speed restriction is generally enforced at 250 knots. Given the speed restriction, aircraft tend to accelerate to a CAS value that is close to the speed restriction and maintain this speed until eclipsing 10,000 feet. As such, the algorithm could be enabled as soon as it is determined the flight is maintaining constant CAS. The algorithm would be disabled once the flight reaches 10,000 feet, but subsequent trajectory predictions would use the adapted weight instead of the nominal value.

This could result in more accurate climb trajectory predictions below 15,000 feet. When the flight reaches 15,000 feet, the algorithm would then be re-enabled using the last adapted weight. This would likely result in a smaller energy rate gap, shorter adaptation time, and greater climb trajectory accuracy and improved conflict detection performance when the algorithm is re-enabled at 15,000 feet.

## Appendix: Constant True Airspeed-Altitude Gradient Assumption

If an aircraft is maintaining constant CAS in climb the true airspeed-altitude gradient ( $dV_T/dh$ ) can be approximated as a constant. This is evident when plotting constant CAS lines on a true airspeed-altitude plot (see Figure 13). The constant CAS lines are approximately linear.



**Figure 13: Constant CAS Lines on True Airspeed vs. Altitude Plot**

Figure 13 was generated using the Standard Atmospheric model<sup>19</sup> and the CAS to true airspeed conversion equation:<sup>20</sup>

$$V_T = a \sqrt{\frac{2}{\gamma - 1} \left( \frac{p_0}{p} \left( 1 + \frac{\gamma - 1}{2} \left( \frac{V_{CAS}}{a_0} \right)^2 \right)^{\frac{\gamma}{\gamma - 1}} - 1 \right)} \quad (16)$$

### Acknowledgments

The authors would like to thank Todd Lauderdale for his help and guidance in developing the ACES code for the experiments conducted in this study.

### References

- <sup>1</sup>Joint Planning and Development Office, “Next Generation Air Transportation System Integrated Plan,” Dec. 12, 2004.
- <sup>2</sup>McNally, D. and Thipphavong, D., “Automated Separation Assurance in the Presence of Uncertainty,” 26<sup>th</sup> International Congress of the Aeronautical Sciences, Sep. 14-19, 2008.
- <sup>3</sup>Thipphavong, D., “Analysis of Climb Trajectory Modeling for Separation Assurance Automation,” AIAA Guidance, Navigation, and Control Conference and Exhibit, Aug. 18-21, 2008.
- <sup>4</sup>McNally, D., et al., “A Near-Term Concept for Trajectory-Based Operations with Air/Ground Data Link Communication,” 27<sup>th</sup> International Congress of the Aeronautical Sciences, Nice, France, Sep. 19-24, 2010.
- <sup>5</sup>National Aeronautics and Space Administration, “Green Aviation: A Better Way to Treat the Planet.”
- <sup>6</sup>Coppenbarger, R.A., “Climb Trajectory Prediction Enhancement Using Airline Flight-Planning Information,” AIAA Guidance, Navigation, and Control Conference, Aug. 1999.
- <sup>7</sup>Coppenbarger, R.A., “Real-Time Data Link of Aircraft Parameters to the Center-TRACON Automation System (CTAS),” 4<sup>th</sup> USA/Europe ATM R&D Seminar, Dec. 3-7, 2001.
- <sup>8</sup>Chan, W., Bach, R., and Walton, J., “Improving and Validating CTAS Performance Models,” AIAA Guidance, Navigation, and Control Conference and Exhibit, Aug. 14-17, 2000.
- <sup>9</sup>Slater, G.L., “Adaptive Improvement of Aircraft Climb Performance for Air Traffic Control Applications,” Proceedings of the IEEE International Symposium on Intelligent Control, Oct. 27-30, 2002.
- <sup>10</sup>Godbole, A.A., “Adaptive Improvement of Climb Performance,” Division of Research and Advanced Studies of the University of Cincinnati, Master of Science Thesis, 2003.
- <sup>11</sup>Schultz, C., “Adaptive Trajectory Prediction Feedback Algorithm for Climbing Flights,” AIAA Region III Student Conference, Apr. 2, 2011.
- <sup>12</sup>Slattery, R., Zhao, Y., “Trajectory Synthesis for Air Traffic Automation,” *Journal of Guidance, Control and Dynamics*, Vol. 20, No. 2, Mar.-Apr. 1997.
- <sup>12</sup>Benjamin, S.G., Devenyi, D., Weygandt, S.S., Brundage, K.J., Brown, J.M., Grell, G.A., Kim, D., Schwartz, B.E., Smirnova, T.G., Smith, T.L., and Mankin, G.S., “An Hourly Assimilation/Forecast Cycle: The RUC,” *Monthly Weather Review*, 132, 495-518.
- <sup>13</sup>“Build 4 of the Airspace Concepts Evaluation System,” AIAA Modeling and Simulation Technologies Conference and Exhibit, No. AIAA 2006-6110, 2006.
- <sup>14</sup>European Organisation for Safety of Air Navigation (Eurocontrol), User Manual for the Base of Aircraft Data (BADA), Revision 3.9, EEC Note No. 11/03/08-08, Eurocontrol Experiment Centre, Apr. 2011.
- <sup>15</sup>Lauderdale, T. A., “The Effects of Speed Uncertainty on a Separation Assurance Algorithm,” 10<sup>th</sup> AIAA Aviation Technology, Integration, and Operations (ATIO) Conference, Fort Worth, TX, 13-15 Sep. 2010.
- <sup>16</sup>Paglione, M., Bayraktutar, I., McDonald, G., and Bronsvort, J., “Lateral Intent Error’s Impact on Aircraft Prediction,” 8<sup>th</sup> USA/Europe ATM R&D Seminar, Jun. 29 – Jul. 2, 2009.

<sup>17</sup>Erzberger, H., Davis, T. J., and Green, S. M., “Design of Center-TRACON Automation System,” *Proceedings of the AGARD Guidance and Control Panel 56<sup>th</sup> Symposium on Machine Intelligence in Air Traffic Management*, 1993, pp. 11.1-11.2

<sup>18</sup>McNally, D., and Gong, C., “Concept and Laboratory Analysis of Trajectory-Based Automation for Separation Assurance,” *Air Traffic Control Quarterly*, Vol. 15(1), pp. 35-63, 2007.

<sup>19</sup>Nation Oceanic and Atmospheric Administration, Nation Aeronautics and Space Administration, and U.S. Air Force, U.S. Standard Atmosphere, 1976, U. S. Government Printing Office, Washington DC, 1976.

<sup>20</sup>Collinson, R. P. G., *Introduction to Avionic Systems*, 3<sup>rd</sup> ed., Springer-Verlag, New York, 2011, pp. 395-396.  
doi: 10.1007/978-94-007-0708-5