(Preprint) AAS 13-736

AUTONOMOUS AEROBRAKING DEVELOPMENT SOFTWARE: PHASE 2 SUMMARY

Alicia D. Cianciolo,[*] Robert W. Maddock,[†] Jill L. Prince,[‡] Angela Bowes,[§] Richard W. Powell,[**] Joseph P. White,[††] Robert Tolson,[‡‡] Daniel O'Shaughnessy,[§§] David Carrelli[***]

NASA has used aerobraking at Mars and Venus to reduce the fuel required to deliver a spacecraft into a desired orbit compared to an all-propulsive solution. Although aerobraking reduces the propellant, it does so at the expense of mission duration, large staff, and DSN coverage. These factors make aerobraking a significant cost element in the mission design. By moving on-board the current ground-based tasks of ephemeris determination, atmospheric density estimation, and maneuver sizing and execution, a flight project would realize significant cost savings. The NASA Engineering and Safety Center (NESC) sponsored Phase 1 and 2 of the Autonomous Aerobraking Development Software (AADS) study, which demonstrated the initial feasibility of moving these current ground-based functions to the spacecraft. This paper highlights key state-of-the-art advancements made in the Phase 2 effort to verify that the AADS algorithms are accurate, robust and ready to be considered for application on future missions that utilize aerobraking. The advancements discussed herein include both model updates and simulation and benchmark testing. Rigorous testing using observed flight atmospheres, operational environments and statistical analysis characterized the AADS operability in a perturbed environment.

INTRODUCTION

Aerobraking uses a planet's atmosphere rather than propulsion to put a spacecraft into a specific orbit. Doing so reduces launch mass or accommodates the mass of additional science instruments rather than propellant. Aerobraking has been used successfully four times, once at Venus and three times at Mars.[1,2,3,4] However, the costs of the ground operations staff and Deep Space Network (DSN) coverage for the three to six months of aerobraking are substantial. The costs are incurred due to the current nature of aerobraking operations, which require continuous DSN coverage and a staff to monitor the atmosphere and spacecraft heating. Additionally, statis-

[*] Aerospace Engineer, Atmospheric Flight and Entry Systems Branch (AFESB), NASA LaRC, Hampton VA 23681.
[†] Aerospace Engineer, AFESB, NASA LaRC, MS 489 Hampton VA 23681.
[‡] Aerospace Engineer, AFESB, NASA LaRC, MS 489 Hampton VA 23681.
[§] Aerospace Engineer, AFESB, NASA LaRC, MS 489 Hampton VA 23681.
[**] Aerospace Engineer, AFESB, Analytical Mechanics and Associates MS 489 Hampton VA 23681.
[††] Senior Project Engineer, AFESB, Analytical Mechanics and Associates, MS 489 Hampton VA 23681.
[‡‡] Professor Emeritus, National Institute of Aerospace, 100 Exploration Way, Hampton VA 23666.
[§§] Principal Staff, Space Department, Johns Hopkins Applied Physics Laboratory, Laurel, MD 20723.
[***] Senior Staff, Space Department, Johns Hopkins Applied Physics Laboratory, Laurel, MD 20723.

tical analysis of future orbits is performed to inform decisions regarding the size, direction, and timing of the maneuvers that maintain the desired flight profile. Substantial cost savings could be realized if the ground-based operations could be automated and moved onboard the spacecraft.

Prior aerobraking missions have made strides to use onboard processing to automate some routine operational functions. The Mars 2001 Odyssey aerobraking mission flew with a periapsis timing estimator as part of its onboard software. An ephemeris model calculated a-priori timing of periapsis and drag pass change in velocity magnitudes that were compared to the spacecraft drag pass accelerometer data. The calculated time of periapsis passage was compared to the actual time of periapsis. Had the software been fully activated, the change in periapsis time could have modified the sequence command timing and reduced the number of ground-based uplinks required. However, the timing estimator was only allowed to operate in shadow mode.

It was the Mars Reconnaissance Orbiter (MRO) mission 4 years later that first demonstrated operational use of a periapsis timing estimator. The updated time estimation algorithm was intended to limit sequence builds to once per day during the prime shift. The better-than-expected performance allowed the estimator to be used extensively when the orbit period dropped below 24 hours. By the time the orbit period dropped below 11 hours, the ground crew was able to uplink 6 day long pass sequences significantly reducing the number of uplinks.[5] The estimator worked well in short orbit periods when the apoapsis altitude is changing slowly. However, ground teams still performed analysis to verify the state update.

The successful demonstration of periapsis timing estimation on MRO prompted the continued investigation into the feasibility of moving additional ground-based processes on board, specifically the ability to allow the spacecraft to determine the direction and magnitude and ultimately command the maneuvers that maintain the desired flight profile. That was the focus of Autonomous Aerobraking Development Software Phase 1 Study.[6,7,8] To accomplish this, aerodynamic and thermal models for a representative spacecraft were developed for both the onboard algorithm known as Autonomous Aerobraking Development Software (AADS) and a ground-based simulation developed for testing purposes. Autonomous ephemeris, atmosphere, and maneuver estimators were also developed and incorporated into AADS. AADS was tested in simulations at three destinations, Mars, Venus, and Titan, and included atmospheric perturbations and sensor (inertial measurement unit (IMU)) measurement errors. Phase 1 simulation analysis of AADS demonstrated that the algorithm could provide state estimates within the 225 s periapsis timing error, used during MRO aerobraking, for more than a week before requiring a ground update, thus eliminating the need for continuous DSN coverage.

The follow-on AADS Phase 2 was a 12-month effort to improve and rigorously test the AADS developed in Phase 1. This paper summarizes progress made in Phase 2 to verify that the AADS algorithms are accurate, robust and ready to be considered for application to future missions that utilize aerobraking. The discussion will focus specifically on the model improvements and simulation and trades that substantially advanced the state-of-the-art of AADS.

MODEL IMPROVEMENTS

AADS consists of four main estimator models: state propagator, atmosphere, thermal and maneuver. The Phase 2 effort focused primarily on AADS performance at Mars because of the amount of flight data for comparison. A few enhancements were made to the thermal model but, since the thermal model is primarily used for Venus missions and is spacecraft specific, it is not discussed here. Details of the thermal model can be found in the AADS Phase 2 final report due out later this year.[9] Descriptions of the updates to the other models are provided below.

State Propagator

The primary function of the State Propagator (SP) is to provide predictions of the 3 degree-of-freedom (3-DOF) spacecraft position and velocity, for use in the atmosphere estimator (AE) and maneuver estimator (ME). This requires a high-fidelity integration of the spacecraft translational equations of motion. Ground operations provide the initial condition for the integration (termed a "ground update"). The spacecraft uses this initial state coupled with onboard force models and measurement data to propagate the ground update forward in time. Because the desired integration interval is long (defined by this project to be at least seven days), the integration must be performed as accurately as possible because the position errors grow quadratically with time.

The most significant model changes from the Phase 1 AADS were made to the SP (formerly called the "ephemeris estimator"). The model was re-architected to allow ready conversion to a flight environment (i.e., real-time processing). Additionally, by modifying the orbital integrations algorithms, performance improvements in orbital integration accuracy, software execution speed, and software memory footprint were also obtained. Unlike the buffered measurement data for batch processing used by AADS in the Phase 1 ephemeris estimator, the Phase 2 SP was designed to run synchronously with the IMU/star tracker output data rates. To ensure maximum flexibility, these rates are user selectable in the software, but were nominally set to 10 Hz for the work discussed here. The key benefit to this modification is that the computing resources required by the AADS on the host spacecraft are dramatically reduced.

The SP is divided into two functions: (1) state determination (SD) or determining the current spacecraft translational state, and (2) orbit propagation (OP) or propagating of the current state forward in time to predict a future orbital condition (periapsis or apoapsis). The SD process integrates the last known state forward in time using the IMU measurements and onboard gravity models to propagate the state to the current time. Once each orbit, the OP process takes the current state and predicts ahead one orbit to provide estimates of the subsequent periapsis and apoapsis states to the AE and ME for corridor maneuver computations. The SD runs in real-time, although at a low rate (~0.01-Hz) as it continually updates the translational state based on the sensed accelerations. The OP is a batch process that only runs once per orbit to feed the AE and ME algorithms.

To accurately perform the SD, all accelerations acting on the vehicle must be included in the integration. These accelerations can be broadly grouped into two categories: gravitational and non-gravitational. In general, the gravitational accelerations are modeled in the software, and the non-gravitational accelerations are measured with the accelerometers. The dominant gravitational acceleration is due to the central body, but other perturbing bodies may also be modeled. For instance, aerobraking at Venus would require the sun gravitational field be included, and Saturn's if aerobraking at Titan. The dynamics of the motion due to gravity change slowly over time, allowing for a large integration time step. Additionally, the gravitational accelerations are computationally expensive, so there is a strong desire to limit the number of evaluations of these models. In contrast, the non-gravitational accelerations, which can change quickly with time, particularly aerodynamic drag and thrust acceleration, necessitate a small integration step. The competing step size requirements for the orbital integration create a computationally stiff system.

The AADS solution to the stiff orbital integration problem is to separate the fast and slow dynamics in the SP, and integrate them with different methods and time steps. The slow dynamics are due to the gravitational accelerations and are evaluated using a variable-step Runge-Kutta-Nystrom (RKN) method that allows minimal computation of the expensive harmonic gravity models. The fast dynamics of the non-gravitational terms is integrated with a simple fixed-step trapezoidal integration of the input accelerometer data at (or close to) the measurement data rate.

This hybrid (fast/slow) integration used in the SP serves to minimize the computational burden without sacrificing accuracy in the resultant state. The architecture changes described above reduces the memory requirements of AADS by a factor of nearly 100 over the Phase 1 integration method.

Atmosphere Estimator

The statistical model used for density dispersions in the Phase 1 AADS AE was found to provide uncertainties of the estimated density and scale height that were much smaller than the natural variability seen during previous aerobraking missions. To improve the uncertainty estimation, the AE was redesigned to use the deviation between the predicted values and the observed values from a number of orbits (N) prior to the one being predicted (i.e., orbit N+1). Two studies were performed: one to determine the optimal averaging method and a second to determine the optimal number of orbits to include in the selected averaging method. Previous aerobraking mission flight data is not used for prediction because scaled season, altitude, time, and location on the planet do not necessarily generate accurate predictions.

The first study considered a number of averaging methods that were tested against previous aerobraking mission data sets. Methods included arithmetic mean, geometric mean, median, and harmonic mean. The goal was to find a method that permitted an unusually large change in density to increase the uncertainty, but not to influence the uncertainty over a long period of time. For example, if the algorithm averages over the previous seven orbits and then the simulation experienced an orbit with an unusually high density then the method would provide a high mean and standard deviation for the next six predicts. This method is intended to protect against situations like that which occurred during the Odyssey aerobraking mission on orbit 106 when the observed heat rate was near the thermal margin limit after several orbits with considerably lower heat rates.

The second study evaluated the number of orbits (N) that could provide a reasonable set of prediction statistics (mean and standard deviation) without changing the variability. It became clear that different approaches provided the best estimates depending on the latitude of periapsis, but the differences were smaller than the predicted variability and not statistically significant.

The results of both studies yielded an uncertainty model that used, for simplicity, the arithmetic mean and the standard deviation from seven orbits (N=7). This approach was tested and performed as well as the ground-based operations simulations.

Maneuver Estimator

For Phase 2 analyses, the ME was updated to utilize the standard deviation (1 sigma uncertainty) on density now provided by the AE. When the ME predicts the spacecraft's next atmospheric pass maximum heat rate relative to the operational corridor, it also calculates the X sigma high (or low, in the case of the altitude corridor) heat rate, where X is specified by an AADS iLoad parameter. The value X is a multiplier on the 1 sigma uncertainty provided by the AE. This additional data is output for informational purposes, but can also be used to modify the AADS ME maneuver logic. If instructed to do so (through an AADS iLoad parameter), the ME can compare the X-sigma estimate against an "immediate action line" (also input through an AADS iLoad). The immediate action line is a limit that, if exceeded, requires the spacecraft to perform a maneuver to raise the periapsis altitude (i.e. lower the heat rate) on the next apoapsis. If, for example, the X=3 sigma uncertainty prediction exceeds the immediate action line, the ME passes into its "bias maneuver" logic which uses the predicted 3 sigma uncertainty value in the maneuver calculation instead of the mean. The intent of this capability is to further protect the spacecraft from any instances where the AE density uncertainty is sufficiently high as to cause concern that an immediate action line violation may occur on the next atmospheric pass.

TRADES AND ANALYSIS

Several simulation updates and trades were performed during Phase 2. AADS, with the model updates described above, was incorporated into the Program to Optimize Simulated Trajectories (POST2). The POST2 simulation, used in Phase 1, was upgraded to an all-C version, which allows for faster run times and greater flexibility, efficiency, and capability in the simulation. Two such studies that utilized the POST2 simulation to characterize the state-of-the art capabilities of AADS are presented here. They include (1) realistic aerobraking mission Operational Readiness Tests (ORT) to characterizes AADS performance capabilities in a flight like operational environment and (2) a statistical Monte Carlo analysis that demonstrates AADS robustness to dispersions. A third trade also presented here, uses the Johns Hopkins University/Applied Physics Laboratory (JHU/APL) Autonomous Aerobraking High Fidelity Simulation (AAHFS) to demonstrate AADS flight software performance through benchmark testing.

Operational Readiness Test

A key advantage of autonomous aerobraking is that it can significantly reduce ground staff workload. However, without a mission to quantify the savings, a flight system training mechanism, known as the Operational Readiness Test, was used to compare the workload of a traditional ground-based operations to an AADS based operations. In addition to quantifying staff workload requirements, the ORT demonstrates AADS capabilities in a flight-like operational environment.

An ORT is designed to evaluate the ability of a team to perform all mission critical tasks prior to the commencement of a mission. The key to a successful ORT is the simulation and processes that recreate the flight and operational experience. By focusing the AADS Phase 2 analysis as an ORT at Mars, the simulation could take advantage of observed density profiles from the three previous aerobraking missions at the planet (from MGS, ODY and MRO) and demonstrate the AADS capabilities in a realistic mission operations setting. The performance and advantages of AADS are evaluated by running two aerobraking ORTs simultaneously: one in the historical ground-based manner (ORT1) and one using AADS (ORT2). So, not only is the AADS prediction and reaction being compared with what happens on a spacecraft (ORT2), but it is also being compared to the decisions that a ground-based team would have made with a similar spacecraft and environment (ORT2 vs. ORT1).

A key advancement in the AADS Phase 2 evaluation was the incorporation of realistic atmosphere profiles from previous Mars aerobraking missions into the POST2 simulation. Assessment of AADS performance in a realistic and perturbed flight environment required realistic atmosphere profiles compared to the bell shaped density versus altitude curves in numerical atmosphere models like the Mars Global Reference Atmosphere Model v2010(MarsGRAM)[10] that was used in the Phase 1 analysis. Due to the differences in the states of the simulated AADS missions compared to the actual Mars aerobraking missions, observed flight atmosphere profiles were scaled in both time and altitude (scale height) to preserve the observed structure.

Reference Mission Simulation: A reference mission is used in the ORT to provide the glide slope (in this case it is a plot of local true solar time of the equatorial crossing of the ascending node vs. time) that all ORT missions attempt to maintain using appropriate maneuver decisions. Remaining on or near the glide slope ensures that the ORT simulation will finish at the required condition. The simulated reference aerobraking mission used to assess AADS, shown in Figure 1, is similar to the Mars Odyssey mission. Odyssey was the most aggressive (maintaining lowest thermal margin) aerobraking mission to date and was considered a stress case with which to evaluate AADS performance. The simulated spacecraft, however, is modeled using the spacecraft mass properties and aerodynamics from the Mars Reconnaissance Orbiter. The reference aerobraking mission begins in an 18-hour orbit period and includes walk-in from a periapsis altitude of 200 km. The walk-in assumes three periapsis altitude-reducing maneuvers performed on alternating apoapses to initiate aerobraking. The flight corridor parameter is a heat rate indicator based on ½ ρv^3 (where ρ is density and v is spacecraft velocity). The desired peak heat rate on each periapsis pass is targeted to remain in a very small corridor (between 0.102 and 0.104 W/cm^2) by performing maneuvers at apoapsis to reach a final apoapsis of 400 km in approximately 70 days. See red dots in Figure 1. A second similar simulation is used to determine the initial flight corridor for both ORT1 and ORT2. This mission is allowed to make a maneuver once every two days, like past aerobraking operations, and targets heat rates to remain between 0.085 and 0.14 W/m^2 also shown in Figure 1. This simulation is known as the operational mission runout. The walk-out phase of the mission that puts the spacecraft into the desired science orbit is not considered in either the reference or operational simulation.

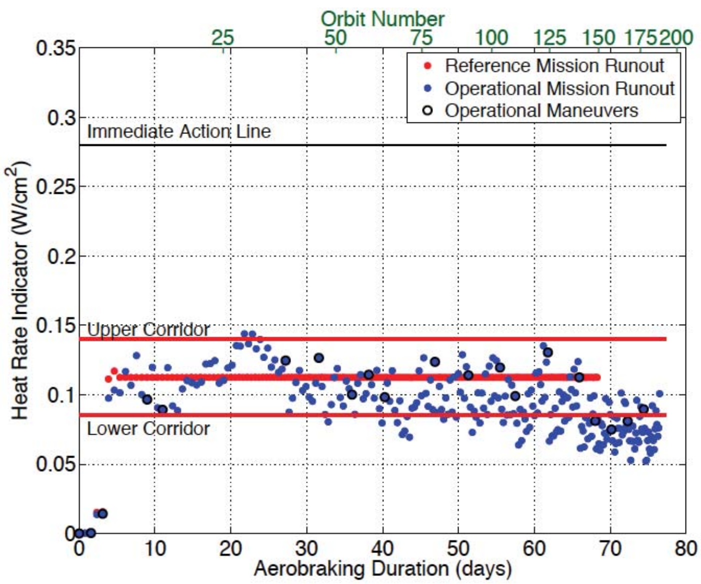

Figure 1. Flight corridor performance for the reference mission run-out.

A reference mission run-out is a standard practice for all aerobraking missions and provides a guideline (glideslope) by which real-time analyses can be compared. The goal of both ORTs was to end at the reference mission's final local true solar time (LTST) of the equatorial crossing of the ascending node which was at 10:46 hours +/-15 minutes. The 15-minute margin was a requirement set by the science instrument operability for the Odyssey aerobraking mission but it also provided flexibility in terminating aerobraking during the end-game phase of the mission. The following section will describe the different simulations and analyses that comprise the ORTs. Then the results of ORT1 and ORT2 are presented.

ORT Simulations and Analysis. Each ORT uses two types of simulations. The first is the *spacecraft flight simulation* that emulates the "real" environment and the spacecraft flight though it. The simulation flies through scaled Mars atmosphere aerobraking flight density profiles rather than MarsGRAM modeled atmospheres. The simulation is intended to emulate the NAV function of uploading a maneuver to the spacecraft and the DSN communication link providing the latest spacecraft state. The second simulation type is the *mission runout simulation.* This simulation uses MarsGRAM modeled atmospheres and allows the user to specify the flight corridor and the maneuver frequency. This aerobraking mission simulation can be "runout" for a specified time interval (e.g. to the end of the mission or just for a week) to understand sensitivities to atmosphere dispersions. This simulation is used for planning and assessing the progress of the aerobraking mission parameters that will be "uploaded" or used in the *spacecraft flight simulation*. The runout simulation is used to evaluate different corridor options and determine the changes to the corridor required to satisfy mission termination constraints. In the case of ORT1, it is also used to evaluate maneuver options and inform the maneuver decision process.

The analysis performed using the *mission runout simulation* is different for ORT1 and ORT2. ORT1, the ground supported mission, must perform analysis each day to determine if and when a maneuver should be made and also its size and direction; all functions that are automated by AADS in ORT2. Due to the time and staff required to build and upload maneuver commands during real time operations, the ground team in ORT1 was allowed only one opportunity each day to make a maneuver and maneuvers could not be made on consecutive days. The ORT1 daily analysis consisted of 13 *mission runout simulations* of the next five to seven days. Each simulation evaluated a burn of different magnitude at the orbit designated for burn opportunity. For example, consider the most recent simulated navigation state received from spacecraft (from the *spacecraft flight simulation*) corresponding to the apoapsis state at orbit 93. The next opportunity to perform a maneuver is on obit 95 that occurs on the next day. The spacecraft has stored onboard a maneuver menu, or a fixed set of maneuver sizes and direction options ranging from $\pm$0.6 m/s. A negative change in velocity implies a burn direction that will lower periapsis altitude, thus increasing heat rate and vice versa for positive values. The daily analysis runs a *mission runout simulation* spanning the next 5 days for each maneuver in the menu. The results, shown in Figure 2, are plotted relative to the desired flight corridor and immediate action line. In this example, if no maneuver is performed, the spacecraft heat rates will likely fall below the lower corridor, extending the mission duration and increasing the error in the final targeted LTST. In this case, mission planners recommended performing a -0.1 m/s maneuver that would keep the periapsis heat rates in the next planning interval within the desired corridor. Once approved by the mission manager, the command was "uploaded" to the *spacecraft flight simulation.* This maneuver choice had to keep the spacecraft in the expected range of conditions for the next 48 hours, as there would be no maneuver option the following day. The *spacecraft flight simulation* would then update the spacecraft state and simulate the next 24 hours of flight. The spacecraft state at the apoapsis nearest the 24 hours would be "down linked" to the ORT1 ground team for them to evaluate the maneuver options in the next two days. The cycle repeated until the aerobraking mission termination conditions were met.

In addition to the daily analysis, there was also weekly analysis performed for ORT1 to assess the overall aerobraking margin to the targeted final LTST defined by the reference mission. This margin is determined by flying three simulated missions that target exactly the heat rate of the upper, middle, and lower corridor. The simulations are initiated with the latest spacecraft state at apoapsis and end at 2800 km apoapsis altitude (the point at which the use of AADS will cease for this ORT). The results are then compared to the reference mission to determine if the corridor should be adjusted to meet the desired termination conditions. Figure 3 shows the LTST results of the three simulations that are initialized with the spacecraft apoapsis state on day 16. The horizontal dashed red lines denote the desired target range at the end of the mission (EOM). In this case, if the mission continues to fly near the lower corridor, the mission will miss the target LTST of the ascending node by 20 minutes. Therefore, a near-term maneuver strategy should focus on remaining in the upper part of the corridor for a few orbits in order to raise the flight at the lower corridor back into the final mission LTST limits during the next weekly runout. Based on the weekly runout results, the team had the option to modify the corridor limits. In this example, the ORT1 ground team opted to increase the lower limit from 0.085 to 0.1 W/cm^2 and also to raise the upper corridor from 0.14 to 0.15 W/cm^2 for the upcoming week. The analysis was repeated with the spacecraft state at the end of each week of the mission.

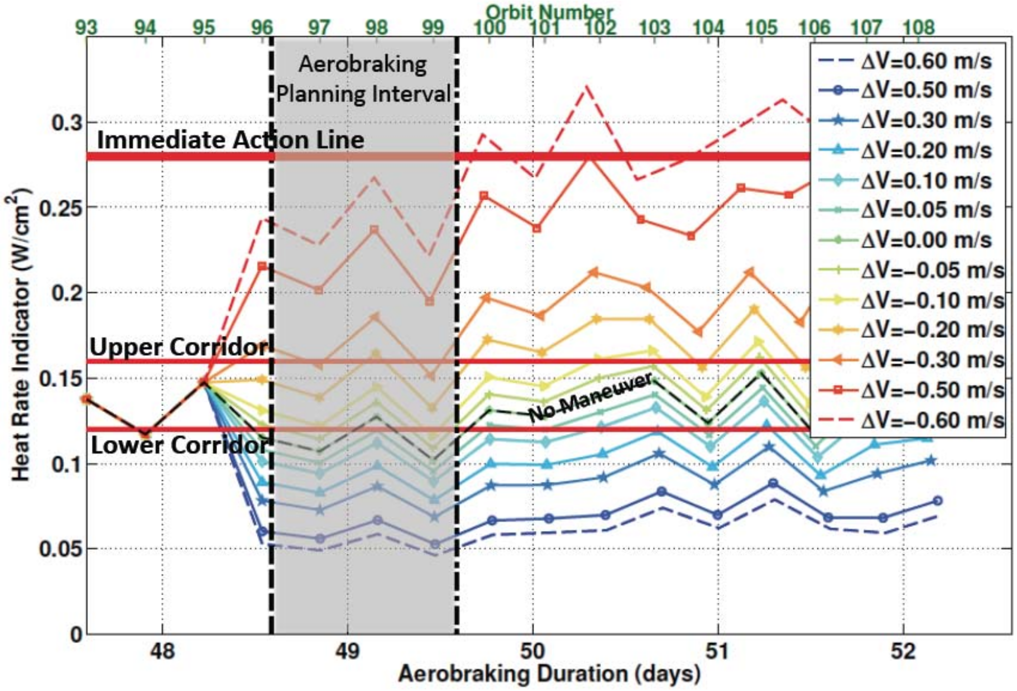

Figure 2. ORT1 Daily Analysis

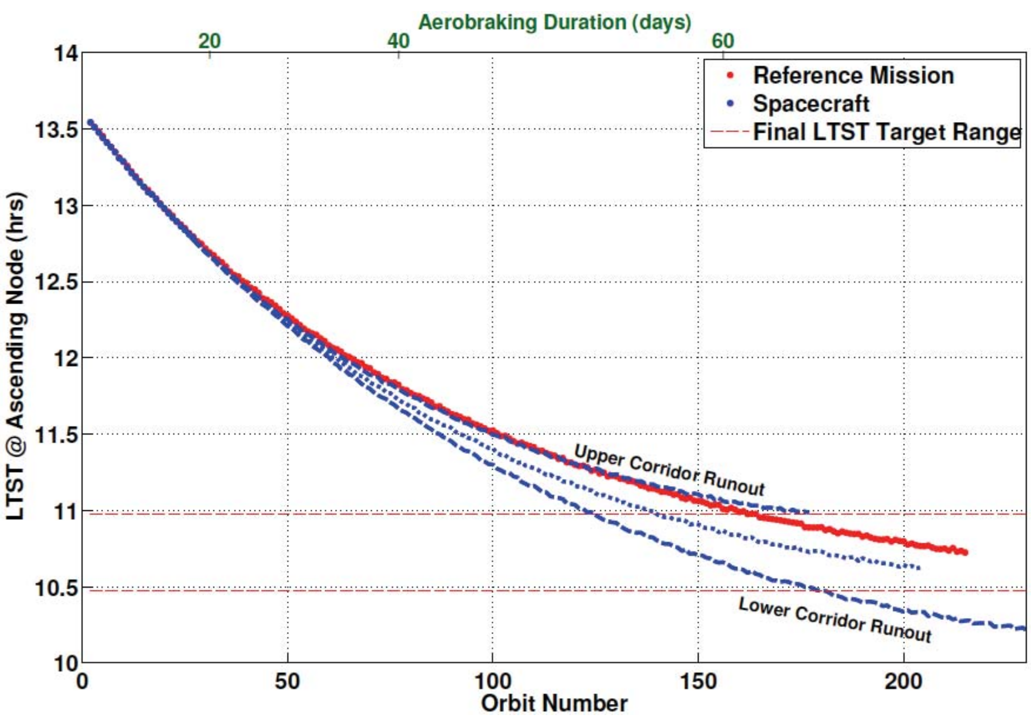

Figure 3. ORT1 Weekly Analysis for Targeted Final LTST

For ORT2, which used AADS to predict the atmosphere, propagate the state and calculate (and execute if needed) a maneuver each orbit, no daily analysis was required. ORT2 evaluated how effective AADS was at limiting propagation errors by evaluating the algorithsms ability to maintain the flight corridor. State updates occurred once per week. Weekly analysis was performed to determine if and how the corridor limits should change. The results are similar to those shown in Figure 3. It is noted that the modifications made to the corridor based on the AADS weekly analysis was different from those selected for ORT1.

ORT Results. For ORT1, the corridor was modified three times throughout the 70-day mission. Figure 4 shows the corridor with both the predicted and actual heat rates observed on each orbit. Because of the lower than expected densities, much of the mission flew near or below the lower corridor (see the blue dots in Figure 4). To remain on the target LTST "glide slope", the lower corridor was raised. Additionally, because of the observed lower than expected variability in the atmospheric density, the ground team also decided to raise the upper corridor. This effectively reduced the thermal margin, but did so to make effective use of the atmosphere when risk was perceived to be low.

Near the end of the mission, when the orbit period is less than 3 hours, there are 8 orbits occurring between maneuver opportunities. Maneuvers are determined based on the latest atmosphere trends provided from the Atmosphere Team and the daily analysis. However, in the ground-based ORT1, the variability in the atmosphere near the end of the mission was not predicted in the model and therefore the heat rate on three orbits violated the immediate action line before a maneuver was performed to raise the periapsis altitude effectively reducing the periapsis heat rate. No automated pop-up command was implemented in the *spacecraft flight simulation.*

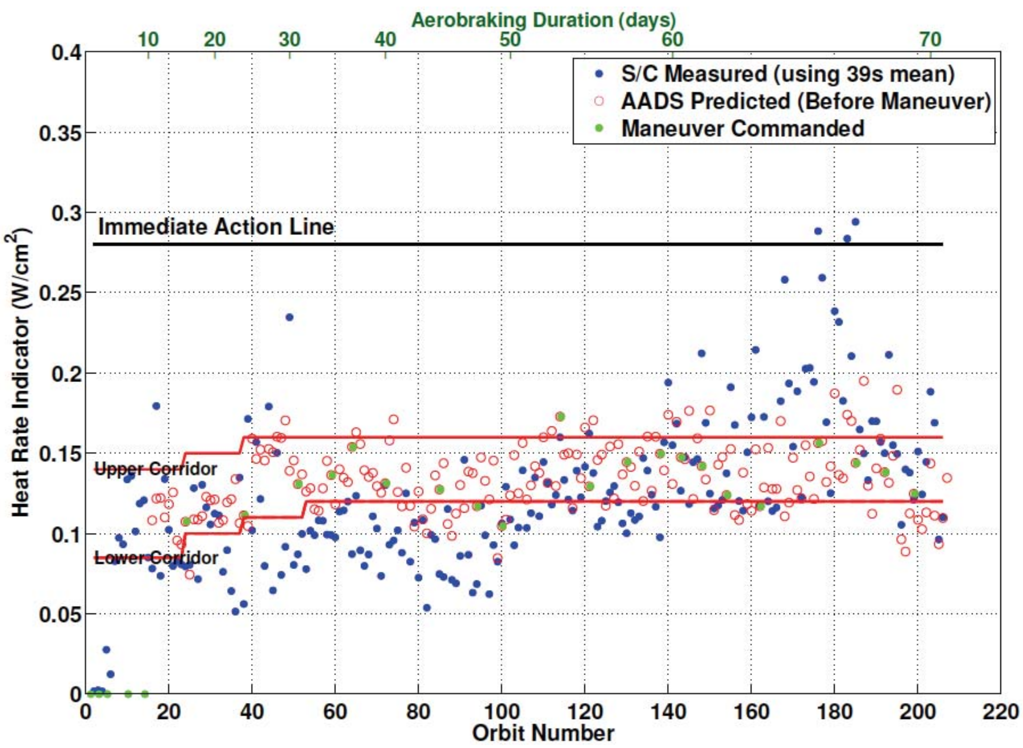

Figure 4. ORT1 Peak Heat Rate

Due to the similarity with ORT1 weekly runout analysis, shown in Figure 3, plots are not provided for ORT2 at the same point in the mission. As with ORT1, ORT2 analysis at the end of week 3 indicated that the lower corridor could be raised slightly to speed up the mission. However, starting after week 4, the upper corridor in the ORT2 simulation was lowered, effectively increasing the margin to the immediate action line, a move contrary to the decisions made in ORT1. Figure 5 shows the heat rate corridor plot for the ORT2 simulated mission. It is noted that the heat rate on orbit 39 and 48 exceeded the immediate action limit line. In ORT2, maneuvers were commanded based only on the predicted value of the heat rate at the next periapsis (red circles in the plot). If the heat rate was within the corridor, no maneuver was performed, and if outside the corridor, a maneuver was performed to place the next periapsis heat rate in the corridor. The maneuver logic and pop-up capability are fully operational. However, the maneuver logic based on the X-sigma high-density prediction (where X = 3) was not active in this simulation.

To determine improvements in performance (to eliminate the immediate action violations), two other versions of the ORT2 were spun off. ORT2a reran the mission with the 3-sigma high-prediction capability activated in the maneuver logic. The result was that no peak heat rates exceeded the immediate action line as intended. Weekly runout analysis did result in a slightly different mission corridor but yielded similar end of mission metrics. The second spin off, ORT 2b, considered the effect of holding a constant corridor for the entire mission. It continued to use the 3-sigma high-density predictions activated in ORT2a. The overall performance comparison between ORT1 and ORT2, ORT2a and ORT2b is shown in Table 1.

In ORT2a, which included the density 3-sigma uncertainties in the maneuver logic, 23 more maneuvers and 75 percent more ΔV (15.3 versus 8.75 m/s) were required compared to ORT2; however, no immediate action violations occurred. The corridor selections in ORT2a are similar to those used in the original ORT2, but the overall mission profile is different because of the additional biasing maneuvers. ORT2b required fewer maneuvers and used less ΔV than either ORT2 or ORT2a. All ORT2 missions required only weekly ephemeris updates.

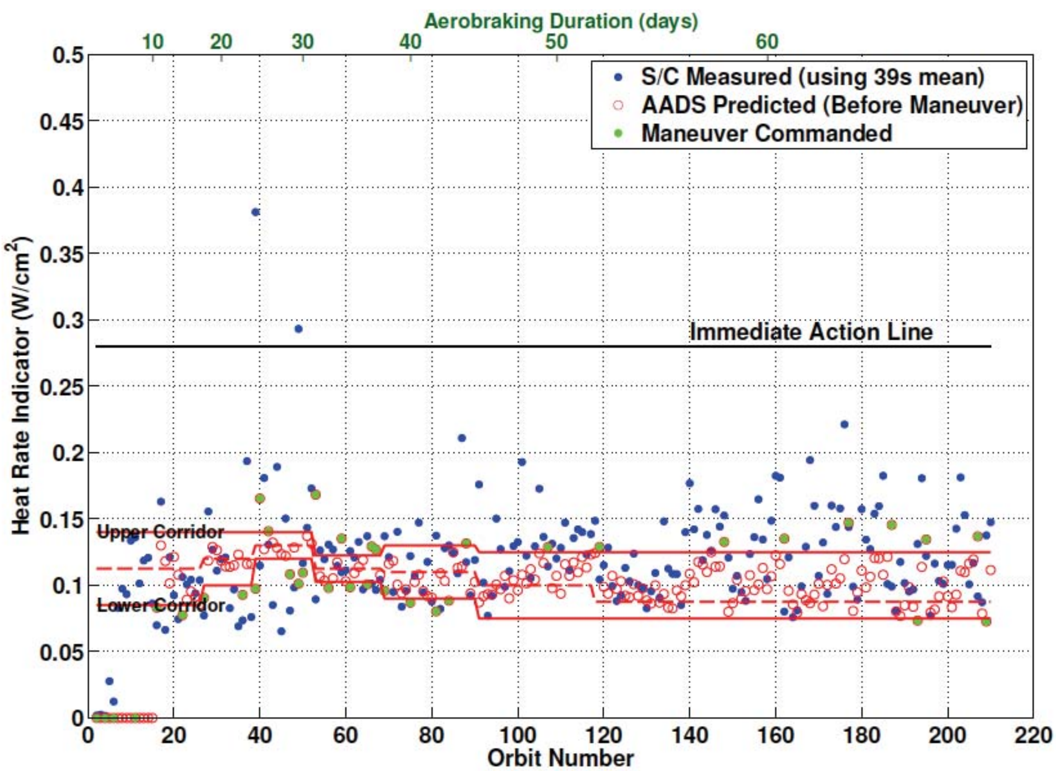

Figure 5. ORT2 Peak Heat Rates

Table 1. ORT Performance Metrics

Simulation	Number of Maneuvers	Maneuver ΔV (m/s)	Duration (days)	Final LTST (hours) Target: 10:46
ORT1 – Ground	22	4.8	70.4	10:39
ORT2 – AADS	33	8.75	69.9	10:43
ORT2a – AADS (3 sigma atmosphere)	56	15.3	67.3	10:49
ORT2b – AADS (con-stant corridor)	26	11.1	65.5	10.51

While ORT2b, with the constant corridor, did meet the final LTST requirements (i.e., target +/- 15 min), flying with a fixed corridor for the entire mission reduces the mission's flexibility to meet termination conditions. Variable corridors allow the spacecraft the ability to catch up in the event that the spacecraft falls behind the glide slope, or provide more margin if the spacecraft has gotten too far ahead. In general, the ORT2s (AADS series) provide both better control of the final LTST and more flexibility primarily because a maneuver can be performed every orbit. ORT2 was able to follow the reference glide slope more closely than the ground-based ORT1. Both ORTs start off a little behind the glide slope, but AADS has more opportunities to take corrective action to get back on track with significantly less analysis while enabling several other corridor and maneuver logic scenarios to be tested. The ORT2 series also verified that spacecraft updates once per week were sufficient to maintain mission requirements.

Monte Carlos Analysis

The objective of the AADS Monte Carlo analysis was to demonstrate that AADS is robust to environmental dispersions like maneuver errors, and aerodynamic and atmosphere uncertainties. The analysis was performed using 2000 3-DOF POST2 simulations and an MRO-like aerobraking mission. The simulation starts in a 32-hour orbit period after the walk-in phase is complete. The flight corridor target for this study is between 0.11 and 0.17 W/cm^2, resulting in a mission that is nominally 135 days. The AADS is not intended to be used for the walk-in or walk-out mission phases and they are not included in the Monte Carlo assessment. Table 2 provides a list and description of the parameters varied in the Monte Carlos simulation.

Figure 6 shows the results of the total heat rate dispersion for a 2000 case Monte Carlo analysis utilizing the MarsGRAM atmosphere. The black dots represent the nominal mission, and the black dashed lines represent the upper and lower corridor limits. The immediate action line, set to provide 100 percent heat rate margin above the upper corridor, has a heat rate of 0.34 W/cm^2; thus all of the perturbed cases have heat rates that remain below the immediate action line.

Figure 7 shows that the total variation in maneuver change in velocity (ΔV) for each of the 2000 missions was just under 3 m/s (3-sigma). The maximum difference from the nominal was approximately 8 m/s, a variation in total ΔV small enough to be accommodated in a typical aerobraking ΔV budget. The green dot in the histograms denotes the nominal AADS mission run-out value while the red dot denotes the mean of the 2000 Monte Carlo simulation. The final aerobraking duration for all 2000 cases fell within +/- 7 days of the nominal AADS mission run-out aerobraking duration. The statistics are shown in Figure 8. Note that the distribution in the output parameters was not expected to be Gaussian due to the fact that many of the processes involved in aerobraking (e.g., utilizing a corridor) are non-Gaussian in nature. Results of the AADS Monte Carlo analysis indicate that AADS is robust to typical spacecraft aerodynamic, spacecraft maneuver, and atmospheric variations experienced at Mars.

Table 2. AADS Monte Carlo Parameters

Parameter	Nominal	Perturbation	Distribution	Rationale
Maneuver Pointing Errors	Inertial α, β, γ	+/- 5 deg	Normal	Angular offset applied to apoapsis velocity vector.
Maneuver timing errors	0.0	+/- 100 s	Normal	Estimated apoapsis time +/- perturbation to see effect of starting burn early/late.
ΔV (m/s) / Maneuver Burn Duration	AADS calculated ΔV	+/- 5 %	Normal	Normally, engine is "on" until AADS calculated ΔV is achieved. To handle burn duration and ΔV parameters, allow applied ΔV= AADS calculated ΔV + perturbation.
Lift Coefficient Multiplier	1	+/- 10 %	Normal	Larger than expected multiplier provides sensitivity analysis.
Drag Coefficient Multiplier	1	+/- 10 %	Normal	Larger than expected multiplier provides sensitivity analysis.
Atmosphere Random Seed	1	1-29999	Integer Uniform	Value used by the atmosphere program to perturb the density and wind profiles. The range is that allowed by the Mars atmosphere program used in the simulation.
Dusttau (Mars only)	0.5	0.1:0.9	Uniform	This determines the dust loading and thus the density and wind profiles that the vehicle will experience. This range provides large variability, but would not include dust storms.

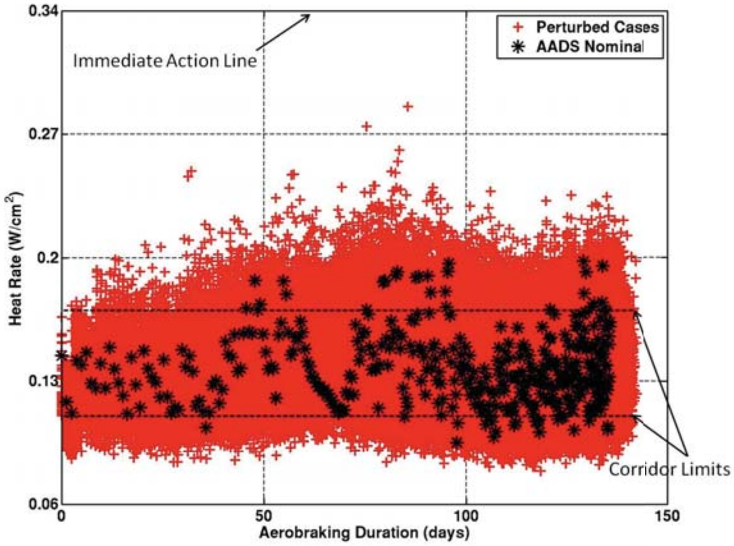

Figure 6. AADS Monte Carlo Heat Rate Distribution

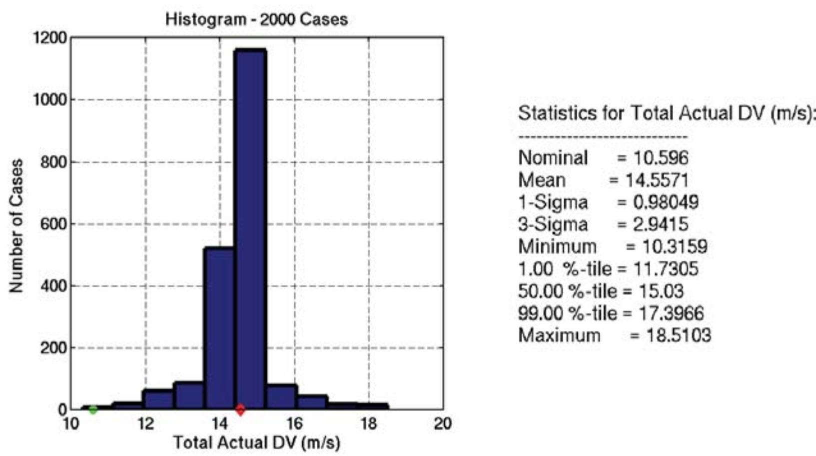

Figure 7. AADS Monte Carlo Results: Total ABM ΔV

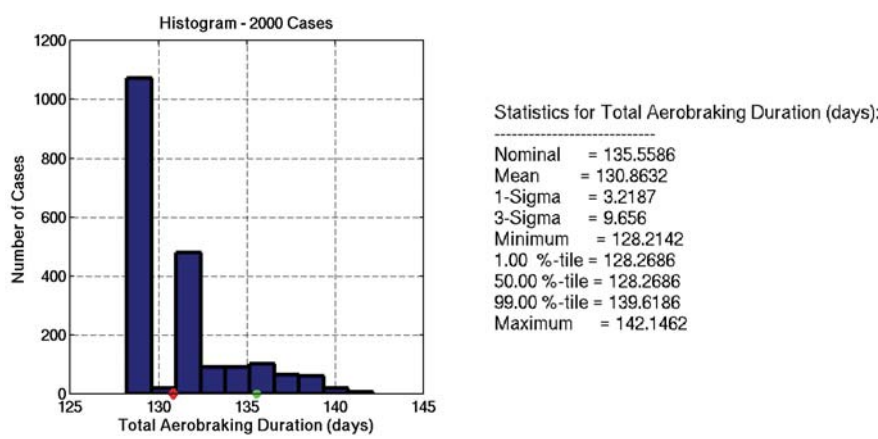

Figure 8. AADS Monte Carlos Results: Aerobraking Duration

Processor Load Analysis Benchmarking

Demonstrating that processor capabilities can support the software requirements is critical to ensuring AADS utility in a flight application. Although the code as written is not yet suitable for implementation on a flight processor, the software is mature enough to perform a preliminary assessment of the feasibility of its use in flight. It is time consuming and expensive to port AADS to a real flight processor, and this task is beyond the scope of the NESC Phase 2 study. However, simple benchmarking was done using the existing code running in a workstation that provides a rough order-of-magnitude assessment of the quantities of interest.

The approach taken for this study was to run AADS in the Autonomous Aerobraking High Fidelity Simulation (AAHFS) to time each portion of AADS that would correspond to a flight software task. AAHFS is a 6-DOF simulation based on the flight software and truth models previously developed for JHU/APL-designed MESSENGER spacecraft simulation and used to evaluate AADS in Phase 1. These times, obtained by the workstation testing using AAHFS, were scaled based on the ratio of the clock speed of the workstation-to-flight processors to estimate the CPU time each task would require on a flight processor. Comparing this scaled time with the expected CPU time available for each task provides an estimate of the processor loading for each task. The testing described in this section used AADS running on a workstation PC with a clock speed of 2670-MHz. As an example, the MESSENGER mission flight processors had a clock speed of 25-MHz, so the ground test times are scaled up by 107 times to estimate the equivalent CPU time required for a notional flight processor. The flight processor clock speed used in this analysis is considered conservative, as MESSENGER processors were purchased nearly a decade ago, and current-day missions are routinely running faster than 25-MHz.

The benchmarking must carefully collect timing information on software functions that are only a part of the expected three tasks that comprise AADS. The tasks are: a high-rate task (notionally 10-Hz) for acceleration/attitude data processing, integration, and buffering; a low-rate task (notionally 0.01-Hz) for evaluation of the gravity models, integration of the spacecraft orbit, and preparation of the least-squares estimation matrices; and a batch process (executed once per orbit) that performs the orbit prediction and the subsequent AE and ME tasks.

The highest-rate process found in AADS is the input data processing and high-rate integration function in the SP. There are few calculations associated with this software task, as it merely performs a trapezoidal integration of the acceleration data and buffers the resultant ΔV for use in the low-rate integrator. Although this task runs at a high rate, it is by far the least concerning of the three processing tasks. It is well known that substantially more complicated algorithms are routinely running in spacecraft flight software at markedly higher rates (greater than 100-Hz), so no further effort was devoted to benchmarking this task.

The low-rate integration task is the most challenging from a processor loading point of view. Benchmarking this task was performed using a simulation at Mars, with a 21-degree and order gravity model, which is currently assumed to be the highest-fidelity gravity model considered for AADS testing (higher ordered gravity models are available, but do not offer a substantial increase in accuracy over the 21-degree and order field for purposes of aerobraking analysis). The results of this benchmarking throughout an entire Mars aerobraking mission are shown in Figure 9a. Each drag pass sees a spike in the execution time for the low-rate integrator due to the shortened time steps. As the mission progresses, these spikes become more frequent as the orbital period drops and more prolonged as the drag pass duration increases, but the peak value is not markedly affected. In the worst case, a flight processor would require approximately 1.6 s to complete the low-rate integration. This reflects a processor loading (for this task only) of less than 2 percent, as the processing frame’s period is 100 s.

The AADS batch process that performs the orbital propagation, together with the atmosphere and maneuver calculations, is by far the most computationally expensive task to complete. Additionally, it runs at a low software priority. Despite the challenges that this task presents, the batch process is not expected to pose an issue for processor loading as there is a substantial amount of time to complete this task. It is assumed that the task will be triggered shortly after the drag pass exit, and must complete with sufficient time to configure the spacecraft for any corridor control maneuver, which means that even in the shortest-period orbits, this task will have more than 20 minutes to complete. Timing tests were done to characterize the expected processor time required to complete this task, and the results are shown in Figure 9b. For this task, the orbital propagation takes longer in the long period orbits, so the worst-case is early in the mission (when the task has much longer to complete). In all cases, the batch process would require much less than 1 percent of the available CPU, presenting no issue for the timely completion of this task. The initial benchmarking assessment indicates that typical processor capabilities can support the software requirements of AADS.

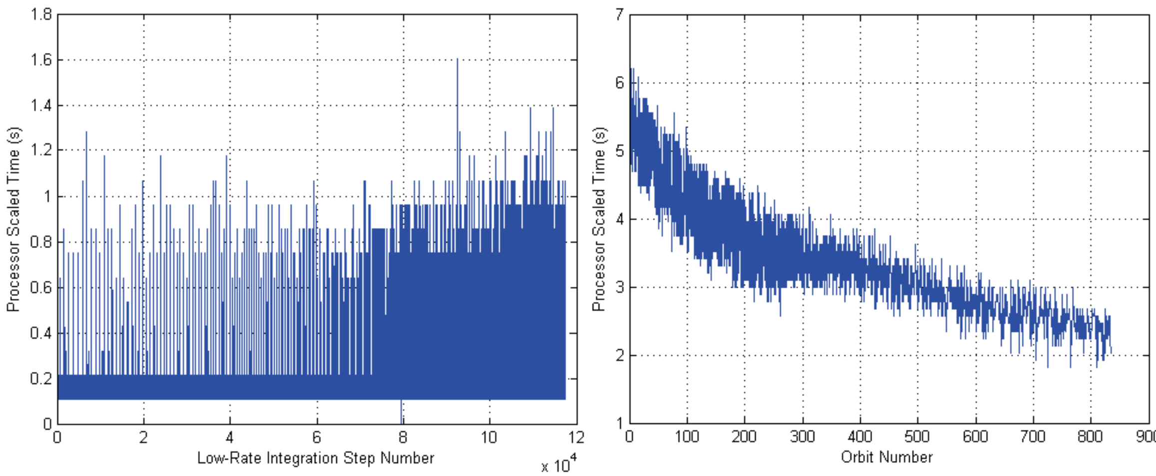

Figure 9a. (Left) AADS Low-Rate Integrator State Determination Scaled Time for Mars Case using 21x21 Central Body Harmonic Gravity Model. 9b. (right) AADS Low-Rate Integrator Orbit Propagator Scaled Time for Mars Case using 21x21 Central Body Harmonic Gravity Model

CONCLUSION

The AADS Phase 2 analysis included a large number of model and simulation upgrades in addition to the multitude of trade, sensitivity and ORT analyses. The architecture changes to the SP improved the accuracy, reduced the processor loading and reduced memory requirements of AADS by a factor of nearly 100 over the Phase 1 model. The modifications to the AE, to allow for user specified multipliers of the 1 sigma density perturbations to affect the onboard maneuver decision, made the mission more robust and less susceptible to violations of the immediate action line. Results of the AADS Monte Carlo analysis indicate that AADS is robust to typical spacecraft aerodynamic, spacecraft maneuver, and atmospheric variations experienced at Mars. Benchmarking demonstrated that there are no indications that the AADS code is computationally prohibitive to a spacecraft computer. The ORT study, comparing traditional ground based methods to AADS, demonstrated that while both approaches achieved the final mission requirements, the AADS Team required far fewer resources (namely no daily meetings and analysis), and only needed weekly updates from the Atmosphere Team for the mission runouts. Though the quantification of actual mission cost savings from reduced workload and DSN coverage is not in the scope of the study, the indications are that it would be substantial.

Additionally, the purpose of evaluating various AADS strategies using ORT scenarios was to understand the features and capabilities of the software. The cases examined for the ORT are just the beginning of our understanding of the flexibility and capability of AADS. For example, by implementing 3-sigma density perturbation bias in the maneuver logic, the number of maneuvers increased (ORT2a); by relaxing the corridor and keeping it constant throughout the mission (ORT2b), the mission uses fewer maneuvers and resulted in a shorter aerobraking phase. Different options could also be considered such as only using the 3-sigma high-density predictions during the periods with the most atmosphere variability. Despite the variety of stress cases and aggressive aerobraking mission ORTs, no modifications to the AADS software were required, demonstrating robustness to an array of conditions. The ORTs also demonstrated how aerobraking mission design and operations paradigms shift when the spacecraft is allowed to calculate and execute maneuvers each orbit. AADS is the logical extension of the current periapsis timing estimator demonstrated on MRO.

This paper attempts to summarize key model updates and highlights only a few of the many studies performed during Phase 2 that considerably advanced the state-of-the-art of the technology to a Technology Readiness Level (TRL) of 5. Future plans include studying the implications of operating AADS in a real-time environment, and then installing the software onto a spacecraft that would use AADS during flight in shadow-mode, where onboard-determined commands would be validated against aerobraking decisions made by the ground staff, would further the TRL to 8.

REFERENCES

[1] D. T. Lyons, "Aerobraking at Venus and Mars: A Comparison of Magellan and Mars Global Surveyor Aerobraking Phases." AAS 99-358.

[2] D. T. Lyons, J. G. Beerer, P. Esposito, M. D. Johnston, "Mars Global Surveyor: Aerobraking Mission Overview." *Journal of Spacecraft and Rockets*. Vol. 36, No. 3, May-June 1999.

[3] J. C. Smith, Jr., J. L. Bell, " 2001 Mars Odyssey Aerobraking." *Journal of Spacecraft and Rockets*. Vol 42, No. 3, May-June 2005.

[4] S. M. Long, et al. "Mars Reconnaissance Orbiter Aerobraking Navigation Operation." AIAA 2008-3349, 2008.

[5] J. D. Chapel, et al. " Attitude Control Performance for MRO Aerobraking and the Initial Science Phase." February 2007. AAS 07-097.

[6] D. G. Murri, et al. "Development of Autonomous Aerobraking (Phase 1)," December 15, 2011. NESC-RP-09-00605/NASA-TM-2012-217328

[7] R. M. Maddock, A. Bowes, R. W. Powell, J. L. Prince, and A. D. Cianciolo, "Autonomous Aerobraking Development Software: Phase One Performance Analysis at Mars, Venus, and Titan," AIAA 2012-5074.

[8] D. Carrelli, D. O'Shaughnessy, T. Strikwerda, J. Kaidy, J. L. Prince, R. W. Powell, "Autonomous Aerobraking for Low-Cost Interplanetary Missions," Acta Astronautica, Available online 22 November 2012, ISSN 0094-5765, http://dx.doi.org/10.1016/j.actaastro.2012.10.041.

[9] D. G. Murri, et al. "Development of Autonomous Aerobraking (Phase 2)," NESC-RP-09-00605. August 2013. Publication Pending.

[10] C. G. Justus, D. L. Johnson, "Mars Global Reference Atmosphere Model, 2001 Version (Mars-GRAM 2001): Users Guide", *NASA/TM-2001-210961. 2001.*