

## AMMO: AN AUTOMATED MULTIPLE MANEUVER OPTIMIZATION SYSTEM

Christopher L. Potts<sup>†</sup>, John R. Michel<sup>‡</sup>, and Behzad Raofi<sup>\*†</sup>

An automated maneuver design capability can provide significant benefits in cost, risk reduction, and science return for interplanetary spacecraft missions. Emphasis must be placed on a general and robust approach to accommodate the diversity of complex missions, both present and future. Maneuver optimization provides a highly desired sophisticated  $\Delta v$  solution, but also increases complexity and introduces significant automation challenges. An Automated Multiple Maneuver Optimization (AMMO) system is presented in this context. Additionally, the prototype system has proven to be extremely successful with Stardust operations support by reducing the design time requirements for commanded  $\Delta v$  by an order of magnitude and providing continuous maneuver design support with limited resources.

### INTRODUCTION

Interplanetary spacecraft missions have evolved from relatively quick hyperbolic reconnaissance to extended orbital presence, in situ measurements, and sample return capability. To achieve the enhanced mission objectives, mission plans require an increasing number of trajectory correction maneuvers and frequent gravity assist encounters during orbital tours. Consequently, the flight path control challenges encountered with current and future interplanetary spacecraft missions continue to advance in complexity and risk.

While the missions are more challenging than ever, the desire always exists to improve efficiency and thereby reduce operation costs. Automating frequently performed tasks in a robust manner is a proficient way to achieve these goals. Executed properly, automation can help achieve many of the time-critical mission requirements while enabling a reduction in both mission risk and operation costs.

---

<sup>†</sup> Member of Engineering Staff, Navigation and Mission Design Section

<sup>‡</sup> Professor of Mathematics, Marietta College, Marietta, Ohio, and JPL Consultant

\* Member AAS

Lowering the cost of interplanetary spacecraft exploration requires the development of capable automation tools. This paper outlines an automated multiple maneuver optimization system designed to routinely determine the optimal  $\Delta v$  requirements from updated orbit determination solutions while satisfying the mission specific constraints. This system has demonstrated its effectiveness through prototype use on operational missions and analysis of upcoming missions.

## OPTIMIZATION

Maneuver optimization has become a prevalent analysis tool to significantly reduce mission propellant requirements. By re-optimizing the trajectory during operations, control variables are adjusted to minimize the cost of correcting spacecraft state variations that occur within the realm of orbit determination uncertainties. For satellite tours or Earth return missions, a global optimization ensures that near-term maneuvers are executed in accordance with the long-term mission goals. The optimization process maintains control of the parameters that define the trajectory requirements of the mission. In the maneuver design process, the optimization step represents a sophisticated general solution to  $\Delta v$  determination. The result is savings in propellant, improved margins in consumable resources, and reduced mission cost.

A significant detriment to using optimization is that the algorithms often require close monitoring and user interaction to achieve an acceptable solution. Frequently, existing algorithms suffer convergence problems in the search for the absolute optimal solution or are ill suited for the specific maneuver optimization application at hand. New algorithms are making progress in terms of reliability and capability, but this improvement comes with an increase in complexity for the general user. Optimization algorithms are generally not conceived for use in an automated approach.

The maneuver optimization problem associated with spacecraft flight operations contains inherent advantages to utilize in the quest for an automated capability. The most significant advantage is the ability to start with a very good initial estimate of the optimal solution. Prior to flight operations, the mission design process defines the optimal reference trajectory. This nominal trajectory provides the initial estimates of the key control variables that define the reference mission. The task is to re-optimize the trajectory after incorporating new estimates of the initial spacecraft state and model variations. The spacecraft state and model updates result from the processing of navigation tracking data acquired in flight. With each optimization cycle, a new reference trajectory is created which provides the initial control state estimates for use in the subsequent optimization. In this manner, each re-optimization process begins with a very good initial estimate of the trajectory control states, which is critical for a reliable optimization solution. In fact, an automated optimization capability provides frequent trajectory updates that in turn promote the reliability of an automated system in somewhat of a self-enabling cycle.

A second flight operation advantage to utilize in an automated optimization algorithm is the recognition of system uncertainties and appropriate control thresholds. Frequently, existing algorithms suffer convergence problems in the search for the absolute optimal solution. In flight operations the emphasis is directed toward an acceptable solution that satisfies the critical mission constraints. An acceptable solution is one that is optimized to the level of the system control capability and not necessarily the absolute numerical optimum. The acknowledgement of system control thresholds is utilized to simplify the optimization problem by removing control variables from the problem definition. As a result, an acceptable solution can be obtained before the optimization algorithm encounters a potential numerical instability.

## ARCHITECTURE OF OPTIMIZATION PROBLEM

The structure used to define the optimization problem is based on the design of D. Byrnes and L. Bright<sup>1</sup>. This approach has been implemented in the CATO (Computer Algorithm for Trajectory Optimization) program, and has proven to be extremely flexible for a wide range of mission applications. An additional advantage is the pre-existence of a software framework for defining and manipulating this type of trajectory structure. For completeness, a brief description of the defining architecture is presented. Further structure details are found in Ref. 1.

The first step in the numerical definition of the maneuver optimization problem is to break up an existing trajectory into discrete segments. The adopted terminology for the boundary between two consecutive legs is a trajectory breakpoint. A unique control point exists for each trajectory leg ( $C_1, C_2, C_3, \dots$ ). The control point consists of 8 variables to define the spacecraft state: the epoch, the spacecraft mass, and 6 parameters to describe the spacecraft position and velocity. The selection of the parameters describing the spacecraft state is extremely flexible, which ultimately correlates with the capability to define specific mission constraints for each trajectory leg. Figure 1 is a schematic of the trajectory structure inherited from Ref. 1.

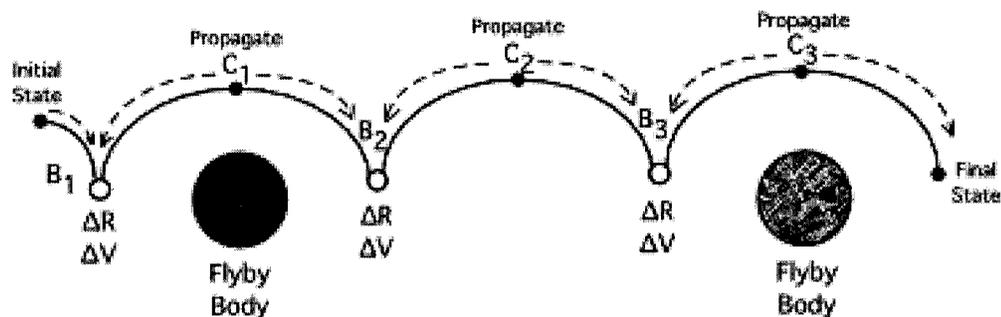


Figure 1: Schematic Representation of an AMMO Trajectory

The number of discrete trajectory legs is somewhat flexible, and gets adjusted by the AMMO system to satisfy the maneuver design requirements as the mission progresses. The user defines the key initial control point locations based upon mission constraints, control parameters of interest, and encounter bodies. An input file identifies the epochs for Trajectory Correction Maneuvers (TCMs) in the mission, which defines the trajectory breakpoint locations ( $B_1, B_2, B_3, \dots$ ). As necessary, the AMMO system automatically creates control points to complete the trajectory leg structure. These control points are placed at the midpoint of any two consecutive breakpoints, with the state parameters determined from the nominal reference trajectory.

## TRAJECTORY INTEGRATION

The initial conditions are obtained from the orbit determination process along with modifications to the spacecraft models and possible updates to the physical properties affecting the spacecraft trajectory (gravitational constants, ephemeris, harmonics, pole orientation, etc.). All the initial control point state information is extracted from the current best estimate of the nominal trajectory, which is readily available during spacecraft operations. Beginning at each control point, the spacecraft's equations of motion are integrated both forward and backward in time to the surrounding breakpoints. The initial state is integrated to the epoch of the first breakpoint,  $B_1$ . Trajectory integration also solves the variational equations to generate state transition partials, which are used later to solve the linear optimization problem.

The piece-wise integration of the trajectory results in two state solutions at each breakpoint epoch:  $X_i^-$  from the previous control point, and  $X_i^+$  from the following control point. Discontinuities in position,  $\Delta R_i = R_i^+ - R_i^-$ , and velocity,  $\Delta V_i = V_i^+ - V_i^-$ , generally exist at each breakpoint. A constraint is applied in the optimization process to eliminate the position discontinuities (within numerical limitations) and create a trajectory continuous in position. The remaining velocity discontinuities represent the velocity changes the spacecraft must execute to complete the mission trajectory.

## LINEAR MINIMIZATION PROBLEM

After numerically integrating the discrete trajectory segments, the original control point states, state transition partials, and breakpoint discontinuities provide a linear model of the original trajectory. Solving the linearized minimization problem provides updates  $\Delta Z$  to the independent variables  $Z$ , which are the trajectory control point parameters.

The minimization problem is of the form:

$$\text{Minimize} \quad \sum_{i=1}^s | A_i x - b_i |$$

subject to linear equality constraints

$$\mathbf{G} \mathbf{x} = \mathbf{h}$$

and simple bound constraints

$$\mathbf{L} \leq \mathbf{x} \leq \mathbf{U},$$

where  $\mathbf{x}$  is an  $n$ -dimensional vector of independent variables, each  $\mathbf{A}_i$  is a 3 by  $n$  matrix, each  $\mathbf{b}_i$  is a 3-dimensional vector,  $\mathbf{G}$  is an  $m$  by  $n$  matrix,  $\mathbf{h}$  is a  $m$ -vector, and  $\mathbf{L}$  and  $\mathbf{U}$  are  $n$ -vectors of lower and upper bounds on  $\mathbf{x}$ , respectively.

In this case,  $\mathbf{x} = \Delta \mathbf{Z}_{\text{free}}$ , the deltas in the free variables of the problem, each  $\mathbf{A}_i = \frac{\partial \Delta \mathbf{V}_i}{\partial \mathbf{Z}_{\text{free}}}$  and each  $\mathbf{b}_i = -\Delta \mathbf{V}_i(\mathbf{Z}_{\text{free}}^c)$ , where  $\mathbf{Z}_{\text{free}}^c$  represents the current values of the free variables. Thus the quantity  $\mathbf{A}_i \mathbf{x} - \mathbf{b}_i$  is just the predicted value for the new velocity difference  $\Delta \mathbf{V}_i$  at the  $i$ th breakpoint based on changing the free variables by  $\mathbf{x} = \Delta \mathbf{Z}_{\text{free}}$ . This is a linear approximation for the true  $\Delta \mathbf{V}_i$ . The number of norms  $s$  summed in the cost function is equal to the number of breakpoints.

The equality constraints  $\mathbf{G} \mathbf{x} = \mathbf{h}$  always contain all the linearized  $\Delta \mathbf{R}_i = 0$  constraints of the form

$$\frac{\partial \Delta \mathbf{R}_i}{\partial \mathbf{Z}_{\text{free}}} \Delta \mathbf{Z}_{\text{free}} = -\Delta \mathbf{R}_i(\mathbf{Z}_{\text{free}}^c).$$

Velocity constraints can also be applied in the form of an equality constraint for breakpoints (or TCMs) where direction and/or magnitude control is desired. Additional mission constraints can also be added to the  $\mathbf{G} \mathbf{x} = \mathbf{h}$  set.

The simple bound constraints  $\mathbf{L} \leq \mathbf{x} \leq \mathbf{U}$  are just

$$l - \mathbf{Z}_{\text{free}}^c \leq \Delta \mathbf{Z}_{\text{free}} \leq u - \mathbf{Z}_{\text{free}}^c$$

where  $l$  and  $u$  are upper and lower bounds on the values of the free variables  $\mathbf{Z}_{\text{free}}$ .

The cost function to minimize is the sum of the  $\Delta \mathbf{V}$  magnitudes at the trajectory breakpoints. Because the objective function is the sum of a set of Euclidean norms, this problem is sometimes called the sum-of-norms (SON) problem. The objective function is non-linear and not differentiable at any  $\mathbf{x}$  for which one of the norms is zero. Because of the non-smooth nature of the cost function, an indirect approach is used to minimize it. The approach is described below.

In AMMO, the SON problem is addressed by iteratively solving a series of closely related weighted least-squares problems, each with the same constraints as the SON problem above. Each least-squares problem is defined by using the reciprocals of the “residuals” remaining from the previous solution as weights. That is, suppose  $\mathbf{x}^c$  is the current solution and let  $w_i = 1/|\mathbf{A}_i \mathbf{x}^c - \mathbf{b}_i|$  for each norm  $i$ . The next problem to solve has the original set of constraints, but a new least-squares objective function:

$$\sum_{i=1}^s w_i |\mathbf{A}_i \mathbf{x} - \mathbf{b}_i|^2.$$

Initially, all the weights  $w_i$  are set to 1. If a  $\Delta V_i$  approaches zero causing the weight to get larger than a prescribed large number ( $10^6$ ), then  $w_i$  is fixed at this large number and the corresponding residual  $\mathbf{A}_i \mathbf{x} - \mathbf{b}_i$  is assumed to be zero. It can be shown that the solutions to the sequence of re-weighted least-squares problems converge to the solution of the SON problem.

A very efficient computer algorithm called DBLSE solves each of the least-squares problems. C. Lawson and R. Hanson developed DBLSE, which is a modified version of their BVLS algorithm<sup>2</sup>. The same algorithm has been used successfully for many years in the linear statistical analysis of maneuver optimization strategies.<sup>3</sup>

Although the original trajectory is highly non-linear, convergence on an optimal solution is possible by applying successive linearized solutions to the original trajectory. The process repeats until the numerically integrated trajectory segments produce a near continuous trajectory in position, and the velocity discontinuities at the trajectory breakpoints are in close agreement with the linear minimization solution. Convergence is achieved when the proposed control point deltas map to changes smaller than 1 mm and 0.01 mm/s in each of the three position and velocity components, respectively, at the previous breakpoint.

The use of a non-linear or a genetic optimization algorithm was considered to replace the linear formulation. The successive linear approximations and least-squares optimization approach was selected based on its relative simplicity and reliability. In a majority of cases, the linear formulation is a very efficient method for solving the spacecraft operations problem of re-optimizing the trajectory given an initial condition perturbation to a nominal trajectory.

## AMMO CONTROL ALGORITHM

Historically, a user monitors each linearized solution and controls the changes applied to the original trajectory. When the solution to the linear approximation of the trajectory is applied to the original trajectory and numerically integrated, the discontinuities between trajectory segments can increase dramatically rather than decrease, as the linear solution would predict. Unfortunately, this is an all too common occurrence. In this case the user usually ends up rejecting the proposed changes from the linear solution and experiments with the control variables in an attempt to get back on a path toward

convergence. The process can be time consuming and frustrating, as it often involves a lot of trial and error.

The AMMO algorithm provides the driver and controller of the optimization process to replace the interaction of an experienced user. The key AMMO objective is to automatically converge on an optimal trajectory solution even when the linear model is a poor representation of the original trajectory. The control algorithm has been modified and improved through the analysis of numerous spacecraft trajectory optimization problems taken from real mission scenarios. The algorithm is designed to monitor each linear solution and provide a focused test of its suitability. For a majority of applications, the linear solution is applied directly and provides rapid convergence. When required, the AMMO algorithm provides feedback control in the successive linear solutions to the original highly nonlinear problem. In this manner, convergence is robustly achieved without requiring any user interaction.

The AMMO feedback and control process begins upon completion of the linear optimization sub-problem. Rather than simply applying the linear corrections to the original trajectory and integrating, a more focused attempt is made to determine the effectiveness of the linear solution. This is highly desirable, since numerically integrating the entire trajectory can be a time consuming step and should not be undertaken without an expectation of improvement. The AMMO algorithm focuses on reducing the position discontinuity at the breakpoint with the largest error. Frequently, the major change in the operations problem is the estimate of the initial spacecraft state, which results in the largest position discontinuity occurring at the first breakpoint. AMMO takes advantage of the trajectory discretization and applies the proposed linear corrections to the control points surrounding the maximum position discontinuity. Consequently, only the two trajectory segments affecting the identified breakpoint need to be integrated. In the case where the first breakpoint is the largest, only one trajectory segment gets integrated since the propagation from the initial conditions to the first breakpoint is a fixed segment.

The AMMO control algorithm also uses the identified breakpoint and integration of the surrounding trajectory segments to determine whether a scale factor should be applied to the linear solution. By applying discrete scale factors (1.0, 0.8, 0.6, 0.4, ...) to the proposed linear solution, the resulting position discontinuity is examined to select the most effective scale value. The algorithm triggers on the first inflection point to avoid unnecessary trajectory integration steps. For example, if a scale factor of 1.0 improves the position discontinuity while the discontinuity increases for a scale factor of 0.8, then no further scale factors are examined.

If the chosen scale factor becomes too small ( $\leq 0.4$ ), the current linear solution provides diminishing returns in the convergence process. The small scale factor is an indication that the optimized solution has exceeded the region of linearity of the original trajectory. Rather than integrate the entire trajectory for a potentially small improvement, a more efficient strategy is to adjust the bound

constraints and compute a new linear optimization solution. Because of the globalization of the optimization problem, this is accomplished by scaling the largest proposed control point change magnitude by a factor of 0.5 and applying this value as a new upper or lower bound (depending on the sign) on the same control point. Notice that care is taken to avoid over-constraining the linear optimization problem by keeping the new bound larger in size than the scaled solution dictates. However, the new bound represents a real constraint on the existing linear solution and will require a reduction in the proposed control point changes in order to satisfy the new bound. More aggressive bound strategies have been examined, but the described approach provides a reliable progression toward convergence.

With each complete trajectory integration, the AMMO algorithm not only locates the largest position discontinuity but also maintains a history of the identified breakpoints. If the linear solution and scale factor for the largest discontinuity is satisfactory, the algorithm verifies that the proposed solution also improves the position errors at the breakpoints in the history list. If not, then a control point bound is computed as previously described and a new linear optimization solution is obtained. This check protects against cases where the linear optimization develops an alternating pattern of improvement at one breakpoint at the expense of another.

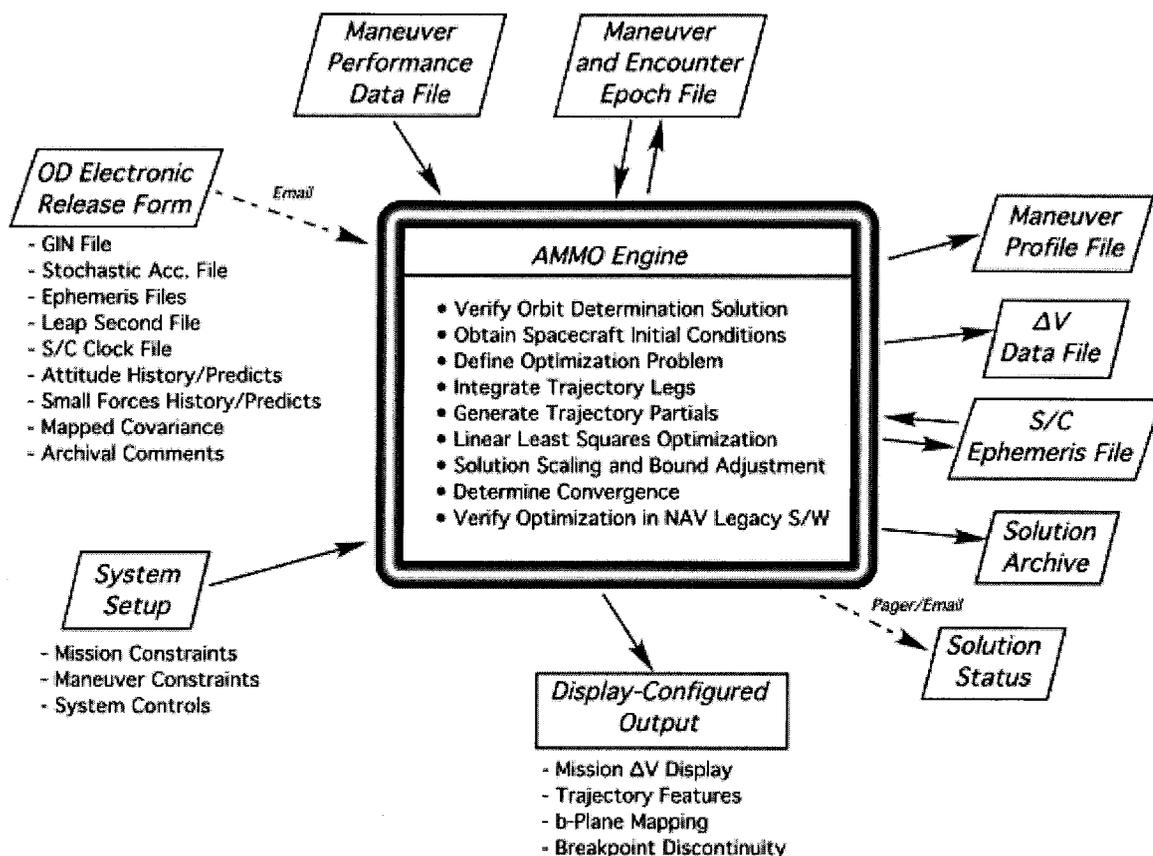
The AMMO control algorithm includes additional control logic for instances where the linear approach is a problematic representation of the original trajectory. If the linear solution and potential scale factors fail to improve the position discontinuity at all, the largest control point change is reduced by an order of magnitude and applied as a bound on that control variable for future iterations. The algorithm also checks the proposed changes in time of closest approach for the controlled encounters. If the proposed change is less than 0.1 seconds, the optimization problem is adjusted to fix the encounter time at the current value. This is an example of recognizing the system uncertainties and applying control thresholds to simplify the optimization problem when necessary to promote convergence.

## **AMMO SYSTEM**

After developing and improving a robust and automated optimization algorithm for use in spacecraft operations, the next step was to integrate the capability with existing navigation software to demonstrate a prototype system for operations. The integrated system could then be applied and modified to accommodate an actual flight operations environment.

The focused nature of flight operations combined with a well-defined mission plan provide the framework to apply a software engine that will routinely determine the optimal  $\Delta v$  requirements from updated orbit determination solutions while satisfying the mission specific constraints. The software design accommodates mission inputs as opposed to individual maneuvers, and adjusts the optimization problem definition to fit the epoch of initial conditions as the

mission progresses. Most of the model inputs to the software engine occur via standard file inputs. Hence when a maneuver execution date changes, the date is manually changed in an existing maneuver epoch file and all subsequent AMMO solutions will utilize the new date. The AMMO engine constantly checks for notification of a new orbit determination (OD) solution. Currently, a new solution is delivered through a standard electronic form that includes all the navigation files that define the latest estimate of the spacecraft trajectory. The release form is sent by email to the AMMO engine, which identifies the new delivery and immediately begins the re-optimization process. Figure 2 shows a schematic of the AMMO system.



**Figure 2: AMMO System Schematic**

The prototype AMMO software has been successfully integrated with the navigation software that has been approved for flight missions. The optimization results are based upon relatively high precision system models, albeit less sophisticated in some respects from the modeling normally used in navigation operations. Consequently, the AMMO system currently provides the optimized targets as inputs to the navigation legacy software to verify the accuracy of the solution. The optimized  $\Delta V$  results from AMMO show very good agreement with the operational software, with differences in  $\Delta V$  magnitude normally on the order of cm/s. This agreement in  $\Delta V$  magnitude is limited by the ability to

duplicate the high precision operation models in the AMMO system. If the  $\Delta V$  magnitude differences increase much beyond this normal comparison, it's usually an indication of a model discrepancy in the AMMO system.

The re-optimized trajectory results are usually available 15 minutes after the OD file is released. Numerous standard files are produced with each trajectory update. These include the maneuver profile file (commanded  $\Delta v$  for the next maneuver in the mission), a  $\Delta v$  data file of complementary information, and an updated spacecraft ephemeris file spanning the end of the mission. An archive directory is created for each OD solution AMMO receives. If desired, a user can recreate and investigate any optimization result produced by AMMO. The AMMO system also creates status displays of the mission  $\Delta v$ , trajectory control events, a trajectory target plot, and graphical results to quickly observe the efficiency of the optimization process. Since the AMMO system is designed to operate continuously, the engine can send out a short email or alphanumeric page notification of the new  $\Delta v$  solution and the number of iterations required in the optimization process.

## **MISSION APPLICATIONS**

### **Genesis**

The AMMO system was used to perform some preliminary tests using Genesis injection error samples. The trajectory was automatically re-optimized to a nominal state after Lissajous Orbit Insertion (LOI). Because of the unique stability properties of the Genesis trajectory<sup>4</sup>, the optimization would frequently converge on a local rather than global minimum. The local minimum case would generally have noticeably higher  $\Delta V$  costs than the reference trajectory. Normally an updated reference trajectory is a vital aid to trajectory convergence. In this special case, starting from the latest reference trajectory promoted convergence on a sub-optimal local solution. The Genesis maneuver optimization problem is unique in that it's best to always use the original pre-launch trajectory for the initial control point estimates. This promotes a local solution closest to the original reference trajectory. The AMMO system converged properly on all Genesis injection samples after this minor modification was implemented.

### **Cassini**

The Cassini tour at Saturn is the type of intense operation environment that the AMMO prototype was designed to support. A robust and automated maneuver design process is essential to accommodate approximately 44 targeted flybys of Titan (orbits as short as 16 days), 7 targeted Icy satellite encounters, and in general 3 maneuvers per encounter.<sup>5</sup> In this scenario, maneuver optimization and a rapid TCM design capability translate directly into significant mission propellant savings. Critical operations support can be required at any time of the day or night, so a manual TCM design process would take a heavy toll on operations personnel over the four-year tour.

The AMMO prototype system has been successfully applied to two separate segments of the Cassini tour. Each segment included approximately 5 Titan encounters, and one segment also had an Icy satellite encounter. The AMMO algorithm converged routinely under both fixed and flexible target control assumptions for each case. Additional analysis is planned to determine the best way to meet the Cassini operation requirements.

### **Europa Orbiter**

A preliminary Europa Orbiter trajectory (Tour 9906-65E) provides the ultimate test to date for the AMMO optimization feedback and control algorithm. The highly non-linear trajectory includes approximately 60 trajectory correction maneuvers over a span of 4.5 years from launch to Europa orbit insertion.<sup>6</sup> The dynamic trajectory contains 16 gravity assist flybys with altitudes as low as 100 km, 2 orbit insertion burns (Jupiter and Europa), 2 non-targeted flybys, and a highly sensitive 3<sup>rd</sup> body capture at Europa.<sup>7</sup> Not surprisingly, applying the linear optimization algorithm to the entire trajectory proved to be numerically unstable. The AMMO algorithm is still able to control convergence in this extreme case. An experienced analyst would find this type of optimization problem extremely challenging and potentially classify the problem as “not worth the effort” required to obtain a globally converged solution. Even with the linear partials continually promoting a divergent solution, the AMMO feedback and control algorithm monitors and adjusts the linear bounds to automatically achieve convergence.

### **Stardust**

From an optimization viewpoint, the Stardust mission trajectory<sup>8</sup> is better behaved and a more stable problem. The AMMO algorithm monitors each iteration, but rarely needs to intervene by applying scale factors or adjusting bounds to the linear solution. The Stardust mission does provide an invaluable opportunity to test the entire AMMO system in a continuous spacecraft operations environment. For over a year now, the AMMO system has been supporting the Stardust mission. During this time, the AMMO system has received approximately 58 orbit determination solutions, and has generated a re-optimized Stardust trajectory usually within 15 minutes of the OD electronic delivery. The AMMO system has provided maneuver optimization support for the last 3 trajectory correction maneuvers, including a rapid design schedule for the final Earth gravity assist maneuver. The legacy Navigation software generates the final TCM products based upon the AMMO optimization results. The TCM products receive a complete manual review before delivery, which is currently the most time consuming aspect of the TCM design procedure. The Stardust project has a devoted AMMO X-terminal located in a secure Navigation Computing Facility for operations. Figure 3 shows the AMMO display results for a preliminary Stardust TCM design.

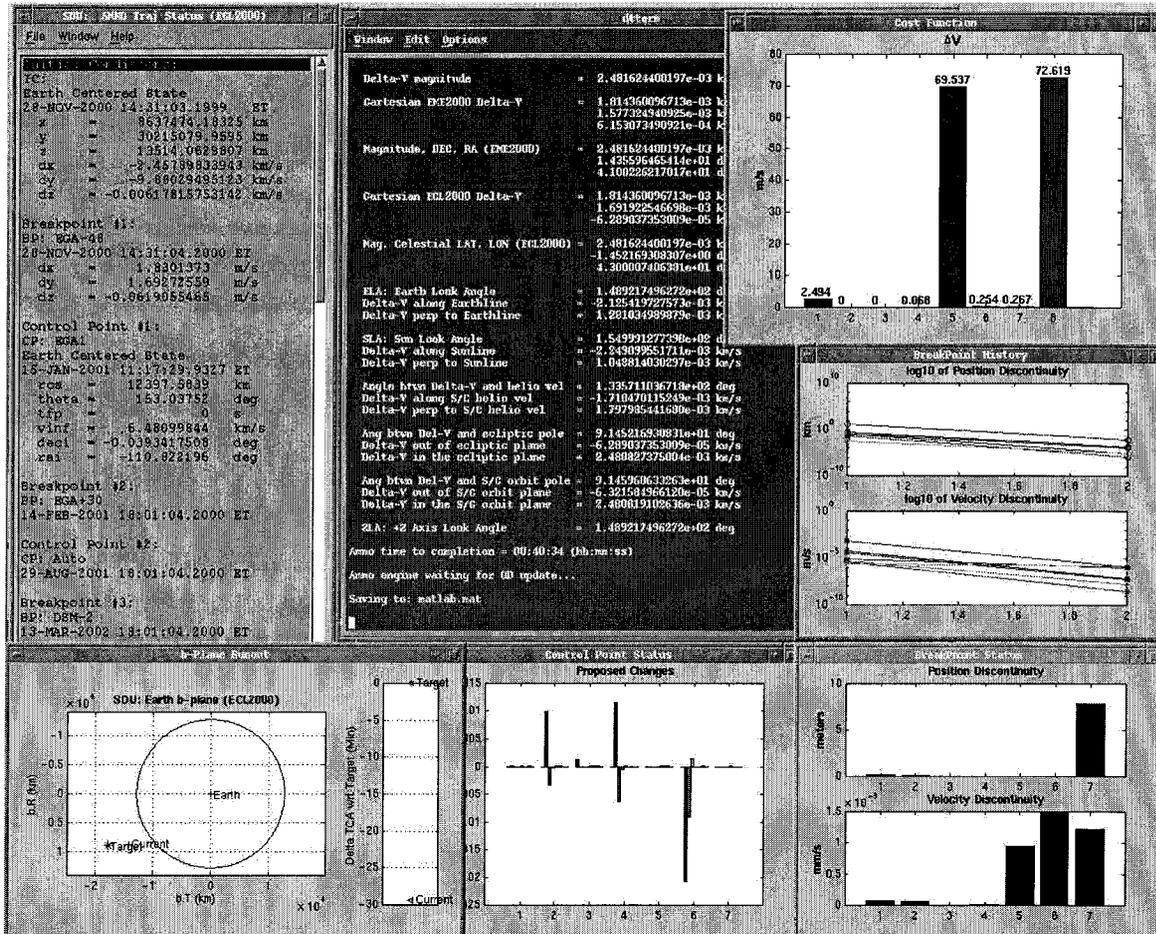


Figure 3: AMMO Display Example

## BENEFITS

An automated ground system for maneuver design significantly reduces workforce requirements, saving time and money. The AMMO system has the potential to ultimately replace a maneuver analyst in the operational determination of trajectory correction maneuvers. The maneuver analyst can place more focus on the critical mission issues and less effort on the repetitive and time consuming support task of re-optimization. In the current era of multi-mission support, additional efficiencies would result from using the same system for maneuver design across multiple projects. The AMMO capability can also be integrated with an automated orbit determination process<sup>9,10</sup> as part of a completely automated navigation system. This proposed system also has potential for future application in an autonomous flight system.

The substantial timesaving can also be transferred into mission performance improvements by utilizing shortened TCM design schedules. The orbit determination process can obtain additional tracking data, and subsequently

improve delivery accuracy for science benefit. For gravity assist targeting, statistical  $\Delta v$  requirements get reduced as a result of improved delivery accuracy.

An automated system improves maneuver reliability and the efficiency of the design solution. The maneuver support level is elevated to 24 hours a day / 7 days a week. Additionally, reference trajectory updates and the subsequent products become routinely available to support trajectory predictions for the Deep Space Network and planning for Sequence and Science activities.

## **FUTURE PLANS**

Future plans include a transition of the AMMO prototype capability into official navigation software and procedures. Currently, there is an ongoing effort to re-implement the legacy navigation software system with a modular design to promote future development efforts. Integration with this task will enable a more efficient AMMO implementation, and eliminate the duplicate modeling effort that currently exists for the prototype system to function with the current navigation software. In the meantime, maintenance of the prototype AMMO system improves operational efficiency, and eliminates the need for repetitive manual analysis.

## **CONCLUSION**

The prototype AMMO system has demonstrated the capability to routinely and automatically re-optimize the spacecraft trajectory in a flight operations environment. This capability has proven to be valuable during time critical mission events and for extended maneuver analysis support with limited resources. Additionally, the control algorithm provides a systematic and effective approach toward solving complex and numerically unstable trajectory optimization problems.

## **ACKNOWLEDGMENT**

This research was carried out at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration.

The AMMO system utilizes as much existing software and algorithms as possible from the Navigation and Mission Design Section at the Jet Propulsion Laboratory. Accordingly, acknowledgment is given to all those associated with the historical development of these software resources. Special thanks go to Larry Bright, Ed Rinderle, Jack Hatfield, Paul Finlayson, and Dennis Byrnes for their helpful overview and insight into the CATO optimization problem. Thanks also go to Jan Ludwinski, the Europa Orbiter project, and the InterPlanetary Network and Information Systems Technology program for their consistent support. Prem Menon, Tung-Han You, and the Stardust project provided the opportunity to successfully demonstrate the AMMO system in an actual flight operations environment.

## REFERENCES

1. D. V. Byrnes and L. E. Bright, L. E., *Design of High-Accuracy Multiple Flyby Trajectories Using Constrained Optimization*, AAS Paper 95-307, AAS/AIAA Astrodynamics Specialist Conference, Halifax, Nova Scotia, Canada, 14-17 August 1995.
2. C. L. Lawson and R. J. Hanson, *Solving Least Squares Problems*, Classics Edition, Classics in Applied Mathematics 15, SIAM Books, (ISBN 0-89871-356-0), 1995.
3. E. H. Maize, *Linear Statistical Analysis of Maneuver Optimization Strategies*, AAS Paper 87-486, AAS/AIAA Astrodynamics Specialist Conference, Kalispell, Montana, 10-13 August 1987.
4. J. L. Bell, M. W. Lo, R. S. Wilson, *Genesis Trajectory Design*, AAS Paper 99-398, AAS/AIAA Astrodynamics Specialist Conference, Girdwood, Alaska, 16-19 August 1999.
5. J. C. Smith, *Description of Three Candidate Cassini Satellite Tours*, AAS Paper 98-106, AAS/AIAA Space Flight Mechanics Meeting, Monterey, California, 9-11 February 1998.
6. B. Raofi, M. D. Guman, and C. L. Potts, *Preliminary Statistical  $\Delta V$  Analysis for a Representative Europa Orbiter Mission*, AIAA Paper 2000-4035, AIAA/AAS Astrodynamics Specialist Conference, Denver, Colorado, 14-17 August 2000.
7. J. R. Johannesen and L. A. D'Amario, *Europa Orbiter Mission Trajectory Design*, AAS Paper 99-360, AAS/AIAA Astrodynamics Specialist Conference, Girdwood, Alaska, 16-19 August 1999.
8. C. L. Yen and E. A. Hirst, *Stardust Mission Design*, AAS Paper 97-707, AAS/AIAA Astrodynamics Specialist Conference, Sun Valley, Idaho, 4-7 August 1997.
9. W. C. Masters and V. M. Pollmeier, *Development of a Prototype Real-Time Automated Filter for Operational Deep Space Navigation*, Third International Conference on Space Mission Operations and Ground Data Systems, Goddard Space Flight Center, Greenbelt, Maryland, 14-18 November 1994.
10. V. M. Pollmeier, P. D. Burkhart, and T. R. Drain, *ARTSN: An Automated Real-Time Spacecraft Navigation System*, Flight Mechanics/Estimation Theory Symposium, Goddard Space Flight Center, Greenbelt, Maryland, 1996.