

EVALUATION OF AN ENERGY-CUTOFF ALGORITHM FOR THE SATURN ORBIT INSERTION BURN OF THE CASSINI-HUYGENS MISSION

Troy Goodson¹

The Cassini-Huygens spacecraft was launched on October 15, 1997 as a joint NASA/ESA mission to explore Saturn. After a 7-year cruise the spacecraft will enter orbit around Saturn on July 1, 2004 for a 4-year investigation of the Saturnian system. This paper describes the navigation-related aspects and analysis of the cut-off algorithm for the orbit-insertion burn. The algorithm is designed to achieve the desired orbital period even after the burn has been restarted due to possible burn interruptions.

The cut-off algorithm uses an approximation to the change in orbital energy due to the maneuver. The targeted change in orbital energy is chosen so that a nominal burn will achieve the desired orbital period. In the case of an interrupted and restarted burn, achieving the target gives only small errors in the final orbital period. The analysis performed breaks down the algorithm's targeting error and demonstrates that the cut-off algorithm is suitable for a wide range of interruption durations.

INTRODUCTION

The Cassini-Huygens spacecraft was launched on October 15, 1997 as a joint NASA/ESA mission to explore Saturn. After a 7-year cruise the spacecraft will enter orbit around Saturn on July 1, 2004 for a 4-year investigation of the Saturnian system. This paper describes the navigation-related aspects and analysis of the cut-off algorithm for the orbit-insertion burn. The algorithm is designed to achieve the desired orbital period even after the burn has been restarted due to possible burn interruptions.

There are a series of approximations that have been made in designing the cut-off algorithm from the perspective of navigation. The most significant approximation is linearization, which for reasonable single-interruption cases, results in about 2 m/s error. Other approximations are the use of a two-body conic orbit to represent the trajectory relative to Saturn, use of an on-board algorithm to compute that conic, a

¹Member of the Technical Staff, Jet Propulsion Laboratory, California Institute of Technology, Pasadena, CA 91109

small timing difference between attitude and acceleration measurements, and the use of Euler integration. The error resulting from these latter approximations is shown to be much smaller than 2 m/s.

An algorithm was presented previously [1] which was referred to as *Smart Burn*, but the algorithm presented here is more compatible with the current implementation of flight software. It is able to handle variable burn-interruption durations. And, it has the feature that the nominal burn is treated identically to failure cases, simplifying the associated design of flight software.

NOMINAL DESIGN

The Saturn-Orbit Insertion (SOI) maneuver is arguably the most important in the Cassini-Huygens mission as the opportunity to achieve an elliptical orbit around Saturn passes in a matter of hours. The spacecraft approaches Saturn with a V_∞ of 5.2 km/s. The inclination of the Saturn-relative orbit with respect to Saturn's equator is about 17° . The spacecraft approaches from the southern hemisphere of Saturn. At a distance of about 158,500 km, Cassini-Huygens makes an ascending ring-plane crossing on July 1, 2004 at about 00:47 UTC. The spacecraft traverses the ring-plane with the high-gain antenna (HGA) in the velocity direction to help protect against debris impacts. Following this ascending ring plane crossing (ARPC), the spacecraft will rotate to follow the constant-rate turn for the maneuver. Throughout the burn, the spacecraft will rotate at about 0.008° per second. On July 1, 2004 at 01:12 UTC, the SOI maneuver begins. Pericrone is reached at about 2:38 UTC and the burn nominally ends at 2:48 UTC (duration of 96 min.). Burn termination, even for nominal execution, is governed by the Energy-Cutoff Burn (ECB) algorithm. For the nominal burn duration, the spacecraft will have rotated by about 46° . Some science activity is planned for the period following SOI. A descending ring plane crossing (DRPC) occurs at a distance of about 158,500 km roughly July 1, 2004 04:34 UTC.

With a nominal cost of 626 m/s, the maneuver slows the spacecraft. Following termination, the Saturn-relative orbit has a period of 116 days (Later, the Periapsis-Raise Maneuver increases that period to 124 days.) SOI primarily alters the semi-major axis of the Saturn-relative orbit; therefore, the burn is best executed nearly parallel to the spacecrafts Saturn-relative velocity. At the start, the acceleration vector lags the anti-velocity vector by about 7° . At the nominal end, the acceleration vector still lags by about 2° .

The cut-off algorithm for SOI and the control of the burn direction are two independent algorithms. These two algorithms both rely on a third component, call the Inertial Vector Propagator (IVP). IVP propagates, among other things, position and velocity on a conic orbit. A conic orbit is used to model the velocity as a function of time during SOI execution for the cut-off algorithm. This state of this orbit at any given time is referred to as *the Cassini-Saturn vector*. A second, circular orbit is used to model the burn direction as a function of time relative to the position of the spacecraft. The burn direction at a given time is referred to as *the CRITICAL_INERT*

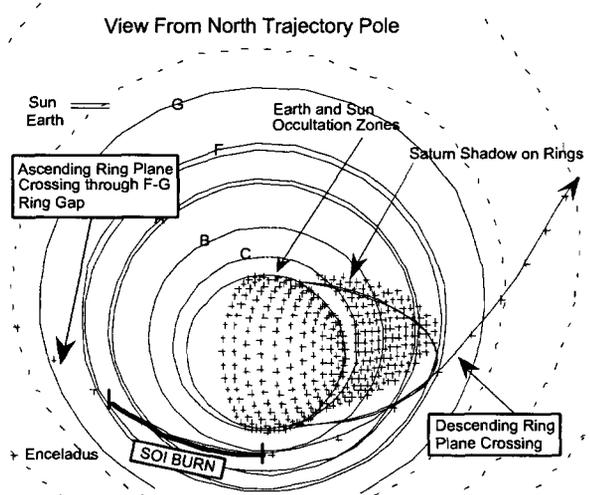


Figure 1: Saturn-relative (cronocentric) trajectory near SOI.9.

vector.

The parameters that constitute a design for SOI are the IVP definition for the Cassini-Saturn vector, the IVP definition for the CRITICAL_INERT vector, the targeted change in energy, the min/max burn cut-off parameters, and some min/max times. The latter two min/max items are beyond the scope of this paper.

SOI is designed first in CATO [2], where the maneuver is chosen to minimize total mission deterministic ΔV . In CATO, SOI is modeled by polynomials of spherical coordinates² of the Earth Mean Equator and Equinox of J2000.0 (EME2000) coordinate system. The ΔV is 625.4 m/s. The acceleration vector starts with a declination of -1.446° and a right ascension of 237.69° , which then increase by $0.00157^\circ/\text{s}$ and $0.00707^\circ/\text{s}$, respectively.

The SOI design is refined using the DPTRAJ [3] software set, SEPV in particular. DPTRAJ's capabilities for modeling thrust direction differ slightly from the spacecraft's onboard flight software's capabilities. Although DPTRAJ can model maneuver thrust directions in a variety of ways, the method most appropriate for SOI is polynomials of spherical coordinates. The maneuver's right ascension and declination may each be polynomial functions of time. The maneuver magnitude as well as bias terms (0th order terms) in the polynomials for right ascension and declination (a total of 3 parameters) may be targeted to 3 trajectory parameters. However, the orientation of the coordinate system in which the spherical coordinates are expressed may not be targeted. For SOI, a coordinate system called VIEW2 [3] is used, where the Y_{view2} axis is along the velocity direction at the start of SOI, the Z_{view2} axis is along the Saturn-relative orbital angular momentum vector, and the X_{view2} axis completes the right-handed system. The polynomial for right ascension has a bias and first-order

²the coordinates are referred to as magnitude, right ascension, and declination

term (262.96° , $0.008^\circ/\text{sec}$), while the polynomial for declination has only a bias term (0.05128° , its first-order term is zero). As a result, the SOI maneuver, as targeted, tracks arcs of constant declination in VIEW2.

The main goal for the SOI burn is achieve a given orbital period. As orbital period for a two-body conic orbit is equivalent to orbital energy, the ECB algorithm is designed on the basis of achieving a certain target for the change to orbital energy.

The ECB algorithm continues execution of the SOI maneuver until the computed change in energy matches the specified target. The computed change in energy accounts for the measured acceleration and, therefore, any periods during which the maneuver is interrupted, viz. while the spacecraft recovers from a fault. For single-interruption cases, Figure 2 shows the ΔV magnitude of SOI as modeled and computed with DPTRAJ.

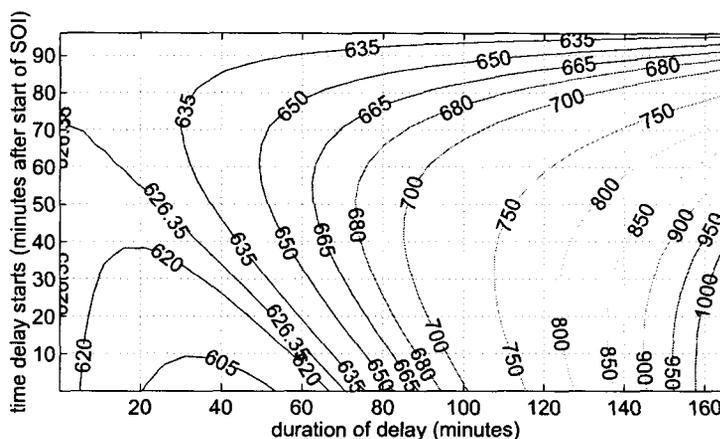


Figure 2: ΔV magnitude of SOI for single-interruption cases. Contours labeled m/s.

Table 1 contains updated parameters for the CRITICALINERT IVP vector definition and Table 2 contains updated parameters for the Cassini-Saturn vector. Note that the start and end times listed in the table are not recommendations from Navigation, they are based on considerations outside of navigation. The ΔE_{target} is set at $17.830 \text{ km}^2/\text{s}^2$.

The IVP definition for the Cassini-Saturn vector was made by determining the position and velocity at pericrone of a trajectory without the SOI maneuver. That state was propagated to an earlier time consistent with the command in the spacecraft command sequence. The CRITICALINERT vector, governed spacecraft onboard flight software capabilities, is modeled using a conic vector centered at the spacecraft. For simplicity, CRITICALINERT is represented with a circular orbit. This choice limits burn direction profiles to great circles as opposed to circles of constant

Table 1: Parameters for CRITICAL_INERT vector.

Head Object	CRITICAL_INERT	Pos X, <i>m</i>	-6.410046131e+02
Base Object	Cassini	Pos Y, <i>m</i>	-7.633099118e+02
Grav Const, m^3/s^2	1.936308242e+01	Pos Z, <i>m</i>	-8.044292668e+01
Start time (UTC)	2004-182T12:00:00	Vel X, <i>m/s</i>	1.031146290e-01
End time (UTC)	2004-183T06:00:00	Vel Y, <i>m/s</i>	-8.944132554e-02
		Vel Z, <i>m/s</i>	2.703155532e-02

Table 2: Parameters for Cassini-Saturn vector.

Head Object	Cassini	Pos X, <i>m</i>	-1.793558482e+09
Base Object	Saturn	Pos Y, <i>m</i>	-1.824737627e+09
Grav Const, m^3/s^2	3.793126773e+16	Pos Z, <i>m</i>	-3.960285223e+08
Start time (UTC)	2004-180T01:28:56	Vel X, <i>m/s</i>	5.977503816e+03
End time (UTC)	2004-186T07:10:00	Vel Y, <i>m/s</i>	4.707756102e+03
		Vel Z, <i>m/s</i>	1.293118804e+03

declination. To make a reasonable match between these two, the IVP definition for CRITICAL_INERT was computed as follows:

\vec{u}_1 = initial SOI burn pointing direction, unit vector, EME2000

\vec{u}_2 = final SOI burn pointing direction, unit vector, EME2000

$$\vec{p} = \vec{u}_1 \times \vec{u}_2 / |\vec{u}_1 \times \vec{u}_2|$$

$$\omega = \text{acos}(\vec{u}_1 \cdot \vec{u}_2) / \Delta t$$

$$\vec{r} = (1000m)\vec{u}_1$$

$$\vec{v} = \omega * \vec{p} \times \vec{r}$$

$$\mu = |\vec{r}|^3 \omega^2$$

Consider (μ, \vec{r}, \vec{v}) to represent a two-body conic orbit at the SOI burn start time, centered at the spacecraft. In particular, this conic is a circle. Propagate backwards (μ, \vec{r}, \vec{v}) as a simple two-body conic from the burn start time to the start time of the CRITICAL_INERT IVP definition.

IVP VS. DPTRAJ BURN MODELING

As discussed above, the DPTRAJ and IVP modeling for the burn pointing direction CRITICAL_INERT are different. The angle between the IVP and DPTRAJ pointing directions is plotted in Figure 3, below. The plot shows a maximum error during the nominal burn of about 0.4°. For SOI maneuvers with interruptions, burn execution

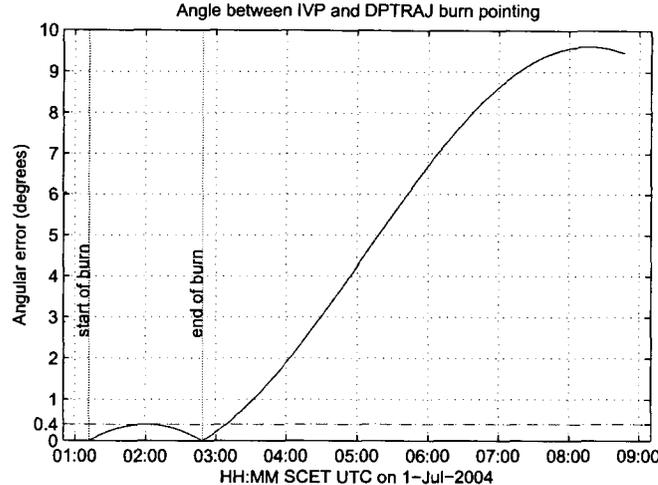


Figure 3: **Angle between the DPTRAJ SOI burn pointing direction from the reference trajectory and the IVP definition in Table 1.** During the nominal SOI burn, the maximum angular error is 0.4° .

will end later and the angular error will increase, but the duration should never be long enough to reach the maximum of about 9.2° . In any case, this error has little effect on the achieved orbital energy as this quantity is second-order in pointing error.

Furthermore, the IVP definition from Table 1 has been modeled in SEPV to see what deterministic error it incurs. These simulations are identical to what produced the reference trajectory 030201 except that SOI's burn direction is from Table 1 and the SOI clean-up maneuver was included. The SOI clean-up maneuver is scheduled to be two days after SOI. Maneuvers after SOI were retargeted to the reference trajectory aimpoints. The maneuvers and targeting strategies have been described in an earlier paper [4].

Two different cases were run; the first with SOI clean-up at SOI + 2 days, the second with SOI clean-up at SOI + 16 days. The resulting ΔV for each maneuver from SOI clean-up to Probe Targeting Maneuver (PTM) is listed in Table 3. The table shows that for the baselined clean-up at SOI + 2 days, the clean-up is about 1.2 m/s and the cost to downstream maneuvers is negligible. For the case of a clean-up at SOI + 16 days, the clean-up is about 3.1 m/s and the impact on downstream maneuvers is actually a savings of about 0.4 m/s. The ΔV cost associated with this pointing difference is clearly quite small.

ERROR ANALYSIS

The ECB algorithm involves several approximations, for most of which analyses appear below. These analyses show that the largest error is due to the assumption of

Table 3: **Deterministic ΔV Cost of Pointing Errors (m/s)**. Three cases are shown: SOI pointing as in the reference trajectory, SOI pointing with IVP and SOI clean-up at +2 days, and clean-up +16 days.

	ref. traj.	+2d clean-up	+16d clean-up
SOI clean-up	0	1.2	3.1
PRM	391.7	391.7	391.3
OTM-006	1.2	1.2	1.2
PTM	15.6	15.6	15.6

linearity with respect to burn ΔV . Other errors arise from the use of Euler’s method for numerical quadrature, use of a conic model for spacecraft state propagation, and timing differences in the spacecraft’s onboard flight software.

LINEARIZATION

Orbital energy for a two-body conic is $E(t) = -\mu/r(t) + (1/2)\vec{V}(t) \cdot \vec{V}(t)$, where $r(t)$ is the spacecraft’s distance and $\vec{V}(t)$ is the spacecraft’s velocity. If an acceleration $\vec{a}_{SOI}(t)$ is applied to the spacecraft, then, as may be found in many astrodynamics textbooks, the change in orbital energy may be computed with

$$\Delta E(t) = \int_0^t \vec{V}(\tau) \cdot \vec{a}_{SOI}(\tau) d\tau \quad (1)$$

Assuming that the influence of \vec{a}_{SOI} is small enough to be linear, then variations in the spacecraft state may be written as

$$\begin{bmatrix} \delta\vec{r}(t) \\ \delta\vec{V}(t) \end{bmatrix} = \Phi(t, t_0) \begin{bmatrix} \delta\vec{r}(t_0) \\ \delta\vec{V}(t_0) \end{bmatrix} + \int_0^t \Phi(t, \sigma) \begin{bmatrix} 0 \\ \vec{a}_{SOI}(\sigma) \end{bmatrix} d\sigma$$

Where $\delta\vec{r}(t_0)$ and $\delta\vec{V}(t_0)$ represent variations in the initial position and velocity vectors, respectively. $\Phi(t, t_0)$ is the transition matrix from the initial time to time t . If we partition the state transition matrix as

$$\Phi(t, \sigma) = \begin{bmatrix} \Phi_{11}(t, \sigma) & \Phi_{12}(t, \sigma) \\ \Phi_{21}(t, \sigma) & \Phi_{22}(t, \sigma) \end{bmatrix}$$

and restrict our attention to the \vec{a}_{SOI} only, meaning $\delta\vec{r}(t_0) = 0$ and $\delta\vec{V}(t_0) = 0$, then we can write an approximation to the spacecraft’s velocity as

$$\vec{V}(t) = \vec{V}_{no-SOI}(t) + \int_0^t \Phi_{22}(t, \sigma) \vec{a}_{SOI}(\sigma) d\sigma$$

where $\vec{V}_{no-SOI}(t)$ is the spacecraft's velocity time-history if SOI is not executed. Substituting this into Equation 1, above, gives

$$\Delta E(t) = \int_0^t \vec{V}_{no-SOI}(\tau) \cdot \vec{a}_{SOI}(\tau) d\tau + \underbrace{\int_0^t \int_0^t [\Phi_{22}(\tau, \sigma) \vec{a}_{SOI}(\sigma)] \cdot \vec{a}_{SOI}(\tau) d\sigma d\tau}_{\text{second-order in } \vec{a}; \text{ throw away}}$$

Throwing out second-order terms gives

$$\Delta E(t) = \int_0^t \vec{V}_{no-SOI}(\tau) \cdot \vec{a}_{SOI}(\tau) d\tau \quad (2)$$

This, then, is the change in energy, linearized by the assumption that the SOI burn ΔV is small compared to the spacecraft velocity.

The interest here is how much of a difference it makes when the cut-off algorithm is based on Equation 1 versus 2. The singular difference between these two equations is that the spacecraft's velocity in Equation 1 includes the effects of the SOI maneuver but Equation 2 uses $\vec{V}_{no-SOI}(t)$, which does not include any effect due to SOI. Consequently, it is useful to note the differences between the planned spacecraft trajectory and one that doesn't account for SOI.

These differences may be seen in Figures 4 and 5. Figure 4 shows the difference in speed in the anti-SOI direction (roughly along track). This is $(\vec{V}(t) - \vec{V}_{no-SOI}(t)) \cdot \vec{a}_{SOI}(t) / |\vec{a}_{SOI}(t)|$. Although integral of the magnitude of the SOI acceleration is 626 m/s, the difference in along-track speed at pericrone is a little over 400 m/s, due to orbital mechanics. The plot shows a negative value because the trajectory with SOI is slower than the trajectory without it.

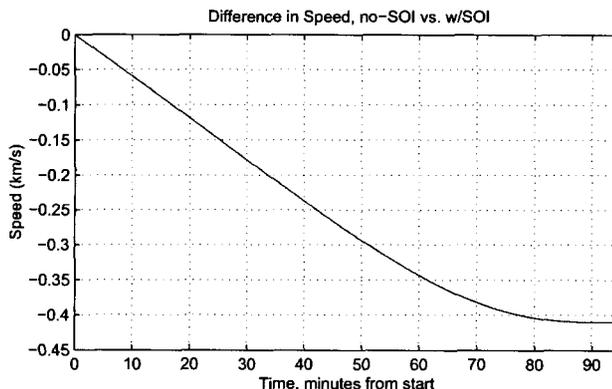


Figure 4: **Speed Difference.** The decrease in speed of the trajectory with SOI, along the anti-SOI direction compared to the no-SOI trajectory.

Figure 5 shows position differences along two directions, the anti-SOI direction and the nearly-radial direction which is perpendicular to the anti-SOI direction yet

still nearly in-plane. This plot shows the SOI trajectory falling behind the no-SOI trajectory by about 1,3005 km. As it falls behind, it also drops below, closer to Saturn by about 750 km.

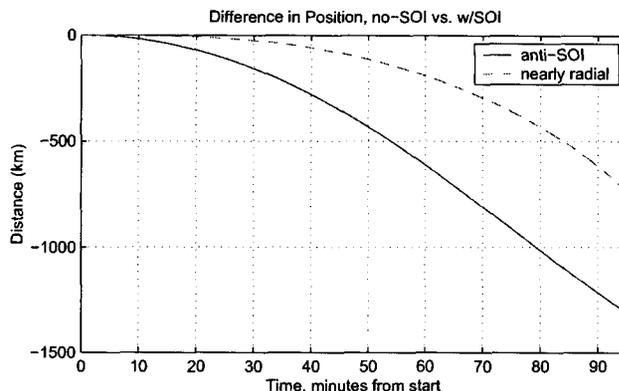


Figure 5: **Position Difference.** The trajectory with SOI falls "behind" and "below" the trajectory without SOI

In order to validate the assumption that SOI is a small acceleration, simulations of the ECB algorithm (with and without this linearization) have been performed. The velocity time-history from the trajectory with SOI was used with Equation 1 to compute a $\Delta E_{target,SOI}$. Then, several different fault times and delay durations were used in simulations with the SEPV software of DPTRAJ, resulting in SOI burn durations such that the target energy was met in each case.

The velocity time-history from the trajectory without SOI was used with Equation 2 to compute a $\Delta E_{target,no-SOI}$. The same set of fault times and delay durations were used to find SOI burn durations such that the target energy was met. The velocity time-history was computed with DPTRAJ and stored in a file, but the ECB algorithm was simulated with MATLAB.

The SOI ΔV results from these two simulations were compared for each fault time and delay duration. A contour plot may be made of this difference in SOI ΔV ; such a plot is shown in Figure 6. This plot shows the difference in SOI ΔV for an algorithm based on Equation 1 versus an algorithm based on Equation 2.

The left-hand border of the plot consists of cases where the delay duration is zero, which is equivalent to absence of a fault; therefore, both algorithms give the same result along the left-hand border. The top border represents SOI cases where the fault occurs just at the end of the nominal execution, so no further execution is needed; therefore, both algorithms also give the same result along the top border. The bottom border represents cases where the beginning of SOI execution has been delayed.

Reasonable delays may be considered to be up to two hours long. For such reasonable delays, the error due to linearization (use of Equation 2) is no more than 2 m/s.

This error is small, about 0.3 % of SOI ΔV magnitude, and clearly shows that SOI's acceleration is within the linear range of this problem.

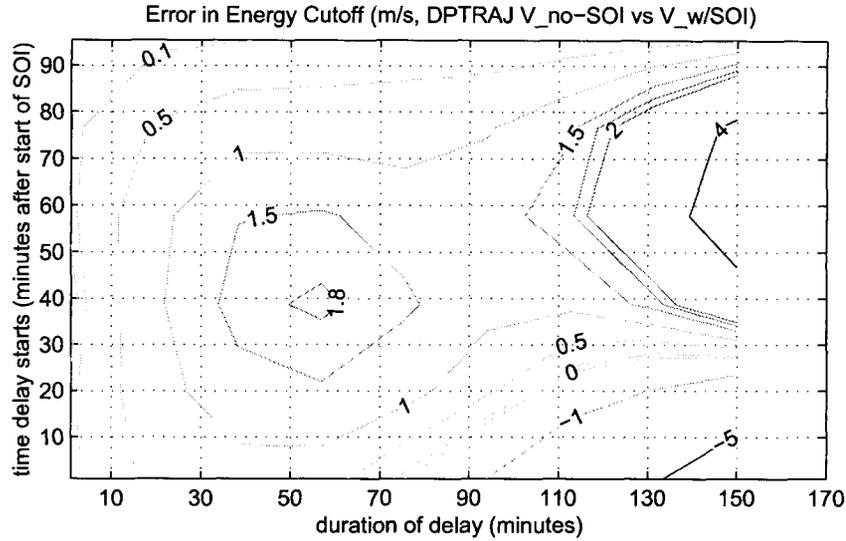


Figure 6: **Contours of Errors due to Linearization.** "time delay starts" is the time when the burn interruption begins. "duration of delay" is the length of the interruption before the burn is restarted. The contours are the ΔV error due to linearization.

USE OF A TWO-BODY CONIC

Of interest is the cost of using a conic trajectory to obtain \vec{V}_{no-SOI} as opposed to a DPTRAJ-generated, without-SOI trajectory. However, the question immediately arises as to how to pick the conic. Although this might be best answered in a detailed study, a couple of simple approaches are evaluated. A first approach is to choose the osculating conic at the start time of SOI. The second is to choose the osculating conic at the pericrone of a trajectory as if SOI were not executed (the no-SOI pericrone). The latter is likely to be more accurate as the velocity error is least when velocity magnitude is greatest; however, results from both choices are presented.

One way to get a feel for the error introduced is to plot the error in velocity vectors in the along-track, cross-track, and out-of-plane directions. Such is plotted in Figure 7, for the conic osculating at the start of SOI, and in Figure 8, for the conic osculating at pericrone. Note that errors³ are typically smaller in the pericrone case, indicating that cut-off errors of the ECB algorithm will be smaller

³particular attention should be paid to along-track errors as these are more indicative of errors in $\vec{V} \cdot \vec{a}$

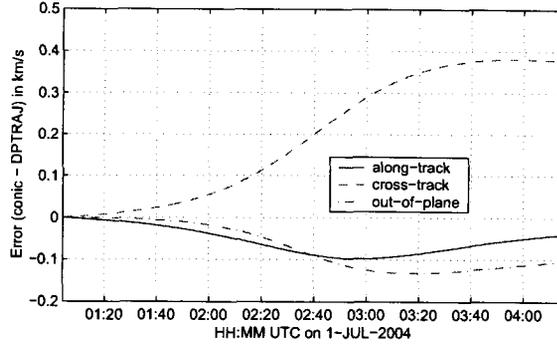


Figure 7: Error (km/s) in conic-propagated S/C velocity vs. DPTRAJ-propagated S/C velocity. Neither includes ΔV_{SOI} . The conic osculates the DPTRAJ trajectory at the nominal start of SOI

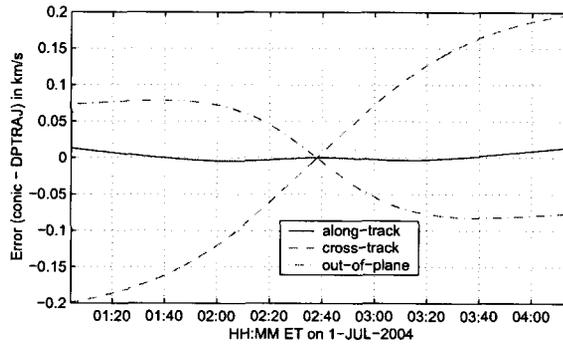


Figure 8: Error (km/s) in conic-propagated S/C velocity vs. DPTRAJ-propagated S/C velocity. Neither includes ΔV_{SOI} . The conic osculates the DPTRAJ trajectory at pericrone

Equation 2 was integrated with the start-of-SOI conic to compute a $\Delta E_{target,start-conic}$ and then to simulate the cut-off algorithm for a variety of burn interruptions. This was repeated for the pericrone conic, producing a $\Delta E_{target,peri-conic}$ and results for the same variety of burn interruptions. The resulting commanded ΔV in either simulation was compared to yet another simulation that was based off a DPTRAJ-generated trajectory. Figure 9 shows the error in commanded ΔV (cut-off ΔV) for a single interruption where the conic osculates at the start of SOI (see Figure 7) versus the DPTRAJ-generated trajectory. The vertical axis marks when the interruption began (in minutes after the start of SOI) and the horizontal axis shows how long the interruption lasted (in minutes). Contours are labeled in meters per second. The only difference between this figure and Figure 10 is that the conic osculates the trajectory at pericrone, (see Figure 8).

These contours both show ΔV errors that grow past 1 m/s for relatively long

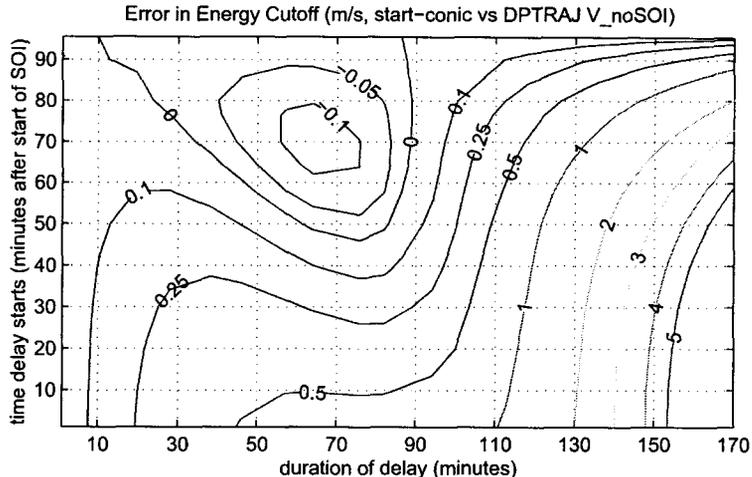


Figure 9: ΔV_{cutoff} error (m/s) for a conic ECB, osculating at the start of SOI, vs. a DPTRAJ-no-SOI ECB. Positive errors indicate that the conic ECB commanded too large a ΔV .

delay durations. Furthermore, it is reasonable to expect, based on earlier plots and discussion, that the pericrone conic will typically outperform the start-of-SOI conic. Even though the pericrone conic is unlikely to be the optimal choice, it serves well. For delay durations of up to two hours, the pericrone-conic results show only about 0.25 m/s error in cutoff ΔV . This is only about 0.04% error and is certainly acceptable.

USE OF IVP

The ECB algorithm will use Cassini’s onboard Inertial Vector Propagator (IVP) [5]. The question arises as to what sort of errors IVP introduces. Here, this is investigated by evaluating the energy integral with an analytical conic propagation and comparing results to those obtained using IVP.

In question is the error introduced by IVP’s conic propagation during the evaluation of the energy integral, Equation 2. For computing $\vec{V}_{no-SOI}(\tau)$, the spacecraft does not have high precision trajectory propagation software on-board, but it does have the IVP software to propagate two-body conic trajectories. However, in order to expedite its calculations, IVP introduces some errors. One of these is that the conic is only recomputed ever RTI (Real-Time Interrupt) which for Cassini-Huygens is about every 1/8 second. These errors are acceptable in terms of what IVP was designed for, but it needs to be shown that the errors are small in terms of SOI’s ECB algorithm.

A straightforward approach for settling the issue is to simulate two energy-based cut-off algorithms, each with different computations of $\vec{V}(t)$; one that uses an analytically-propagated conic⁴ and another that uses IVP simulation data. Note that for both

⁴except that there is no avoiding an iterative solution to Kepler’s equation

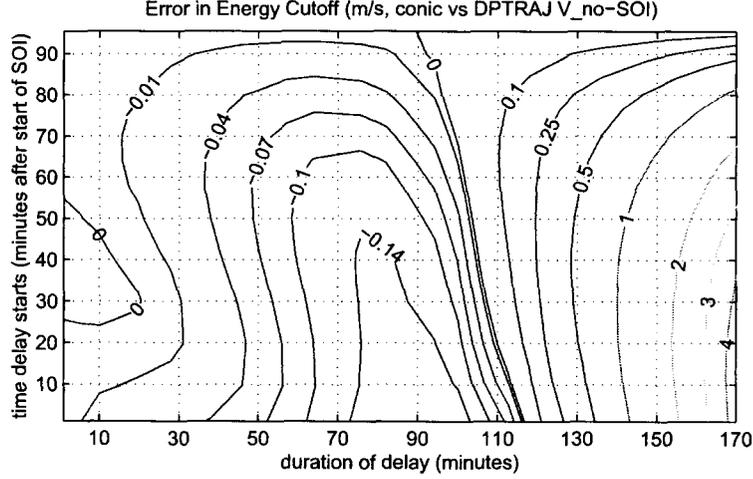


Figure 10: ΔV_{cutoff} error (m/s) for a conic ECB, osculating at pericrone, vs. a DPTRAJ-no-SOI ECB. Positive errors indicate that the conic ECB commanded too large a ΔV .

ECB simulations, appropriate targets are computed, $\Delta E_{target,conic}$ and $\Delta E_{target,IVP}$. The ΔV error is then the difference in cut-off ΔV computed by these algorithms for various single-interruption cases.

The data shown in Figure 11 are calculated as follows. First, $\Delta E_{target,IVP}$ and $\Delta E_{target,conic}$ are computed. Then, a conic-based cut-off algorithm is simulated for each point in the (start-of-interruption time, interruption duration) grid, giving both $t_{end,conic}$ and ΔV_{conic} at each grid point using $t_{end,conic}$ as the burn-termination time. Next, $\vec{V}_{IVP}(t) \cdot \vec{a}(t)$ is integrated to compute a ΔE , call it ΔE_1 , at each grid point.

$$\Delta E_1(i) = \int_{t_o}^{t_{fail}(i)} \vec{V}_{IVP}(\tau) \cdot \vec{a}(\tau) d\tau + \int_{t_{fail}(i)+t_{delay}(i)}^{t_{end,conic}(i)} \vec{V}_{IVP}(\tau) \cdot \vec{a}(\tau) d\tau$$

where t_{fail} is the start-of-delay time and t_{delay} is the interruption duration. Note that the start/end-times of both integrals have to be rounded to the next soonest RTI; this is the aforementioned 1-RTI discretization. The cut-off time of the IVP-based ECB algorithm is estimated with

$$\vec{V}_{IVP}(t_{end,conic}(i)) \cdot \vec{a}(t_{end,conic}(i))(t_{end,IVP}(i) - t_{end,conic}(i)) \approx \Delta E_{target,IVP} - \Delta E_1(i)$$

$$t_{end,IVP}(i) \approx t_{end,conic}(i) - \frac{(\Delta E_1(i) - \Delta E_{target,IVP})}{\vec{V}_{IVP}(t_{end,conic}(i)) \cdot \vec{a}(t_{end,conic}(i))}$$

where i indexes all the points in the (start-of-interruption time, interruption duration) grid. The ΔV error was also estimated linearly, using

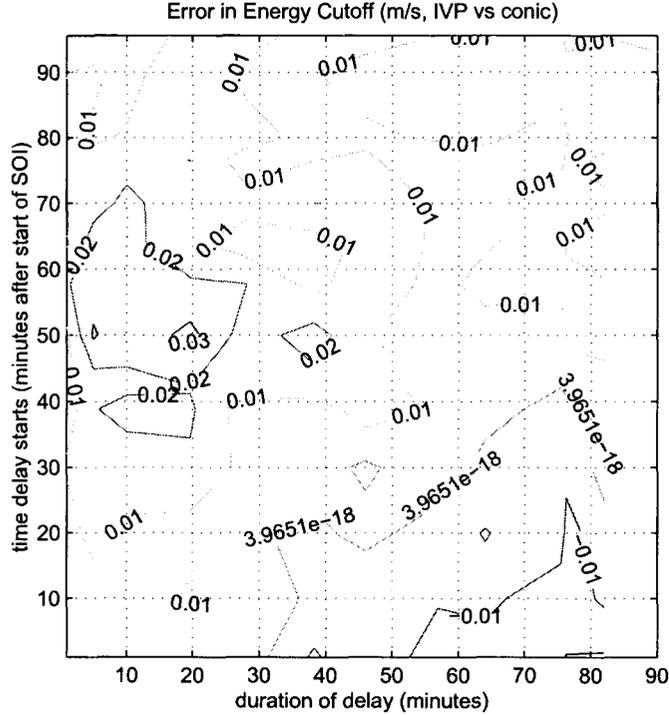


Figure 11: Contours of ΔV cutoff error (m/s) for an IVP-based algorithm vs. a conic-based algorithm. Positive errors indicate that the IVP-based algorithm commanded too large a ΔV .

$$\Delta V_{IVP}(i) \approx \Delta V_{conic}(i) + (t_{end,IVP}(i) - t_{end,conic}(i)) |\vec{a}(t_{end,conic}(i))|$$

It is $\Delta V_{IVP}(i) - \Delta V_{conic}(i)$ that is plotted in Figure 11.

The error shown is about 30 mm/s. The size of this error is, in large part, due to multiple 1-RTI discretizations. Each such discretization could be up to 1/8 sec, with an acceleration of $0.1m/s^2$, that's about 12.5mm/s per discretization. Clearly, these discretizations are a primary contributor to the IVP propagation error. At the same time, this ΔV error is on the order of mm/s which is clearly acceptable.

EULER QUADRATURE ERROR

The baseline plan for the energy-based ECB algorithm is to use a simple Euler quadrature. Intuitively, this error should be small. That intuition is justified with a couple of back-of-the-envelope calculations.

With Euler quadrature one may approximate the integral of some function $\int_a^b f(t)dt$ as a summation over fixed-length intervals $\sum_{i=1}^N f(t_i) ((b-a)/N)$ where $t_i = a + (i-1)(b-a)/N$. An upper bound on the error due to that approximation is $(1/2)Nh^2 f'_{max}$,

where N is the number of intervals, h is $(b-a)/N$, and f'_{max} is maximum of the function's first derivative over the integration interval. Using $Nh = b - a$, this can be re-written as $(1/2)(b - a)hf'_{max}$. $(b - a)$ is about 95.5 minutes and h is one RTI, or 1/8 seconds.

However, f'_{max} requires a little more discussion. For the ECB algorithm, $f = \vec{V} \cdot \vec{a}$, which gives

$$f' = \frac{d}{dt} (\vec{V} \cdot \vec{a}) = \dot{\vec{V}} \cdot \vec{a} + \vec{V} \cdot \dot{\vec{a}}$$

$|\dot{\vec{V}}|$ is μ/r which is a maximum at closest-approach, when r is 805,000 km. So $|\dot{\vec{V}}|_{max}$ is about $6 \times 10^{-3} km/s^2$. $|\dot{\vec{a}}|_{max}$ is F/m_{min} , giving about $1 \times 10^{-4} km/s^2$. $|\ddot{\vec{a}}|_{max}$ is only about $2 \times 10^{-8} km/s^3$. Finally, $|\vec{V}|_{max}$ is about $31 km/s$. If all the vectors lined up, then f'_{max} would be about

$$(6 \times 10^{-3}) \times (1 \times 10^{-4}) + 31 \times (2 \times 10^{-8}) \approx 1 \times 10^{-6} km^2/s^4$$

Of course, the vectors don't quite line up and upon closer examination, a better approximation for f'_{max} is about $3 \times 10^{-7} km^2/s^4$.

Putting all of this back together gives $(1/2)(b-a)hf'_{max} = (1/2) \times (96min \times 60s/min) \times (1/8s) \times (3 \times 10^{-7} km^2/s^4) \approx 1 \times 10^{-4} km^2/s^2$. Given that the nominal change in energy is about $17.8 km^2/s^2$ and the nominal ΔV is about $626 m/s$, a rough scaling, $626 m/s \times (1 \times 10^{-4} km^2/s^2) / (17.8 km^2/s^2)$, says that this is about $4 mm/s$ error in ΔV . This error is much smaller than those accepted, already.

IVP/ATE TIMING

The design of the ECB algorithm ignores the 1/2 RTI mismatch between IVP velocity data and ATE⁵ accelerometer data. There is a mismatch because, while IVP computes updates to the state of Cassini relative to Saturn every RTI, the accelerometer data from ATE is actually an accumulation of acceleration over the past RTI. One interpretation of the ATE data is that it represents the average acceleration, occurring 1/2 RTI ago, implying a 1/2 RTI offset between the ATE and IVP data.

An error of this sort should give it's largest contribution if ΔE_{target} is computed assuming no mismatch or offset but, during execution, such an offset indeed contributes to the $\Delta E_{estimate}$, as computed via Equation 2. The easiest case to judge errors from is the nominal, without any restart or delay. We are interested in

$$\begin{aligned} \Delta E_{target} - \Delta E_{estimate} &= \left(\sum_{i=1}^N \vec{V}(t_i) \cdot \vec{a}(t_i) \right) \Delta t - \left(\sum_{i=1}^N \vec{V}(t_i) \cdot \vec{a}(t_i + RTI/2) \right) \Delta t \\ &< (N\Delta t) |\vec{V}|_{max} |\vec{a}(t_i) - \vec{a}(t_i + RTI/2)|_{max} \end{aligned}$$

⁵Attitude Estimator

where ΔE_{target} is computed without the 1/2 RTI offset, $\Delta E_{estimate}$ is computed for the nominal burn duration but includes the 1/2 RTI offset, Δt is the step size, t_i are the time nodes, and both summations include the same number of time nodes. The right-most expression is the upper-bound on the error introduced by the 1/2 RTI offset. That expression for the upper-bound is convenient because it only involves the maximum difference in acceleration profiles.

Across one RTI, the dominate change in \vec{a} is due to rotation; at $0.008^\circ/s$ for one RTI, the rotation angle is 0.001° . The length $|\vec{a}|$ is roughly $1 \times 10^{-4} km/s^2$, so $|\vec{a}(t_i + RTI/2) - \vec{a}(t_i)|$ is approximately $\sin(0.001 \times \pi/180) * (1 \times 10^{-4} km/s^2) \approx 2 \times 10^{-9} km/s^2$. So

$$\begin{aligned} & (N\Delta t)|\vec{V}|_{max}|\vec{a}(t_i) - \vec{a}(t_i + RTI/2)|_{max} \\ & \approx (96min) \times (60sec/min) \times (31km/s) \times (2 \times 10^{-9} km/s^2) \approx 3.5 \times 10^{-4} km^2/s^2 \end{aligned}$$

Given that the nominal change in energy is about $17.8 km^2/s^2$ and the nominal ΔV is about $626 m/s$, a rough scaling, $626 m/s \times (3.5 \times 10^{-4} km^2/s^2)/(17.8 km^2/s^2)$, puts this at about $12 mm/s$ error in ΔV . This upper bound is fairly conservative, a simulation⁶ of the algorithm suggests this error might only be $2 mm/s$.

CONCLUSION

Obviously, there are a myriad important details The cut-off error for SOI incorporates several approximations. These include linearization (assuming that ΔV_{SOI} is small relative to the cronocentric velocity), use of two-body conic propagation instead of numerical integration, using on-board algorithms for conic propagation, and Euler quadrature. Also, the algorithm absorbs errors due differences between flight and ground software's modeling of burn direction and a slight timing offset between accelerometer data and conic propagation. Out of all of this, the largest error is $\approx 2 m/s$, due to linearizing the energy integral.

The criteria typically used to evaluate such ΔV error is to evaluate the magnitude of the SOI clean-up maneuver that would fix said error. The nominal SOI clean-up location is two days after SOI, at which point SOI magnitude errors scale by approximately 4.7 into SOI clean-up magnitude. The $2 m/s$ error above then becomes almost $10 m/s$ at the clean-up location. Recall that this penalty is only paid if the burn is interrupted. In any case, an SOI clean-up maneuver of $10 m/s$ is a small enough fraction of the $130 m/s$ mission ΔV margin (95%).

References

- [1] D. L. Gray and G. M. Brown. Fault-Tolernat Guidance Algorithms for Cassini's Saturn Orbit Insertion Burn. *Proceedings of the American Control Conference (ACC 905)*, June 1998.

⁶computing ΔE_{target} assuming the 1/2 RTI offset

- [2] D. V. Byrnes and L. E. Bright. Design of high-accuracy multiple flyby trajectories using constrained optimization. *Advances in the Astronautical Sciences, (Proceedings of the 1995 AAS/AIAA Astrodynamics Conference Aug 14-17 1995)*, 90, #1:121, August 1995.
- [3] *DPTRAJ-ODP User's Reference Manual, Vol. 1*. JPL Internal Document 630-336.
- [4] N. J. Strange, T. D. Goodson, and Y. Hahn. Cassini Tour Redesign for the Huygens Mission. *Proceedings of the AIAA/AAS Astrodynamics Specialist Conference, AIAA 2002-4720*, August 2002.
- [5] Gurkirpal Singh. Cassini Inertial Vector Propagator (IVP) Algorithm. *JPL IOM 3455-97-068 (Internal Document)*, November 1997.