OPTIMISATION OF WORST CASE PERIGEE RAISING STRATEGY

M J Smalley & D A Young

British Aerospace, Space and Communications Division, Stevenage, UK.

ABSTRACT

A method of optimising the apogee burns to maximise the worst-case beginning of life mass for a geosynchronous satellite is presented. The propellant budget of a spacecraft has to be designed to take into account the 99th percentile worst case. Optimising the worst-case rather than the nominal transfer strategy results in a reduction in the propellant mass which would be required for the manoeuvre.

A computer program, which has been written to implement the algorithm is briefly described and the results of two case studies are presented. One of these case studies employs a single apogee engine burn with the thrust vector following a slew about a single spacecraft axis. The second case study employs a multi-burn apogee engine manoeuvre with an inertially fixed thrust vector. Both cases show significant reductions in the propellant required for the 99th percentile case.

Keywords: Apogee Engine Firing, Multi-burn AEF, Worst Case Requirements.

1. INTRODUCTION

Of current interest in spacecraft mission analysis is the optimisation of apogee engine firings using multiple long duration burns so as to minimise the propellant requirement for achieving the target orbit.

Previous methods which have been used perform the optimisation in a piecewise manner in that each stage of the mission (launch, apogee engine firings and the station acquisition) are optimised separately. This will not necessarily yield the true optimum strategy for the overall transfer.

In addition to this the propellant has to be budgeted to take into account dispersions in the performance of the launch vehicle, spacecraft propulsion and attitude control subsystems, and the accuracy of the orbit determination. In order to minimise the propellant required it is necessary to optimise the worst case mission subject to these tolerances rather than to optimise the nominal case and then take into account the propellant requirements due to the dispersions.

This paper presents a technique which optimises for the worst-case beginning-of-life mass of a satellite considering all stages of the mission before BOL as a single problem.

A program has been written modelling this technique and results from the software are presented in the paper.

2. THEORY

2.1 Statement of Problem

Optimisation strategies currently exist whereby the transfer strategy which yields the minimum propellant requirement to achieve the target orbit is determined. The spacecraft propellant budget, however, has to be such that dispersions have to be considered typically to a probability level of 99% (2.58σ). Since the variation in the launch vehicle injection errors and burn execution errors give rise to non-symmetric errors about the nominal case, the worst case will therefore be penalising the mission propellant requirements. Figure 1a illustrates a one dimensional example of this.

The figure shows the effect on the propellant required as a function of the duration of the Apogee engine burn. If the apogee engine is permitted to burn too long then the orbit obtained will be higher than the orbit required and station acquisition propellant will have to be used to reduce the orbit, cancelling out the effects of the apogee engine burn. However, if the apogee engine is turned off too soon then the orbit obtained will fall short of the requirement. Hence the station acquisition propellant will again have to be used to correct the orbit, however, in this case it is being used to supplement the apogee engine. Thus the inefficiency is due only to the difference between the performance of the station acquisition thrusters and the apogee engine. The worst case, and hence the propellant requirement, will be the case where the burn is longest.

Figure 1(b) shows the case where the nominal burn time is chosen such that the worst case, taking into account the possible deviation is minimised, and although yielding a higher nominal propellant requirement results in a saving in the overall required propellant.

Proceedings of the Second International Symposium on Spacecraft Flight Dynamics, Darmstadt, FR Germany, 20–23 October 1986 (ESA SP-255, Dec. 1986)

Figure 1(a)

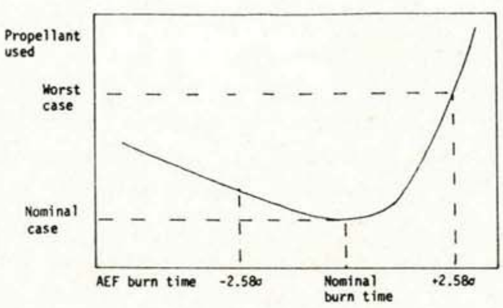
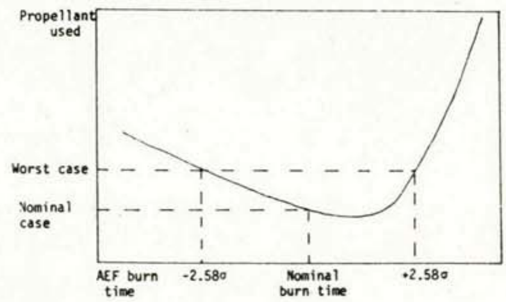


Figure 1(b)



2.2 Worst Case Optimisation

It is desired to be able to optimise the worst case deviation and thereby achieve the best beginning-of-life mass for the worst case rather than for the nominal case.

This is achieved by making the function of the optimisation the worst value which can be found in the region about the nominal point. This region is defined by the tolerances on the values of the optimisation variables due to the dispersions and errors being considered.

This problem may be considered to be an optimisation within an optimisation whereby at each function evaluation the value is maximised in the given subspace.

The method for this is as follows:-

Consider the optimisation of a function F of n variables. We require to find a value for the vector of optimisation variables $\underline{x}$ such that the maximum value of the function in a subspace centred on $\underline{x}$ is a minimum.

The outer optimisation will be performed using an unconstrained optimisation algorithm which requires the first derivatives to be calculated.

The function evaluation method for this algorithm is as follows:

($\underline{x}$ is the optimisation variable vector)

Step F1 Calculate a starting point y_0 for the internal optimisation by calculating the gradients of the function at the nominal point $\underline{x}$ and extrapolating in the direction of steepest ascent of the function to the surface defined by the tolerances.

$$\text{i.e. } \underline{y}_0 = \underline{x} + \underline{a}^T \underline{\sigma} \quad (1)$$

where $\underline{a}$ is the normalised gradient vector

$\underline{\sigma}$ is the vector of tolerances for each variable.

Step F2 Using y_0 as an initial point for the internal optimisation find $\underline{y}^*$ such that $F(\underline{y}^*)$ is maximised and $\underline{y}^*$ lies on the n-dimensional ellipsoid defined by the locus of tolerances centred on x

$$\text{i.e } \sum_i \left(\frac{y_i^* - x_i}{\sigma_i} \right)^2 = 1 \quad (2)$$

Step F3 Return the function value $F(\underline{y}^*)$.

The gradient may be calculated as follows:

Step G1 Calculate the function value $(F(\underline{y}^*))$

Step G2 Calculate the analytic derivatives of the function at the point $\underline{y}^*$

(These will be the same as if the derivative were calculated from changes in the value $F(y^*)$ due to variances in $\underline{x}$ which would have to be performed numerically, involving n internal optimisations).

The function which is to be optimised is the mass of the spacecraft at BOL rather than the mass of the spacecraft after AEF. This has the advantage that the drift orbit does not require to be specified before the optimisation is performed but will be chosen by the algorithm so as to minimise the sum of the mass required for the AEF and the mass required for the station acquisition.

In addition, the orbital elements of the transfer orbit will be included as optimisation variables (those which can be varied) such that the transfer orbit may be optimised together with the transfer to the on-station position.

Thus the entire transfer and drift orbit stages of the mission may be optimised in parallel rather than serially.

2.2.1 Internal Optimisation.

The initial optimisation, required in Step F2 of the function evaluation above requires that $F(\underline{y}^*)$ is maximised subject to the constraint that $\underline{y}^*$ lies on the boundary of the region defined by the tolerances. This introduces the assumption that the worst case will occur at the extremes of the variations, which has been shown to be reasonable for the problem being considered. This simplification allows the internal optimisation to be performed using an unconstrained optimisation algorithm where the optimisation variables are n-1 polar coordinates which define the ellipsoid of worst case values centred on the nominal point. This also eliminates any problems due to the scaling of the optimisation variables.

2.3 Orbit Evolution

The optimisation algorithm requires the BOL mass and the derivatives of the BOL mass with respect to the optimisation variables. To obtain these an orbit evolution has to be carried out from injection into transfer orbit through the coast and thrust arcs up to the start of station acquisition.

2.3.1 Function Evaluation. The orbit is evolved in the following way. The osculating orbital elements at injection, along with the length of the first coast arc are used to calculate the position of the satellite at the end of the first coast arc after taking into account the principal secular and cyclic J2 gravitational perturbations. The first thrust arc is then evolved by integrating with a fixed time step through the burn. This process is then repeated for each subsequent coast arc and thrust arc until after the final burn whereupon the station acquisition propellant requirement is determined to give the final spacecraft mass.

The transfer orbit to be evolved is defined by the optimisation variables which are

r_a the injection apogee radius
i the injection inclination
ω the injection argument of perigee
Ω the injection right ascension of ascending mode
RA, Dec or α, β, δ_0, $\dot{\delta}$, define the burn attitude
ν the length of the coast arc
t the length of the thrust arc
τ the magnitude of the thrust during the burn
ε defines the out of plane dispersion in attitude for single-axis slew burn
I the specific impulse of the station acquisition thrusters.

The thrust can be inertially fixed or a single-axis slew.

For an inertially fixed burn it is defined as

$$\underline{T} = \tau \begin{pmatrix} \cos \text{Dec} \cos \text{RA} \\ \cos \text{Dec} \sin \text{RA} \\ \text{Sin Dec} \end{pmatrix} \qquad (3)$$

and for a single-axis slew burn it is

$$\underline{T} = \tau \begin{pmatrix} \cos\delta \cos\alpha - \sin\delta \sin\alpha \cos\beta \\ \cos\delta \sin\alpha + \sin\delta \cos\alpha \cos\beta \\ \sin\delta \sin\beta \end{pmatrix} \qquad (4)$$

where $\delta = \delta_0 + \dot{\delta} t$
t is time since start of burn.

The thrust control laws are illustrated in Figure 2.

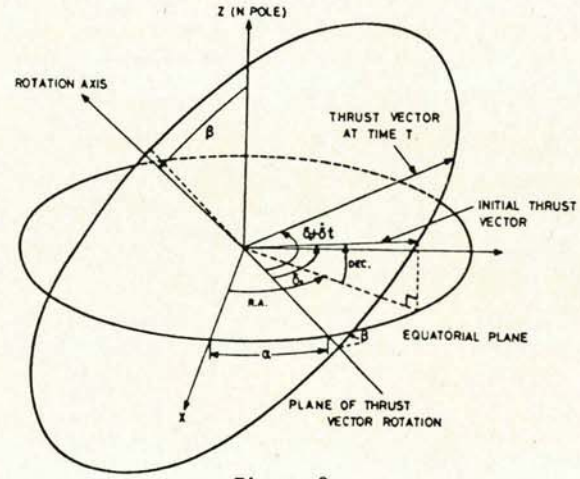


Figure 2

To evolve the orbit the equation of state

$$\ddot{\underline{r}} = -\frac{\mu}{r^3}\underline{r} + \frac{1}{m}\underline{T} \qquad (5)$$

is used, where $\underline{r} = (x, y, z)$
$\ddot{\underline{r}} = (\ddot{x}, \ddot{y}, \ddot{z})$
$r = (x^2 + y^2 + z^2)^{\frac{1}{2}}$

m is the instantaneous mass of the spacecraft
μ is the gravitational constant

For a coast arc $\underline{T} = 0$, hence the position of the satellite at the end of the coast arc of given length can be calculated directly after making allowance for the perturbations.

However, the thrust arc has to be integrated over using a Runge-Kütta routine to update the position of the satellite during the burn.

After the final apogee engine firing the satellite has a mass, M. But the final on-station mass, M_F is given by

$$M_F = M - M_{SA} \qquad (6)$$

where M_{SA} is the station acquisition mass.

To obtain the station acquisition propellant requirement calculate the ΔV requirement. This is determined in two parts as a circularisation or east-west ΔV, and a plane change or north-south ΔV. It assumes a synchronous orbit is required and that the drift orbit is near-synchronous.

$$\Delta V_{NS} = V_S \,(2 - 2\cos\theta)^{\frac{1}{2}} \qquad (7)$$

$$\cos\theta = \sin i \sin i_{OBJ} \cos(\Omega_{OBJ} - \Omega) + \cos i \cos i_{OBJ} \qquad (8)$$

where i is current inclination
Ω is current RA
i_{OBJ} is objective inclination
Ω_{OBJ} is objective RA
V_S is synchronous velocity

$$\Delta V_{EW} = \frac{V_S\, V_N}{4a_s} \qquad (9)$$

where a_s is synchronous radius

$$\text{and } V_N = |a_s + 2\Delta a_D - r_a| + |a_s + 2\Delta a_D - r_p| \qquad (10)$$

where r_a is drift orbit apogee radius
r_p is drift orbit perigee radius

$$\text{and} \quad \Delta a_D = \frac{d.a_s}{540} \qquad (11)$$

with d the required drift rate in degrees per day to achieve target longitude.

The station acquisition propellant is given by the rocket equation

$$M_{SA} = M\left[1 - \exp\left\{-\frac{(\Delta V_{NS} + \Delta V_{EW})}{g.I}\right\}\right] \quad (12)$$

2.3.2. Derivative Evaluation. The optimisation algorithm requires the derivatives of the objective function with respect to each of the optimisation variables.

Differentiating equation (6) in section 2.3.1 with respect to the general optimisation variable, ζ gives

$$\frac{\partial M_F}{\partial \zeta} = \frac{\partial M}{\partial \zeta} - \frac{\partial M_{SA}}{\partial \zeta} \quad (13)$$

The $\frac{\partial M}{\partial \zeta}$ terms are all zero except for the derivative of the end of final burn spacecraft mass with respect to the thrust arc duration. These terms are given by the respective flow rates for each burn from the liquid apogee engine.

The $\frac{\partial M_{SA}}{\partial \zeta}$ derivative can be calculated from the expression

$$\frac{\partial M_{SA}}{\partial \zeta} = \sum_{i=1}^{6} \frac{\partial M_{SA}}{\partial K_i} \cdot \frac{\partial K_i}{\partial \zeta}$$

where K_i are the state variables x, y, z, $\dot{x}$, $\dot{y}$, $\dot{z}$

The $\frac{\partial K_i}{\partial \zeta}$ terms are evaluated in parallel with the evolution of the spacecraft orbit. Initial values are calculated for the $\frac{\partial K_i}{\partial \zeta}$ and they are updated through the burns using a 4th order Runge-Kutta integration technique,

The Runge-Kutta requires the evaluation of $\frac{d}{dt}\left(\frac{\partial K_i}{\partial \zeta}\right)$. These are determined from the state equation and previous iterations. The derivative $\frac{d}{dt}\left(\frac{\partial x}{\partial \zeta}\right)$ is equivalent to the previous $\frac{\partial \dot{x}}{\partial \zeta}$ term.

The expression $\frac{d}{dt}\left(\frac{\partial \dot{x}}{\partial \zeta}\right)$ can be obtained from the state equation (6) by differentiating it with respect to ζ to give

$$\frac{d}{dt}\left(\frac{\partial \dot{\underline{r}}}{\partial \zeta}\right) = -\left(\frac{\mu}{r^3}\right)\left[\frac{\partial \underline{r}}{\partial \zeta} - \frac{3}{r^2}\left(\underline{r} \cdot \frac{\partial \underline{r}}{\partial \zeta}\right)\underline{r}\right] + \frac{1}{m}\left(\frac{\partial \underline{T}}{\partial \zeta}\right) - \frac{\underline{T}}{m}\frac{\partial m}{\partial \zeta} \quad (15)$$

The $\frac{\partial M_{SA}}{\partial K_i}$ terms are calculated analytically from the station acquisition equations in section 2.3.1.

2.3.3 Launch Vehicle Optimisation. It is possible to optimise the performance of a launch vehicle by choosing injection parameters to maximise the spacecraft injection mass.

The launch vehicle user guide provides information relating the injection mass to injection orbit perigee height, apogee height inclination and argument of perigee. The program has injection optimisation variables: apogee height, inclination argument of perigee and right ascension of ascending mode. It uses these to determine an injection mass for the spacecraft and uses the mass trade off values for these optimisation variables to set initial values for $\frac{\partial M}{\partial \zeta}$ derivatives. The initial values for the $\frac{\partial K_i}{\partial \zeta}$ terms are determined analytically from orbital equations relating the position and velocity of the satellite to the injection orbital parameters. These are evolved in the method described in section 2.3.1 and 2.3.2 in order to determine the derivatives of the BOL mass with respect to these optimisation variables.

3. APPLICATION

A program was written to apply the theory described above.

The requirements for the software were as follows:

- o Transfer orbit optimisation

  Options of

  a) Optimisation of Ariane transfer orbit parameters r_a, ω, i,

  b) Optimisation of STS transfer orbit parameters r_a, ω, i, Ω

  c) Fixed injection mass into a fixed transfer orbit.

- o AEF Optimisation

  a) Up to 5 burns

  b) Single axis slew or inertially fixed control laws.

  c) Burns not necessarily at Apogee, (i.e. Perigee top-up manoeuvres)

- o Station Acquisition model.
- o Optimisation of Nominal Strategy or Worst Case Strategy.
- o Facility to fix any of the optimisation variables
- o Confidence analysis to ensure worst case has been found.

Further details of the reasons and application of these requirements are discussed below:-

3.1 Fixing Optimisation Variables

The facility to fix the values of optimisation variables allows tolerances to be considered on parameters which cannot be optimised . For example, the thrust delivered by the LAE is not a variable which can be modified but it is subject to errors. Thus the nominal value is required to be fixed for evaluations of the nominal BOL mass but the errors must be included in the worst case optimisation.

This facility can also be used to consider injection errors into a standard GTO.

3.2 Control Laws

The mathematical representation of the control laws is given in equations (3) and (4). It can be seen that the inertially fixed law is a special case of the single axis slew law with

$$\begin{aligned} \beta &= 90° \\ \dot{\delta} &= 0 \\ \delta_0 &= \text{Dec} \\ \alpha &= \text{RA} \end{aligned} \qquad (15)$$

Thus by fixing the variables for β and $\dot{\delta}$ the control laws for the single axis slew may be used without modification.

The requirements on the thrust vector pointing for the attitude control system will normally be to maintain the thrust vector to within a certain fixed tolerance for the duration of the burn. The worst case for this error will occur if there is an initial set up error of this magnitude and it remains so for the duration of the burn. This can be easily modelled as an error in the slew plane (i.e. a tolerance on δ_0) and an error (normally of equal magnitude) in the direction perpendicular to the slew plane. This direction is a function of α and β, and, because these are variables, cannot be modelled as errors in α and β.

This error requires the inclusion of an additional optimisation variable ε which is fixed at zero for the nominal function evaluations but has a non-zero error.

The errors in α, β and $\dot{\delta}$ can then be set to zero for the single axis slew case. For an inertially fixed control law β is constant and hence the errors can be expressed in α and δ_0 (since the directions are perpendicular and ε need not be used.

3.3 Confidence Analysis

The value returned by the internal optimisation will be a maximum, but it cannot be ensured that the value is the global maximum and not a local maximum. Thus, in order to assess the confidence that the final optimised worst case value is statistically near to the true worst case a confidence analysis is performed.

This takes the form of a Monte-Carlo evaluation of a large number of function evaluations for points on the boundary of the tolerance region and determines the percentage of points found which were worse than the 'worst-case' returned by the optimisation.

A switch was installed which gives the option to consider points both inside the tolerance boundary as well as on it so as to enable a confidence check on the assumption that the worst case will lie on the boundary of the region to be performed.

4. CASE STUDIES

We shall present two case studies to show the propellant mass saving which may be obtained using a worst case optimisation technique.

Case 1 is an Ariane 3 launched geosynchronous spacecraft with a single apogee engine burn, the thrust vector being controlled by a slew about a single spacecraft axis. The transfer orbit parameters will be optimised in this example.

The second case is an Ariane 4 launched geosynchronous spacecraft which uses the Ariane Standard Geostationary Transfer Orbit. The AEF is performed by means of three burns with the thrust vector fixed in inertial space for each burn.

For both cases we first performed an optimisation of the nominal strategy and, by performing a worst case function evaluation about the optimisee nominal point, determined the worst case BOL mass. We then performed a worst case optimisation and present the results for comparison. For both the worst case determinations a confidence analysis was performed to assess the validity of the results.

4.1 Case 1

This considers a spacecraft launched on a dedicated flight of the Ariane 3 launch vehicle thus allowing the transfer orbit elements to be optimised. The orbit circularisation and inclination removal is performed by means of a single burn of the liquid apogee engine which lasts for typically 90 minutes. For the duration of this burn the thrust vector pointing will be controlled by means of a single axis slew control law.

Table 1 shows the tolerances for each variable and the pertinent values of the optimisation variables for both runs.

Table 2 shows both the nominal and worst case beginning of life masses for the two optimisations and clearly shows the gain which can be achieved for the worst case. This gain is equivalent to 65.7% of the random component of the propellant requirement. As expected the nominal performance is degraded slightly but this is of little importance since the propellant budget design will be based on the worst case value.

Comparing the optimised parameters for the two nominal strategies shows that the principal difference is the reduction of the burn time. This is to be expected since the worst possible scenario is if the engine over-performs. (This is also indicated by the worst case for the Nominal Optimisation case which has a large increase in the engine thrust). In order to allow for over-performance the burn is deliberately under-sized and the station acquisition thrusters used to complete the manoeuvre.

For both the 'worst-cases' a confidence analysis was performed with 10,000 random points taken on the boundary and a further 10,000 points in the subspace. For the worst-case corresponding to the nominal optimisation none of the 20,000 points gave a worse BOL mass than that given in Table 2 and for the worst-case obtained by the worst-case optimisation only one point (on the boundary) gave a worse BOL mass. Considering the complexity of the problem and the mathematical impossibility of performing a global optimisation these are very encouraging results.

Table 1

Case 1 Optimised Variables

Variable	Tolerance	Nominal Optimisation		Worst Case Optimisation	
		Nominal Point	Worst Case Position	Nominal Point	Worst Case Point
TO ra (km)	139.58	42100.67	42096.00	42175.89	42176.15
TO i (deg)	0.0593	6.338	6.337	6.291	6.292
TO ω (deg)	0.3431	178.416	178.44	178.443	178.44
α (deg)	0.0	114.351	114.351	114.357	114.357
β (deg)	0.0	7.540	7.540	7.571	7.571
δ_0 (deg)	0.3767	234.895	234.89	234.949	234.950
$\dot{\delta}$ (deg)	0.0	3.1138×10^{-3}	3.2557×10^{-3}	3.2557×10^{-3}	3.1557×10^{-3}
ν (rads)	3.646×10^{-4}	21.775	21.775	21.778	21.778
T (seconds)	0.96	6118.12	6118.04	6032.57	6032.57
τ (N)	4.165	491.680	495.72	491.680	490.685
ε (degs)	0.3767	0.0	-0.08437	0.0	0.3655
R_{CT} I_{SP} (s)	3.0	294.0	293.95	294.0	293.895

Table 2

Case 1 Achieved Parameters

	Nominal Optimisation	Worst Case Optimisation	Difference
Nominal M_{BOL} (Kg)	1542.900	1541.164	-1.736
Worst Case M_{BOL} (Kg)	1523.772	1536.344	+12.572

4.2 Case 2

Case 2 concerns an Ariane 4 dual launch into a standard geostationary transfer orbit. Circularisation at geosynchronous height is achieved by three inertially fixed burns at apogees 3, 5 and 7.

Table 3 gives the values of the optimisation variables and their respective tolerances. As for Case 1 the principal differences are the reduction of the burn times.

Table 4 shows the nominal and worst case BOL masses obtained after both a nominal and a worst case optimisation. For this example the saving represents 45.4% of the random component of the propellant requirement for this manoeuvre.

A confidence analysis was performed on both optimisation cases with 10,000 points on the boundary and 10,000 points inside the boundary. None of these points produced a function evaluation worse than the worst-case determined by the optimisation.

5. Conclusions

A method for optimising the worst-case propellant requirement for the perigee raising and inclination removal manoeuvre and subsequent station acquisition has been presented.

Several possible burn strategies can be modelled for liquid bi-propellant engines, including inertially fixed or single-axis slew burns. A multi-burn strategy can also be accommodated.

The optimisation algorithm considers the optimisation variables in parallel and thus determines a true optimum. A nominal or a worst case optimisation can be performed to determine the beginning-of-life spacecraft mass.

To illustrate the savings in the random component of the propellant budget available by optimising for the worst case beginning-of-life spacecraft mass, two case studies have been presented. These examples reflect the possible burn strategies available for optimisation. Common to both cases

Table 3

Case 2 Optimised Variables

Variable	Tolerance	Nominal Optimisation: Nominal Point	Nominal Optimisation: Worst Case Position	Worst Case Optimisation: Nominal Point	Worst Case Optimisation: Worst Case Point
T.0 r_a((km)	69.66	42353.199	42351.297	42353.199	42353.313
T.0 i (deg)	0.077	7.0	6.9863	7.0	6.9997
T.0 ω (deg)	0.413	178.0	177.9798	178.0	177.993
1st Burn R.A (deg)	0.27	254.44	254.449	254.404	254.401
1st Burn Dec (deg)	0.27	-7.31544	-7.35654	-7.324	-7.323
1st Burn ν (rad)	0.0	15.7019	15.7019	15.7005	15.7005
1st Burn T (secs)	1.0	831.170	831.109	826.141	826.088
1st Burn τ (N)	2.0	486.5	486.866	486.5	486.574
2nd Burn R.A (deg)	0.27	255.841	255.855	255.744	255.742
2nd Burn Dec (deg)	0.27	-7.44121	-7.51306	-7.51451	-7.55892
2nd Burn ν (rad)	0.0	12.5196	12.5196	12.5207	12.5207
2nd Burn T (secs)	1.0	1209.08	1209.02	1204.05	1204.11
2nd Burn τ (N)	2.0	486.5	487.029	486.5	486.444
3rd Burn R.A (deg)	0.8	257.053	257.142	256.951	256.675
3rd Burn Dec (deg)	0.8	-7.50019	-7.97735	-7.61281	-8.28559
3rd Burn ν (rad)	0.0	12.5302	12.5302	12.5316	12.5316
3rd Burn T (secs)	1.0	751.422	751.361	746.326	746.520
3rd Burn τ (N)	7.65	486.5	491.257	486.5	484.151
R_{CT} I_{SP} (s)	3.0	291.0	290.961	291.0	291.230

Table 4

Case 2 Achieved Parameters

	Nominal Optimisation	Worst Case Optimisation	Difference
Nominal M_{BOL} (Kg)	699.471	698.906	-0.565
Worst Case M_{BOL} (Kg)	695.235	697.157	+1.922

are the significant reductions in the random component of the required propellants. Case 1 which considers one single-axis slew burn provides a saving of 65.7% in the random component of the propellant budget for a worst-case optimisation compared with the random propellant requirement for a nominal optimisation. Case 2 models three inertially-fixed burns and the reduction in random propellant for this example is 45.4%.

An added facility of the software is a confidence analysis of the optimised worst case point to indicate whether this point is a global or local maximum. Because the software is computationally fast many thousands of function evaluations can be achieved in a short space of time, typically a few minutes, in order to determine whether the worst case is a local or global maximum to a high degree of confidence.

In conclusion, an optimisation algorithm has been devised which allows for parallel optimisation, can consider one or more single-axis slew or inertially fixed burns, achieves significant reductions in worst case propellant requirements, and is computationally fast at achieving a result.