Paper AAS 05–392

American Astronautical Society

American Institute of Aeronautics and Astronautics

Optimisation of Low-Thrust Orbit Transfers Using the Q-Law for the Initial Guess

Anastassios E. Petropoulos

Seungwon Lee

Jet Propulsion Laboratory,
California Institute of Technology,
Pasadena, CA 91109-8099, USA.

AAS/AIAA Astrodynamics Specialists Conference

Lake Tahoe, California, August 7-11, 2005
AAS Publications Office, P.O. Box 28130, San Diego, CA 92198

AAS 05–392

OPTIMISATION OF LOW-THRUST ORBIT TRANSFERS USING THE Q-LAW FOR THE INITIAL GUESS

Anastassios E. Petropoulos* **and Seungwon Lee†**

The Q-law is a Lyapunov feedback control law for effecting low-thrust orbit transfers between arbitrary pairs of orbits around a central body, provided the eccentricity remains below unity. The Q-law can automatically introduce judiciously positioned coast arcs to reduce propellant consumption (at the expense of longer flight-time). Because the Q-law has been shown to produce results representative of optimal orbit transfers in the literature, in this work we use the Q-law to generate initial guesses for optimising a complex orbit transfer around the Earth. The Q-law produces transfer trajectories with different flight-times and propellant masses (depending on the amount of coasting). Each of the Q-law transfer trajectories is then optimised using the Static Dynamic Control algorithm. Initial guesses based on the Q-law, but with Q-law parameters optimised by a genetic algorithm, are also optimised. The optimal solutions obtained are compared with the Q-law initial guess. The speed of optimisation, and nature of the optimal solutions found, are compared with optimal solutions obtained using the previously available, simplistic initial guesses. The Q-law initial guesses and especially the genetic-algorithm-Q-law initial guesses provide many-fold reductions in optimisation run time over the simple initial guess, and they furthermore have performance representative of the optimal solutions found, unlike the simple initial guess.

INTRODUCTION

The Q-law[1–3] is a Lyapunov feedback control law for effecting low-thrust orbit transfers between arbitrary pairs of orbits around a central body, provided the eccentricity remains below unity. The Lyapunov function of the Q-law is based on the optimal rates of change of the orbit elements over not only the thrust direction but also over the anomaly on the osculating orbit. As such, while Lyapunov feedback control is inherently locally optimal in reducing the Lyapunov function, the nature of its Lyapunov function makes the Q-law "quasi-globally optimal" in reducing the propellant consumption. Furthermore, the Q-law can automatically introduce judiciously positioned coast arcs to reduce propellant consumption (at the expense of longer flight-time). Because the Q-law has been shown[2] to produce results representative of optimal orbit transfers in the literature, in this work we use the Q-law to generate initial guesses for optimising a complex, many-revolution orbit transfer which involves a large plane change and target values for all orbit elements save true anomaly. The optimisation problem is that of minimising propellant consumption for a number of different, but fixed, flight times. The optimisation is performed using the software program called "Mystic", written by Whiffen and based on his Static Dynamic Control algorithm.[4,5]

We assess the utility of using initial guesses from the Q-law by also performing optimisation runs based on simple, non-Q-law initial guesses. The run times needed for convergence and the nature of the optimal solutions found (as compared to each other and to the initial guess) are then

*Senior Member of the Engineering Staff, Jet Propulsion Laboratory, California Institute of Technology, Pasadena, CA 91109-8099, USA. Member AAS. Member AIAA. Email: Anastassios.Petropoulos@jpl.nasa.gov.

†Member of the Technical Staff, Jet Propulsion Laboratory, California Institute of Technology, Pasadena, CA 91109-8099, USA. Email: Seungwon.Lee@jpl.nasa.gov.

compared. A number of different parameters appear in the Q-law, which, while having nominal values, can be adjusted so as to reduce propellant mass. Thus, in addition to the initial guesses generated using the nominal Q-law, initial guesses based on the Q-law with adjusted parameters are also considered. In particular, the parameter values used here are taken from Ref. [6], where the same orbit transfer is considered and a genetic algorithm (GA) is used to find sets of parameter values which improve on the performance of the nominal Q-law solutions. The GA-optimised Q-law, or simply GA-Qlaw, can offer, for a given flight time, up to about 30% propellant savings over the nominal Q-law for the complex orbit transfer being considered here.

The motivating factor for trying to find good initial guesses is that optimisation problems such as the one considered here are difficult to solve due to the large number of revolutions: When indirect methods are used, the already great sensitivity of the solution to the initial value of the costates is further increased; when direct methods are used, the dimension of the search space is significantly enlarged. When "goodness" of an initial guess is measured based on comparing propellant mass and flight time against optimal solutions, the Q-law and GA-Qlaw perform remarkably well. The goal of the present work is to test the hypothesis that this sort of goodness translates into significantly reduced run times when optimisation runs are seeded with the Q-law and GA-Qlaw initial guesses.

METHOD

Definition of Orbit Transfer

The orbit transfer being considered is around the Earth and involves the initial and target orbit elements shown in Table 1, expressed in the reference frame of Earth Ecliptic and Equinox of J2000. Classical orbit elements are used, namely $(a, e, i, \Omega, \omega)$, for semimajor axis, eccentricity, inclination, longitude of the ascending node, argument of periapsis, respectively. The initial true anomaly is taken as zero and the final true anomaly is free, *i.e.,* it is a phase-free transfer. (Of course, since the transfer trajectories will involve many revolutions, the initial and final values for the true anomaly do not significantly affect the overall structure the transfer.) For simplicity, the Earth is modelled as a point mass, with gravitational parameter 398600.43 km^3/s^2. Again for simplicity, the gravity of the Moon is ignored. The only other body whose gravity is taken into account is the Sun, as required by Mystic's formulation of the dynamics. The initial epoch for the transfer is 20 May 2003. The motion of the Earth about the Sun is modelled using full ephemerides, but the Earth is placed at the location of the Earth-Moon barycentre, to remove the slight wobble in the Earth's Sun-centered motion which can create a multiplicity of local minima in the transfer problem. An initial spacecraft mass of 2000 kg is assumed, and the ion engine is assumed to deliver a constant specific impulse of 2000 s, and a maximum thrust of 2 N. It is worth noting that the reference frame and initial epoch assumptions are largely immaterial, given the simplifications made to the dynamics and the fact that the initial and target orbits are both relatively deep in the Earth's gravity well.

Table 1: Orbit Transfer Definition

	a (km)	e	i (deg.)	Ω (deg.)	ω (deg.)
Initial Orbit	24505.9	0.725	0.06	180	180
Target Orbit	26500	0.7	116	180	-90

Transfer trajectories are sought for a number of different, but fixed, flight times, ranging from nearly the minimum flight time, to almost twice the minimum flight time. This orbit transfer is very similar to Case E of Refs. 2,3 (there is a difference of 0.12° in the initial orbit plane), and identical to Case E of Refs. 6.

Optimisation Using Mystic

Mystic is based on the Static Dynamic Control algorithm, which best fits into the direct method category, although, unlike other direct methods, the explicit time dependence of the optimisation problem is not removed by parameterisation.[4,5] There are two main reasons for using Mystic as the optimiser. The first is that since it is based essentially on direct methods, it can be given an entire thrust profile as an initial guess, not just the state, thrust and derivatives at the initial time as would be done in the case of indirect methods. The second is that, although it can take very long to do so, Mystic converges very robustly from even very poor initial guesses, unlike many other optimisers.

Mystic divides the trajectory into a series of management periods, each of fixed duration. The optimisation variables are the inertial direction and magnitude of the thrust on each management period; both are held fixed for the duration of the management period. The cost function is essentially the propellant mass, although the constraint of being at the target orbit at the final time is enforced by means of penalty functions. The initial guess to Mystic is simply a series of thrust vectors in inertial space, one for each management period. The management period duration was fixed at 0.025 days for all cases. For the most part, this is sufficiently brief to give a reasonable distribution of management periods over each revolution of the transfer.

A number of parameters control the behaviour of Mystic's optimisation algorithm. These parameters were typically left at the same standard settings for all optimisation runs. One exception is for the trust region size (or, more precisely, the upper limit on the trust region size). The concept of a trust region is common to many optimisers; it is essentially a region where a quadratic approximation of the cost function in terms of the optimisation variables can be trusted to be sufficiently good with respect to the full, non-linear cost function. The optimiser is not permitted to take a step which is larger than the trust region. Thus, while an optimiser will estimate the size of a trust region, it is useful to set an upper limit on the size to prevent the optimiser from taking too large a step. Mystic measures the trust region in units of force.

Optimisation runs using the standard settings were performed for all initial guesses. Some runs which did not converge or converged very slowly were rerun with variant settings. These variants are made clear in the Results section. In particular, several additional runs were made for some of the initial guesses using a four-tiered scheme for the trust region sizes. The tiers were demarcated by dividing the number of management periods into four roughly equal, consecutive parts. The first tier, that is, roughly the first quarter of the trajectory in terms of time, had the standard trust region size. Each subsequent tier had double the trust region size of the previous tier. The rationale behind this tiered scheme is that the final state of the spacecraft is much more sensitive to changes in thrust at the beginning of the transfer than at the end of the transfer. Passing on this knowledge to the optimiser by imposing smaller upper limits on the trust region size nearer to the beginning of the trajectory may help the optimiser converge more quickly.

Generating Q-Law Initial Guesses

The Q-law, being a feedback algorithm, will return a thrust vector given the current spacecraft state. Thus, to generate an initial guess, the initial spacecraft state is propagated forward in time using the Q-law to provide the thrust history. However, because of Mystic's management period structure, the Q-law is called only at the beginning of every management period. The Q-law will estimate the state at the temporal midpoint of the management period, and use this estimated state to return a thrust vector which will be used as the inertial thrust direction for the entire management period. Another factor worth noting is that although the Q-law is based on two-body dynamics, because it is a feedback algorithm, it can still function well as a control law when perturbations are present to the two-body dynamics, as is the case here. The magnitude of the thrust vector returned by the Q-law will always be either zero (when coasting) or the maximum thrust level.

The behaviour of the Q-law is controlled by the target orbit elements and by a number of weights and other parameters as explained in Ref. 3. Using the notation of Ref. 3, these parameters and their nominal values (where applicable) are $W_{œ} = 1$, where $œ = (a, e, i, \omega, \Omega)$, $m = 3$, $n = 4$, $r = 2$, $b = 0$, η_{acut}, $W_P = 1$, $k = 100$, $r_{p\text{min}}$. The latter three parameters are for enforcing a periapsis radius constraint. For this transfer, the nominal value of $r_{p\text{min}}$, the minimum permitted periapsis radius, is taken to be 6578 km. The absolute effectivity cut-off, η_{acut}, does not have a nominal value; instead it is adjusted manually until the desired flight time is obtained, as this parameter controls the amount of coasting on the transfer: Zero means never coast, unity means always coast. The related parameter, η_{rcut}, the relative effectivity cut-off, is ignored in this transfer.

The term "Q-law initial guess" will refer to initial guesses generated using either the nominal Q-law parameters and target orbit elements, or, in two cases, a small number of intuitively-made variations from the nominal values. These variations will be made clear in the Results section.

Generating GA-Qlaw Initial Guesses

As mentioned in the Introduction, the parameters of the Q-law are amenable to optimisation, in particular, by means of genetic algorithms. In Ref. 6, a GA is described and results are presented for the orbit transfer being considered here, with the only differences being that 1) the Sun's gravity is ignored, and 2) a continuously varying thrust direction is permitted, that is, there is no management period structure. Ref. 6 presents a Pareto front (trade-off curve) in flight time and propellant mass, based on varying the Q-law parameters using the GA.

To generate "GA-Qlaw initial guesses" here, we take the following steps: 1) use the Q-law parameters for the trajectories on the Pareto front that are closest to the flight times of interest here, and 2) adjust η_{acut} until the target orbit is met (this is needed to compensate for the two differences mentioned above.)

One additional GA-Qlaw initial guess is generated for the near-minimum-flight-time case using the same methods as in Ref. 6, but where a solution is sought having not only the shortest possible flight time, but also a "reasonable" number of revolutions. The rationale for this will become clearer in the Results section, but is essentially that the number of revolutions is an important attribute of the minimum-flight-time case and that Mystic will not easily add or subtract revolutions from the initial guess since a local minimum will normally be found before a full revolution is added or removed in the course of the optimisation.

Generating Simple Initial Guesses

Ideally, for a fair comparison, the simple initial guess should not use any knowledge of the optimal solution, just as the Q-law and GA-Qlaw guesses use no knowledge of the optimal solution. However, to avoid potentially prohibitively long run times in optimising such guesses, an advantage will be given to the simple initial guesses. Two attributes of the initial guess that assist the optimisation process in Mystic, particularly for many-revolution transfers, are: 1) The "right" number of revolutions (*i.e.,* a locally optimal solution with that number of revolutions should exist), and 2) A "reasonable" semimajor axis profile (*i.e.,* a locally optimal solution with roughly the same profile should exist). The first attribute is particularly important.

Thus, the simple initial guesses used here were designed to have the same number of revolutions as the optimal solutions found using the Q-law initial guesses (but GA-Qlaw guess, for the 64.9-day case), and a roughly similar semimajor axis profile. (As expected from intuition, the semimajor axis profile has a roughly trapezoidal shape since the large plane change demands that the orbit be enlarged to perform the plane change and overall transfer more efficiently.) The guesses were generated by dividing the trajectory into three parts. In the first part, thrust of magnitude equal to some fraction of the maximum thrust is applied along the velocity vector. The second part is a coast period, and the third part uses thrust against the velocity vector, again at some fraction of the maximum thrust. The profiles of the other orbit elements are ignored in generating the simple initial guesses.

Finding the Minimum Flight Time

Mystic is not designed to minimise the time of flight of a trajectory. Thus, in order to find the minimum flight time, an iterative procedure must be followed, where Mystic repeatedly searches for propellant-optimal solutions with different, fixed, flight times until the minimum flight time that still yields a feasible trajectory (one that meets the target orbit) is found. This iterative process was seeded using the minimum flight time found by the GA-Qlaw in Ref. 6 (60.3 days), but with a Q-law initial guess. The minimum flight time found with the iterative procedure was 59.7 days, of which less than about 0.5 days involved coasting. That is, given that minimum flight time trajectories generally have continuous thrust, the minimum flight time for this transfer must be only slightly below 59.7 days.

Cases Considered and Computers Used

Propellant-optimal transfers are found for each of the following flight times in days: 59.7, 64.9, 81, 82, 95, 105. These flight times are designated 1 through 6, respectively. For each flight time, save the 81 day, optimisation runs were performed using all three types of initial guess: Simple, Q-law, and GA-Qlaw, designated S, Q, and G, respectively. For the 81-day case, only a Q-law initial guess was used.

The optimisation was performed on two different types of computers. For the 59.7-day case, Mystic was run on a Sun workstation of type SunBlade 1000, with a 900 MHz-clock-speed UltraSPARC central processing unit (CPU). The remaining cases were run on individual nodes of a cluster of linux machines with Intel Xeon chips having a clock speed of 3.06 GHz, a front-side bus speed of 533 MHz, a 512 kB L2 cache, and a 1Mb L3 cache. The relatively small L2 and L3 sizes are not of great importance for this particular problem. For each of the runs, the Mystic process was typically the only CPU-intensive job running on the CPU. The CPU time required for each run was recorded. To measure the relative speeds of the machines, one run was done on both machines: the linux nodes were found to be about 39% faster than the Sun workstation.

RESULTS

The various initial guess trajectories and the corresponding trajectories found by Mystic will be denoted using trajectory identifiers such as 2Q1 and 2Q1o1. The first character (2 here) designates the flight time (64.9 days here); the second character (Q here) designates the initial guess type (Q-law here); the third character designates different versions of the given type of initial guess (1 for nominal, 2 for the alternative version); the fourth character, if present, designates a trajectory obtained by Mystic based on the given initial guess (o means optimal, f means run completed but failed to converge, t means optimisation run was terminated due to excessive run time); the fifth character, if present, designates different settings used in the optimisation run (1 means nominal settings, 2 mean the alternative settings). (Nominal initial guess versions and optimisation settings are described in the Method section above.) In other words, three-character designators, such as 2Q1, indicate initial guesses, while five-character designators, such as 2Q1o1, indicate trajectories

Table 2: Initial Guesses and Corresponding Trajectories from Optimiser

Traj. ID	Residuals[1] a (km)	e	i (deg.)	Ω (deg.)	ω (deg.)	Revs	m_{prop} (kg)	CPU (hrs.)	Num. Iters.
59.7-Day Trajectories									
1Q1	7546.2	0.0537	-21.733	16.861	22.056	80.53	526.0		
1Q1o1	**2.0**	**-0.0047**	**-0.008**	**0.003**	**0.015**	**81.81**	**520.9**	**89.23**[a]	**38925**
1Q2	10935.9	0.1135	13.063	42.174	31.430	76.27	455.1		
1Q2o1	**2.4**	**-0.0043**	**-0.007**	**0.002**	**0.014**	**78.06**	**517.6**	**80.67**[a]	**35230**
1G1	1032.6	-0.0797	-0.004	1.717	3.764	88.93	526.0		
1G1f2	7.8	-0.0279	-0.050	0.017	0.096	88.61	526.0	45.98[a,b]	20927
1G2	1635.4	-0.0294	-0.004	0.465	1.933	72.52	526.0		
1G2o1	**3.0**	**-0.0044**	**-0.007**	**0.002**	**0.014**	**73.14**	**523.1**	**4.93**[a]	**2669**
1S1	65.0	-0.0059	-115.940	0.138	89.372	80.18	237.7		
1S1t1	403.1	0.1910	-57.111	12.821	76.886	80.20	309.1	121.20[a]	41778
1S1o2	**2.0**	**-0.0045**	**-0.007**	**0.002**	**0.015**	**79.81**	**520.7**	**71.17**[a,b]	**30693**
64.9-Day Trajectories									
2Q1	7317.0	0.0572	-19.430	17.599	19.660	87.12	571.8		
2Q1o1	**2.0**	**-0.0022**	**-0.004**	**0.000**	**0.007**	**88.26**	**503.8**	**19.50**	**11235**
2G1	228.4	-0.0050	-0.007	0.034	0.727	81.39	513.2		
2G1o1	**0.8**	**-0.0019**	**-0.003**	**0.000**	**0.007**	**81.37**	**488.2**	**6.21**	**3705**
2S2	2048.3	-0.0169	-115.940	0.162	90.709	81.23	248.1		
2S2o1	**0.8**	**-0.0019**	**-0.003**	**0.000**	**0.007**	**81.56**	**488.1**	**30.42**	**18924**
81-Day Trajectories									
3Q2	99.6	0.0006	-0.018	0.035	0.027	113.39	695.4		
3Q2o1	**0.8**	**-0.0016**	**-0.003**	**0.001**	**0.006**	**113.49**	**475.3**	**20.50**	**9990**
82-Day Trajectories									
4Q1	998.2	0.0009	-0.015	0.031	0.118	115.01	720.5		
4Q1o1	**0.4**	**-0.0016**	**-0.003**	**0.000**	**0.002**	**115.17**	**463.4**	**30.49**	**13650**
4G1	243.1	-0.0015	-0.007	0.045	0.515	89.49	491.2		
4G1o1	**1.0**	**-0.0012**	**-0.002**	**0.000**	**0.006**	**89.42**	**441.1**	**19.88**	**10519**
4S1	1992.7	-0.0135	-115.940	0.154	90.382	115.18	159.6		
4S1o1	**1.3**	**-0.0015**	**-0.003**	**0.001**	**0.007**	**115.59**	**455.9**	**233.78**	**104351**
4S1o2	**0.6**	**-0.0016**	**-0.003**	**0.001**	**0.004**	**115.52**	**455.2**	**63.25**	**28699**
95-Day Trajectories									
5Q1	13.5	-0.0012	0.010	0.000	0.041	132.33	686.1		
5Q1o1	**0.6**	**-0.0014**	**-0.002**	**0.001**	**0.006**	**132.27**	**444.6**	**36.98**	**14455**
5G1	-21.7	0.0018	-0.000	0.033	0.013	99.41	488.3		
5G1o1	**0.9**	**-0.0011**	**-0.002**	**0.000**	**0.002**	**99.36**	**420.6**	**32.75**	**15854**
5S1	-234.8	0.0033	-115.940	0.176	89.881	132.19	191.5		
5S1o1	**0.1**	**-0.0013**	**-0.003**	**0.001**	**0.006**	**132.26**	**441.7**	**294.90**	**121959**
5S1o2	**-7.1**	**-0.0020**	**0.001**	**0.014**	**0.047**	**132.27**	**441.7**	**85.70**	**34762**
105-Day Trajectories									
6Q1	41.5	-0.0010	-0.007	0.038	0.049	146.18	663.0		
6Q1o1	**68.9**	**0.0016**	**0.007**	**0.055**	**0.211**	**146.10**	**434.7**	**33.68**	**12012**
6G1	294.9	-0.0007	-0.008	0.008	0.098	109.53	502.2		
6G1o1	**6.0**	**-0.0007**	**-0.001**	**0.004**	**0.016**	**109.28**	**406.8**	**112.57**	**49112**
6G1o2	**-3.5**	**-0.0011**	**-0.002**	**0.003**	**0.005**	**109.31**	**407.2**	**92.53**	**43119**
6S1	1961.6	-0.0160	-115.940	0.175	90.326	146.09	162.4		
6S1t1	1883.0	0.2310	-53.093	13.017	80.419	146.13	222.3	252.36	99000
6S1o2	**65.9**	**0.0019**	**0.000**	**0.069**	**0.270**	**146.51**	**427.5**	**83.21**	**30168**

[1]Residual = $æ_f - æ_T$ (for $æ = a, e, i$) or $\cos^{-1}[\cos(æ_f - æ_T)]$ (for $æ = \Omega, \omega$). $(æ_f, æ_T)$ =(final,target) values for $æ$.
[a]Run time is for UltraSPARC chip. Run time on Xeon chip would be about 39% faster.
[b]Run stopped and resumed multiple times with new optimisation settings.

Table 3: Q-law Parameters Used in GA-Qlaw Initial Guesses

Traj. ID	1G1	1G2	2G1	4G1	5G1	6G1
W_a	15.9	11.5	12.9	8.5	15.7	9.5
W_e	3.3	2.0	2.6	1.4	3.5	1.4
W_i	69.5	69.9	50.2	61.6	61.4	55.9
W_Ω	6.5	11.6	30.9	26.2	14.3	30.0
W_ω	4.8	5.0	3.4	2.3	5.1	3.3
η_{acut}	0	0	0.29	0.26	0.29	0.31
m	3.07	3.34	4.94	8.65	10.87	10.14
n	6.70	6.52	5.99	5.16	4.66	5.16
r	2.05	4.78	2.96	3.34	3.62	3.34
W_p	0.118	0.001	0.296	0.226	0.317	0.23
$r_{p\text{min}}$	6646.7	7017.0	6649.7	6603.9	6602.2	6642.5
k	205.3	52.3	160.6	205.3	393.0	890.3

Table 4: Variant Initial Guesses and Optimisation Settings

Traj. ID	Variation from nominal
1Q2	Q-law target inclination set to 179°, to keep semimajor axis high for a longer time. Uses relative effectivity cutoff of 0.05.
1G1f2	Optimiser needed greater damping initially. Damping manually lowered part way through run. Trust region upper limit also lowered manually after damping lowered.
1G2	Fewer revolutions than 1G1.
1S1o2	Optimiser needed lower target penalty weight at first. Increased manually part way through run.
2S2	Semimajor axis profile based on 2G1o1, not 2Q1o1.
3Q2	Artificially raised the minimum periapsis radius constraint by resetting $r_{p\text{min}}$ and k, to try find different class of local minima.
4S1o2	Tiered scheme for trust regions
5S1o2	Tiered scheme for trust regions
6G1o2	Tiered scheme for trust regions
6S1o2	Tiered scheme for trust regions

computed by the optimiser starting with the initial guess indicated by the first three characters of the designator.

Summary data for the initial guess trajectories and the trajectories found by Mystic are presented in Table 2 which shows the the trajectory identifier, the residuals in meeting the target orbit elements, the number of revolutions performed by the spacecraft around the Earth, the propellant mass expended, the CPU time needed for the Mystic optimisation run, and the number of iterations performed by Mystic. Trajectories are grouped by flight time, and within each flight time, each type of initial guess is listed above the corresponding trajectory or trajectories found by Mystic. Data for converged optimistation runs that used nominal settings are in normal-size bold font, while converged runs with variant optimisation settings are shown in small bold font. Initial guesses and unconverged optimisation runs are shown in normal font.

The absolute effectivity cutoff values used in the Q-law initial guesses were zero for flight times of 82 days or less, 0.24 for the 95-day flight time, and 0.3 for the 105-day flight time. The parameters used for the GA-Qlaw initial guesses are shown in Table 3. Those initial guesses and optimisation runs which used non-nominal or non-standard values for the parameters are listed in Table 4 along with a description of the changes in the parameters.

Trajectory plots are shown in Figs. 1-4 for the 59.7-day case 1Q2o1 (optimal, Q-law derived), the 105-day case 6Q1o1 (optimal, Q-law derived), the 105-day case 6G1 (GA-Qlaw initial guess), and the optimal 105-day case 6G1o1 derived therefrom. All the plots are at the same scale and have the same vantage point. The Earth's sphere is shown in blue, and thrust vectors appear in red. For each of the flight times considered, plots of the time histories of several orbit elements are made for most of the initial guesses and optimal solutions found (see Figs. 5-10). In addition, a plot of propellant mass versus flight time is shown in Fig. 11 for the optimal solutions found, for the Q-law and GA-Qlaw initial guesses, and for the GA-Qlaw data of Ref. [6] (in which, we recall, the management period structure is absent and a continuously varying thrust direction is permitted). Propellant mass is not shown for the simple initial guesses because the residuals for these guesses are very large, and so there is no purpose in comparing the mass to the optimal one. All of the Mystic run times listed in Table 2 are plotted in Fig. 12 to better illustrate the effects of the different initial guess types and optimisation settings.

DISCUSSION OF THE RESULTS

Qualitative Features of the Trajectories

One basic question that can be immediately answered is whether the various initial guesses meet the target orbit. This information is contained in Table 2 and partly in Figs. 5-10. As expected, none of the simple initial guesses reach the target orbit. They do roughly match the target semimajor axis and eccentricity, but the residuals in all the other elements, particularly inclination, are large. The Q-law initial guesses reach the target orbit for flight times of 81 days and higher. For flight times below 81 days, even though continuous thrust is used, the Q-law initial guess falls a little short of the target orbit. The GA-Qlaw initial guesses reach the target orbit (within a reasonable tolerance) for all the flight times considered, including the near-minimum flight time of 59.7 days.

Immediately apparent in comparing the initial guess versus the optimal solution time histories of the orbit elements is that the semimajor axis profile of the optimal solution is always very close to that of the initial guess (Figs. 5-10). This is in large part due to the fact that there is typically at least one local minimum associated with a given number of revolutions. So the optimiser will normally find the locally optimal solution associated with the number of revolutions present in the initial guess, or a very close number of revolutions. Since the semimajor axis profile is the sole determinant of the number of revolutions, it will not change significantly in the optimal solution.

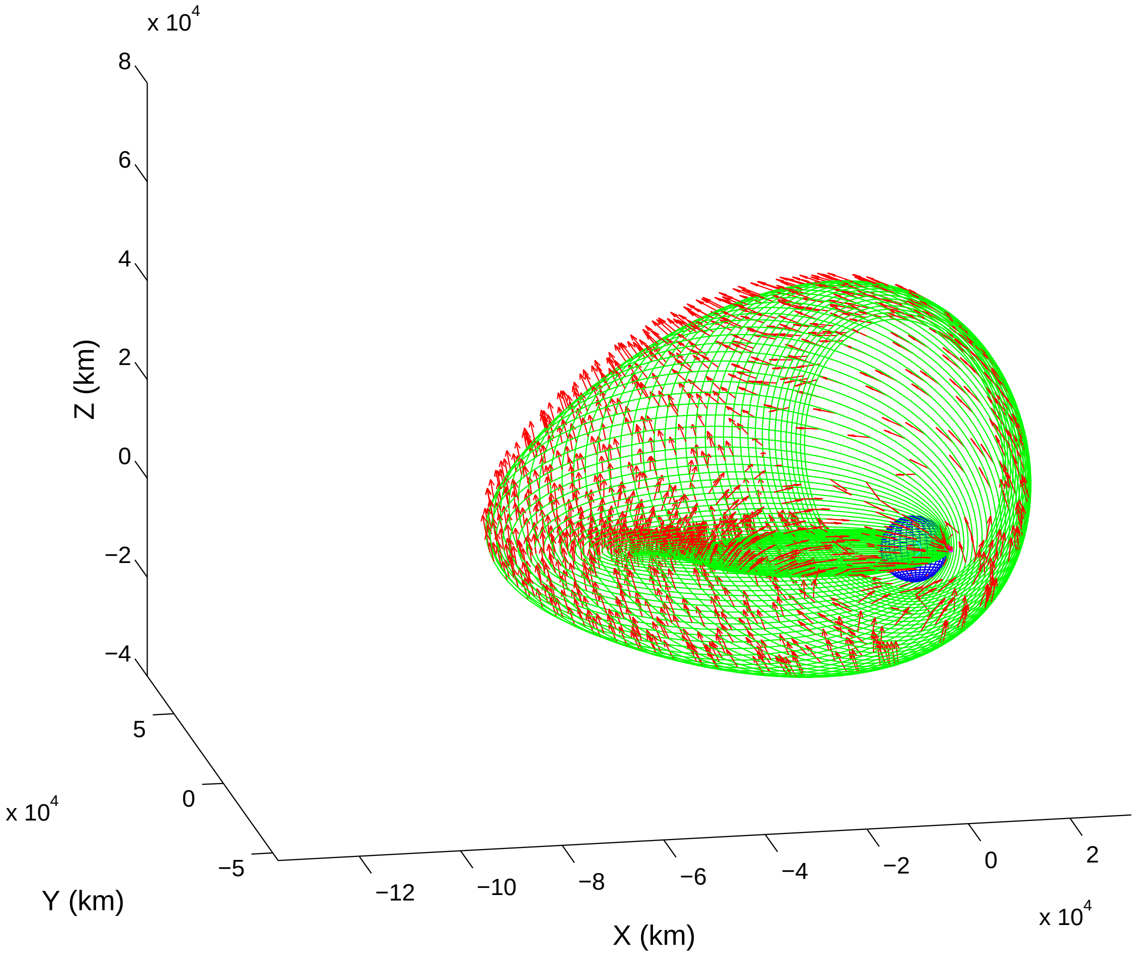

Figure 1 Trajectory plot for the 59.7-day case 1Q2o1: Optimal trajectory derived from Q-law guess 1Q2.

The scenario of the same number of revolutions being produced by a different semimajor axis profile is discounted, because, in searching for such a profile, the optimiser would likely have to change the number of revolutions during intermediate steps — but it will not do so because of the attraction to the local minimum involving the initial number of revolutions and semimajor axis profile.

Another obvious feature of the semimajor axis profiles is that the semimajor axis grows quite large, thereby permitting a more efficient plane change, and then shrinks back to the target value. The efficiency gained for the plane change outweighs the cost of overshooting the target semimajor axis size. Different degrees of overshooting provide different local minima, as is particularly apparent by comparing the Q-law-derived solutions with the GA-Qlaw-derived solutions. The larger overshoots of the GA-Qlaw solutions (see Figs. 5-10) provide a better local minimum (see Table 2 or Fig. 11).

Figure 11 shows, as expected, that the GA-Qlaw-derived solutions have the lowest propellant mass for a given flight time: lower than the Q-law derived solutions by about 0% at 59.7 days to about 6.4% at 105 days. The solutions derived from the simple initial guesses have performance very similar to the optimal solutions on which their semimajor axis profile was modelled: The GA-Qlaw-derived solution for the 64.9-day flight time, and the Q-law-derived solutions for the

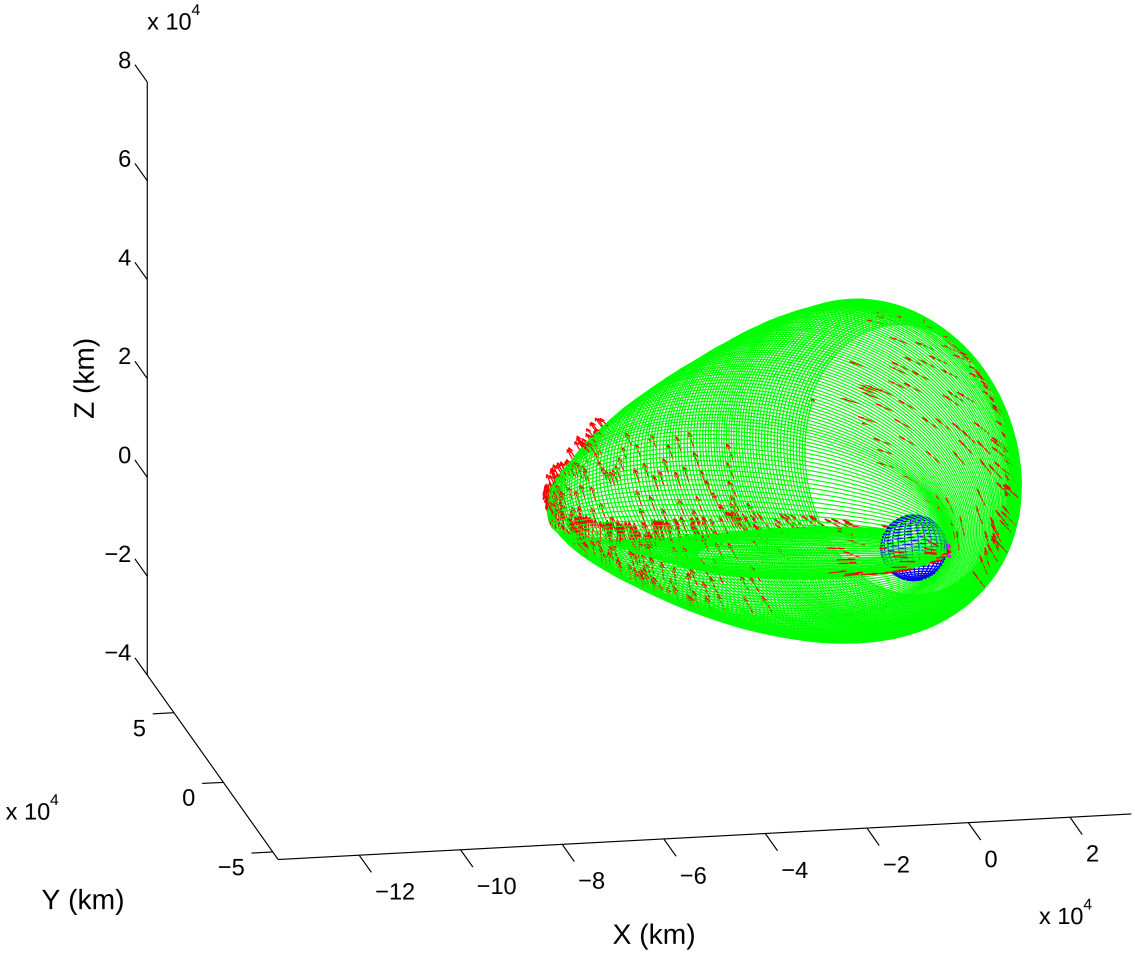

Figure 2 Trajectory plot for 105-day case 6Q1o1: Optimal trajectory derived from Q-law guess 6Q1.

other flight times. Again as expected, for those initial guesses that reach the target orbit (within a reasonable tolerance), the difference in propellant mass for the initial guess and the optimal solution is smaller in the case of the GA-Qlaw guesses than in the case of the Q-law guesses.

An unexpected feature does arise in Fig. 11, however. The GA-Qlaw guess performs increasingly worse than the corresponding optimal solution, as the flight time increases. One possible explanation for this is that the number of management periods increases with the flight time. The Q-law feedback algorithm is designed to take advantage of a continuously variable thrust direction, and so inefficiencies are introduced when a management period structure is imposed. These inefficiencies are compounded as the trajectory progresses, meaning that longer trajectories will see a greater performance degradation than shorter ones. The management period structure used here, results, for example, in trajectory arcs of about 120° spanning a single management period during some periapsis passages. This rather large arc will cause the Q-law performance to degrade, since the Q-law is not expecting to hold the thrust direction inertially fixed for such a large arc. The optimiser, in contrast, is able to use the inertially fixed direction much more effectively (for example, an inertially fixed thrust direction is well suited to changing eccentricity). Of note is that the GA-Qlaw solutions for continuously varying thrust direction are only between about 1% and

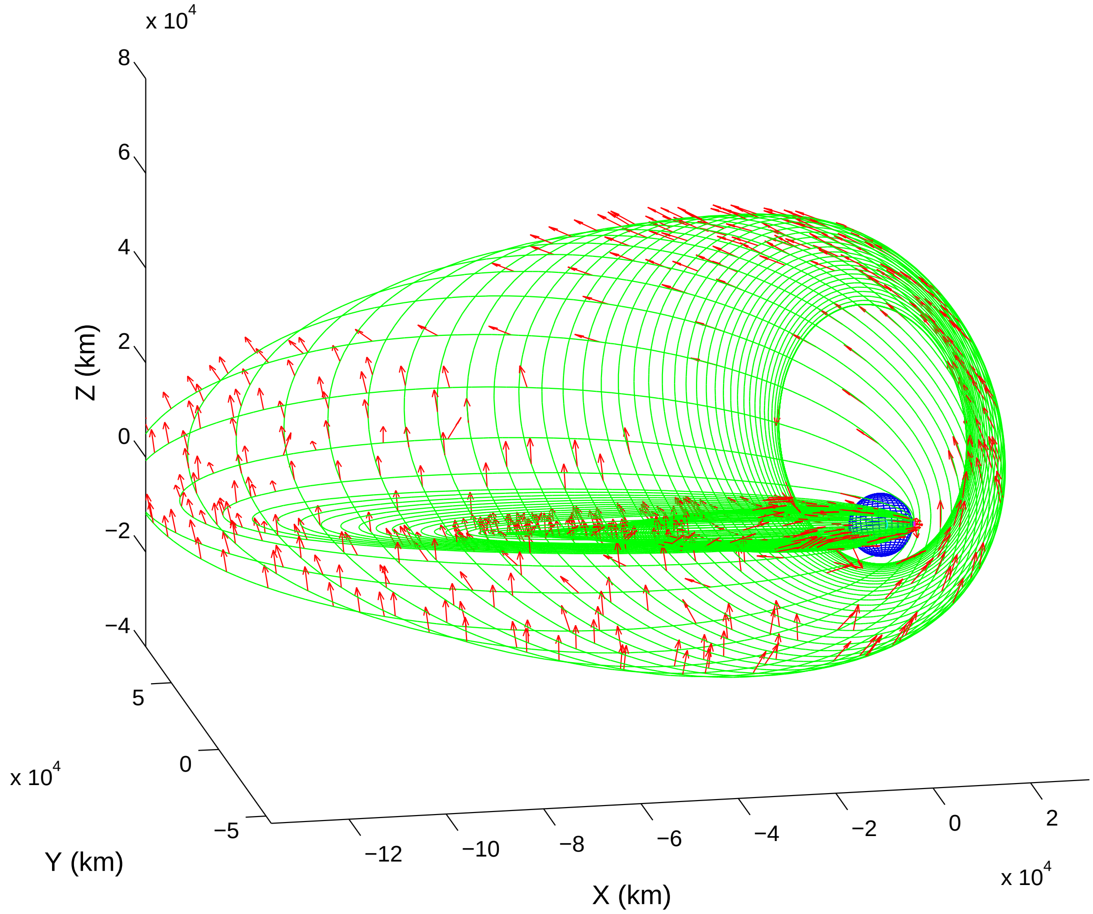

Figure 3 Trajectory plot for 105-day case 6G1: GA-Qlaw initial guess trajectory (meets target).

8% worse than the optimal, GA-Qlaw-derived solutions found here, and only up to 2% worse than the optimal, Q-law-derived solutions found here. (Ref. [6] shows that for simpler orbit transfers, propellant consumption of GA-Qlaw trajectories lies within about 0.5% of that for optimal ones.)

0.1 CPU Run Times: The Role of Initial Guesses and Optimisation Settings

Mystic is an exceptionally robust optimiser. However, it was not designed to handle trajecotries of more than about 100 revolutions — the run time simply becomes too large in most cases. Here we have demonstrated that the choice of initial guess, primarily, and the choice of optimisation settings, secondarily, can greatly impact the CPU run time. Of note is that Mystic converged for almost all the trajectories of Table 2, even for most of the ones based on the simple initial guesses (high residuals).

Figure 12 shows that at the near-minimum-flight-time case of 59.7 days, an initial guess with the right number of revolutions and semimajor axis profile and with low residuals drastically reduces the CPU run time: When comparing runs with the same optimisation settings, the reduction is by a factor of between 16 and well upwards of 25 (see cases 1Q2o1, 1G2o1, and 1S1t1 in Table 2). For flight times of 64.7 days and higher, the simple initial guesses with nominal optimisation settings

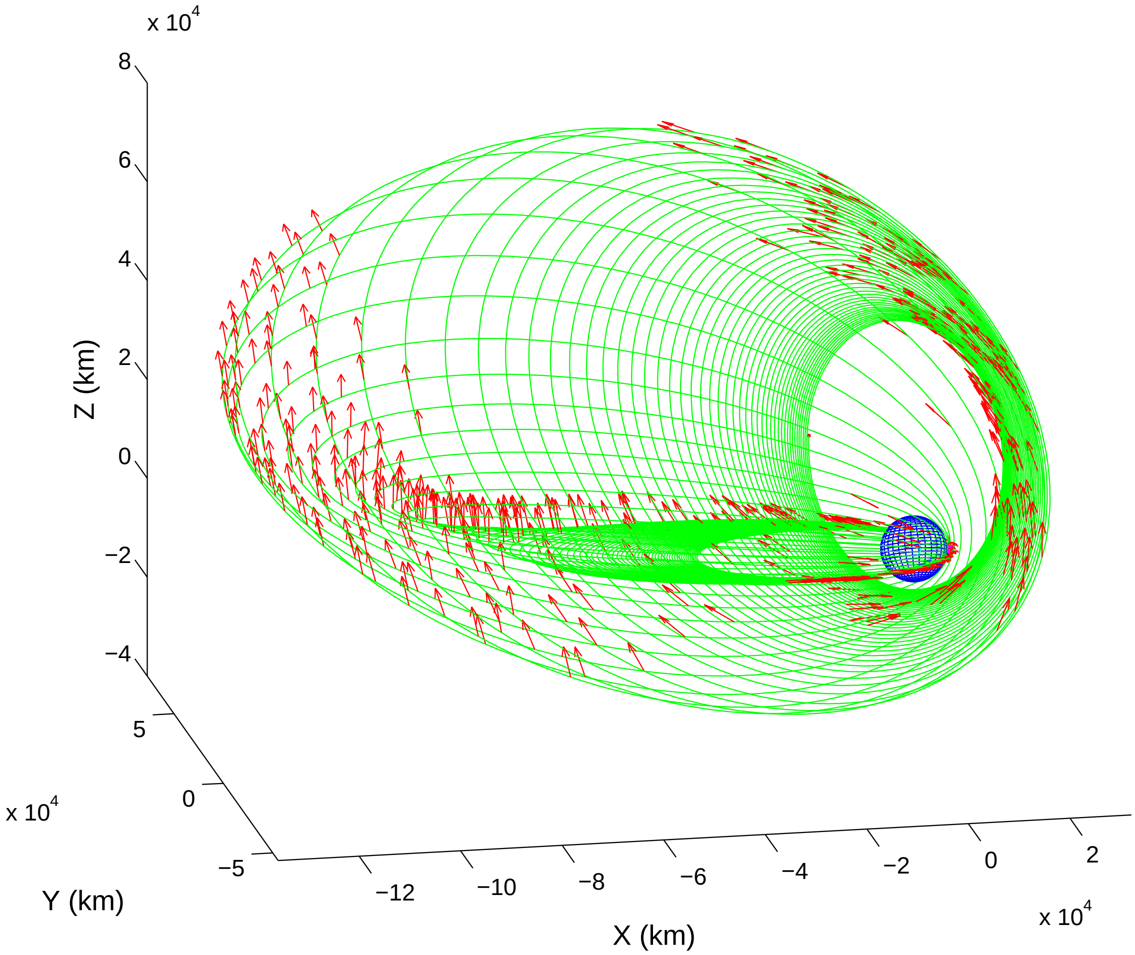

Figure 4 Trajectory plot for 105-day case 6G1o1: Optimal trajectory derived from the GA-Qlaw guess 6G1, shown in Fig. 3.

take about 5 to 12 times longer to run than the Q-law or GA-Qlaw guesses. As expected, the run times for the GA-Qlaw guesses were generally shortest, followed by the run times for the Q-law guesses. One unexpected feature is that the run time for the Q-law initial guesses is relatively flat, while the run time for the GA-Qlaw guesses rises steadily and shows a large upward spike for the 105-day case. A possible explanation for this spike in run time lies again in the management period structure. The GA-Qlaw initial guess, due to its far fewer revolutions, has, on average, many more management periods per revolution than the other initial guesses. This provides the optimiser much finer control over the time-history of the orbit elements, making the trajectory harder to optimise. This difficulty, added to the increased overall number of management periods, causes the run time to shoot upwards.

The role of the optimisation settings is also clearly illustrated in Fig. 12. Altering these settings seems to help primarily cases with very large residuals: For example, the tiered trust region scheme provided speed-ups by factors of between 3.7 and 3.4 for the simple-guess cases 5S1 and 6S1, but by a factor of only 1.2 for the GA-Qlaw case 6G1. Other optimisation settings can help runs that are proceeding prohibitievly slowly, typically those with large residuals, such as case 1S1.

Lastly, we note that we have not included the run times for the GA-Qlaw itself, that is, running

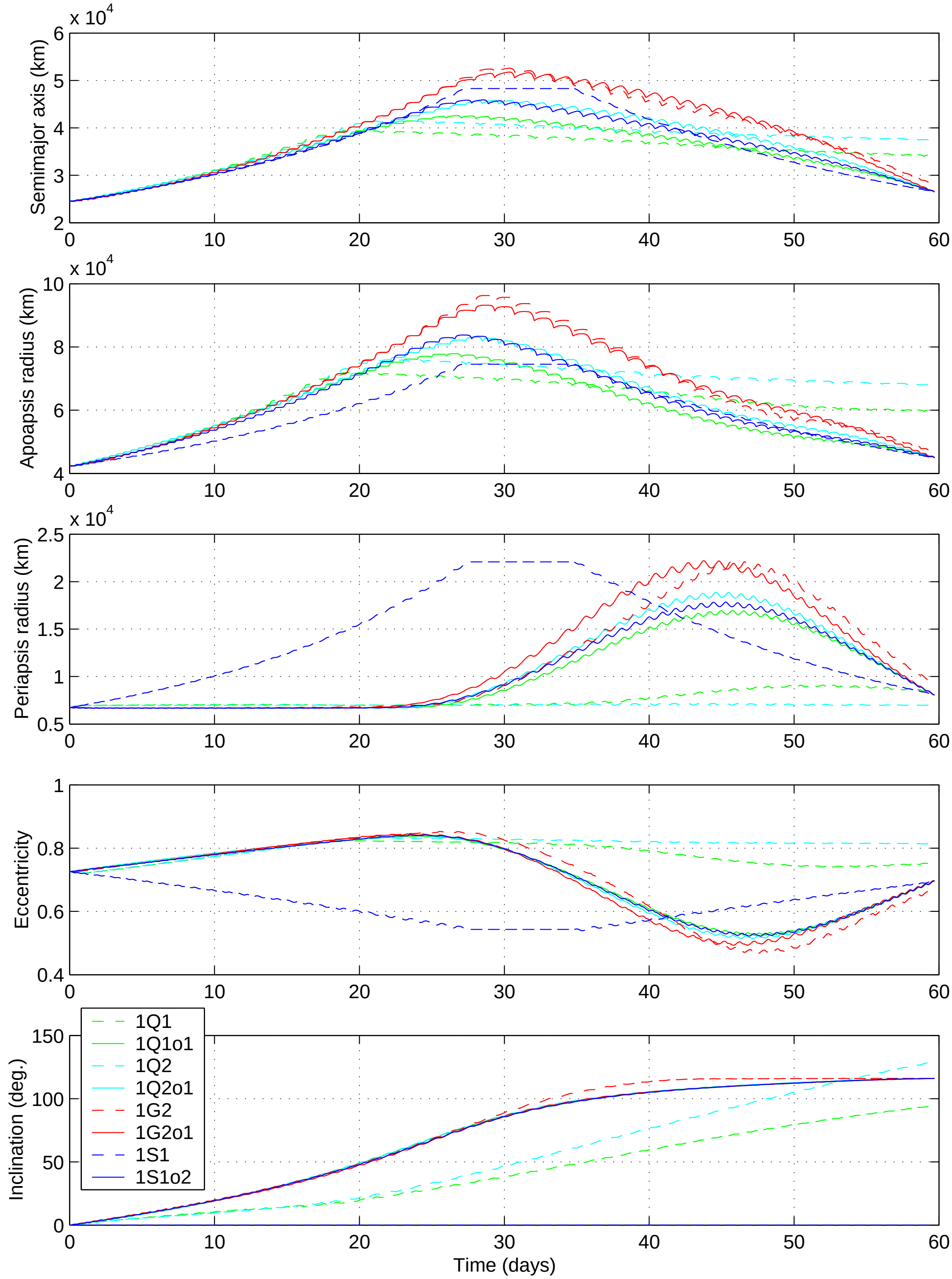

Figure 5 Orbit element time history for initial guesses (dashed lines) and the corresponding optimal solutions (solid lines) for the 59.7-day case.

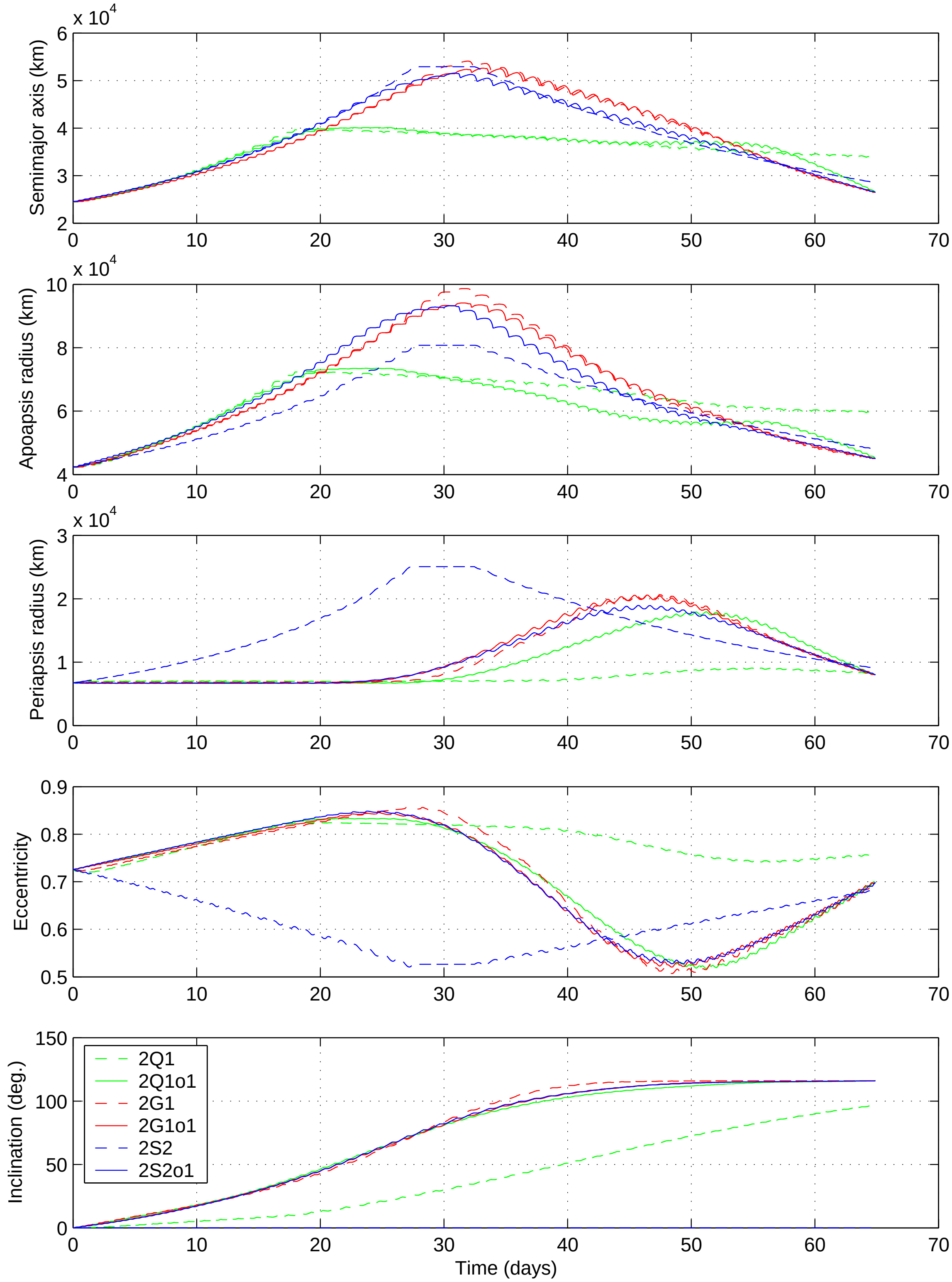

Figure 6 Orbit element time history for initial guesses (dashed lines) and the corresponding optimal solutions (solid lines) for the 64.9-day case.

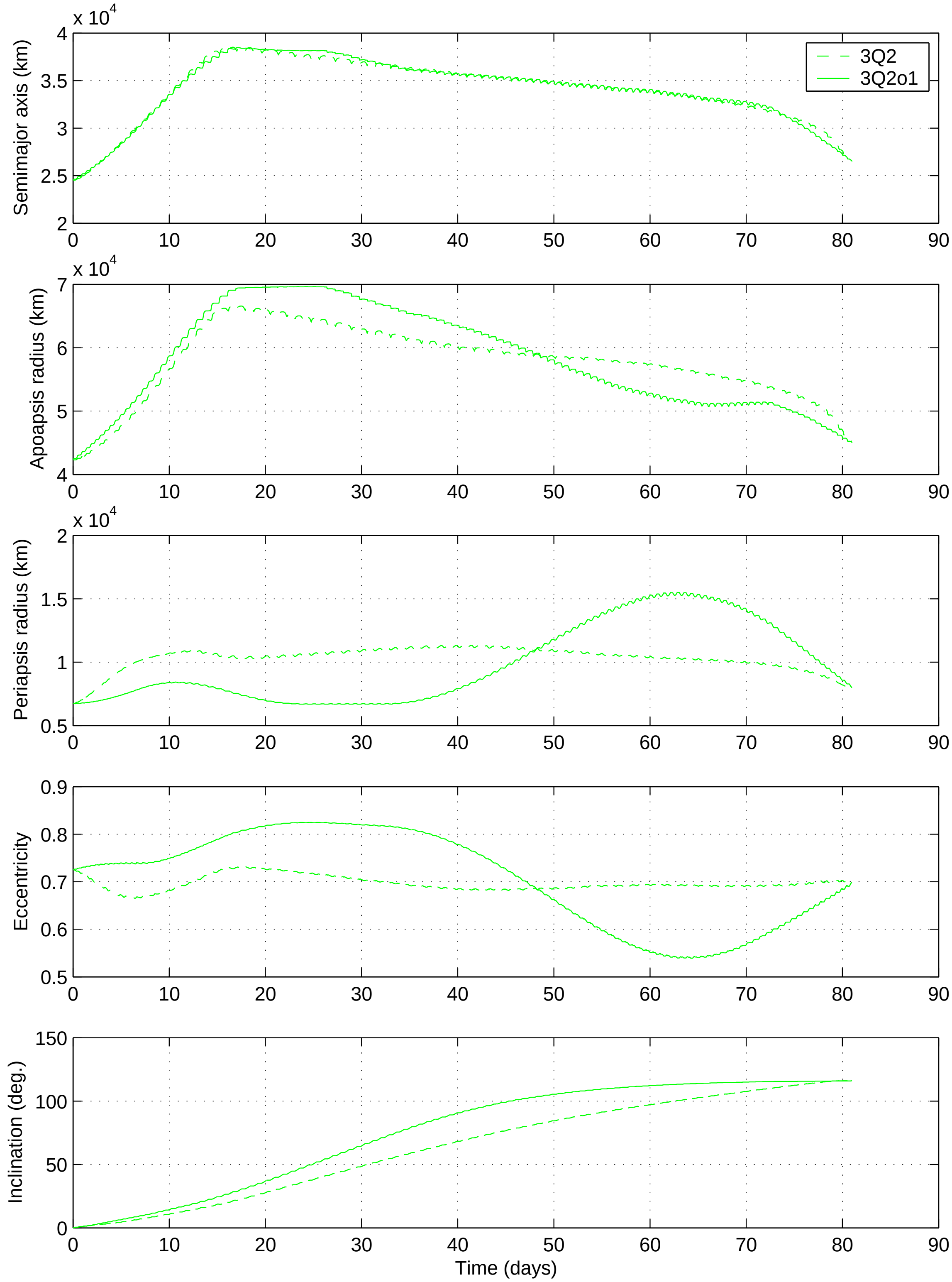

Figure 7 Orbit element time history for a variant Q-law initial guess (dashed line) and the corresponding optimal solution (solid line) for the 81-day case.

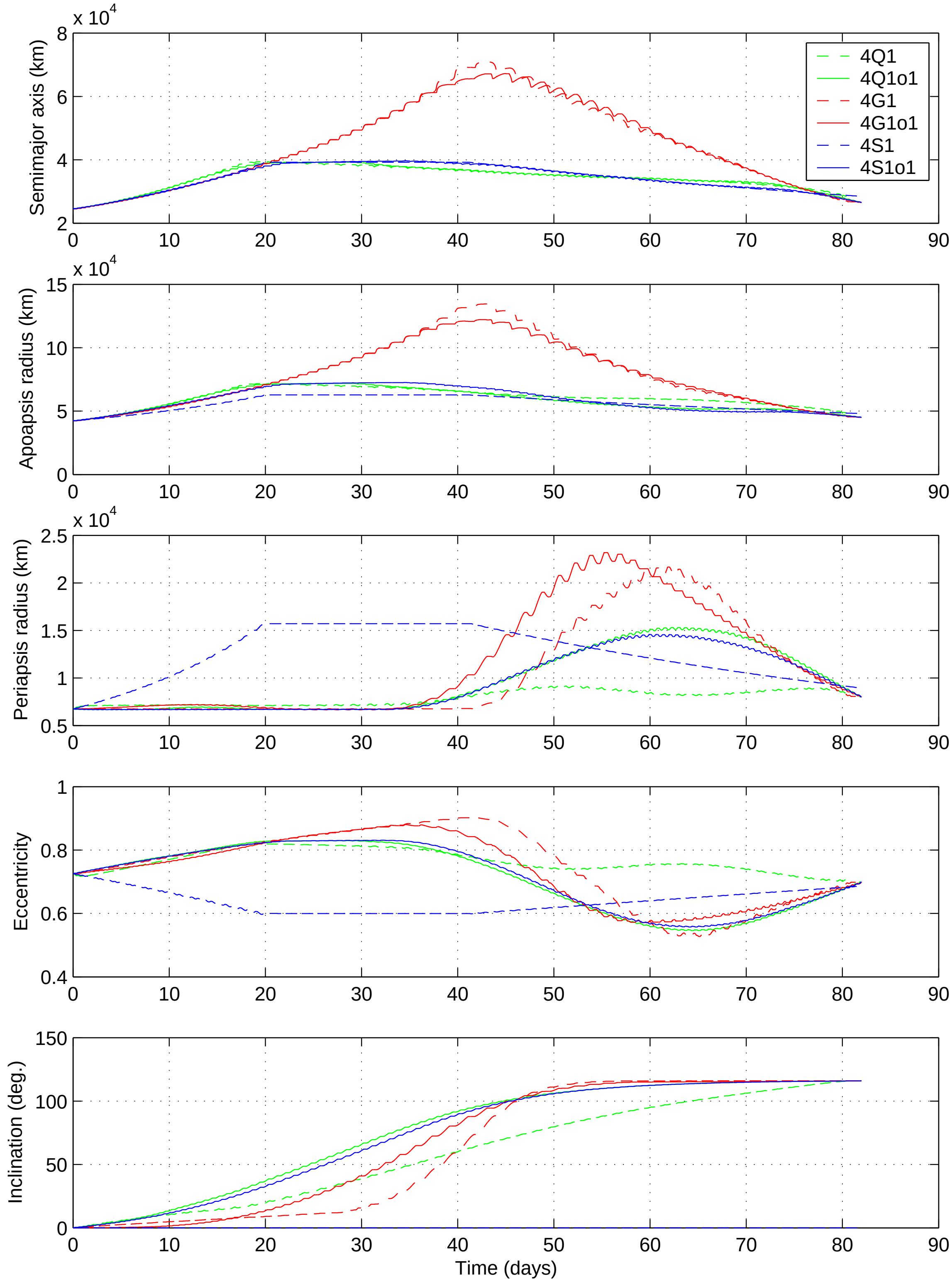

Figure 8 Orbit element time history for initial guesses (dashed lines) and the corresponding optimal solutions (solid lines) for the 82-day case.

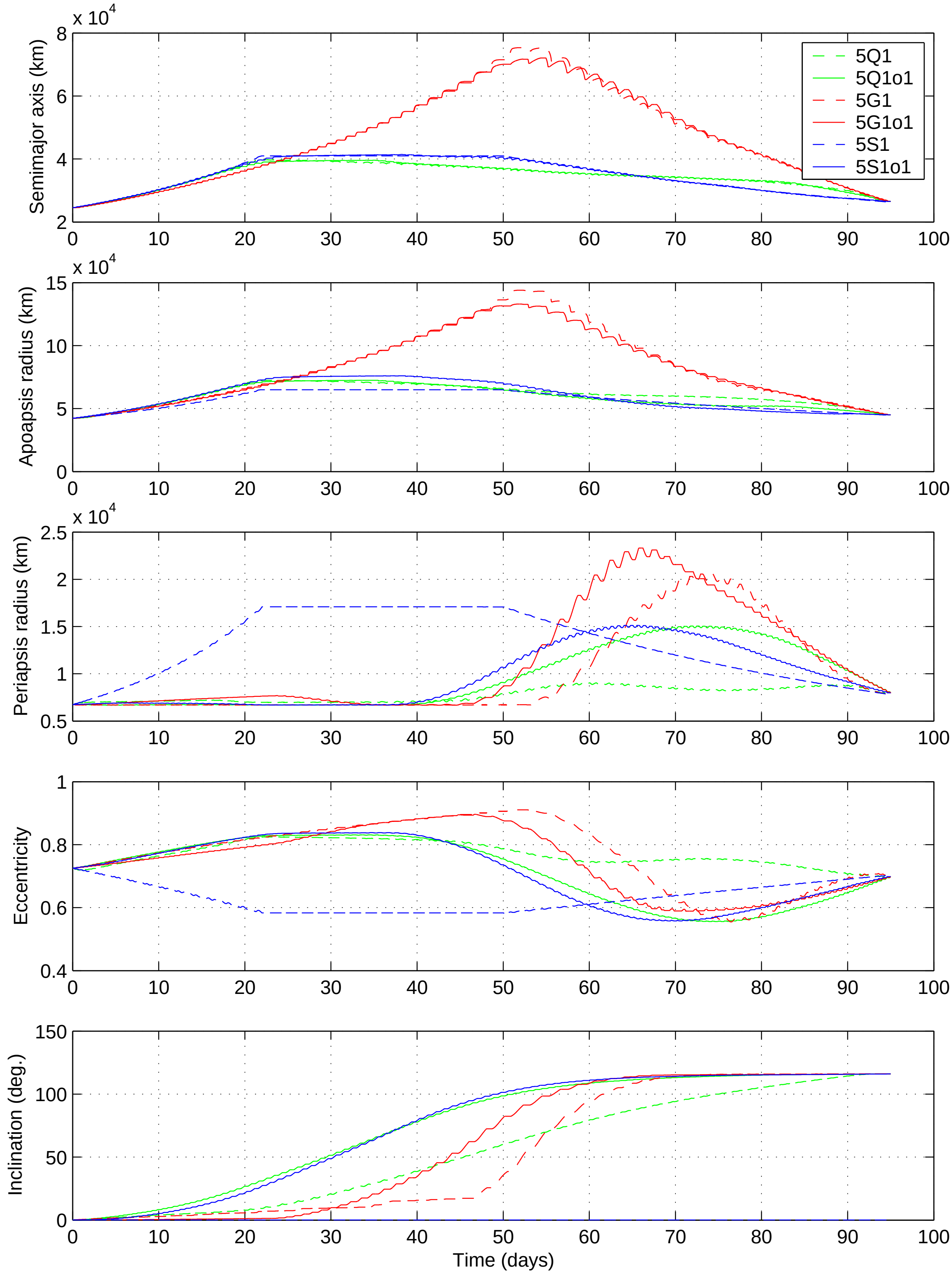

Figure 9 Orbit element time history for initial guesses (dashed lines) and the corresponding optimal solutions (solid lines) for the 95-day case.

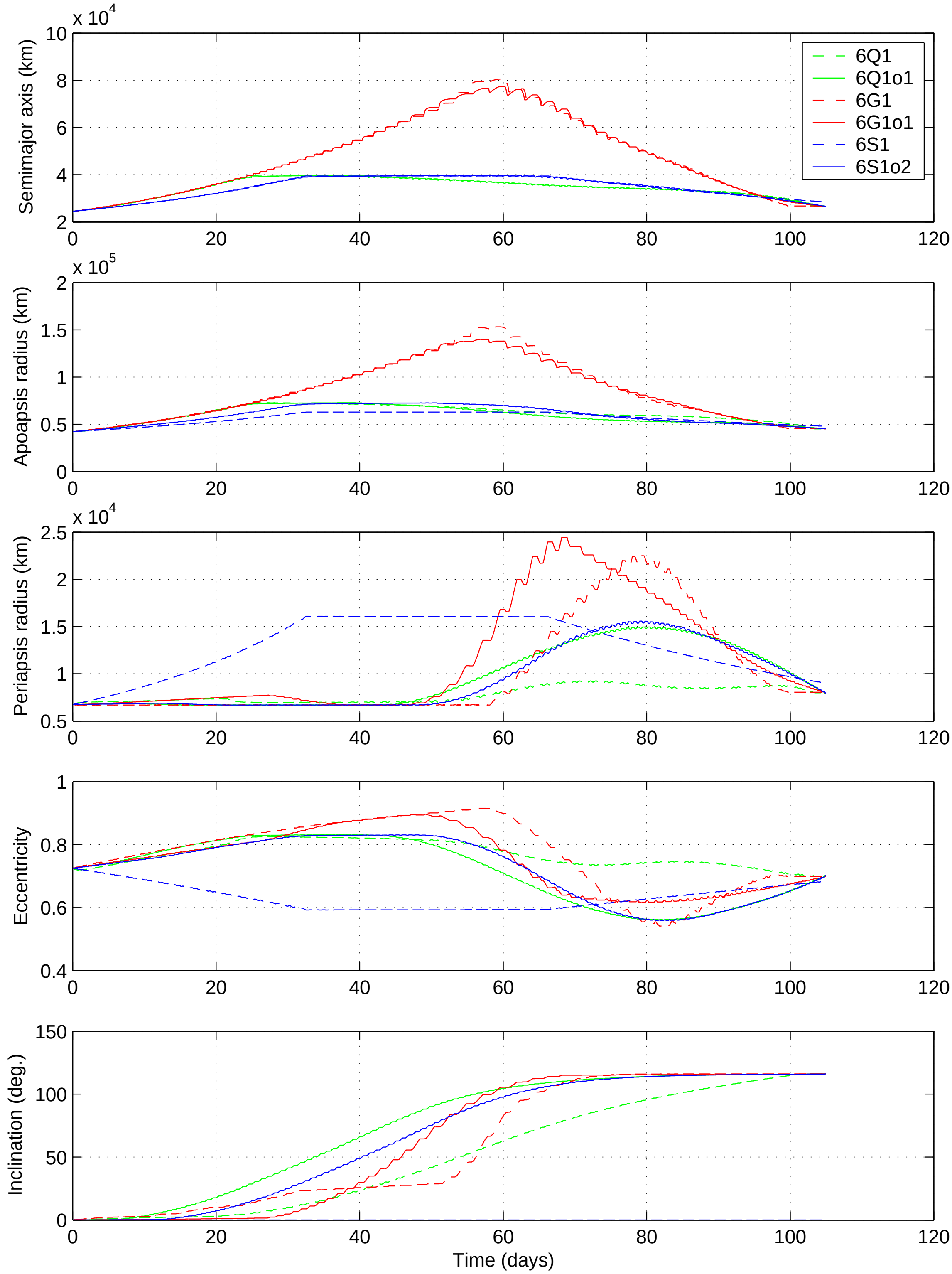

Figure 10 Orbit element time history for initial guesses (dashed lines) and the corresponding optimal solutions (solid lines) for the 105-day case.

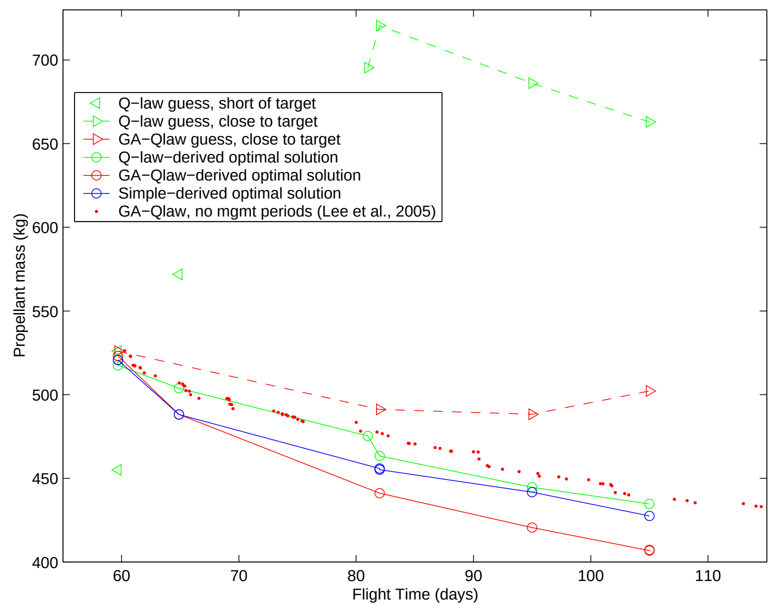

Figure 11 Propellant mass versus flight time for all the Q-law and GA-Qlaw initial guesses and all the optimal solutions listed in Table 2. Data are also shown for the GA-Qlaw trajectories (all meeting the target orbit) of Lee *et al.* (Ref. [6]), where a continuously varying thrust direction is permitted (no management periods used).

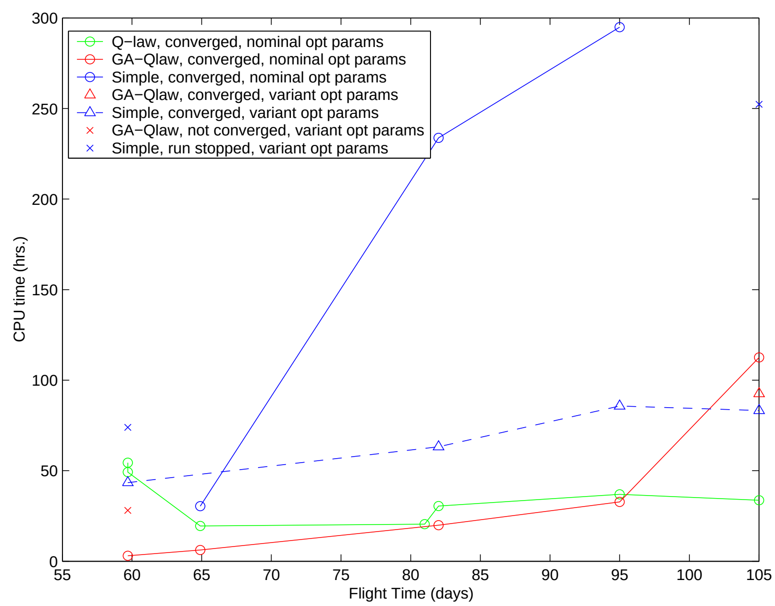

Figure 12 Mystic run times for all cases in Table 2. The times for the 59.7-day trajectories are scaled down by 39% from the Sun workstation times to allow comparison with the other times (linux) on the plot.

the genetic algorithm to find better Q-law parameters. The reason is that the GA-Qlaw algortithm of Ref. [6] generates trajectories *en masse* — an entire Pareto front containing many trajectories is computed in a single run on a parallel-processor. The initial guesses used here came from a run described in Ref. [6], which took about 41.3 hours total CPU time (4.13 hours wall-clock time since 10 processors were used) to generate 200 Pareto-optimal trajectories (having flight times between 60 and 500 days). (The run was performed on a cluster of performance similar to the one used here.) Thus, the time needed per Pareto-optimal trajectory is about 12 minutes on average. If the algorithm of Ref. [6] were to be rewritten to find single trajectories, the run time should be on about this order (*i.e.,* 12 minutes). Hence, since the run time to generate a GA-Qlaw trajectory is so small, even if not precisely defined, that it is not included in the CPU run times needed to obtain the final optimal solution in Mystic.

CONCLUSIONS

The GA-Qlaw typically provides initial guesses that converge between about five and thirty times faster than a simple initial guess. The Q-law typically provides initial guesses that converge between about five and ten times faster than a simple initial guess. In some cases, the convergence time for a simple initial guess can be reduced by a factor of three by using a tiered trust region scheme in the optimisation run. In addition, for the complex transfer considered here, the Q-law initial guesses provide reasonable estimates of the optimal performance, and the GA-Qlaw initial guesses (using continuously varying thrust directions) provide exceptional estimates of the optimal performance. The run time and trajectory-performance benefits seen here in using the Q-law and the GA-Qlaw for the initial guess will likely extend to other optimisers and to other orbit transfers, and possibly to transfers where significant perturbations to the two-body problem are present, and so should serve as a useful mission design tool which allows mission designers to explore the parameter space more fully and more quickly.

ACKNOWLEDGEMENTS

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

REFERENCES

1. Petropoulos, A.E., "Simple Control Laws for Low-Thrust Orbit Transfers," AAS Paper 03-630, AAS/AIAA Astrodynamics Specialist Conference, Big Sky, Montana, 03-07 Aug. 2003.
2. Petropoulos, A.E., "Low-Thrust Orbit Transfers Using Candidate Lyapunov Functions with a Mechanism for Coasting," AIAA 2004-5089, AIAA/AAS Astrodynamics Specialist Conference, Providence, Rhode Island, 16-19 Aug. 2004.
3. Petropoulos, A.E., "Refinements to the Q-Law for Low-Thrust Orbit Transfers," AAS 05-162, AAS/AIAA Space Flight Mechanics Conference, Copper Mountain, Colorado, 23-27 Jan. 2005.
4. Whiffen, G. J. and Sims, J. A., "Application of a Novel Optimal Control Algorithm to Low-Thrust Trajectory Optimization," AAS/AIAA Space Flight Mechanics Meeting, AAS Paper 01-209, Santa Barbara, California, Feb. 11-15, 2001.
5. Whiffen, G.J., "Optimal Low-Thrust Orbit Transfers around a Rotating Non-Spherical Body" AAS/AIAA Space Flight Mechanics Meeting, AAS Paper 04-264, Maui, Hawaii, Feb. 8-12, 2004.
6. Lee, S., von Allmen, P., Fink, W., Petropoulos, A.E., and Terrile, R., "Design and Optimization of Low-Thrust Trajectories," Paper 2.1202, IEEE Aerospace Conference, Big Sky, Montana, 05-12 Mar. 2005.