AAS 15-370

LOW-THRUST TRAJECTORY OPTIMIZATION IN DROMO VARIABLES

Juan L. Gonzalo*, and Claudio Bombardelli†

The Dromo orbital propagator was recently introduced by Peláez et al., and has been under active development. It has proven to be an excellent propagation tool, both in terms of accuracy and computational cost. In this article, we explore its applicability to the solution of optimal control problem in low-thrust missions. To this end, an optimal control formulation based in Dromo and a direct transcription method is used to solve several LEO-GEO and escape from Earth problems; the obtained results clearly show the suitability of this orbital propagator for such purposes.

INTRODUCTION

The *Dromo* orbital propagator was initially introduced by Peláez et al.[1] and has been under active development since then.[2,3] Other than showing excellent performance in terms of propagation speed and accuracy the regularized formulation of Dromo has proven to be particularly advantageous to find approximate solutions of basic astrodynamics problems.[4,5,6] Recently, the Dromo formulation has been employed as a basis for the solution of multi-revolution low-thrust optimization problems in Earth orbit[7] showing a dramatic improvement when compared to a cartesian based formulation. The reason for the improvement comes form three important characteristics of the method. First of all, its good performance in both precision and computational cost. Secondly, the state of the object is described by a set of redundant generalized orbital parameters, whose variations are related in magnitude and time scale with that of the control; this greatly improves the numerical behavior compared to a Cowell formulation, were the variations of the state due to the acceleration exerted by the low-thrust engine are normally very small compared with the characteristic variations along one orbit. Finally, the same formulation is valid for elliptic, hyperbolic and parabolic orbits, making it applicable to many different scenarios and easily allowing for optimal trajectories that combine several types of orbit. Other non-singular element methods have been successfully used by other authors[8] with positive results.

In this article, a formulation of the Optimal Control Problem based on a direct transcription method and the *Dromo* orbital propagator is used to solve two types of Earth-bound problems: LEO to GEO transfers and escapes from Earth orbit. The proposed formulation has a strong focus on efficiency, taking advantage of the sparsity of the discretized dynamic problem and using analytic expressions for the gradients of the equations. Approximated analytic results for the case of very low tangential thrust are used[4] to construct an initial guess as close to the true optimum as possible.

*PhD candidate, Space Dynamics Group, School of Aerospace Engineering, Technical University of Madrid (UPM).
†Research associate, Space Dynamics Group, School of Aerospace Engineering. Technical University of Madrid (UPM).

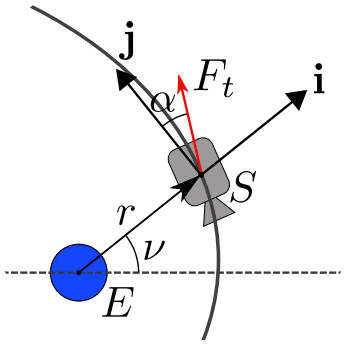

Figure 1. Schematic representation of the problem.

The structure of the article is the following. First we introduce the model employed for the optimization problem with all simplifying assumptions. The equations of motion in Dromo variables are then introduced and employed to formulate the planar, constant thrust control problem. Next, the optimal control problem solution strategy using a non-linear programming approach is described. Finally, results for the optimization of a number of LEO to GEO and LEO to escape trajectories are discussed.

DESCRIPTION OF THE PROBLEM

Let us consider a spacecraft S of mass m orbiting around a primary E of gravitational parameter μ. Its initial radial position is R_0, measured from the center of the primary, and its initial angular position is ν_0, measured from the initial eccentricity vector (i.e. ν_0 is the initial true anomaly). The only forces acting on the spacecraft are the attraction of the primary, which from now on is assumed to be the Earth, and the constant, in-plane thrust produced by its propulsion system F_t. The mass of the propellant is considered to be negligible compared to the initial mass of the spacecraft, so the later can be taken as constant as a first approximation. Under these conditions, the Optimal Control Problem (OCP) of determining the thrust orientation profile α to perform a series of Earth-bound, planar, optimum orbital maneuvers is studied.

All equations and variables considered hereafter are expressed in non-dimensional form: to this end, three characteristic magnitudes are introduced for length (R_0), mass (m) and time (n_0^{-1}), where n_0 is the angular frequency of the circular orbit of radius R_0 around the primary, $n_0 = \sqrt{\mu/R_0^3}$. The remaining magnitudes are derived from them, as shown in table 1.

Table 1. Characteristic values used to introduce non-dimensional variables.

Length	Time	Mass	Velocity	Acceleration	Force
R_0	$n_0^{-1} = \sqrt{R_0^3/\mu}$	m	$R_0 n_0$	$R_0 n_0^2$	$m R_0 n_0^2$

EQUATIONS OF MOTION. DROMO ORBITAL FORMULATION

In the Dromo orbital formulation, initially proposed by Peláez et al.[1] and further developed by multiple authors,[2,9] a fictitious time θ is introduced through a change of independent variable given by a Sundman transformation. The variation of parameters technique is then used to obtain expressions for the derivatives of seven generalized orbital elements, which along the non-dimensional physical time τ describe the state of the particle. These elements are constant for the unperturbed problem, but evolve in the presence of perturbing forces (that is, any force expect the gravitational attraction of the primary). Furthermore, it is possible to separate these orbital elements into two groups: the first three describe the geometry of the orbit in its plane, while the other four give the orientation of said plane. Consequently, the later are constant for the planar problem (when there is no out-of-plane component of the perturbation), and the motion of the particle is described by a 4-dimensional state vector:

$$(\tau, q_1, q_2, q_3) \tag{1}$$

whose evolution is given by the following system of four differential equations:

$$\frac{\mathrm{d}\tau}{\mathrm{d}\theta} = \frac{1}{q_3 s^2} \tag{2}$$

$$\frac{\mathrm{d}}{\mathrm{d}\theta}\begin{bmatrix} q_1 \\ q_2 \\ q_3 \end{bmatrix} = \frac{1}{q_3 s^3}\begin{bmatrix} s\sin\theta & (s+q_3)\cos\theta \\ -s\cos\theta & (s+q_3)\sin\theta \\ 0 & -q_3 \end{bmatrix}\begin{bmatrix} \mathbf{f}_p \cdot \mathbf{i} \\ \mathbf{f}_p \cdot \mathbf{j} \end{bmatrix} = \mathbf{F} \tag{3}$$

with

$$s = q_3 + q_1\cos\theta + q_2\sin\theta \tag{4}$$

$$r = \frac{1}{q_3 s} \tag{5}$$

$$u = \frac{\mathrm{d}r}{\mathrm{d}\tau} = q_1\sin\theta - q_2\cos\theta \tag{6}$$

where $\mathbf{f}_p$ is the total perturbing force, r is the orbital radius, u is the radial velocity and s is the transversal velocity, all of them in non-dimensional form. Versors $\mathbf{i}$ and $\mathbf{j}$ are oriented in the radial and transversal directions respectively, and they define a *Local Vertical - Local Horizontal (LVLH)* reference frame. A careful examination of these equations reveals three issues worth noting. First of all, Equation (2) is uncoupled from Equations (3), so the problem can be solved for (q_1, q_2, q_3) without calculating the dimensionless time. Secondly, θ coincides with the true anomaly of the Keplerian orbit for the unperturbed problem. As a consequence, the initial condition for θ in both perturbed and unperturbed problems is $\theta_0 = \nu_0$. Finally, the method presents only two singular cases: one when the trajectory reaches a point at the infinite ($r = \infty$), and other when the angular momentum vanishes ($h = 0$). Nevertheless, these are uncommon situations.

It is convenient to define relations between Dromo parameters and the classical orbital elements. The following formulas have been compiled from several authors,[1,9,2] and particularized for the planar case

$$q_1 = \frac{e}{h}\cos\Delta\gamma, \qquad q_2 = \frac{e}{h}\sin\Delta\gamma, \qquad q_3 = \frac{1}{h} \tag{7}$$

$$e = \frac{\sqrt{q_1^2 + q_2^2}}{q_3}, \qquad \Delta\gamma = \tan^{-1}\left(\frac{q_2}{q_1}\right), \qquad h = \frac{1}{q_3}, \tag{8}$$

$$\nu = \tan^{-1}\left(\frac{q_1 \sin\theta - q_2 \cos\theta}{q_1 \cos\theta + q_2 \sin\theta}\right), \quad E = \frac{q_1^2 + q_2^2 - q_3^2}{2}, \quad a = \frac{1}{q_3^2 - q_1^2 - q_2^2} \tag{9}$$

where e is the eccentricity, h is the angular momentum, E is the total energy, a is the semimajor axis and $\Delta\gamma$ is the angular drift between the variations of the fictitious time and the true anomaly[2]

$$\Delta\theta = \Delta\nu + \Delta\gamma.$$

In the particular case of planar orbit, $\Delta\gamma$ coincides with the angle between the initial and osculating eccentricity vectors,[2] and θ becomes the inertial angular position of the particle measured from the initial eccentricity vector. Finally, note that Equations (7) can also be used to calculate the initial values for the Dromo orbital elements, knowing that $\Delta\gamma(\theta_0) = 0$ and $h_0 = \sqrt{1 + e_0 \cos\nu_0}$.

To close this introduction to Dromo, it is interesting to point out some interesting properties regarding its application for studying Optimal Control Problems using the Direct Method:

- The same formulation is valid for any kind of orbit, allowing to solve a large variety of optimization problems.
- Speed and precision. This is capital, since a large number of orbits have to be computed while solving the NLP subproblem obtained from the discretization of the continuous OCP.
- Good behavior when integrated with fixed-step routines, which are normally used when solving OCPs using the Direct Method.
- The state varies in the same time scale as the perturbation. The main perturbation term is the thrust of the engine, whose orientation is the control law to be optimized; therefore, *state and control vary in the same time scale*.

Planar, Constant Thrust Problem

The motion of the particle in the planar case is governed by the ODE system formed by Equations (2) and (3), where (τ, q_1, q_2, q_3) is the state vector. To close the problem, it only remains to define the perturbing force, $\mathbf{f}_p$. The only perturbation considered in this work is the thrust produced by the propulsion system, whose orientation along the orbit will be the control function in the family of OCPs studied in the following section. The magnitude of the thrust is assumed to be constant, and it is defined in its non-dimensional form as

$$\varepsilon = \frac{F_t}{m\mu/R_0^2} = \frac{A_t}{\mu/R_0^2} \tag{10}$$

where F_t is the thrust exerted by the propulsion system, and $A_t = F_t/m$ is the corresponding acceleration. The study is centered in propulsion systems of high specific impulse I_{sp}, such as ionic, plasma or nuclear thermal engines, so ε and the specific fuel consumption take very small values. Taking the later into account, for a first approximation the mass of the vehicle is assumed to be constant.

Projecting $\mathbf{f}_p$ into the LVLH frame yields

$$\begin{bmatrix} \mathbf{f}_p \cdot \mathbf{i} \\ \mathbf{f}_p \cdot \mathbf{j} \end{bmatrix} = \begin{bmatrix} a_r \\ a_\theta \end{bmatrix} = \varepsilon \begin{bmatrix} \sin\alpha \\ \cos\alpha \end{bmatrix}, \quad \varepsilon = \sqrt{a_r^2 + a_\theta^2} \tag{11}$$

where a_r is the radial component of the perturbing acceleration, a_θ is the transversal component of the perturbing acceleration, and α is the angle formed by the acceleration vector and the transversal direction. While expressing the perturbing forces in terms of α is convenient, normally the angle β formed by the acceleration and the instantaneous velocity vector is of greater interest. The relation between α and β is given by:

$$\alpha = \beta + \arctan\left(\frac{e\sin\nu}{1+e\cos\nu}\right) \tag{12}$$

Therefore, the problem can be posed in terms of α, which yields simpler equations than using β, and then the later can be recovered by applying Equation (12).

DIRECT METHOD FOR OPTIMAL CONTROL PROBLEMS

OCPs can be solved using either Direct or Indirect Methods; for this study, only the former are considered. In Direct Methods, the original, continuous OCP must be transcribed into a discrete Non-Linear Programming (NLP) problem, which can then be solved using a suitable algorithm. This implies approximating the continuous functions and equations involved by a set of discrete parameters and constraints, as well as defining an adequate objective function to minimize. To this end, a grid of M uniformly spaced nodes is defined for the independent variable. To ease the construction of the grid for free-final-time problems, a new change of independent variable is introduced

$$\theta = \theta_0 + (\theta_f - \theta_0)t\,, \quad t \in [0,1] \tag{13}$$

so the position of the grid nodes in t is constant regardless of the values of θ_0 and θ_f. Then, the discrete values for Dromo fictitious time, state and control at the k-th node can be written as:

$$t^k = \frac{k-1}{M-1}, \quad \theta^k = \theta_0 + (\theta_f - \theta_0)t^k, \quad \tau^k = \tau(\theta^k), \quad \mathbf{q}^k = \mathbf{q}(\theta^k), \quad \alpha^k = \alpha(\theta^k).$$

The equations of motion are enforced by transcribing them into a set of *defect constraints*, using a suitable method from the Implicit Runge-Kutta family of algorithms for the integration of initial value problems. Implicit methods are usually preferred for the direct transcription of dynamics because they have better stability and greater order for the same number of steps, while their normally higher computational cost is compensated by the fact that they are going to be solved iteratively inside the NLP algorithm. Additionally, the Runge-Kutta family of methods is chosen because they respond better to fast variations in the problem, and behave better with relative large steps. The equations corresponding to the Trapezoidal and Hermite-Simpson Separated methods are given in Appendix B, particularized for the chosen orbital formulation and the problem at hand. Note that, since the equations for the physical time and the generalized orbital elements are uncoupled, only the former are considered for the definition of the defect constraints. In those cases where physical time plays a role, be it as part of the objective function or as a constraint of the OCP, the corresponding equations will added.

The OCP may introduce additional constraints apart from those associated with dynamics. For the purposes of this study, it is important to consider the *fixed final energy* constraint, whose equation is:

$$g^E(\mathbf{x}) \equiv \frac{1}{2}\left[(y_1^M)^2 + (y_2^M)^2 - (y_3^M)^2\right] - E_f^* = 0 \tag{14}$$

where E_f^* is the desired final energy; note that the particular case of *escape* can be obtained by setting $E_f^* = 0$. It is also interesting to considered the *fixed final time* constraint

$$g^\tau(\mathbf{x}) \equiv \tau_f^* - \tau_f = 0 \tag{15}$$

where τ_f^* is the desired final time, and τ_f can be determined using the equations presented in the next paragraph for the minimum-time objective function.

To close the transcription of the OCP, a suitable expression must be given for the objective function. Multiple objective functions can be defined depending on the OCP; for this study, only the cases of maximum final energy and minimum total time are considered. The maximum energy case is a *problem of Mayer*, since the objective function only contains terms depending on the final value of the state:

$$J(\mathbf{x}) = -E = -\frac{1}{2}\left(q_1^2 + q_2^2 - q_3^2\right)\big|_{\theta=\theta_f} = -\frac{1}{2}\left[(q_1^M)^2 + (q_2^M)^2 - (q_3^M)^2\right] \tag{16}$$

On the other hand, the minimum transfer time is a *problem of Lagrange*, since it is expressed as a quadrature. This is due to the physical time being a *dependent* variable in Dromo formulation. The quadrature for the final non-dimensional time τ_F can be obtained from Equation (2), yielding:

$$J = \tau_F = \int_{\theta_0}^{\theta_F} \frac{1}{q_3 s^2}\mathrm{d}\theta = \int_{\theta_0}^{\theta_F} m(q_1, q_2, q_3, \theta)\mathrm{d}\theta \tag{17}$$

To transcribe (17), a numerical scheme consistent with the one used for the dynamics is recommended. Therefore, a trapezoidal quadrature is used for the Trapezoidal transcription

$$J(\mathbf{x}) = \sum_{k=1}^{M-1} \frac{h_k}{2}(m^k + m^{k+1}) = \frac{h_1}{2}m^1 + \frac{1}{2}\sum_{k=2}^{M-1}(h_k + h_{k-1})m^k + \frac{h_{M-1}}{2}m^M$$

while a Simpson scheme is preferred for the HSS transcription

$$J(\mathbf{x}) = \sum_{k=1}^{M-1} \frac{h_k}{6}(m^k + 4\overline{m}^{k+1} + m^{k+1})$$

where m^k is the value of the integrand at the k-th node, $\overline{m}^k$ is the value of the integrand at the middlepoint between the k-th and the $(k+1)$-th nodes, and $h_k = \theta^{k+1} - \theta^k$.

To finalize this section on the direct transcription of the OCP, there are some issues regarding computational efficiency and memory usage worth noting. NLP algorithms normally require to compute the Jacobian and Hessian of the constraints and the objective function (or a suitable approximation). Since the size of the grid needed to adequately transcribe most real life problems is large, so will be the resulting Jacobian. However, constraints associated with dynamics normally depend on a small number of adjacent nodes (two for the Trapezoidal and HSS methods previously considered), so their gradient matrix is *sparse*: taking advantage of this can dramatically reduce the computational effort required for both calculating the gradients and solving each NLP step. Furthermore, the gradient matrix of the defect constraints can be constructed *analytically*, using the expressions given in Appendix A (for brevity sake, the resulting matricial expressions are omitted; they can be found in Reference 7). Analogously, the contribution of the defect constraints to the Hessian matrix shows a strong sparsity. And since the analytical expressions for the gradients of the

constraints are known, the Hessian can be computed numerically using the method of finite differences of the gradients. By following the procedure proposed by Coleman, Garbow and Moré[10] for sparse problems, only 6 evaluations of the gradients of the defect constraints are needed to compute a forward differences estimate of the Hessian (10 for a centered differences estimate).

NUMERICAL EXPERIMENTS

The problem of finding the optimum thrust orientation control law in the two-body planar, constant thrust scenario previously presented is formulated, using Dromo to propagate the orbit and the Direct Method to transcribe the OCP as a NLP problem. A third party software package is used to numerically solve this NLP problem, particularly MATLAB's global optimization toolbox and its *interior point* algorithm. The algorithm selection is based on its support for sparse, large scale problems, a highly desirable property when dealing with problems coming from the discretization of dynamical systems, as already discussed.

One key aspect not yet considered is the construction of a good enough initial guess for the iterative NLP solver. The quality of said initial guess not only affects the number of iterations needed to reach the solution, but it is also a deciding factor for the convergence of the NLP algorithm. For the purposes of this study, a simple yet good enough initial guess can be constructed knowing that, for the two-body problem with constant thrust, the maximum instantaneous energy raise is achieved using tangential thrust.[4] Furthermore, this is also the asymptotic solution when $\varepsilon \to 0$ for maximizing the orbital energy increase during a fixed period of time. Taking these results into account, an initial guess is constructed propagating the orbit for the tangential thrust case until a certain stopping criterion is reached. Said stopping criterion will depend on the characteristics of the OCP: for escape problems, it is good enough to propagate the tangential thrust orbit until escape is reached; while for orbit transfers various criterion involving q_3 (which is linked to the angular momentum) or the orbital energy can be considered.

All the numerical experiments were run in a computer with an Intel© Core™ i5-3450 processor with 4 physical cores at 3.10 GHz, and 8 GB of DDR3 RAM at 1600 MHz. Parallel computation was not enabled. The function and constraints tolerances for the *Interior-point* algorithm were set to 10^{-10}.

The escape from an equatorial LEO is now considered, for different values of the initial eccentricity e_0 and the non-dimensional thrust parameter ε. In all cases, the maneuver starts at the perigee of the orbit ($\nu_0 = 0$), which is placed at an altitude of $r_{p0} = 300\,\text{km}$. The numerical results have been computed using the Trapezoidal discretization, with a number of grid nodes depending on the value of ε (the lower the thrust the larger the number of revolutions needed to escape, thus requiring more nodes to get a fine enough grid). The first example, Figures 2, corresponds to an initially circular orbit and a thrust parameter of $\varepsilon = 10^{-3}$ ($a = 8.1346\,10^{-3}\,\text{m/s}^2$), and is calculated using a grid of $M = 700$ nodes. The NLP algorithm only took 5 iterations to converge, for a total computational time of $1.1126\,\text{s}$. The thrust orientation remains close to the tangential direction during most of the maneuver, oscillating around it. This supports the already introduced notion that the tangential control law is the low-thrust asymptotic solution for maximizing the orbital energy increase in a given time. The orbit shows the typical spiraling shape of low-thrust maneuvers, while the figures for the eccentricity and the semimajor axis confirm the similarities between the optimum and tangential control laws. A closer look at the eccentricity curves reveals that escaping using tangential thrust requires *less orbits* but *more time* than the optimum solution: this implies that the vehicle move faster in the optimum case. Also noteworthy is that, while both the eccentricity and the am-

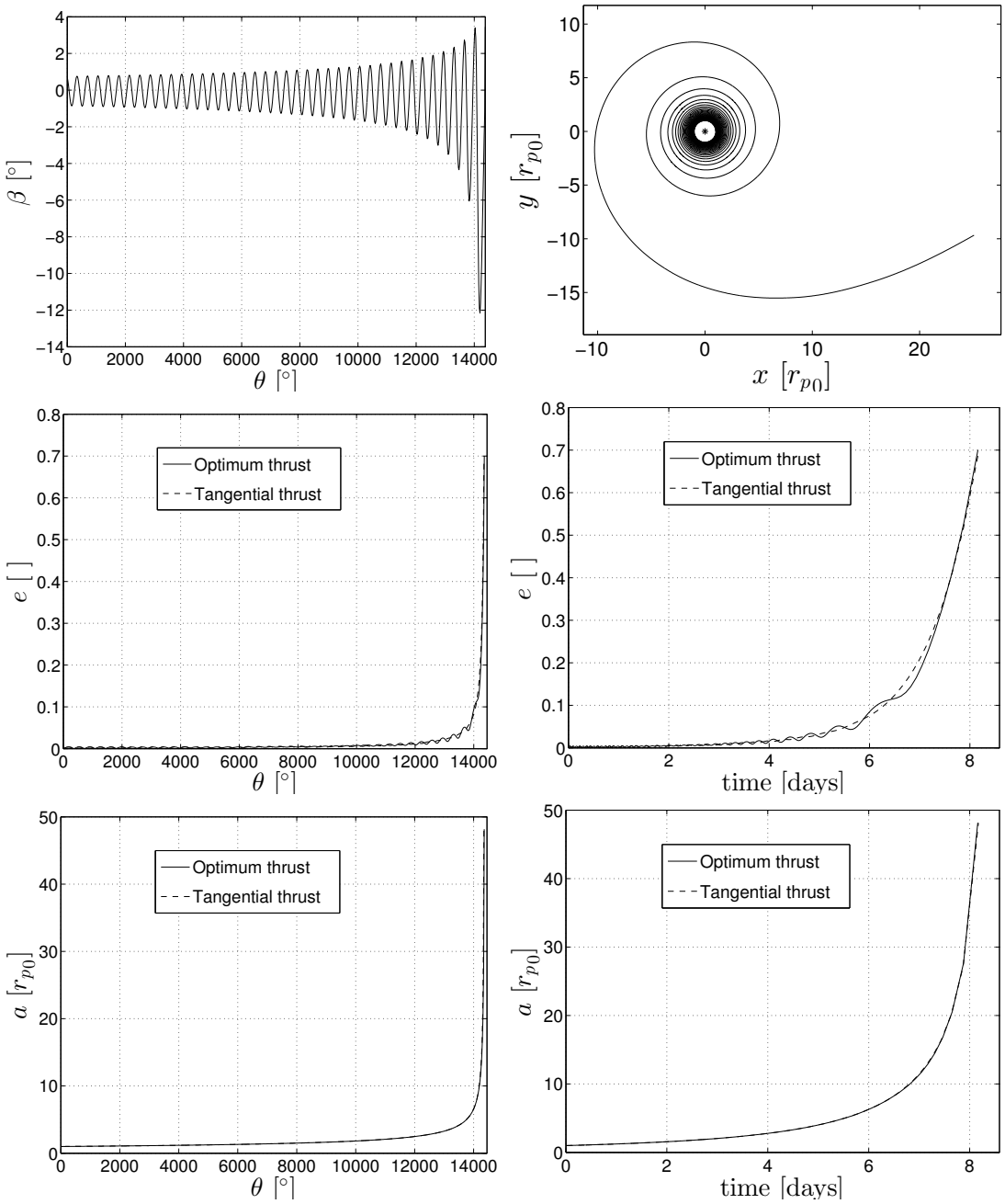

Figure 2. LEO-escape maneuver, for $e_0 = 0$ and $\varepsilon = 10^{-3}$.

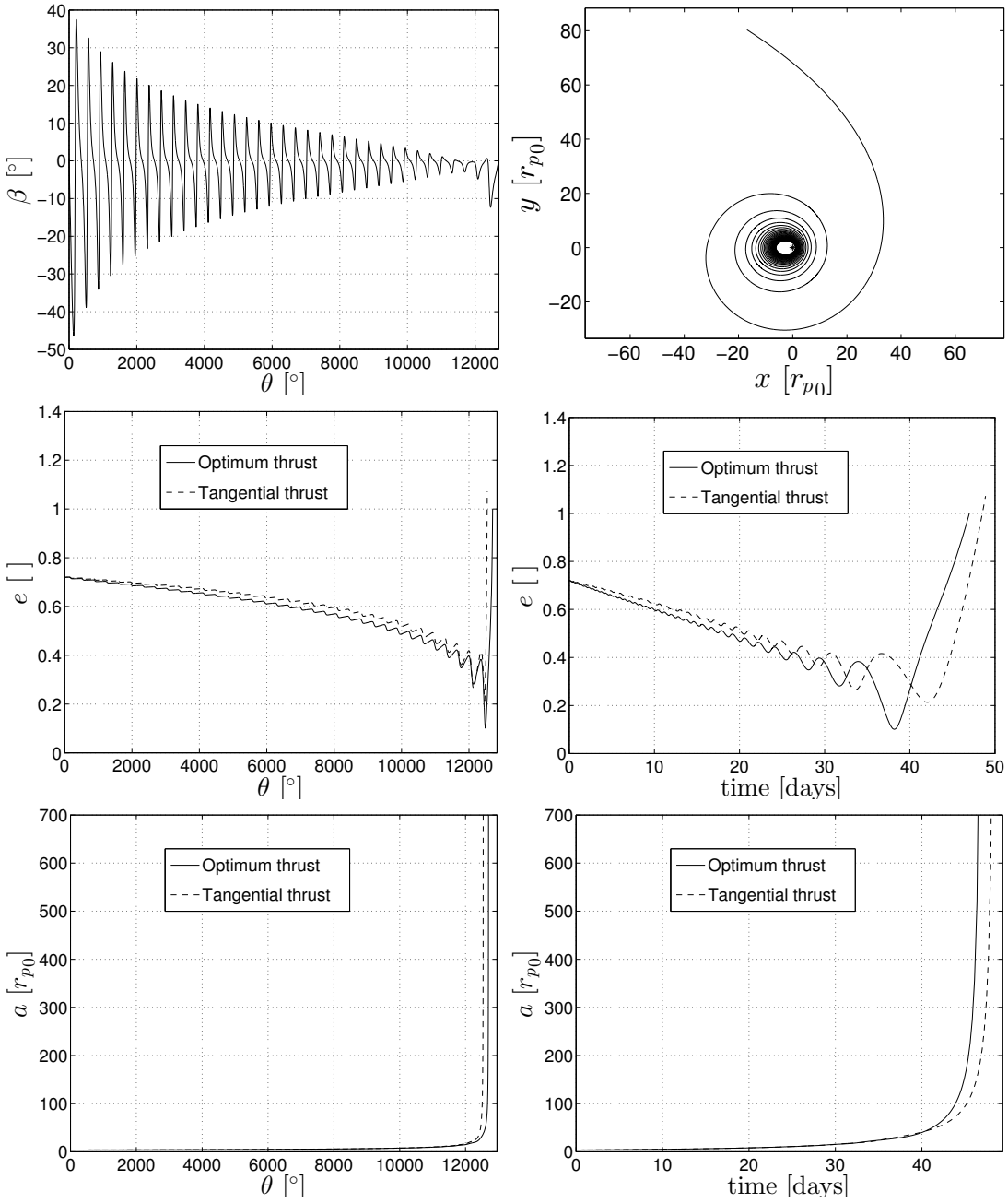

Figure 3. LEO-escape maneuver, for $e_0 = 0.72$, $\varepsilon = 10^{-4}$.

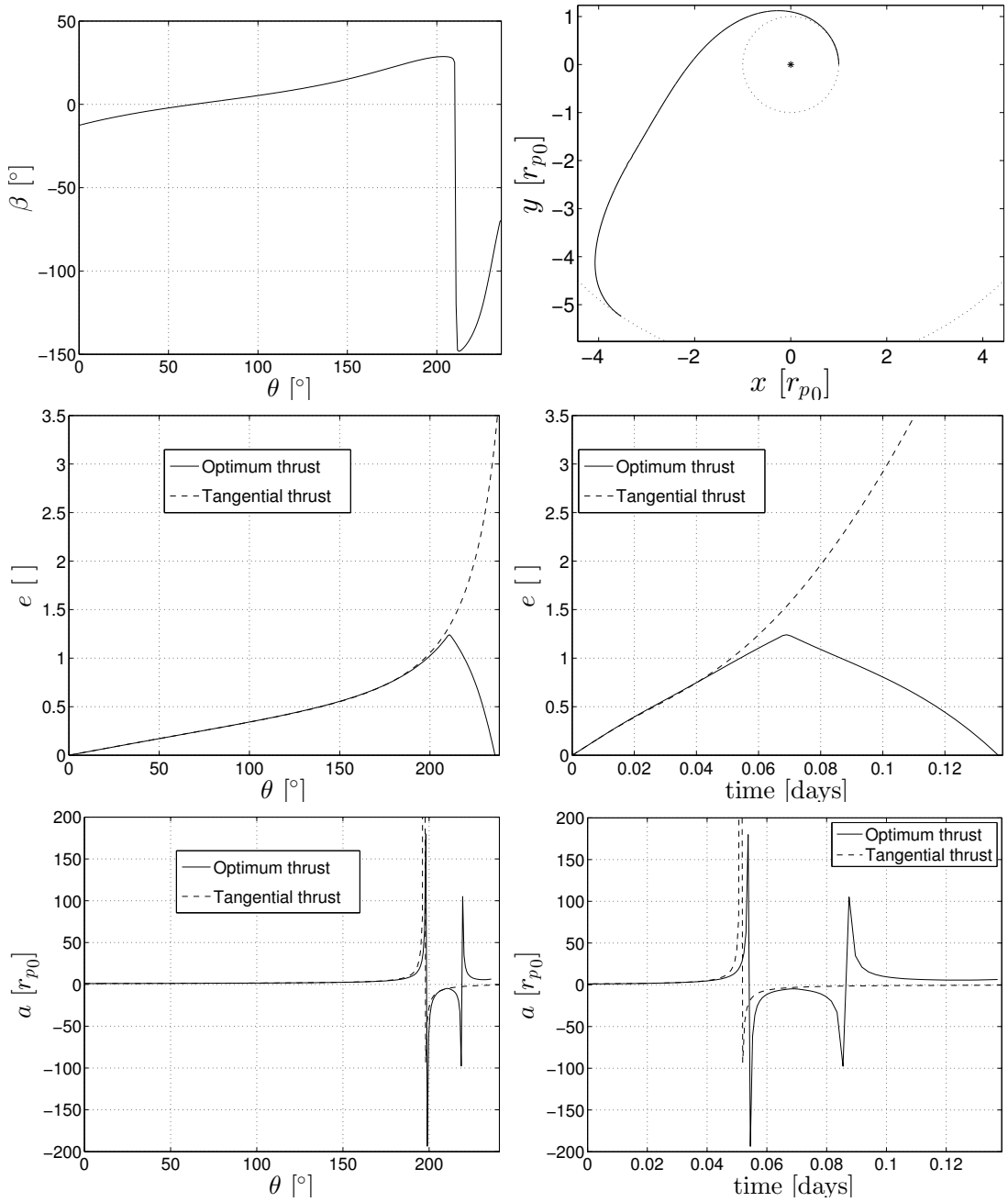

Figure 4. LEO to GEO transfer, for $e_0 = 0$ and $\varepsilon = 10^{-1}$.

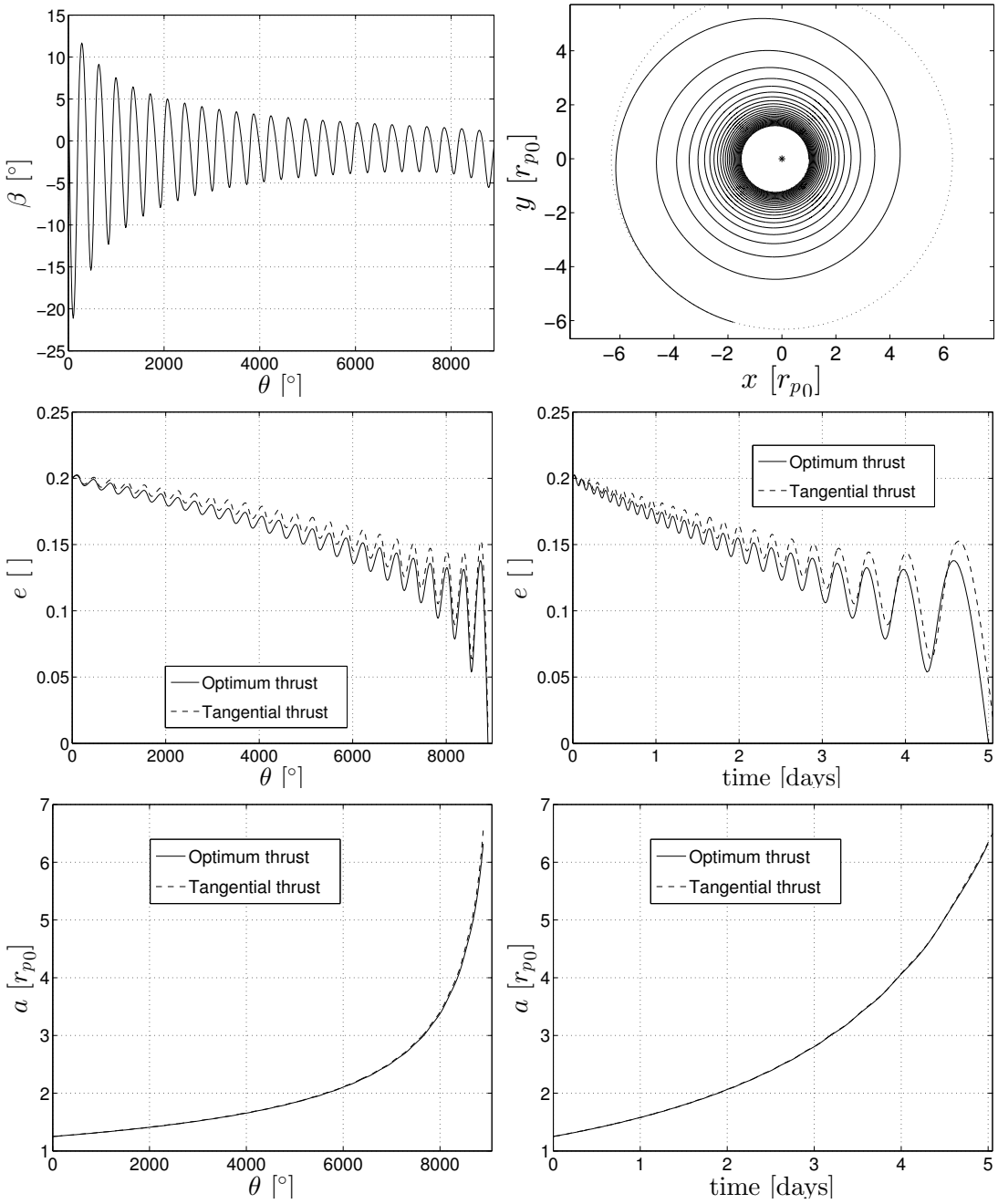

Figure 5. LEO to GEO transfer, for $e_0 = 0.2$ and $\varepsilon = 10^{-3}$.

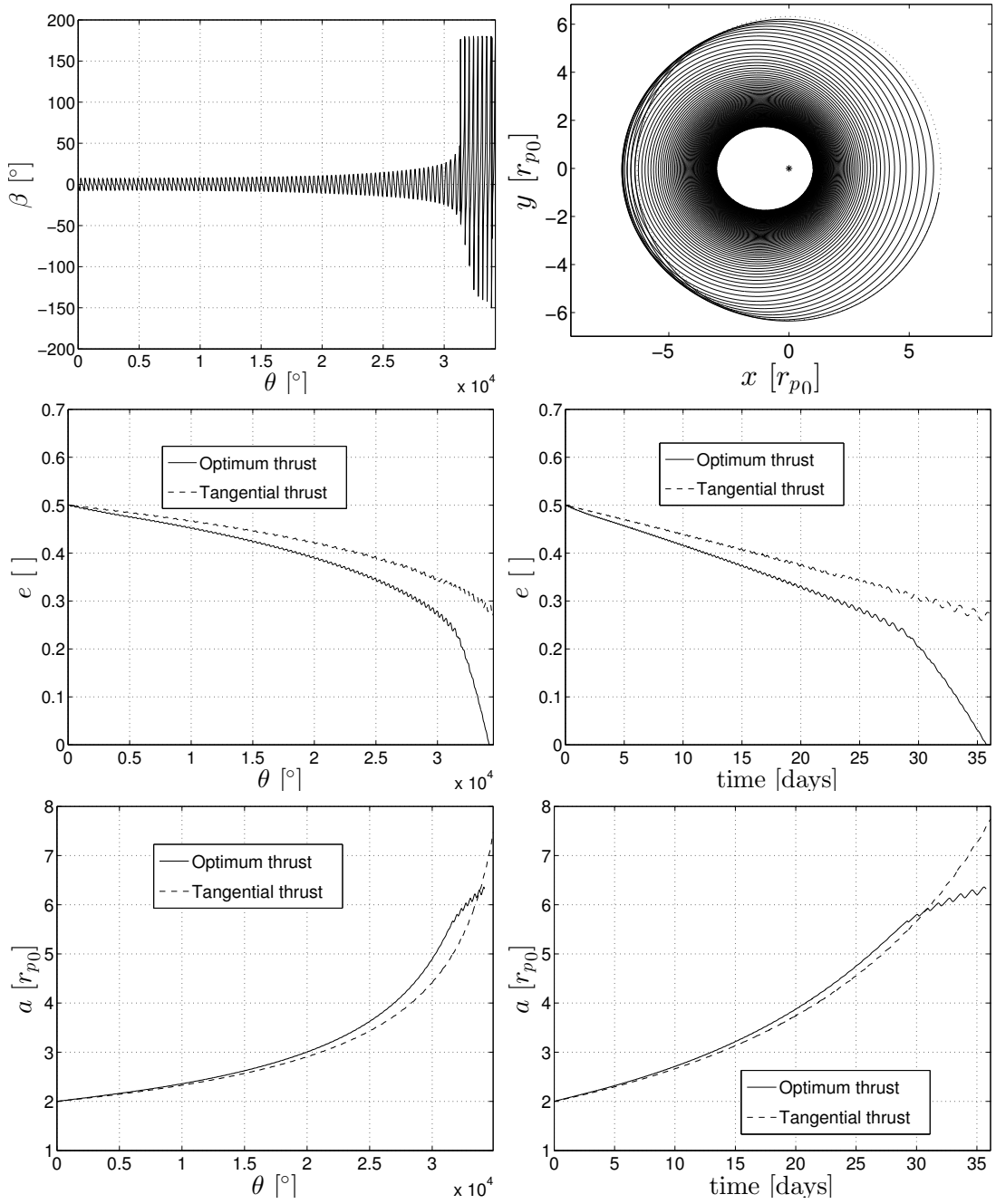

Figure 6. LEO to GEO transfer, for $e_0 = 0.5$ and $\varepsilon = 10^{-4}$.

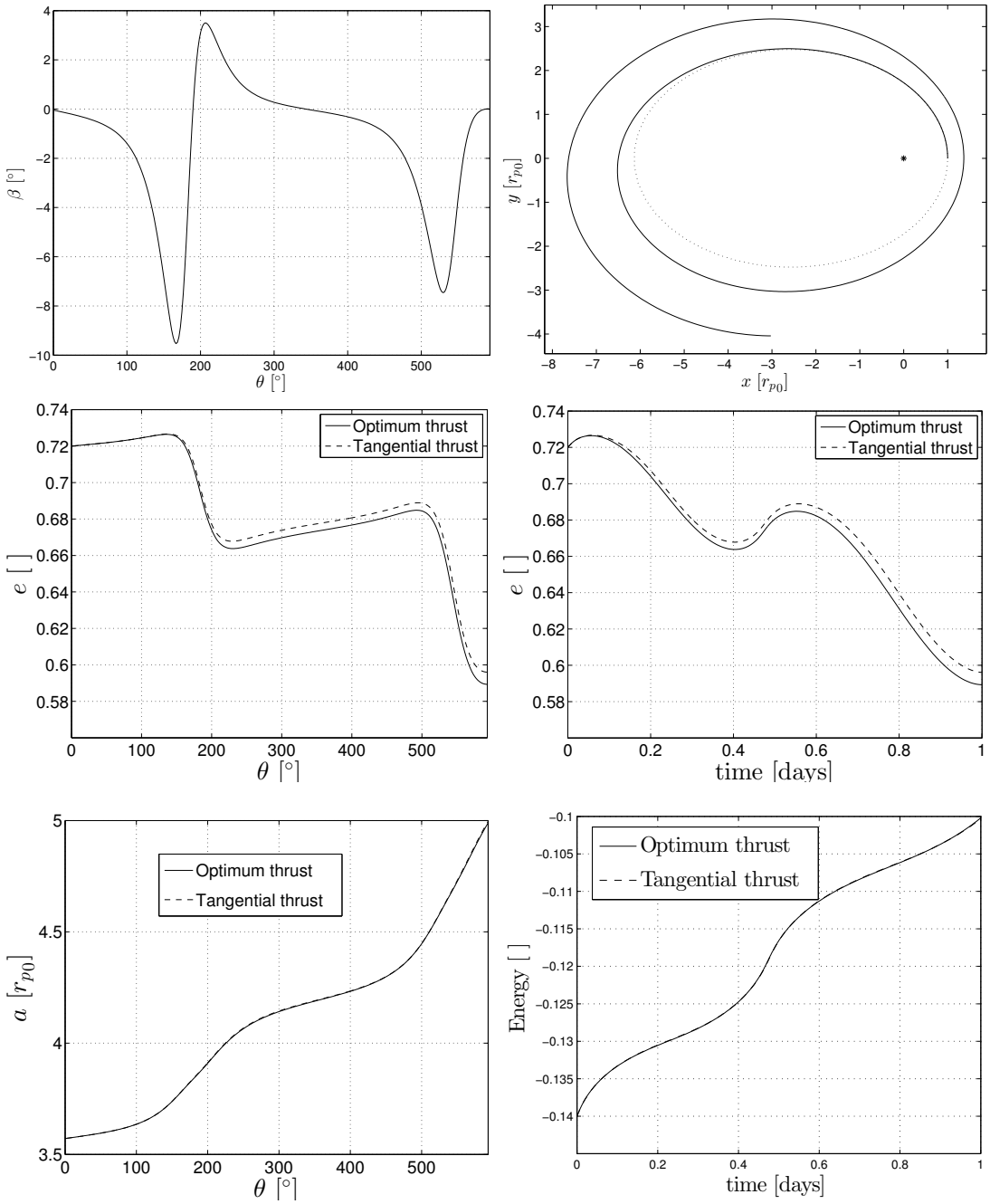

Figure 7. Maximum energy increase from a GTO in 1 day, for $\varepsilon = 10^{-3}$

plitude of β increase with time, said variations tend to concentrate at the end of the maneuver. A much more eccentric initial orbit is considered in Figures 3, calculated for $e_0 = 0.72$, $\varepsilon = 10^{-4}$ ($a = 8.1346\,10^{-4}$ m/s^2) and $M = 5000$. In this case the convergence of the NLP algorithm was slower, with 42 iterations and 417.8463 s of computational time. A specially low value for thrust has been selected to obtain more interesting results, since the high initial eccentricity reduces the number of revolutions needed to escape. Thrust orientation angle β still oscillates around the tangential direction, but it no longer resembles a sinusoidal. It presents rapid variations near the apogee of the orbit, while the evolution in the vicinity of the perigee becomes smoother. There is also a qualitative change in the evolution with time of the eccentricity and the amplitude of β. A tendency to circularize the orbit reducing both magnitudes appears during most of the maneuver, followed by a strong increase near the end. The differences between the optimum and tangential thrust control laws grow bigger towards the end of the trajectory, still verifying that the optimum escape takes place in less time but with more orbits than the tangential thrust escape.

The next set of OCPs corresponds to three LEO to GEO transfers. Same as before, the maneuver starts at the perigee ($\nu_0 = 0$), which takes place at an altitude of $r_{p0} = 300$ km, and the discretization is performed using the Trapezoidal method. The results for the first case, calculated with $e_0 = 0$, $\varepsilon = 10^{-1}$ ($a = 8.1346\,10^{-1}$ m/s^2) and $M = 300$, are represented in Figures 4. The convergence of the NLP algorithm was reached in 18 iterations and 3.2362 s. The value chosen for ε is rather large, not having a low-thrust application, but it serves to illustrate one of the benefits of the Dromo formulation. The curves for the eccentricity reveal that the orbit, which is elliptic during most of the maneuver, briefly becomes hyperbolic towards the end. Such a solution is possible because a formulation valid for all kind of orbits is used, offering a higher flexibility for the optimization process. Aside from this, the figures for e and a also include the curves corresponding to the tangential thrust control law: both cases are similar during the first part of the maneuver, but they separate as the optimum control law circularizes the orbit to fulfill the constraints. The next test problem, corresponding to $e_0 = 0.2$, $\varepsilon = 10^{-3}$ ($a = 8.1346\,10^{-3}$ m/s^2) and $M = 700$, is shown in Figures 5. This trajectory also showed a good behavior of the NLP algorithm, which required 18 iterations and 8.065 s to converge. The lower thrust leads to a typical spiraling orbit, with β oscillating around a mean value slightly separated from the tangential direction. This mean value is negative, indicating that thrust shows a small bias towards the inner part of the orbit. It is also worth noting that the amplitude of β decreases with time, while the oscillations in the eccentricity grow larger. The last LEO to GEO transfer, Figures 6, is calculated for $e_0 = 0.5$, $\varepsilon = 10^{-4}$ ($a = 8.1346\,10^{-3}$ m/s^2) and $M = 5000$. Unlike the previous cases, the convergence of this test problem presented some issues, requiring 200 iterations and 401.7085 s. The most notable characteristic of this transfer is that thrust actually points in the direction *opposite* to velocity during some parts of the orbit, near the end of the maneuver. This decelerating regions take place around the perigee, and are associated to the need of circularizing the orbit to meet the final constraints. A closer inspection of the curves for the eccentricity and the thrust orientation angle reveals that, instead of gradually reducing the eccentricity along the maneuver, it is more efficient to follow a nearly tangential thrust profile during most of it, and circularize the orbit at the end.

Finally, the problem of determining the thrust orientation control law that maximizes the final energy E_f reached in a given time departing from a GTO ($e_0 = 0.72$) is considered. The maneuver starts at the perigee, at an altitude of $r_{p0} = 300$ km, lasts one day and is performed with $\varepsilon = 10^{-3}$ ($a = 8.1346\,10^{-3}$ m/s^2). The OCP is discretized using the HSS method and a 300 points grid; under these conditions, the NLP algorithm took 14 iterations and 5.8399 s to converge. The

results, represented in Figures 7, show that the thrust orientation angle remains close to the tangential direction, but with a bias towards the inner part of the orbit due to its high eccentricity. It is checked that β becomes zero at the perigee, and experiments the greatest variations near the apogee. Furthermore, its value is negative for the first half of the orbit, when moving from the perigee to the apogee, and positive during the other half. Regarding the eccentricity and the semimajor axis, both combine a secular and an oscillatory term, the later having the same period of the orbit. The secular component for the eccentricity is decreasing, indicating that the orbit is circularized as the energy increases, while for the semimajor axis it is increasing. The improvement with respect to the tangential thrust case is small, as depicted in the curve for the energy.

CONCLUSION

A new formulation for solving optimal control problems of space trajectories, restricted to the planar, two bodies, low constant thrust case, has been defined using Dromo orbital propagator and a Direct transcription method. The formulation has been developed with computational efficiency as a key aspect, obtaining analytical expressions for the gradients of the constraints coming from dynamics and exploiting the sparsity of the involved Jacobian and Hessian matrices. The availability of a good initial guess based on the tangential thrust problem has allowed for fast and stable solutions.

The results obtained have confirmed Dromo as a great orbital propagator for this type of problems. This is due to several interesting properties. First of all, the same formulation is valid for all kinds of orbits. Secondly, it shows very good speed and precision, and behaves well when integrated with fixed-step routines (used in the transcription process). Finally, the state varies in the same time scale as the perturbation, which in this case is also the control.

The numerical experiments studied with the new formulation have yielded some interesting conclusions. In escape and energy increase problems, the optimum thrust profile approaches the tangential thrust case for small values of the non-dimensional thrust, and separates as it grows. Depending on the mission, the eccentricity of the initial orbit and the thrust, the differences between both control laws can be quite important or negligible. Regarding LEO to GEO transfers, it has been observed than the eccentricity reduction needed to reach the final circular orbit tends to concentrate at the final part of the maneuver.

ACKNOWLEDGMENT

This work has been supported by the Spanish Ministry of Education, Culture and Sport through its FPU Program (reference number FPU13/05910), and by the Spanish Ministry of Economy and Competitiveness within the framework of the research project "Dynamical Analysis, Advanced Orbital Propagation, and Simulation of Complex Space Systems" (ESP2013-41634-P).

APPENDIX A: ANALYTICAL GRADIENTS OF DROMO EQUATIONS

The Jacobian matrix for the defect constraints can be analytically constructed from the gradient of the Right Hand Side of Equations (3). This is highly desirable, since the usage of analytical expressions for the gradients yields lower computational times and better precision. Consider

$$\mathbf{F} = \mathbf{R} \cdot \begin{bmatrix} a_r \\ a_\theta \end{bmatrix}$$

where $\mathbf{F}$ is the RHS of Equations (3). Then, the gradient with respect to the reduced state vector $\mathbf{q}^{\mathrm{T}} = (q_1, q_2, q_3)$ is

$$\nabla_{\mathbf{q}}\mathbf{F}^{\mathrm{T}} = \begin{pmatrix} \dfrac{\partial}{\partial q_1}\mathbf{F} & \dfrac{\partial}{\partial q_2}\mathbf{F} & \dfrac{\partial}{\partial q_3}\mathbf{F} \end{pmatrix}$$

where

$$\frac{\partial}{\partial q_i}\mathbf{F} = \frac{\partial}{\partial q_i}\mathbf{R} \cdot \begin{bmatrix} a_r \\ a_\theta \end{bmatrix}$$

since the thrust acceleration vector does not depend on the state vector in the formulation used. Performing the partial derivatives and grouping terms:

$$\frac{\partial}{\partial q_1}\mathbf{R} = \frac{\cos\theta}{q_3 s^4} \begin{bmatrix} -2\sin\theta s & -\cos\theta(2s+3q_3) \\ 2\cos\theta s & -\sin\theta(2s+3q_3) \\ 0 & 3q_3 \end{bmatrix}$$

$$\frac{\partial}{\partial q_2}\mathbf{R} = \frac{\sin\theta}{q_3 s^4} \begin{bmatrix} -2\sin\theta s & -\cos\theta(2s+3q_3) \\ 2\cos\theta s & -\sin\theta(2s+3q_3) \\ 0 & 3q_3 \end{bmatrix}$$

$$\frac{\partial}{\partial q_3}\mathbf{R} = \frac{1}{q_3^2 s^4} \begin{bmatrix} -\sin\theta s(s+2q_3) & -\cos\theta(2q_3^2+(s+q_3)^2) \\ \cos\theta s(s+2q_3) & -\sin\theta(2q_3^2+(s+q_3)^2) \\ 0 & 3q_3^2 \end{bmatrix}$$

The gradient with respect to the control $\mathbf{u} = \alpha$ is:

$$\nabla_{\mathbf{u}}\mathbf{F} = \frac{\partial}{\partial\alpha}\mathbf{F} = \mathbf{R} \cdot \frac{\partial}{\partial\alpha}\begin{bmatrix} a_r \\ a_\theta \end{bmatrix} = \mathbf{R} \cdot \varepsilon \begin{bmatrix} \cos\alpha \\ -\sin\alpha \end{bmatrix}$$

Finally, the derivative with respect to the independent variable θ is given by

$$\frac{\partial}{\partial\theta}\mathbf{F} = \frac{\partial}{\partial\theta}\mathbf{R} \cdot \begin{bmatrix} a_r \\ a_\theta \end{bmatrix} = \frac{1}{q_3 s^4} \begin{bmatrix} D_{11} & D_{12} \\ D_{21} & D_{22} \\ D_{31} & D_{32} \end{bmatrix} \cdot \begin{bmatrix} a_r \\ a_\theta \end{bmatrix}$$

where:

$$\begin{aligned}
D_{11} &= s\left[2q_1 - (-s + 2q_1\cos\theta + 2q_2\sin\theta)\cos\theta\right] \\
D_{12} &= \cos\theta(2s+3q_3)(q_1\sin\theta - q_2\cos\theta) - s\sin\theta(s+q_3) \\
D_{21} &= -2\cos\theta s(q_1\sin\theta - q_2\cos\theta) + s^2\sin\theta \\
D_{22} &= \sin\theta(q_1\sin\theta - q_2\cos\theta)(2s+3q_3) + s\cos\theta(s+q3) \\
D_{31} &= 0 \\
D_{32} &= -3q_3(q_1\sin\theta - q_2\cos\theta)
\end{aligned}$$

The gradient of the RHS of Equation (2) is also needed to analytically construct the Jacobian for certain objective functions (particularly, minimum-time problems). Let us define

$$m = \frac{1}{q_3 s^2}$$

where m is the RHS of Equation (2). Then

$$\nabla_{\mathbf{q}} m = -\frac{1}{q_3 s^3} \begin{bmatrix} 2\cos\theta \\ 2\sin\theta \\ 2 + s/q_3 \end{bmatrix}$$

$$\nabla_u m = 0$$

$$\frac{\partial}{\partial\theta} m = \frac{2}{q_3 s^3}(q_1 \sin\theta - q_2 \cos\theta).$$

APPENDIX B: DEFINITION OF THE DEFECT CONSTRAINTS

The process of transcribing the problem dynamics into a discrete set of non-linear defect constraints is briefly outlined in this Appendix, using the Trapezoidal and Hermite-Simpson Separated methods.

The *Trapezoidal method* is a 2-stages, 3rd order, IRK scheme, given by equation:

$$\mathbf{q}^{k+1} = \mathbf{q}^k + \frac{h_k}{2}(\mathbf{F}^{k+1} + \mathbf{F}^k)$$

with

$$\mathbf{F}^k = \mathbf{F}(\mathbf{q}^k, \mathbf{u}^k, \theta^k), \quad h^k = (\theta_f - \theta_0)(t^{k+1} - t^k),$$

where $\mathbf{F}$ is the right hand side of Equations (3), $\mathbf{q}^k = \left(q_1^k, q_2^k, q_3^k\right)^{\mathrm{T}}$ is the state at the k-th node and $\mathbf{u}^k = \alpha^k$ is the control at the k-th node. Expressing it as a defect constraint yields

$$\boldsymbol{\xi}_k(\mathbf{x}) \equiv \mathbf{q}^{k+1} - \mathbf{q}^k - \frac{h_k}{2}(\mathbf{F}^{k+1} + \mathbf{F}^k) = 0$$

where $\mathbf{x}$ is the $(4M+1)$-th dimensional optimization variable:

$$\mathbf{x}^{\mathbf{T}} = \left(\theta_f, \mathbf{q}^1, \alpha^1, \ldots, \mathbf{q}^k, \alpha^k, \ldots \mathbf{q}^M, \alpha^M\right)$$

Then, applying these defect constraints to all nodes of the grid except the last one, a $3(M-1)$-th dimensional column vector of non-linear defect constraints is reached

$$\mathbf{c}(\mathbf{x})^{\mathrm{T}} = \begin{bmatrix} \boldsymbol{\xi}_1(\mathbf{x}) & \boldsymbol{\xi}_2(\mathbf{x}) & \ldots & \boldsymbol{\xi}_{M-1}(\mathbf{x}) \end{bmatrix}$$

The *Hermite-Simpson method* is a 3-stages, 4th order IRK scheme. Its main equations are:

$$\mathbf{0} = \overline{\mathbf{q}}^{k+1} - \frac{1}{2}(\mathbf{q}^{k+1} + \mathbf{q}^k) - \frac{h_k}{8}(\mathbf{F}^{k+1} - \mathbf{F}^k) \qquad \text{Hermite interpolant}$$

$$\mathbf{0} = \mathbf{q}^{k+1} - \mathbf{q}^k - \frac{h_k}{6}\left(\mathbf{F}^{k+1} + 4\overline{\mathbf{F}}^{k+1} + \mathbf{F}^k\right) \qquad \text{Simpson quadrature}$$

$$\text{with} \quad \overline{\mathbf{F}}^{k+1} = \mathbf{F}(\overline{\mathbf{q}}^{k+1}, \overline{\mathbf{u}}_{k+1}, \theta^k + h_k/2)$$

where $\overline{\mathbf{q}}^{k+1}$ and $\overline{\mathbf{u}}^{k+1}$ are, respectively, the state and the control at the middle point between nodes k and $k+1$. Two different formulation arise, depending on whether the state at the middle points

is included in the optimization variable or not: said formulations are called Hermite-Simpson Separated (HSS) and Hermite-Simpson Compressed (HSC) respectively. Only the former is considered here, which leads to a $(8M-3)$-th dimensional optimization variable:

$$\mathbf{x}^T = \left(\theta_F, \mathbf{q}^1, \alpha^1, \overline{\mathbf{q}}^2, \overline{\alpha}^2, \mathbf{q}^2, \alpha^2, \ldots, \overline{\mathbf{q}}^M, \overline{\alpha}^M, \mathbf{q}^M, \alpha^M\right)$$

and the following set of 6 defect constraints per node (except for the last one):

$$\boldsymbol{\xi}_k(\mathbf{x}) = \begin{bmatrix} \overline{\mathbf{q}}^{k+1} - \frac{1}{2}(\mathbf{q}^{k+1} + \mathbf{q}^k) - \frac{h_k}{8}(\mathbf{F}^{k+1} - \mathbf{F}^k) \\ \mathbf{q}^{k+1} - \mathbf{q}^k - \frac{h_k}{6}\left(\mathbf{F}^{k+1} + 4\overline{\mathbf{F}}^{k+1} + \mathbf{F}^k\right) \end{bmatrix}$$

Consequently, the $6(M-1)$-th dimensional column vector of non-linear constraints derived from dynamics is:

$$\mathbf{c}(\mathbf{x})^{\mathrm{T}} = \begin{bmatrix} \boldsymbol{\xi}_1(\mathbf{x}) & \boldsymbol{\xi}_1(\mathbf{x}) & \ldots & \boldsymbol{\xi}_{M-1}(\mathbf{x}) \end{bmatrix}$$

REFERENCES

[1] J. Peláez, J. Hedo, and P. d. Andrés, "A special perturbation method in orbital dynamics," *Celestial Mechanics and Dynamical Astronomy*, Vol. 97, No. 2, 2007, p. 131150.

[2] G. Baù, C. Bombardelli, and J. Peláez, "A new set of integrals of motion to propagate the perturbed two-body problem," *Celestial Mechanics and Dynamical Astronomy*, Vol. 116, No. 1, 2013, pp. 53–78.

[3] G. Baù and C. Bombardelli, "Time Elements for Enhanced Performance of the Dromo Orbit Propagator," *The Astronomical Journal*, Vol. 148, No. 3, 2014, p. 43.

[4] C. Bombardelli, G. Baù, and J. Peláez, "Asymptotic solution for the two-body problem with constant tangential thrust acceleration," *Celestial Mechanics and Dynamical Astronomy*, Vol. 110, No. 3, 2011, pp. 239–256.

[5] C. Bombardelli, "Analytical formulation of impulsive collision avoidance dynamics," *Celestial Mechanics and Dynamical Astronomy*, Vol. 118, No. 2, 2014, pp. 99–114.

[6] J. L. Gonzalo and C. Bombardelli, "Asymptotic Solution for the Two Body Problem with Radial Perturbing Acceleration," *Advances in the Astronautical Sciences*, No. AAS 14-226, Santa Fe, New Mexico, USA, AAS/AIAA, January 26-30 2014.

[7] J. L. Gonzalo, "Perturbation Methods in Optimal Control Problems Applied to Low Thrust Space Trajectories," Master's thesis, ETSI Aeronáuticos, Technical University of Madrid (UPM), 2012.

[8] J. T. Betts, *Practical Methods for Optimal Control and Estimation Using Nonlinear Programming, Second Edition*. SIAM, 2010.

[9] H. Urrutxua, M. Sanjurjo-Rivo, and J. Peláez, "DROMO Propagator Revisited," *Advances in the Astronautical Sciences*, No. AAS 13-488, Kauai, Hawai, USA, 23rd AAS/AIAA Space Flight Mechanics Meeting Proceedings, February 2013.

[10] T. F. Coleman, B. S. Garbow, and J. J. Moré, "Software for Estimating Sparse Hessian Matrices," *ACM Transactions on Mathematical Software*, No. 11, 1985, pp. 363–377.