IAC-08-C1.2.5

A FAST SECOND-ORDER ALGORITHM FOR PRELIMINARY DESIGN OF LOW-THRUST TRAJECTORIES

Gregory Lantoine
Georgia Institute of Technology, Atlanta, Georgia, 30332-0150, USA
gregory.lantoine@gatech.edu

Ryan P. Russell
Georgia Institute of Technology, Atlanta, Georgia, 30332-0150, USA
ryan.russell@gatech.edu

ABSTRACT

Solar system exploration missions require demanding propulsion capabilities. From that point of view, low-thrust technology is increasingly considered since it uses propellant more efficiently. However, the optimization of the resulting trajectories is usually difficult and time consuming. In this paper, we develop a fast algorithm for preliminary design of low-thrust trajectories for spacecraft in a near Keplerian environment. The method achieves order of magnitude speed improvements at the expense of a minor accuracy reduction in the model fidelity. The approach is based on differential dynamic programming, a proven second-order technique that relies on Bellman's Principle of Optimality and successive minimization of quadratic approximations. Whereas traditional formulations require on expensive numerical integrations to solve the problem, our approach takes advantage of the well-known analytic partial derivatives of Keplerian motion to enable considerable faster computations. Preliminary numerical results are presented and compared to existing algorithms to illustrate the performance and the robustness of our approach. It is shown that our method provides similar results and is fast, easy to use, and enjoys small convergence sensitivities. It is therefore highly valued for preliminary design where large trade spaces have to be assessed rapidly.

Nomenclature

Φ^1	First-order state transition matrix
Φ^2	Second-order state transition matrix
$\mathbf{x}$	State vector
$\mathbf{X}$	Augmented state vector
$\Delta\mathbf{v}$	Velocity impulse vector
$\mathbf{F}$	Transition function vector
$\mathbf{p}$	Equality constraint vector
$\mathbf{h}$	Inequality constraint vector
m	Mass
g_0	Reference gravity acceleration value
I_{sp}	Specific impulse of the engine
$T_{\max}$	Maximum thrust of the engine
μ	Gravitational parameter
a	Semi-major axis
ψ	Universal variable
S_n	n^{th} universal function
E	Universal eccentric anomaly
F	Universal hyperbolic anomaly
f, g	Lagrange coefficients

Subscript

k	Node number
N	Total number of segments

Superscript

$-$	Quantity prior to impulse
$+$	Quantity after impulse

Conventions

$\mathbf{q}$	Vectors are bolded
δq	Small increment of the dummy variable q
J^*	Control-free quantity
J_q	First partial derivative of J w.r.t q
J_{qq}	Second partial derivative of J w.r.t q
$\dot{q}$	Partial derivative of q w.r.t time
$A \bullet B$	When A is an array and B is a second-order tensor: $(A \bullet B)_{ij} = \sum_p A(p)B(p,i,j)$

Acronyms

DDP	Differential Dynamic Programming
HDDP	Hybrid Differential Dynamic Programming
STM	State Transition Matrix
TOF	Time of Flight
JPL	Jet Propulsion Laboratory

Paper IAC-08-C1.2.5, 59th International Astronautical Congress, Glasgow, Scotland, Sep 29 - Oct 3, 2008.

INTRODUCTION

Due to its better efficiency compared to chemical propulsion, it is well known that continuous low-thrust propulsion offers substantial benefits for solar system exploration missions, like reduced spacecraft mass, increased payload capacity, or higher flexibility in trajectory design. With the increasing interest in low-thrust propulsion, optimization tools for feasibility analysis and preliminary design of the resulting trajectories are becoming more important to mission designers.

Those preliminary design tools must present characteristics desirable for the mission conceptual phase. First a short computational run is necessary to be able to explore a large design space and quickly assess a large number of potential trajectories. Secondly, convergence should be robust and not highly sensitive to initial guesses in order to explore automatically and efficiently the design space without strong expertise. In addition, the accuracy should be sufficient enough to not overlook any promising solutions.

However, optimization of low-thrust problems is numerically challenging and time-consuming since the thrust is operated for significant periods of the mission time, which leads to a large set of independent variables to optimize. To overcome this issue, a widespread strategy is to rely on analytical closed form expressions to avoid expensive numerical integrations. Many existing preliminary tools implement this idea. Originally developed by Boeing and later refined by Sauer at JPL, CHEBYTOP is based on an approximation of the problem using Chebychev polynomials.[1] This allows the optimal control problem to be solved in closed form. However CHEBYTOP suffers in accuracy because thrust constraints are not handled naturally and motion is restricted to the heliocentric sphere of influence. Another interesting tool, MIPELEC, was developed by CNES and uses differential equations averaged over each orbit.[2] It is averaging differential equations over one orbit, which yields analytical expressions for the mean state variables. However MIPELEC is limited to time-optimal problems and is best suited for very low-thrust trajectories with many revolutions. MALTO developed at JPL,[3] and GALLOP, developed at Purdue University,[4] use the well-proven technique of Sims and Flanagan[5] by modelling continuous thrusting as a series of impulses around a single gravitational body. This simplification of the problem can reduce dramatically the number of variables, and existing analytical results of the two-body problem can be advantageously exploited. The model is akin to using a low order euler integration scheme on the thrust- but a highly accurate integrration method on the keplerian motion. While n-body, oblateness, and other perturbations may be included, the model is best suited for near keplerian problems. Numerical comparisons show good agreement with high-fidelity tools.[6] This impulsive perturbation formulation of the problem is selected for this study and is described in detail in the next section.

The goal of this study is to combine the convergence and low dimension benefits of DDP methods with the speed and simplicity of the impulsive perturbation model. In MALTO and GALLOP, the trajectory structure leads to a constrained nonlinear programming (NLP) problem, which is solved directly using the nonlinear solver SNOPT.[7] However, even with the simplified formulation of the problem, the number of control variables grows as the square of the flight time. Therefore, the many revolution problem can be difficult to converge even with the sparse capabilities of SNOPT. The resulting increased dimensionality causes direct methods to become computationally prohibitive ('curse of dimensionality'). In addition, only first-order derivatives are used by SNOPT (second order derivatives are approximated at best), so convergence is slower than pure second-order methods.

To solve complex problems, differential dynamic programming (DDP) is a proven second-order technique based on Bellman's Principle of Optimality and successive minimization of quadratic approximations.[8,9] Like direct methods, DDP is known to be robust to poor initial guesses since it also includes a parameterization of the control variables. However, it is not as sensitive to the resulting high dimensional problem because DDP transforms this large problem into a succession of low dimensional subproblems. In this paper, we build upon a recently developed hybrid differential dynamic programming (HDDP) algorithm for robust low-thrust optimization.[10] It uses first- and second-order state transition matrices (STMs) to obtain the partial derivatives required for optimization, and combines DDP with nonlinear mathematical programming techniques to increase robustness and efficiency. In prelimnary testing, this method shows better convergence performance than traditional DDP formulation. But HDDP uses numerical integrations to obtain the required partials, which is computationally intensive. A problem with n states generally requires n^2 and n^3 additional equations to be integrated for the first- and second-order STMs. To make this algorithm more appropriate for preliminary design, we extend it by employing the analytic STMs of the two-body problem.

In this paper, we intend to demonstrate the value of using keplerian STMs in the optimization process.

Perturbations such as thrust, n-body, or oblateness effects can be included approximately to make this algorithm more general. Comparisons are made with unmodified HDDP. The results strongly support the utility of the analytic state transition matrices whenever possible.

PROBLEM FORMULATION

Discretization scheme

In Ref. 10, HDDP employs a piecewise constant control formulation. It is an accurate model but partial derivatives cannot be computed analytically using that scheme. Therefore we select a less accurate discretization scheme where the trajectory is subdivided into segments consisting of an impulse followed by a coast period. The impulsive Δv's are the controls of this model and approximate the low thrust acceleration over a segment. In addition to a dramatic reduction in the number of variables and the availability of analytic partial derivatives, another advantage of this formulation is that sensitivities with respect to the impulses are the same as those with respect with the state velocity components. Therefore, unlike HDDP (or traditional DDP), we do not require extra equations of motion for the control sensitivities.

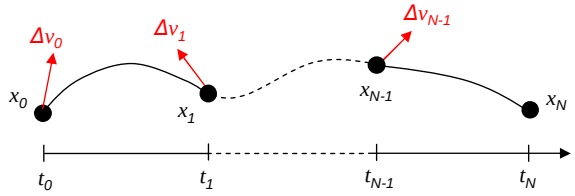

Figure 1: Impulsive discretization scheme.

The state $\mathbf{x}_k$ of the spacecraft at a node point is given by its position $\mathbf{r}_k$ and velocity $\mathbf{v}_k$. Each impulse Δv_k leads to a velocity discontinuity before and after the impulse given by $\mathbf{v}_k^+ = \mathbf{v}_k^- + \Delta\mathbf{v}_k$. Also, since we will see that the mass of the spacecraft appears in some equations, it must be computed alongside the trajectory, but is not part of the state vector (a further reduction in the problem dimension). The mass discontinuity due to the impulse is obtained from the rocket equation:

$$m_k^+ = m_k^- \exp(-\frac{\Delta v_k}{g_0 Isp}) \tag{1}$$

Dynamics

The mapping between the states that form the segment boundaries is defined on each segment by a transition function F_k :

$$\mathbf{x}_{k+1} = F_k(\mathbf{x}_k, \Delta\mathbf{v}_k) \tag{2}$$

To reduce computational time, the coast arcs are computed analytically using two-body mechanics with respect to a primary body using a standard Kepler solver through the "f and g" procedure presented by Bate et al.[11] If the position and velocity are known at a given instant, then the position and velocity at any later time are found in terms of a linear combination of the initial values. Therefore we can get a closed form expression of the transition function:

$$\mathbf{x}_{k+1} = \begin{bmatrix}\mathbf{r}_{k+1}\\ \mathbf{v}_{k+1}^-\end{bmatrix} = \begin{bmatrix}f\mathbf{r}_k + g(\mathbf{v}_k^- + \Delta\mathbf{v}_k)\\ \dot{f}\mathbf{r}_k + \dot{g}(\mathbf{v}_k^- + \Delta\mathbf{v}_k)\end{bmatrix} = F_k(\mathbf{x}_k, \Delta\mathbf{v}_k) \tag{3}$$

The Lagrange coefficients f, g and their time derivatives in these expressions are functions of initial conditions and change in anomaly. Because the independent variable is time, the final solution requires iteration. We use the classic Newton-Raphson method.[11]

Since $\Delta\mathbf{v}_k$ is part of the transition function, $\mathbf{v}_k^-$ is defined from now on to be the value of the velocity at node k and superscript '-' can be dropped. Examples of the type and scope of the problems that can be solved using this model include all unperturbed and perturbed keplerian trajectories about a single celestial body for orbit transfers, rendezvous, intercepts, arrival and capture, departure and escape. We will see later how to account for perturbations.

Objective function

Following the HDDP framework, the objective function to be minimized is defined by:

$$J = \sum_{k=0}^{N-1} L_k(\mathbf{x}_k, \Delta\mathbf{v}_k) + \varphi(\mathbf{x}_N) \tag{4}$$

For minimum fuel problems, we want to minimize the total sum of the impulsive maneuvers, so $L_k = \Delta v_k$ and $\varphi = 0$. For minimum time problems, $L_k = 0$ and $\varphi = t_f - t_0$.

Constraints

For each node $k = 0...N$, constraints are written in the general form p_k and h_k to account for path and control constraints. Terminal constraints are also included:

$$\begin{aligned} p_k(x_k, \Delta\mathbf{v}_k) &= 0\\ h_k(x_k, \Delta\mathbf{v}_k) &\leq 0 \end{aligned} \tag{5}$$

One important example for h_k is the limitation of the magnitude of the impulse by the total amount of Δv that could be produced by the low-thrust engine over the duration of the segment. This is to ensure that the impulse discretization scheme models accurately the corresponding low-thrust propulsion system.

$$\Delta v_k \leq \Delta v_{\max,k} = \frac{T_{\max}}{\bar{m_k}} \Delta t \tag{6}$$

This formula slightly underestimates the maximum velocity impulse since only the mass at the beginning of the segment is used, while it should be linearly decreasing over the segment. This expression therefore leads to a more conservative trajectory design, but feasibility is guaranteed.

Summary

The optimal control problem we are considering is reduced to the following form:

$$\min_{\Delta V_0,\ldots,\Delta V_{N-1}} J = \sum_{k=0}^{N-1} L_k(\mathbf{x}_k, \Delta\mathbf{v}_k) + \varphi(\mathbf{x}_N)$$
$$\text{s.t.} \begin{cases} \mathbf{x}_{k+1} = F_k(\mathbf{x}_k, \Delta\mathbf{v}_k) & k = 0...N-1 \\ p_k(\mathbf{x}_k, \Delta\mathbf{v}_k) = 0 & k = 0...N \\ h_k(\mathbf{x}_k, \Delta\mathbf{v}_k) \leq 0 & k = 0...N \end{cases} \tag{7}$$

This formulation has a dynamic structure that is well adapted to apply Bellman's Principle of Optimality which describes the process of solving problems involving time successive decision making. An approach for the solution using the recently introduced HDDP algorithm[10] is overviewed in the next section.

OVERVIEW OF HDDP

This method uses dynamic programming locally around a current trajectory to generate an optimal feedback control law. Being a second-order method, it relies on quadratic Taylor series expansions about the nominal solution. After initializing the problem once with an initial guess, each iteration can be divided in two parts: the backward and forward sweeps*. In the backward sweep, coefficients of the control law that update the current thrust control policy are determined through the computations of the sensitivities of the objective function with respect to small variations of the control sequence. In the forward sweep, the new states and objective value of the system are calculated, alongside with state transition matrices.

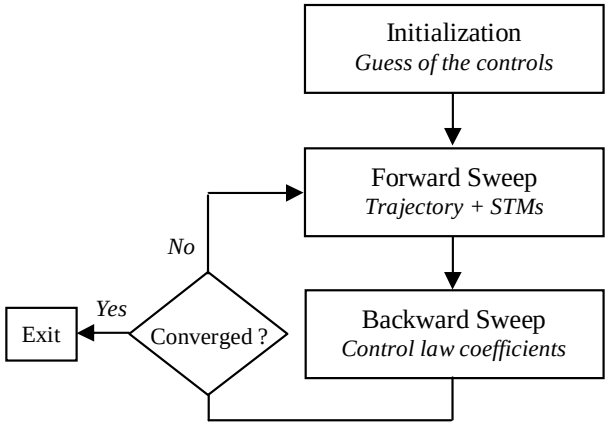

Figure 3: Main steps of HDDP.

In all DDP algorithms, the objective is approximated by a quadratic Taylor series expansion. Noting that the impulse $\delta\mathbf{v}$ is the control term, the change in the objective is expressed up to second order as:

$$\delta J_k \approx J_{x,k}^T \delta\mathbf{x}_k + J_{\Delta v,k}^T \delta(\Delta\mathbf{v}_k) + \delta\mathbf{x}_k^T J_{x\Delta v,k} \delta(\Delta\mathbf{v}_k)$$
$$+\frac{1}{2}\delta(\Delta\mathbf{v}_k)^T J_{\Delta v\Delta v,k} \delta(\Delta\mathbf{v}_k) + \frac{1}{2}\delta\mathbf{x}_k^T J_{xx,k} \delta\mathbf{x}_k \tag{8}$$

One specificity of HDDP is the derivation of the coefficients of this quadratic approximation with the help of the first-order and second-order state transition matrices. STMs are useful tools for our problem because they enable mapping of the perturbations in the spacecraft state from one time to another. The main recursive equations of the HDDP backward sweep are the following:

$$J_{x,k}^T = L_{x,k}^T + J_{x,k+1}^{*T} \Phi_k^1 \tag{9a}$$
$$J_{xx,k} = L_{xx,k} + \Phi_k^{1T} J_{xx,k+1}^* \Phi_k^1 + J_{x,k+1}^{*T} \bullet \Phi_k^2 \tag{9b}$$

Note that contrary to Ref. 10, we map only the partial derivatives with respect to the states since the ones with respect to the controls can be deduced easily from those (see Eq. 10). If spherical coordinates are used to represent controls, derivatives have to be modified using the chain rule from transformation relationship between cartesian and spherical coordinates.

$$J_{\Delta v,k} = J_{x,k}(4:6)$$
$$J_{\Delta v\Delta v,k} = J_{xx,k}(4:6, 4:6)$$
$$J_{x\Delta v,k} = J_{xx,k}(1:6, 4:6) \tag{10}$$

Making the gradient of Eq. 8 with respect to the control vanish, we get the usual control law given in Eq. 11. This control law should be modified if

*Here, the term sweep is used to indicate an iteration over all time steps.

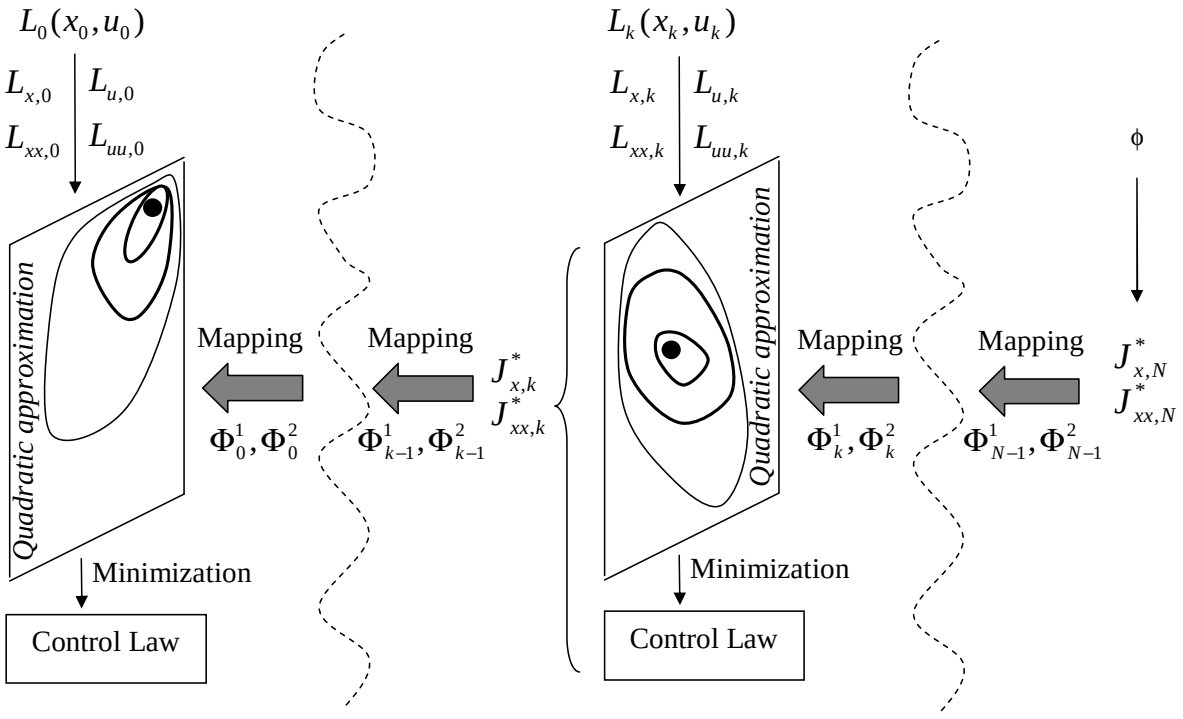

Figure 2: General procedure to generate required derivatives ($u = \Delta\mathbf{v}$).

$J_{\Delta v \Delta v,k}$ is not positive definite or if constraints are present. We omit description of the different techniques for the conciseness of the paper and interested readers may refer to Ref. 10 for detailed theory.

$$\delta(\Delta v_k) = -J^{-1}_{\Delta v \Delta v,k} J_{\Delta v,k} - J^{-1}_{\Delta v \Delta v,k} J_{\Delta v x,k} \delta x_k \quad (11)$$

Finally, to be able to use Eq. 9a and Eq. 9b recursively, control-dependant derivatives of the quadratic approximation are replaced with their control-free counterparts by using Eq. 11.

$$J^*_{x,k} = J_{x,k} - J^T_{\Delta v,k} J^{-1}_{\Delta v \Delta v,k} J_{\Delta v x,k} \quad (12a)$$

$$J^*_{xx,k} = J_{xx,k} - J^T_{\Delta v x,k} J^{-1}_{\Delta v \Delta v,k} J_{\Delta v x,k} \quad (12b)$$

Figure 2 summarizes the backward sweep procedure. One disadvantage of HDDP is that it requires the computation of the STMs by integrating a set of $n^2 + n^3$ equations, which is time consuming (note that the second order STM can be reduced to the dimension $(n^3 + n^2)/2$ if symmetries of the Hessian are considered). Fortunately, for the specific problem formulation described in section 2, it is possible to obtain analytic expressions of the STMs.

ANALYTIC EXTENSION

Modifications in the HDDP algorithm are discussed to take advantage of the existing two-body analytic state transition matrices. The only effects considered in this analytic method are the inverse square law gravity forces. Perturbations are treated in a particular fashion in the last subsection. The analytic method relies heavily on the f and g solution of the Kepler problem found in Eq. 3. We emphasize that the universal formulation allows us to handle hyperbolic trajectories without modification although the examples we consider later are elliptical.

Keplerian STMs

The analytic forst-order Keplerian state transition matrix developed by Goodyear,[12] Battin[13] and others[14–17] is well known. It is computed via universal variables to obtain the derivatives of the Lagrangian representation coefficients of Eq. 3, since:

$$\begin{bmatrix} \delta\mathbf{r}_{k+1} \\ \delta\mathbf{v}_{k+1} \end{bmatrix} = \begin{bmatrix} \delta f I & \delta g I \\ \delta \dot{f} I & \delta \dot{g} I \end{bmatrix} \begin{bmatrix} \mathbf{r}_0 \\ \mathbf{v}_0 \end{bmatrix} + \begin{bmatrix} f I & g I \\ \dot{f} I & \dot{g} I \end{bmatrix} \begin{bmatrix} \delta\mathbf{r}_0 \\ \delta\mathbf{v}_0 \end{bmatrix} \quad (13)$$

Pitkin builds upon this formulation to obtain both first- and second-order STMs.[18] In our algorithm we follow the same process as Pitkin. The details of the derivation are tedious and will not be reproduced here. There are minor differences to make the computations more efficient. The universal variable is found from the solution of the Kepler problem. Also, we prefer to use the closed form expression of the universal functions given by Goodyear[12] instead of truncated series as proposed by Pitkin.

Table 1 details the dramatic computational gains achieved using the analytic STMs. An improvement of three orders of magnitude can be obtained for the STM computations. Figure 4 shows that the main computational burdens for numerical HDDP are the STM computations and the trajectory integration while the remaining calculations are negligible. On the other hand, using an analytic approach allows a relatively even distribution of the computational burden amongst the trajectory, STM, and sweep calculations. Note that the trajectory computation is non-trivial despite its low dimension due to the iterations required to solve Keplers problem.

Steps	Analytic HDDP	Num. HDDP
STM Comp.	0.38 s	186 s
Traj. Comp.	0.32 s	23.3 s
Back. Sweep	0.16 s	0.14 s
Other	0.05 s	0.05 s

Table 1: Execution times of HDDP steps in Matlab for a representative problem using 150 nodes.

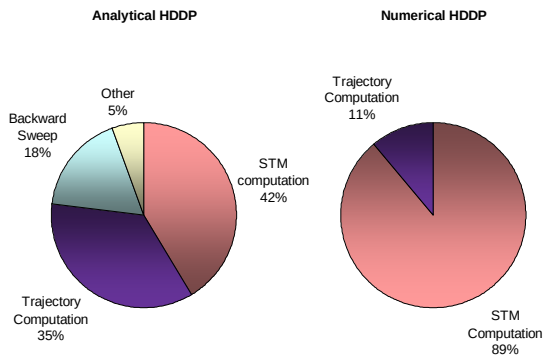

Figure 4: Execution time contributions.

Time derivatives

This subsection presents a method to solve the problem with variable beginning and ending time. Since our approach does not require any integrations, no normalization of time is needed. First, to obtain derivatives with respect to time of flight, the states are augmented with this new variable:

$$\mathbf{X} = \begin{bmatrix} \mathbf{r} \\ \mathbf{v} \\ TOF \end{bmatrix} \tag{14}$$

The STMs are augmented with the corresponding extra derivatives. Eq. 15, Eq. 16a and Eq. 16b show the augmented first- and second-order STMs.

$$\Phi^1 = \begin{bmatrix} \Phi^1_{rr} & \Phi^1_{rv} & \Phi^1_{rt} \\ \Phi^1_{vr} & \Phi^1_{vr} & \Phi^1_{vt} \\ 0_{1\times 3} & 0_{1\times 3} & 1 \end{bmatrix} \tag{15}$$

$$\Phi^2_r = \begin{bmatrix} \Phi^2_{rrr} & \Phi^2_{rrv} & \Phi^2_{rrt} \\ \Phi^2_{rvr} & \Phi^2_{rvr} & \Phi^2_{rvt} \\ 0_{1\times 3} & 0_{1\times 3} & 0 \end{bmatrix} \tag{16a}$$

$$\Phi^2_v = \begin{bmatrix} \Phi^2_{vrr} & \Phi^2_{vrv} & \Phi^2_{vrt} \\ \Phi^2_{vvr} & \Phi^2_{vvr} & \Phi^2_{vvt} \\ 0_{1\times 3} & 0_{1\times 3} & 0 \end{bmatrix} \tag{16b}$$

The missing derivatives with respect to time of flight are found by differentiating with respect to time the expressions of Pitkin. Detailed expressions of the derivatives of the representation coefficients are given in the Appendix.

$$\Phi^1_{rt} = \dot{f}\mathbf{r}_k + \dot{g}\mathbf{v}_k \tag{17a}$$

$$\Phi^1_{vt} = \ddot{f}\mathbf{r}_k + \ddot{g}\mathbf{v}_k \tag{17b}$$

$$\Phi^2_{rrt} = \frac{\partial \dot{f}}{\partial r}\mathbf{r}_k + \frac{\partial \dot{g}}{\partial r}\mathbf{v}_k + \dot{f}I \tag{17c}$$

$$\Phi^2_{rvt} = \frac{\partial \dot{f}}{\partial v}\mathbf{r}_k + \frac{\partial \dot{g}}{\partial v}\mathbf{v}_k + \dot{g}I \tag{17d}$$

$$\Phi^2_{vrt} = \frac{\partial \ddot{f}}{\partial r}\mathbf{r}_k + \frac{\partial \ddot{g}}{\partial r}\mathbf{v}_k + \ddot{f}I \tag{17e}$$

$$\Phi^2_{vvt} = \frac{\partial \ddot{f}}{\partial v}\mathbf{r}_k + \frac{\partial \ddot{g}}{\partial v}\mathbf{v}_k + \ddot{g}I \tag{17f}$$

On the other hand, initial time derivatives are easy to obtain using a simple chain rule:

$$J_{t_0} = J_X(t_0)^T \frac{\partial \mathbf{X}_0}{\partial t_0} \tag{18a}$$

$$J_{Xt_0} = J_{XX}(t_0)\frac{\partial \mathbf{X}_0}{\partial t_0} \tag{18b}$$

$$J_{t_0t_0} = \frac{\partial \mathbf{X}_0}{\partial t_0}^T J_{XX}(t_0)\frac{\partial \mathbf{X}_0}{\partial t_0} \tag{18c}$$

where $\frac{\partial \mathbf{X}_0}{\partial t_0} = \begin{bmatrix} \frac{\partial \mathbf{r}_0}{\partial t_0} \\ \frac{\partial \mathbf{v}_0}{\partial t_0} \\ 0 \end{bmatrix}$.

Most orbit problems by nature are highly nonlinear in the initial and final times. Therefore, care must be taken to maintain small trust regions in the time dimensions of the problem in order to avoid overstepping the quadratic region of validity from iteration to iteration.

Perturbations

Like thrust, effects of perturbation forces $\mathbf{f}_p$ can be approximated by a series of impulses at each node to account for the perturbing acceleration over the whole segment. The transition function and the STMs are modified accordingly.

$$\Delta \mathbf{v}_{P,k} \approx \mathbf{f}_P(\mathbf{x}_k, t_k)\Delta t \tag{19}$$

NUMERICAL RESULTS

Variable Time Earth-Mars Transfer

An example problem for a simple Earth-Mars transfer is presented to test the performance of our algorithm. Numerical data used for the transfer are given in Table 1 and are taken from Oberle[19] who studied Earth-Mars transfers intensively using his software BNDSCO. It is a multiple shooting indirect method that relies on optimal control theory to solve effectively boundary-value problems, yielding accurate solutions without discretization.[20]

Parameter	Value
m_0	679.78 kg
$t_{f\,guess}$	300 days
$g_0 I_{sp}$	5.5809×10^4 m/s
$T_{\max}$	0.56493 N

Table 2: Data of the Earth-Mars problem

A fixed equally-spaced mesh of 50 nodes is used. The final time is left free to test the expressions of the time derivatives provided in the previous section. Initial position is at Earth and Mars orbit is targeted. The initial guess for each impulsive δv is zero and the flight time guess is that of a hohman transfer. The solution converges easily and we can see in Figure 5 that analytic HDDP results match well with those of standard HDDP and BNDSCO. The switching structure is reproduced a little less accurately because of the more approximate discretization with impulses. The optimized final time is 297 days and is consistent with the result of Oberle (296.5 days). The next example demonstrates the power of our method in the context of using a high-resolution mesh grid.

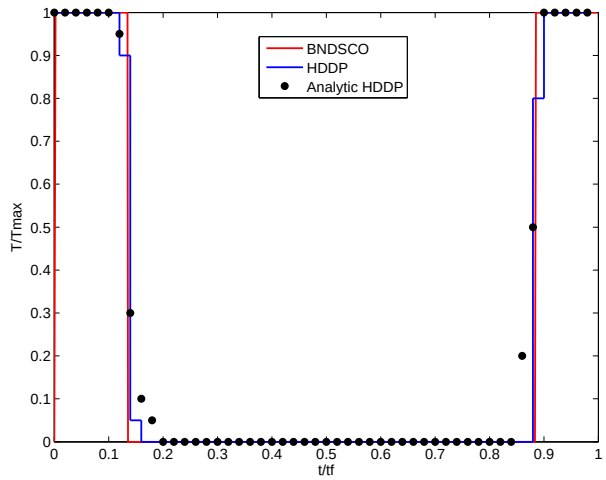

Figure 5: Time evolution of thrust.

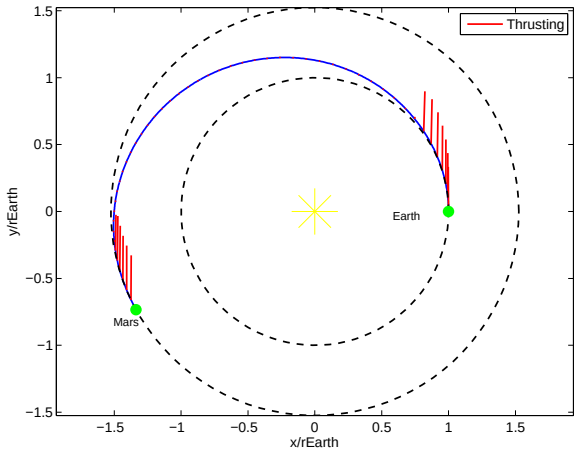

Figure 6: Trajectory of the Earth-Mars transfer.

Multi-rev Orbital Transfer

Finally, we examine a more complicated problem about the minimum fuel optimization of a multi-revolution, low-thrust, circle-to-circle, planar orbital transfer between LEO and GEO orbits. Numerical data used for the transfer are given in table 2. Time of flight is fixed and is chosen to be long enough to allow multi-rev optimal trajectories.

Parameter	Value
r_0	20000 km
r_f	42000 km
m_0	1000 kg
t_f	4 days
I_{sp}	2000 s
$T_{\max}$	5 N

Table 3: Data of the orbital transfer problem

A fixed equally-spaced mesh of 700 nodes is used. This large number of nodes is enabled by the speed of the algorithm and the fact that DDP methods successively solve multiple small problems as opposed to a single large dimensioned problem. For the initial guess, we provide a simple feasible, spiral trajectory where each impulse is applied tangent to the velocity. Results are compared with another recent second-order feedback control algorithm[21] termed as the Modified Gradient, and with the indirect solver T3D dedicated to orbital transfers.[22] Since T3D is an indirect method that does not discretize controls, it gives "exact" locally optimal solutions. The solution produced by T3D is therefore considered as the benchmark solution.

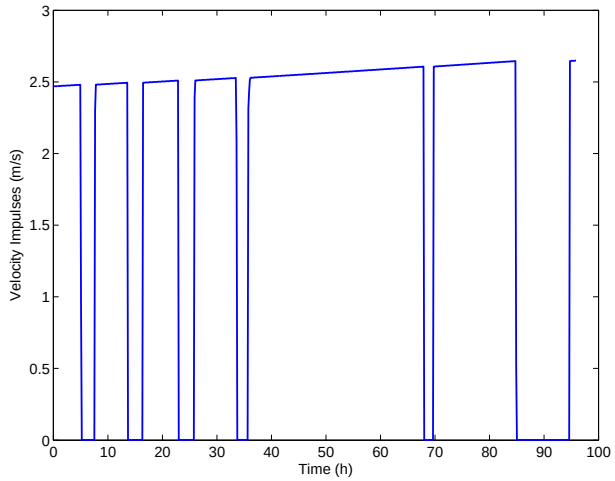

Figure 7: Evolution of Δv impulses with Analytic HDDP.

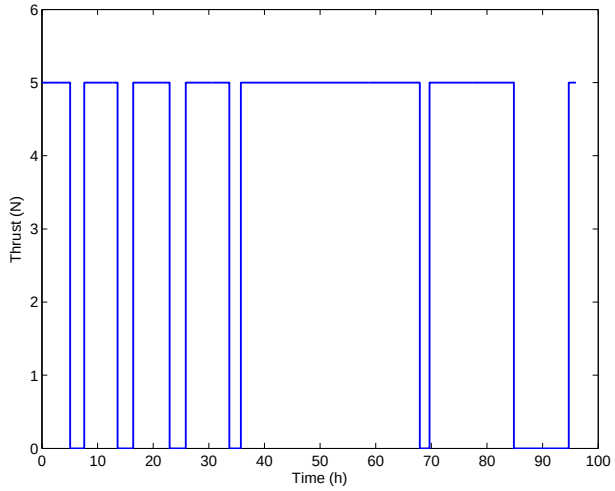

Figure 8: Thrust evolution with T3D.

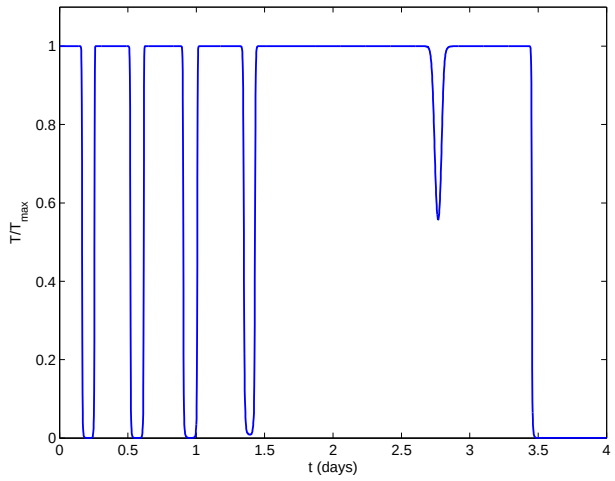

Figure 9: Thrust evolution with Modified Gradient.

Figure 7 to Figure 9 compare the thrust structure of the Analytic HDDP solution with the ones of T3D and Modified Gradient. Despite the complexity of the structure with multiple bangs, our solution agrees with T3D more closely than Modified Gradient. The improved accuracy of the Analytic HDDP method is achievable because we can use a high-resolution grid and perform many iterations. However, note that Modified Gradient uses zero-thrust as an initial guess to find solution. In this example, when the zero thrust guess is provided to the Analytic HDDP, the algorithm encounters no problems walking towards a converged solution. However, the solution is a different local minimum with a slightly less favorable final mass.

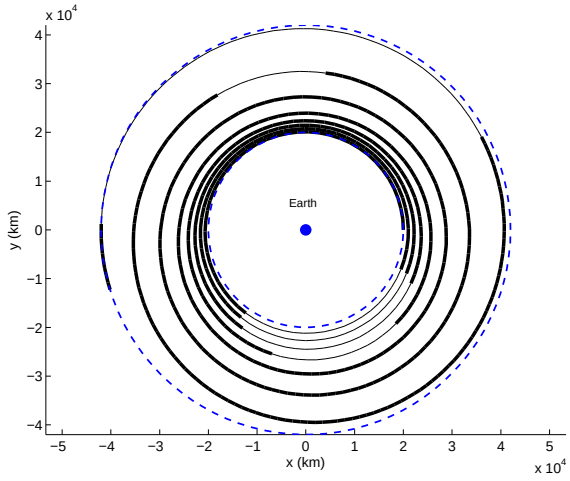

Figure 10: Trajectory of the LEO-to-GEO transfer.

CONCLUSION

We presented an innovative low thrust trajectory optimization and analysis tool intended for preliminary mission feasibility studies. Its strength lies in its fast computational speed achieved because the method eliminates the need for time-consuming numerical integration and provides for rapid computation of the derivatives of the objective. The method is local and gradient based; therefore, the solution is inherently dependent on the initial guess. However, the method retains the robust convergence qualities as is typical with all DDP based methods. Accordingly, we demonstrate robust convergence for a traditionally difficult multi-rev problem when starting from trivially simple initial guesses.

We suggest our tool to be used as a basis at early stages of space mission design when high accuracy is not required or as an initial approximation for finding an exact solution. Increased accuracy can be obtained by numerically integrating parts of the trajectory under non-Keplerian influence. In addition, for pure two-body problems, high accuracy can be achieved with high resolution grids thanks to the

speed of the algorithm. Complex two-body problems like multi-revolution trajectories can also become manageable.

In future work we intend to test this algorithm on a larger spectrum of representative examples. In addition, gravity-assist trajectories can be implemented by treating the flyby body as a zero-point mass and modeling the effect of the flyby as a discrete impulse. Finally, we plan to make direct comparisons with state-of-the-art solvers like SNOPT.

APPENDIX

Following the work of Pitkin, the analytical expressions of the derivatives of the Lagrange representation coefficients f and g are given here:

$$\ddot{f} = -\frac{\mu}{r_k}\left(\frac{\dot{S}_1}{r_{k+1}} - \frac{S_1\dot{r}_{k+1}}{r_{k+1}^2}\right) \tag{20a}$$

$$\ddot{g} = -\mu\left(\frac{\dot{S}_2}{r_{k+1}} - \frac{S_2\dot{r}_{k+1}}{r_{k+1}^2}\right) \tag{20b}$$

$$\frac{\partial \dot{f}}{\partial r} = -\frac{\dot{f}\left(r_{k+1}\frac{\partial r_k}{\partial r} + r_k\frac{\partial r_{k+1}}{\partial r}\right) + \mu\frac{\partial S_1}{\partial r}}{r_k r_{k+1}} \tag{20c}$$

$$\frac{\partial \dot{f}}{\partial v} = -\frac{\dot{f}\left(r_{k+1}\frac{\partial r_k}{\partial v} + r_k\frac{\partial r_{k+1}}{\partial v}\right) + \mu\frac{\partial S_1}{\partial v}}{r_k r_{k+1}} \tag{20d}$$

$$\frac{\partial \dot{g}}{\partial r} = \frac{(1-\dot{g})\frac{\partial r_{k+1}}{\partial r} - \mu\frac{\partial S_2}{\partial r}}{r_{k+1}} \tag{20e}$$

$$\frac{\partial \dot{g}}{\partial v} = \frac{(1-\dot{g})\frac{\partial r_{k+1}}{\partial v} - \mu\frac{\partial S_2}{\partial v}}{r_{k+1}} \tag{20f}$$

$$\begin{aligned}\frac{\partial \ddot{f}}{\partial r} =& \frac{\mu}{r_k^2}\frac{\partial r_k}{\partial r}\left(\frac{\dot{S}_1}{r_{k+1}} - \frac{S_1\dot{r}_{k+1}}{r_{k+1}^2}\right) \\ &+ \frac{\mu}{r_k}\left(\frac{1}{r_{k+1}}\frac{\partial \dot{S}_1}{\partial r} - \frac{\dot{S}_1}{r_{k+1}^2}\frac{\partial r_{k+1}}{\partial r}\right. \\ &\left. - \frac{\partial S_1}{\partial r}\frac{\dot{r_{k+1}}}{r_{k+1}^2} - \frac{S_1}{r_{k+1}^2}\frac{\partial \dot{r}_{k+1}}{\partial r} + \frac{2S_1\dot{r}_{k+1}}{r_{k+1}^3}\right)\end{aligned} \tag{20g}$$

$$\begin{aligned}\frac{\partial \ddot{f}}{\partial v} =& \frac{\mu}{r_k}\left(\frac{1}{r_{k+1}}\frac{\partial \dot{S}_1}{\partial v} - \frac{\dot{S}_1}{r_{k+1}^2}\frac{\partial r_{k+1}}{\partial v} - \frac{\partial S_1}{\partial v}\frac{\dot{r}_{k+1}}{r_{k+1}^2}\right. \\ &\left. - \frac{S_1}{r_{k+1}^2}\frac{\partial \dot{r}_{k+1}}{\partial v} + \frac{2S_1\dot{r}_{k+1}}{r_{k+1}^3}\right)\end{aligned} \tag{20h}$$

$$\begin{aligned}\frac{\partial \ddot{g}}{\partial r} =& -\frac{\mu}{r_{k+1}}\left(\frac{\partial \dot{S}_2}{\partial r} - \frac{\dot{S}_2}{r_{k+1}}\frac{\partial r_{k+1}}{\partial r} - \frac{\partial S_2}{\partial r}\frac{\dot{r}_{k+1}}{r_{k+1}}\right. \\ &\left. - \frac{S_2}{r_{k+1}}\frac{\partial \dot{r}_{k+1}}{\partial r} + \frac{2S_2\dot{r}_{k+1}}{r_{k+1}^2}\right)\end{aligned} \tag{20i}$$

$$\begin{aligned}\frac{\partial \ddot{g}}{\partial v} =& -\frac{\mu}{r_{k+1}}\left(\frac{\partial \dot{S}_2}{\partial v} - \frac{\dot{S}_2}{r_{k+1}}\frac{\partial r_{k+1}}{\partial v} - \frac{\partial S_2}{\partial v}\frac{\dot{r}_{k+1}}{r_{k+1}}\right. \\ &\left. - \frac{S_2}{r_{k+1}}\frac{\partial \dot{r}_{k+1}}{\partial v} + \frac{2S_2\dot{r}_{k+1}}{r_{k+1}^2}\right)\end{aligned} \tag{20j}$$

The derivatives of the radius are also required:

$$\dot{r_{k+1}} = r_1\dot{S}_0 + \sigma_0\dot{S}_1 + \mu\dot{S}_2 \tag{21a}$$

$$\begin{aligned}\frac{\partial \dot{r}_{k+1}}{\partial r} =& \dot{S}_0\frac{\partial r_k}{\partial r} + \dot{S}_1\frac{\partial \sigma_0}{\partial r} + \left(r_k\frac{\partial \dot{S}_0}{\partial \alpha} + \sigma_0\frac{\partial \dot{S}_1}{\partial \alpha}\right. \\ &\left. + \mu\frac{\partial \dot{S}_2}{\partial \alpha}\right)\frac{\partial \alpha}{\partial r} + (\sigma_0\dot{S}_0 \\ &+ (\mu + \alpha r_k)\dot{S}_1)\frac{\partial \psi}{\partial r} + (\sigma_0 S_0 \\ &+ (\mu + \alpha r_k)S_1)\frac{\partial \dot{\psi}}{\partial r}\end{aligned} \tag{21b}$$

$$\begin{aligned}\frac{\partial \dot{r}_{k+1}}{\partial v} =& \dot{S}_1\frac{\partial \sigma_0}{\partial v} + \left(r_k\frac{\partial \dot{S}_0}{\partial \alpha} + \sigma_0\frac{\partial \dot{S}_1}{\partial \alpha}\right. \\ &\left. + \mu\frac{\partial \dot{S}_2}{\partial \alpha}\right)\frac{\partial \alpha}{\partial v} + (\sigma_0\dot{S}_0 \\ &+ (\mu + \alpha r_k)\dot{S}_1)\frac{\partial \psi}{\partial v} + (\sigma_0 S_0 \\ &+ (\mu + \alpha r_k)S_1)\frac{\partial \dot{\psi}}{\partial v}\end{aligned} \tag{21c}$$

$$\frac{\partial r_k}{\partial r} = \frac{\mathbf{r}_k}{r_k} \tag{21d}$$

Other parameters are defined by:

$$\sigma_0 = \mathbf{r}_k^T\mathbf{v}_k, \quad \frac{\partial \sigma_0}{\partial r} = \mathbf{v}_k, \quad \frac{\partial \sigma_0}{\partial v} = \mathbf{r}_k \tag{22a}$$

$$\alpha = -\mu/a, \quad \frac{\partial \alpha}{\partial r} = 2\mu\frac{\mathbf{r}_k}{r_k^3}, \quad \frac{\partial \alpha}{\partial v} = 2\mathbf{v}_k \tag{22b}$$

Universal functions S_0 and S_1 depend on universal variable ψ obtained from Kepler solution:

If $a > 0$:

$$\Delta E = \psi\sqrt{\mu/a} \tag{23a}$$

$$\dot{\Delta E} = \sqrt{\mu/a}/r_{k+1} \tag{23b}$$

$$S_0 = cos(\Delta E) \tag{23c}$$

$$S_1 = \frac{sin(\Delta E)}{\mu/a} \tag{23d}$$

If $a < 0$:

$$\Delta F = \psi\sqrt{-\mu/a} \tag{24a}$$

$$\dot{\Delta F} = \sqrt{-\mu/a}/r_{k+1} \tag{24b}$$

$$S_0 = cosh(\Delta F) \tag{24c}$$

$$S_1 = \frac{sinh(\Delta F)}{-\mu/a} \tag{24d}$$

Higher order universal functions satisfy:

$$S_n = \alpha S_{n+2} + \psi^n/n! \tag{25}$$

The derivatives of the universal functions w.r.t. α are for $n = 1, 2, 3$:

$$\frac{\partial S_n}{\partial \alpha} = (\psi S_{n+1} - nS_{n+2})/2 \tag{26a}$$

$$\frac{\partial S_0}{\partial \alpha} = S_2 + \alpha\frac{\partial S_2}{\partial \alpha} \tag{26b}$$

Derivatives of universal functions w.r.t. time are found using Eq. 23b, Eq. 24b, Eq. 25 and the chain rule. Derivatives of universal variable found in Eq. 21b and Eq. 21c are:

$$\frac{\partial \psi}{\partial r} = -\frac{(r_k \frac{\partial S_1}{\partial \alpha} + \sigma_0 \frac{\partial S_2}{\partial \alpha} + \mu \frac{\partial S_3}{\partial \alpha}) \frac{\partial \alpha}{\partial r} + S_1 \frac{\partial r_k}{\partial r} + S_2 \frac{\partial \sigma_0}{\partial r}}{r_k + 1} \tag{27a}$$

$$\frac{\partial \psi}{\partial v} = -\frac{(r_k \frac{\partial S_1}{\partial \alpha} + \sigma_0 \frac{\partial S_2}{\partial \alpha} + \mu \frac{\partial S_3}{\partial \alpha}) \frac{\partial \alpha}{\partial v} + S_1 \frac{\partial r_k}{\partial v} + S_2 \frac{\partial \sigma_0}{\partial v}}{r_k + 1} \tag{27b}$$

Derivatives of universal functions w.r.t. states are found using Eq. 27a, Eq. 27b, Eq. 25 and the chain rule.

References

[1] D. W. Hahn, B. F. Itzen, and F. T. Johnson, "Chebychev trajectory optimization program /chebytop/ final report," technical report nasa-cr-73359, NASA, July 1969.

[2] R. Epenoy and S. Geffroy, "Optimal low-thrust transfers with constraints - generalization of averaging techniques," *Acta Astronautica*, vol. 41, no. 3, pp. 133–149, 1997.

[3] J. A. Sims, P. Finlayson, E. Rinderle, M. Vavrina, and T. Kowalkowski, "Implementation of a low-thrust trajectory optimization algorithm for preliminary design." No. AIAA-2006-674, Aug. 2006. AAS/AIAA Astrodynamics Specialist Conference and Exhibit, Keystone, CO.

[4] T. T. Mcconaghy, T. J. Debban, A. E. Petropoulos, and J. M. Longuski, "Design and optimization of low-thrust trajectories with gravity assists," *Journal of spacecraft and rockets*, vol. 40, no. 3, pp. 380–387, 2003.

[5] J. A. Sims and S. N. Flanagan, "Preliminary design of low-thrust interplanetary missions." No. AAS 99-338, Aug. 1999. AAS/AIAA Astrodynamics Specialist Conference, Girdwood, Alaska.

[6] T. Polsgrove, L. Kos, R. Hopkins, and T. Crane, "Comparison of performance predictions for new low- thrust trajectory tools." No. AIAA 2006-6742, Aug. 2006. AAS/AIAA Astrodynamics Specialist Conference and Exhibit, Keystone, CO.

[7] P. E. Gill, W. Murray, and M. A. Saunders, "SNOPT: An SQP algorithm for large-scale constrained optimization," *SIAM journal on optimization*, vol. 12, no. 4, pp. 979–1006, 2002.

[8] D. H. Jacobson and D. Q. Mayne, *Differential Dynamic Programming*. New York, N.Y.: Elsevier Scientific, 1970.

[9] D. Q. Mayne, "A second-order gradient method for determining optimal control of non-linear discrete time systems," *International Journal of Control*, vol. 3, pp. 85–95, 1966.

[10] G. Lantoine and R. P. Russell, "A hybrid differential dynamic programming algorithm for robust low-thrust optimization." No. AIAA 2008-6615, Aug. 2008. AIAA/AAS Astrodynamics Specialist Conference and Exhibit, Honolulu, Hawaii.

[11] R. Bate, D. Mueller, and J. White, *Fundamentals of Astrodynamics*. New York: Dover Publications, 1971.

[12] W. H. Goodyear, "A general method for the computation of cartesian coordinates and partial derivatives of the two-body problem," technical report nasa cr-522, NASA, Sept. 1966.

[13] R. H. Battin, *An Introduction to the Mathematics and Methods of Astrodynamics*. AIAA Education Series, Reston, Virginia: American Institute of Aeronautics and Astronautics, revised edition ed., 1999.

[14] H. Sperling, "Computation of keplerian conic sections," *ARS Journal*, vol. 31, pp. 660–661, 1961.

[15] S. H. Herrick, "Universal variables," *Astronomical Journal*, vol. 70, pp. 309–315, 1965.

[16] W. W. Lemmon and J. E. Brooks, "A universal formulation for conic trajectories-basic variables and relationships," report 3400-601 9-tu000, TRW/Systems, Redondo Beach, CA, Feb. 1965.

[17] G. J. Der, "An elegant state transition matrix." Paper AIAA-1996-3660, AIAA/AAS Astrodynamics Conference, San Diego, CA, July 1996.

[18] E. T. Pitkin, "Second transition partial derivatives via universal variables," *Journal of Astronautical Sciences*, vol. 13, p. 204, Jan. 1966.

[19] H. J. Oberle and K. Taubert, "Existence and multiple solutions of the minimum-fuel orbit transfer problem," *Journal of Optimization Theory and Applications*, vol. 95, pp. 243–262, Nov. 1997.

[20] H. J. Oberle and W. Grimm, "Bndsco - a program for the numerical solution of optimal control problems," internal report no. 515-89/22, Institute for Flight Systems Dynamics, DLR, Oberpfaffenhofen, Germany, 1989.

[21] J. T. Olympio, *Optimisation and Optimal Control Methods for Planet Sequence Design of Low-Thrust Interplanetary Transfer Problems with Gravity Assists*. PhD thesis, Ecole des Mines de Paris, Oct. 2008.

[22] T. Dargent and V. Martinot, "An integrated tool for low thrust optimal control orbit transfers in interplanetary trajectories," in *Proceedings of the 18th International Symposium on Space Flight Dynamics*, (Munich, Germany), p. 143, German Space Operations Center of DLR and European Space Operations Centre of ESA, Oct. 2004.