A Continuous Feedback Optimal Control based on Second-Variations for Problems with Control Constraints

Joris T. Olympio

Abstract

The paper describes a continuous second-variation algorithm to solve optimal control problems where the control is defined on a closed set. A second order expansion of a Lagrangian provides linear updates of the control to construct a locally feedback optimal control of the problem. Since the process involves a backward and a forward stage, which require storing trajectories, a method has been devised to accurately store continuous solutions of ordinary differential equations. Thanks to the continuous approach, the method adapts implicitly the numerical time mesh. The novel method is demonstrated on bang-bang optimal control problems, showing the suitability of the method to identify automatically optimal switching points in the control.

1 Introduction

The current paper is concerned with the resolution of optimal control problems with bang-bang structure. Most optimal control problem solvers use a direct problem formulation\cite{5} that transforms the optimal control problem into a nonlinear program\cite{1} \cite{2} \cite{3} \cite{4}. Indeed, one could argue that direct methods offer the most straightforward formulation to a generic optimal control solver, and they are quite robust. Alternative methods, for instance indirect methods\cite{6}, require formulating a boundary value problem and convergence is usually difficult. Neither direct, nor indirect method-based algorithms can be both robust and accurate in computing the optimal control solution. Besides, direct and indirect approach, one should explore second-order methods.

Second-order methods appear as very good alternative. In the 1960’s, while trying to elegantly solve LQP researchers devised second-order methods such as second-variation methods or differential dynamic programming (DDP)\cite{7} \cite{8} \cite{9} \cite{10} \cite{11} \cite{12}. The classical second order algorithms for optimal control problems use dynamic programming equations. Referred as differential dynamic programming (DDP), DDP provides a simple way of computing an optimal control through a quadratic expansion of the Hamilton-Jacobi-Bellman value function around a nominal trajectory\cite{9}. Over the past 50 years, these methods have also been tested and adapted on non-LQP problems, and in the space community, DDP based methods seem to get a new lease of life. Current computer architecture allow better programing of the methods thanks to significant increase in memory capacity and processing unit speed. So far, Whiffen \cite{13}\cite{14} certainly proposed the best improvement of DDP method to provide a generic tool for high-fidelity low-thrust space trajectory optimisation. The major difficulty in low-thrust optimal control problems is the control is often of a bang-bang structure because the control is linear in the dynamics. Some work using DDP then consider simplifications in the dynamics\cite{15}, or consider a quadratic cost to the designing of low-thrust trajectories to asteroids to improve convergence rate\cite{16}. To the author knowledge there is not a lot of supporting theories for convergence and behaviour of DDP.

Second-order methods can also be devised using Lagrangian expansions, rather than the Bellman’s value function. The approach has been introduced by Bullock\cite{17}\cite{18} and Bryson and extended by others for space trajectory optimization\cite{19}\cite{20}. Although the basis is different from the derivation of DDP the end results are quite similar. Both DDP and gradient-based Lagrangian methods compute a linear control update iteratively using gradient of the Hamiltonian function. However, the derivations based on the expansion of the Lagrangian rather than the Hamilton-Jacobi-Bellman value function provide a better theoretical framework.

The algorithm presented in this paper uses as baseline an expansion of the Lagrangian. The novelty of the approach is however many fold. This paper proposes an approach for approximating the continuous solution through
interpolation, thus improving the backward-forward integrations, and increasing the accuracy of the sensitivity matrices and the feedback coefficients. A method for handling dynamical control constraints without using penalization is presented. Penalization of the control or the state for constraints is one of the major drawbacks of current DDP and second-variation methods. That is, a regularisation of the Lagrangian is also presented to improve the convergence rate.

In section 2, the main problem is introduced with notations and assumptions. The methodology is developed in section 3. This includes the second-order developments, the generation of an optimal feedback control law, the conditions of optimality, and the treatment of state constraints. Section 4 presents the optimal control update, and deals with the particular case of control constraints, which is important for the bang-bang problem class. Section 5 introduced a practical solution method to produce continuous solution for the problem, with an adaptive implicit mesh. The algorithm of the method is presented in section 6. Eventually, the last section, section 7 illustrates the method, and demonstrates that the method can find the bang-bang optimal control without requiring any insights on the switching structure.

2 Problem Statement

2.1 Preliminaries

To simplify notation, the following conventions are used. Scalars are denoted in italics, while column vectors, matrices and tensors are in bold. In addition, column vectors use lowercase and matrices use uppercase.

And for derivation,
\[ F_x = \frac{\partial F}{\partial x}, \quad F_{xy} = \frac{\partial^2 F}{\partial x \partial y} \]
where \( F \) is a general scalar function. For clarity of the expressions, when the points of evaluation are omitted it is assumed that the expression is evaluated around a nominal trajectory \([\hat{x}(t), \hat{u}(t), \hat{p}]\). The dimension of the state \( x(t) \) is \( n_x \), and the dimension of control \( u(t) \) is \( n_u \).

2.2 Formulation

The developments include the case where a scalar constant parameter \( p \) is included and needs to be optimised along with the control.

The dynamics are described by an ordinary differential equation
\[
\frac{dx}{dt} = f(x, u, p; t), \quad t \in [t_0, t_f], \quad u(t) \in U, \quad p \in P \quad (1)
\]
\[ 0 = \varphi(x(t_0), x_0, p_0) \quad (2) \]
where \( \varphi : \mathbb{R}^{n_x} \times \mathbb{R} \rightarrow \mathbb{R}^{n_p} \) describes initial conditions on the state at date \( t_0 \) and unknown parameter \( p_0 \). Functional \( f \), of state function \( x \in C([t_0, t_f], \mathbb{R}^{n_x}) \) and unknown constant parameter \( p \in P \subset \mathbb{R} \), is assumed at least twice continuously differentiable, \( f \in W^2_2(\mathbb{R}^{n_x} \times U \times P) \), with the Sobolev space
\[ W^2_p(\Omega) = \{ f \in L^2(\Omega) : \forall \alpha \in \mathbb{N}, |\alpha| \leq k, \partial^\alpha f \in L^p(\Omega) \} \]

Furthermore, the control is assumed a bounded measurable function, \( u \in L^2([t_0, t_f], U) \). Let the control region \( U \subset \mathbb{R}^{n_u} \) be defined by the box constraints:
\[ u^l_i \leq u_i \leq u^u_i, \quad 1 \leq i \leq n_u \quad (3) \]
where \( u^l_i \in \mathbb{R} \) and \( u^u_i \in \mathbb{R} \) are respectively the lower and upper bounds on the \( i^th \) component of the control.

Let terminal constraints be defined as
\[ \psi(x(t_f), p_f) = 0 \quad (4) \]
where $\psi : \mathbb{R}^{n_x} \times \mathbb{R} \rightarrow \mathbb{R}^{n_y}$ is assumed twice continuously differentiable. Scalar $p_f$ is an unknown parameter. With very few modifications, these problem parameters can also be slack variables of possible terminal inequality constraints.

Consider the minimisation of an objective function $J$ written under the Mayer form,

$$J = J(x(t_f), p_f) \rightarrow \min$$

where $J : \mathbb{R}^{n_x} \times \mathbb{R} \rightarrow \mathbb{R}$ is a functional of the vector $x(t_f)$ and the parameter $p$. It is assumed well defined (Jacobian of maximum rank), and at least twice continuously differentiable.

A second order method is presented to minimise the objective function $J$, Eq. (5), subjects to the state constraints Eqs. (2) and (4), and the control constraint Eq. (3), with iterative updates on the control $u(t)$ and the parameters $p, p_0$ and $p_f$.

### 3 Second Order Equations

#### 3.1 Second Order Equations

For this problem, an extended terminal value function is introduced,

$$\phi(x, v, p_f, t_f) = J(x(t_f), p_f) + v^T \psi(x(t_f), p_f) + \psi(x(t_f), p_f)^T C \psi(x(t_f), p_f)$$

where $v \in \mathbb{R}^{n_y}$ is a non-zero constant Lagrange multiplier vector for the constraints $\psi$, and $C > 0$ is a square diagonal regularisation matrix, $C \in M^{n_y \times n_y}(\mathbb{R})$. Let the augmented Lagrangian $\mathcal{L} \in W^2_2(\mathbb{R}^{n_x} \times U \times P \times \mathbb{R}^{n_y})$ of the constrained optimal control problem be defined with

$$\mathcal{L}(x, u, p, p_0, p_f, v) = \phi(x, v, p_f, t_f) + \eta^T \varphi(x_0, p_0, t_0) + \int_{t_0}^{t_f} \lambda^T \left( f(x, u, p; t) - \frac{dx}{dt} \right) dt$$

It accounts for initial conditions Eq. (2), terminal constraints Eq. (4) and objective function Eq. (5). Costate vector $\lambda$ and Lagrange multiplier $\psi$ have been introduced for the dynamics and the initial constraints Eq. (2) respectively.

While most second-order methods published to date only use penalisation to reach constraint satisfaction, the presented method introduces Lagrange multipliers to have a quicker and better satisfaction of the constraints. As part of a min-max problem, $V$ should be minimised for $u$ and $p$, while maximised for $v$. Indeed, the constrained maximisation supposes the existence of a min-max and max-min problem, or also the existence of a saddle point. The regularisation matrix $C$ of the augmented Lagrangian (Eq. (7)) is used to regularise the Lagrangian and find a saddle point of $\lambda$, where $\psi_\theta = \theta$.

Let the Hamiltonian $H \in W^2_2(\mathbb{R}^{n_x} \times \mathbb{R}^{n_x} \times U \times \mathbb{R})$ be defined as:

$$H(x, \lambda, u, p; t) = \lambda(t)^T f(x, u, p; t)$$

since the objective function $J$ (Eq. 5) does not include Lagrangian terms (integral terms).

Looking for control updates $\delta u$ of the unknowns of the problem are $\delta x$, $dp$ and $dV$, such that a control update can be expressed in the form

$$\delta u = \alpha + \beta \delta x + \gamma dp + \omega dV$$

with $\alpha \in \mathbb{R}^{n_u}$, $\beta \in M^{n_u \times n_x}(\mathbb{R})$, $\gamma \in \mathbb{R}^{n_u}$ and $\omega \in M^{n_u \times n_y}(\mathbb{R})$. $\bar{u}$ is the current nominal control and $\epsilon_u \in [0, 1]$ is an update factor. This control update can be seen as a feedback control law on the state $x$.

Since $J$, $f$, $\varphi$ and $\psi$ are assumed at least twice continuously differentiable, the development of the Lagrangian $\mathcal{L} \in W^2_2(\mathbb{R}^{n_x} \times U \times \mathbb{R})$ to second order, around the nominal trajectory $\bar{x}$ and control $\bar{u}$ and nominal parameter values, yields,
\[ d\mathcal{L}(\bar{x}, \bar{u}, p, p_0, p_f, \nu) = d\phi + d\left((\eta^T \phi)\right) \\
- \lambda_u^T \delta x_f + \lambda_0^T \delta x_0 \\
- \delta \lambda_f^T \delta x_f + \delta \lambda_0^T \delta x_0 \\
+ \int_{t_0}^{t_f} \left( \frac{d\lambda + \delta \lambda}{dt} \right)^T \delta x - \delta \lambda \frac{dx}{dt} \right) dt + \int_{t_0}^{t_f} dH dt \\
+ o(||\delta x||^2, ||\delta \lambda||^2, dp^2) \] (11)

with:

\[ dH = H_x \delta x + H_u \delta u + H_p dp + f \delta \lambda \]

\[ + \frac{1}{2} \delta x^T H_{xx} \delta x + \frac{1}{2} \delta u^T H_{uu} \delta u + \frac{1}{2} H_{pp} dp^2 \]

\[ + \delta u^T H_{ux} \delta x + \delta x^T H_{ux} \delta u + \delta x^T H_{xp} dp + H_{pu} \delta x dp \]

\[ + \delta \lambda^T f_u \delta x + \delta \lambda^T f_u \delta u + \delta \lambda^T f_p dp \]

\[ + \delta \lambda^T f_x \delta \lambda + \delta u^T f_u \delta \lambda + f_p \delta \lambda dp \] (12)

Residual term \( o(||\delta x||^2, ||\delta \lambda||^2, dp^2) \) describes the higher order terms of the development, and,

\[ o(||\delta x||^2, ||\delta \lambda||^2, dp^2) = \Delta \mathcal{L} = d\mathcal{L}(x, u, p, p_0, p_f, \nu) \]

\[ \Delta \mathcal{L} = \mathcal{L}(\bar{x}, \bar{u}, \bar{p}, \bar{p}_0, \bar{p}_f, \bar{\nu}) - \mathcal{L}(x, u, p, p_0, p_f, \nu) \] (13)

\[ \mathcal{L} \] is a measure of the improvement owing to the updated control law (Eq. (10)). The second term of the right hand side must thus be of the same order of magnitude of \( \Delta \mathcal{L} \). For linear or quadratic problems in \( x, u \) and \( p \), the residual term is zero.

**Lemma 1** (Riccati Transformation). The costate vector is related to the other variables with the transformation,

\[ \delta \lambda = R \delta x + K d\nu + T dp \] (15)

\[ with \ R \in \mathbb{R}^{n_x \times n_x}(\mathbb{R}), \ K \in \mathbb{R}^{n_x \times n}\(\mathbb{R}\) and \ T \in \mathbb{R}^{1 \times n_x}(\mathbb{R}).

**Proof.** The state transition matrix \( \Phi(t_f, t) \) for the dynamics \( f \) yields,

\[ x(t_f) = \Phi_{11}(t_f, t)x(t) + \Phi_{12}(t_f, t)\lambda(t) + \Phi_{13}(t_f, t)p \] (16)

\[ \lambda(t) = \Phi_{21}(t_f, t)x(t_f) + \Phi_{22}(t_f, t)\lambda(t_f) + \Phi_{23}(t_f, t)p \] (17)

\[ p = \Phi_{31}(t_f, t)x(t_f) + \Phi_{32}(t_f, t)\lambda(t_f) + \Phi_{33}(t_f, t)p \] (18)

The transversality conditions for the terminal constraints provide the value for \( \lambda(t_f) \). The transition matrix is invertible. Combining equations (16) and (17), and considering small variations of the variables \( x, p \) and \( \nu \) yield:

\[ \delta \lambda(t) = (1 - \Phi_{21}(t_f, t_f)\Phi_{12}(t_f, t_f))^{-1} \Phi_{21}(t_f, t_f)\Phi_{11}(t_f, t_f)\delta x(t) \\
+ (1 - \Phi_{21}(t_f, t_f)\Phi_{12}(t_f, t_f))^{-1} (\Phi_{21}(t_f, t_f)\Phi_{13}(t_f, t_f) + \Phi_{23}(t_f, t_f)) \delta p \\
+ (1 - \Phi_{21}(t_f, t_f)\Phi_{12}(t_f, t_f))^{-1} \Phi_{22}(t_f, t_f) \psi^T_{x_f} d\nu \]

**Assumption 1.** Consider a nominal trajectory \( \bar{x} \) and a nominal control \( \bar{u} \), possibly not optimal. The solutions of Eqs. (25, 26, 27) are bounded with the following assumptions (24, 25, 26):
Legendre-Clebsch condition $H_{uu} > 0$. This discards any problems where the control appears linearly. A procedure that guarantees this condition in the method shall be presented hereafter.

- $H_{xx} - H_{xu} H_{uu}^{-1} H_{ux}$ is positive-semidefinite.
- $R(t_f)$ is positive-semidefinite

The first assumption is a necessary condition to ensure that no singular arcs are encountered. However, this has no influence on the local optimality of the final solution.

**Definition 1.** Considering a quadratic model $(Q, R)$, the Trust-Region problem consists in minimizing the quadratic expansion for the solution to be within a given ball $B(x, r)$,

$$\min_p \left( f + p^T R + \frac{1}{2} p^T Q p \right)$$

s.t.

$$\|p\| \leq \Delta$$

where $p = \delta x$ and $\Delta$ is called the Trust-Region radius. Furthermore, the solution $p^*$ is solution of linear problem

$$(Q + \lambda I) p = -R$$

for some $\lambda \geq 0$ such that $Q + \lambda I$ is positive definite.

Applying the Trust-Region algorithm [27] [28] [29] results then in the computation of a shift matrix $S_{uu} = \lambda I$ such that the Hessian $H_{uu} \leftarrow H_{uu} + \lambda I$ is positive-definite. A counterpart of applying the Trust-Region method to correct the Hessian is a limitation of the control update amplitude $\delta u$ in accordance to the selection of Trust-Region radius $\Delta$, which eventually has an influence on result of Eq. (14).

Transformation of Eq. (15) allows to compute the feedback coefficients in Eq. (9). This yields, assuming $H_{uu} > 0$,

$$\alpha = -H_{uu}^{-1} H_u$$
$$\beta = -H_{uu}^{-1} (H_{xu} + R^T f_u)^T$$
$$\gamma = -H_{uu}^{-1} (H_{pu} + T^T f_u)^T$$
$$\omega = -H_{uu}^{-1} (K^T f_u)^T$$

And, using the conditions of optimality, ordinary differential equations for $R, K, T$ and the costate vector $\lambda$ are obtained. Thus,

$$\frac{d\lambda^T}{dt} = -H_x + H_u H_{uu}^{-1} (H_{ux} + H_{uu} \lambda R)$$
$$\frac{dx}{dt} = H_{\lambda}^T = f$$
$$\frac{dR}{dt} = H_{xx} + R^T f_x + f_x R + \beta^T (H_{ux} + f_x R)$$
$$\frac{dK}{dt} = K^T f_x + \omega^T (H_{ux} + f_x R)$$
$$\frac{dT}{dt} = H_{px} + T^T f_x + \gamma^T (H_{ux} + f_x R) + f_p R$$

These ODEs allow of computing a bounded update $\delta u$ of the current nominal control $\bar{u}$ through Eq. (12). If the truncation error defined by Eq. (14) is limited, this control update should minimize the Lagrangian of the problem and to some extent the terminal constraints violation and/or the terminal cost $J$.

Additionally, to decrease the constraint violation, an update of the Lagrange multiplier $\nu$ should be provided, and similarly, an update for the parameter $p$ should be produced to minimize the Lagrangian.
3.2 Parameter and Lagrangian Vector Updates

**Proposition 1.** The following transformations are considered

\[ d\psi = K^T \delta x + Qd\nu + Vd\nu \quad (28) \]
\[ d\tau = T^T \delta x + V^Td\nu + Wd\nu \quad (29) \]

This can be demonstrated by computing the ODEs and identifying the similarities with previous ODEs. This yields additional ODEs:

\[ -\frac{dV}{dt} = K^T (f_u \gamma + f_p) \quad (30) \]
\[ -\frac{dQ}{dt} = K^T f_u \omega \quad (31) \]
\[ -\frac{dW}{dt} = H_{pp} + \gamma H_{up} + T^T (f_u \gamma + f_p) + f_p T \quad (32) \]

Eventually, assuming \( W(t_0) < 0 \), the problem parameter update is,

\[ dp = -W(t_0)^{-1} \phi_{vp} \quad (33) \]

Also, assuming \( Q(t_0)^{-1} \) exists, lagrange multipliers are updated with the linear update

\[ d\nu = \varepsilon_\nu Q(t_0)^{-1} (\delta \phi - V(t_0)dp) \quad (34) \]

If \( Q(t_0) < 0 \), \( d\nu \) defines a direction for the constraints reduction, and \( \varepsilon_\nu \in [0, 1] \) is thus determined to minimise the constraints \( \psi \).

The ODE system (Eqs. (30-32)) is integrated backwards from some terminal conditions depending on the terminal constraints \( \psi \) and the terminal cost \( J \), while the state dynamics (Eq. (1)) are integrated forward. This results in numerically decoupled schemes, which yield to numerical robustness.

3.3 Terminal Conditions

The formulation allows the handling of various types of constraints without changing of the backward integrated ODE, since the treatment of constraints has an influence only on the terminal conditions.

The following terminal conditions are used to integrate backwards the ODEs (Eqs. (25), (26), (27), (30), (31), (32), (35), (36), (37), (38), (39), (40), (41)):

\[ R(t_f) = J_{xx}(t_f) + \phi_{xx}(t_f) \quad (35) \]
\[ K(t_f) = \phi_{x\nu}(t_f) \quad (36) \]
\[ Q(t_f) = 0 \quad (37) \]
\[ T(t_f) = \phi_{x\nu}(t_f) \quad (38) \]
\[ V(t_f) = \phi_{vp}(t_f) \quad (39) \]
\[ W(t_f) = \phi_{pp}(t_f) \quad (40) \]
\[ \lambda(t_f) = \phi_x(t_f) \quad (41) \]

The satisfaction of the constraints \( \psi \) depends on the Lagrange multiplier \( \nu \).

**Lemma 2.** Assuming \( H_{uu}^{-1} \) is positive definite, the Jacobian \( \psi_x(x(t_f)) \) of the constraints with respect to the terminal state \( x_f \) has full rank \( n_x \), and \( Q(t_0) \) must be negative definite, then the Lagrange multipliers update \( d\nu \) gives the (local) direction of minimisation of the constraints \( \psi \).

**Proof.** See [19].
3.4 Parameters Determination

The unknown parameters for the initial conditions and the dynamics do not follow the same update rule as of \( dp \). Because the equations of the sensitivity matrices, \((R, K, \text{ and } Q)\) and \((T, V, \text{ and } W)\), are integrated backwards, setting the initial constraint \( \varphi \) does not change the terminal conditions of integration Eqs. (35, 36, 37, 38, 39, and 40). Assuming the initial constraint \( \varphi \) is well defined (maximum rank condition), the update formula for the parameters \( p \) is

\[
d p_0 = -\varphi^{-1} p \varphi_p \quad (42)
\]

There is no need to update the Lagrange vector \( \eta \) introduced in Eq. (11). One may want to include a relaxation factor to control the update \( \delta p_0 \). Likewise, for the parameter part of the terminal constraint, \( \varphi \)

\[
d p_f = -\varphi^{-1} p \varphi_p \quad (43)
\]

4 Treatment of Control Constraints

4.1 Acceptance of Control Update

Once the backward equations are integrated from their respective terminal conditions, and around a nominal trajectory, the control update is computed using Eqs. (9) and (10). Note that according to Eq. (34), \( \delta u \) depends on \( \epsilon_n \).

Both \( \epsilon_n \) and \( \epsilon_u \) permit of adjusting the control update to yield an improvement of the extended value function, and to ensure that the second order developments truncation error (Eq. (14)) is small enough. If Lagrange multipliers \( \nu \) are not used, a linesearch procedure is used to find \( \epsilon_u \) that minimizes the merit function. When \( \epsilon_n \) is used, the direction of minimization is given by a linear combination of \( \alpha \) and \( d \nu \). One approach is to perform linesearches with \( \epsilon_u \) and \( \epsilon_n \) successively, the purpose being to find quickly an improvement of the merit function, regardless whether it is or not an strict minimizer.

Indeed, as in [9], the validity of the second order development can be checked using

\[
\Delta L = \int_{t_0}^{t_f} \left[ H(\bar{x}, \bar{\lambda}, \bar{u}, p, t) - H(\bar{x}, \bar{\lambda}, \bar{u}, p, t) \right] dt
\]

\[
\Delta L \approx (\Delta \mathcal{L})_{d \nu = 0} \quad (44)
\]

Clearly,

\[
\lim_{\epsilon_n \to 0} \Delta L = \lim_{\epsilon_u \to 0} (\Delta \mathcal{L})_{d \nu = 0} = 0 \quad (45)
\]

For \( \epsilon_u \) small enough, \( \Delta L \) gets close to zero thus ensuring the second order developments are assumed valid. If in addition, the value \( \Delta L \) of Eq. (44) is negative then the control update \( d u \) can be accepted.

Under these conditions the method generates iterates of control \( u'(t) \) that minimize the Lagrangian \( \mathcal{L} \).

4.2 Bang-Bang Control and Control Constraints

According to Pontryaguin Maximum Principle[6], when the control belongs to a compact set \( U \), for the problem class considered where the control appears linearly in the Hamiltonian, the optimal control has a bang-bang structure. Obviously, this result does not come from the variational equations (that seek \( \nabla_u H = 0 \)), but directly from the maximisation of the Hamiltonian over the compact set. In the problem formulation, and Eq. (3), closed set on the control is assumed. However, the development of the method, referred to calculus of variations, assume the Lagrangian is differentiated on an open set.

Constraints on the control can be set such that the control is implicitly limited. One approach is to formulate the control constraint (Eq. 3) using a barrier function \( \rho \) that is incorporated into the cost [13][15]. Thus, when the control violates bounds at any time, the solution gets penalised and eventually the solver corrects the violation. This approach seems to provide practical results, and also as the control bounds may not be respected in the initial iterations of the
solver, the solver can go through a forbidden space of the search space and possibly find solutions with that extra controllability. This approach, however, does not ensure the optimal control solution is an admissible control (e.g. \(u \notin U\)).

The approach considered here is to explicitly force the control to not exceed the desired limits. The principle is to identify a continuous function \(m : \mathbb{R} \to U\) and to introduce a slack variable \(v_i\) such that the box constraint Eq. \([3]\) can be replaced with the explicit equality constraint

\[
u_i - m(v_i) = 0 \quad (47)
\]

Indeed, one can thus replace \(u_i\) accordingly in the developments, and solve the problem for \(v_i\) directly instead of \(u_i\). Since the compact set \(U\) is then not used, the gradient \(\nabla v_H\) can be used for the computation of the control updates, and this also provides an indication of the optimality of the control \(u\) (Weierstrass condition). Such an application of \(m\) can be for instance,

\[
m(v_i) = u_l + (u_i - u_l) \sin^2 v_i
\]

This approach, although numerically more challenging than penalization, makes sure that the control satisfies the problem description and never violates the box constraints Eq. \([3]\).

### 4.3 Heuristic for Convergence of Bang-Bang Problems

Some transformations \(m : \mathbb{R} \to U\) may allow, however, triviality or singularity to the problem. For instance, when Lagrange multipliers \(\nu\) are not used, for the \(i\)-th component of the control \(u\), some transformations \(m\) yield,

\[
\exists t \in [t_0, t_f], m(v_i(t)) = 0 \Rightarrow (H_\nu(t) = 0, \delta v_i(t) = 0) \quad (48)
\]

No updates are made even though the control solution is not optimal.

A diagonal matrix \(D > 0\) is thus introduced in the Hamiltonian,

\[
H(x, \lambda, u, p; t) = \lambda^T f(x, u, p; t) + u^T Du \quad (49)
\]

This is equivalent to adding a quadratic Lagrange term in the cost \(J = J(x(t_f), u, p_f)\). As a result, control \(u\) is smoothed.

To obtain a bang-bang saturated control, \(D\) is updated and decreased periodically with the iterations for the control to be in a neighbourhood of an optimal bang-bang control solution.

### 5 Continuous Solution Approximation

#### 5.1 Problematic

To compute the feedback control coefficients, \(\alpha, \beta, \gamma\) and \(\omega\), a backwards integration (e.g. Eqs. \([25], [26], [27]\)) is done around a nominal trajectory \(\{x, p\}\) that results from a forward integration. It is thus not possible to solve all the ODEs concurrently. It is necessary to store continuous solution of the ODEs, that is both the trajectory (e.g. \(x\) and \(\lambda\)), and the sensitivity matrices (e.g. \(R, K\)) to then compute the control update. To respect the continuous developments of previous section, an efficient and reliable continuous data storage mechanism has to be devised.

#### 5.2 Current Discrete Approximation Approaches

Often, the alternative is to work with the best discrete time approximate solutions and the definition of a time mesh where the solution can be evaluated.

Many approaches have been proposed in the literature to produce a discrete control time mesh. The approach of keeping the control constant for fixed durations \([15], [13]\) produces good results when the number of nodes has been well selected. Likewise, the control can be kept constant over multiple Runge-Kutta time steps for smooth control \([16]\). But, since the control is used in the forward integration but not during the backwards integration, using the Runge-Kutta (RK) time steps alone does not seem appropriate for the bang-bang problem considered. Runge-Kutta integration is only time-reversal when used with a constant step size \([30]\). Others \([31], [32]\) propose a refinement procedure based
on an integer programming problem, and consider the minimisation of the continuity error and the minimisation of number of the points to add.

When the control is of a bang-bang structure, an other approach should be followed, because a constant time mesh for the control is likely to miss the switching times of the optimal control. The usual approaches increase the number of points close to the discontinuity. This has a major drawback of increasing the number of data points to store.

The choice of a discrete, or multi-stage, formulation could be argued in regards to potential applications, since it is always simpler to implement a piecewise control. From a theoretical point of view, however, it is necessary to propose a scheme for limiting the approximation made on the control. Properties of the continuous control solution (e.g. switching structure, singular arcs) can eventually be used to produce a good discrete control solution.

5.3 Continuous Trajectory Approximation

The trajectory and the control are initially sampled at $N$ given node points that can be described by a grid $t_i \in [t_0, t_f]$, $i = 1 \ldots N$. The trajectory and the control are thus exactly defined at the nodes, and an approximation to the continuous solution is constructed between the node points. Consider an explicit adaptive step-size Runge-Kutta integration method. It is the baseline for solving the ODE systems in the current paper. An explicit Runge-Kutta integration method uses the following iterative equations for integrating,

$$y_{n+1} = y_n + h \sum_{i=1}^{s} b_i k_i$$  \hspace{1cm} (50)

$$k_i = f \left( t_n + c_i h, y_n + h \sum_{j=1}^{s} a_{ij} k_j \right),$$  \hspace{1cm} (51)

where $a_{ij}$, $b_i$ and $c_i$ are constant scalars, and $s$ is the order of the method. Shampine’s method for dense output computation in Runge-Kutta computation is used\cite{Shampine1988}. One can introduce a polynomial $S(x)$, and an error function $e(x)$ such that,

$$y(x) = S(x) + e(x)$$  \hspace{1cm} (52)

$$\frac{de}{dx} = f(x, S(x)) - \frac{S(x)}{dx}, \quad e(x_0) = 0$$

Shampine\cite{Shampine1988} thus defines polynomials $P(x)$ of maximum degree $m + 1$ ($m$ being the number of sub-integrations in one subinterval of length $h$),

$$\frac{dP(x)}{dx} = \sum_{i=0}^{m} l_i m(x) f(x_i, e_i + S(x_i))$$  \hspace{1cm} (53)

where $l_i m$ are Lagrange basis polynomials of $k$th-order $m$. It is demonstrated there exists an integer $n$, and constants $c_1$ and $c_{2,m}$ such that

$$\left\| \frac{d^2 y(x)}{dx^2} - \frac{dS(x)}{dx} \right\| \leq c_1 h^n$$  \hspace{1cm} (54)

$$\| y(x) - P(x) \| \leq c_2 h^{2 + \min(m,n+1)}$$  \hspace{1cm} (55)

That is, $P$ is a better higher order approximation of $y$ that $S$ is. An iterative scheme is used to construct $P(x)$. The procedure starts with a simple Euler integration model for $S(x)$ with $m = 1$, and then increases the order $m$ by evaluating successively $S(x)$ and $P(x)$. The construction of $S$ and $P$ is dependant of the extrapolation method (e.g. Runge-Kutta method), and $P$ is only then available at the end of the integration over the interval of length $h$.

For a fifth order explicit RK method, $\forall x \in [x_n, x_{n+1}]$,

$$y(\bar{x}) = r_1 + \bar{x}(r_2 + (1 - \bar{x})(r_3 + \bar{x}(r_4 + (1 - \bar{x})r_5)))),$$  \hspace{1cm} (56)
where

\[ \mathbf{r}_1 = \mathbf{y}_n \]  
\[ \mathbf{r}_2 = h \sum_{i=1}^{k} b_i \mathbf{k}_i \]  
\[ \mathbf{r}_3 = h \mathbf{k}_1 - (\mathbf{y}_{n+1} - \mathbf{y}_n) \]  
\[ \mathbf{r}_4 = -h (\mathbf{k}_2 + \mathbf{k}_1) \]  
\[ \mathbf{r}_5 = h (d_1 \mathbf{k}_1 + d_3 \mathbf{k}_3 + d_4 \mathbf{k}_4 + d_5 \mathbf{k}_5 + d_6 \mathbf{k}_6 + d_7 \mathbf{k}_7) \]

and \( d_i \) are constants of the method. Globally, the solution for the state trajectory and the sensitivity matrices can be stored as a \( C^1 \) approximation. For a fifth order RK method, five coefficients for each element of the ODE system, plus the range of validity of the polynomials are stored.

Consequently, during the forward integration, the trajectory state \( \mathbf{x}(t) \) is stored continuously, using Eq. (56), and this continuous approximation is used during the backward integration. Likewise, during the backward integration, the costate \( \lambda(t) \) and the sensitivity matrices \( \mathbf{R}(t), \mathbf{K}(t), \mathbf{T}(t), \mathbf{V}(t) \) continuous approximation are stored. Thus, all matrices and vectors involved in the computations follow a high resolution continuous approximation. Eventually, the backward stage being fully continuous, a continuous control update \( \delta \mathbf{u} \) is available.

Additionally, to obtain the best trajectory and the most accurate optimal control, from the continuous control update \( \delta \mathbf{u} \), an approach should be devised to construct a continuous approximation of the nominal control \( \bar{\mathbf{u}} \).

### 5.4 Continuous Nominal Control Approximation

The continuous control approximation is constructed a-posteriori, from the results of the backward integration. In regards to Eq. (10), the continuous control update should be added to the nominal control \( \bar{\mathbf{u}} \) to provide an updated control \( \mathbf{u} \) that can be used as the new nominal control for the following iteration during the backward integration. Consider a nominal control \( \bar{\mathbf{u}} \), during the forward recursion, when integrating ODE Eq. (11). According to section 5.3, control update \( \delta \mathbf{u} \) can be constructed accurately using RK interpolant of matrices \( \mathbf{R}, \mathbf{K} \) for all \( t \in [t_0, t_f] \), and likewise, an accurate updated control \( \mathbf{u} \) can be computed. This online produced nominal control has however to be stored, such that it can be used again for the backward recursion and future iterations. Practically, this is a difficult problem since neither \( \bar{\mathbf{u}} \) nor \( \delta \mathbf{u} \) have a readily available closed form, in general.

Consider \( t_k \) the RK nodes of the forward integration with the control \( \bar{\mathbf{u}} + \epsilon_\mathbf{u} \delta \mathbf{u} \) with values \( \mathbf{u}(t_k) \). The problem is to completely determined \( \mathbf{u} \) such that

\[ |J(\bar{\mathbf{u}} + \epsilon_\mathbf{u} \delta \mathbf{u}) - J(\mathbf{u})| \leq \eta_1 \]  
\[ ||\phi(\bar{\mathbf{u}} + \epsilon_\mathbf{u} \delta \mathbf{u}) - \phi(\mathbf{u})|| \leq \eta_2 \]

where \( J \) and \( \phi \) are, respectively, the target merit function value and the constraints, computed with the improved control \( \bar{\mathbf{u}} + \delta \mathbf{u} \). \( \eta_1 \) and \( \eta_2 \) are given small tolerances. For very sensitive problems, the discrepancy between this new control \( \mathbf{u} \) and the control \( \bar{\mathbf{u}} + \epsilon_\mathbf{u} \delta \mathbf{u} \) can result in large differences in the target merit function value. This difference can possibly be bigger than the expected improvement (Eq. (44)) thus resulting in a failure of the method to converge properly.

Each component of \( \mathbf{u} \) is thus approximated between the coarse grid nodes to follow a piecewise-polynomial function, or spline (linear function, cubic spline interpolant). Intermediate nodes at \( t_{k+1/2} \) can be tested, and added if it results in a better approximation of \( \bar{\mathbf{u}} + \epsilon_\mathbf{u} \delta \mathbf{u} \). Eventually, \( \mathbf{u} \to \bar{\mathbf{u}} \).

### 6 Second Order Algorithm and Implementation

#### 6.1 The Algorithm

The algorithm is implemented (C-SOGO) and coded in C++. It uses efficient linear algebra classes with shared-memory parallelisation. A high level description of the algorithm is the following:

Given:
• convergence tolerance on the constraints $\eta_{\psi}$, and the optimality condition $\eta_H$.

• Trust-Region radius for the control $\Delta u > 0$.

• Trust-Region radius for Lagrange multipliers $\Delta \nu > 0$, and parameter $\Delta \rho > 0$.

• a nominal control $\bar{u}(t)$ for $t \in [t_0, t_f]$.

• a nominal Lagrange multiplier $\bar{\nu} \in \mathbb{R}^{n\psi}$.

• a regularisation matrix $C \in \mathbb{R}^{n\psi \times n\psi}$.

• initial state $x_0 = x(t_0)$.

Step 0 Initialisation

Step 0.a. Initialise iteration counter.

Step 0.b. Set $\varepsilon_u^* = 0$ and $\varepsilon_{\nu}^* = 0$.

Step 1 Nominal trajectory

Step 1.a. Using a nominal control $\bar{u}(t)$, compute the state trajectory $x(t)$.

Step 1.b. Compute nominal merit function value $\bar{L}$.

Step 2. Check terminal criteria

Compute the terminal constraints $\psi$ Eq. (4). If constraints $\|\psi\| > \eta_{\psi}$ go to Step 3.

If optimality condition $\max(\Delta L, \|H_u\|_{\infty}) > \eta_H$ go to Step 3.

Otherwise, an optimal solution has been found. Stop.

Step 3 Backwards Step

Step 3.a. Computation of the terminal conditions with Eqs. (36), (37), (35) and Eqs. (38), (40), (36).

Step 3.b. Computation of the Shift matrix $S_{uu}$ thanks to the Trust-Region algorithm (see definition 1) for $H_{uu}(t) + S_{uu}(t) > 0$ for all $t \in [t_0, t_f]$. The Shift matrix $S_{uu}$ shall ensure the boundedness of the ODE solutions.

Step 3.c. Integration of Eqs. (25), (26), and (31) and computation of continuous time approximation of $\lambda(t), R(t), K(t)$ and $Q(t)$ for all $t$. During the integration, $H_{uu}$ is shifted with $S_{uu}$. As the equations are integrated, in accordance to sec. 5.3 a mesh and polynomials are constructed to provide a continuous approximation of them.

Step 3.d. Likewise, if the problem includes a parameter, integrate Eqs. (27), (30), and (32) and compute continuous time approximations of $T(t), V(t)$ and $W(t)$ for all $t$.

Step 4 Computation of Lagrange multipliers $\bar{\nu}$

Step 4.a. If $Q(t_0) < 0$ go to 4.c

Step 4.b. Apply Trust-Region algorithm to have a negative definite $Q(t_0)$.

Step 4.c. Computation of Lagrange multipliers $\bar{\nu}$ for the constraints using Eq. (34).

Step 5. Forward Integration Loop
**Step 5.a.** Integrate the dynamical state equations Eq. (1) with the control \( u + \varepsilon_u \delta u \), where the control update \( \delta u \) is computed using Eq. (10). The continuous approximation of sensitivity matrices \( R(t) \), \( K(t) \), and \( Q(t) \) accurately provide the feedback coefficients \( \alpha \), \( \beta \) and \( \gamma \). To produce the improving control update \( \delta u \), algorithm A1 is used to find \( \varepsilon_u \) and \( \varepsilon_\nu \) that minimise the merit function Eq. (6).

**Step 5.b.** Evaluation of the constraints \( \psi(x; t_f) \), the objective function \( J(x; t_f) \), and the merit function value \( L(x, \nu; t_f) \).

**Step 5.c.** If improvement in merit function, \( L < \bar{L} \), take \( \varepsilon_u^* = \varepsilon_u \) and \( \varepsilon_\nu^* = \varepsilon_\nu \). Go to step 6.

**Step 5.d.** Improvement could not be found. Conditions of validity of the second order development might not be meet. Reduce Trust-Region radius \( \Delta_u \). Increase Iteration counter. Go to step 3.

**Step 6.** Accept current control.

**Step 6.a.** Constructing a continuous time approximation of the new nominal control \( \bar{u} \leftarrow \bar{u} + \varepsilon_u^* \delta u \).

**Step 6.b.** Update Lagrange multipliers, \( \nu = \bar{\nu} + \varepsilon_\nu^* d \nu \).

**Step 6.c.** Update nominal merit function value \( \bar{L} \).

**Step 6.d.** Update regularisation matrix \( C \).

**Step 7.** Continue with the next iteration. Increase iteration counter. Go to step 2.

Algorithm A1 is either a line-search method to get \( \varepsilon_u \in [0, 1] \) that minimises \( \mathcal{L} \) when \( \nu \) is not used, or otherwise, a simple two-dimensional \( [\varepsilon_u, \varepsilon_\nu] \)-grid search method to get the additional parameter \( \varepsilon_\nu \in [0, 1] \). On the current iteration, linesearch, or gridsearch, is stopped when a sufficient decrease is obtained.

Convergence is achieved when both the constraints \( \psi \) and the optimality norm are satisfied to a given tolerance. A norm of optimality can be given by Eq. (44) and \( H_u \).

7 **Examples**

The initial motivation of this work has been the optimization of low-thrust space trajectories, which present a bang-bang optimal control structure. Both presented examples have affine control dynamics with the control defined on a closed set, and thus the optimal control should be bang-bang.

**7.1 Double Integrator Problem**

The dynamics

\[
\mathbf{f}(\mathbf{x}, \mathbf{u}; t) = \frac{d}{dt} \begin{bmatrix} p \\ v \\ u \end{bmatrix} = \begin{bmatrix} v \\ u \end{bmatrix}
\]

where \( p \) is the 1-D position of the particle, and \( v \) is its velocity. The control bounds are

\[
-1 \leq u \leq 1
\]

Owing to the control constraints, the transformation \( u = 1 - 2\sin^2 v \) is used, and \( v \) is the new control to seek.

Terminal constraints are

\[
\phi(x; t_f) = \begin{bmatrix} p(t_f) = 0 \\ v(t_f) = 0 \end{bmatrix}
\]

And the initial conditions are \( p(0) = 1, v(0) = 1 \).

The objective is to minimize the total time,

\[
J(x; t_f) = t_f \rightarrow \max
\]
The time of flight $t_f$ is thus considered as a parameter of the problem. The integration of the equations is thus done through a change of variable $\tau = t/t_f$ such that the integration is done on $[0, 1]$ with varying Jacobian $d\tau$. That is we should use

$$\tilde{f}(x, u, t_f; \tau) = f(x, u; \tau t_f)$$

(68)

Figure 1: Iterates of a simple control problem. The coarse grid includes 3 node points. Left, continuous control iterates. Right, gradient of the Hamiltonian with respect to the continuous control.

Figure 1 shows the iterates of program to find the solution. The control is approximated with a constrained cubic spline to prevent overshoot at intermediate points (with constrained cubic splines, the slope is imposed rather than the curvature). At the plateau of the control, in figure 1 the control is thus approximated with straight lines.

Using as initial guess $t_f = 3.5$ we found an optimal time $t_f^* = 3.449581$.

One can note how the number of points varies. The comparison to the smoothness of the control or the optimality of the solution is not straightforward, however.

It is arguable that the control obtained with the method is not strictly bang-bang in the sense that it can reach intermediate values on a set of measure not null. However, from a numerical point of view, for an optimality tolerance small enough the error to the true bang-bang solution should be negligible. And, possibly, the solution obtained can be refined in a later stage with the switching time as parameter since an optimal control structure is then known. The result of the refinement is likely to be negligible if the optima
7.2 Simple Orbital Transfer Problem

The problem is the one of transferring a spacecraft from one circular orbit to another with higher radius. The terminal orbit is sought in the same plane as the initial orbit, and thus the dynamics is simplified to be two-dimensional. The spacecraft is propelled with continuous and constant thrust. With the gravitational constant $\mu$, the dynamics are then,

$$\frac{d}{dt} \begin{bmatrix} r \\ v_r \\ v_\theta \\ m \end{bmatrix} = \begin{bmatrix} \frac{v_r^2}{r} - \frac{\mu}{r^2} + \frac{F}{r} \sin u \\ r v_r v_\theta^2 + \frac{F}{r} m \delta \cos u \\ -\frac{F}{V_e} \end{bmatrix}$$

State elements $r, v_r, v_\theta$ and $m$ denote respectively the radial position, the radial velocity, the ortho-radial velocity and the mass of the spacecraft. $F$ is the thrust force ($0 \leq F \leq F_0 = 5N$) and $V_e$ is the exhaust velocity ($V_e = g_0 I_{sp} = 19612m/s$). The controls are thus the steering angle $u$ and the thrust amplitude $F$.

The initial orbit is circular of radius $r_0 = 20000$ km. The constraint is to be on a circular orbit of radius $r_f = 42000$ km at final time. Using the cylindrical coordinate, this yields the constraints,

$$\phi(x; t_f) = \begin{bmatrix} v_r \\ v_\theta - \sqrt{\frac{\mu}{r}} \\ (r - r_f)^2 \end{bmatrix}$$

The time of flight is fixed to $t_f = 4$ days. The initial mass is 1000 kg. The objective is to maximize the final mass.

$$J(x; t_f) = m(t_f) \rightarrow \max$$

The final optimal mass is $m(t_f)^* = 932.15$ kg and the spacecraft make about 8 revolutions around Earth. The optimal control is depicted on figure 3 and the optimal trajectory is on figure 2. As can be seen on figure 3, the optimal control is of bang-bang type. The solver is able to find accurately the commutation points, while no insight of the control structure is initially provided. Such problem has usually many local optimal that can be sorted with respect to the number of revolutions. That is some solutions may have more, or less, than 8 revolutions for a constant time of flight. In general however, the best control is to thrust at the perigee (where the cost and the terminal constraints are more sensible to thrust) and coast at the apogee, as depicted in figure 2.
Figure 2: Optimal trajectory of the orbital transfer problem.

Figure 3: Control of the orbital transfer problem.
8 Conclusions

The solving of optimal control problems with a continuous backward-forward sweep algorithm based on a second-order variations is presented. A second order expansion of a Lagrangian provides linear updates of the control to construct a locally feedback optimal control of the problem. The control updates are computed following backward and forward stages. Ordinary differential equations are integrated backward around a state trajectory that has been computed forwardly in an earlier stage. The method uses an accurate continuous approximation of the ODE solutions to ensure precision of the control updates.

The method was demonstrated on two examples with bang-bang optimal control solution. It was shown that the solver can find accurately the switching structure of the solution, without any insight.

References

[1] I. Michael Ross and Fariba Fahroo. Legendre pseudospectral approximations of optimal control problems. 2003.

[2] A. V. Rao, D. A. Benson, C. L. Darby, M. A. Patterson, C. Francolin, , and G. T. Huntington. Algorithm 902: Gpops, a matlab software for solving multiple-phase optimal control problems using the gauss pseudospectral method,. ACM Transactions on Mathematical Software, 37(2):1–39, April 2010. doi 10.1145/1731022.1731032

[3] V.M. Becerra. Solving complex optimal control problems at no cost with psopt. In Proc. IEEE Multi-conference on Systems and Control, pages 1391–1396, September 2010.

[4] J.T. Betts and W. P. Huffman. Sparse optimal control software, socs. Technical report, Boeing Information and Support Services, Seattle, Washington, 1997.

[5] J.T. Betts. Survey of numerical methods for trajectory optimization. Journal Guidance, Control and Dynamics, 21(2):193 – 207, 1998. doi [10.2514/2.4231]

[6] L.S. Pontryagin. The Mathematical Theory of Optimal Processes. John Wiley and Sons Inc, 1962.

[7] S. R. Mc Reynolds and A. E. Jr. Bryson. A successive sweep method for solving optimal programming problems. Technical Report AD0459518, Harvard University, 1965.

[8] D.H. Jacobson and D.Q. Mayne. Differential Dynamic Programming. American Elsevier, New York, 1969.

[9] D.A. Mayne. A second order gradient method for determining optimal trajectories for nonlinear discrete-time systems. Int. J. Control, 3:85–95, 1966.

[10] Mitter S.K. Successive approximation methods for the solution of optimal control problems. Automatica, (3):135–149, 1966.

[11] S.B. Gershwin and D.H. Jacobson. A discrete-time differential dynamic programming with application to optimal orbit transfer. AIAA Journal, 8(9):1616–1626, 1970.

[12] P. Dyer and S. R McReynolds. Optimization of control systems with discontinuities and terminal constraints. IEEE Transactions on automatic Control, 14(3), 1969.

[13] G.J. Whiffen and J.A. Sims. Application of a novel optimal control algorithm to low-thrust trajectory optimization. In AAS/AIAA Space Flight Mechanics Meeting, Feb 2001. AAS 01-209.

[14] G.J. Whiffen and J.A. Sims. Application of the sdc optimal control algorithm to low-thrust escape and capture including fourth-body effects. In 2nd International Symposium on Low-Thrust Trajectories, Toulouse, June 2002. 18-20 June.

[15] G. Lantoine and R.P. Russel. A hybrid differential dynamic programming algorithm for low-thrust optimization. In AIAA/AAS Astrodynamics Specialist Conference and Exhibit, 18-21 August 2008, Honolulu, Hawaii, 2008.
[16] C. Colombo, M. Vasile, and G. Radice. Optimal low-thrust trajectories to asteroids through an algorithm based on differential dynamic programming. *Celestial Mechanics and Dynamical Astronomy*, 105(1):75 – 112, 2009. doi[10.1007/s10569-009-9224-3]

[17] T.E. Bullock. Computation of optimal controls by a method based on second variations. Technical report, Stanford University, 1966.

[18] T. Bullock and G. Franklin. A second-order feedback method for optimal control computations. *IEEE Transactions on Automatic Control*, 12(6):666 – 673, 1967.

[19] 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, October 2008.

[20] J.T. Olympio. Algorithm for low thrust optimal interplanetary transfers with escape and capture phases. In *AIAA/AAS Astrodynamics Specialist Conference and Exhibit*, Honolulu, Hawaii, August 2008. 2008-7363.

[21] P.E. Gill, W. Murray, M. A. Saunders, and M.H. Wright. Some theoretical properties of an augmented lagrangian merit function. Technical Report SOL 86-6R, Stanford University, Sep 1986.

[22] M. R. Hestenes. Multiplier and gradient methods. *Journal of Optimization Theory and Applications*, 4(5):303–320, 1969. doi[10.1007/BF00927673]

[23] M.J.D. Powell. *A method for nonlinear constraints in minimization problems*, pages 283–298. Academic Press, optimization edition, 1969.

[24] R.E. Kalman. Contributions to the theory of optimal control. *Boletin de la Sociedad Matematica Mexicana*, 5(102):102–119, 1960.

[25] R.S. Bucy. Global theory of the riccati equation. *Journal of Computer and System Sciences*, 1(4), 1967. doi[10.1016/S0022-0000(67)80025-4]

[26] D.H. Jacobson. New conditions for boundedness of the solution of a matrix riccati differential equation. *Journal of Differential Equations*, 8(2):258–263, 1970. doi[10.1016/0022-0396(70)90005-7]

[27] J.F. Rodriguez, J.E. Renaud, and L.T. Watson. Trust region augmented lagrangian methods for sequential response surface approximation and optimization. *Journal of Mechanical Design*, 15:58–66, 1997. doi[10.1115/1.2826677]

[28] A.R. Conn, N.I.M. Gould, and P.L. Toint. *Trust-region methods*. MPS-SIAM Series on Optimization 1. Society for Industrial and Applied Mathematics, 2000.

[29] T.F. Coleman and A. Liao. An efficient trust region method for unconstrained discrete-time optimal control problems. *Computational Optimization and Applications*, 4:47–66, 1993.

[30] D. Stoffer. Variable steps for reversible integration methods. *Computing*, 55(1):1–22, 1995. doi[10.1007/BF02238234]

[31] J. Laurent-Varin, J.F. Bonnans, N Berend, M. Haddou, and C. Talbot. Interior-point approach to trajectory optimization. *Journal of Guidance, Control and Dynamics*, 30(5), 2007. doi[10.2514/1.18196]

[32] J.T. Betts and W.P. Huffman. Mesh refinement in direct transcription methods for optimal control. *Optimal control applications and method*, 19(1):1 – 21, 1998. doi[10.2514/2.4231]

[33] Lawrence F. Shampine. Interpolation for variable order, runge-kutta methods. *"Computers and Mathematics with Applications"*, 14(4):255–260, 1987. doi[10.1016/0898-1221(87)90133-7]