Indirect approach for optimal control in Orekit

Romain Serra

February 2025

Change log

October 24 - 1st version (Orekit 12.2)

February 25 - 2nd version (Orekit 13.0)

1 Introduction

This document explains the foundations of Orekit (version 13.0) implementation of indirect optimal control for space trajectories. The library provides the user with a model for the so-called adjoint variables as defined in Pontryagin's Maximum Principle (PMP) for a range of cost functions, along with a solver for these optimality conditions. The latter is limited to fixed-time, orbit-to-orbit problems, in the form of a single shooting algorithm, and users are encouraged to develop their own tools for other applications. To know more about optimal control theory and their application in astrodynamics, the reader is referred to the literature, for example [6, 4, 3]. In a nutshell, optimal control is a branch of optimization set in infinite dimension (since the decision domain is a function space) and the indirect approach consists in deriving optimality conditions before solving for them (as opposed to the direct one, based on discretizing beforehand). The PMP provides first-order, necessary conditions of optimality for a class of problems.

The state vector $\mathbf{x}$ is made of the position $\mathbf{r}$ and velocity $\mathbf{v}$ vectors in some reference frame along with the mass m. Let $\mathbf{T}$ be the thrust force. Let us assume that the exhaust speed is constant, so that the mass equation is:

$$\dot{m}(t) = -\alpha|\mathbf{T}(t)|, \tag{1}$$

where $\alpha \geq 0$ and $|.|$ is the Euclidean norm. Given a set of external accelerations $\mathbf{a}_1, \ldots, \mathbf{a}_n$, according to Newton's second law, the equations of motion read:

$$\begin{aligned} \dot{\mathbf{r}}(t) &= \mathbf{v}(t) \\ \dot{\mathbf{v}}(t) &= \frac{\mathbf{T}(t)}{m(t)} + \sum_i \mathbf{a}_i(t, \mathbf{x}(t)) \end{aligned} \tag{2}$$

Let us adopt a Lagrange form of the cost function J:

$$J = \int_{t_0}^{t} L(\tau, \mathbf{x}(\tau), \mathbf{T}(\tau)) \mathrm{d}\tau \tag{3}$$

The minimization of $\int L$ under the differential equations (1-2) defines an optimal control problem. The addition of initial (at t_0) or final (at t_f) conditions is possible and these times can be fixed or not. Note that, except when stated otherwise, it shall be assumed that the thrust magnitude is bounded by $\bar{T} > 0$.

2 Adjoint equations

Following the PMP's framework and discarding the so-called abnormal case, the Hamiltonian H is:

$$H(t, \mathbf{x}, \mathbf{p}, \mathbf{T}) = -L(t, \mathbf{x}, \mathbf{T}) + < \mathbf{p}_r, \mathbf{v} > + < \mathbf{p}_v, \frac{\mathbf{T}}{m} + \sum_i \mathbf{a}_i(t, \mathbf{r}, \mathbf{v}, m) > -\alpha |\mathbf{T}| p_m, \quad (4)$$

where $\mathbf{p} = (\mathbf{p}_r, \mathbf{p}_v, p_m)$ is the adjoint vector, which can be seen as an extension to the Lagrange multipliers prominent in finite-dimension optimization. Note that if $\alpha \neq 0$, H is not differentiable when $\mathbf{T} = 0$. However it is still a locally Lipschitz function in the control vector which is enough to apply the PMP. The latter states first of all that $\frac{d\mathbf{p}}{dt} = -\frac{\partial H}{\partial \mathbf{x}}$. From now on, let us assume that L does not explicitly depend on the state variables. One then has the following linear equations for the adjoint position and velocity:

$$\begin{aligned} \dot{\mathbf{p}}_r &= - < \mathbf{p}_v, \sum_i \frac{\partial \mathbf{a}_i}{\partial \mathbf{r}} > \\ \dot{\mathbf{p}}_v &= -\dot{\mathbf{p}}_r - < \mathbf{p}_v, \sum_i \frac{\partial \mathbf{a}_i}{\partial \mathbf{v}} > \end{aligned} \quad (5)$$

Furthermore, the PMP says that the optimal control maximizes the Hamiltonian at all times. Assuming that $L = L(t, |\mathbf{T}|)$, then it comes that the thrust force must be aligned with the adjoint velocity i.e. $|\mathbf{p}_v|\mathbf{T} = |\mathbf{T}|\mathbf{p}_v$. One can then rewrite Eq. (4) as:

$$H = -L + < \mathbf{p}_r, \mathbf{v} > + |\mathbf{p}_v| \frac{|\mathbf{T}|}{m} + < \mathbf{p}_v, \sum_i \mathbf{a}_i > -\alpha |\mathbf{T}| p_m \quad (6)$$

Now the adjoint mass equation can also be written:

$$\dot{p}_m = |\mathbf{T}| \frac{|\mathbf{p}_v|}{m^2} - < \mathbf{p}_v, \sum_i \frac{\partial \mathbf{a}_i}{\partial m} > \quad (7)$$

In Orekit, Eq. (5) and (7) are encoded in *(Field)CartesianAdjoint-DerivativesProvider*. The individual contributions to it (and to the Hamiltonian) from the different accelerations $\mathbf{a}_i$ are in the implementation of the interface *(Field)-CartesianAdjointTerm*. The native ones cover attraction from a point-mass body (central or not, as a third body or not) and the J_2 effect, as well as non-inertial forces (in a rotating frame).

Note that the PMP also states that $\dot{H} = \frac{\partial H}{\partial t}$, so when t_0 or t_f is not fixed, $-H$ can be seen as the adjoint of time. It also means that for an autonomous system (the dynamics not depending explicitly on time), the Hamiltonian is constant over the optimal trajectory.

At this point, the complete relationship between $\mathbf{T}$ and $\mathbf{p}$ is still undetermined and depends on the actual definition of L. This logic is to be implemented via the *(Field)CartesianCost* interface and the library comes with a few built-in cases, described next. Note that if one wishes to compute J along, one can use *getCostDerivativeProvider* to add to the propagator $\int L$ as an additional integration variable.

2.1 Fuel cost

A favored choice in trajectory design is to minimize the mass depletion a.k.a. the fuel expenditure. Let us choose the control vector as $\mathbf{w} = \mathbf{T}/\bar{T}$ and the cost integrand as $L = L_1 = \bar{T}|\mathbf{w}|$. In this case, one has:

$$H = \bar{T}|\mathbf{w}|S + < \mathbf{p}_r, \mathbf{v} > + < \mathbf{p}_v, \sum_i \mathbf{a}_i > \tag{8}$$

where $S = \frac{|\mathbf{p}_v|}{m} - \alpha p_m - 1$. The maximization of the Hamiltonian tells us that $|\mathbf{w}| = \bar{T}$ when $S > 0$ and zero otherwise. The control is said to be bang-bang and S is referred to as the switching function. Note that it is assumed here and will be also in the other cases that the S does not vanish on an interval, which would be called a singular arc.

In Orekit, to preserve the accuracy of numerical integration, events detectors are to be included in propagation to properly handle the singularities when the switches occur. This is why *CartesianCost* has a *getEventDetectors* method.

2.2 Energy cost

The previous case is difficult to solve for in practice, hence a usual alternative is the so-called cost energy, where L is proportional to the squared Euclidean norm of thrust (typically with a scaling factor $\frac{1}{2}$ although the exact form may vary). It is sub-optimal regarding fuel consumption and the gap is somewhat captured by the following Cauchy-Schwarz inequality:

$$\int_{t_0}^{t_f} |\mathbf{T}| dt \leq \sqrt{(t_f - t_0) \int_{t_0}^{t_f} |\mathbf{T}|^2 dt} \tag{9}$$

Let us now go over the different versions implemented in Orekit, as inheritors of *(Field)-AbstractCartesianEnergy*.

2.2.1 Unbounded thrust

In this subsection, let us consider that the thrust is not bounded i.e. $\bar{T} = +\infty$.

Zero mass flow If $\alpha \simeq 0$, the mass is considered constant and can be dropped from the state vector, as well as the corresponding adjoint variable. It is then more convenient to choose the control vector as the acceleration $\mathbf{u} = \frac{\mathbf{T}}{m}$ rather than the force, so that $L = L_2 : \mathbf{u} \mapsto \frac{1}{2}|\mathbf{u}|^2$. Plugging this into Eq.(4), it becomes:

$$H = -\frac{1}{2}|\mathbf{u}|^2 + < \mathbf{p}_v, \mathbf{u} > + < \mathbf{p}_r, \mathbf{v} > + < \mathbf{p}_v, \sum_i \mathbf{a}_i > \tag{10}$$

The maximization readily gives $\mathbf{u} = \mathbf{p}_v$. Thus the control law is now smooth. Note that an advantage of choosing $\mathbf{u}$ over $\mathbf{T}$ is that this relationship with the adjoint vector does not depend on the mass at all. Another one is that the vanishing control $\mathbf{p}(t) = 0 \, \forall t$ can be used as initial guess in shooting methods (see 3 for details on this technique).

Non-zero mass flow Let us go back to $\alpha \neq 0$ and the control vector being $\mathbf{T}$, so that $J = \frac{1}{2} \int |\mathbf{T}|^2$. The Hamiltonian is a polynomial of degree 2 w.r.t. the thrust magnitude. Let us define $\tilde{S} = S + 1 = \frac{|\mathbf{p}_v|}{m} - \alpha p_m$, so that:

$$H = -\frac{1}{2}\left(|\mathbf{T}| - \tilde{S}\right)^2 + \frac{1}{2}\tilde{S}^2 + < \mathbf{p}_r, \mathbf{v} > + < \mathbf{p}_v, \sum_i \mathbf{a}_i > \tag{11}$$

When $\tilde{S}$ is negative, then maximizing H is equivalent to $|\mathbf{T}| = 0$. Otherwise, $|\mathbf{T}| = \tilde{S}$. So the control law is continuous but when $\tilde{S}$ vanishes, its derivative is not.

2.2.2 Bounded thrust

Let us now consider that the thrust magnitude has an upper bound $\bar{T} < +\infty$. In this case, let us define the control vector as $\mathbf{w} = \mathbf{T}/\bar{T}$. The maximization of the Hamiltonian is identical to the previous analysis, except when $\tilde{S} > \bar{T}$, in which case $|\mathbf{T}| = \bar{T}$.

2.3 Penalized fuel cost

An intermediate between the fuel cost and a smoother one can be sought. This can be done by adding a penalty term $\varepsilon \bar{T} G(|\mathbf{u}|)$ to L, where ε is a weight in $[0, 1]$. In a homotopic approach, one would start at 1 with the full penalty and move towards the left, solving problems closer and closer to the one defined in 2.1 (see [1] for more details). The Hamiltonian then reads:

$$H = \bar{T}|\mathbf{w}|\left(-1 - \alpha p_m + \frac{|\mathbf{p}_v|}{m}\right) - \varepsilon \bar{T} G(|\mathbf{w}|) + < \mathbf{p}_r, \mathbf{v} > + < \mathbf{p}_v, \sum_i \mathbf{a}_i > \tag{12}$$

In Orekit, *(Field)PenalizedCartesianFuelCost* offers the framework for any penalty. There are two native implementations, as described next.

2.3.1 Quadratic penalty

With $G = G_q : a \mapsto \frac{1}{2}\bar{T}a^2 - a$, one gets the same problem than 2.2.2 when $\varepsilon = 1$. For $\varepsilon > 0$, the Hamiltonian is:

$$H = -\frac{\varepsilon}{2}\left(\bar{T}|\mathbf{w}| - S_\varepsilon\right)^2 + \frac{\varepsilon}{2}S_\varepsilon^2 + < \mathbf{p}_r, \mathbf{v} > + < \mathbf{p}_v, \sum_i \mathbf{a}_i > \tag{13}$$

where $S_\varepsilon = \frac{1}{\varepsilon}\left(\frac{|\mathbf{p}_v|}{m} - \alpha p_m - 1\right) + 1$. The maximization of the Hamiltonian is very similar to 2.2.2 and implies the following:

$$|\mathbf{w}| = \begin{cases} 0 & \text{if } S_\varepsilon < 0 \\ S_\varepsilon/\bar{T} & \text{if } 0 \leq S_\varepsilon < \bar{T} \\ 1 & \text{if } S_\varepsilon \geq \bar{T} \end{cases} \tag{14}$$

2.3.2 Logarithmic barrier

Another choice for the penalty is $G = G_l : a \mapsto \log(a) + \log(1-a)$. It is not defined for $\varepsilon = 0$ or 1 and hence is referred to as a barrier function. The derivative of the Hamiltonian w.r.t. the norm is:

$$\frac{dH}{d|\mathbf{w}|} = \bar{T}(S - \varepsilon G') \tag{15}$$

Thus $1/|\mathbf{w}|$ is solution of the following equation in the unknown x on $[1, +\infty)$:

$$x + \frac{1}{1 - x^{-1}} = \frac{S}{\varepsilon} \tag{16}$$

so it is the largest root of a second-order polynomial:

$$x^2 - 2\left(\frac{S}{2\varepsilon} + 1\right)x + \frac{S}{\varepsilon} \tag{17}$$

After some algebraic manipulations it comes that:

$$|\mathbf{w}| = \frac{2\varepsilon}{S + 2\varepsilon + \sqrt{S^2 + 4\varepsilon^2}} \tag{18}$$

It is worth mentioning that the control law is smooth in this case. Beware that a homotopic method cannot start at 1, but it can still use the quadratic penalty case for instance to be initialized for some value of ε.

2.4 Flight duration cost

Another cost of choice in trajectory design is the time of flight, when one wants to reach the target as soon as possible. In that case $J = t_f - t_0 = \int dt$. This implies that t_f is free. Now assuming that all the $\mathbf{a}_i$ are independent of m, then $\dot{p}_m \geq 0$ and the transversality condition on the adjoint mass ($p_m(t_f) = 0$, see 3) implies that it is a negative function. Hence $\tilde{S}$ is always positive and the maximization of the Hamiltonian gives $|\mathbf{T}| = \bar{T}$. The optimal control consists in firing at full thrust constantly.

3 Indirect solver

3.1 Single shooting

The PMP also provides so-called transversality equations, linked to the boundary conditions on the state. If the terminal Cartesian variables are fixed, then the corresponding adjoint variables are free. If the terminal mass is free, then the adjoint mass must vanish. Thus for fixed times (t_0, t_f) (hence excluding the case of 2.4), fixed initial state $(\mathbf{x}_0, m_0)$ and fixed terminal Cartesian vector $\mathbf{x}_f$, then the only non-trivial transversality equation is $p_m(t_f) = 0$. This can be seen as solving a two-point boundary value problem, in the state and adjoint space. The shooting approach consists in starting from a guess on the value of $\mathbf{p}_0 = \mathbf{p}(t_0)$ and iterate on it via differential correction until the conditions are satisfied. The single version uses propagation from t_0 to t_f (in contrast to splitting the interval for robustness purposes as done in so-called multiple shooting). In summary, it requires solving for the following non-linear system:

$$\begin{cases} \mathbf{x}(t_f) - \mathbf{x}_f &= \mathbf{0} \\ p_m(t_f) &= 0 \end{cases} \tag{19}$$

seen as a function of $\mathbf{p}_0$.

3.2 Implementation details

In Orekit, finding a zero of the shooting function is done with a simple Newton-Raphson approach, encoded in *NewtonFixedBoundaryCartesianSingleShooting* (note that its ancestor class *AbstractIndirectShooting* is very generic and can be leveraged upon for custom use). It is worth mentioning that it works fine with backward propagation ($t_0 > t_f$), which is useful when the mass is known at the end rather than at the beginning.

The present implementation of single shooting utilizes Automatic Differentiation to obtain the derivatives needed by the Newton update. In practice, it means that $\mathbf{p}_0$ defines the variables of a so-called Taylor Differential Algebra, in which all steps of the integration scheme are performed. When a switch occurs, it requires special care: one needs to introduce an extra intermediate variable, as done for example in [5] and explained thereafter. Let t_s be the time when the switching function $(t, \mathbf{x}, \mathbf{p}) \mapsto g(t, \mathbf{x}, \mathbf{p})$ reaches a critical value, say 0, detected during integration and triggering a change in the control law. From propagation up to t_s, one has a Taylor expansion of $(\mathbf{x}, \mathbf{p})(t_s)$ at order 1 in $\mathbf{p}_0$. Let us then introduce $\mathrm{d}t = t - t_s$ as an additional independent variable. By evaluating the differential equations in this new algebra, one can form the adequate expansion of the state $\mathbf{x}(t_s) + \dot{\mathbf{x}}(t_s)\mathrm{d}t$ and the adjoint, before evaluating g. Let us now now swap $\mathrm{d}t$ and g as variables, the latter becoming independent. This is done by inverting a so-called Taylor map [2], which at first-order boils down to a matrix inversion. Next, by setting the variation of g to zero, one goes back to the original algebra and gets the derivatives taking into account the fact that the switch occurred. The propagation can finally resume with the next integration step.

3.3 Examples

The reader can refer to the *FixedBoundarySingleShooting* file in the Orekit tutorials repository for examples with the energy cost applied to heliocentric trajectories. In particular, it is shown how the solver can be used sequentially, adding up constraints, using the obtained adjoint vector as initialization for the next one, starting with the cost from 2.2.1 and a no-control guess $\mathbf{p} = 0$. Note that the shooting method works with both *Orbit*- (with *CARTESIAN OrbitType*) and *AbsolutePVCoordinates*-based numerical propagation.

References

[1] R. Bertrand and R. Epenoy. New smoothing techniques for solving bang–bang optimal control problems—numerical results and statistical interpretation. *Optimal Control Applications and Methods*, 23(4):171–197, 2002.

[2] M. Berz. *Modern Map Methods in Particle Beam Physics.* Academic Press, 1999.

[3] M. Cerf. Optimization techniques ii: Discrete and functional optimization. In *Optimization Techniques II.* EDP Sciences, 2023.

[4] G. Colasurdo and L. Casalino. Indirect methods for the optimization of spacecraft trajectories. *Modeling and Optimization in Space Engineering*, pages 141–158, 2013.

[5] P. Di Lizia, R. Armellin, A. Morselli, and F. Bernelli-Zazzera. High order optimal feedback control of space trajectories with bounded control. *Acta Astronautica*, 94(1):383–394, 2014.

[6] E. Trélat. Optimal control and applications to aerospace: some results and challenges. *Journal of Optimization Theory and Applications*, 154:713–758, 2012.