AAS 19-297

A SURVEY OF THE METHODS AVAILABLE FOR THE DESIGN OF MANY-REVOLUTION LOW-THRUST PLANETOCENTRIC TRAJECTORIES

Pradipto Ghosh*

The mission use of electric propulsion (EP) systems providing continuous, low thrust has steadily increased over the past several years. As EP technology continues to mature, the upward trend in its efficient utilization is expected to persist. A particularly challenging and long-studied low-thrust trajectory design problem is Electric Orbit Raising, i.e. the design of many-revolution, long duration transfers in the planetocentric space. Numerous methods, both analytical and numerical, optimal and sub-optimal, open-and-closed-loop, and involving models of various degrees of complexity, have been proposed. This paper attempts to characterize and classify the more popular of these methods and their variants.

INTRODUCTION

Investigation into the methods for the design of multi-revolution, low-thrust trajectories in the planetocentric space has received fresh impetus in recent years, due, in no small parts, to the rapid development and increasing adoption of Electric Propulsion (EP) technology in missions involving Geostationary communication satellites. Continuous thrust generation using EP is not a new concept *per se*, having been first tested in flight in the 1960's.[1] Even though the primary advantage of low thrust levels, namely, a large accumulated ΔV while expending relatively little propellant mass compared to chemical propulsion systems, was recognized early on, its adoption has historically been gradual. In the 1980's and 90's, EP found applications in station-keeping maneuvers for GEO satellites, but its serious use as primary propulsion for orbit raising applications was deferred until the early 2000's, perhaps mainly because of its perceived risk as a still-maturing technology, and also because of delayed revenues resulting from long transfer duration. In 2001, the Artemis mission fired its ion engines for 18 months, a major portion of its GTO to GEO journey. Launched in 2003, SMART-1 was the first spacecraft to complete a full GTO to GEO transfer and beyond (a lunar orbit) using low-thrust propulsion exclusively. Since then, several all-electric, low-thrust orbit raising missions have flown successfully, one of the more recent ones being the 4-month-long transfer of the Eutelsat 172B (launched June 2017) to GEO using only EP, in particular, a SPT-100 Hall Effect Thruster.[2]

The problem of designing multi-revolution (hundreds or even thousands of revolutions) low-thrust trajectories around a central body, is, however, non-trivial, research into which started more than half a century ago and continues to this day.[3–6] Starting with the early work of Lawden and Edelbaum, the majority of trajectory design methods have tackled the problem from an optimization, although not necessarily from a modern Optimal Control Theory, perspective.[3,4] Since computational power was at a premium in the early days, especially in 1960's, 70's and the early 80's, many of the early trajectory optimization algorithms are based on lower-order and/or one-revolution-averaged dynamical models. Averaging the dynamics allowed larger integration step sizes, and a smaller state-space dimension allowed for the derivation of convenient, semi-analytical formulae for estimating the maneuver duration, ΔV, and the thrust program.

Propellant and flight-time optimization of spiraling, low-thrust trajectories using indirect (Pontryagin's Minimum Principle) and direct (e.g. transcription-with-nonlinear-programming) methods have been reported in numerous works. Developed in the mid-1970's and still used by NASA for geocentric trajectory

*Astrodynamics Engineer, Analytical Graphics Inc., pghosh@agi.com, Senior Member, AIAA

optimization, SEPSPOT is an indirect-method-based software that utilizes orbital averaging instead of direct integration of the osculating orbital dynamics to keep program run time in check.[7] A relatively early work by Scheel and Conway used a Runge-Kutta (R-K) parallel shooting method to compute an equatorial LEO-to-GEO transfer trajectory with about 100 revolutions.[8] Their model was of medium fidelity, involving lunar perturbation and the Earth J_2 effect. However, applying a direct method to many-revolution transfers has its challenges, the greatest of which is the large number of decision variables (upto hundreds of thousands) and constraints (of similar order). Generating an initial guess for such a direct method is also non-trivial, absent an *a priori* intuition for a simple thrusting strategy, in terms of thrust spherical angles in a satellite-centric frame, for instance. In spite of this, several authors have solved the many-revolution optimization problem using various flavors of direct collocation with nonlinear programming, including Betts, who reports a 578-revolution spiral using the $\mathbb{SOCS}$ software.[9] More recently, Graham and Rao have solved month-long, minimum-time LEO-to-GEO transfer problems using a variable-order Legendre-Gauss-Radau orthogonal collocation method.[10] Aziz *et al.* have used a Hybrid Differential Dynamic Programming (HDDP) algorithm to optimize a 500-revolution spiral trajectory under J_2 and lunar gravity.[11]

Apart from trajectory design from within pure Calculus of Variations (CoV) or Computational Optimal Control frameworks, there are other methods as well, such as those based on the "blending" of thrust vectors that instantaneously maximize orbit element rates, examples of which are the Kleuver method and Q-law.[12,13] Such methods tend to be computationally more tractable compared to collocation-based ones modeling osculating dynamics, yet, at the same time allow accounting for practical mission scenarios such as intermittent thrusting, thus making them attractive as "seed" methods for high-fidelity, high-dimensional optimization-based methods.

In the light of the foregoing discussion, the low-thrust, planetocentric, many-revolution trajectory design methods surveyed in this paper are classified into the following three categories:

i) Simplified Model-based Analytical/Semi-analytical Methods (SMASM)

ii) Computational Optimal Control Methods (COpC)

iii) Sub-optimal Trajectory Design Methods (SubTD)

There is considerable variation among these trajectory design formalisms in terms of the fidelity of the physics modeled and the assumptions made in order to produce a solution. Some account for eclipsing and accordingly introduce a natural bang-off-bang thrusting pattern expected from a typical Solar Electric Propulsion (SEP) engine, while some others that assume constant thrust acceleration over the transfer duration. Again, some formulations consider more realistic engine models than others, with a throttable thrust magnitude, a dedicated power processing unit, and solar cell performance degradation in the Van Allen radiation belts, while there are others assume ideal models of Constant Thrust and Specific Impulse engines. In terms of the resulting control policy, some of the surveyed methods provide feedback laws useful for guidance purposes, while the majority solve the open-loop problem from which feedback policies may subsequently be synthesized. In a similar vein, some of these methods formulate path constraints, such as thrust direction or minimum periapsis, while others do not. Thus, a developer interested in implementing a low-thrust, multi-revolution, trajectory design library, or integrating an existing one into a larger software system, has several options at their disposal. If the intention is to develop a quick trajectory "calculator" for initial phases of a mission, perhaps with a view to implementing more complex methods at a later stage, some of the reduced-order semi-analytical methods (SMASM) or sub-optimal ones (SubTD) may be sufficient. On the other hand, a high-fidelity, high-precision library may be designed based on collocation methods, one that will intelligently select a computationally inexpensive "seed" method depending upon a user setting, such as the choice of whether to account for shadowing or whether a rendezvous solution is requested instead of an orbit transfer solution. The purpose of this paper is not to present an exhaustive review of literature dealing with low-thrust, multi-revolution spiral trajectory design methods, nor to document the algorithmic details of each method, but rather to summarize and categorize the more common ones with the hope of providing initial guidance to the trajectory design and optimization software tool developer in selecting a tool or method appropriate for their specific requirement in terms of development effort and fidelity/accuracy goals.

SIMPLIFIED MODEL-BASED ANALYTICAL/SEMI-ANALYTICAL METHODS

The Edelbaum-Kèchichian Method

In his seminal analysis, Edelbaum presented analytical solutions to the problem of maximizing the inclination change between two circular orbits of specified semi-major axes under constant thrust acceleration and fixed transfer time.[3] With the thrust yaw angle (control) held constant per revolution, Edelbaum used the following one-revolution-angle-averaged, circular orbit GVEs for velocity and inclination:

$$\overline{\frac{di}{dt}} = \frac{2}{\pi}\frac{\alpha \sin \mathscr{y}}{V} \tag{1}$$

$$\overline{\frac{dV}{dt}} = -\alpha \cos \mathscr{y} \tag{2}$$

and then posed the fixed-time inclination-maximization problem in the form of an ordinary parameter optimization problem using velocity as the independent variable:

$$\max_{\mathscr{y}} \int_{V_0}^{V_f} \overline{\frac{di}{dV}} dV, \;\; \text{s.t} \int_{V_0}^{V_f} \overline{\frac{dt}{dV}} dV = \text{constant} \tag{3}$$

Although Eq. (3) results in an expression for the optimal control angle $\mathscr{y}^*$, and closed-form formulae for the maximum inclination change $\Delta\bar{i}^*$ and the maneuver $\Delta\bar{V}^*$, the formula for $\Delta\bar{i}^*(\bar{V})$ is a piece-wise function involving a somewhat inconvenient conditional expression (Eqs. (44a-b) in reference [3]). It must also be noted that the orbit is assumed to remain quasi-circular during the maneuver, an assumption that can be justified if the constant yaw angle switches sign at the orbital anti-nodes, effectively making $\Delta\bar{h} = 0, \ \Delta\bar{k} = 0$ (and also $\Delta\bar{p} = 0, \ \Delta\bar{q} = 0$) per revolution. Later, Kèchichian, using time as the independent variable, reformulated the original Edelbaum problem as an Optimal Control Problem and, through direct application of the PMP, obtained a closed form expression for the control history $\mathscr{y}^*(t)$ valid for the entire transfer duration.[14] Apart from resulting in an ephemeris-generating algorithm, the Kèchichian formulation also provides a single formula for the evolution of the orbit inclination as function of time. The Optimal Control formulation consists in minimizing the Lagrange cost (maneuver duration, and for constant acceleration, also minimum ΔV) $\int_0^{t_f} 1.dt$ subject to Eqs. (1, 2). For a spacecraft equipped with a constant-thrust-acceleration engine, the procedure for generating a minimum-time, multi-revolution trajectory between the given initial conditions $\{a_0, \ i_0\}$ and final conditions $\{a_f, \ i_f\}$, can be summarized as follows:

i) Compute the initial optimal yaw angle from:

$$\tan \mathscr{y}_0^* = \frac{\sin(\frac{\pi}{2}|i_f - i_0|)}{\sqrt{\frac{a_f}{a_0}} - \cos(\frac{\pi}{2}|i_f - i_0|)} \tag{4}$$

ii) Calculate the simulation/maneuver duration from:

$$t_f^* = \frac{1}{\alpha}\sqrt{\frac{\mu}{a_0}}\Big(\cos \mathscr{y}_0^* - \frac{\sin \mathscr{y}_0^*}{\tan[\frac{\pi}{2}|i_f - i_0| + \mathscr{y}_0^*]}\Big) \tag{5}$$

iii) Compute the optimal thrust yaw and pitch angles from:

$$\mathscr{y}^*(t) = \frac{\sqrt{\frac{\mu}{a_0}} \sin \mathscr{y}_0^*}{\sqrt{\frac{\mu}{a_0}} \cos \mathscr{y}_0^* - \alpha t} \tag{6}$$

$$\mathscr{p}^*(t) = 0 \tag{7}$$

iv) Numerically integrate the *equinoctial* GVEs forward in the $[0, t_f]$ interval with the known initial states using controls Eqs. (6,7). However, during simulation, care must be taken to only change $\mathscr{y}^*$ from one revolution to the next while holding it constant within a revolution. Also, the yaw sign must switch at the anti-nodes to prevent drift in the average eccentricity and line of nodes under the above two-body-and-thrust model.

Analytical expressions are also available for the semi-major axis and inclination as function of time:

$$a^*(t) = \frac{a_0}{1 + \frac{a_0}{\mu}\alpha^2 t^2 - 2\sqrt{\frac{a_0}{\mu}}\alpha t \cos \mathcal{Y}_0^*} \tag{8}$$

$$|\Delta i^*(t)| = \frac{2}{\pi}\Big[\tan^{-1}\Big(\frac{\alpha t - \sqrt{\frac{\mu}{0}}\cos \mathcal{Y}_0^*}{\sqrt{\frac{\mu}{a_0}}\sin \mathcal{Y}_0^*}\Big) + \frac{\pi}{2} - \mathcal{Y}_0^*\Big] \tag{9}$$

Note that as is the case with many trajectory optimization methods based on angular-position-averaged dynamical models, the Edelbaum-Kèchichian method is not intrinsically suitable for rendezvous problems because the spacecraft position in the orbit does not appear in the dynamics. However, the method has proved suitable for quick estimation of flight time and propellant consumption even for design techniques formulated under less restrictive assumptions, including cases in which shadows are modeled.[15,16]

Kèchichian has extended the analytic averaging approach for circular orbits by accounting for the averaged J_2 effect.[17] An extra state, Ω, was appended Eqs. (1, 2) by including its averaged rate $\overline{\frac{d\Omega}{dt}}$ due to J_2. With this added effect, the orbit now rotates around the instantaneous line of nodes, which itself exhibits precession. The averaged, 3-state trajectory optimization problem is again formulated and solved via an indirect method, with piecewise constant thrust yaw angle switching signs at the relative anti-nodes, although closed-form analytical expressions such as those presented in Eqs. (4- 9) are no longer available. The corresponding semi-analytical method is therefore rendered somewhat less competitive relative to other purely numerical, averaging-based direct methods discussed below. In reference [15], Kleuver presents a semi-analytic algorithm for predicting the flight-time for circle-to-circle inclination change maneuvers in the presence of shadows. Key results from the Edelbaum-Kèchichian formulation are utilized, but the Kleuver method does not produce an ephemeris and thrust attitude history that could be exploited by a COpC method. An analytical method for quasi-circular orbit transfer in the presence of shadows was developed in reference [18] by extending the Edelbaum method described above. Thrust orientation is constrained to maintain circularity. In reference [19], an analytical method is developed from geometric considerations for circular orbit raising in the presence of earth shadow. The thrusting strategy is to maximize the Δa per revolution while keeping Δe to zero under eclipsing. However, both of these methods consider the special case of *co-planar* transfers only, and are sub-optimal due to steering losses incurred in constraining the eccentricity.

The Wiesel-Alfano Method

Yet another semi-analytical, low-thrust orbit transfer algorithm based on a two-state dynamical model was developed by Wiesel and Alfano.[20] Similar to the original Edelbaum method, the Wiesel-Alfano method also determines an inclination-maximizing yaw steering control law for a *single revolution* circular transfer, but instead of using averaged dynamics, it uses an *osculating* two-state model. This results in a continuously-varying yaw profile in each revolution as opposed to Edelbaum's square-wave yaw profile (constant magnitude, sign switches at anti-nodes). Furthermore, in separately considering a "fast time scale" problem (per orbit) and a "slow time scale" problem (the entire transfer), the Wiesel-Alfano method is able to remove the constant acceleration restriction inherent in the Edelbaum-Kèchichian method, replacing it instead with a more realistic constant-thrust-constant-I_{sp} engine model. Acceleration is assumed to be constant only within a revolution, but increases from revolution to revolution, thus accounting for propellant loss. Earth-eclipsing is not modeled, thereby limiting the method's utility for SEP missions.

The osculating counterparts of Eqs. (1, 2) with f as the independent variable and $\{i, a\}$ as the dependent ones are:

$$\frac{di}{df} = \frac{a^2}{\mu}\alpha \sin \mathscr{y}(f) \cos f \tag{10}$$

$$\frac{da}{df} = \frac{2a^3}{\mu}\alpha \cos \mathscr{y}(f) \tag{11}$$

Note that as before, pure tangential thrusting is assumed, thus $\mathscr{p} = 0$. Also, α is the engine acceleration magnitude assumed constant over the short time scale of one orbit, but not on the longer time scale, as explained below. The objective of the fast time scale problem is to program $\mathscr{y}(f)$ so as to maximize the one-revolution inclination change with a specified change in semi-major axis, Δa:

$$\max_{\mathscr{y}} \mathscr{J}(\mathscr{y}) = \int_0^{2\pi} \frac{a^2}{\mu}\alpha \sin \mathscr{y} \cos f df, \quad \text{s.t}\ \mathscr{C}(\mathscr{y}) = \int_0^{2\pi} \frac{2a^3}{\mu}\alpha \cos \mathscr{y} df - \Delta a = 0 \tag{12}$$

The standard one-variable optimization necessary condition $\frac{\partial}{\partial \mathscr{y}}(\mathscr{J} + \lambda \mathscr{C}) = 0$ leads to the following optimal yaw program valid per orbit with initial semi-major axis a:[20]

$$\mathscr{y}^*(f) = \tan^{-1}\Big(\frac{\cos f}{\sqrt{\frac{1}{u} - 1}}\Big), \text{ where } u := \frac{1}{4\lambda^2 a^2 + 1} \tag{13}$$

With this control, analytical expressions are derivable for the changes in a, i, e and Ω per orbit, i.e. over intervals of $\delta \mathscr{T} = 2\pi\sqrt{\frac{a^3}{\mu}}$ ([20]):

$$\delta a = \frac{8a^3\alpha}{\mu}\sqrt{1-u}\, K(u), \quad K(\cdot): \text{ elliptic integral of the first kind} \tag{14}$$

$$\delta i = \frac{4a^2\alpha}{\mu}\Big[\sqrt{\frac{1}{u}}\, E(u) + \Big(\sqrt{u} - \sqrt{\frac{1}{u}}\Big)\, K(u)\Big], \quad E(\cdot): \text{ elliptic integral of the second kind} \tag{15}$$

$$\delta e = 0 \tag{16}$$

$$\delta \Omega = 0 \tag{17}$$

The long time scale dynamics can now be modeled by considering the per-orbit evolution of the states $\{a, i\}$ in the "long" time scale $\mathscr{T}$, over which the acceleration is no longer a constant:

$$\frac{\delta a}{\delta \mathscr{T}} = \frac{4a^{3/2}}{\pi\sqrt{\mu}}\frac{\alpha_0}{1 - \dot{m}\mathscr{T}}\sqrt{1-u}\, K(u) \tag{18}$$

$$\frac{\delta i}{\delta \mathscr{T}} = \frac{2}{\pi}\sqrt{\frac{a}{\mu}}\frac{\alpha_0}{1 - \dot{m}\mathscr{T}}\Big[\sqrt{\frac{1}{u}}\, E(u) + \Big(\sqrt{u} - \sqrt{\frac{1}{u}}\Big)\, K(u)\Big] \tag{19}$$

Here $\alpha_0 = \frac{|\boldsymbol{T}|}{m_0}$ is the initial thrust acceleration, $\boldsymbol{T}$ the thrust, and $\dot{m}$ the mass flow rate. Changing the independent variable in Eqs. (18, 19) from $\mathscr{T}$ to the velocity-like variable

$$\mathscr{V} = -\frac{\alpha_0}{\dot{m}} \ln(1 - \dot{m}\mathscr{T}) \tag{20}$$

Wiesel and Alfano solve the following minimum-ΔV transfer problem with $u(\mathscr{V})$ as the control variable:

$$\mathscr{W}\mathscr{A} \begin{cases} \min\limits_{u} \int_0^{\mathscr{V}_f} 1.d\mathscr{V} \\ \frac{\delta a}{\delta \mathscr{V}} = \frac{4a^{3/2}}{\pi\sqrt{\mu}}\sqrt{1-u}\, K(u) \\ \frac{\delta i}{\delta \mathscr{V}} = \frac{2}{\pi}\sqrt{\frac{a}{\mu}}\Big[\sqrt{\frac{1}{u}}\, E(u) + \Big(\sqrt{u} - \sqrt{\frac{1}{u}}\Big)\, K(u)\Big] \\ [a(0),\ i(0)]^T = [a_0,\ i_0]^T \text{specified} \\ [a(\mathscr{V}_f),\ i(\mathscr{V}_f)]^T = [a_f,\ i_f]^T \text{specified},\ \mathscr{V}_f \text{ free} \end{cases} \tag{21}$$

Solution of $\mathscr{W}\mathscr{A}$ via a numerical technique such as the shooting method would yield $a(\mathscr{V}_f), i(\mathscr{V}_f), u(\mathscr{V}_f)$. From the resulting $a, i, \mathscr{V}_f$ map, it is easy to read off the $\mathscr{V}_f$ value for a given set of terminal conditions, which in turn can be mapped to the total transfer time using Eq. (20). Note that unlike the Edelbaum-Kèchichian procedure which provided purely analytical formulas for estimating the transfer time, ΔV, $a(t)$ and $i(t)$, the Wiesel-Alfano method requires the solution of an optimal control problem for the long time scale transfer, although analytic expressions are available for some of the quantities of interest, e.g. one-revolution change in the elements, with the corresponding thrust program, Eqs. (13-17). A method for generating an ephemeris and control time history is however, not explicitly discussed in the Wiesel-Alfano paper. In order to produce this information, useful for higher-fidelity direct trajectory optimization methods, for example, the following procedure could be adopted:

i) Numerically solve the minimum-ΔV optimal control problem for a circle-to-circle transfer based on Eqs. (18, 19), i.e. use $\mathscr{T}$ as the independent variable instead of $\mathscr{V}$. This should yield $u(\mathscr{T})$

ii) Numerically integrate the GVEs (in Keplerian elements) forward with the pitch profile given by Eq. (13), interpolating to evaluate u at the current time step.

COMPUTATIONAL OPTIMAL CONTROL METHODS

An Averaging-based Indirect Method for Orbit Transfers

One of the earliest low-thrust geocentric trajectory optimization tools incorporating high-fidelity physics and sophisticated numerics was the Solar Electric Control Knob Setting Program by Optimal Trajectories or SECKSPOT (subsequently renamed to SEPSPOT and currently maintained by NASA Glenn Research Center) developed by Sackett, Malchow and Edelbaum.[7] SEPSPOT models eclipses to faithfully account for the operation of SEP engines, includes an option to account for the delay in thruster start-up upon exiting shadow, considers thrust level decay caused by solar cell power degradation due to the damaging effect of charged particle flux, incorporates the averaged J_2 effect, and has the ability to add thrust attitude constraints. In spite of being a sophisticated and time-tested software program, the main deficiency of SEPSPOT in the light of more recent (over the past three decades) developments in computational optimal control is the fact that it is based on an indirect method, and thus involves a user to provide, in addition to the initial (or from) and final (or to) orbit states, also initial guesses for co-states. As is well-known in the Optimal Control literature, the co-states are notoriously hard to estimate, which typically limits the usefulness of indirect method-based software.[21] To reduce simulation time, SEPSPOT utilizes numerically-averaged orbit states and co-states in the Euler-Lagrange formulation, which allows large integration steps (days) to be taken compared with an unaveraged system of differential equations (minutes). A brief outline of the SEPSPOT averaging principle follows.

Let $\boldsymbol{\xi} = [a\ h\ k\ p\ q]^T$ denote the five equinoctial elements of a satellite at a given instant. The Gauss Variational Equations (GVEs) describing the evolution of $\boldsymbol{\xi}$ can be compactly written as:

$$\dot{\boldsymbol{\xi}} = \alpha(t)\frac{\partial \boldsymbol{\xi}}{\partial \dot{\boldsymbol{r}}}\hat{\boldsymbol{u}} = \alpha(t)\mathbb{P}(\boldsymbol{\xi}, F)\hat{\boldsymbol{u}} \tag{22}$$

where $\mathbb{P}(\boldsymbol{\xi}, F) \in \mathbb{R}^{5\times 3}$ is the velocity partials matrix derived in detail by Sackett *et al.*[7] . For a minimum-time transfer problem with a Mayer cost, $\mathscr{J} = t_f$, the *unaveragaed* Hamiltonian is:

$$\mathscr{H} = \alpha(t)\boldsymbol{\lambda}_\xi^T\mathbb{P}(\boldsymbol{\xi}, F)\hat{\boldsymbol{u}} \tag{23}$$

where $\boldsymbol{\lambda}_\xi \in \mathbb{R}^{5\times 1}$ is a vector of costates associated with $\boldsymbol{\xi}$. From Pontryagin's Minimum Principle (PMP) the thrust program that point wise-minimizes the Hamiltonian, and the corresponding minimized unaveraged Hamiltonian are:

$$\hat{\boldsymbol{u}}^* = -\frac{\mathbb{P}^T(\boldsymbol{\xi}, F)\boldsymbol{\lambda}_\xi}{\|\mathbb{P}^T(\boldsymbol{\xi}, F)\boldsymbol{\lambda}_\xi\|} \tag{24}$$

$$\mathscr{H}^* = -\alpha(t)\left\|\mathbb{P}^T(\boldsymbol{\xi}, F)\boldsymbol{\lambda}_\xi\right\| \tag{25}$$

A first-order approximation to the states and the costates, $[\tilde{\boldsymbol{\xi}},\ \tilde{\boldsymbol{\lambda}}_\xi]$ which follow slower dynamics compared to their osculating counterparts is then derived from the *averaged* Hamiltonian $\mathscr{H}^*_{\text{av}}$ as follows. Define:

$$\mathscr{H}^*_{\text{av}} = \frac{1}{\mathscr{T}} \int_{t+\frac{\mathscr{T}}{2}}^{t-\frac{\mathscr{T}}{2}} \mathscr{H}^*(\tilde{\boldsymbol{\xi}},\ \tilde{\boldsymbol{\lambda}}_\xi, F(t))dt = \frac{1}{\mathscr{T}} \int_{-\pi}^{\pi} \mathscr{H}^*(\tilde{\boldsymbol{\xi}},\ \tilde{\boldsymbol{\lambda}}_\xi, F(t))\frac{dt}{dF}dF \tag{26}$$

Here $\mathscr{T}$ is the orbital period at fixed (constant) values of $\tilde{\boldsymbol{\xi}}$, and pure Keplerian motion is assumed over the averaging interval. Letting

$$w(\tilde{\boldsymbol{\xi}}) = \frac{1}{\mathscr{T}}\frac{dt}{dF} = \frac{1}{2\pi}(1 - \tilde{k}\cos F - \tilde{h}\sin F) \tag{27}$$

the Eq.(26) may be simplified to:

$$\mathscr{H}^*_{\text{av}} = \int_{-\pi}^{\pi} \mathscr{H}^*(\tilde{\boldsymbol{\xi}},\ \tilde{\boldsymbol{\lambda}}_\xi, F(t))w(\tilde{\boldsymbol{\xi}})dF \tag{28}$$

The Euler-Lagrange differential equations in terms of the first-order-approximate states and costates for the constraint-free problem under continuous thrusting are then derived in terms of the averaged Hamiltonian, which is itself computable using an $N-$point Gaussian quadrature:

$$\dot{\tilde{\boldsymbol{\xi}}} = (\frac{\partial \mathscr{H}^*_{\text{av}}}{\partial \tilde{\boldsymbol{\lambda}}_\xi})^T = \int_{-\pi}^{\pi} (\frac{\partial \mathscr{H}^*}{\partial \tilde{\boldsymbol{\lambda}}_\xi})^T w(\tilde{\boldsymbol{\xi}})dF \tag{29}$$

$$\dot{\tilde{\boldsymbol{\lambda}}}_\xi = -(\frac{\partial \mathscr{H}^*_{\text{av}}}{\partial \tilde{\boldsymbol{\xi}}})^T = -\int_{-\pi}^{\pi} [(\frac{\partial \mathscr{H}^*}{\partial \tilde{\boldsymbol{\xi}}})^T w + \mathscr{H}^*(\frac{\partial w}{\partial \tilde{\boldsymbol{\xi}}})^T]dF \tag{30}$$

$$\hat{\tilde{\boldsymbol{u}}}^* = -\frac{\mathbb{P}^T(\tilde{\boldsymbol{\xi}}, F)\tilde{\boldsymbol{\lambda}}_\xi}{\left\|\mathbb{P}^T(\tilde{\boldsymbol{\xi}}, F)\tilde{\boldsymbol{\lambda}}_\xi\right\|} \tag{31}$$

The two-point boundary value problem is solved using standard techniques,[21] but large integration steps are now allowed because the short-period variations in the states and costates have been eliminated through averaging. In SEPSPOT, this procedure has been generalized to include thrust attitude constraints, some environmental perturbations (J_2, SRP and $3^{rd.}$-body gravity, but not drag), shadow effects etc.[7] Note also, that since averaging of the dynamics have been performed relative to the spacecraft angular position, this particular formulation is not suitable for solving orbital rendezvous problems, where the target orbit specification would include all the six orbital states. Numerical examples of GTO-to-GEO transfers of spacecraft with an initial thrust acceleration level $\sim 10^{-4}\text{m/s}^2$, $I_{sp} = 2900$ seconds and transfer time 100-200 days are documented in reference [7].

Averaging-based Indirect Methods for Rendezvous

Averaging-based indirect methods for many-revolution problems, in addition to offering the convenience of reduced sensitivity to costate initial guesses, are also advantageous because they allow parameterization of an optimal control problem in terms of relatively few parameters, namely, initial costate values, other constant Lagrange multipliers, and the terminal time if free. Such methods are not only restricted to orbit transfer scenarios, but have also been extended to handle rendezvous cases.[22] In reference 22, the authors have used a multiple time scales formulation to reduce the original minimum-time rendezvous problem to an averaged optimal control problem, which is then formulated as a two-point boundary value problem. The resulting averaged control (the thrust steering angles) obtained by the application of PMP, is then substituted back into the osculating dynamics and numerically integrated forward. The minimum-time rendezvous problem in terms of osculating variables can be posed as:

$$\mathscr{P}\begin{cases}\min\limits_{|\boldsymbol{\alpha}|\leq\epsilon}\ \int_0^{t_f} 1.dt\\ \frac{d\boldsymbol{\zeta}}{dt}=\boldsymbol{f}(\boldsymbol{\zeta},L)\boldsymbol{\alpha}\\ \frac{dL}{dt}=g(\boldsymbol{\zeta},L)+\boldsymbol{h}^T(\boldsymbol{\zeta},L)\boldsymbol{\alpha}\\ [\boldsymbol{\zeta}(0),\ L(0)]^T=[\boldsymbol{\zeta}_0,\ L_0]^T\text{specified}\\ [\boldsymbol{\zeta}(t_f)-\boldsymbol{\zeta}_f,\ \psi(L(t_f),t_f)]^T=[0,\ 0]^T\text{specified},\ t_f\text{ free}\end{cases}\tag{32}$$

where ϵ, the thrust acceleration upper-bound, is a "small" parameter ($\sim 10^{-4}$ m/s^2 or less), $\boldsymbol{\zeta}=[n\ h\ k\ p\ q]^T$ are the slowly-evolving states, L is the fast state dominated by Keplerian dynamics, $\boldsymbol{f}(,\cdot,)$ is the (slow) state dynamics matrix periodic in L, $g(,\cdot,)$ and $\boldsymbol{h}(,\cdot,)$ are the two-body and the perturbation influences on L, respectively, and $\psi(\cdot)$ is a known function of the terminal longitude. See references [22, 23] for detailed mathematical expressions for the problem functions. Changing the independent variable to L, or more specifically to $\mathscr{L}:=\epsilon L$, introducing a new variable $\phi:=\epsilon t$, and normalizing the control vector to $\boldsymbol{\alpha}_s=\frac{\boldsymbol{\alpha}}{\epsilon}$, problem $\mathscr{P}$ can be recast into the following Mayer form:[22]

$$\tilde{\mathscr{P}}\begin{cases}\min\limits_{|\boldsymbol{\alpha}_s|\leq 1}\ \phi(\mathscr{L}_f)\\ \frac{d\boldsymbol{\zeta}}{d\mathscr{L}}=\frac{\boldsymbol{f}(\boldsymbol{\zeta},\frac{\mathscr{L}}{\epsilon})}{g(\boldsymbol{\zeta},\frac{\mathscr{L}}{\epsilon})}\boldsymbol{\alpha_s}\\ \frac{d\phi}{d\mathscr{L}}=\frac{1}{g(\boldsymbol{\zeta},\frac{\mathscr{L}}{\epsilon})}\\ [\boldsymbol{\zeta}(\mathscr{L}_0),\ \phi(\mathscr{L}_0)]^T=[\boldsymbol{\zeta}_0,\ 0]^T\text{specified}\\ [\boldsymbol{\zeta}(\mathscr{L}_f)-\boldsymbol{\zeta}_f,\ \psi(\phi(\mathscr{L}_f),\mathscr{L}_f)]^T=[0,\ 0]^T\text{specified},\mathscr{L}_f\text{ free}\end{cases}\tag{33}$$

Note that although the derivatives in the LHS of problem $\tilde{\mathscr{P}}$ are taken with respect to regular or slow' "time" $\mathscr{L}$, the dynamics are a function of the fast "time" $\frac{\mathscr{L}}{\epsilon}$, which, in turn, means that $\tilde{\mathscr{P}}$ falls under the category of a more general two-timescale optimal control problem:[24]

$$\mathscr{P}^\epsilon\begin{cases}\min\limits_{\boldsymbol{u}\in\mathscr{U}}\ \int_0^{t_f}\mathbb{L}(\boldsymbol{x}(t),\boldsymbol{u}(t),t,\frac{t}{\epsilon})dt\\ \text{s.t }\frac{d\boldsymbol{x}}{dt}=\boldsymbol{F}(\boldsymbol{x}(t),\boldsymbol{u}(t),t,\frac{t}{\epsilon})\\ \boldsymbol{x}(0)=\boldsymbol{x}_0,\ \Psi(\boldsymbol{x}(t_f),t_f)\text{ specified}\end{cases}\tag{34}$$

which itself is a perturbation of the *averaged* optimal control problem:

$$\overline{\mathscr{P}}\begin{cases}\min\limits_{\boldsymbol{v}\in\mathscr{V}}\ \int_0^{t_f}\overline{\mathbb{L}}(\bar{\boldsymbol{x}},\boldsymbol{v}(\cdot),t)dt\\ \text{s.t }\frac{d\bar{\boldsymbol{x}}}{dt}=\overline{\boldsymbol{F}}(\bar{\boldsymbol{x}},\boldsymbol{v}(\cdot),t)\\ \bar{\boldsymbol{x}}(0)=\boldsymbol{x}_0,\ \Psi(\bar{\boldsymbol{x}}(t_f),t_f)\text{ specified}\end{cases}\tag{35}$$

Here, $\bar{\boldsymbol{x}}$ is the *averaged* state with dependence only on the slow/normal time scale, $\boldsymbol{v}(t,\frac{t}{\epsilon})$ is the *averaged* control with feedback dependence on the fast time, and $\overline{\boldsymbol{F}}$ and $\overline{\mathbb{L}}$, are the *averaged* dynamics and Lagrange cost integrand respectively:

$$\overline{\boldsymbol{F}} = \frac{1}{\mathscr{T}}\int_0^{\mathscr{T}}\boldsymbol{F}(\bar{\boldsymbol{x}},\boldsymbol{v}(t,\theta),t,\theta)d\theta\tag{36}$$

$$\overline{\mathbb{L}} = \frac{1}{\mathscr{T}}\int_0^{\mathscr{T}}\mathbb{L}(\bar{\boldsymbol{x}},\boldsymbol{v}(t,\theta),t,\theta)d\theta\tag{37}$$

with $\overline{\boldsymbol{F}}$ periodic in $\theta=\frac{t}{\epsilon}$ with period $\mathscr{T}$. It can be shown that under certain assumptions, the control obtained by solving the two-point boundary value problem based on the averaged Hamiltonian system:

$$\frac{d\bar{\boldsymbol{x}}}{dt} = \overline{\boldsymbol{F}}(\bar{\boldsymbol{x}},\boldsymbol{v}(\cdot),t)\tag{38}$$

$$\frac{d\bar{\boldsymbol{\lambda}}}{dt} = -(\frac{\partial\overline{\boldsymbol{F}}^T}{\partial\bar{\boldsymbol{x}}}\bar{\boldsymbol{\lambda}}+\frac{\partial\overline{\mathbb{L}}}{\partial\boldsymbol{x}})\tag{39}$$

with appropriate boundary conditions is nearly optimal for the original problem $\mathscr{P}^\epsilon$ (or $\tilde{\mathscr{P}}$).[22–24] Summarizing, the above-described method utilizes an averaged dynamical system *including* the satellite angular position as an independent variable, allowing for the specification of the final true longitude, as is required for rendezvous problems. Additionally, the two-timescale averaging technique results in better numerical conditioning in the sense that numerical integration of Eqs. (38, 39) requires a step size $\sim h$ instead of $\sim \frac{h}{\epsilon}$ for the TPBVP corresponding to problem $\tilde{\mathscr{P}}$. This averaging-based indirect method has been extended solve both minimum-fuel and minimum-time problems, and included eclipsing using a cylindrical shadow model. Earth oblateness and thrust-direction constraints with Constant Specific Impulse (CSI) engines. For GTO-to-GEO rendezvous and orbit transfer test cases with initial acceleration level $\sim 10^{-4}\mathrm{m/s}^2$ and $I_{sp} = 2000$ seconds, involving hundreds of revolutions and days, see references [22, 23]. These principles are implemented in the low-thrust optimization software MIPELEC developed by the French space agency CNES.

The T_3D optimal low-thrust trajectory optimization package for multi-revolution, transfers also uses averaging techniques to compute minimum-time and minimum-fuel orbit transfer and rendezvous problems.[25] However, its main difference with the MIPELEC implementation is that it retains time as an independent variable, while keeping L as a state. This is advantageous in evaluating time-dependent perturbations in the force model.

An Averaging-based Direct Method For Orbit Transfers

In reference [16], Kleuver has used a direct shooting method to solve both minimum-time and minimum-propellant multi-revolution orbit transfer problems. In that work, the *structure* of the controls (the thrust program) is first derived from PMP as functions of costates. Then the costates, which are treated as optimization decision variables, are expressed in terms of node polynomial coefficients.The co-state parameterized controls are finally substituted back into the GVEs, which are then iteratively propagated forward in time until the associated cost function is numerically minimized and the terminal constraints met. The result of this process is expected to be the optimal node-polynomial coefficients. In Kleuver's analysis, the control structure is derived based on reduced-order dynamics in the $a - e - i$ space following the argument that initial and terminal conditions for most transfer problems are expressed only in terms of these elements An extension of this method using the modified equinoctial elements in presented in reference [26], in which the authors use collocation instead of a direct shooting approach.

Let $\boldsymbol{x} = [a\ e\ i\ \Omega\ \omega\ f\ m]^T \in \mathbb{R}^7$ be the satellite state evolving according to the mass-augmented GVEs compactly expressed as:

$$\frac{d\boldsymbol{x}}{dt} = \mathscr{F}(\boldsymbol{x}, \boldsymbol{\alpha}, t) \tag{40}$$

Let $\boldsymbol{x}_r = [a\ e\ i]^T \subset \boldsymbol{x}$. The Mayer cost (minimum-time/propellant) Hamiltonian for $\boldsymbol{x}_r$ is

$$\mathscr{H}(\boldsymbol{x}_r, \boldsymbol{\lambda}_r, \mathscr{p}, \mathscr{y}) = \lambda_a \dot{a} + \lambda_e \dot{e} + \lambda_i \dot{i} \tag{41}$$

and, with the thrust acceleration resolved in the NTW or Frenet frame, the GVEs for $a,\ e,\ i$ are:

$$\frac{da}{dt} = \frac{2\alpha a^2 v}{\mu} \cos \mathscr{p} \cos \mathscr{y} \tag{42}$$

$$\frac{de}{dt} = \frac{\alpha}{v}[2(e + \cos f) \cos \mathscr{p} \cos \mathscr{y} + \frac{r}{a} \sin \mathscr{p} \cos \mathscr{y} \sin f] \tag{43}$$

$$\frac{di}{dt} = \frac{\alpha r}{h} \sin \mathscr{y} \cos(\omega + f) \tag{44}$$

The *structure* of the optimal control $\{\mathscr{p}^*, \mathscr{y}^*\}$ for this path-constraint-free problem can be obtained from the direct application of the PMP:

$$\frac{\partial \mathscr{H}}{\partial \mathscr{p}}\Big|_{\mathscr{p}=\mathscr{p}^*} = 0, \quad \frac{\partial \mathscr{H}}{\partial \mathscr{y}}\Big|_{\mathscr{y}=\mathscr{y}^*} = 0 \tag{45}$$

which simplifies to:[16]

$$
\begin{aligned}
\sin \not{p}^* &= \frac{\lambda_e \frac{r}{a} \sin f}{\sqrt{4[\lambda_a \frac{a^2 v^2}{\mu} + \lambda_e (e + \cos f)]^2 + \lambda_e^2 \frac{r^2}{a^2} \sin^2 f}} \\
\cos \not{p}^* &= -\frac{2[\lambda_a \frac{a^2 v^2}{\mu} + \lambda_e (e + \cos f)]}{\sqrt{4[\lambda_a \frac{a^2 v^2}{\mu} + \lambda_e (e + \cos f)]^2 + \lambda_e^2 \frac{r^2}{a^2} \sin^2 f}} \\
\sin \mathscr{y}^* &= -\frac{\lambda_i \frac{rv}{h} \cos(\omega + f)}{\sqrt{\lambda_i^2 (\frac{rv}{h})^2 \cos^2(\omega + f) + (\lambda_a \cos^2 \not{p}^* \frac{2a^2 v^2}{\mu})^2 + \lambda_e^2 [2(e + \cos f) \cos \not{p}^* + \frac{r}{a} \sin f \sin \not{p}^*]^2}} \\
\cos \mathscr{y}^* &= -\frac{\lambda_a \frac{2a^2 v^2}{\mu} \cos \not{p}^* + \lambda_e [2(e + \cos f) \cos \not{p}^* + \frac{r}{a} \sin f \sin \not{p}^*]}{\sqrt{\lambda_i^2 (\frac{rv}{h})^2 \cos^2(\omega + f) + (\lambda_a \cos^2 \not{p}^* \frac{2a^2 v^2}{\mu})^2 + \lambda_e^2 [2(e + \cos f) \cos \not{p}^* + \frac{r}{a} \sin f \sin \not{p}^*]^2}}
\end{aligned}
\tag{46}
$$

The (time-varying) thrust magnitude is not a control variable in this formulation, and is assumed to be evaluated from an engine model $|\boldsymbol{T}| = \frac{2\eta\mathscr{P}}{g_0 I_{sp}}$. It is clear from Eqs. (46) that the steering laws demonstrate (osculating) state-feedback structure, and are also functions of the current costates. In preparation for the application of a direct shooting method, the controls are discretized as:

$$
\lambda_\kappa \approx \sum_{\jmath=0}^{\mathscr{M}} (\tilde{\lambda}_\jmath)_\kappa \ell_\jmath(t), \;\; \kappa \in \{a,\ e,\ i\} \tag{47}
$$

where $\ell_\jmath(\cdot)$ is a node-polynomial basis and $\tilde{\boldsymbol{\lambda}} = [\tilde{\boldsymbol{\lambda}}_a\ \tilde{\boldsymbol{\lambda}}_e\ \tilde{\boldsymbol{\lambda}}_i]$ are the free parameters to be optimized. The angle-averaged dynamic optimization problem for the 7-state system of Eq. (40) can now be formulated in terms of the following system of differential-algebraic equations:

$$
\begin{aligned}
\min_{\tilde{\boldsymbol{\lambda}}} \quad & t_f \text{ or } - m_f \\
\text{s.t.} \quad & \frac{d\bar{\boldsymbol{x}}}{dt} = \frac{1}{P}\int_{-\pi}^{\pi} \mathscr{F}(\bar{\boldsymbol{x}}, \tilde{\boldsymbol{\lambda}}) \frac{dt}{dE} dE \\
& \cos f = \frac{\cos E - e}{1 - e\cos E} \\
& \sin f = \frac{\sin E \sqrt{(1 - e^2)}}{1 - e\cos E} \\
& \bar{\boldsymbol{x}}(0) = \boldsymbol{x}_0 \\
& [\bar{a}(t_f)\ \bar{e}(t_f)\ \bar{i}_{t_f}]^T = [a_f\ e_f\ i_f]^T
\end{aligned}
\tag{48}
$$

where $\frac{dt}{dE} = \sqrt{\frac{a^3}{\mu}}(1 - e\cos E)$ assuming pure Keplerian motion. Note that as in the case of Eq. (28), the evaluation of the orbital averaging integral in Eq. (48) can be performed using any $N-$point quadrature formula (reference [16] used a 20-point trapezoidal rule), while the numerical integration itself may be accomplished with a fixed-step integrator with step-size $\sim$ days. In Kleuver's original work, semi-major axis a is used as the independent variable instead of time. The following points are worth noting at this stage:

i) Similar to the SEPSPOT formulation, but unlike the MIPELEC technique, the Kleuver method is based on mean states obtained by averaging over an anomaly angle. Thus, the knowledge of the spacecraft's position in the orbit is lost, making the method unsuitable for solving rendezvous problems. However, SEPSPOT, MIPELEC and T_3D all solve the trajectory optimization problem via an indirect method as opposed to Kleuver's direct method, the latter being typically more robust to poor initial guesses compared to indirect methods. Furthermore, the Kleuver method offers the robustness of direct methods without sacrificing computational speed because the formulation utilizes averaging. It has also been adapted to a pure collocation-type approach in which both the states and the controls (costates) are discretized and the state dynamics are enforced at discrete node points through *implicit* integration instead of *explicit* integration of the direct shooting method.[26]

ii) On the other hand, SEPSPOT, MIPELEC and T_3D provide built-in support for path constraints (thrust direction), whereas the Kleuver method natively does not support path constraints. In fact, attempting to add path constraints will cause above-described formulation to lose some of its simplicity because in such a case, the derivation of the expressions for $\mathscr{p}^*$ and $\mathscr{q}^*$ must involve minimizing $\mathscr{H}$ among the set of *admissible* controls.[27]

iii) If SEP-generated thrust is the only (extra-two-body) osculating perturbation under consideration, then the lower and upper limits of the averaging integral in Eq. (48) must be replaced with E_{ex} and E_{in} respectively, where E_{ex} is the Earth shadow exit angle and E_{in} the Earth shadow entrance angle. In reference [16], these angles have been calculated from a cylindrical shadow model following the algorithm detailed in reference [28]. If other perturbations, such as gravitational harmonics, SRP, drag, third-body etc., are modeled as well, then averaging must be carried out over the entire orbit (SEP thrust being active over only a part of it).

iv) Referring to Eqs. (42-44), it is easy to see that the optimal controls would still be valid if additional perturbations are modeled, but only if their average influence does not affect rates of change of $\{a,\ e,\ i\}$. If they do, the optimal controls must be re-derived.

Test cases have been reported by Kleuver in reference [16] for minimum-time and minimum-fuel LEO-to-GEO and GTO-to-GEO transfers with initial acceleration level $\sim\ 10^{-4}\mathrm{m/s}^2$ and duration $\sim\ 200$ days, both with and without eclipsing. In his original work with control (costate) parameterization, Kleuver obtained excellent agreement with SEPSPOT using as few as 7 NLP variables; utilizing more NLP variables to parameterize the costates (over a denser mesh) resulted in marginal cost improvement only. The maximum run-time was reported to be about 33.seconds. The initial guess for the transfer time was directly obtained from the Edelbaum-Kèchichian method (Eq. (5) multiplied by 1.2 to account for eclipsing), while the costate guesses are derivable from the same formulation. In the Falck and Dankanich collocation approach to solving the averaged optimal control problem of Eq. (48), a minutely improved cost was obtained (198.6 days vs. Kleuver's 198.99 days), but the number of NLP variables rose to $\sim\ 400$, a still minuscule dimension compared with osculating-model-based optimization cases discussed in a later section.

Precision-integrated Indirect Methods

Several authors have addressed the problem of determining minimum-time and minimum-fuel spiral trajectories under osculating dynamical models using different variable sets.[29–34] Computing minimum-fuel multi-revolution trajectories is more challenging compared to minimum-time ones because of multiple thrust-coast switches and the possibility of the presence of singular arcs. These difficulties are further exacerbated if eclipsing and extra-two-body gravitational perturbations are modeled. It is perhaps due to this reason, coupled with the problem-dependent manner of estimating the initial costates, that the successful application of precision-integrated indirect methods to medium/high-fidelity, path-constrained, $100+$ revolution, multi-month optimal trajectories has been relatively rare. Kèchichian, in references [29, 31], presented the numerical solution of minimum-time LEO-to-GEO transfers with intermediate, constant thrust acceleration ($\sim\ 10^{-2}$ m/s^2) using equinoctial elements augmented with true longitude. Thrusting was assumed to be continuous throughout the transfer, the duration of which was a fraction of a day, and analytical expressions for the state and co-state dynamics were derived with known expressions for the J_2 through J_6 perturbing accelerations in a satellite-fixed frame. In a series of papers, Taheri *et al.* have used a variety of sophisticated numerical techniques to solve the fuel-optimal spiral problem using an indirect shooting method, but none of their work present results for, or discusses their framework's suitability to, many-revolution transfers with forced coasting under eclipsing.[32–35] In reference [36], Haberkorn and co-authors have adopted a homotopy-single-shooting approach to solve a very low-thrust ($\sim\ 10^{-5}$ m/s^2), year-long, fixed-time, minimum-fuel rendezvous problem with 754 revolutions and 1786 thrust-coast switches. In that work, the authors employ an MEE two-body-plus-constant-thrust model to maximize the final mass based on the usual mass dynamics, i.e., with:

$$\dot{m} = -\frac{T}{g_0 I_{sp}} \|\hat{\boldsymbol{u}}(t)\| \tag{49}$$

minimize:

$$\min_{\|\hat{\boldsymbol{u}}\| \leq 1} \quad \int_0^{t_f} \|\hat{\boldsymbol{u}}(t)\| \, dt \tag{50}$$

In order to tackle the TPBVP convergence issues arising from the bang-bang control structure of (see the derivation in reference [36] for details), two homotopic criteria:

$$C_1 \quad = \quad \int_0^{t_f} (\lambda \|\hat{\boldsymbol{u}}(t)\| + (1-\lambda) \|\hat{\boldsymbol{u}}(t)\|^2) dt \tag{51}$$

$$C_2 \quad = \quad \int_0^{t_f} \|\hat{\boldsymbol{u}}(t)\|^{2-\lambda}) dt, \;\; \lambda \in [0, 1] \tag{52}$$

are defined to smoothly transition the solution from the relatively tractable energy problem ($\lambda = 0$) to the mass problem by three separate continuation methods. The problem execution time was reported to be about 7 hours (circa 2004).

Osculating Dynamics-based Direct Methods

As alluded to earlier, the main attraction of the "direct" computational optimal control methods is that these methods are typically less sensitive to the initial guesses for the solve-for parameters in the transcribed numerical problem. In indirect methods, numerically solving the Euler-Lagrange system involves providing initial guesses of the costates/path-multipliers to the solver, the challenges of which are extensively documented in the literature.[21] Most direct methods, on the other hand, discretize the state-dynamics through polynomial approximations (piecewise/local or global) to the states and the controls along an estimated trajectory, and it is these polynomial coefficients that finally become the solve-for parameters for a non-linear programming problem (NLP). Another distinct advantage of osculating dynamics-based direct method is the ease with perturbations can be added to the right-hand-side of the GVEs, without requiring a re-derivation of the necessary conditions for optimality. For many-revolution orbit transfer missions powered by milli-g level (or less) engine accelerations, straight application of the direct transcription methods to the osculating dynamical equations results in a very large NLP, which has historically presented difficulties for NLP solver libraries in terms of speed, convergence and memory. This has been the main motivation behind the use of averaged state dynamics in the direct methods discussed earlier. The averaging operation over the spacecraft angular position effectively removes the dependence of the thrust direction on the rapidly-varying angular position, thus allowing the optimal control to be faithfully captured with fewer parameters, and hence reduce the NLP size. However, it is also the case that averaging removes information from the problem; for example, in order to synthesize real-time feedback controls from a set of pre-generated open-loop trajectories, access to the *osculating* controls would be necessary. Fortunately, the last two decades have witnessed great strides in the power of *sparse* numerical optimization libraries, such as SNOPT, IPOPT and more recently WORHP, that take advantage of the sparsity pattern of the NLP constraint Jacobian matrices generated from sophisticated direct transcription/collocation schemes.[37–39] This factor, coupled with the availability of Automatic Differentiation libraries such as ADOL-C, CasADi, CppAD etc. that easily interface with gradient-based optimizers, has led multiple researchers to apply direct transcription/collocation methods to the perturbed, osculating GVE description of many-revolution, low-thrust transfers. Three direct numerical methods applied to many-revolution, low-thrust transfers are briefly examined: Runge-Kutta Parallel Shooting, Collocation, and Differential Dynamic Programming (DDP), or its variant, Hybrid DDP (HDDP). R-K Parallel Shooting and collocation are both Direct Transcription methods, and although trajectory optimization using collocation was reported earlier in the literature, R-K Parallel Shooting seems to have been applied before collocation to the specific task of optimizing many-revolution geocentric trajectories,[8] and therefore described first.

Runge-Kutta Parallel Shooting: An early application of a direct transcription method to the optimization of many-revolution orbit-raising was by Scheel and Conway, who used a R-K parallel shooting method to discretize the problem, and used the *dense* NLP solver NZSOL to solve the resulting NLP.[8] The R-K parellel shooting method discretizes both the states and controls at a set of time nodes, and utilizes sequential *explicit* numerical integration in creating the non-linear dynamical "defect" constraints. The feature that makes R-K parallel shooting particularly attractive for solving multi-revolution spiral problems is the fact that this discretization scheme allows for the controls to be discretized over a denser mesh than the states, a feature not inherent to traditional collocation schemes[21] For example, using a 3-step version of this method, the number of control samples is approximately 3 times the number of state samples. This is particularly advantageous because it is known that all the equinoctial or elliptical states for many-revolution spirals vary slowly, excepting the angular position and the thrust direction, the latter oscillating with a period equal to the orbit period. Briefly, the R-K discretization of the standard optimal control problem:

$$\mathbb{P}\begin{cases}\min\limits_{\boldsymbol{u}\in\mathscr{U}} \int_0^{t_f} \mathbb{L}(\boldsymbol{x}(t),\boldsymbol{u}(t),t)dt \\ \text{s.t } \frac{d\boldsymbol{x}}{dt} = \boldsymbol{F}(\boldsymbol{x}(t),\boldsymbol{u}(t),t) \\ \boldsymbol{x}(0)=\boldsymbol{x}_0,\ \Psi(\boldsymbol{x}(t_f),t_f) \text{ specified}\end{cases} \tag{53}$$

leads to the following standard NLP:

$$\begin{aligned} \min_{\boldsymbol{\xi}} \quad & \mathscr{J}(\boldsymbol{\xi}) \\ \text{s.t.} \quad & \mathscr{C}(\boldsymbol{\xi}) \leq \mathbf{0} \\ & \boldsymbol{a} \leq \boldsymbol{\xi} \leq \boldsymbol{b} \end{aligned} \tag{54}$$

with cost function $\mathscr{J}(\cdot)$, functional constraints $\mathscr{C}(\cdot)$, decision variable vector $\boldsymbol{\xi}$, and decision variable box bounds $[\boldsymbol{a},\ \boldsymbol{b}]$. The decision variables comprise the discrete values of the states and controls at their respective mesh points, and other free parameters such as the final time, see reference [8] for details. Scheel and Conway used the equinoctial elements to solve a minimum-time, 100.6 revolution, equatorial-LEO-to-GEO orbit raising problem (i.e. not requiring a plane change) for a spacecraft equipped with an intermediate-thrust (0.0032 $\mathrm{m/s^2}$) and 2000 sec. I_{sp} engine. The J_2 and lunar gravity perturbations were modeled, but eclipsing was not considered, i.e., thrusting was assumed to be continuous. With time as the independent variable, 5 states ($a,\ h,\ k,\ l,\ m$) and one control (the thrust pitch angle), 60 state discretization nodes, and 6 R-K steps between each node pair for control discretization, the resulting NLP had 1027 variables and 300 constraints, moderately-sized NLP by today's standards. An initial guess was provided by propagating the GVEs with constant tangential thrust. In order to avoid the problem of sampling the trajectory too densely in the low-earth region and too sparsely at the GEO regime, Scheel and Conway have recommended placing the discretization nodes unevenly in time but evenly in true longitude.

Collocation: Collocation is perhaps the most widely-used of all transcription methods, and the literature on the theoretical and computational aspects of it is vast, and continuously growing. Therefore, an attempt will not be made here to provide a detailed account of this technique; for a somewhat recent tutorial on the topic and its relation with other computational optimal control methods, cf. reference [40]. The principal idea behind collocation is to represent the state and control histories as polynomials over a time grid, and solve for the coefficients of these polynomials by requiring that the DAEs describing the state evolution be enforced at some or all of these grid points. Similar to R-K Parallel Shooting, collocation methods also transcribe an optimal control problem of the type described by Eq. (53) to an NLP of the form Eq. (54), which is subsequently handed over to a numerical optimization library. An early application of local collocation algorithms (trapezoidal and Hermite-Simpson) to a minimum-propellant, low-thrust, many-revolution LEO-to-LEO orbit raising was documented by Betts in reference [9]. Betts modeled the problem using a set of modified equinoctial elements (MEE), a throttlelable CSI engine of acceleration level $\sim 10^{-4}\mathrm{m/s^2}$, and earth oblateness perturbations upto J_4. With 7 states (6 MEE + mass), 4 controls (thrust magnitude + three unit vectors), one path constraint (thrust unit vector magnitude), and a 13782 node trapezoidal discretization, the SQP solver solved an NLP with

152593 variables and 110974 constraints. The same problem was also discretized over 27742 nodes via a Hermite-Simpson collocation method, resulting in an NLP with 416123 variables and 249674 constraints. The maneuver was executed in 578 revolutions in approximately 40 days. Mesh refinement was used in both the cases, and time, instead of true longitude, was used as the independent variable. Although the NLP size in this case is about 3 orders of magnitude higher compared to the Scheel and Conway R-K parallel shooting discretization, note that a direct comparison might be unfair because the former work used fewer states and a thrust acceleration about 2 orders of magnitude higher. The program runtime was reported to be about 62 minutes on an SGI Origin 2000 server computer.

More recently, Betts[41] and Graham and Rao[10] have applied direct transcription to the osculating MEE GVEs to solve optimal LEO-to-GEO orbit raising problems with intermittent SEP thrusting. Both researches report using the conical projection eclipsing model instead of the cylindrical shadow model utilized in references [16, 23], and both use L as the independent variable instead of time, the latter to effectively facilitate a more favorable grid size and hence limit the NLP dimension. Partitioning the state space into slow variables, a fast variable, and mass, references [10, 41] cast the perturbation equations in MEE $\mathbb{Q} := [\boldsymbol{\varphi},\ L]^T$:

$$\frac{d\boldsymbol{\varphi}}{dt} = \boldsymbol{f}(\boldsymbol{\varphi}, L)\epsilon \tag{55}$$

$$\frac{dL}{dt} = g(\boldsymbol{\varphi}, L) + \boldsymbol{h}(\boldsymbol{\varphi}, L)\epsilon \tag{56}$$

$$\frac{dm}{dt} = -\frac{\|\boldsymbol{T}\|}{g_0 I_{sp}} \tag{57}$$

into the form:

$$\frac{d\boldsymbol{\varphi}}{dL} = \frac{\boldsymbol{f}(\boldsymbol{\varphi}, L)\epsilon}{g(\boldsymbol{\varphi}, L) + \boldsymbol{h}^T(\boldsymbol{\varphi}, L)\epsilon} \tag{58}$$

$$\frac{dt}{dL} = \frac{1}{g(\boldsymbol{\varphi}, L) + \boldsymbol{h}(\boldsymbol{\varphi}, L)\epsilon} \tag{59}$$

$$\frac{dm}{dL} = -\frac{\|\boldsymbol{T}\|}{g_0 I_{sp}(g(\boldsymbol{\varphi}, L) + \boldsymbol{h}(\boldsymbol{\varphi}, L)\epsilon)} \tag{60}$$

by changing the independent variable to L. Here, $g(, \cdot,)$ is a 2π-periodic function of L. Both papers decompose the trajectory into a sequence of thrust-coast segments, and optimize the thrust profile within each sunlit segment. However, in each case, significant effort has been devoted to obtain an initial guess for the number of phases and the true longitudes at which shadow entrance and exit occurs. Continuity constraints are imposed on the states $\{P(L),\ h(L),\ k(L),\ p(L),\ q(L),\ m(L)\}$ at the the shadow exit and entrance interfaces, and only natural motion is considered during the coast phases. In the Betts work, a minimum-propellant $28.5°$ inclination LEO to GEO transfer with an SEP engine with initial acceleration of $\sim 10^{-3}$ m/s^2 is reported to be accomplished in slightly more than 43 days, requiring about 248.5 revolutions and 363 thrust-coast phases. The final true longitude was free. With successive mesh refinement, the number of NLP (SQP) variables was 262240, and the number of NLP constraints was 175599. The problem size is thus comparable to reference [9], although it is interesting to note that the problem run-time (with a very good initial guess) is only about 17.5 minutes on a *desktop PC* (compared with a server computer pre-2000's), attesting to giant leaps in affordable computing power achieved in the 15 years separating the two works. In the Graham and Rao paper, minimum-time GTO-to-GEO test cases were solved using a Gauss Pseudospectral Method (GPM) with IPOPT as the NLP solver. The results, one with 89 revolutions in 65.9 days and another with 165 revolutions in 121.22 days, were compared with similar test cases reported by Kleuver via averaging, and agreement was seen to be close; a slightly lower cost was obtained in the former case and slightly higher cost in the other case. However, as noted previously, that the shadow models used in the two works were different. Furthermore, the NLP dimensions and run times were not reported in reference [10], although these may be assumed to be comparable to the Betts work, as opposed to $\sim 10s$ of parameters and ~ 5 seconds in Kleuver's.

Hybrid Differential Dynamic Programming (HDDP) : Aziz and co-authors have recently applied the HDDP algorithm, originally introduced by Lantoine and Russell [42], to the optimization of multi-revolution spirals.[11] HDDP extends the original Differential Dynamic Programming (DDP) algorithm[43] by incorporating constraint handling through an augmented Lagrangian method, multi-phase support, an efficient quadratic programming (QP) solver, and state propagation using state-transition matrices (STM). DDP starts with a nominal/guess trajectory, which is divided into a number of sequential "stages". The final objective is to compute the optimal controls valid over each stage. The following QP problem is numerically solved backward at each stage k to obtain controls for all the stages (more specifically, control updates to the guess trajectory):

$$\tilde{\mathscr{J}}_k^*(\boldsymbol{x}_k) = \min_{\boldsymbol{u}_k} \quad [\tilde{\mathbb{L}}_k(\boldsymbol{x}_k, \boldsymbol{u}_k) + \tilde{\mathscr{J}}_{k+1}^*(\boldsymbol{x}_{k+1})], \qquad k = 0, \ldots, N \tag{61}$$

where the tilde denotes local quadratic approximation of the quantities of interest about the nominal trajectory. At the conclusion of this backward sweep, the controls thus obtained are used to propagate (via STMs) the states in a forward sweep, and the process is repeated until user-defined convergence criteria are met. Therefore, if $\boldsymbol{u}_k \in \mathbb{R}^m$, then HDDP sequentially solves $N+1$ sub-problems each of size m rather than solve a single large NLP of size $m(N+1)$ as in a direct-shooting approach. In reference [11], the authors use HDDP to solve several minimum-fuel, fixed-revolution problems from the literature. Instead of using equinoctial, modified equinoctial or dynamical (see Sreesawet and Dutta [44]) slow variables, Cartesian position-velocity components in the gravitating-body-centered inertial frame are used as dynamical states. However, to regulate the integration step size and therefore render the problem dimensions manageable, the independent variable is changed from time to true, eccentric and mean anomalies through Sundman transform, which allows taking a fixed number of steps (100) per revolution without compromising accuracy. With the thrust magnitude and spherical angles as controls, the problem size with this formulation is then $300N_{rev}$. Following HDDP, Aziz and co-authors have solved a gravitational-perturbation-free 2000-revolution fuel-optimal transfer problem, and also a 500-revolution transfer problem with J_2 and lunar gravity perturbations, with STM computations and stage-to-stage propagation parallelized on a supercomputer. Path constraints, such as minimum altitude or bounded thrust direction, are not considered. The method, is however, computationally expensive, with main contribution from the STM computations; the 2000 revolution transfer reportedly took 48 hours before being terminated. The run-time and number of NLP iterations were also reported to increase with N_{rev}, albeit in an unpredictable manner. The algorithm is shown to faithfully capture the bang-bang control structure expected from minimum-fuel problems.

SUB-OPTIMAL TRAJECTORY DESIGN METHODS

SubTD methods consider the problem of trajectory design from a *targeting* perspective rather than an *optimization* one, i.e., the objective is to derive a thrust program with a view toward attaining a set of target orbit elements (e.g. ideal GEO), not necessarily accompanied by the minimization of an associated path integral. Such methods typically consider models of low to medium fidelity (e.g. most do not inherently account for eclipsing or path constraints), as the main motivation behind their use would be to generate, without excessive computational or development effort, ephemeris and control history for subsequent downstream utilization, e.g. as nominal trajectories for direct methods.

One SubTD approach is the so-called "thrust-blending" method in which the main idea is to blend or combine the thrust programs that individually cause the highest instantaneous rate of change of the orbital elements of interest. Mathematically, such a control law may be expressed as:

$$\boldsymbol{T}_\kappa^*(\boldsymbol{e}) = \arg\max_{\boldsymbol{T}_\kappa} \quad \frac{d\kappa}{dt}, \qquad \kappa \in \{a, e, i, \Omega, \omega\} \tag{62}$$

$$\boldsymbol{T}_{blend} = \sum_\kappa W_\kappa(t) \boldsymbol{T}_\kappa^*(\boldsymbol{e}) \tag{63}$$

where the (generally time-varying) weights W_κ can be tuned to leverage the physics of the problem. The rates $\frac{d\kappa}{dt}$ are obtained from the GVEs. It may be noted that the analytically derived control laws of the type of

Eq. (63) display a feedback structure. In [12], Kleuver introduced such a guidance algorithm based on a two-body-with-thrust set of GVEs, targeting only the elements a, e and i. A constant-power, constant I_{sp} engine with thrust-to-mass ratio $\sim 10^{-5}$ m/s^2 provides the thrust, but the thrust direction is analytically computed as a blend of directions that maximize, with respect to the pitch and yaw angles respectively, $\dot{a}$, $\dot{e}$ and $\dot{i}$. However, analytical expressions are not available for the weights, which makes the control law semi-heuristic. In order to effect a transfer from a circular, inclined $(28.5°)$ LEO to a standard GEO, Kleuver, modulated the weights over the maneuver duration so that during the initial phase (55% of the transfer duration), pitch steering is almost along-velocity, with the eccentricity weight decreasing linearly. This quickly raises the semi-major axis, but is accompanied by a slow rise in eccentricity, which is gradually reduced to nearly zero over the second half of the transfer with a quadratic weighting function. The inclination weighting function was chosen to be linearly decreasing with two different rates throughout the transfer so as to consistently reduce the inclination, more so during the second half when it is more economical to do so. Upon integrating the two-body-thrust dynamical equations with this control law and thrusting when in sunlight only, this algorithm found a transfer with duration $\sim 9\%$ higher than the corresponding minimum-time SEPSPOT solution, and ~ 8 % higher than an averaging-based direct method similar to the one presented in reference [16]. A similar blending technique, called Directional Adaptive Guidance (DAG), was adopted by Falck and co-authors [45] based on an earlier work by Ruggiero *et al.*[46] The DAG is shown to solve months-long (200+ days) GTO-to-GEO and LEO-to-GEO transfer with a high (constant) I_{sp} (3300 s) engine model identical to that used by Kleuver.[16] The control law of Eq. (63) is modified to the following form, which automatically causes the thrust to be turned off as the target elements κ_f are achieved:

$$\boldsymbol{T}_{blend} = \sum_{\kappa} W_\kappa \frac{\kappa_f - \kappa}{\kappa_f - \kappa_0} \boldsymbol{T}_a^*(\boldsymbol{e}),\ W_\kappa \text{ tunable constants} \tag{64}$$

Thrust is turned off during eclipses, although the specific eclipse model employed in that work is not mentioned. Additionally, the DAG method also accommodates a mechanism for coasting based on maneuver efficiency, i.e., if the vehicle angular position in the orbit is such that an element rate-of-change is much lower (or smaller than a threshold) compared with its maximum-attainable value, thrust is turned off to lower propellant consumption. This concept is similar to the Q-law[13] maneuver efficiency, which was historically introduced earlier and discussed next as part of Lyapunov-based transfer methods.

Several researchers have addressed the trajectory design problem from a Lyapunov control design framework.[13,47–50] The basic idea is to design asymptotically stable feedback or guidance laws for targeting specific orbit elements, resulting in orbit transfer[13,48] or rendezvous.[50] Chang *et al.* in reference [48] have used the dynamical variables angular momentum $\boldsymbol{h}$ and (scaled) eccentricity vector $\boldsymbol{A} = \mu\boldsymbol{e}$ instead of the more commonly encountered geometric ones, such as Keplerian, equinoctial or MEE, to define the orbit. The authors introduce a Lyapunov function $V(\boldsymbol{r}, \dot{\boldsymbol{r}}) = \|\boldsymbol{h} - \boldsymbol{h}_f\|^2 + \frac{1}{2}\|\boldsymbol{A} - \boldsymbol{A}_f\|^2$ to quantify the error between the current state and the target orbit, using which an asymptotically stabilizing control law (thrust program causing $\dot{V} \leq 0$) is derived. The method generates free-final-time transfer (not rendezvous) trajectories under a purely two-body-thrust model. Eclipsing is not considered in the original work, but the method was successfully exploited as a collocation seed method in reference [41] to produce an initial guess trajectory for the sunlit passes of an SEP spacecraft.

The Q-law is a Lyapunov feedback control law originally developed in 2003 and subsequently refined in 2005 by Petropoulos "with the aim of improving approximations to and initial guesses for, propellant-optimal, low-thrust orbit transfers which involve specified changes in all orbit elements except true anomaly".[13] Intermittent thrusting is incorporated, although coast periods are motivated by propellant saving rather than eclipsing. Furthermore, the original analysis is based on a two-body-plus-(constant) thrust model described in terms of the classical Keplerian elements, although an MEE formulation has been reported in [51]. At the core of the Petropoulos Q-law is the candidate Lyapunov function or proximity quotient that quantifies the proximity between the current and target orbit elements:

$$Q = (1 + W_P P) \sum_{\kappa} W_\kappa S_\kappa \left[\frac{d(\kappa, \kappa_f)}{\dot{\kappa}_{max}}\right]^2 \tag{65}$$

where P is a minimum-periapsis penalty function, W_P is the periapsis-penalty weight, W_κ are non-negative scalar weights, $d(,\cdot,)$ is a distance function, $\dot{\kappa}_{max}$ is maximum rate of change of the element κ over the thrust pointing spherical angles and the true anomaly (as modeled by the respective GVEs), and S_κ are scalars, unity except S_a, that help convergence to the final semi-major axis. Refer to [13] for more details on synthesizing an analytical expression for Q. Note that the quantity within the square brackets is a "time-to-go"-like quantity, so that Q embodies a blended, minimum time to attain the target. Following the standard Lyapunov control design paradigm, the thrust program at any point during the transfer can be obtained by minimizing $\dot{Q}$ and can be obtained analytically:[6,51]

$$[\not{p}^* \; \not{y}^*] = \underset{[\not{p} \; \not{y}]}{\arg\min} \quad \dot{Q}, \tag{66}$$

In order to ensure thrusting over the "most effective" (in terms the reduction of Q) range of angular positions of the orbit, Q-law incorporates the concept of thrust effectivity or efficiency by defining absolute and relative efficiencies:

$$\eta_a = \frac{\dot{Q}_n}{\dot{Q}_{nn}}, \; \eta_r = \frac{\dot{Q}_n - \dot{Q}_{nx}}{\dot{Q}_{nn} - \dot{Q}_{nx}} \tag{67}$$

where

$$\dot{Q}_n = \underset{[\not{p} \; \not{y}]}{\arg\min} \quad \dot{Q}, \quad \dot{Q}_{nn} = \underset{f}{\arg\min} \quad \dot{Q}, \quad \dot{Q}_{nx} = \underset{f}{\arg\max} \quad \dot{Q}, \tag{68}$$

The quantities $\dot{Q}_{nn}$ and $\dot{Q}_{nx}$ are determined numerically. Propellant saving can be achieved if thrusting is allowed only above user-specified values of η_a or η_r, although at the expense of increased the maneuver duration. Petropoulos provides examples of such trade-off scenarios through multi-month, several-hundred-revolution trajectories. The $Q-$law has been used in obtaining sub-optimal spiral trajectories in numerous researches including in the interplanetary trajectory design tool Mystic,[52] and most recently by Jackson and co-authors in a part-$Q-$law/part optimal transfer.[6] In [6], a Q-law transfer is made up to a certain fraction $(25\%,\ 50\%,\ 75\%)$ of approximately a 4-month-long GTO-to-GEO journey, beyond which a collocation-based direct method takes over. This results in an NLP smaller in dimension compared with one that would have been obtained from optimizing the entire transfer.

Recently, Dutta and his co-workers have adopted a sub-optimal orbit raising algorithm to solve several LEO-to-GEO and GTO-to-GEO planar and 3-dimensional orbit transfer test cases with eclipsing, involving up to thousands of revolutions and ~ 1 year+ transfer time.[44,53] The slow variables of the problem are the angular momentum and eccentricity vectors as in reference [48]. Thrust pitch and yaw angles are programmed in the sunlit portion of each orbital revolution so as to minimize the deviation of the per-revolution terminal states from the GEO target states, i.e., the objective is to minimize:

$$\mathscr{J} = w_i(h_{GEO} - h_{2\pi})^2 + w_2 \left\| \boldsymbol{e} \right\|_{2\pi}^2 + w_3 \left\| \boldsymbol{h}_{XY} \right\|_{2\pi}^2 \tag{69}$$

by optimizing the thrust pointing angle history within the sunlit arc of each revolution. Here, $\boldsymbol{h}_{XY}$ projection of $\boldsymbol{h}$ on the equatorial inertial plane. The sunlit arc within each revolution is subdivided into sub-arcs, within each of which a piecewise linear control profile is assumed; the optimization decision variables are then the thrust angle values at the delimiting nodes. The algorithm is terminated once the state targets are met to within a specified tolerance. The computation burden is small, ~ 10's of seconds, although the algorithm has been reported to suffer from an "optimality gap" compared to the results published for identical problems solved in reference [16]. However, a direct comparison may not be justified because both the problem formulations and numerical approaches of the two sources differ.

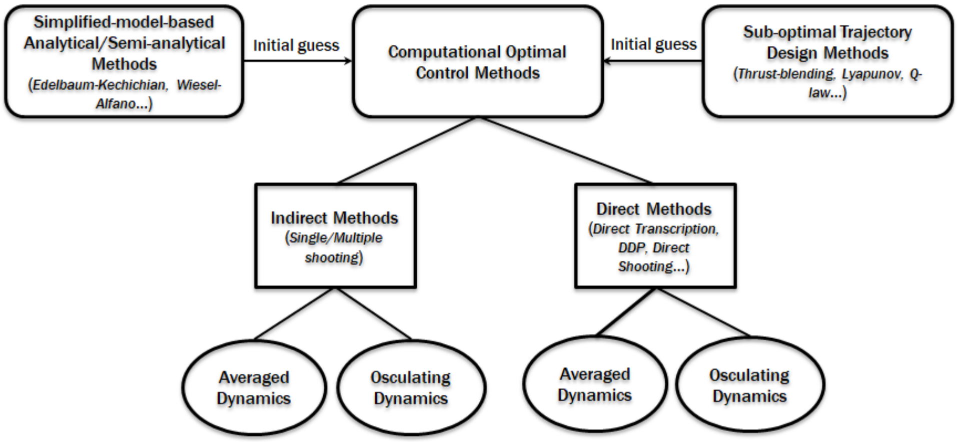

Figure 1. A Classification of Many-revolution Trajectory Generation Methods

CONCLUSION

A classification of the existing methods for the design and optimization of multi-revolution, low-thrust, spiral trajectories around a gravitating central body was studied, a summary of which appears in Fig.(1). The increasing commercial adoption of Solar Electric Propulsion systems, especially in GEO-bound communication satellites, combined with a healthy projected growth in the small satellites market, is expected to motivate continued effort into efficient trajectory design software supporting such applications. Over the past 6 decades, numerous methods have been proposed to address this problem; some of the early methods, such as the Edelbaum analytical method, adopted reduced-order models and simplifying assumptions (e.g. quasi-circular nature of the transfer orbit) with a view to developing quick, convenient analytical formulae, while more recent researches were seen to have produced purely numerical algorithms leveraging the power of supercomputers, e.g. HDDP. The available methods were categorized into three broad types, namely, Simplified Model-based Analytical/Semi-analytical Methods, Computational Optimal Control Methods, and Sub-optimal Trajectory Design Methods. The first and the third types are *trajectory design* methods, rather than *optimization methods*, and in a software implementation, would be ideal as seed methods for the more computationally involved (and potentially higher fidelity) Optimal Control methods. Among the latter types, those that are based on averaged dynamics appear to be the most promising because they retain the advantages of direct methods, such as extensibility to higher fidelity force models without the need to re-derive the optimality necessary conditions, with the added benefit of tractable NLP size.

NOTATION

$[a\ e\ i\ \omega\ \Omega\ f\ m]$	[semi-major axis eccentricity inclination argument of periapsis RAAN true anomaly spacecraft mass]
$[h\ k\ p\ q]$	$[e\sin(\omega+\Omega)\ e\cos(\omega+\Omega)\ \tan(\frac{i}{2})\sin\Omega\ \tan(\frac{i}{2})\cos\Omega]$
$\boldsymbol{e}$	$[a\ e\ i\ \omega\ \Omega\ f]$
F, E, l, L	Eccentric Longitude, Eccentric anomaly, Mean Longitude, True longitude
$\boldsymbol{q}$	$[P\ h\ k\ p\ q]$
P	Semi-latus rectum
$\boldsymbol{\alpha}(t), \alpha(t)$	Engine thrust acceleration vector, magnitude at time t
$\hat{\boldsymbol{u}}(t), \dot{\boldsymbol{r}}$	Thrust direction at time t, spacecraft inertial velocity
μ, n	Earth gravitational parameter, spacecraft mean motion
$\mathscr{p}, \mathscr{y}$	Thrust pitch, yaw angle
r, h	Satellite radial distance from the attracting center and specific angular momentum
$I_{sp}, \mathscr{P}, \eta$	Propulsion system specific impulse, generated power, power efficiency

REFERENCES

[1] M. Martinez-Sanchez and J. E. Pollard, "Spacecraft Electric Propulsion - An Overview," *Journal of Propulsion and Power*, Vol. 14, No. 5, 1998, pp. 688–699.

[2] D. Lev *et al.*, "The Technological and Commercial Expansion of Electric Propulsion in the Past 24 Years," 35^{th} *International Electric Propulsion Conference, Georgia Institute of Technology*, Atlanta, Georgia, 2017.

[3] T. Edelbaum, "Propulsion Requirements for Controllable Satellites," *ARS Journal*, Vol. 31, 1961, pp. 1079–1089.

[4] D. Lawden, *Optimal Trajectories for Space Navigation*. Butterworths Mathematical Texts, 1963.

[5] Z. P. Olikara, "Framework for Optimizing Many-revolution Low-thrust Transfers," *2018 AAS/AIAA Astrodynamics Specialist Conference*, Snowbird, UT, 2018.

[6] J. L. Shannon, M. T. Ozimek, J. A. Atchison, and C. M. Hartzell, "Q-Law Aided Direct Trajectory Optimization For The High-Fidelity, Many-Revolution Low-Thrust Orbit Transfer Problem," *2019 AAS/AIAA Space Flight Mechanics Meeting*, Maui, HI, 2019.

[7] L. L. Sackett, H. L. Malchow, and T. N. Edelbaum, "Solar Electric Geocentric Transfer With Attitude Constraints: Analysis," Tech. Rep. NASA CR–134927, The Charles Stark Draper Laboratory, Inc., Cambridge, Massachusetts, 1975.

[8] W. A. Scheel and B. A. Conway, "Optimization of Very-Low-Thrust, Many-Revolution Spacecraft Trajectories," *Journal of Guidance, Control and Dynamics*, Vol. 17, No. 6, 1994, pp. 1185–1192.

[9] J. T. Betts, "Very Low-Thrust Trajectory Optimization Using a Direct SQP Method," *Journal of Computational and Applied Mathematics*, Vol. 120, 2000, pp. 27–40.

[10] K. F. Graham and A. V. Rao, "Minimum-Time Trajectory Optimization of Multiple Revolution Low-Thrust Earth-Orbit Transfers," *Journal of Spacecraft and Rockets*, Vol. 53, No. 2, 2016, pp. 289–303.

[11] J. D. Aziz, J. S. Parker, D. J. Scheeres, and J. A. Englander, "Low-Thrust Many-Revolution Trajectory Optimization via Differential Dynamic Programming and a Sundman Transformation," *The Journal of the Astronautical Sciences*, Vol. 65, No. 2, 2018, pp. 205—-228.

[12] C. A. Kluever, "Simple Guidance Scheme for Low-Thrust Orbit Transfers," *Journal of Guidance, Control and Dynamics*, Vol. 21, No. 6, 1998, pp. 1015–1017.

[13] A. E. Petropoulos, "Refinements to the Q-law for Low-Thrust Orbit Transfers," 15^{th} *AAS/AIAA Space Flight Mechanics Conference*, Copper Mountain, CO, 2005.

[14] J. A. Kèchichian, "Reformulation of Edelbaum's Low-Thrust Transfer Problem Using Optimal Control Theory," *Journal of Guidance, Control and Dynamics*, Vol. 20, No. 5, 1997, pp. 988–994.

[15] C. A. Kluever, "Using Edelbaum's Method to Compute Low-Thrust Transfers with Earth-Shadow Eclipses," *Journal of Guidance, Control, and Dynamics*, Vol. 34, No. 1, 2011, pp. 300–303.

[16] C. A. Kluever, "Low-Thrust Trajectory Optimization Using Orbital Averaging and Control Parameterization," *Spacecraft Trajectory Optimization* (B. Conway, ed.), ch. 5, pp. 112–138, Cambridge University Press, 2010.

[17] J. A. Kèchichian, "Optimal Low-Thrust Transfer in General Circular Orbit Using Analytic Averaging of the System Dynamics," *The Journal of the Astronautical Sciences*, Vol. 57, No. 1–2, 2009, pp. 369–392.

[18] G. Colasurdo and L. Casalino, "Optimal Low-Thrust Maneuvers in Presence of Earth Shadow," *AIAA/AAS Astrodynamics Specialist Conference, AIAA Paper 2004-5087*, Providence, RI, 2004.

[19] J. A. Kèchichian, "Low-Thrust Eccentricity-Constrained Orbit Raising," *Journal of Spacecraft and Rockets*, Vol. 35, No. 3, 1998, pp. 327–335.

[20] W. E. Wiesel and S. Alfano, "Optimal Many-Revolution Orbit Transfer," *Journal of Guidance, Control, and Dynamics*, Vol. 8, No. 1, 1985, pp. 155–157.

[21] J. T. Betts, *Practical Methods for Optimal Control and Estimation Using Nonlinear Programming*. Society for Industrial and Applied Mathematics, 2009.

[22] S. Geffroy, R. Epenoy, and J. Noailles, "Averaging Techniques In Optimal Control For Orbital Low-Thrust Transfers And Rendezvous Computation," $11^{th.}$ *International Astrodynamics Symposium*, Gifu, Japan, 1996.

[23] S. Geffroy and R. Epenoy, "Optimal Low-Thrust Transfers With Constraints–Generalization Of Averaging Techniques," *Acta Astronautica*, Vol. 41, No. 3, 1997, pp. 133–149.

[24] F. Chaplais, "Averaging and Deterministic Optimal Control," *SIAM Journal on Control and Optimization*, Vol. 25, No. 3, 1986, pp. 767–780.

[25] T. Dargent, "Averaging Technique In T_3D: An Integrated Tool For Continuous Thrust Optimal Control In Orbit Transfers," $24^{th.}$ *AAS/AIAA Space Flight Mechanics Meeting*, Santa Fe, NM, 2014.

[26] R. D. Falck and J. W. Dankanich, "Optimization of Low-Thrust Spiral Trajectories by Collocation," *AIAA/AAS Astrodynamics Specialist Conference*, Minneapolis, MN, 2012.

[27] A. E. Bryson and Y.-C. Ho, *Applied Optimal Control: Optimization, Estimation and Control*, ch. 3. Taylor and Francis, 1975.

[28] B. Neta and D. A. Vallado, "On Satellite Umbra/Penumbra Entry and Exit Positions," *Journal of the Astronautical Sciences*, Vol. 46, No. 1, 1997, pp. 91–103.

[29] J. A. Kèchichian, "Trajectory Optimization Using Nonsingular Orbital Elements and True Longitude," *Journal of Guidance Control Dynamics*, Vol. 20, Sept. 1997, pp. 1003–1009.

[30] T. Haberkorn, P. Martinon, and J. Gergaud, "Low Thrust Minimum-Fuel Orbital Transfer: A Homotopic Approach," *Journal of Guidance Control Dynamics*, Vol. 27, Nov. 2004, pp. 1046–1060.

[31] J. A. Kèchichian, "Inclusion of Higher Order Harmonics in the Modeling of Optimal Low-Thrust Orbit Transfer," *The Journal of the Astronautical Sciences*, Vol. 56, 2008, pp. 41–70.

[32] E. Taheri, I. Kolmanovsky, and E. Atkins, "Enhanced Smoothing Technique for Indirect Optimization of Minimum-Fuel Low-Thrust Trajectories," *Journal of Guidance Control Dynamics*, Vol. 39, No. 11, 2016, pp. 2500–2511.

[33] V. Arya, E. Taheri, and J. L. Junkins, "Efficient Computation Of Optimal Low Thrust Gravity Perturbed Orbit Transfers," *2019 AAS/AIAA Space Flight Mechanics Meeting*, Maui, HI, 2019.

[34] J. L. Junkins and E. Taheri, "State Vector Representations for Low-Thrust Trajectory Optimization," *2018 AAS/AIAA Astrodynamics Specialist Conference*, Snowbird, UT, 2018.

[35] R. Woollands, E. Taheri, and J. L. Junkins, "Efficient Computation Of Optimal Low Thrust Gravity Perturbed Orbit Transfers," *2019 AAS/AIAA Space Flight Mechanics Meeting*, Maui, HI, 2019.

[36] T. Haberkorn, P. Martinon, and J. Gergaud, "Low Thrust Minimum-Fuel Orbital Transfer: A Homotopic Approach," *Journal of Guidance Control Dynamics*, Vol. 27, No. 6, 2004, pp. 1046–1060.

[37] P. E. Gill, W. Murray, and M. A. Saunders, "SNOPT: An SQP Algorithm for Large-Scale Constrained Optimization," *SIAM Rev.*, Vol. 47, Jan. 2005, pp. 99–131.

[38] A. Wächter, "Short Tutorial: Getting Started With Ipopt in 90 Minutes,"

[39] C. Bueskens and D. Wassel, "The ESA NLP Solver WORHP," *Modeling and Optimization in Space Engineering* (G. Fasano and J. D. Pintér, eds.), Vol. 73, pp. 85–110, Springer New York, 2013.

[40] B. A. Conway, "A Survey of Methods Available for the Numerical Optimization of Continuous Dynamic Systems," *Journal of Optimization Theory and Applications*, Vol. 152, No. 2, 2012, pp. 271–306.

[41] J. T. Betts, "Optimal Low–Thrust Orbit Transfers With Eclipsing," *Journal of Optimization Theory and Applications*, Vol. 36, No. 2, 2015, pp. 218–240.

[42] G. Lantoine and R. P. Russell, "A Hybrid Differential Dynamic Programming Algorithm for Constrained Optimal Control Problems. Part 1: Theory," *Journal of Optimization Theory and Applications*, Vol. 154, 2012, pp. 382–417.

[43] D. Jacobson and D. Q. Mayne, *Differential Dynamic Programming*. American Elsevier Publishing, 1970.

[44] S. Sreesawet and A. Dutta, "A Novel Methodology For Fast And Robust Computation Of Low-Thrust Orbit-Raising Trajectories," $27^{th.}$ *AAS/AIAA Space Flight Mechanics*, San Antonio, TX, 2017.

[45] R. D. Falck, W. K. Sjauw, and D. A. Smith, "Comparison of Low-Thrust Control Laws for Applications in Planetocentric Space," 50^{th} *AIAA/ASME/SAE/ASEE Joint Propulsion Conference*, Cleveland, OH, 2014.

[46] A. Ruggiero, P. Pergola, S. Marcuccio, and M. Andrenucci, "Low-Thrust Maneuvers for the Efficient Correction of Orbital Elements," 32^{nd} *International Electric Propulsion Conference*, Wiesbaden, Germany, 2011.

[47] M. R. Ilgen, "Low Thrust OTV Guidance Using Lyapunov Optimal Feedback Control Techniques," *1993 Astrodynamics specialist conference*, Victoria, Canada, 1993.

[48] D. E. Chang, D. F. Chichka, and J. E. Marsden, "Lyapunov-Based Transfer Between Elliptic Keplerian Orbits," *Discrete and Continuous Dynamical Systems - Series B*, Vol. 2, No. 1, 2002, pp. 57–67.

[49] S. Lee, P. von Ailmen, W. Fink, A. E. Petropoulos, and R. J. Terrile, "Design And Optimization Of Low-Thrust Orbit Transfers," *2005 IEEE Aerospace Conference*, Big Sky, MT, 2005.

[50] S. Hernandez and M. R. Akella, "Lyapunov-Based Guidance for Orbit Transfers and Rendezvous in Levi-Civita Coordinates," *Journal of Guidance, Control, and Dynamics*, Vol. 37, No. 4, 2014, pp. 1170–1181.

[51] Gàbor I. Varga and Josè M. Sànchez Pèrez, "Many-Revolution Low-Thrust Orbit Transfer Computation Using Equinoctial Q-Law Including J_2 and Eclipse Effects," *2015 AAS/AIAA Astrodynamics Specialist Conference*, Vail, CO, 2016.

[52] G. J. Whiffen, "Mystic: Implementation of the Static Dynamic Optimal Control Algorithm for High-Fidelity, Low-Thrust Trajectory Design," *2006 AIAA/AAS Astrodynamics Specialists Conference*, Keystone, Colorado, 2006.

[53] A. Dutta and L. Arora, "Objective Function Weight Selection for Sequential Low-Thrust Orbit-Raising Optimization Problem," *2019 AAS/AIAA Space Flight Mechanics Meeting*, Maui, HI, 2019.