Study of analytical solutions for low-thrust trajectories

Juan Luis Cano Rodríguez
Escuela Técnica Superior de Ingenieros Aeronáuticos
Universidad Politécnica de Madrid

March 1, 2017

Acknowledgements

The history of scientific and technological progress is full of examples of visionaries whose contributions were rejected by the scholar community or mocked by the press of the time for being too fanciful or belonging to the realm of fantasy and deemed impossible. While this fact might come as non surprising for many, in writing this thesis I have found particular examples of astonishing significance which have made me feel both an enormous embarrassment and an incommensurable admiration, being the 1920 editorial of The New York Times dismissing Robert H. Goddard ideas for space propulsion (and its correction one day after the Eagle successfully landed on the moon, 49 years later) a somewhat recent, paradigmatic example. I would like to express my gratitude to all those determined, brilliant individuals for being a source of inspiration for the generations that followed them.

Open source software is no exception to this rule, with the additional handicap that most software that we take for granted today has been (or continues to be) developed by volunteers against all odds and without specific funding in their free time, when their children are sleeping or as pure procrastination during their full time jobs. This Master thesis would not be possible without the tremendous, often unpaid and unrecognized effort of all those volunteers, to whom I also express my greatest thankfulness.

In writing this thesis I have lost countless hours looking for old but valuable scientific papers and books that were impossible to find (not even available for purchase) despite all my best efforts, which, to all effects and in my humble opinion, can be considered as disastrous as the destruction of the Library of Alexandria. The academic publishing industry naturally holds accountable for this regretful loss of human knowledge. I would like to thank from the bottom of my heart all the people that have left behind this fallacious system that hinders human progress and that have openly shared all their knowledge, for free. In return of all that wisdom and as my very modest contribution, this Master thesis has been written in English to serve a bigger audience and published in the open.

Moving now to more personal acknowledgements, I would like to thank my advisor, professor Manuel Sanjurjo, for all the tremendous patience and support he has shown since this thesis started. I am sure mentoring a student that occasionally finds some time on weekends to do real work is not an easy task. Many thanks to David Morante as well, and also to my *de iure* supervisor, professor Jesús Peláez, for clearing all the administrative obstacles. I would like to mention Helge Eichhorn and Frazer McLean as well, who helped me shape the original `poliastro` paper.

Going back in time a little bit more, I cannot omit Mario Merino from this list. During my freshman year he showed true dedication for his students, helping those in need, encouraging collaboration among them and motivating us to go beyond what was expected for us. His vision for a better University has inspired me ever since, and his always kind advice was determinant for me to apply for an Erasmus scholarship, which still is the best decision I have ever made. It is possible to be a genius and still have time for the rest of us, and Mario taught us so.

Internal motivation is a necessary condition for successfully accomplishing any task, but external motivation is, no doubt, an invaluable help. I would like to thank my friends and coworkers for encouraging me in the final stages of writing this thesis, especially Lourdes, who sparked the determination I still needed despite everything was in place, and my good friend Álex, for his always accurate insights and for making this Python journey a worthwhile, entertaining experience. I cannot finish these acknowledgements without thanking my parents for making sure I could have all the opportunities I wanted. This is as sincerest as it can be, for I know there is no such thing as a "self-made person" and I have enjoyed privileges that others can only dream of, thanks to a great extent to my family. *Papá, mamá, hermano, muchas gracias por todo*.

Lastly, I would also like to thank professor Michèle R. Lavagna in a very special way for giving me freedom to start this project four years ago in Milan, Italy, and for accepting my negative to take the final oral exam, effectively failing my favourite subject, ever. *Grazie mille per tutto*.

Contents

"A planet is the cradle of mind, but one cannot live in a cradle forever."

KONSTANTIN E. TSIOLKOVSKY (1857 - 1935)

Chapter 1

Introduction

1.1 The conquest of space

Since in 1609 German mathematician and astronomer Johannes Kepler published his book *Astronomia nova*, containing the most famous of all transcendental equations, the motion of the celestial bodies has attracted the attention of the greatest minds in human history, even sparkling entire new fields in mathematics [Battin, 1999]. It is easy to imagine that if even Kepler's equation, the one that captures the essence of the two-body problem in its most restricted form, already has this mathematical intricacy, any further development will carry away similar or greater complexity.

$$M = E - e \sin E$$

Figure 1.1: The Kepler equation.

Almost three centuries later, in 1903, Russian rocket scientist Konstantin E. Tsiolkovsky first explained in his article *Exploration of Outer Space by Means of Rocket Devices* precise conditions for artificial objects to reach the orbit of the Earth, making a huge leap from the mere observation of the celestial bodies and the science fiction stories that had inspired him to the real possibility of going to space. Quoting [Siddiqi, 2000]:

> In his most revolutionary idea, he proposed that humans could hope to fly to very high altitudes and ultimately into outer space only by using liquid-propellant rockets. One of his most important conclusions was that a rocket would be capable of carrying up a cargo of any size, and develop any speed desired, as long as the rocket was sufficiently large and the ratio of the mass of the propellant to the mass of the entire rocket was large enough – a relationship that is known as the Tsiolkovsky Equation.

$$\Delta v = v_e \ln \frac{m_0}{m_f}$$

Figure 1.2: The Tsiolkovsky equation.

If we define the term Astrodynamics as the branch of Mechanics that studies practical problems concerning the motion of rockets and other artificial objects through space, Tsiolkovsky's contribution might well be considered its starting point, and many others ensued before they could be tested in practice during the second half of the 20th century. In 1919 Yuri V. Kondratyuk conceived the gravitational slingshot or flyby to accelerate a spacecraft through interplanetary flight and suggested a mission profile for a Lunar landing [Siddiqi, 2000], in 1925 Walter Hohmann conjectured that the minimum-fuel transfer between two coplanar circular orbits consists of two tangent impulses along the line of apses (although this result was not proved until almost forty years later in [Lawden, 1963]) and in 1926 Hermann J. Oberth observed that the velocity gain of an impulsive maneuver is higher when the kinetic energy is maximum (nowadays known as the Oberth effect). The severe limitations in weight and available energy for such kind of travels were already apparent for these pioneers, who were, in some way, anticipating the need to optimize on board fuel consumption.

At the same time, important practical advances were being made in the field of rocketry. In 1920 Robert H. Goddard published in his article *A Method of Reaching Extreme Altitudes* several ideas and experimental results regarding the study of rockets, in particular that the exhaust speed of their combustion gases could be greatly increased by using convergent-divergent or De Laval nozzles, and in 1926 he launched the world's first liquid-fueled rocket, which reached an altitude of 12 meters, lasted 2 seconds and averaged about 100 kilometers per hour. After this remarkable event, amateur rocket societies started to form in America and Europe and bigger and more sophisticated rockets were built and launched in the following years.

To this day, chemical propulsion (whether using liquid fuel as pioneered by Goddard and later used by the Saturn V or solid fuel as already used by the Chinese in the 13th century) continues to be the only way to escape the gravitational well of the Earth. What is not so well known is that many of the people involved in the early development of astrodynamics and astronautics in the 20th century shared a common interest in an alternative method: electric propulsion. Goddard had already reflected on the use of "electrons moving with the velocity of light" as a method of propulsion as early as 1906, and Tsiolkovski wrote in 1911 about the use of electricity "to produce a large velocity for the particles ejected from a rocket device" [Choueiri, 2004]. Oberth even envisioned electric propulsion as a real possibility for attitude control in his 1929 book *Ways to Spaceflight* and already predicted its mass saving capabilities.

Despite all this enthusiasm, however, there were far too many obstacles at the time to make it a reality: first of all, the understanding of atomic physics was still in its infancy, with the nature of the electrons still unclear and the discovery of the proton not happening until 1920, hindering rigorous studies. Besides, electric propulsion was definitely not useful to attain orbital velocity, and on the other hand it requires a much more complex mathematical analysis (we will expand on these aspects in 1.2). For the following fifteen years after the publication of Oberth's book all the focus shifted away from electrical propulsion, until the tremendous advances regarding the production of intense ion currents during the forties brought back the debate about its feasibility [Choueiri, 2004].

Electric spacecraft propulsion was not the only idea that appeared those years that fell dormant because it was well ahead of its time. In 1925, twenty five years after Pyotr N. Lebedev demonstrated the pressure of light, Friedrich A. Zander published a technical paper titled *Problems of flight by jet propulsion: interplanetary flights* where he discussed several challenges associated with space travel, including re-entry into the Earth atmosphere, and first analyzed with great detail the use of radiation pressure as a mean of propulsion, an idea suggested years earlier by Tsiolkovski. Zander proposed the use of extremely thin aluminum mirrors and computed the required surface and admissible stresses, laying the first bricks of beam-powered propulsion. The first formal technology and design effort for a solar sail did not begin until 1976 and to this day there are no flight proven solutions.

In 1950 George F. Forbes presented the element that was missing for a complete mathematical analysis, the study of low-thrust trajectories, by showing for the first time that in some cases they were more efficient than the high-thrust counterparts in his Masters thesis *The trajectory of a powered rocket in space*. This, and the contribution of Lyman S. Spitzer, who in his 1952 paper *Interplanetary Travel Between Satellite Orbits* suggested that for an ion propulsion system to be technically useful, an acceleration of three centimeters per square second would be enough ("sufficient to attain a velocity of 15 km/sec in two months" [Spitzer Jr, 1952]), provided the building blocks for the electrical propulsion to have the attention it deserved.

Finally, the decade of the sixties served as the definitive *tour de force* of chemical propulsion, which powered the rockets used during the space race that started in 1957 with the launch of Sputnik I by the Soviet Union and that culminated with in 1969 with the first manned landing on the Moon by the United States of America. In the meanwhile, in 1964 the first mission to demonstrate the use in orbit of an ion thruster, SERT-1, was launched, whereas a milestone comparable in importance arrived as late as 1998 with the mission Deep Space 1. With these achievements, the early ideas of the visionaries of the beginning of the century were finally proven real and humanity was in the position of using these technologies to "leave the cradle".

1.2 Chemical versus electric propulsion

After the historical perspective given in 1.1 we dedicate the following sections to introduce some important theoretical concepts, with the objective of justifying the importance of low-thrust orbit analysis within the context of orbit optimization and the pursuit of analytical solutions in this area. We start with the distinction of chemical versus electric propulsion and the main similarities and differences between them – this does not intend to be an in-depth description of the subsystems and technologies that are involved, but rather an overview of their most essential aspects and how do they affect their performance and use cases (for such in-depth analysis we direct the readers to well known references in the subject such as [Sutton and Biblarz, 2016]). In A a thorough list of spacecraft propulsion methods has been included for reference.

Except for very small satellites, nearly all space missions require on-board propulsion systems, which have a major impact on spacecraft mass [Curran et al., 1993]. Some sources estimate that the current cost of sending one kilo of payload into low Earth orbit lies around 20 000 €, and therefore any effort to decrease the launch mass of the spacecraft leads to significant savings [Choueiri, 2009].

We can define propulsion in a broad sense as the act of changing the motion of a body [Sutton and Biblarz, 2016]. Spacecraft propulsion methods can be classified according to the energy source that is used to produce thrust, which can be chemical, nuclear or electric.

Chemical propulsion uses the chemical potential stored in the propellants, usually of reducing (fuel) and oxidizing nature, that is released in a high-pressure combustion reaction between the two. The reaction produces a flow of hot gases (between 2500 and 4100 °C) that can be accelerated in a convergent-divergent (or De Laval) nozzle to achieve supersonic velocities (1800 to 4300 m/s). In this case, therefore, the power source and the propellants are the same elements. Chemical propulsion systems can be themselves subdivided according to the state of the propellants: *solid propellant rocket motors* use a solid material that contains all the chemical elements that are needed for complete burning, whereas *liquid propellant rocket engines* use liquid propellants that are fed under pressure from separate tanks into a thrust chamber, where they react.

Electric propulsion comprises a family of systems that use an electric power source which is physically separate from the mechanism that produces the thrust itself. In *electro-thermal propulsion* the propellant is heated by heated resistors or electric arcs and then expanded to supersonic velocity as it is done in chemical propulsion systems. On the other hand, *ion and plasma drives* use electric and magnetic fields to accelerate ions or highly ionized plasmas, therefore not applying thermodynamic expansion to the propellant.

Beam-powered propulsion or directed energy propulsion refers to a broad family of methods, some of them experimental or speculative, that use an external power source to propel the spacecraft. Some of them concentrate beamed energy to heat a stored propellant, such as solar thermal rockets, while others use passive devices that take advantage of the radiation pressure, such as solar or laser-based sails.

Lastly, **nuclear propulsion** is a proposed spacecraft propulsion technology that delivers heat to a working fluid by means of a nuclear reaction, either by atomic fission or fusion. While they would be extremely useful for shortening interplanetary travel time and the first are at least feasible with today's technology, to date no nuclear thermal rocket has flown and they present too many technological and environmental problems, so we won't discuss more about them.

To this day, no optimal spacecraft propulsion method exists and when designing an space mission one or more of them must be chosen according to the desired requirements and the available technology. Figure 1.3 displays a graphical summary of performance characteristics of several spacecraft propulsion methods according to their energy source. As mentioned before, the only method that has enough power to escape the gravitational field of the Earth is chemical propulsion, whether in the form of solid rockets or liquid engines: the rest of the methods provide an acceleration (or, equivalently, thrust) that is orders of magnitude below and that does not reach the required thrust-to-drag ratio to lift the spacecraft to the atmosphere.

However, chemical propulsion is not without limitations. If we recover the Tsiolkovski equation

$$\Delta v = v_e \ln \frac{m_0}{m_f} \tag{1.1}$$

we notice that the obtained increment in velocity Δv is directly proportional to the *effective exhaust velocity* v_e and that increases with the *mass fraction* m_0/m_f. Effective exhaust velocity can be also written as

$$v_e = I_{sp} g_0 = \frac{T}{\dot{m}}$$

where I_{sp} is the specific impulse, g_0 is the standard gravity, T is the exerted thrust and $\dot{m}$ is the mass flow rate. Therefore, v_e is a measure of the *efficiency* of the propulsion system. If our effective exhaust velocity (or, equivalently, specific impulse) is low, we will need a higher mass fraction to gain the required velocity increment. Notice that this equation is ideal and does not take into account loses due to atmospheric drag or the gravitational forces.

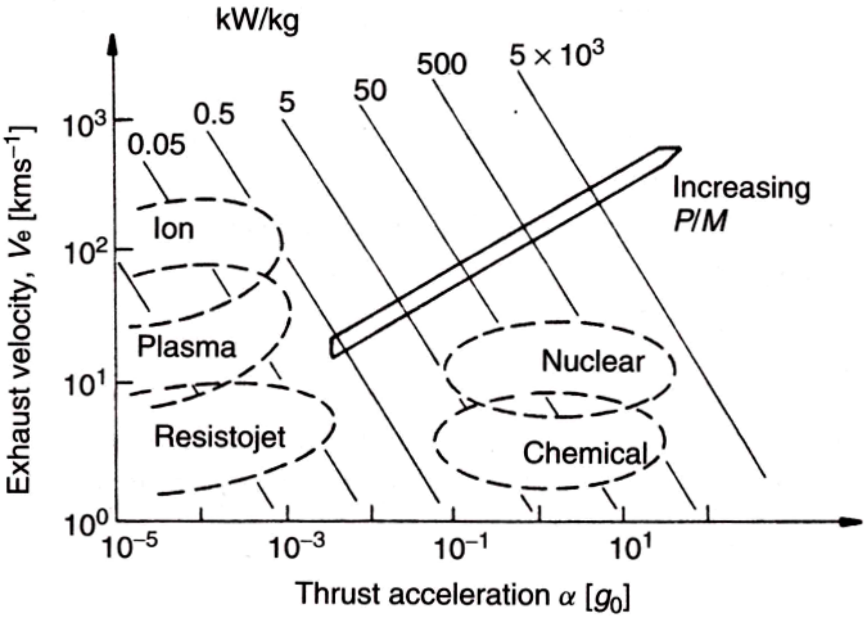

Figure 1.3: Performance characteristics of spacecraft propulsion methods according to energy source.

While the maximum thrust achievable with chemical propulsion is more than 10^6 newton, the specific impulse using current technology is in the range of 250 to 500 seconds approximately. This is a comparatively low value for the efficiency and creates the need of carrying high amounts of propellant mass, both for expendable rockets that raise the payload into orbit and also for attitude control and orbit maintenance systems for the operation of the spacecraft [Curran et al., 1993].

Fortunately, the specific impulse for ion thrusters and plasma drives can be one or two orders of magnitude higher, therefore leading to much lower propellant mass requirements. However, although these systems are more efficient in terms of the consumed mass for a given trajectory, due to the limitation of electrical power on board the spacecraft the available thrust is significantly lower than chemical rockets. As a result, low-thrust systems in general and electric propulsion systems in particular must be used during long periods of time that can range between weeks and months.

This sole fact renders the mathematical treatment of low-thrust trajectories much more complicated: on the one hand they cannot be treated as ballistic arcs anymore since there will be a permanent force acting on the spacecraft in addition of the gravitational field, and on the other hand there will be an infinite number of possible trajectories between two points and fixed times [Woodcock et al., 2002]. In the following sections we analyze the consequences of these two facts.

1.3 The perturbed two-body problem

The equation of the perturbed relative motion of two bodies is [Battin, 1999]

$$\frac{\mathrm{d}^2\mathbf{r}}{\mathrm{d}t^2} + \frac{\mu}{r^3}\mathbf{r} = \mathbf{a}_d \tag{1.2}$$

where $\mathbf{r}$ is the position vector of one body with respect to another, in our case the spacecraft with respect to its attractor, and $\mathbf{a}_d$ is the non-Keplerian or perturbation acceleration. This equation arises when analyzing the n-body problem, in which case the perturbation acceleration is due to the gravitational attraction of the other $n-2$ bodies, but $\mathbf{a}_d$ can be more general and include the action of a low-thrust device. This action can be controlled in the case of electric propulsion systems or be subject to external factors in the case of beam-powered propulsion systems such as the radiation pressure from the Sun.

The most straightforward method for determining the position $\mathbf{r}(t)$ and velocity $\mathbf{v}(t)$ in the case of a nonzero perturbing acceleration is a direct numerical integration of 1.2 in rectangular coordinates, known as Cowell's method [Battin, 1999]. However, this equation becomes singular when the distance between the two bodies approaches zero, is nonlinear and its solution is unstable, so a number of regularizations and transformations have been proposed in the literature [Baù et al., 2013].

Another approach to integrate the equations of motion is to use a set of orbital elements as dependent variables and then apply variational techniques. This technique, called *variation of parameters*, assumes that we can use the solution of the unperturbed system to represent the complete solution [Vallado, 2001]. The advantage is that most orbital elements stay constant for the case of pure Keplerian motion, and depending on the nature of the perturbing forces, the effects on the orbit can be circumscribed to a small subset of orbital elements. Furthermore, we can easily separate the short period, long period and secular effects on each of those elements. Two forms of variational equations exist: the Lagrangian form assumes the perturbation function can be derived from a potential, and the Gaussian form removes this constraint [Battin, 1999]. The former is useful when analyzing conservative perturbations, like the effects of the non-spherical shape of the Earth, whereas the latter is more general and works for arbitrary perturbation accelerations.

There are many forms of the Gauss planetary equations, depending on the element set in use and the reference frame chosen for projecting the perturbation acceleration components. We reproduce here the form derived by [Battin, 1999], which uses classical orbital elements $(a, e, i, \Omega, \omega, M)$ and polar coordinates (f_r, f_θ, f_h):

$$\frac{\mathrm{d}a}{\mathrm{d}t} = \frac{2a^2}{h}\left(e\sin\theta f_r + \frac{p}{r}f_\theta\right) \tag{1.3}$$

$$\frac{\mathrm{d}e}{\mathrm{d}t} = \frac{1}{h}\left(p\sin\theta f_r + \left((p+r)\cos\theta + re\right)f_\theta\right) \tag{1.4}$$

$$\frac{\mathrm{d}i}{\mathrm{d}t} = \frac{r\cos\left(\omega+\theta\right)}{h}f_h \tag{1.5}$$

$$\frac{\mathrm{d}\Omega}{\mathrm{d}t} = \frac{r\sin\left(\omega+\theta\right)}{h\sin i}f_h \tag{1.6}$$

$$\frac{\mathrm{d}\omega}{\mathrm{d}t} = \frac{1}{he}\left(-p\cos\theta f_r + (p+r)\sin\theta f_\theta\right) - \frac{r\sin\left(\omega+\theta\right)\cos i}{h\sin i}f_h \tag{1.7}$$

$$\frac{\mathrm{d}M}{\mathrm{d}t} = n + \frac{b}{ahe}\left(\left(p\cos\theta - 2re\right)f_r - (p+r)\sin\theta f_\theta\right) \tag{1.8}$$

$$\tag{1.9}$$

Where a is the semimajor axis, e the eccentricity, i the inclination, Ω the right ascension of the ascending node, ω the argument of periapsis, θ the true anomaly, M the mean anomaly, $(\omega + \theta) = \omega + \theta$ the argument of latitude, r the distance to the center of mass of the attractor (or equivalently, the modulus of the position vector $r = |\mathbf{r}|$, $p = a(1-e^2)$ the *semilatus rectum* and h the specific angular momentum. The three components of the disturbing force are mutually orthogonal: f_r points outwards along the position vector, f_h is perpendicular to the plane of the orbit and f_θ completes the trihedron. Notice that for small values of the eccentricity or the inclination the variation of some parameters over one orbit can be very large, and for circular ($e = 0$) or equatorial ($i = 0$) orbits the equations become singular [Vallado, 2001]. In these cases, a different set of orbital elements would have to be chosen. A recent development in this regard can be found in [Baù et al., 2013].

With these equations, a known perturbation acceleration or guidance law and an appropriate numerical integration scheme we would be able to integrate the equations of motion 1.2 and compute the variation of the dependent variables in time, apart from the numerical errors of the method and the available precision of the computer. However, if we have full control of $\mathbf{a}_d$ we are now in charge of selecting the appropriate control law to minimize the travel time, the fuel consumption or any other cost function of our desire, subject to the constraints of our problem. This is further explained in the following section.

1.4 Trajectory optimization

The need to optimize the trajectory of a spacecraft comes from the stringent mass fraction constraints discussed in 1.2 and predates the introduction of low-thrust systems. One example is the design of the Venus-Earth-Earth Gravity Assist (VEEGA) trajectory for the Galileo spacecraft, which took advantage of three flybys to achieve a high escape velocity while leaving enough propellant to perform the Jupiter orbit insertion and a tour around its moons [D'Amario et al., 1989]. This mission had the additional constraint of visiting two asteroids, so the design of its trajectory involved checking closest approaches with several thousand objects and reoptimizing for each candidate.

In any case, apart from this cases of parametric optimization, the use of continuous thrust devices imposes the need of modeling and determining the full time history of the thrust magnitude and direction, effectively becoming an optimal control problem [Conway, 2010]. These optimization problems are particularly difficult for several reasons: the concurrence of short period and very long period phenomena, the indetermination of the terminal conditions or even the structure of the problem itself (which might be subject to optimization) and the presence of discontinuities as perfectly valid trajectories (namely impulsive ΔV changes) are some of them.

There are several numerical optimization methods for solving these kind of problems, generally divided into two families:

Indirect methods usually make use of the analytical necessary conditions from the calculus of variations to form a two-point boundary-value problem associated with the equations of motion 1.2. Their advantage is that the obtained solution is smooth and accurate, but on the other hand they tend to get stuck in local optimum and it is therefore difficult to give a reasonable initial guess.

Direct methods, on the other hand, transform the optimal control problem into a parameter optimization problem by discretization. There are many different kinds of direct solution methods, depending on what degree of parametrization is chosen.

Although some smoothness properties of the solution are lost in the discretization, they converge more easily than indirect methods and are more flexible and robust in the face of problem modifications or changes in the formulation. The great majority of optimal space trajectories are nowadays determined by direct methods [Conway, 2010].

There exist also *hybrid methods* that combine the previous two and try to bring the advantages of both of them.

In some very restricted cases there are analytical solutions of the problem. They usually make use of very strong restrictions, such as fixing the acceleration instead of the thrust [Battin, 1999] or restricting to: very low-thrust transfers between circular orbits. Even though they are of very narrow applicability and difficult to develop, analytical solutions are always less expensive in terms of computation power and provide unique physical insight on the problem under study [Lawden, 1963]. It is within this context where we propose collecting several of these analytical solutions, implementing them in software and publishing them in the open, as will be explained in 1.6.

1.5 Orbital mechanics software

There are several software packages devoted to orbital mission analysis, flight control, orbital optimization and other topics related to Astrodynamics and Space Operations. While it is not our intention to make an exhaustive list, we would like to mention two open source alternatives that are in use nowadays and that might overlap in some way with our own solution.

- `Orekit` is a space dynamics library written in Java that aims at providing accurate and efficient low level components for the development of flight dynamics applications. It is developed by an international community led and sponsored by CS Systèmes d'Information. Although it does not have complex continuous thrust maneuvers build in, it features the necessary building blocks to program them*.

- `GMAT` is an open-source space mission analysis tool designed to model, optimize, and estimate spacecraft trajectories in flight regimes ranging from low Earth orbit to lunar applications, interplanetary trajectories, and other deep space missions. While Orekit target audience is mostly developers, GMAT is more appropriate for the end user. It is developed in the open by NASA.

We now comment how we have used the Python programming language to develop our own solution from scratch and its advantages and drawbacks.

1.5.1 Python

The Python programming language was started by Guido van Rossum in 1989 as a successor to the ABC language, and v1.0 was released in 1994†. It is therefore not new, and in fact it predates the Java language, first released in 1996. On the other hand, Python first uses for scientific purposes appeared as early as 1995, with the creation

*See https://www.orekit.org/static/apidocs/org/orekit/forces/maneuvers/ConstantThrustManeuver.html

†http://python-history.blogspot.com.es/2009/01/brief-timeline-of-python.html

```
while count < numiter:
    y = norm_r0 + norm_r + A * (psi * c3(psi) - 1) / c2(psi)**.5
    # ...
    xi = np.sqrt(y / c2(psi))
    tof_new = (xi**3 * c3(psi) + A * np.sqrt(y)) / np.sqrt(k)

    if np.abs((tof_new - tof) / tof) < rtol:
        # Convergence check
        break
    else:
        count += 1
        if tof_new <= tof:  # Bisection check
            psi_low = psi
        else:
            psi_up = psi
        psi = (psi_up + psi_low) / 2
```

Figure 1.4: Fragment of BMW-Vallado algorithm for Lambert's problem in Python. Notice its resemblance to pseudocode.

of a special interest group on numerical arrays [Millman and Aivazis, 2011]. However, in recent times the ecosystem has greatly improved, with the application of Open Development principles, the increasing interest and involvement of private companies and the generous funds given to projects like IPython [Perez and Granger, 2007] and Jupyter. Nowadays, it is one of the most used languages in fields like Astronomy [Momcheva and Tollerud, 2015] as well as small-to-medium Data Science, and heavily trusted for teaching undergraduate Computer Science in top universities [Guo, 2014]. In figure 1.4 we can see a fragment of one of the Bate-Mueller-White algorithm to solve Lambert's problem implemented in Python.

One of the most important differences between Python and compiled languages like Fortran or C is its dynamic typing nature. The variety of type systems across has traditionally been a major source of debate among programmers, and in fact some studies suggest that there is "a small but significant relationship between language class and defects" [Ray et al., 2014]. Languages featuring dynamic typing, as it is the case with Python, are often easier to write and read but more difficult to debug, as there are no guarantees about the types of the arguments, and have worse performance. While there is an increasing interest in developing type inference systems (see for instance the Julia and Scala languages), these are extremely difficult to set up for languages like Python [Cannon, 2005].

1.5.2 poliastro

The code developed for this Master thesis is based on `poliastro`, an open source, pure Python library for Astrodynamics and Orbital Mechanics focused on interplanetary applications and released under the MIT license[Cano Rodríguez et al., 2017]. To overcome the limitations mentioned in 1.5.1 regarding the performance of the Python programming language, `poliastro` relies on `numba`, an open source library which can infer types for array-oriented and math-heavy Python code and generate optimized machine instructions using the LLVM compiler infrastructure.

Numba works by inferring the types of the variables of a Python function and refining them in several stages until it generates assembly code for the desired platform. In

```
# --- LINE 29 ---
#   $103.3 = unary(fn=-, value=psi)  :: float64
#   $103.4 = global(gamma: <built-in function gamma>)  :: [...]
#   $const103.5 = const(int, 5)  :: int64
#   $103.6 = call $103.4($const103.5)  :: (int64,) -> float64
#   $103.7 = $103.3 / $103.6  :: float64
#   delta = $103.7  :: float64

delta = (-psi) / gamma(2 + 2 + 1)
```

Figure 1.5: Example of numba annotation along corresponding line of Python code.

Version	Minimum	Maximum	Median	Relative	IQR
Intel ifort, `-O2`	594620.8	654121.4	623536.2	**1.0**	25861.2
GNU gfortran, `-O2`	358478.2	505127.0	454613.6	**0.729**	68265.5
poliastro, numba	197610.9	206153.2	203615.8	**0.327**	3296.5
poliastro, pure Python	3502.7	3703.0	3639.6	**0.006**	65.6

Table 1.1: Benchmarking results: solutions per unit time (higher is better).

figure 1.5 we see a small fragment of the code that implements the Stumpff functions in poliastro along with its so-called Numba Intermediate Representation, which is the first stage of the optimization process.

In [Cano Rodríguez et al., 2016] detailed benchmarks were conducted to compare the performance of core Astrodynamics algorithms written in Python as featured in `poliastro` and in Fortran, measuring the number of solutions computed per unit time. The tests were performed in a virtualized CentOS machine to avoid interferences from the outside world and provide homogeneous results. The results are summarized in table 1.1.

On the other hand, `poliastro` relies on well-tested, community-backed libraries for low level astronomical tasks, such as Astropy[Robitaille et al., 2013] and jplephem. Astropy provides several utilities for handling astronomical times and performing reference frame conversions and also allows the user to introduce quantities directly with its physical units, improving the usability of the software and reducing the probability of unit errors. jplephem, on the other hand, presents a Python layer to interact with the JPL ephemeris files to compute the position and velocities of the celestial bodies with high precision.

1.6 Contribution of this Master thesis

While in theory one can always resort to numerical methods to solve orbital optimization problems, in practice having analytical solutions is highly valuable: not only they are usually quicker to compute and can be used as initial guesses for iterative algorithms with small radius of convergence, but also they serve as a vehicle to understand the nature of the problem and unveil physical relations that can be lost in numerical solutions. Anyhow, analytical solutions are very difficult to find in general, and very few examples of mathematically tractable cases exist [Battin, 1999].

On the other hand, although there are some existing or ongoing projects that already implement one or more of these algorithms (see for instance [Carlo et al., 2016], which presents a MATLAB optimization package) we have been unable to find open source implementations. Aside the philosophical implications of free software, it would be desirable to have a working implementation of these steering programs openly available, royalty-free and easy to obtain which can be then audited by a wide majority of researchers and not those in possession of a specific software license. The main object of this work is to produce such software (see appendix B for installation instructions).

Lastly, each of the guidance laws considered for study are validated against numerical examples available in the literature, with the objective of explicitly stating the margin of error between our software and the used references, and on the other hand serve as inspiration or starting point for future research projects that want to include other guidance laws or improve the existing ones.

"As for almost every other application of mathematics to practical affairs, therefore, the analytical and numerical approaches are here complimentary rather than alternative."

DEREK F. LAWDEN (1919 - 2008)

Chapter 2

Methodology

2.1 Cases in study

For this study we have considered a variety of low-thrust guidance laws that are available in the literature. Due to the scarcity of mathematically tractable examples in the field, we have focused our attention in the ideal case of fixed-acceleration maneuvers, which effectively take the variation of mass of the spacecraft out of consideration [Battin, 1999]. Thanks to this assumption, the required time of flight for each maneuver will be:

$$t_f = \frac{\Delta V_{\text{total}}}{f}$$

where f is the specific thrust, or equivalently the magnitude of the perturbing acceleration.

On the other hand, and again to adhere ourselves to straightforward analytical solutions, we have selected those references that make no assumptions about the natural orbital perturbations and do not take into account real world phenomena like eclipses. However, [Kéchichian, 1997] notes that it is useful to develop suboptimal solutions that are as realistic as possible while retaining the analytic nature, so we might consider them for a future work.

Lastly, for the implementation and discussion of results we chose the guidance laws that had accompanying numerical validation cases we could find in the literature. This is our final selection of guidance laws for low-thrust trajectories:

- Optimal transfer between circular coplanar orbits $a_0 \rightarrow a_f$ [Edelbaum, 1961, Burt, 1967]
- Optimal transfer between circular inclined orbits $(a_0, i_0) \rightarrow (a_f, i_f), e = 0$ [Edelbaum, 1961, Kéchichian, 1997]
- Quasi-optimal eccentricity-only change $e_0 \rightarrow e_f$ [Pollard, 1997]
- Simultaneous eccentricity and inclination change $(e_0, i_0) \rightarrow (e_f, i_f), a_0 = a_f$ [Pollard, 2000]
- Argument of periapsis adjustment $\omega_0 \rightarrow \omega_f$ [Pollard, 1998]

The assumptions for developing each of these guidance laws is explained in the corresponding section of this chapter.

2.2 Equations of motion

To validate the analytical solutions provided by the guidance laws in study, we have compared the results of the computations with a numerical integration of the equations of motion. To accomplish that, we have performed a reduction of order on equation 1.2:

$$\mathbf{u} \longleftarrow \begin{pmatrix} \mathbf{r} \\ \mathbf{v} \end{pmatrix} = \begin{pmatrix} x \\ y \\ z \\ v_x \\ v_y \\ v_z \end{pmatrix}$$

to arrive to a system of equations of the form:

$$\frac{\mathrm{d}\mathbf{u}}{\mathrm{d}t} = \begin{pmatrix} \dot{\mathbf{r}} \\ \dot{\mathbf{v}} \end{pmatrix} = \begin{pmatrix} v_x \\ v_y \\ v_z \\ -\mu\frac{x}{r^3} + a_{dx} \\ -\mu\frac{y}{r^3} + a_{dy} \\ -\mu\frac{y}{r^3} + a_{dz} \end{pmatrix} = \mathbf{f}(t, \mathbf{u}, \mu, \mathbf{a}_d) \tag{2.1}$$

where $\mathbf{a}_d = \mathbf{a}_d(t, \mathbf{u}, \mu)$ is the guidance law in study. In all the cases we have considered there is no explicit dependence with time, its presence being an implementation detail.

To finally integrate the equations of motion we have used a Runge-Kutta method of order 8(5,3) available in SciPy as `dop853`*. This method is based on an 8(6) method by Dormand and Prince modified to use a 5th order error estimator with 3rd order correction [Hairer et al., 1993]. It is highly efficient, requiring fewer function evaluations for the same relative tolerance than other popular integrators, and provides a cheap numerical approximation of the solution between the integration points that can be used for plotting and event detection (dense output, see again [Hairer et al., 1993] for details)†. Unless noted otherwise, we chose a maximum relative tolerance of 10^{-10}.

The validation of each guidance law is always done in two steps: first the analytical results of the corresponding paper for time of flight t_f and cost ΔV are confirmed, and then an integration of the equations of motion for a time t_f is perform and the final orbital elements are compared to the expected ones.

2.3 Planar maneuvers

Let us first focus on the planar maneuvers, those that only affect the semimajor axis a, eccentricity e and argument of periapsis ω of the orbit. While some of them can be obtained as particular cases of more complicated guidance laws studied in 2.4, for the sake of simplicity and clarity of exposition we decided to consider them separately.

*This method is also recommended for Orekit, see `https://www.orekit.org/site-orekit-8.0/apidocs/org/orekit/propagation/numerical/NumericalPropagator.html`

†Nonetheless, the dense output is not exposed to SciPy, see `https://github.com/scipy/scipy/blob/v0.18.1/scipy/integrate/dop.pyf`.

2.3.1 Semimajor axis change

In this section we discuss several strategies to perform a transfer between two coplanar, near-circular orbits, and we select the tangential one among the other solutions for practical considerations.

Burt studied the secular changes in the classical orbital elements originated by several thrust programs assuming that the perturbing forces are "sensibly constant over one orbital period" and also "small enough to produce a negligible variation" of each element during that period [Burt, 1967]. For the semimajor axis in particular, and using results previously obtained by King-Hele, he considers two guidance laws: applying thrust in a direction that is within the orbital plane and perpendicular to the radius vector (f_θ) and along the direction of the velocity (f_T). The increment of semimajor axis a for each of them is:

$$\left.\frac{\widetilde{\mathrm{d}a}}{\mathrm{d}t}\right|_{f_\theta} = \frac{2a^{3/2}}{\mu^{1/2}}\left(1 - e_0^2\left(\frac{a_0}{a}\right)^{3/2}\right)^{1/2} f_\theta$$

$$\left.\frac{\widetilde{\mathrm{d}a}}{\mathrm{d}t}\right|_{f_T} = \frac{2a^{3/2}}{\mu^{1/2}}\left(1 - \frac{1}{4}e_0^2\frac{a_0}{a} + \mathcal{O}(e_0^4)\right) f_T$$

While the indirect variation of eccentricity is:

$$\left.\frac{e}{e_0}\right|_{f_\theta} = \left(\frac{a_0}{a}\right)^{3/4}$$

$$\left.\frac{e}{e_0}\right|_{f_T} = \left(\frac{a_0}{a}\right)^{1/2}\left(1 + \frac{3}{16}e_0^2\left(1 - \frac{a_0}{a}\right) + \mathcal{O}(e_0^4)\right)$$

For $a_f < 4a_0$ the secular increase in semimajor axis is greater for f_T, whereas for $a_f > 4a_0$ the increase is greater for f_θ (see figure 2.1 and [Burt, 1967]). On the other hand, we notice that if we enlarge the orbit in both cases the eccentricity gets lower, and vice-versa. For eccentric orbits and large increases of semimajor axis, the reduction of eccentricity can be significant. For this reason, for $e > 0.843$ Burt suggests alternative strategies that only affect one orbital element.

In the absence of numerical example to validate these strategies, we opted to focus on the near-circular case, where the eccentricity will remain essentially the same when enlarging the orbit and the difference between tangential and perpendicular is negligible. This is the starting point of Edelbaum's original analysis, where he already noticed that the tangential program is not necessarily optimal, in agreement with the results of Burt. Quoting [Edelbaum, 1961]:

> It is demonstrated in (17) [Lawden 1958, AN] that the optimum thrust direction for escape with low thrust is halfway between the tangent to the orbit and the normal to the radius vector. This result is also largely applicable to transfer between circular orbits. For the purposes of this report, the difference between this program and tangential thrust can be considered negligible.

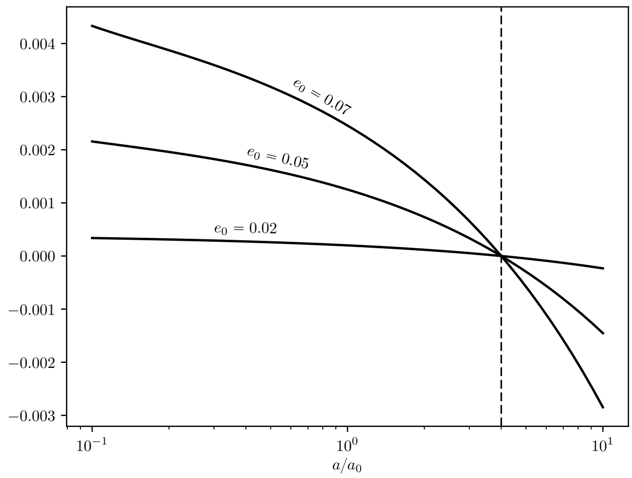

Figure 2.1: Difference of $\frac{da}{dt}$ between tangential (f_T) and perpendicular (f_2) thrust.

Kéchichian later reformulated Edelbaum's theory by changing the independent variable, arriving to an algorithm that is much easier to implement in a computer program [Kéchichian, 1997]. The problem of coplanar circle-to-circle transfer is actually a particular case of the more general problem studied by these authors, which is explained in 2.4.1.

The required ΔV will be:

$$\Delta V = |V_0 - V_f| = \left| \sqrt{\frac{\mu}{a_0}} - \sqrt{\frac{\mu}{a_f}} \right|$$

and the required time of flight $t_f = \Delta V / f$ as explained at the beginning of the chapter. As we were unable to find satisfactory validation examples for this otherwise simple case, it will be treated jointly with the combined semimajor axis and inclination transfer.

2.3.2 Eccentricity change

Although Edelbaum already presented an optimal strategy for small changes of eccentricity, [Pollard, 1997] suggests an alternative approach: consider four simplified, in-plane steering laws and compute the change of the orbital elements for each of them. The original control law from Edelbaum is $\tan \alpha_{\text{opt}} = 1/2 \tan \theta$, where α is the angle between the velocity vector and the projection of the thrust onto the orbital plane, or pitch angle[‡]. The required velocity increment would be [Edelbaum, 1961]:

$$\Delta V_{\text{opt}} = \frac{\pi}{4\,\mathrm{E}(3/4)} \sqrt{\frac{\mu}{a}} \Delta e \simeq 0.649 \sqrt{\frac{\mu}{a}} \Delta e$$

where E is the complete elliptic integral of the second kind. Fortunately, the optimal steering law is very close to a fixed inertial direction in space, and hence applying

[‡]Notice that this guidance law no longer fits into the assumptions used by Burt, since the components of the thrust are no longer constant over one orbital period.

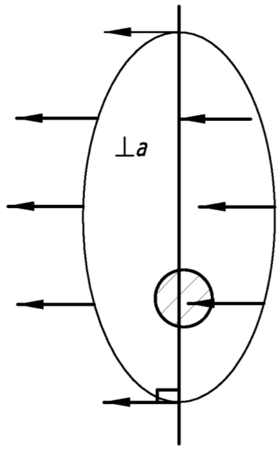

Figure 2.2: Steering law perpendicular to the semimajor axis.

thrust along a fixed direction perpendicular to the semimajor axis of the ellipse yields a required velocity increment only 3 % higher [Pollard, 1997]:

$$\Delta V = \frac{2}{3}\sqrt{\frac{\mu}{a}}\,|\arcsin e_0 - \arcsin e_f| \simeq 0.667\sqrt{\frac{\mu}{a}}\Delta e$$

The last approximate relation only holds for small eccentricities. The secular change of eccentricity in this case is:

$$\left|\frac{\mathrm{d}\tilde{e}}{\mathrm{d}t}\right| = \frac{3}{2}f\sqrt{\frac{a}{\mu}}\sqrt{1-e^2}$$

Pollard also considered the possibility of discontinuous thrust, but we will not study it in this work.

Using a geometrical reasoning, it is straightforward to see that the condition for applying thrust in a direction that is orthogonal to the semimajor axis of the orbit is $f_r = f\sin\theta$ and $f_\theta = f\cos\theta$. However, this approach can lead to numerical instabilities when the eccentricity is low and the semimajor axis itself is ill-defined. In our implementation we opted for fixing the thrust direction $\mathbf{a}_d = f\hat{\mathbf{p}}$ at the beginning of the integration:

$$\hat{\mathbf{p}} = \begin{cases} \mathbf{e}/e & \text{if } e > 0.001 \\ \mathbf{r}_0/r & \text{otherwise} \end{cases}$$

where $\mathbf{e}$ is the eccentricity vector.

To validate this guidance law we have employed an example found in [Ruggiero et al., 2011] consisting in the disposal of a Sun-synchronous orbit by increasing its eccentricity, described in table 2.1. Notice that some values are not explicitly written in the paper and we had to infer them from the results.

On the other hand, although it is not mentioned in the original paper, a change of sign is needed depending on the sign of $e_f - e_0$, so we tested the reverse case as well using the same data.

Attractor	Earth
Specific thrust	$2.4 \cdot 10^{-7}$ km/s^2
Altitude	900 km
Initial eccentricity e_0	0
Final eccentricity e_f	0.1245 (reverse-engineered from results)
Time of flight t_f	29.697 days (reverse-engineered from results)
Cost ΔV	0.6158 km/s

Table 2.1: Disposal of a Sun-synchronous orbit from [Ruggiero et al., 2011].

2.3.3 Argument of periapsis adjustment

According to [Pollard, 1998], we can adjust the argument of periapsis ω while holding the semimajor axis a and the eccentricity e constant by applying thrust along a direction that is parallel to the semimajor axis of the orbit. The resulting secular rate of change of ω is:

$$\frac{\widetilde{\mathrm{d}\omega}}{\mathrm{d}t} = \mp \frac{3}{2} f \sqrt{\frac{a}{\mu}} \frac{\sqrt{1-e^2}}{e} + \dot{\omega}$$

where again we are not considering the possibility of discontinuous thrust and $\dot{\omega}$ is the natural apsidal rotation due to a the oblateness of the attractor. For the case of the Earth, this would be [Vallado, 2001]:

$$\dot{\omega} = \frac{3nR_{\oplus}^2 J_2}{4a^2(1-e^2)^2}(4 - 5\sin^2 i)$$

where n is the mean motion and $J_2 = 1.08262668 \cdot 10^{-3}$ is the second degree harmonic. The required ΔV considering the apsidal rotation is:

$$\Delta V = \frac{\Delta\omega}{\frac{3}{2}\,\mathrm{sign}\,(\Delta\omega)\sqrt{\frac{a}{\mu}}\frac{\sqrt{1-e^2}}{e} + \frac{\dot{\omega}}{f}} \tag{2.2}$$

To validate this steering law we have taken another example from [Ruggiero et al., 2011], in this case the periapsis correction of a standard Soyuz geostationary transfer orbit (GTO) summarized in table 2.2. The shape of the orbit has been taken from the Soyuz Users Manual, issue 2 revision 0.

The example of Ruggiero does not take into account the natural apsidal rotation, and so the question opens whether this is a sensible choice or not. Looking at the denominator of equation 2.2, we are therefore interested in comparing $\dot{\omega}/f$ with the rest. Dimensional analysis yields:

$$\frac{\frac{\dot{\omega}}{f}}{\frac{3}{2}\,\mathrm{sign}(\Delta\omega)\sqrt{\frac{a}{\mu}}\frac{\sqrt{1-e^2}}{e}} \sim \frac{\frac{\dot{\omega}}{f}}{\frac{1}{V}} \sim V\frac{\dot{\omega}}{f} \sim \frac{\mu R_{\oplus}^2 J_2}{a^4 f} \approx \frac{1}{5}$$

Notice that in this case $\sqrt{1-e^2} \sim e$. The natural apsidal rotation is by no means negligible with respect to the action of the thrust, but we will accept not taking it into account in the absence of better numerical validation examples.

Attractor	Earth
Specific thrust	$2.4 \cdot 10^{-7}$ km/s^2
Altitude of apoapsis $r_a - R_\oplus$	35950 km
Altitude of periapsis $r_p - R_\oplus$	250 km
Initial argument of periapsis ω_0	178 deg
Argument of periapsis change $\Delta\omega_0$	5 deg
Time of flight t_f	12 days (approximate)
Cost ΔV	0.2489 km/s

Table 2.2: Periapsis correction of a standard Soyuz geostationary transfer orbit from [Ruggiero et al., 2011].

2.4 Non planar maneuvers

We now study maneuvers that affect the orbital elements which define the plane of the orbit as well as the rest, specifically the inclination i. Although there are some guidance laws to modify Ω available in the literature (see [Kéchichian, 2010, Pollard, 1997]), we have not considered them in this work because, on the one hand, the $(a_0, \Omega_0) \to (a_f, \Omega_0), e = 0$ proposed by Kéchichian can be treated in the same way as the $(a_0, i_0) \to (a_f, i_0), e = 0$ studied in section 2.4.1 and, on the other hand, Pollard suggests that taking advantage of the natural nodal regression and its dependence on altitude is preferred.

The advantage of using combined maneuvers is evident when we consider the required increment of velocity for an inclination change of a circular orbit using an impulsive maneuver [Vallado, 2001]:

$$\Delta V = 2V_0 \sin \frac{\Delta i}{2}$$

For moderate values of Δi, the required ΔV can be of the same order of the velocity itself, which is usually in the range of tenths of kilometers per second for low Earth orbits. It is therefore an extremely expensive maneuver that is better performed at large distances of the attractor, where the velocity decreases. In the following sections we will therefore study two steering programs that combine a change in inclination with another orbital element.

2.4.1 Combined semimajor axis and inclination change

Among other results, and as previously mentioned, [Edelbaum, 1961] presented an extremely interesting guidance law for a combined change of semimajor axis and inclination, which was later reformulated in [Kéchichian, 1997] to simplify the algorithms involved and potential software implementations. We comment here the most salient part of the derivation.

The thrust is applied along the tangential and out of plane components, resulting $f_T = f \cos\beta$ and $f_h = f \sin\beta$ where β, called the thrust yaw angle, will be the control parameter. If β is held constant during each revolution (hence piecewise constant) and we switch the sign at the orbital antinodes, the only orbital elements that will experiment secular variation will be a and i. Averaging out the orbital position θ transforms β in a continuous function of time, resulting in the following equations of motion:

$$\frac{\mathrm{d}\tilde{i}}{\mathrm{d}t} = \frac{2f\sin\beta}{\pi V}$$
$$\frac{\mathrm{d}\tilde{V}}{\mathrm{d}t} = -f\cos\beta$$

Here Kéchichian departs from Edelbaum and casts these equations as a minimum time transfer problem between initial and final i and V. After applying techniques from the calculus of variations, he arrives to the control law

$$\tan\beta(t) = \frac{V_0\sin\beta_0}{V_0\cos\beta_0 - f\cdot t}$$

in terms of time t and the parameters of the problem, where

$$\tan\beta_0 = \frac{\sin\left(\frac{\pi}{2}\Delta i\right)}{\frac{V_0}{V} - \cos\left(\frac{\pi}{2}\Delta i\right)}$$

Both quantities can be easily computed without ambiguity by using appropriate `arctan2` routines. The time history of both the velocity and the relative inclination are also known:

$$V = \sqrt{V_0^2 - 2V_0 f\cdot t\cos\beta_0 + f^2t^2}$$

$$i - i_0 = \frac{2}{\pi}\left(\arctan\left(\frac{f\cdot t - V_0\cos\beta_0}{V_0\sin\beta_0}\right) + \frac{\pi}{2} - \beta_0\right)$$

As will be seen graphically in the next chapter, the orbit smoothly gets larger to reduce the velocity and perform most of the inclination change far away from the attractor and then shrinks back to the required final semimajor axis. We note that for $\Delta i = 2$ rad $= 114.59$ deg both β_0 and β become zero: this represents the limit case when the spacecraft reaches infinity and then the inclination change is performed at no cost. In this case, continuous thrust is no longer advantageous and a bitangent impulsive maneuver is preferred.

To validate this guidance law we used two LEO to GEO examples explained in great detail in [Kéchichian, 1997] and summarized in table 2.3.

Attractor	Earth
Specific thrust	$3.5\cdot 10^{-7}$ km/s^2
Initial semimajor axis a_0	7 000 km
Final semimajor axis a_f	42 166 km
Initial inclination i_0	28.5 deg and 90 deg
Final inclination i_f	0 deg
Time of flight t_f	191.26295 days and 335 days
Cost ΔV	5.78378 km/s and 10.13 km/s

Table 2.3: LEO to GEO transfers from [Kéchichian, 1997].

2.4.2 Combined eccentricity and inclination change

Following previous works in the topic, [Pollard, 2000] presented several continuous thrust strategies, in particular a control law for a combined eccentricity and inclination change. Instead of deriving an optimal steering program, Pollard studies how applying an in-plane acceleration perpendicular to the semimajor axis of the orbit with the intent of changing the eccentricity affects in turn the inclination, and adjusts the yaw angle so the final eccentricity and inclination requirements are met at the same time. In this case, the secular variation of eccentricity will be

$$\left|\frac{\widetilde{\mathrm{d}e}}{\mathrm{d}t}\right| = \frac{3}{2} f \cos|\beta| \sqrt{\frac{a}{\mu}} \sqrt{1-e^2}$$

where β is again the yaw angle. Notice that setting $\beta = 0$ recovers the case studied in section 2.3.2. Under the assumption that the sign of the yaw angle reverses at minor axis crossings, the secular rate of change of the inclination will be

$$\left|\frac{\widetilde{\mathrm{d}i}}{\mathrm{d}t}\right| = \frac{f \sin|\beta|}{\pi} \sqrt{\frac{a}{\mu}} |\cos\omega| \frac{2(1+e^2)}{\sqrt{1-e^2}}$$

Note that the inclination change will be more efficient when $|\cos\omega|$ approaches 1, that is, $\omega \approx 0$ deg or $\omega \approx 180$ deg. Dividing these two rates of change and integrating assuming ω constant yields the control law [Pollard, 2000]:

$$\tan|\beta| = \left| \frac{3\pi(i_f - i_0)}{4\cos\omega \left(\log\left(\frac{e_f+1}{e_0+1}\frac{e_0-1}{e_f-1}\right) - e_f + e_0\right)} \right|$$

The required ΔV for this case will be:

$$\Delta V = \frac{2}{3}\sqrt{\frac{\mu}{a}} \frac{|\arcsin e_0 - \arcsin e_f|}{\cos|\beta|}$$

Since no numerical validation cases are explicitly given for this guidance law, we have extracted five data points from the plots of the original paper using an online software[§]. All five have been used for the analytical validation and only one (and its reverse) for the numerical validation. The cases are summarized in table 2.4.

	Case 1	Case 2	Case 3	Case 4	Case 5
Attractor	Earth				
Specific thrust	$2.4 \cdot 10^{-7}$ km/s2				
Semimajor axis a	42 164 km				
Initial eccentricity e_0	0.1	0.2	0.4	0.6	0.8
Final inclination i_f	20 deg			16 deg	
Yaw angle β **(deg)**	83.043	76.087	61.522	40	16.304
Cost ΔV **(km/s)**	1.6789	1.6890	1.7592	1.7241	1.9799

Table 2.4: Cases extracted from [Pollard, 2000].

[§]http://arohatgi.info/WebPlotDigitizer/app/

"Even if rounding error vanished, numerical analysis would remain. Approximating numbers, the task of floating-point arithmetic, is indeed a rather small topic and may even be a tedious one."

LLOYD N. TREFETHEN (1955)

Chapter 3

Results

3.1 Planar maneuvers

For the planar maneuvers, we will skip the semimajor-axis only case as explained in the previous chapter since we were not able to find a validation example, and only briefly comment the numerical results of the other two maneuvers. More details will be given for the more complicated combined maneuvers.

3.1.1 Eccentricity change

For the two cases in consideration (the one described in the previous chapter and the one corresponding to a reverse change in eccentricity) we were able to recover the expected results from [Pollard, 1997] with a relative error of 10^{-4}. This can be regarded as a nice result, given that the actual values were not present in the original paper and had to be computed or interpreted from the plots.

The same relative and absolute errors of 10^{-4} are obtained for the expected eccentricity after integrating the equations of motion, either with a positive or a negative change in eccentricity. This validates our intuition of reversing the direction of the thrust depending on the sign of Δe.

3.1.2 Argument of periapsis adjustment

For the only validation example studied in this case, we were able to match the results in [Ruggiero et al., 2011] within a relative tolerance of 10^{-2}. We observe a moderate accuracy in this case due to the fact that the time of flight given by the original paper is approximate and because of the intermediate computations needed to obtain the orbital parameters of the GTO orbit from the information given in the Soyuz User Manual. On the other hand, we did not find any validation example that did take into account the natural apsidal rotation of the orbit, so we openly question the applicability of this example. In any case, the function as it is implemented now, albeit not validated, accepts a precomputed value of $\dot{\omega}$.

Regarding the integration of the trajectory we obtained better results and managed to approximate the final argument of periapsis with a relative error of 10^{-4}.

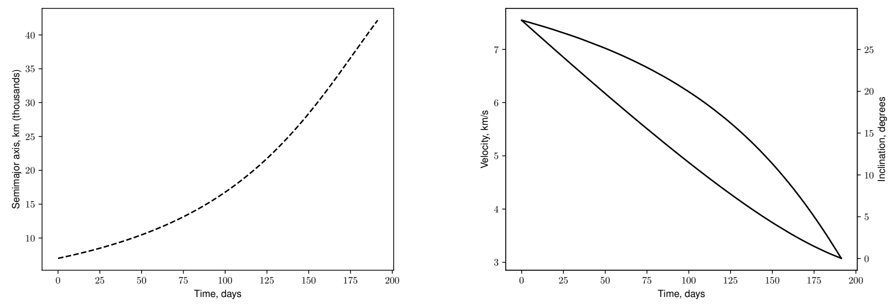

Figure 3.1: Evolution of the combined a and i transfer orbit for the case $i_0 = 28.5$ deg

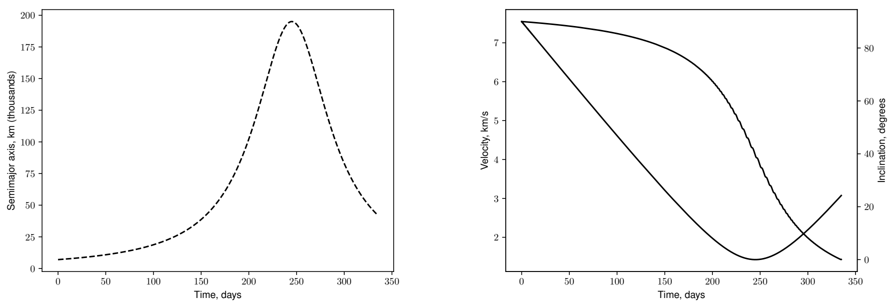

Figure 3.2: Evolution of the combined a and i transfer orbit for the case $i_0 = 90$ deg

3.2 Non planar maneuvers

3.2.1 Combined semimajor axis and inclination change

For all the evaluated cases we were able to recover the original results of [Kéchichian, 1997], with a varying degree of accuracy. For both the expected time of flight t_f and cost ΔV, a relative error of 10^{-5} was achieved for the first case, 10^{-3} for the second and 10^{-2} for the singular case. With the figures given in the original paper, we are unable to assess if the loss of accuracy is due to problems in our algorithm or simply the fact that not all decimal places are included in the text. For the sake of completeness, our results to machine precision are summarized in table 3.1.

Regarding the numerical validation, after integrating the equations of motion for a time $t = t_f$ for the two non singular cases we recover the expected values for final semimajor axis a and inclination i with a relative error of 10^{-5} for the former and an absolute error of 10^{-3} for the latter. Notice that the relative error makes no sense if we are comparing to zero. The eccentricity does not experiment significant growth and stays equal to zero with an absolute tolerance of 10^{-2}.

For completeness, we also include the plots of the time history for the semimajor axis, inclination and velocity for the two non singular cases (figures 3.1 and 3.2), recording the evolution during the integration of the equations of motion. We notice how the semimajor axis significantly increases for the second case, and how most of the inclination change takes place at that moment.

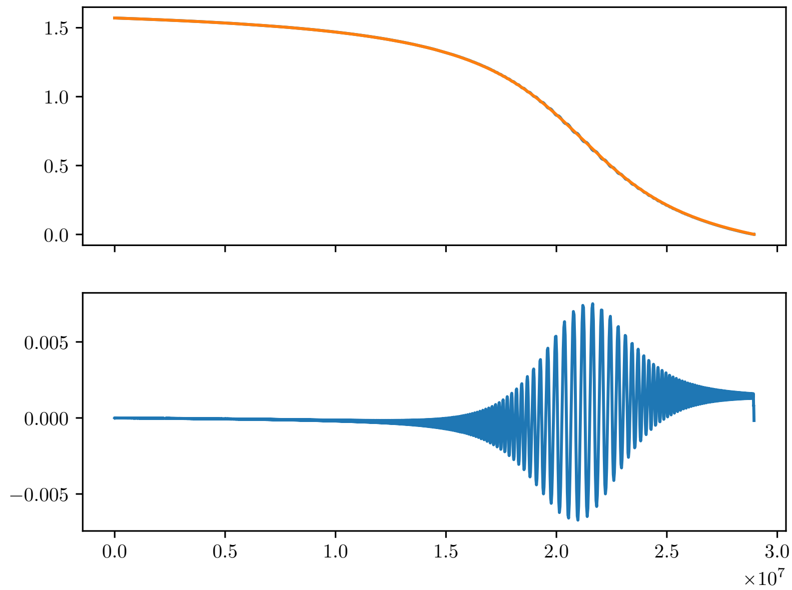

Figure 3.3: Difference between analytical and numerical value for the inclination in the combined a and i transfer orbit for the case $i_0 = 90$ deg

	Time of flight t_f			**Cost** ΔV		
	Expected	Computed	ε	Expected	Computed	ε
Case 1	191.26295	191.262282913	$0.35 \cdot 10^{-5}$	5.78378	5.7837714353	$0.15 \cdot 10^{-5}$
Case 2	335.0	335.033933749	$0.10 \cdot 10^{-3}$	10.13	10.1314261566	$0.14 \cdot 10^{-3}$
Case 3	351.0	351.211665646	$0.60 \cdot 10^{-3}$	10.61	10.6206407691	$0.10 \cdot 10^{-2}$

Table 3.1: Analytical results of the combined semimajor axis and inclination change.

To observe the differences between the numerical and the analytical solution, in figure 3.3 we have plotted the difference between the analytical evolution of the inclination and the numerical solution. While this difference is small compared to the absolute value of the inclination, we notice and oscillatory behavior that is due to the differences within each revolution. In fact, we notice that the frequency of the oscillations decreases as the orbit gets larger, and then rises again.

3.2.2 Combined eccentricity and inclination change

For each of the data points extracted from the plots in [Pollard, 2000], we were able to recover the expected values of yaw angle β and cost ΔV with a relative error of 10^{-2}. This case is even more challenging since the precision of the extracted data is limited by the resolution of the plots, even with the help of an automated software. Again, for improving the accuracy more numerical test cases would be needed.

The results of the integration of the third case ($e_0 = 0.4, i_f = 20.0$ deg) display similar accuracy: the expected final eccentricity and inclination are recovered with a

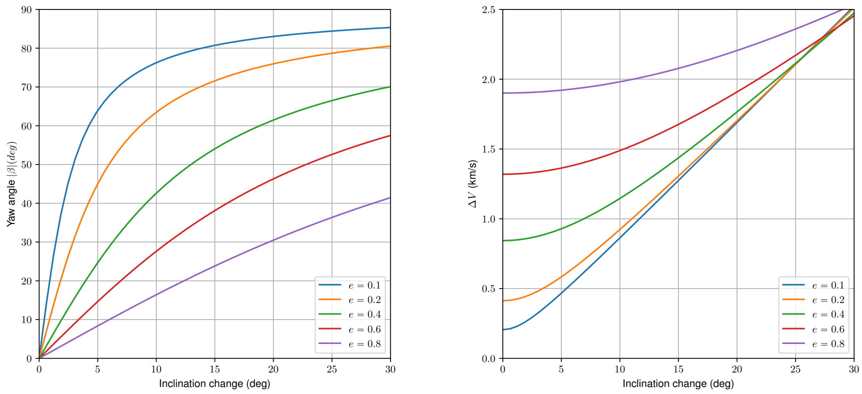

Figure 3.4: Charts of yaw angle and velocity requirements in terms of inclination and eccentricity as found in [Pollard, 2000].

relative error of 10^{-2} and 10^{-1} respectively. These tolerances are the worst for all the guidance laws in study, although still acceptable. The reverse change in eccentricity has been studied as well to test the algorithm against an initial circular orbit, yielding the same results.

We reproduce here the charts that can be found in [Pollard, 2000] using our own formulas, where it can be seen that the general trend of the plots is respected (figure 3.4).

On the other hand, we also represent the evolution of the eccentricity during the integration of the equations of motion versus the theoretical trend, given by:

$$e = \sin\left(\pm\frac{3}{2} f \cos|\beta| \sqrt{\frac{a}{\mu}} t + \arcsin e_0\right)$$

The discrepancy in the final part of the transfer corresponds with the numerical tolerance achieved for the integration.

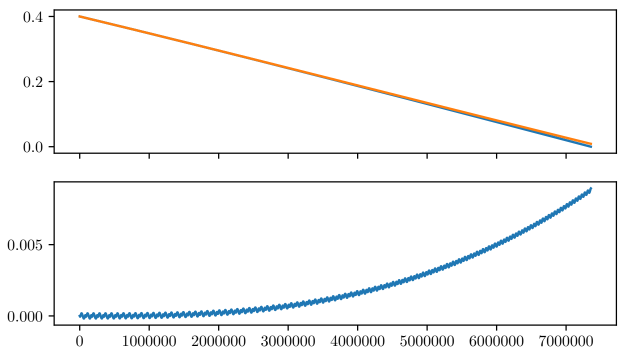

Figure 3.5: Evolution of the eccentricity in the combined e and i transfer orbit.

"But in introducing me simultaneously to skepticism and to wonder, they taught me the two uneasily cohabiting modes of thought that are central to the scientific method."

CARL E. SAGAN (1934 - 1996)

Chapter 4

Conclusions

Here we summarize the main conclusion of this work and propose some ideas for possible improvements and future areas of work. Regarding the collection of validation data and the survey of scientific articles related to analytical solutions of continuous thrust orbits, our main findings are:

- We could not find validation cases for some of the guidance laws. While it is definitely useful to have steering programs available, along with detailed mathematical derivations and the corresponding explanations, it is desirable to count also with accompanying numerical examples to improve the reproducibility of the results. Otherwise it is not possible to confirm whether potential software implementations are correct or not.

- For other articles in the literature, the numerical examples where presented in the form of plots and charts. Although having some form of graphical representation is better than nothing, accurately extracting data points from charts is extremely difficult and the attainable precision is limited by factors like the quality and resolution of the image.

- Even when there are some numerical applications of the proposed guidance laws, often the results do not have enough decimal places to clearly determine if the discrepancies are due to natural floating point arithmetic problems or implementation issues.

For all these reasons, we therefore recommend to include tables, appendices or companion software that supports the theoretical work presented in the paper. We have done so by uploading all the supporting software online, and also the scripts that were used to generate the plots and charts wherever they were relevant.

Regarding the process of implementing those steering programs in software:

- Implementation details are important, and not always given in the literature. Sometimes some equation or formula is perfectly defined from a mathematical point of view, but can present ill-defined behavior that is easily avoidable by rewriting some terms or using a different algorithm.

- While validating the guidance laws we occasionally found that it is not enough to check for the cases available in the literature. Sometimes singular behavior or corner cases are not tested and can lead to unexpected behavior. From a user experience point of view, one should consider adding guards, use defensive programming techniques, monitor the evolution of the integration or add timeouts.

From a user experience point of view, thinking about the potential users of the software:

- By decoupling the integration of the equations of motion, the control laws and the computation of the associated quantities our software is easy to reuse and extend.
- More work regarding the conversion of reference frames would be desirable, since it is too low-level now and prone to error. Incorporating velocities into Astropy reference frames for proper transformation is already on the roadmap*.
- The use of high-level features, like implicit element conversion and physical unit handling, makes the software easier to use but also has a noticeable effect on performance. Special care has been taken not to place highly dynamical or introspective code inside the evaluation of the perturbation acceleration, such as unit conversion, but there is probably more room for optimizations†.

And lastly, regarding the aspects more related to the mathematics that support the theoretical aspects of orbital mechanics in general and continuous thrust trajectories in particular:

- Focusing on analytical solutions, while arguably harder and of less immediate applicability, can be a powerful tool to gain insight about the physical phenomena under study. The evolution of velocity and inclination for the combined semimajor axis and inclination transfer or the periodic numerical noise that appears when comparing the result of the integration of the equations of motion with the analytical evolution of the elements are good examples. However, the current trend in looking for analytical solutions involves the use of Hamiltonian mechanics, calculus of variations and complicated transformations [da Silva Fernandes et al., 2016].
- The choice of orbital elements affects the convergence of the algorithms, and for classical orbital elements in particular the performance of singular orbits is hurt (equatorial, circular or both). In these cases the advantages of using formulations such as the variation of parameters are lost, since the variation of some elements within one orbital revolution are no longer small and can even lead to high numerical errors. Other sets of orbital elements could be explored, particularly equinoctial elements or DROMO elements [Peláez et al., 2006].
- Practical considerations should be taken into account, even in a preliminary design phase. For instance, for the combined semimajor axis and inclination maneuver, if the radius of the orbit increases too much the spacecraft can go through the Van Allen radiation belt, posing a threat to electronic systems [Kéchichian, 1997]. Another practical consideration that could be taken into account would be studying the efficiency efficiency of the maneuvers and the possibility of using discontinuous thrust for further propellant savings [Petropoulos, 2003]. Lastly, accounting for the periods of shadow and various eclipses can be of special importance if the spacecraft is solar-powered [Kéchichian, 1997].

*See https://github.com/astropy/astropy/issues/4344

†See https://github.com/poliastro/poliastro/pull/140

Appendix A

Methods of spacecraft propulsion

List of spacecraft propulsion methods as obtained from Wikipedia*.

*https://en.wikipedia.org/wiki/Spacecraft_propulsion#Table_of_methods

Method	$I_{sp}g_0$ (km/s)	Thrust (N)	Firing duration	Max. Δv (km/s)	TRL
Solid-fuel rocket	<2.5	<107	Minutes	7	9
Hybrid rocket	?	?	Minutes	>3	9
Monopropellant rocket	1 ~3	0.1 ~100	Milliseconds ~minutes	3	9
Liquid-fuel rocket	<4.4	<107	Minutes	9	9
Electrostatic ion thruster	15 ~210	?	Months ~years	>100	9
Hall-effect thruster (HET)	8 ~50	?	Months ~years	>100	9
Resistojet rocket	2 ~6	10−2 ~10	Minutes	?	8
Arcjet rocket	4 ~16	10−2 ~10	Minutes	?	8
Field emission electric propulsion (FEEP)	100 ~130	10−6 ~10−3	Months ~years	?	8
Pulsed plasma thruster (PPT)	20	0.1	2,000 ~10,000 hours	?	7
Dual-mode propulsion rocket	1 ~4.7	0.1 ~107	Milliseconds ~minutes	3 ~9	7
Solar sails	299792	9/km2 ~230/km2	Indefinite	>40	5
Tripropellant rocket	2.5 ~5.3	0.1 ~107	Minutes	9	6
Magnetoplasmadynamic thruster (MPD)	20 ~100	100	Weeks	?	6
Nuclear–thermal rocket	9	107	Minutes	>20	6
Propulsive mass drivers	0 ~30	104 ~108	Months	?	6
Tether propulsion	?	1 ~1012	Minutes	7	6
Air-augmented rocket	5 ~6	0.1 ~107	Seconds ~minutes	>7?	6
Liquid-air-cycle engine	4.5	103 ~107	Seconds ~minutes	?	6
Pulsed-inductive thruster (PIT)	10 ~80	20	Months	?	5
Variable-specific-impulse magnetoplasma rocket...	10 ~300	40 ~1,200	Days ~months	>100	5
Magnetic-field oscillating amplified thruster	10 ~130	0.1 ~1	Days ~months	>100	5
Solar–thermal rocket	7 ~12	1 ~100	Weeks	>20	4
Radioisotope rocket	7 ~8	1.3 ~1.5	Months	?	4
Nuclear–electric rocket	?	?	?	?	4

Continued on next page

Method	$I_{sp}g_0$ (km/s)	Thrust (N)	Firing duration	Max. Δv (km/s)	TRL
Orion Project (near-term nuclear pulse propuls...	20 ~100	109 ~1012	Days	30 ~60	3
Space elevator	?	?	Indefinite	>12	3
Reaction Engines SABRE	30/4.5	0.1 ~107	Minutes	9.4	3
Magnetic sails	145 ~750, solar wind	2/t	Indefinite	?	3
Mini-magnetospheric plasma propulsion	200	1/kW	Months	?	3
Beam-powered/laser	?	?	?	?	3
Launch loop/orbital ring	?	104	Minutes	11 ~30	2
Nuclear pulse propulsion (Project Daedalus' dr...	20 ~1,000	109 ~1012	Years	15000	2
Gas-core reactor rocket	10 ~20	103 ~106	?	?	2
Nuclear salt-water rocket	100	103 ~107	Half-hour	?	2
Fission sail	?	?	?	?	2
Fission-fragment rocket	15000	?	?	?	2
Nuclear–photonic rocket	299792	10−5 ~1	Years ~decades	?	2
Fusion rocket	100 ~1,000	?	?	?	2
Antimatter-catalyzed nuclear pulse propulsion	200 ~4,000	?	Days ~weeks	?	2
Antimatter rocket	10,000 ~100,000	?	?	?	2
Bussard ramjet	2.2 ~20,000	?	Indefinite	30000	2

Appendix B

Source code

The source code corresponding to the implementation of all the guidance laws studied in this work, as well as the corresponding test cases for the analytical and numerical validation, some ancillary scripts to produce several tables and plots of this report and the sources of the report itself are all available as an online repository at this URL:

https://github.com/Juanlu001/pfc-uc3m/

The software has been released under the MIT license, a permissive license that allows remixing and commercial use subject to some simple requirements. The complete text is copied at the end of this appendix.

The code is written with reproducibility in mind and it should be easy to install a Python development environment and run the code. These are the required steps to do so using Continuum Analytics Anaconda distribution:

1. Install Anaconda Python 3.6 from https://www.continuum.io/downloads.

2. Download latest development version of the project from https://github.com/Juanlu001/pfc-uc3m/archive/master.zip

3. Create a virtual environment:

```
$ cd pfc-uc3m
$ conda env create  # Will read environment.yml
$ source activate pfc36
(pfc36) $
```

4. Run tests

```
(pfc36) $ cd pfc-uc3m
(pfc36) $ pytest -vvv
```

Bibliography

[Battin, 1999] Battin, R. H. (1999). *An introduction to the mathematics and methods of astrodynamics*. Aiaa.

[Baù et al., 2013] Baù, G., Bombardelli, C., and Peláez, J. (2013). A new set of integrals of motion to propagate the perturbed two-body problem. *Celestial Mechanics and Dynamical Astronomy*, 116(1):53–78.

[Burt, 1967] Burt, E. (1967). On space manoeuvres with continuous thrust. *Planetary and Space Science*, 15(1):103–122.

[Cannon, 2005] Cannon, B. (2005). *Localized type inference of atomic types in python*. PhD thesis, CALIFORNIA POLYTECHNIC STATE UNIVERSITY San Luis Obispo.

[Cano Rodríguez et al., 2016] Cano Rodríguez, J. L., Eichhorn, H., and McLean, F. (2016). poliastro: An astrodynamics library written in python with fortran performance. In *Proc. 6th International Conference on Astrodynamics Tools and Techniques (ICATT)*.

[Cano Rodríguez et al., 2017] Cano Rodríguez, J. L., Lorenzo Márquez, A., Ombredanne, P., Cañardo Alastuey, J., and Cody (2017). poliastro/poliastro: poliastro 0.6.0.

[Carlo et al., 2016] Carlo, M. D., Martin, J. M. R., and Vasile, M. (2016). Camelot - computational-analytical multi-fidelity low-thrust optimisation toolbox. In *6th International Conference on Astrodynamics Tools and Techniques (ICATT)*. ESA.

[Choueiri, 2004] Choueiri, E. Y. (2004). A critical history of electric propulsion: The first 50 years (1906-1956). *Journal of Propulsion and Power*, 20(2):193–203.

[Choueiri, 2009] Choueiri, E. Y. (2009). New dawn for electric rockets. *Scientific American*, 300(2):58–65.

[Conway, 2010] Conway, B. A. (2010). *Spacecraft trajectory optimization*, volume 29. Cambridge University Press.

[Curran et al., 1993] Curran, F., Brophy, J., and Bennet, a. (1993). The nasa electric propulsion program. In *29th Joint Propulsion Conference and Exhibit*, page 1935.

[da Silva Fernandes et al., 2016] da Silva Fernandes, S., das Chagas Carvalho, F., and de Moraes, R. V. (2016). Optimal low-thrust transfers between coplanar orbits with small eccentricities. *Computational and Applied Mathematics*, 35(3):803–816.

[D'Amario et al., 1989] D'Amario, L. A., Byrnes, D. V., Johannesen, J. R., and Nolan, B. G. (1989). Galileo 1989 veega trajectory design. *Journal of the Astronautical Sciences*, 37:281–306.

[Edelbaum, 1961] Edelbaum, T. N. (1961). Propulsion requirements for controllable satellites. *ARS Journal*, 31(8):1079–1089.

[Guo, 2014] Guo, P. (2014). Python is Now the Most Popular Introductory Teaching Language at Top U.S. Universities. http://web.archive.org/web/20160218070508/http://cacm.acm.org/blogs/blog-cacm/176450%2Dpython%2Dis%2Dnow%2Dthe%2Dmost%2Dpopular%2Dintroductory%2Dteaching%2Dlanguage%2Dat%2Dtop%2Dus%2Duniversities/fulltext.

[Hairer et al., 1993] Hairer, E., Wanner, G., and P. Nørsett, S. (1993). *Solving ordinary differential equations*. Springer-Verlag, Berlin.

[Kéchichian, 1997] Kéchichian, J. A. (1997). Reformulation of edelbaum's low-thrust transfer problem using optimal control theory. *Journal of Guidance, Control, and Dynamics*, 20(5):988–994.

[Kéchichian, 2010] Kéchichian, J. A. (2010). Analytic representations of optimal low thrust transfer in circular orbits. *Spacecraft Trajectory Optimization*, 29:139.

[Lawden, 1963] Lawden, D. F. (1963). *Optimal trajectories for space navigation*. Butterworths.

[Millman and Aivazis, 2011] Millman, K. J. and Aivazis, M. (2011). Python for Scientists and Engineers. *Comput. Sci. Eng.*, 13(2):9–12.

[Momcheva and Tollerud, 2015] Momcheva, I. and Tollerud, E. (2015). Software Use in Astronomy: an Informal Survey. *ArXiv e-prints*.

[Peláez et al., 2006] Peláez, J., Hedo, J. M., and de Andrés, P. R. (2006). A special perturbation method in orbital dynamics. *Celestial Mechanics and Dynamical Astronomy*, 97(2):131–150.

[Perez and Granger, 2007] Perez, F. and Granger, B. E. (2007). IPython: A System for Interactive Scientific Computing. *Comput. Sci. Eng.*, 9(3):21–29.

[Petropoulos, 2003] Petropoulos, A. E. (2003). Simple control laws for low-thrust orbit transfers.

[Pollard, 1998] Pollard, J. (1998). Evaluation of low-thrust orbital maneuvers. In *34th AIAA/ASME/SAE/ASEE Joint Propulsion Conference and Exhibit*, page 3486.

[Pollard, 2000] Pollard, J. (2000). Simplified analysis of low-thrust orbital maneuvers. Technical report, DTIC Document.

[Pollard, 1997] Pollard, J. E. (1997). Simplified approach for assessment of low-thrust elliptical orbit transfers. In *25th International Electric Propulsion Conference, Cleveland, OH*, pages 97–160.

[Ray et al., 2014] Ray, B., Posnett, D., Filkov, V., and Devanbu, P. (2014). A large scale study of programming languages and code quality in github. In *Proceedings of the 22nd ACM SIGSOFT International Symposium on Foundations of Software Engineering - FSE 2014*. Association for Computing Machinery (ACM).

[Robitaille et al., 2013] Robitaille, T. P. et al. (2013). Astropy: A community Python package for astronomy. *Astronomy & Astrophysics*, 558:A33.

[Ruggiero et al., 2011] Ruggiero, A., Pergola, P., Marcuccio, S., and Andrenucci, M. (2011). Low-thrust maneuvers for the efficient correction of orbital elements. In *32nd International Electric Propulsion Conference*, pages 11–15.

[Siddiqi, 2000] Siddiqi, A. A. (2000). Challenge to apollo: the soviet union and the space race, 1945-1974.

[Spitzer Jr, 1952] Spitzer Jr, L. (1952). Interplanetary travel between satellite orbits. *Journal of the American Rocket Society*, 22(2):92–96.

[Sutton and Biblarz, 2016] Sutton, G. P. and Biblarz, O. (2016). *Rocket propulsion elements*. John Wiley & Sons.

[Vallado, 2001] Vallado, D. A. (2001). *Fundamentals of astrodynamics and applications*, volume 12. Springer Science & Business Media.

[Woodcock et al., 2002] Woodcock, G., Farris, R., Johnson, L., Jones, J., Kos, L., and Trausch, A. (2002). Benefits of nuclear electric propulsion for outer planet exploration. In *38 th AIAA/ASME/SAE/ASEE Joint Propulsion Conference and Exhibit, Indianapolis, IN*.