Optimally Controlling a Rocket into a Stable Orbit

Nick Seinsche
Mathematical Institute
Albert-Ludwigs-Universität
Freiburg, Germany
nick.seinsche@students.uni-freiburg.de

Michael Kindsvater
Mathematical Institute
University of Freiburg
Freiburg, Germany
michael.kindsvater@merkur.uni-freiburg.de

***Abstract*—In the following report we addess the problem of controlling a rocket that has just launched from a planet into a stable circular orbit around that planet using minimal controls in two and three dimensions. The two body-problem makes up the underlying dynamics which we modify to add rocket controls. After specifying conditions on reaching orbit, our problem may be formulated as a continuous optimal control problem which will further be discretized and solved by using CasADi's IPOPT. Finally the results will be discussed.**

***Index Terms*—cost optimal control, rockets, differential equations**

I. The Two-Body Problem

Newton's law of universal gravitation states that the force with which two objects attract each other is proportional to the product of their masses divided by the square of their distance.

Furthermore *Newton's* second law of motion implies that the attraction force is equal to the product of the objects mass and acceleration.

If the center points of the stellar objects are parametrized with respect to time via

$$x_i : [0, T] \to \mathbb{R}^d \qquad i \in \{1, 2\} \quad d \in \{2, 3\} \quad T > 0$$

and we gather them along with their velocities into a single vector,

$$z(t) := (x_1(t), x_2(t), x_1'(t), x_2'(t))^\top \in \mathbb{R}^{4d},$$

the laws above motivate the following first order, ordinary, non-linear and autonomous system of differential equations:

$$\begin{aligned} m_1 x_1''(t) &= \gamma \cdot \frac{m_1 \cdot m_2}{\|x_1(t) - x_2(t)\|^2} \cdot \frac{x_2(t) - x_1(t)}{\|x_1(t) - x_2(t)\|} \\ m_2 x_2''(t) &= \gamma \cdot \frac{m_1 \cdot m_2}{\|x_1(t) - x_2(t)\|^2} \cdot \frac{x_1(t) - x_2(t)}{\|x_1(t) - x_2(t)\|} \end{aligned} \tag{1}$$

where γ is the *Gravitational constant* and $m_1, m_2 > 0$ are the masses of the first and second body respectively. Notice that the forces oppose each other. For the purpose of this report, it is convenient to rewrite (1) as a system of first order differential equations, in order to apply numerical methods later. This is accomplished by the follwing expression:

$$z'(t) = F(z(t)) := \begin{pmatrix} x_1'(t) \\ x_2'(t) \\ f_1(x_1(t), x_2(t)) \\ f_2(x_1(t), x_2(t)) \end{pmatrix} \tag{2}$$

where f_1 and f_2 are given by the righthand side of the equations in (1) after diving by m_1 and m_2 respectively. Indeed, for a given initial value, both (1) and (2) define initial value problems with equivalent solutions.

II. Applying Controls & Dynamics

The first body, i.e. x_1, will be the rocket. It will have negligable mass compared to the stellar body, i.e. x_2, which it will orbit. We assume the rocket to be maneuverable. It has a main engine and gas thrusters that exert some force in a given direction. The sum of these forces will be considered the controls $u(t)$ of the rocket. They are the influence on the acceleration of the first body at a given time $t \in [0, T]$. Therefore the right hand side of (2) needs to be modified to incorporate these additional forces. In two dimensions this yields:

$$\tilde{f}_1(x_1, x_2, u) = f_1(x_1, x_2) + \frac{1}{m_1} \begin{pmatrix} r(t) \cdot \sin(t) \\ r(t) \cdot \cos(t) \end{pmatrix} \tag{3}$$

where the control vector is given by $u(t) = \begin{pmatrix} r(t) & \theta(t) \end{pmatrix}^\top$. In the case of three dimensions, the control vector may be written in spherical coordinates and thus

$$\tilde{f}_1(x_1, x_2, u) = f_1(x_1, x_2) + \frac{r(t)}{m_1} \begin{pmatrix} \cos(\varphi(t)) \cdot \sin(\theta(t)) \\ \cos(\theta(t)) \\ \sin(\varphi(t)) \cdot \sin(\theta(t)) \end{pmatrix}, \tag{4}$$

where the control vector is given by the triple

$$u(t) = \begin{pmatrix} r(t) & \varphi(t) & \theta(t) \end{pmatrix}^\top. \tag{5}$$

Notice that the multiplication with m_1^{-1} is necessary to convert the exerted force to the respective acceleration of the body.

Again, since the mass of the rocket is negligable compared to the stellar orbject it will orbit, it may be assumed that the position of the second body is constant and the velocity is zero for every timepoint $t \in [0, T]$. This simplifies the resulting dynamics to the following:

$$z'(t) = f(z(t), u(t)) := \begin{pmatrix} x_1'(t) \\ \tilde{f}_1(x_1(t), x_2(0), u(t)) \end{pmatrix} \tag{6}$$

where $z(t) := (x_1(t), x_1'(t))^\top \in \mathbb{R}^{2d}$ and $x_2(0) = 0 \in \mathbb{R}^d$, being the initial position of the planet.

III. Specifying an Orbit and Orbital Speed

Let $s > 0$ be the radius of the planet and $d_{\text{orbit}} > s$ be the radius of a circular orbit that the rocket shall reach, i.e. $d_{\text{orbit}} - s$ is the distance from the rocket to the surface of the planet once the rocket has reached this orbit. The orbital speed v_{orbit} is the speed the rocket has to attain in order to stay in the circular orbit defined above. This speed is given by the following expression:

$$v_{\text{orbit}} = \sqrt{\frac{(m_2 + m_1)\cdot\gamma}{d}} \approx \sqrt{\frac{m_2\cdot\gamma}{d}}, \tag{7}$$

In the light of our two dimensional Problem the following conditions are sufficient for the rocket achieving a stable orbit around the planet at time T:

(i) The distance of the rocket to the center of the planet is d_{orbit},
(ii) the velocity of the rocket is v_{orbit},
(iii) the velocity vector of the rocket is tangential to the orbital path or equivalently perpendicular to the orbit normal.

These conditions are illustrated in Fig. 1.

Once the above requirements are met, no further controls are necessary for the rocket to stay in the circular orbit of distance $d_{\text{orbit}} - s$ above the surface of the planet.

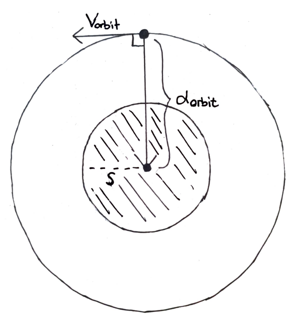

Fig. 1: Illustration of the orbital constraints for the two dimensional case.

In three dimensions, however, this is not sufficient for specifying circular orbits because rotations of the orbit are not taken into account. Therefore let $Q \in \mathbb{R}^{3\times 3}$ be a rotation matrix that describes the rotation of the orbital circle with center point $(0, 0, 0)$ and radius d_{orbit} inside the horizontal x-z-plane. The resulting rotated orbit is then given by the set of points

$$O = \left\{ Q \cdot \begin{pmatrix} d_{\text{orbit}}\cdot\cos(2\pi t) \\ 0 \\ d_{\text{orbit}}\cdot\sin(2\pi t) \end{pmatrix} \in \mathbb{R}^3 \;\middle|\; t \in [0,1] \right\}. \tag{8}$$

Specifying orbits in this way allows the three dimensional case to be treated as the two dimensional problem embedded into the x-z-plane of the three dimensional real space by imposing the terminal constraints above on $Q^\top x(T)$ instead of $x(T)$, i.e. imposing orbital contraints after a linear transform of the terminal orbit into the horizontal x-z-plane. Since in three dimensions, both the position and the velocity of the rocket at time T have an additional degree of freedom compared to the two dimensional case, two additional contraints are necessary:

(iv) The rocket lies in in the orbital plane defined by Q,
(v) the velocity vector of the rocket is contained in the orbital plane defined by Q.

IV. Continuous Time Optimal Control Problem

The problem stated in the abstract can now be formulated as a continuous time optimal control problem of the following form:

$$\begin{aligned}
\underset{z(\cdot),u(\cdot)}{\text{minimize}} \quad & \int_0^T r(t)\,dt && && (9)\\
\text{subject to} \quad & z'(t) = f(z(t), u(t)) && t \in [0,T] && (10)\\
& z(0) = \bar{z}_0 && && (11)\\
& g(z(T)) = 0 && && (12)\\
& p_{\text{min}} \le p(u'(t)) \le p_{\text{max}} && t \in [0,T] && (13)\\
& h_{\text{min}} \le h(z(t), u(t)) \le h_{\text{max}} && t \in [0,T] && (14)
\end{aligned}$$

where $T > 0$ is the time horizon, $\bar{z}_0 \in \mathbb{R}^{2d}$ is an initial state, $g : [0,T] \to \mathbb{R}^{n_g}$, $h : [0,T] \to \mathbb{R}^{n_h}$ and $p : [0,T] \to \mathbb{R}^{n_p}$ constraints, and $p_{\text{max}}, p_{\text{min}} \in \mathbb{R}^{n_p}$, $h_{\text{max}}, h_{\text{min}} \in \mathbb{R}^{n_h}$ upper and lower bounds. Since the cost function is given by the integral over the thrust r (with the thrust being contraint to non-negative values), a solution to this problem will describe an orbital maneuver with minimal thrust and therefore minimal fuel usage.

The constraints come in three different types. Firstly, the function g describes the terminal contraints, constraints on the final state of the rocket to ensure a stable orbit as discussed in section III. Since the norm and dot product are both invariant under multiplication with rotation matricies we have:

(i) Reach orbital height

$$g_1(z(T)) = \|x_1(T)\|_2 - d_{\text{orbital}} \tag{15}$$

(ii) Reach orbital velocity

$$g_2(z(T)) = \|x_1'(T)\|_2 - v_{\text{orbital}} \tag{16}$$

(iii) Velocity vector is perpendicular to the normal of the desired orbit

$$g_3(z(T)) = \langle x_1'(T), x_1(T)\rangle \tag{17}$$

To account for the third dimension there need to be two constraints added.

(iv) The rocket is on the orbital plane specified by Q

$$g_4(z(T)) = \langle Q^\top \cdot x_1(T), (0,1,0)\rangle \tag{18}$$

(v) Velocity vector is perpendicular to the binormal of the orbit, i.e. the velocity vector lies in the orbital plane defined by Q

$$g_5(z(t)) = \langle Q^\top \cdot x_1'(T), (0,1,0)\rangle \tag{19}$$

The second type of contraint imposes limitations on the derivative of the controls u. This is a non-standard constraint and is not treated in [1]. However, since the rocket is subject to mechanical limits, it is intuitively clear why $u(t+\tau)$ must be dependent on $u(t)$ for small $\tau > 0$ in order to receive meaningful optimal controls. Therefore, the function p limits how quickly we can change the controls of the rocket. Note that local differentiability of u is a sufficient condition for this constraint to be well defined. Again, the second angle θ is only relevant in the 3D case:

(i) Rate of change of thrust is limited

$$\begin{aligned} p_1(u'(t)) &= r'(t) \\ -\frac{r_{\text{max}}}{c_1} \leq p_1 &\leq \frac{r_{\text{max}}}{c_1} \end{aligned} \tag{20}$$

(ii) Rate of change of angle φ is limited

$$\begin{aligned} p_2(u'(t)) &= \varphi'(t) \\ -\frac{\pi}{c_2} \leq p_2 &\leq \frac{\pi}{c_2} \end{aligned} \tag{21}$$

(iii) Rate of change of angle θ is limited

$$\begin{aligned} p_3(u'(t)) &= \theta'(t) \\ -\frac{\pi}{c_2} \leq p_3 &\leq \frac{\pi}{c_2} \end{aligned} \tag{22}$$

Lastly, h sets bounds for the controls and the state of the rocket during the maneuver:

(i) The thrust is limited by a constant r_{max}

$$\begin{aligned} h_1(z(t), u(t)) &= r(t) \\ 0 \leq h_1 &\leq r_{\text{max}} \end{aligned} \tag{23}$$

(ii) Don't get too close to the planet

$$\begin{aligned} h_2(z(t), u(t)) &= \|x_1(t)\|_2 \\ (1+\delta) \cdot d_{\text{surface}} \leq h_2 &< \infty \end{aligned} \tag{24}$$

where $\delta > 0$ is a constant.

V. Discretization and Solving Approach

We want to solve this OCP in discrete time using *direct multiple shooting* or more specific the simultanious approach discussed in [1]. To do so, we choose a number of discretization points $N+1$ and divide the time horizon $[0, T]$ with discrete time points $t_0 = 0, t_1 = h, \ldots, t_k = k \cdot h, \ldots, t_N = T$, with h the step size $h = \frac{T}{N}$. Similarly, we discretize the functions for the state and the controls of the rocket, i.e. $z_k = z(t_k) = (x_k, x'_k)^\top$ and $u_k = u(t_k) = (r_k, \varphi_k, \theta_k)^\top$.

Now, we can formulate our discrete nonlinear optimal control problem.

$$\underset{\substack{z_0,\ldots,z_N \\ u_0,\ldots,u_{N-1}}}{\text{minimize}} \quad \sum_{k=0}^{N-1} I(r_k, h)\, dt \tag{25}$$

$$\text{subject to} \quad z_{k+1}(t) = F(z_k, u_k, h) \quad k = 0, \ldots, N-1 \tag{26}$$

$$z_0 = \bar{z}_0 \tag{27}$$

$$G(z_N) = 0 \tag{28}$$

$$P_{\text{min}} \leq P(u_k, u_{k+1}, h) \leq P_{\text{max}} \quad k = 0, \ldots, N-2 \tag{29}$$

$$H_{\text{min}} \leq H(z_k, u_k) \leq H_{\text{max}} \quad k = 0, \ldots, N \tag{30}$$

To discretize the objective, we use the function $I(r, h) = r \cdot h$ which, summed up, realizes a Riemann sum.

We approximate the dynamics $z' = f(z, u)$ using the RK4-integrator with the same step size h. That way, we can simulate the dynamics over the period of one time step and get $z_{k+1} = F(z_k, u_k, h)$ using a single RK4-step.

The terminal constraints can be used exactly as in the continuous formulation, since $t_N = T$ is a discretization point.

The function p is a bit more complicated to discretize, because we don't know the derivative of the isolated points of u_k. To achieve our goal of limiting the amount our controls can change, it is sufficient to limit the finite differences of neighboring points.

(i) Amount of change of thrust is limited

$$\begin{aligned} P_1(u_k, u_{k+1}, h) &= \frac{|r_{k+1} - r_k|}{h} \\ 0 \leq P_1 &\leq \frac{r_{\text{max}}}{60} \end{aligned} \tag{31}$$

(ii) Amount of change for consecutive angles φ_k, φ_{k+1} is limited

$$\begin{aligned} P_2(u_k, u_{k+1}, h) &= \frac{|\varphi_{k+1} - \varphi_k|}{h} \\ 0 \leq P_2 &\leq \frac{\pi}{48} \end{aligned} \tag{32}$$

(iii) Amount of change for consecutive angles θ_k, θ_{k+1} is limited

$$\begin{aligned} P_3(u_k, u_{k+1}, h) &= \frac{|\theta_{k+1} - \theta_k|}{h} \\ 0 \leq P_3 &\leq \frac{\pi}{48} \end{aligned} \tag{33}$$

This is the same as piecewise linearly interpolating the values and then limiting the resulting function's piecewise derivative. Since the boundary values P_{min} and P_{max} are constant over time, we never violate the boundaries, even between points.

Dealing with h is more straightforward again. We simply only impose the constraints at the existing discretization points.

To formulate this resulted non-linear optimization problem within Python, we used the interface of the symbolic framework CasADi [2]. To solve it, the interior point method based solver IPOPT [3] was used.

VI. Results

In this section, we want to present two examples. In both examples, we chose numbers and constants that we found worked well without specific units in mind. Using real life units was beyond the scope of this project but would have produced similar results. The exact values used and the implementation along with interactive, animated versions of the figures are available in this GitHub repository, by running the respective Python scripts: https://github.com/Nick-Seinsche/NumOptCtrlProject.

In the first example in two dimensions, we consider a rocket that is already at the height $(1+\delta)\cdot s$ with $\delta = 0.1$ and has a certain speed and is pointed at a slight angle. The calculated controls and the resulting trajectory can be seen in Figs. 2, 3 and 4.

In the second example we consider a three dimensional system. We used similar initial values, but this time we don't start perpendicularly below the target orbit. That way, controls involving both φ and θ are necessary to reach the correct orbit. Results are displayed in Figs. 5, 6 and 7.

In both two and three dimensions the optimal trajectory looks similar. There is an initial rocket burn that causes the rocket to miss the planet at the opposite side of the start point, i.e. the contraint (24) is active. This is followed by an orbital injection burn that raises the altitude to the desired orbit. In three dimensions, another smaller burn is needed to align with the orbital plane defined by the rotation matrix Q. The steepness of the thrust change seen in Figs. 3 and 6 is directly limited by the upper bound in (31).

Also using CasADi, we were able to verify the linear independance constraint qualification (LICQ, cf. [1]) at the optimal values, by checking the rank of the jacobian of the function that results by stacking all the constraints on top of each other.

VII. Further considerations

The current dynamics do not incorperate weight loss of the rocket due to burning of propellant. In the following it will be shortly outlined, how the dynamics can be modified to incorporate this:

We augment the dynamics by adding a Function $w : [0,T] \to \mathbb{R}$ which tracks the weight of the rocket over time. With the initial value $w(0) = m_1$ and the notation as in (4) and (6), we have in three dimensions

$$\begin{pmatrix} z'(t) \\ w'(t) \end{pmatrix} = f(z(t), u(t), w(t)) := \begin{pmatrix} x_1'(t) \\ \tilde{f}_1(x_1(t), 0, u(t), w(t)) \\ -e(u(t)) \end{pmatrix}$$

where $\tilde{f}_1(x_1(t), 0, u(t), w(t))$ is given by

$$\tilde{f}_1(x_1, x_2, u, w) = f_1(x_1, x_2) + \frac{r(t)}{w(t)} \begin{pmatrix} \cos(\varphi(t)) \cdot \sin(\theta(t)) \\ \cos(\theta(t)) \\ \sin(\varphi(t)) \cdot \sin(\theta(t)) \end{pmatrix}$$

and $e : [0, r_{\max}] \to \mathbb{R}_{\geq 0}$ translates the current force exerted by the rocket to weightloss of the rocket due to fuel consumption. Assuming that the main engine of the rocket provides all of the thrust, the expression $r/e(r)$ is up to a constant related to the *specific impulse* of it's main engine, i.e. how efficiently the fuel is used to create thrust. Futhermore, a contraint $w(T) \geq m_{1,\text{dry}} > 0$ where $m_{1,\text{dry}}$ is the mass of the rocket without fuel might be sensible to ensure that the rocket may not burn more propellant than it has onboard.

Another consideration is the extension to non-circular orbits. It may be of interest to find the optimal controls to a truly

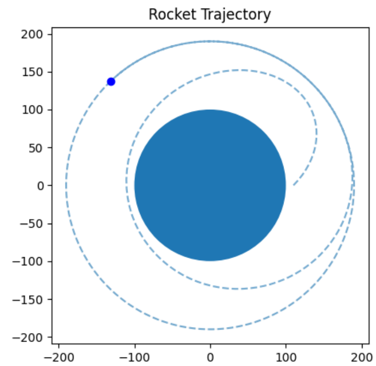

Fig. 2: Visualization of the trajectory (indicated by the dashed line) of the rocket (indicated by the blue dot) using optimal controls around a circular planet in two dimensions

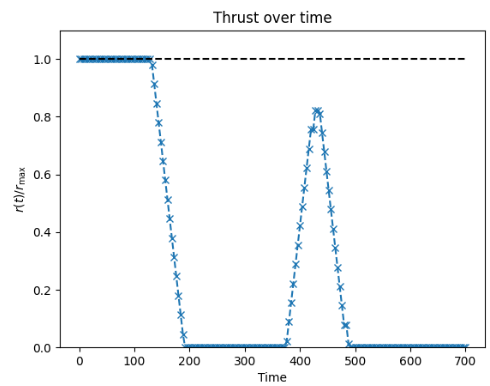

Fig. 3: Plot of the thrust divided by the maximum thrust over time of the trajectory depicted in Fig. 2. The thrust controls at the timepoints $t_0 = 0, \ldots, t_{175} = 700$ are indicated by the crosses.

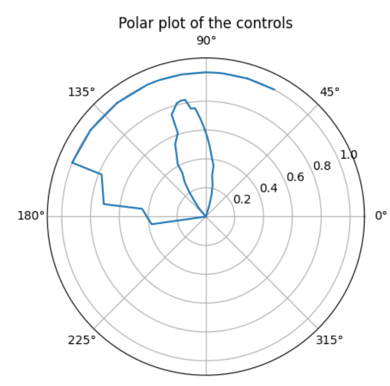

Fig. 4: Polar plot of the path of the control vector $u(t) = (r(t), \theta(t))$ over time for the trajectory depicted in Fig. 2. Again the thrust is divided by the maximum thrust.

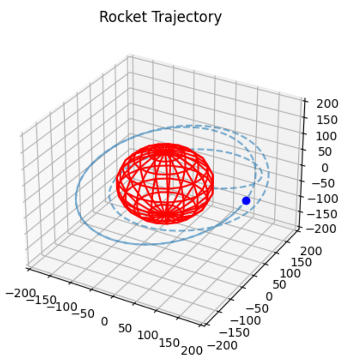

Fig. 5: Visualization of the trajectory (indicated by the dashed line) of the rocket (indicated by the blue dot) using optimal controls around a spherical planet (indicated by the red lines) in three dimensions.

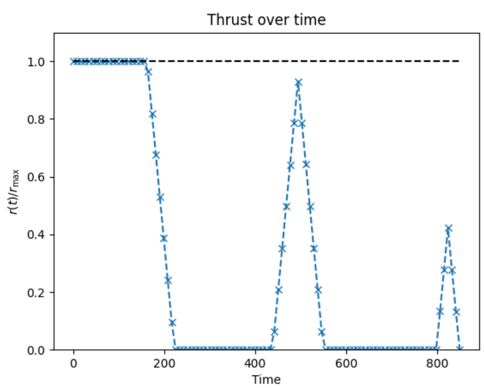

Fig. 6: Plot of the thrust divided by the maximum thrust over time of the trajectory depicted in Fig. 5. The controls at the timepoints $t_0 = 0, \ldots, t_{98} = 850$ are indicated by the crosses.

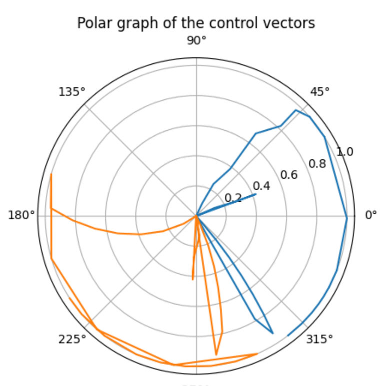

Fig. 7: Polar plot of the path of the controls $(r(t), \varphi(t))$ (in blue) and $(r(t), \theta(t))$ (in orange) which make up the controls $u(t)$ over time for the trajectory depicted in Fig. 6. Again the thrust is divided by the maximum thrust.

elliptical orbit, too. This, however, is not achieved by extenting Q to matricies which may stretch space. This is because in the case of a stable elliptical orbit of the rocket around the planet, the planet would be located at one of the focal points of the orbital ellipse. Using a simple linear transform would result in the planet being located at the barycenter of the ellipse instead. In general, one would want a bijective linear function that maps focal points (of a true ellipse) to focalpoints (of a circle). Such a function does not exists since it would not be injective.

An idea to solve this in two dimensions would be to define a $d_{\text{orbit}} : [0, 2\pi] \to \mathbb{R}_{>0}$ which parametrizes the elliptical orbit. (In the situation of a circular orbit, d_{orbit} was a constant). In this case the orbital contraint (ii) might be realized using the *Vis-Viva-Equation* and (iii) might be realized by comparing $x'(T)$ with $\frac{d}{d\theta} d_{\text{orbit}}(\theta_T)$, the latter of which is tangential to the orbital path. Here θ_T is the angle of the rocket towards the planet at time T which would have to be calculated by transforming $x_1(T)$ to polar coordinates. During the implementation it became clear that it might be more sensible to address the problem of non-circular orbits by formulating the dynamics in polar coordinates instead.

Summary

In the first section I, we derived a set of equations that govern the motion of two massive bodies from *Newton's* law of universal gravitaion. These equations form a system of differential equations.

In section II we introduced the two bodies. The first would be our actuated rocket and the second would be the planet to be orbited. We modified the dynamics of motion to give us control of the rocket.

Next, section III discussed sufficient conditions for the rocket to be in orbit of the planet. These are conditions that the rocket would have to satisfy by the end of the orbital maneuver.

With all this information, in section IV, we could formulate an optimal control problem to which the solution would yield controls for a successful orbital maneuver.

To solve this problem, section V dealt with discretizing and implementing the previous equations and functions.

Finally, we present solutions to two example problems in section VI.

In section VII, we concluded the report with a brief discussion on further things that could have been considered in the modelling and formulation of the problem.

References

[1] S. Gros and M. Diehl, "Numerical Optimal Control (draft manuscript)," https://www.syscop.de/files/2020ss/NOC/book-NOCSE.pdf, April 27, 2022.

[2] J. A. E. Andersson, J. Gillis, G. Horn, J. B. Rawlings, and M. Diehl, "CasADi — A software framework for nonlinear optimization and optimal control," Mathematical Programming Computation, 11(1):1–36, 2019.

[3] A. Wächter and L. T. Biegler, "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming," Mathematical Programming, 106(1):26–57, 2006.