Modeling and Simulation of Aerospace Systems

Drag Free Control of GOCE Spacecraft

Group 22:
André Oliveira - 10670413
Ernest Sebastian Constantine - 10607280
Gonçalo Eduardo Cascalho Raposo - 10670412

Contents

1 The real system

This report describes the modeling and simulation of the Drag Free and Attitude Control System (DFACS) that was used on the Gravity Field and Steady-State Ocean Circulation Explorer (GOCE) satellite.

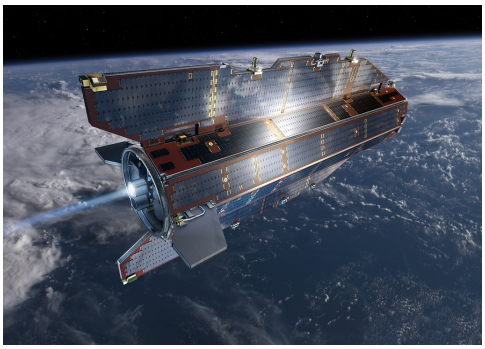

Figure 1: Artist's view of GOCE

The spacecraft was designed with the scientific purpose of determining the Earth's steady state gravity field anomalies with an accuracy of $1 \times 10^{-5} m/s^2$, and determining the geoid height with an accuracy of between 1 to 2 cm, at length scales down to 100 km [1]. To achieve this, the spacecraft carried a highly sensitive gravity gradiometer consisting of three accelerometers which measured gravitational gradient along three orthogonal axes.

The type of mission carried out by the GOCE satellite required that it had to fly on an orbit very close to the Earth's surface in a Sun-synchronous orbit. The spacecraft orbited at a comparatively low altitude of 254.9 km. Therefore, the atmospheric drag acting on the spacecraft could not be neglected as it led to faster orbit decay. It was shaped like an arrow to minimize the drag and stabilize the spacecraft as it orbited through the residual air in the thermosphere. It also carried an ion propulsion system which continuously compensated the air drag without the vibration of conventional chemical propulsion, thus reducing the error in the highly sensitive gravity gradiometer sensor. The ion propulsion system used xenon as the fuel, which is stored in the fuel tank with a capacity of 40 kg [2]. It ejected xenon ions at velocities exceeding 40000 m/s.

The goal of the DFACS is to mitigate the effect of the atmospheric drag and other perturbing accelerations on the decay of the orbit, by measuring these disturbances and compensating them using a controllable propulsion system (ion thruster). The system is designed to maintain the spacecraft at a desired orbital height.

The DFACS consists of three sub-systems that require modeling, which are the accelerometer, the flow control valve, and the ion thruster. The accelerometer measures the acting accelerations and outputs a voltage signal. This signal will be processed by a control system that will then control the produced thrust by means of a flow control valve. The valve changes the mass flow rate of the flowing Xenon, thus affecting the thrust. When the thrust equals the drag, the accelerometer should measure no acceleration, corresponding to the desired equilibrium.

2 The physical model

As mentioned in section 1, the real system is made up of different sub-systems. The definition of the physical model will be done for each part separately for simplicity. After arriving at the physical model, it is then needed to see how the different parts interact with each other, which is shown in the flow chart in section 5.

2.1 Orbit

To study the orbit of the spacecraft, it was considered a two body and central-force problem [3]. A two body analysis is a very good approximation, since other surrounding bodies (i.e. sun, planets, ...) are much farther, which results in weaker forces and, therefore, those interactions can be neglected. The central-force analysis is also a valid approximation, because since the Earth's mass is much greater than the spacecraft's mass, then the center of mass of the system will be very close to the center of mass of the Earth and, therefore, one can assume that the Earth is fixed in space and the center of this problem.

As is well known, the orbit is ruled by the gravitational interaction between the Earth and the spacecraft. The solution of this central force problem is a conic section. In this case, the conical will be an ellipse very close to a circle.

Besides the two body and central-force approximations, others were made. Firstly, the Earth was considered as an ellipsoid, instead of its true shape, resulting only in the J2 perturbation [4], whereas the true shape would result in a much more complex gravity potential[3]. Another assumption was

describing the atmosphere by the same atmospheric model [4] (time-invariant) all around, which for a polar orbit is an even more far-fetched approximation (the atmosphere is thinner at the poles).

As for perturbations of the orbit, it was only considered the effects from J_2, the atmospheric drag and the thrust of the spacecraft.

Despite the considered model for the orbit and its approximations, it doesn't affect the study of this system in the way of determining whether the spacecraft's thrust is able to follow the drag or not.

2.2 Accelerometer

The considered MEMS accelerometer is capacitive and measures the acceleration of a charged inertial mass, which can be used to compute the forces acting on the spacecraft. This mass is electrostatically suspended between two charged plates so that as the mass moves, the capacitance changes [5]. To arrive at a simple physical model of this effect, we reduce it to two ideal parallel plate capacitors in series, in which one of the plates of each capacitor is the inertial mass. Doing this, the resistance of the real capacitors is being neglected. Also, the impedance of the wires is neglected since the impedance of the capacitors is many times bigger.

When the accelerometer mass moves away from its equilibrium position, it is brought back to the center position by means of an electrical force, due to its movement and the control voltage applied to the electrodes [6]. Furthermore, the mass displacement must translate to an output signal that will then control the propulsion system to compensate for the drag. This control subsystem can be modeled by connecting the inertial mass to one of the inputs of an operational amplifier while feedbacking the output through a capacitor back into the input, as well as into the plates of the capacitors [5]. To make sure that the mass is stable, that is, that the electrostatic force acting on the mass is leading the mass to equilibrium, a voltage bias on the plates must also be applied [6]. When making these assumptions, we assume an ideal amplifier, that is, the gain is infinite, the input impedance is infinite, the output impedance is zero and all dissipative, limited bandwidth, noise, and input capacitive effects are ignored [7]. Also, the impedance of the circuitry connecting the different elements is neglected.

Several models of the accelerometer subsystem were considered. First, the output of the amplifier was fed into a PD controller before being applied to the electrodes of the capacitors. However, the derivative part led to errors due to the simulation being run in discrete time since the controller is non-causal, that is, the output depends on future values of the input. Therefore, another model is considered in which the controller is a simple proportional gain. Also, a spring and a damper were added to the inertial mass. This was done in order to model the friction on the walls of the enclosed mass and improve the results of the numerical simulation. The final accelerometer physical model can be seen in Fig. 2.

2.3 Flow Control Valve

The flow control valve has the function of controlling the flow of the xenon gas from the xenon tank to the ion thruster, which produces the thrust. It consists of a spool, acting as a valve, which can be closed or opened with a certain aperture, connected to a linear actuator. The linear actuator consists of a spring, a solenoid, an armature, and a proportional integral (PI) controller. The armature is physically linked with the spool so their displacement is the same. The spool has the shape of a choked nozzle. The pressure at the inlet of the spool is the pressure of the tank, and the pressure at the outlet is near zero, due to the vacuum of space. This condition makes sure that the flow at the minimum section is always chocked with a Mach number equal to one.

The spring has a function of closing the valve when there is no electrical current. The solenoid produces a magnetic field when there is an electrical current. This magnetic field produces an attractive force between the opposing stator and armature pole pieces. The armature will then move to close the air gap and minimize the reluctance of the magnetic circuit. An ideal linear DC motor will be considered, such that force generated is proportional to the current.

The PI controller receives the output voltage of the accelerometer, which corresponds to the acceleration of the satellite. The integrative gain assures that the steady-state difference goes to zero. The controller outputs a current which will control the force applied on the spool, affecting its position. The position of the spool controls the area of the choked section of the nozzle, which has a Mach number equal to one. Thus, the position of the spool directly determines the mass flow rate of the xenon gas.

The shape of the valve is considered to be rectangular so that the position of the spool directly correlates with the sectional area of the valve. The U shape of the end of the spool is modified so that the pressure applied on the spool is equal either the valve is opened or closed. To compute the pressure force acting on it, the pressure was assumed to be constant in this entire surface. When the spool is

fully closed, the applied pressure is the tank's pressure, while if it is opened, the pressure is the throat pressure at Mach 1.

The fluid properties in the chocked nozzle (pressure, density, temperature, and speed of sound) can be found using the assumption of ideal gas and isentropic flow. With the Mach number known to be one and the area of the chocked nozzle known from the spool position, the mass flow rate of the xenon gas can be computed. These assumptions result in figure 3.

2.4 Ion Thruster

The ion thruster has the function of generating the thrust required to compensate the aerodynamic drag of the satellite, which will maintain the orbital altitude of the satellite around the desired value. It ionizes the xenon flow and accelerates it at high speed using a strong magnetic field, producing thrust. The assumption made is that the high voltage for the acceleration grid is always available, the efficiency of the ionization process is 100 %, and the beam of ions is unidirectional, singly ionized, and monoenergetic. The thrust depends on the mass flow rate of the xenon, acceleration grid voltage, and the properties of the xenon ion (mass and electron charge). A simple model is illustrated in figure 4.

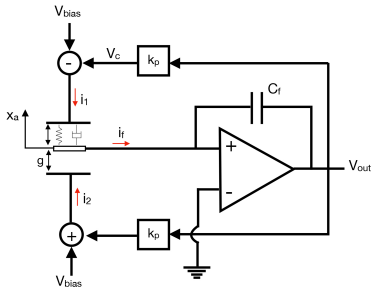

Figure 2: Physical model of the accelerometer.

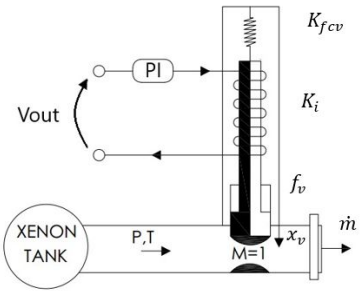

Figure 3: Physical model of the flow control valve.

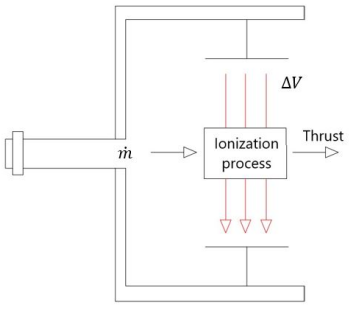

Figure 4: Physical model of the thruster.

3 The mathematical model

3.1 Orbit

As requested, Gauss Planetary Equations were used to model the variation in time of the orbital parameters. These equations can be derived from doing a simple force and torque balance. The acting forces would be the gravitational force, calculated through Newton's law of universal gravitation, and any exterior forces (e.g. perturbations). Gauss Planetary Equations can be represented as in Eq. (1) [8]:

$$\begin{cases} \frac{\mathrm{d}a}{\mathrm{d}t} = \frac{2a^2 v}{\mu} a_t \\ \frac{\mathrm{d}e}{\mathrm{d}t} = \frac{1}{v}\left[2(e+\cos(f))a_t - \frac{r}{a}\sin(f)a_n\right] \\ \frac{\mathrm{d}i}{\mathrm{d}t} = \frac{r\cos(\omega+f)}{h} a_h \\ \frac{\mathrm{d}\Omega}{\mathrm{d}t} = \frac{r\sin(\omega+f)}{h\sin(i)} a_h \\ \frac{\mathrm{d}\omega}{\mathrm{d}t} = \frac{1}{ev}\left[2\sin(f)a_t + \left(2e + \frac{r}{a}\cos(f)\right)a_n\right] - \frac{r\sin(\omega+f)\cos(i)}{h\sin(i)} a_h \\ \frac{\mathrm{d}f}{\mathrm{d}t} = \frac{h}{r^2} - \frac{1}{ev}\left[2\sin(f)a_t + \left(2e + \frac{r}{a}\cos(f)\right)a_n\right] \end{cases} \tag{1}$$

where the states are: a as semi-major axis, e as the eccentricity, i as inclination, Ω as longitude of ascending node, ω as argument of periapsis, and f as true anomaly - and the parameters are: v as the orbital velocity, μ as the gravitational parameter, r as the distance between the two bodies, h as the specific angular momentum, and a_t, a_n and a_h as the components of the "external acceleration".

The parameters used in Eq. (1) can be calculated as in Eq. (2) [3]:

$$v = \sqrt{\mu\left(\frac{2}{r} - \frac{1}{a}\right)} \qquad r = \frac{a(1-e^2)}{1+e\cos(f)} \qquad h = \sqrt{\mu a(1-e^2)} \tag{2}$$

The "external acceleration" is the acceleration caused by the external forces in a reference frame rotated by the matrix in Eq. (3) [4]. In this case, the external forces are the J2 perturbation, the drag, and the thrust.

$$\left[\hat{\mathbf{t}}, \hat{\mathbf{n}}, \hat{\mathbf{h}}\right] = \left[\frac{\mathbf{v}}{||\mathbf{v}||}, \hat{\mathbf{h}} \times \hat{\mathbf{t}}, \frac{\mathbf{r} \times \mathbf{v}}{||\mathbf{r} \times \mathbf{v}||}\right] \qquad \mathbf{a}_{\text{tnh}} = \left[\hat{\mathbf{t}}, \hat{\mathbf{n}}, \hat{\mathbf{h}}\right]^T \mathbf{a}_{\text{car}} \tag{3}$$

However, to construct this matrix it's necessary to have the vectors for the velocity and angular momentum of the spacecraft. One can obtain those vectors in the Earth-centered inertial (ECI) frame (same as the Cartesian frame) by calculating them in the orbital plane and then rotating it accordingly to ω, i and Ω [3], as represented in Eq. (5).

$$R_x(\theta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\theta) & -\sin\theta \\ 0 & \sin\theta & \cos\theta] \end{bmatrix} \qquad R_z(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1] \end{bmatrix} \tag{4}$$

$$\begin{aligned} \mathbf{r}^k &= [r\cos f, r\sin f, 0]^T & \mathbf{v}^k &= \left[-\frac{\mu}{h}\sin f, \frac{\mu}{h}(e+\cos f), 0\right]^T \\ \mathbf{r}^c &= R_z(\Omega)R_x(i)R_z(\omega)\mathbf{r}^k & \mathbf{v}^c &= R_z(\Omega)R_x(i)R_z(\omega)\mathbf{v}^k \end{aligned} \tag{5}$$

The J2 perturbation causes a force that can be derived from the gradient of its potential [3]. In the ECI frame, the acceleration can be calculated as in Eq. (6) [4]:

$$\begin{cases} a_x^{J2} = x\frac{\mu}{r^3}\left[\frac{3}{2}J_2\left(\frac{R_{eq}}{r}\right)^2\left(5\left(\frac{z}{r}\right)^2 - 1\right)\right] \\ a_y^{J2} = y\frac{\mu}{r^3}\left[\frac{3}{2}J_2\left(\frac{R_{eq}}{r}\right)^2\left(5\left(\frac{z}{r}\right)^2 - 1\right)\right] \\ a_z^{J2} = z\frac{\mu}{r^3}\left[\frac{3}{2}J_2\left(\frac{R_{eq}}{r}\right)^2\left(5\left(\frac{z}{r}\right)^2 - 3\right)\right] \end{cases} \tag{6}$$

This vector has then to be rotated to the *tnh* frame accordingly to Eq. (3).

The acceleration caused by the thrust is simply given by equation Eq. (7). The force caused by drag is calculated by Eq. (8) and the resulting acceleration by Eq. (9).

$$a_t^T = T/m \tag{7} \qquad D = \frac{1}{2}\rho v^2 C_D A \tag{8} \qquad a_t^D = -D/m \tag{9}$$

where the drag coefficient can be calculated from the ballistic coefficient $C_D = \frac{m}{A\beta}$.

The air density ρ depends on the altitude H. In the simulation, H was calculated with the Earth being an ellipsoid, Eq. (10), and calculating the closest point in that surface whose normal intersects the coordinates of the spacecraft. This can be achieved by solving Eq. (11). These two equations can be gathered in Eq. (13) and the system was solved with Newton's method.

$$F(x,y,z) = \left(\frac{x}{R_a}\right)^2 + \left(\frac{y}{R_a}\right)^2 + \left(\frac{z}{R_b}\right)^2 - 1 = 0 \tag{10}$$

$$(u,v,w) - (x,y,z) = \lambda \cdot \nabla F(x,y,z), \qquad \lambda \in \mathbb{R} \tag{11}$$

$$H = \|(u,v,w) - (x,y,z)\| \tag{12}$$

$$\begin{cases} x\left(1 + \frac{\lambda}{R_a^2}\right) - u = 0 \\ y\left(1 + \frac{\lambda}{R_a^2}\right) - v = 0 \\ z\left(1 + \frac{\lambda}{R_b^2}\right) - w = 0 \\ \left(\frac{x}{R_a}\right)^2 + \left(\frac{y}{R_a}\right)^2 + \left(\frac{z}{R_b}\right)^2 - 1 = 0 \end{cases} \tag{13}$$

where R_a is the Earth's equatorial radius and R_b the polar radius; (u,v,w) is the spacecraft coordinates and (x,y,z) is the coordinates of the point in the Earth's surface; and H is the altitude of the spacecraft.

Analyzing the state space model, Eq. (1), it's clear that it isn't defined when a, e, i, v, r, h and μ are zero, which is relevant, since the model can't be used in those cases. Although is unlikely or even impossible to have some of those values equal to zero, some are more common, for example, in the case of a circular orbit ($e = 0$) or an equatorial orbit ($i = 0$). Given that, in this problem, the orbit is neither circular or equatorial, this model can be used.

3.2 Accelerometer

When writing the mathematical model of the accelerometer, we need to arrive at an expression for V_{out}, since this voltage is carried out to the next subsystem. In order to do this, one must consider the characteristic equations of the parallel plates capacitor and the ideal operational amplifier written in Eq. (14) and Eq. (15) [7].

$$C = \frac{\epsilon A}{d}, \qquad i_C = C\frac{dV_C}{dt} \qquad (14) \qquad i_- = i_+ = 0A, \qquad V_- = V_+ \qquad (15)$$

In Eq. (14), A is the area of the plates, d the distance between plates and ϵ is the absolute permittivity of the medium between the plates. i_C and V_C are the values of current and voltage at the terminals of the capacitor with capacitance C and in Eq. (15), i_- and i_+ are the currents at the plus and minus terminals of the amplifier. The same goes for the voltages V_- and V_+. In this case, since V_- is connected to the ground, $V_- = V_+ = 0$.

The charged mass and two plates can be seen as two capacitors, as said before. According to the defined axis for the mass position x_a, the capacitances are defined in Eq. (16).

Then, to derive the differential and algebraic equations, we will consider the circuit described in the physical system. Taking into account that $i = \frac{dQ}{dt}$ [7], since the charge in the plates of a capacitor is given by $Q = CV$ and C_1 and C_2 are not constant, the current flowing into/from a capacitor is not given by Eq. (14), but instead by $i = \frac{dC}{dt}V + C\frac{dV}{dt}$.

However, since the capacitance C_f does not change, we can use Eq. (14) to write the current in the feedback capacitor. Knowing this, one can write the currents i_1, i_2 and i_f given in Eq. (17).

$$C_1 = \frac{\epsilon A}{g - x_a}, \qquad C_2 = \frac{\epsilon A}{g + x_a} \qquad (16) \qquad \begin{cases} i_1 = C_1\frac{\dot{x_a}}{g-x_a}(V_c - V_{bias}) + C_1\frac{dV_C}{dt} \\ i_2 = -C_2\frac{\dot{x_a}}{g+x_a}(V_c + V_{bias}) + C_2\frac{dV_C}{dt} \\ i_f = -C_f\frac{dV_{out}}{dt} \end{cases} \qquad (17)$$

Finally, we note that the proportional gain makes it so that $V_c = k_p V_{out}$. Applying the conservation of charge to the node linking all three capacitors, we get that $i_1 + i_2 = i_f$. These expressions, together with the ones derived in Eq. (17), give the 1st order differential equation for V_{out}, already in state space form, in Eq. (18).

$$\frac{dV_{out}}{dt} = \frac{\dot{x}_a}{1 + k_p\frac{C_1+C_2}{C_f}}\left[\frac{C_2}{C_f}\frac{V_c + V_{bias}}{g + x_a} - \frac{C_1}{C_f}\frac{V_c - V_{bias}}{g - x_a}\right] \qquad (18)$$

To arrive at the equation of motion of the charged mass it is required to arrive at an expression of the electrical forces acting on the mass. The energy stored on the electrical field of a capacitor is given by $U = \frac{1}{2}CV^2$ [7], with the corresponding values of C and V. Since the electric field can be considered conservative in this application, the force can be given by Eq. (19) [9].

$$F = -\frac{\partial U}{\partial x_a} = -\frac{1}{2}\frac{\partial C}{\partial x_a}V^2 - CV\frac{\partial V}{\partial x_a} \qquad (19)$$

As seen in Eq. (19), given the expression for the energy, to compute the forces we need to compute $\frac{\partial V_c}{\partial x_a}$. Using the proportional gain relation written above and noting that applying the chain rule we have Eq. (20), $\frac{dV_c}{dx_a}$ is written in Eq. (21).

$$\frac{dV_{out}}{dx_a} = \frac{dV_{out}}{dt}\frac{dt}{dx_a} = \frac{dV_{out}}{dt}\frac{1}{\dot{x_a}} \qquad (20)$$

$$\frac{dV_c}{dx_a} = \frac{k_p}{1 + k_p\frac{C_1+C_2}{C_f}}\left[\frac{C_2}{C_f}\frac{V_c + V_{bias}}{g + x_a} - \frac{C_1}{C_f}\frac{V_c - V_{bias}}{g - x_a}\right] \qquad (21)$$

Then, computing the derivative in Eq. (19) for each of the capacitors to get both electric forces acting on the mass, the corresponding F_1 and F_2 are given in Eq. (22).

$$\begin{cases} F_1 = -C_1(V_c - V_{bias})\left[\frac{1}{2}\frac{V_c - V_{bias}}{g - x_a} + \frac{k_p}{1+k_p\frac{C_1+C_2}{C_f}}\left[\frac{C_2}{C_f}\frac{V_c+V_{bias}}{g+x_a} - \frac{C_1}{C_f}\frac{V_c - V_{bias}}{g - x_a}\right]\right] \\ F_2 = -C_2(V_c + V_{bias})\left[-\frac{1}{2}\frac{V_c + V_{bias}}{g + x_a} + \frac{k_p}{1+k_p\frac{C_1+C_2}{C_f}}\left[\frac{C_2}{C_f}\frac{V_c+V_{bias}}{g+x_a} - \frac{C_1}{C_f}\frac{V_c - V_{bias}}{g - x_a}\right]\right] \end{cases} \qquad (22)$$

When applying Newton's 2nd Law to the seismic mass, it is needed to take into account the fact that the mass is suspended inside of the spacecraft, and therefore the acceleration suffered by the spacecraft (due to the modeled drag and thrust) has to be taken into account since the frame centered on the mass isn't inertial. Finally, using Hooke's Law to model the force produced by the spring with constant k, and a linear damping with constant c for the damper, the acceleration of the mass is written, already in state space form, in Eq. (23). Together with Eq. (18), these make the ODEs for this subsystem, where the states are x_a, v_a and V_{out}. Obviously, x_a is constricted to the available gap between $x_a = g$ and $x_a = -g$.

$$\begin{cases} \dot{x}_a = v_a \\ \dot{v}_a = \dfrac{D-T}{m_{\mathrm{SC}}} + \dfrac{F1+F2-kx_a-cv_a}{m} \end{cases} \tag{23}$$

3.3 Flow Control Valve

In writing the mathematical model of the flow control valve, it is more convenient to divide it into three parts. The first part is the proportional integral (PI) system, the second part is the equation of motion for the spool, and the third part is the calculation of the xenon mass flow rate.

The PI system receives the input V_{out} from the accelerometer and outputs the current to the linear actuator. The output of the PI controller is the sum of one term proportional to V_{out} and another proportional to its integral (V_i), with the respective gains as shown in Eq. (24) and Eq. (25).

$$i = K_{p_v} V_{out} + K_{iv} V_i \tag{24}$$

$$\dot{V}_i = V_{out} \tag{25}$$

The equation of motion for the spool is based on a mass-spring-damper system with the linear actuator as an external force. The force generated by the actuator is proportional to the current $F_{actuator} = K_i i$.

$$\dot{x}_v = v_v \tag{26}$$

$$\dot{v}_v = \frac{1}{m}(K_i i - cv - K_{f_{cv}} x_v - F_0 + F_{\mathrm{press}}) \tag{27}$$

where m is the mass of the spool, K_i is the actuator constant, c is the valve friction coefficient, $K_{f_{cv}}$ is the spring coefficient, F_0 is the preload, and F_{press} is the force due to the pressure at the section. As mentioned, this force was assumed to be the same either the valve was closed (tank pressure) or opened (flow pressure).

Due to the shape of the section of the spool, the throat area A^* can be computed as in Eq. (28). To compute the mass flow rate of the xenon, one needs to know the properties at the throat, which can be calculated as in Eq. (29).

$$A^* = x_v \cdot w \tag{28}$$

$$\begin{cases} T^* = \frac{2T_0}{\gamma+1} \\ p^* = p_0(\frac{\gamma+1}{2})^{\frac{\gamma}{1-\gamma}} \\ \rho^* = \rho_0(\frac{\gamma+1}{2})^{\frac{1}{1-\gamma}} \end{cases} \tag{29}$$

where w is the width of the section. T^*, p^*, and ρ^* are the temperature, pressure, and density in the throat, respectively, and T_0, p_0, and ρ_0 are properties of the tank [10].

The differential equations used in this part of the model were Eq. (25), Eq. (26) and Eq. (27). Notice that the displacement x_v has the physical constraints of the valve orifice dimensions, that is, a minimum value of 0 and a maximum value corresponding to the height of the orifice when completely opened.

3.4 Ion Thruster

The mathematical model of the Ion Thruster is based on Newton's 2nd law. The ions are accelerated to a high exhaust velocity using an electrical power source. The produced force can be computed as follows

$$T = \frac{dm_p}{dt} v_{ex} = \dot{m} v_i \tag{30}$$

$$v_i = \sqrt{\frac{2qV_b}{M}} \tag{31}$$

,where $\dot{m}$ is the ion mass flow rate and v_i is the ion velocity. By conservation of energy, the ion exhaust velocity is given by Eq. (31), where V_b is the net voltage through which the ions are accelerated, q is the charge, and M is the ions' mass.

The mass flow rate of the xenon can be computed with Eq. (32). Since the Mach number in the throat is equal to one, the velocity of the fluid equals the speed of sound - Eq. (33).

$$\dot{m} = \rho^* A^* v_f \tag{32}$$

$$v_f = \sqrt{\gamma R T^*} \tag{33}$$

where R is the gas constant (63.33 J/KgK for Xenon).

The ion thruster model has no differential equations. However, the algebraic equations are used to calculate the generated thrust.

3.5 Linearization

Since the orbital part of the dynamics is oscillatory, instead of around an equilibrium point, the linearization was done for a nominal condition, chosen in this case as the final value of the states after running the simulation for one hour. It is important to note that the derivatives in the Jacobian matrix used in the linearization were computed using finite differences, because the right-hand side functions cannot

be written in closed form since, throughout the simulation, it uses logical conditions, an optimization method and the evaluation of a look-up table (to compute the drag).

Both the nominal condition states and the obtained eigenvalues are shown below in Eq. (34), Table 1 and Fig. 5.

$$\begin{gathered} a_n \simeq 6648.37\,\text{km} \qquad e_0 \simeq 0.0051 \qquad i_0 = \pi/2\,\text{rad} \qquad \Omega_0 = 0\,\text{rad} \qquad \omega_0 \simeq 0.179\,\text{rad} \qquad f \simeq 3.996\,\text{rad} \\ x_{a\,0} \simeq 5.9 \times 10^{-8}\,\text{m} \qquad v_{a\,0} \simeq 3.9 \times 10^{-10}\,\text{m/s} \qquad V_{out\,0} \simeq 0.0011\,\text{V} \\ V_{i\,0} \simeq 11.20\,\text{Vs} \qquad x_{v\,0} \simeq 9.6 \times 10^{-9}\,\text{m} \qquad v_{v\,0} \simeq 9.8 \times 10^{-13}\,\text{m/s} \end{gathered} \tag{34}$$

Table 1: Eigenvalues of the linearized model around a nominal condition

-75.000000318050 + 152.15022370272i
-75.000000318050 - 152.15022370272i
-0.3067541702889 + 0.4599271552618i
-0.3067541702889 - 0.4599271552618i
-0.0114910238173 + 0.0000000000000i
-0.0000348815918 + 0.0007046643534i
-0.0000348815918 - 0.0007046643534i
-0.0000030195781 + 0.0000759078086i
-0.0000030195781 - 0.0000759078086i
0.0000015151369 + 0.0000000000000i
-0.0000000000000 + 0.0000000000000i
-0.0000000001969 + 0.0000000000000i

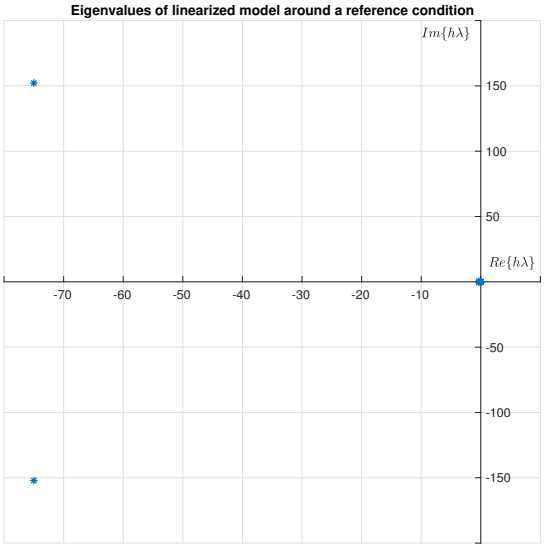

Figure 5: Plot of linearization eigenvalues

As expected, we are dealing with a stiff system, since the order of magnitude of the real part of the eigenvalues spans across several values, from 10^{-6} to 10^{2}.

We can see that the magnitude of two of the eigenvalues is close to zero, which correspond to integrator states in the ODEs.

The one eigenvalue with a positive real part is due to the secular effect introduced by the $J2$ perturbation, which is not compensated since there is no control on the orbit, apart from negating the drag. Regarding the system, these will only affect the orbital parameters, rotating the orbit in space.

4 Numerical integration

Analyzing the poles of the linearized system, it's clear that the system at study is stiff. Therefore, a stiff ODE solver should be chosen. As explained in Matlab's documentation: "Try `ode15s` when `ode45` fails or is inefficient and you suspect that the problem is stiff. Also use `ode15s` when solving differential algebraic equations (DAEs)."[11] Since the system at study is stiff and consists of a set of DAEs, the first solver that was used in this problem was `ode15s`, which gave good results.

`ode15s` is a variable-step, variable-order (VSVO) solver based on the numerical differentiation formulas (NDFs) of orders 1 to 5 [11]. NDF is a multistep implicit method and can be seen as a modification of backward differentiation formula (BDF). The algorithm can be written as [12]:

$$\begin{gathered} y' = f(t,y) \\ y(t_0) = y_0 \end{gathered} \qquad \begin{gathered} t_n := t_0 + n \cdot h \\ f_n := f(t_n, y_n) \\ n = 0, 1, ..., \end{gathered} \qquad \sum_{j=0}^{k} \alpha_j y_{n+j-k+1} = h \sum_{j=0}^{k} \beta_j f_{n+j-k+1} \tag{35}$$

Table 2: Coefficients of the NDF methods. k represents the number of steps, α_k and β_k the coefficients in Eq. (35) and p is the order of the method.

k	α_0	α_1	α_2	α_3	α_4	α_5	β_k	p
1	-1	1					1	1
2	$\frac{1}{3}$	$-\frac{4}{3}$	1				$\frac{2}{3}$	2
3	$-\frac{2}{11}$	$\frac{9}{11}$	$-\frac{18}{11}$	1			$\frac{6}{11}$	3
4	$\frac{3}{25}$	$-\frac{16}{25}$	$\frac{36}{25}$	$-\frac{48}{25}$	1		$\frac{12}{25}$	4
5	$-\frac{12}{137}$	$\frac{75}{137}$	$-\frac{200}{137}$	$\frac{300}{137}$	$-\frac{300}{137}$	1	$\frac{60}{137}$	5

In the simulation, `ode15s` was kept with variable step and variable order, and therefore, it would use a more advanced algorithm compared to the one in Eq. (35). In this way, one can grant a more efficient solve, since it can "relax" or "'thighten" the solver, however it's suitable, this is, the tolerance allowed. In this case, the absolute tolerance was set to 10^{-8} and the relative tolerance to 10^{-6}, which grants at least 6 significant digits or 8 decimal places (in Matlab, tolerance is calculated as `|e(i)| <= max(RelTol*abs(y(i)),AbsTol(i))`[11]).

The stability domain of multistep methods can be calculated from Eq. (36) [13]. Its boundary can be calculated if one traces the solution of Eq. (36) for $z = e^{i\theta}$, $\theta \in [0, 2\pi)$. The stable regions can be distinguished from the unstable regions by testing if the roots of the stability polynomial for $h\lambda$, Eq. (37), check: $|(h\lambda)_k| < 1$ [13].

$$h\lambda = \frac{\sum_{j=0}^{k} \alpha_j z^j}{\sum_{j=0}^{k} \beta_j z^j} \quad , \quad z \in \mathbb{C} : |z| \leq 1 \tag{36}$$

$$\sum_{j=0}^{k} (\alpha_j - h\lambda\beta_j)\, z^j \tag{37}$$

This analysis was done for NDF methods of order 1 to 5 and can be seen in Fig. 6. As one can see, the methods are implicit and, therefore, work well with stiff systems, since almost all of the entire left half-plane is modeled as stable.

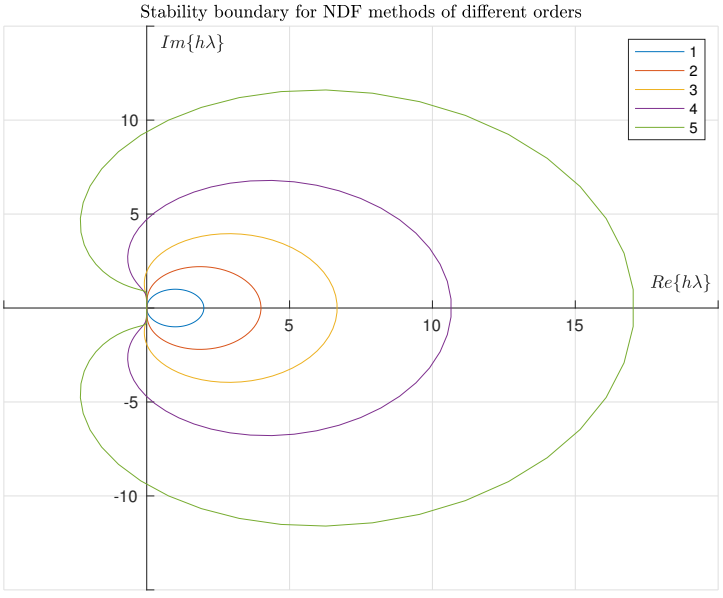

Figure 6: Stability boundaries for NDF methods of order 1 to 5. The region inside the curves is unstable and the outer region is stable.

5 The simulation framework

Taking into advantage the fact that the system is made up of several subsystems connected only by input/output variables, each iteration of the simulation is also run in parts consisting of the same subsystems, as shown in the flow chart below.

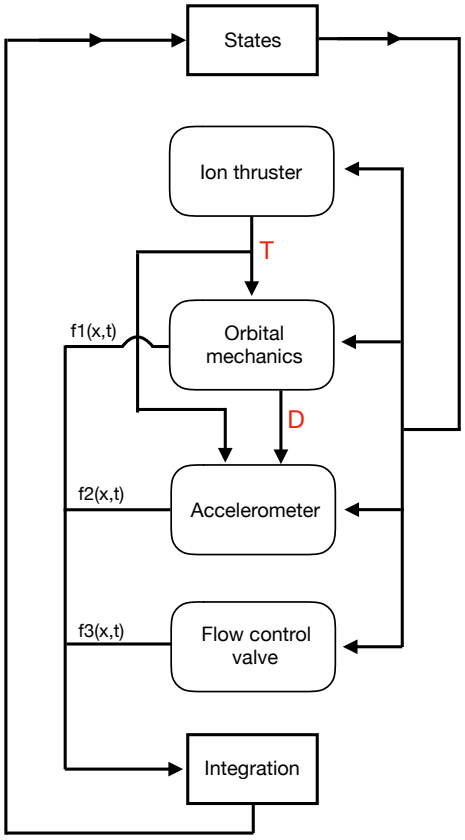

Figure 7: Flow chart of the implemented simulation

Fig. 7 describes what happens in each iteration of the simulation. The input states to each block-/subsystem were not written explicitly in order to make the figure less complex. Here follows a more detailed description:

- The ion thruster block receives as input the position of the spool and computes the provided thrust.
- The orbital mechanics and accelerometer blocks receive as input the respective state variables as well as the variables written in red, in the figure (namely, the drag and thrust).
- The flow control valve, apart from the respective state variables, also receives as input the state V_{out}.
- The three main subsystems then give as output the right-hand sides of the corresponding differential equations, denoted as $f_i(x,t)$ in Fig. 7.

6 System response

In this report, we are modeling and simulating the DFACS system in nominal and off-nominal conditions. The nominal condition is when the spacecraft is maintaining an altitude of 254.9 km while all systems are operating normally. The off-nominal condition is when some of the subsystems are not operating normally. In specific, we considered a malfunction in the accelerometer and in the flow control valve during a fixed time interval.

6.1 Normal conditions

A simulation was run for 5 orbital periods, which corresponds to, approximately, 15 hours. The initial values for the states were:

$$\begin{aligned} a_0 \simeq 6663.044\,\text{km} \qquad e_0 = 0.0045 \qquad i_0 = \pi/2\,\text{rad} \qquad \Omega_0 = 0\,\text{rad} \qquad \omega_0 = 0\,\text{rad} \qquad f = 0\,\text{rad} \\ x_{a\,0} = 0\,\text{m} \qquad v_{a\,0} = 0\,\text{m/s} \qquad V_{out\,0} = 0\,\text{V} \\ V_{i\,0} = 0\,\text{Vs} \qquad x_{v\,0} = 0\,\text{m} \qquad v_{v\,0} = 0\,\text{m/s} \end{aligned} \tag{38}$$

where the lines correspond to the orbit, accelerometer and flow control valve parameters, respectively. These initial parameters are equivalent to the spacecraft starting the orbit at the periapsis, which is over the equator, and moving North. The accelerometer has it's mass centered and, therefore, isn't measuring any acceleration. The flow control valve is closed and, therefore, the spacecraft isn't thrusting.

The parameters were mostly defined accordingly to the table given in the project guidelines. Some of the optimizable parameters were changed and are presented in Eq. (39) (accelerometer parameters) and Eq. (40) (flow check valve parameters).

$$k_{p_a} = 1, \qquad k_{d_a} = 0\,\text{s}, \qquad c = 0.2\,\text{Ns/m}, \qquad k = 0.1\,\text{N/m} \tag{39}$$

$$K_{p_v} = 10^{-4}\,\text{A/V}, \qquad K_{i_v} = 3 \times 10^{-3}\,\text{A/Vs} \qquad K_i = 2 \times 10^{-3}\,\text{N/A} \tag{40}$$

Some important results regarding the orbit are shown in Fig. 8.

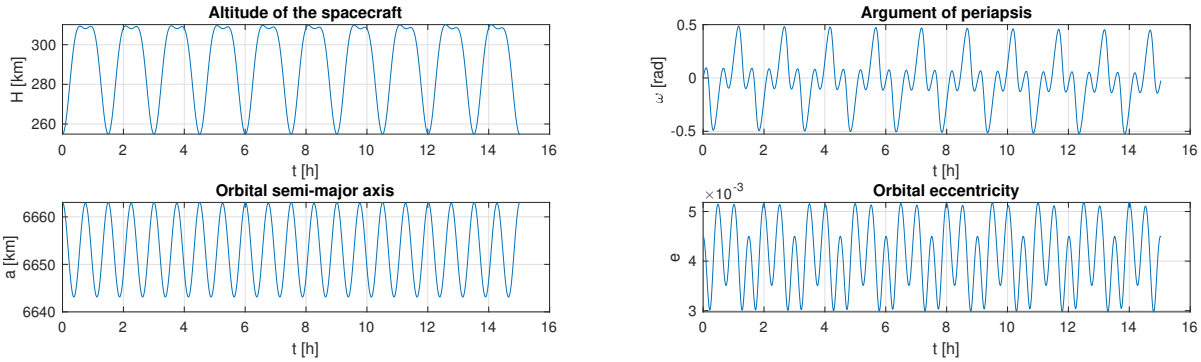

Figure 8: Orbit related plots: altitude, argument of periapsis, semi-major axis and eccentricity.

As mentioned, the spacecraft starts at the periapsis and, therefore, the initial altitude is a minimum. Analyzing the altitude plot, the minimums would correspond to passing over the equator and the maximums to passing over the poles. However, the mean value of the argument of periapsis is slowly decreasing (secular effect) and, after some time, the previous correspondence wouldn't be true. The semi-major axis and eccentricity oscillate over time, but the mean value seems to be stable, which means that the orbit doesn't decay. The oscillation in the semi-major axis is due to the small delay of the thrust following the drag. A quicker controller would result in oscillations with smaller amplitude.

The relevant results concerning the accelerometer subsystem are represented in Fig. 9. In this figure, the plots on each line correspond to the same variable in different but continuous time spans. The plots were divided in time in order to highlight the initial transient.

One can see from the plots that the accelerometer behaves in exactly the way it is expected to. Initially, given that the simulation starts with $T = 0$, the seismic mass is pushed up in the x_a axis due to the acceleration caused by drag. Then, V_{out} measures the mass displacement and sends this information to the next subsystem which will increase the thrust. It's also possible to observe that, as intended, the electrical force acting on the mass is re-balancing, that is, it draws the mass back to equilibrium. This can be seen on the plots, on the right, of the electrical force and seismic mass position since they are always opposite in sign.

From the graphs on the right one can conclude that after the initial transient, the thrust follows the drag almost exactly so that the effects on the orbit height are mitigated. However, this does not mean that the mass is exactly at $x_a = 0$, as can be seen in the position of the seismic mass plot. The mass oscillates around the equilibrium position, which means that there is a delay in the thrust response – it is always behind the drag. Since the drag oscillates, the thrust is periodically a tiny amount bigger and smaller than the drag. This periodic behavior can, therefore, be seen in all the four plots on the right. Once again, the amplitude of the oscillations depends on the speed of the thrust response.

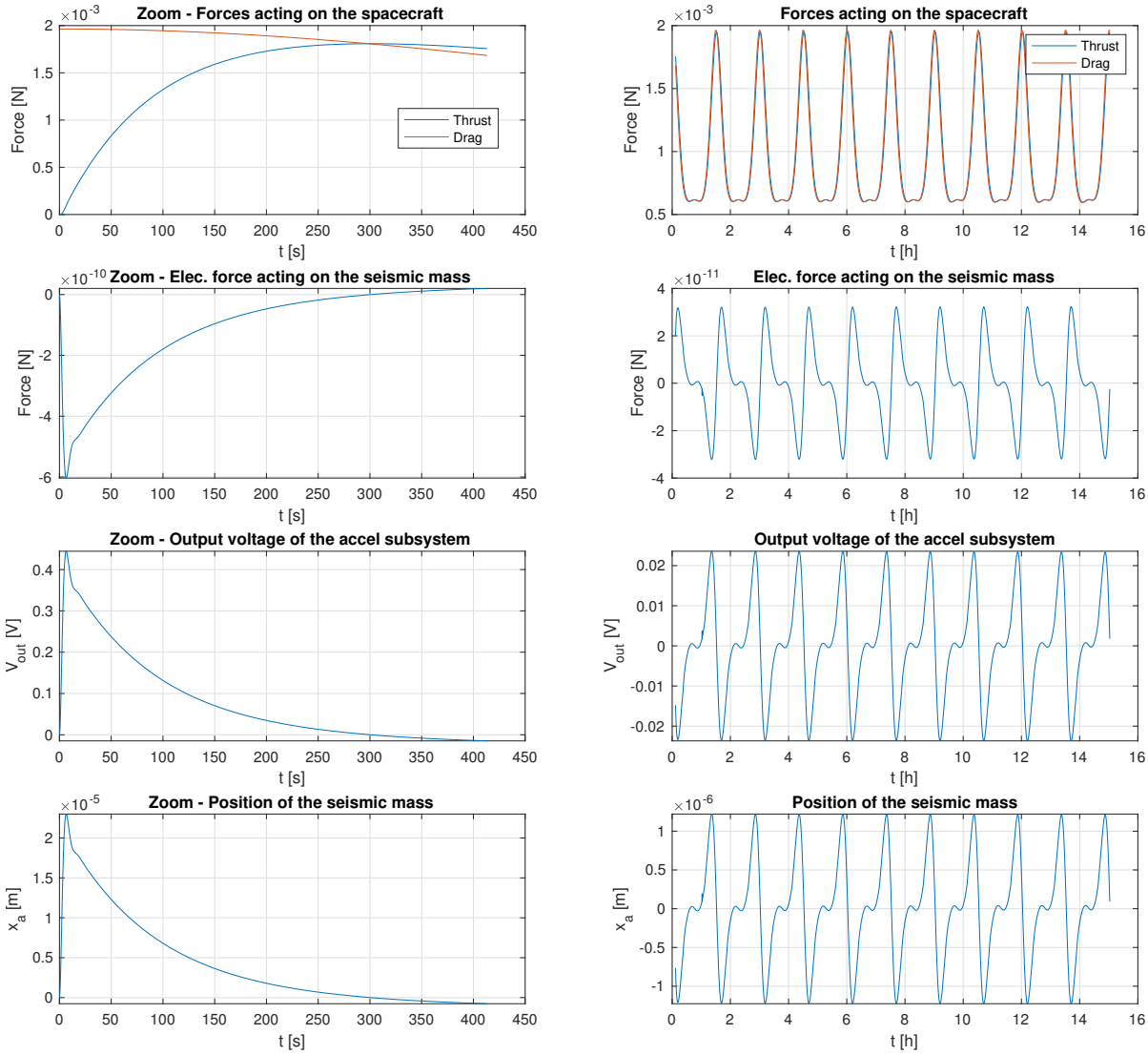

Figure 9: Accelerometer related plots: force acting on spacecraft, electrical force, output voltage and seismic mass position. On the left: first $\sim$ 400s. On the right: rest of the simulation.

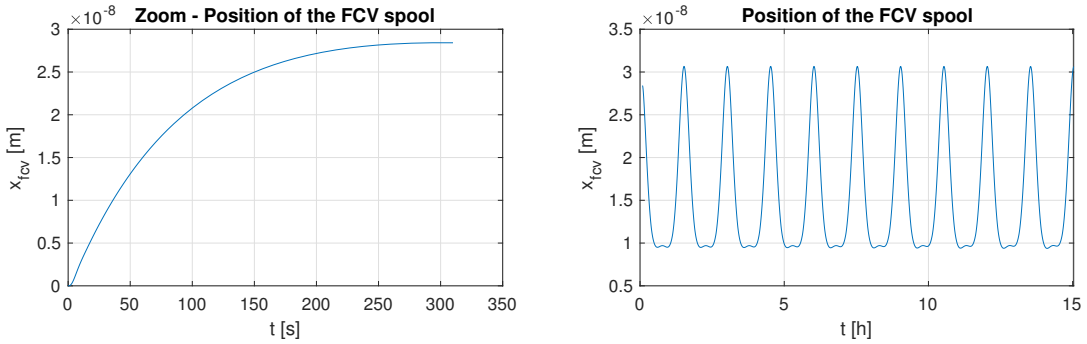

Figure 10: Position of the spool of the flow control valve subsystem. On the left: first $\sim$ 300s. On the right: rest of the simulation

Finally, as can be seen in Fig. 10, the position of the spool also has a periodic tendency, which is imposed by the accelerometer. Initially, as seen on the plot on the left, the valve opens in order to increase the thrust due to the initial discrepancy in the forces. Throughout the rest of the simulation, the position of the spool oscillates around a non-zero value, which goes to show the effect of the integral part of the controller. As expected, the shape of the spool position curve is similar to the torque curve.

6.2 Off-nominal conditions

Two off-nominal conditions were studied:

- the accelerometer seismic mass gets stuck: $x_a = \text{const} \Leftrightarrow v_a = 0$;
- the valve spool mass gets stuck: $x_v = \text{const} \Leftrightarrow v_v = 0$

These were modeled to occur at $t = 3000\,\text{s}$ and to last $\Delta t = 1200\,\text{s}$, returning to normal conditions at $t = 4200\,\text{s}$. To do so, the simulation was normally run until $t = 3000\,\text{s}$, when the values of the final states were saved. The adequate changes were made to those values, in order to model the off-nominal condition, and then they were used as the initial states values of a new simulation, with the initial time being the time at which the previous simulation was interrupted. This simulation went for the off-nominal condition duration and the derivatives were modified accordingly. Lastly, a third simulation was run with the normal system response, in order to model the system recovering

6.2.1 Accelerometer seismic mass stuck

In order to test the ability of the system recovering from a problem in the accelerometer, for example, its mass getting stuck, the velocity and acceleration of the accelerometer's mass was forced to zero during the off-nominal condition duration.

The system response, regarding the thrust with respect to the drag, and the position of the accelerometer's mass can be seen in Fig. 11.

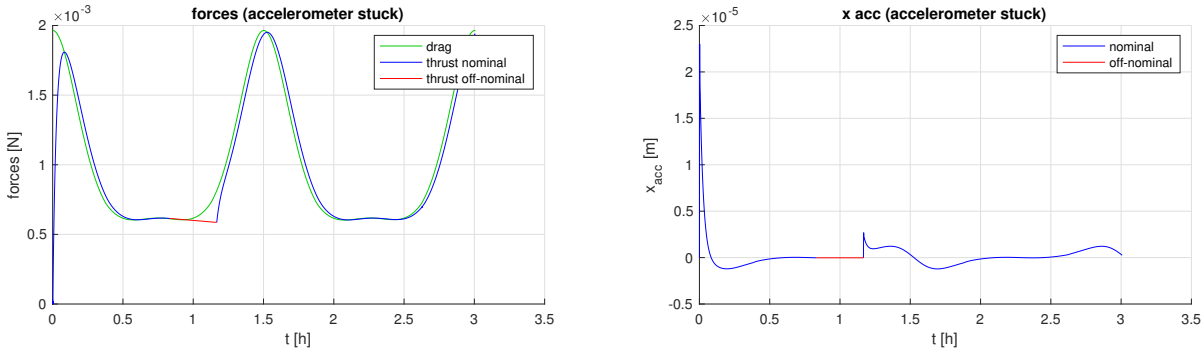

Figure 11: Results for the off-nominal condition of accelerometer malfunction.

This off-nominal condition doesn't affect the system that much, since the accelerometer getting stuck would result in a constant output voltage, not necessarily stopping the thrust. As one can see, the thrust maintains its rate of change during the off-nominal period, which will affect the system differently depending on the time at which the malfunction occurs.

6.2.2 Valve spool mass stuck

A problem in the valve was modeled as its mass getting stuck. In this case, its velocity and acceleration were forced to zero, which results in a constant thrust. The results can be seen in Fig. 12.

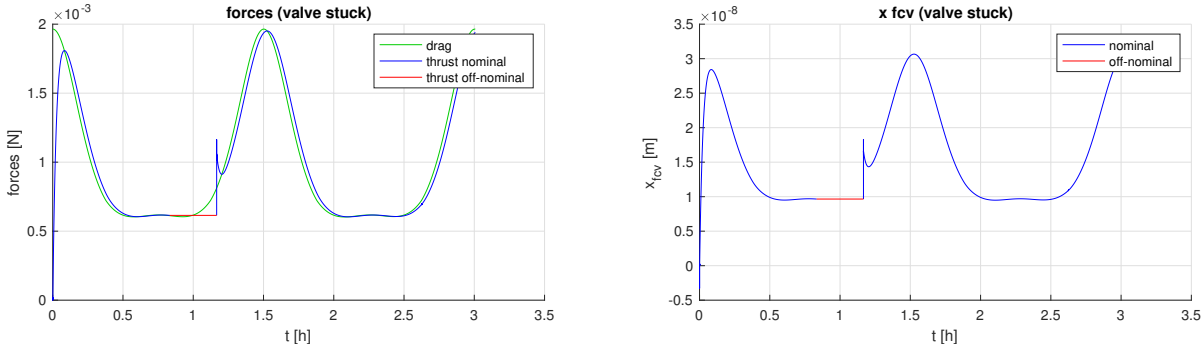

Figure 12: Results for the off-nominal condition of flow control valve malfunction.

As expected, the consequences of this condition are more significant, since the valve's mass getting stuck would result in the thrust being constant for that amount of time. Once again, the effect on the system will depend on the instant at which the malfunction occurs. Nonetheless, when the system is able to return to its normal conditions, it reacts pretty quickly to assure equilibrium, from what one can assume that the system is very robust to the modeled off-nominal conditions.

7 System optimization

Since the goal of GOCE is to maintain a drag free orbit, then it's desired for the thrust to follow the drag as accurately as possible. Hence, the objective function, this is, the cost function, was

$$J = \|D - T\|^2 \tag{41}$$

The parameters that were identified as optimizable were: K_{p_a}, c, k, m_{fcv}, K_{p_v}, K_{i_v}. Notice that K_i wasn't included in the optimization, since it only affects the force proportionally to the current, and the latter depends linearly of K_{p_v} and K_{i_v} – Eq. (42). Therefore, one of these three parameters can be omitted in the optimization.

$$F_{actuator} = K_i \cdot i = K_i K_{p_v} V_{out} + K_i K_{i_v} V_i \tag{42}$$

In order to optimize this system, `fmincon()` was used. As said before, the minimizing function was Eq. (41). To calculate its value, the simulation was run for 2 orbital periods with the same initial values as in the "normal" simulation. In the end, the norm of the difference between the drag and the thrust is computed and returned.

Some lower and upper bounds were defined for the parameters. In order for the optimization not to diverge, those bounds were defined as an order of magnitude lower and higher than the initial values.

Nonlinear constraints were also defined. One to keep the simulation time lower than 10s, since the contrary would make the optimization slow and expensive. Another to make sure that no events are observed (during the ODE solve) since these events would be physical boundaries that would be exceeded, that is, the positions of the accelerometer and the spool.

`fmincon()` was run with the algorithm `active-set` (it isn't large-scale [11]), and with a step tolerance of 10^{-6}. The initial parameters and the ones obtained after the optimization are shown:

$$\begin{cases} K_{p_a} = 1 \\ c = 0.2\,\text{Ns/m} \\ k = 0.1\,\text{N/m} \\ m_{fcv} = 0.2\,\text{kg} \\ K_{p_v} = 1 \times 10^{-4}\,\text{A/V} \\ K_{i_v} = 3 \times 10^{-3}\,\text{A/(Vs)} \end{cases} \Rightarrow \begin{cases} K_{p_a} \simeq 0.8852 \\ c \simeq 0.0988\,\text{Ns/m} \\ k \simeq 0.0818\,\text{N/m} \\ m_{fcv} \simeq 0.1040\,\text{kg} \\ K_{p_v} \simeq 6.03 \times 10^{-4}\,\text{A/V} \\ K_{i_v} \simeq 0.0174\,\text{A/(Vs)} \end{cases} \tag{43}$$

The optimization took about 13 min and the value of the cost function was decreased about 58.7%. Note that since the valve starts closed, and therefore the thrust is zero, there will always be a difference between the drag and the thrust at the start. The iterations of the optimization can be seen in Fig. 13.

Iter	F-count	f(x)	Max constraint	Line search steplength	Directional derivative	First-order optimality	Procedure
0	7	0.000726437	0				
1	18	0.000716439	0	0.0625	-3.59e+03	4.74e+03	
2	26	0.000360946	0	0.5	-2.73e+03	3.55e+03	
3	41	0.000332117	0	0.00391	-112	1.68e+03	
4	52	0.000301121	0	0.0625	-12.5	17.4	
5	69	0.000300195	0	0.000977	-8.5	13	
6	87	0.000300101	0	0.000488	-2.18	13.2	

Figure 13: Iterations of `fmincon()` applied as described above.

The change in the system response, regarding the thrust, can be seen in Fig. 14. Comparing both figures, they seem pretty similar. However, it's clear that there was an improvement in the speed of response of the thrust. Especially in the start of the simulation (left graphs), the thrust equals the drag much faster. It's also possible to observe some little oscillations, which can be understood as a loss in damping in order to increase the speed of the system's response. Throughout the simulation, the thrust follows the drag much quicker after the optimization.

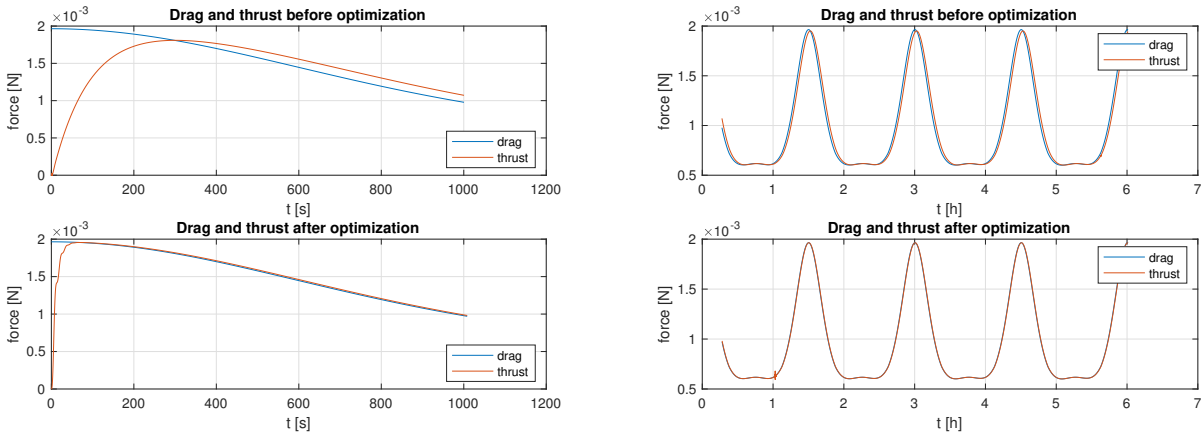

Figure 14: Drag and thrust before and after optimization. Plots on the left show the first 1000 s and plots on the right show the rest of the simulation during 2 orbits.

References

[1] A. Bacchetta, L. Colangelo, E. Canuto, S. Dionisio, L. Massotti, C. Novara, M. Parisch, and P. Silverstrin, "From GOCE to NGGM: Automatic control breaktroughs for european future gravity mission," *IFAC PapersOnLine 50-1*, p. 1, 2017.

[2] D. Muzi, R. Floberghagen, and M. Drinkwater, "GOCE: ESA'S FIRST GRAVITY MISSION," *Envisat Symposium 2007*, p. 6, 2007.

[3] P. J. S. Gil, "Elementos de mecânica orbital," *Instituto Superior Técnico*, p. 204, December 2015.

[4] C. Colombo, "Orbital mechanics: Orbit perturbations," *Politecnico di Milano*, 2017.

[5] E. G. K. I. L. Baginsky, "Capacitive mems accelerometers for measuring high-g accelerations," *Optoelectronics Instrumentation and Data Processing*, pp. 2–3, May 2017.

[6] M. Bao, *Analysis and Design Principles of MEMS Devices*. ELSEVIER, 2005.

[7] F. Topputo, "Course notes of "modeling and simulation of aerospace system"," *Politecnico di Milano*, 2018-2019.

[8] R. H. Battin, *An Introduction to the Mathematics and Methods of Astrodynamics, Revised Edition*. American Institute of Aeronautics and Astronautics, Inc., 1999.

[9] J. Raymond A. Serway, John W. Jewett, *Physics*. Physical Sciences, Physics and Astronomy, 2014.

[10] J. John D. Anderson, *Modern Compressible Flow: With Historical Perspective*. McGraw-Hill Publishing Company, 1990.

[11] *MATLAB Documentation*. https://www.mathworks.com/help/index.html.

[12] R. V. Ryuichi Ashino, Michihiro Nagase, "Behind and beyond the matlab ode suite," *Osaka Kyoiku University*, p. 24, January 2000.

[13] M. Embree, "Lectures 37 and 38: Linear multistep methods: Absolute stability." http://www.math.vt.edu/people/embree/math5466/lecture37_38.pdf.