MSAS – Assignment #2: Modeling

Marcello Pareschi, 252142

Exercise 1

To perform a preliminary analysis and evaluate a proposed satellite configuration, it is requested to model an active thermal control system and its associated mechanisms to assess the system's capability to maintain the temperatures within a certain range $[T_{min}; T_{max}]$. The satellite body is a rectangular cuboid (1.5 [m] height, 0.5 [m] side), with fixed solar panels (0.5 [m] side, 0.95 [m] length) located on the +Y-axis. The satellite flies with a fixed inertial attitude, with the satellite's Y-axis always directed towards the Sun. The satellite has two extendable radiators, which are located on the ±X-axis faces. The radiators are connected to the satellite body with electrically controlled hinges that can rotate along the Z-axis direction. The satellite with its radiators in open and closed configurations is represented in Figure 1.

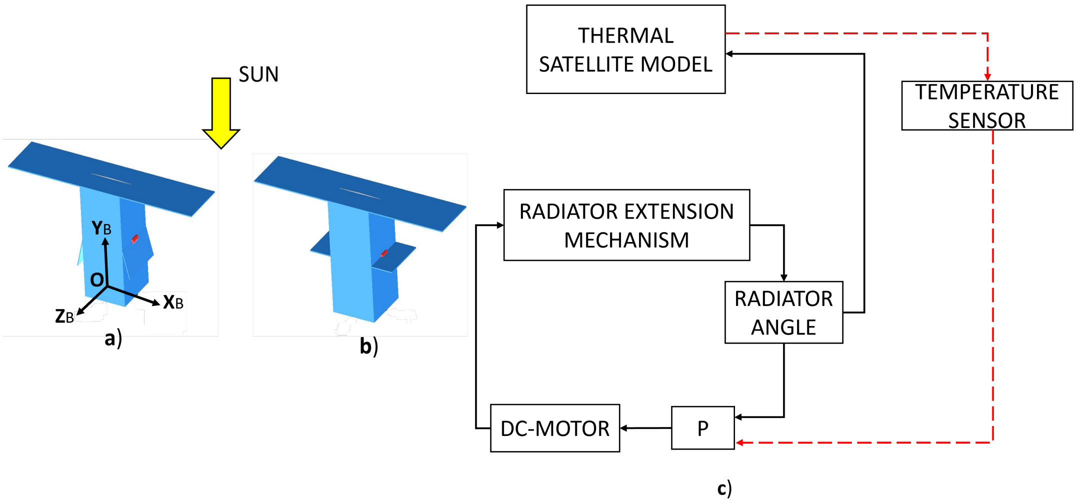

Figure 1: Satellite configuration: **a**) Closed radiator; **b**) Open radiator; **c**) Thermal control logic overview

The extension mechanism is based on a DC motor which gives the needed torque to move the radiator to the desired position. In Figure 2 the physical model of the mechanism is shown. The angle θ represents the rotation of the radiator with respect to the Z-axis and is measured counter-clockwise. For a rotation angle $\theta = 0$ deg, the radiators are in fully open configuration, for $\theta = -0.4\pi$ the radiators are in stowed (closed) configuration. The main parameters of the electro-mechanical model are reported in Table 1.

Parameter	**Symbol**	**Value**	**Units**
Resistance	R	0.1	Ω
Inductance	L	0.001	H
Motor constant	k_m	0.3	Nm/A
Radiator mass	m_r	0.2	kg
Radiator length	L_r	0.5	m

Table 1: Characteristics of the thermal control mechanism.

The radiator has a lower emissivity side, facing deep space in the closed configuration, and a higher emissivity side which is hidden when the radiator is in the stowed position. It is assumed

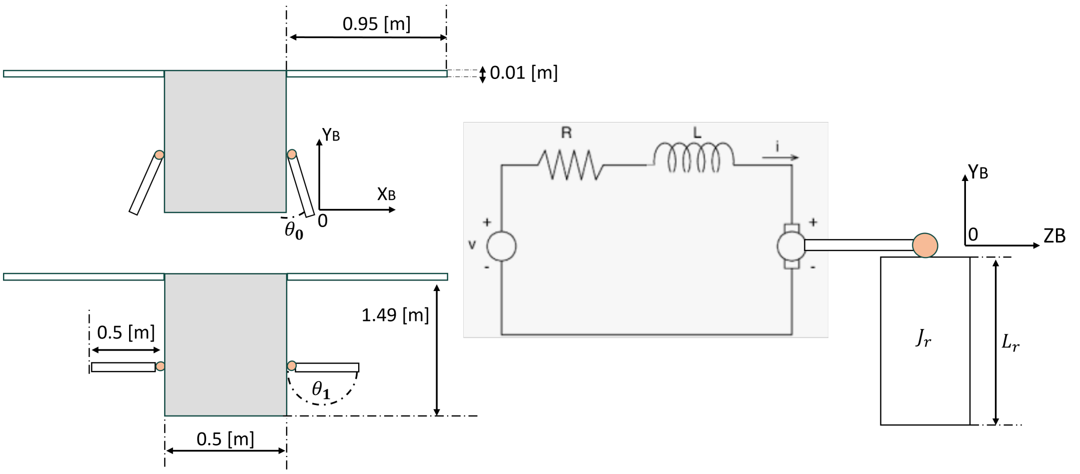

Figure 2: Schematics of the radiator extension mechanism

that radiative heat exchange among the satellite surfaces (re-radiation) is negligible and that no other thermal radiation sources (e.g., planets) are present. As a result, the thermal control of the satellite is regulated mainly by the radiator angle θ, as shown in Figure 1 c). By opening and closing the radiator, the effective area of the high- and low-emissivity sides exposed to deep space increases and decreases respectively. This results in a linear variation of the emissivity ϵ as a function of θ that can be expressed as in Eq. (1):

$$\epsilon(\theta) = \epsilon_{min} + \left[\frac{\epsilon_{max} - \epsilon_{min}}{0.4 \cdot \pi}\right](\theta(t) + 0.4 \cdot \pi) \tag{1}$$

When the radiator is in a closed configuration with $\theta(t) = -0.4\pi$ [rad] the emissivity is at its minimum $\epsilon_{min} = 0.01$. In fully open configuration instead, when $\theta(t) = 0$ [rad], the radiator emissivity reaches its maximum value $\epsilon_{max} = 0.98$. Using a lumped nodes thermal modeling, the satellite can be divided into 5 components (the two solar panels, the main body, and the two radiators). The radiators and solar panels are connected to the main body node by a conductive path, while all nodes are thermally radiating towards deep space (see Figure 3).

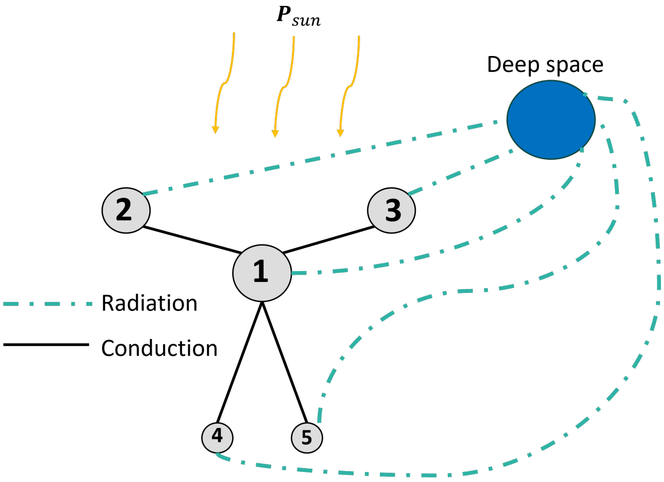

Figure 3: The thermal lumped nodes model

The solar radiation from the Sun is acting only on the two solar panel nodes and on the top face of the main-body node (the radiators are always in shadow). The temperature of deep space is constant and equal to T_{ds}= 3 [K]. The characteristics of the thermal network and its components are given in Table 2.

Parameter	Symbol	Value	Unit
Sun Power	P_{Sun}	1350	W/m^2
Heat capacity	C_1	1.5×10^5	J/K
Heat capacity	C_2, C_3	1187.5	J/K
Heat capacity	C_4, C_5	30	J/K
Thermal conductance	G_{12}, G_{13}, G_{14}, G_{15}	10	W/K
Absorptivity	α_1	0.6	-
Absorptivity	α_2, α_3	0.78	-
Emissivity	ϵ_1	0.45	-
Emissivity	ϵ_2, ϵ_3	0.75	-

Table 2: Thermal parameters of the system.

Part 1: causal modeling (9 points)

Considering the following main constitutive equations of the system:

$$C_i \frac{dT_i}{dt} = \sum_i^n (Q_{in}^i - Q_{out}^i) \tag{2}$$

$$V_{in} = k_p \cdot (T_1 - T_1^{ref}) \tag{3}$$

$$J_r \ddot{\theta} = \tau_{in} \tag{4}$$

where Eq. (2) is the conservation of energy at a thermal node; Eq. (3) is a proportional control law in which V_{in} is the input voltage applied on the DC-motor, and T_1^{ref} is the reference temperature of the main-body; and Eq. (4) is the simplified rotational radiator dynamics where the input torque is linked to current flowing into the motor through the relation: $\tau_{in}= k_m \cdot i$. Assuming that: the upper limit $\hat{T}_{max}$ and lower limit temperature $\hat{T}_{min}$ allowed for **node 1** are 300 [K] and 290 [K] respectively, the motor control is capable of maintaining the angle θ within the physical limits, all the thermal nodes start with a temperature of 298.15 [K], and that the radiator is closed at the beginning of the simulation ($\theta(0) = -0.4\pi$):

1. Formulate the full system of non-linear ODEs making explicit the state variables of the whole simulation.
2. Considering a simulation time of at least 50 [h], define the value for the proportional gain k_p such that: the temperature is kept around the target $T_1^{ref} = 294.15$ [K], and within $t = 10$ [h] the maximum temperature oscillations are less than 0.1% of T_1^{ref}. Plot the resulting temperature evolution over time on all thermal nodes.
3. Discuss the results and the ode used in Matlab for the integration.

Part 2: acausal modeling (6 points)

Using the Modelica standard library, reproduce in Dymola the physical model of the thermal-electro-mechanical system described above. You can build your **own model block** in Modelica to implement specific signals such as $\epsilon(\theta)$ or the control strategy. Please note that when you create a new model block (see in Table 3) you can insert input/output variables and your equations (see as an example the **Text View** of the block: *Modelica/Blocks/Math/Add*). Then, simulate it in OpenModelica comparing the results with the ones obtained using the causal model.

(15 points)

Physical element	*Modelica library*
Rigid-body	Modelica/Mechanics/MultiBody/Parts
Revolute joint	Modelica/Mechanics/MultiBody/Parts
Fixed-translational element	Modelica/Mechanics/MultiBody/Parts
DC-motor	Modelica/Electrical/Machines/BasicMachines/DCMachines
Thermal node	Modelica/Thermal/HeatTransfer/Components
Thermal resistances	Modelica/Thermal/HeatTransfer/Components
Logical switch	Modelica/Blocks/Logical/Switch
Hysteresis	Modelica/Blocks/Logical/Switch
Temperature sensor	Modelica/Thermal/HeatTransfer/Sensors
Your model block	File/New/Block

Table 3: Modelica libraries.

1 Part 1: Causal Modeling

1.1 Mathematical modeling

The satellite thermal model is based on the lumped parameters approach (LPA). According to this method, the satellite surfaces are modeled as thermal nodes with a certain thermal capacity, and their temperatures were taken as state variables. The thermal nodes are capable of exchange heat each other, with the Sun and with the deep space. While the various sources of heat transfer will be discussed in detail below, it is important to first highlight the simplifications made regarding the radiative heat exchange. The following assumptions are applied to the different nodes:

- node one, which is representative of the main body, receives the solar radiation only on the top face area $A_1^{top} = 0.25$ m^2, while emits toward deep space from its total surface $A_1 = 3.5$ m^2
- nodes two and three, associated to the solar panels, absorb the solar radiation from the top surface, while emit towards the deep space from the bottom surface. Both node two and three have area $A_2 = A_3 = 0.475$ m^2. The solar panels have ha thickness of 0.01 m, which was neglected. Thus the total solar panels side area, which is of 0.048 m^2, was neglected. As a consequence, the disspating area used in nodes two and three is $\approx 5\%$ smaller than the real solar panel disspating area
- the surface area of the radiators, corresponding to nodes four and five, is $A_4 = A_5 = 0.25$ m^2, the side area was neglected, as it was not provided.

Given these assumptions, it is possible to write down the ODEs ruling the temperature of each thermal node writing the energy balance (Equation 2) on each node. Thermal power is positive when entering the system, causing an increase in the node temperature. The conductive exchange between the nodes is governed by a unique thermal conductance G. The radiator emissivity $\varepsilon(\theta)$ varies with the radiator angle θ as stated in Equation 1.

$$\begin{aligned}
C_1\frac{dT_1}{dt} &= \alpha_1\, A_1^{top}\, P_{sun} + G\sum_{j=2}^{5}(T_j - T_1) - \varepsilon_1\, A_1\, \sigma\,\left(T_1^4 - T_{DS}^4\right)\\
C_2\frac{dT_2}{dt} &= \alpha_2\, A_2\, P_{sun} + G\,(T_1 - T_2) - \varepsilon_2\, A_2\, \sigma\,\left(T_2^4 - T_{DS}^4\right)\\
C_3\frac{dT_3}{dt} &= \alpha_3\, A_3\, P_{sun} + G\,(T_1 - T_3) - \varepsilon_3\, A_3\, \sigma\,\left(T_3^4 - T_{DS}^4\right)\\
C_4\frac{dT_4}{dt} &= G\,(T_1 - T_4) - \varepsilon\,(\theta)\, A_4\, \sigma\,\left(T_4^4 - T_{DS}^4\right)\\
C_5\frac{dT_5}{dt} &= G\,(T_1 - T_5) - \varepsilon\,(\theta)\, A_5\, \sigma\,\left(T_5^4 - T_{DS}^4\right)
\end{aligned}$$

The equation governing the angular position of the radiator is a second order ODE, which models an ideal DC motor with no damping assumption, as stated in Equation 4. The torque τ_{in} is proportional to the current flowing into the circuit, the inertia of the radiator can be computed as $J_r = 1/3\, m_r\, L_r^2$. The second order ODE can be rewritten as a first order ODE's system introducing the radiator angular velocity ω:

$$\begin{cases}\dot{\theta} = \omega\\ J_r\,\dot{\omega} = k_m\, i\end{cases}$$

The ODE governing the dynamics of the electric circuit illustrated in figure can be written applying the Kirchhoff's voltage law:

$$L\frac{di}{dt} = V_{in} - Ri - k_m\omega$$

where L and R denote the inductance and resistance od the DC motor, respectively, and $V_{in} = k_p\,(T_1 - T_{ref})$ is the input voltage applied to the DC motor. The system under analysis can be rewritten in state form by taking as state vector

$$\mathbf{x} = \begin{bmatrix} T_1 & T_2 & T_3 & T_4 & T_5 & \theta & \omega & i \end{bmatrix}^T$$

The system of non-linear ODEs describing the system dynamics in state-space form is then:

$$\begin{cases} \dfrac{dT_1}{dt} &= \dfrac{1}{C_1}\left(\alpha_1\, A_1^{top}\, P_{sun} + G \sum_{j=2}^{5} (T_j - T_1) - \varepsilon_1\, A_1\, \sigma\, \left(T_1^4 - T_{DS}^4\right)\right) \\ \dfrac{dT_2}{dt} &= \dfrac{1}{C_2}\left(\alpha_2\, A_2\, P_{sun} + G\,(T_1 - T_2) - \varepsilon_2\, A_2\, \sigma\, \left(T_2^4 - T_{DS}^4\right)\right) \\ \dfrac{dT_3}{dt} &= \dfrac{1}{C_3}\left(\alpha_3\, A_3\, P_{sun} + G\,(T_1 - T_3) - \varepsilon_3\, A_3\, \sigma\, \left(T_3^4 - T_{DS}^4\right)\right) \\ \dfrac{dT_4}{dt} &= \dfrac{1}{C_4}\left(G\,(T_1 - T_4) - \varepsilon\,(\theta)\, A_4\, \sigma\, \left(T_4^4 - T_{DS}^4\right)\right) \\ \dfrac{dT_5}{dt} &= \dfrac{1}{C_5}\left(G\,(T_1 - T_5) - \varepsilon\,(\theta)\, A_5\, \sigma\, \left(T_5^4 - T_{DS}^4\right)\right) \\ \dfrac{d\theta}{dt} &= \omega \\ \dfrac{d\omega}{dt} &= \dfrac{k_m}{J_r} i \\ \dfrac{di}{dt} &= \dfrac{1}{L}\left(-Ri - k_m\omega + k_p\,(T_1 - T_{ref})\right) \end{cases} \tag{5}$$

1.2 Numerial simulation

Once the system of ODEs in Equation 5 was properly defined, the right-hand side (RHS) was implemented in a MATLAB function, incorporating constraints to ensure that the radiator angles remain within the specified boundaries: $\theta_{min} = -0.4\pi$ rad and $\theta_{max} = 0$ rad. Specifically, the control is deactivated by setting the proportional gain to zero, effectively simulating a null input voltage in the DC motor. This occurs when θ reaches its lower boundary while T_1 is below the target temperature or when θ reaches the upper boundary while T_1 exceeds the target temperature. With this implementation, the current does not immediately drop to zero; however, the transient is expected to be very short.

At the initial time, both the circuit current and the radiator angular velocity were set to zero. The radiator angle was initialized at $\theta(0) = -0.4\pi$ rad, corresponding to the fully closed position. All thermal nodes were assigned an initial temperature of 298.15 K.

Additional considerations were made regarding the tuning of the parameter k_p. The proportional gain was set to $k_p = 6 \cdot 10^{-5}$, a value carefully selected to ensure that T_1 remains within $T_{ref} \pm 0.1\% T_{ref}$ after 10 hours. While a larger gain would result in faster convergence, it would also lead to excessive angular velocity rates for the radiators, potentially causing mechanical damage in a real mission scenario.

Once that the main steps of the numerical implementation were clarified, the obtained results for a simulation of 50 h can be discussed. Figure 4 shows the time evolution of the main body temperature. It can be noted that T_1 converges to $T_{ref} = 294.15\ K$, as requested. Two horizontal lines, corresponding to $T_{ref} \pm 0.1\% T_{ref}$, are included to demonstrate that after ten hours the temperature oscillations remain within the desired boundaries.

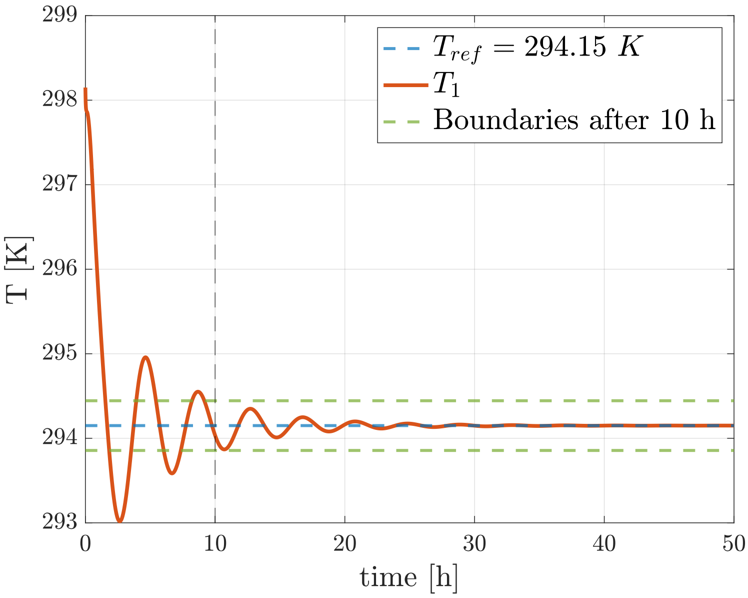

Figure 4: Main body temperature time evolution

The time evolution of the radiator position angle is depicted in Figure 5, along with the time evolution of the angular velocity. It can be observed that, since the initial temperature of node one, corresponding to the main body, is higher than the target temperature, the radiators are immediately opened by the control logic to dissipate heat. The maximum position angle is reached and maintained for some time, during which the angular velocity is zero. Additionally, as shown in Figure 5, when the system reaches the steady-state condition of $T_1 = T_{ref}$, the radiator angle converges to a value of $\overline{\theta} \approx -36.0525$ deg. After the transient phase, the angular velocity converges to 0 rad/s.

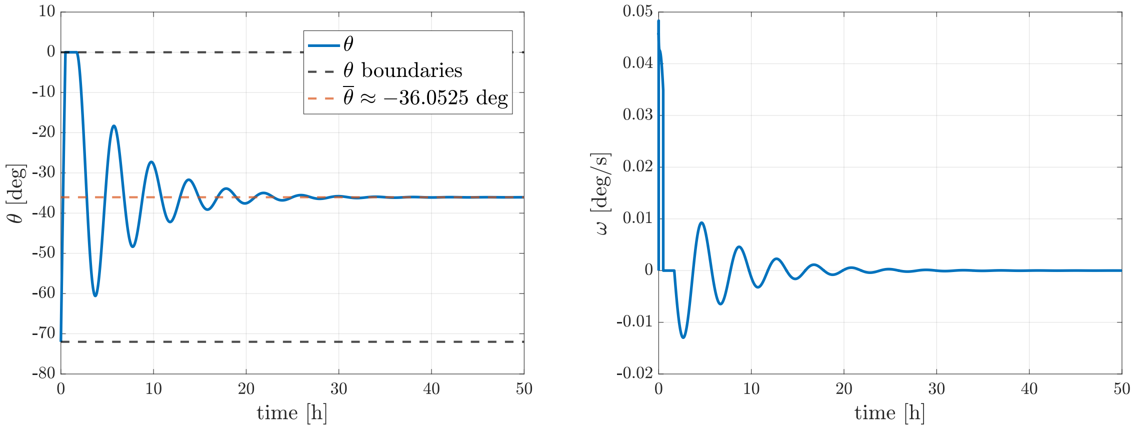

Figure 5: Radiator position angle (left) and angular velocity (right) time evolution

Figure 6 illustrates the time evolution of the temperature of the solar panels, corresponding to nodes two and three, and the radiators, corresponding to nodes four and five. A single plot is provided for both solar panels since their thermal balance and initial conditions are identical, resulting in the same temperature evolution. The same reasoning applies to the radiators' dynamics.

In general, the thermal response of a system is strongly influenced by the thermal capacity of its nodes and the net exchanged power. Figure 6 shows that the temperature of the solar panels initially increases by approximately 27 K in a short time. This rapid evolution, compared to the slower temperature change of the main body, is due to the much lower thermal capacity of the solar panels. At the beginning of the simulation, their temperature rises because the absorbed solar power from the top face exceeds the radiative heat dissipation from the back face into deep space. A steady-state condition is reached when the temperature difference between the solar panel nodes and the main body node becomes sufficient for conductive heat transfer to remove the excess heat that cannot be dissipated via radiation.

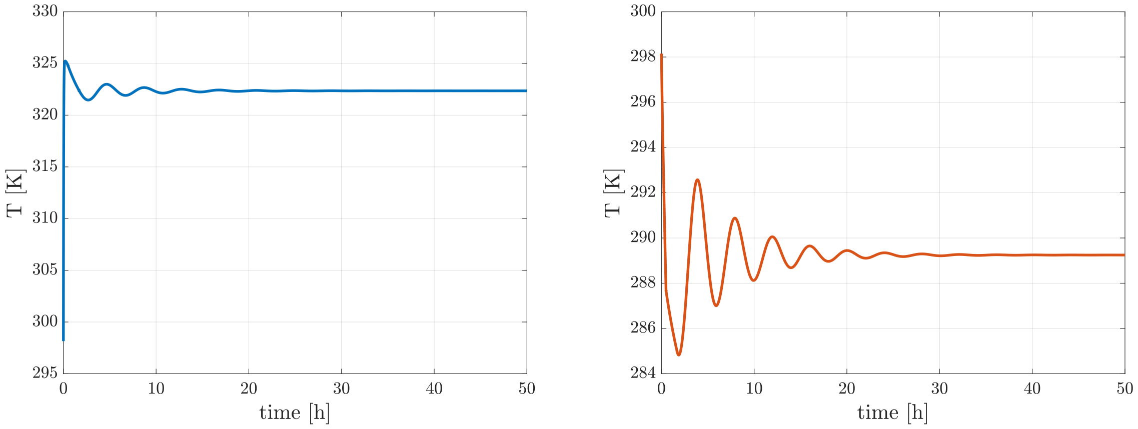

Figure 6: Time evolution of solar panel temperature (left) and radiator temperature (right).

Figure 6 also depicts the time evolution of the radiator temperature. It can be observed that the radiators reach a lower steady-state temperature compared to the other nodes. This occurs because the radiators are not exposed to direct solar radiation; thus, the only incoming power is the conductive heat transfer from the main body.

At the beginning of the simulation, a rapid temperature drop is observed because, initially, the radiators share the same temperature as the main body. Since conductive heat exchange is minimal at this stage, the radiators predominantly dissipate heat into deep space via radiation. Additionally, it can be noted that the radiators exhibit larger temperature variations compared to the other nodes. This behavior is expected due to their significantly lower thermal capacity.

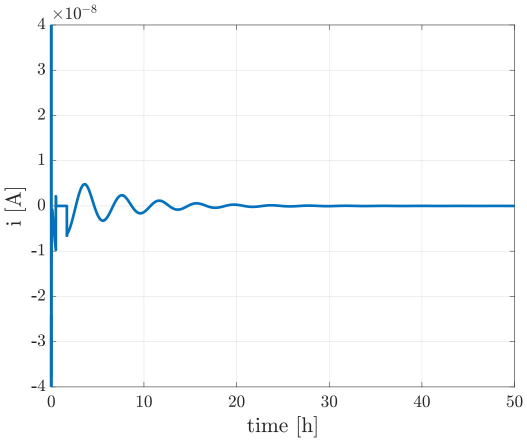

Figure 7: Current time evolution

Figure 7 shows the time history of the current flowing through the circuit. An initial sharp peak can be observed, which occurs because, in the early instants of the simulation, the difference between T_1 and T_{ref} is significant. Since the input voltage is proportional to this temperature difference, a high current is initially generated. As the radiator approaches the fully open position, the control logic disables the input voltage, causing the current to drop. However, a small overshoot is present due to the inductor in the circuit, which introduces a brief transient even after the voltage is set to zero. Eventually, the current stabilizes to zero as the system reaches steady-state conditions and the temperature difference between T_1 and T_{ref} becomes negligible.

1.3 ODE selection

To accurately integrate the system of nonlinear ODEs presented in Equation 5, MATLAB's `ode15s` solver was employed. This choice was motivated by the stiffness of the system, which arises from the multi-physics nature of the problem. Specifically, the electrical circuit dynamics evolve much more rapidly than the mechanical system, while the thermal model operates on an even slower timescale. `ode15s` is particularly well-suited for such stiff systems, as it provides a balance between numerical stability and computational efficiency throughout the simulation.

2 Part 2: Acausal Modeling

2.1 Acausal Model Description

The acausal model of the system under analysis was implemented in Dymola exploiting built-in blocks from the Modelica libraries togheter with custom blocks to deal with the variable emissivity of the radiators and the control logic. The model, illustrated in Figure 8, was divided into three subsystems: the thermal model, the electromechanical model and the control logic. This section will briefly discuss the blocks used for each subsystem and present the results, followed by a comparison with the results obtained from causal modeling.

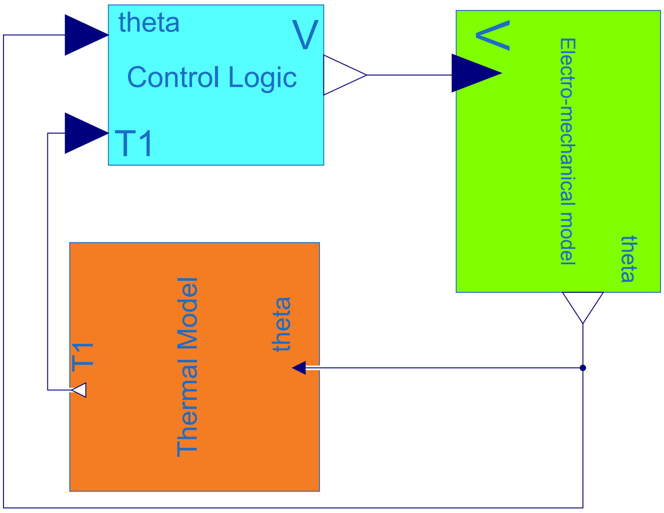

Figure 8: Complete acausal model in Dymola

In Figure 9 is illustrated the the acausal model od the thermal subsystem. The five nodes of the lumped-parameter thermal model of the satellite—representing the main body, the solar panels, and the radiators—were modeled using the *Heat Capacitor* block. The initial temperature of each block was set to $T_0 = 298.15$ K, in accordance with the specifications provided for the exercise. To model the incoming solar flux on the top face of the main body and the solar panels, a *Fixed Heat Flow* block was connected to each thermal node exposed to the Sun. The solar power entering each exposed node was computed as $Q_{\text{Sun}} = \alpha^{(i)} A^{(i)} q_{\text{Sun}}$, where $\alpha^{(i)}$ is the absorptivity, $A^{(i)}$ the surface area exposed to the Sun, and q_{Sun} the solar flux density. As a result, 202.5 W are absorbed by the main body, while 500.175 W are absorbed by each solar panel node.

Conductive heat exchange between nodes was modeled by connecting the thermal masses through *Thermal Conductor* blocks. Radiative heat exchange was implemented using *Body Radiation* blocks, linking the main body, solar panels, and radiators to deep space. Deep space was modeled using a *Fixed Temperature* block set to 3 K. To accurately model the radiative behavior of the radiators, two custom blocks were developed and integrated into the system:

- The first custom block is a modified version of the *Body Radiation* block, capable of accepting a variable emissivity as input.
- The second block computes the equivalent emissivity as a function of the radiator angle, according to Equation 1.

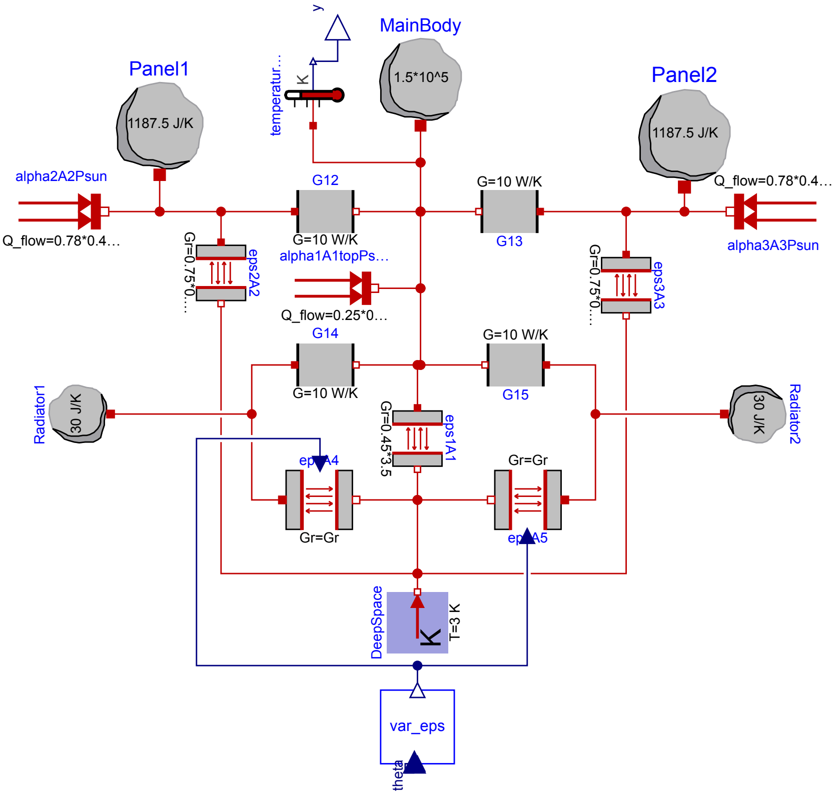

Figure 9: Thermal model in Dymola

The DC motor, which generates the torque required to rotate the radiators to their desired position, was modeled as an RL series circuit with an additional back electromotive force (EMF), which opposes the applied voltage, as illustrated in Figure 10. The voltage was supplied using a *Signal Voltage Source* block, which receives as input the signal output from the *Control Logic*. The resistance and inductance of the circuit were modeled using *Resistor* and *Inductor* blocks, respectively. The back EMF was implemented using the *Rotational EMF* block, with a motor constant set to $k_m = 0.3$ Nm/A.

The rotational dynamics of the radiator was captured using the *Rotational Inertia* block, which was connected to the mechanical output of the EMF block. An *Angle Sensor* was placed at the end of the chain to measure the angular displacement θ, which serves as an input to the *Control Logic* block.

Although the satellite includes two radiators, only one electromechanical model was implemented. This is justified by the thermal symmetry of the system with respect to the Y-axis, which ensures that both radiators receive the same commanded angle. Moreover, this modeling choice reduces the computational load of the simulation.

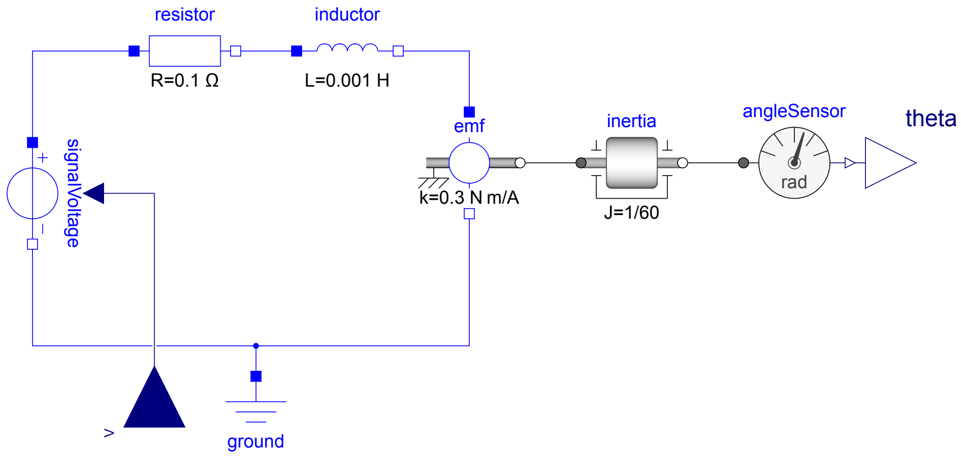

Figure 10: Electro-mechanical model in Dymola

The *Control Logic* block, depicted in Figure 11, was implemented to replicate the active control algorithm developed in MATLAB. The block receives as inputs the main body temperature T_1 and the radiator angle θ, which are the outputs of the *Thermal Model* and *Electromechanical Model*, respectively. Its output is the input voltage to be applied to the DC motor.

The condition $-0.4\pi \leq \theta \leq 0$ is verified using an *AND* port, which leverages the Modelica built-in blocks *lessEqual* and *greaterEqual*. The output of the *AND* port is connected to a *Logical Switch* block, ensuring that the input voltage is set to zero if θ exceeds the specified boundaries. The control is re-activated when the main body temperature falls below the reference temperature of 294.15 K.

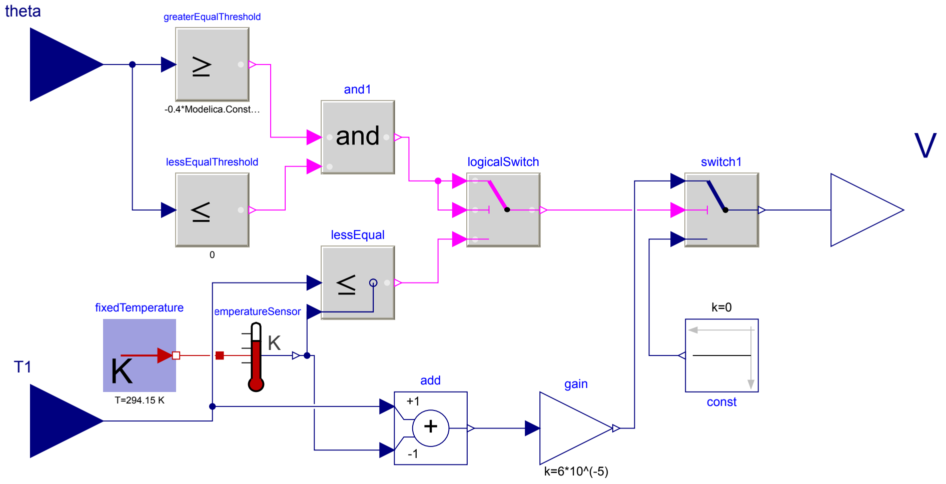

Figure 11: Control Logic in Dymola

2.2 Simulation setup and Results

After having described the model, this section presents the simulation setup and the corresponding results. The simulation was executed in *OpenModelica* over a time span of 50 hours, using the DASSL integrator with a tolerance of 10^{-12}. DASSL was selected due to its suitability for simulating stiff multi-physics systems.

To facilitate a direct comparison between the causal and acausal modeling approaches, the simulation results from *OpenModelica* were exported to MATLAB in *CSV* format and subsequently plotted in the MATLAB environment. Figure 12 depicts the time history of T_1, T_2, T_4, θ, ω, and i, obtained from both the causal and acausal models. Although the satellite includes two solar panels and two radiators, only one of each is shown, due to the thermal symmetry of the system.

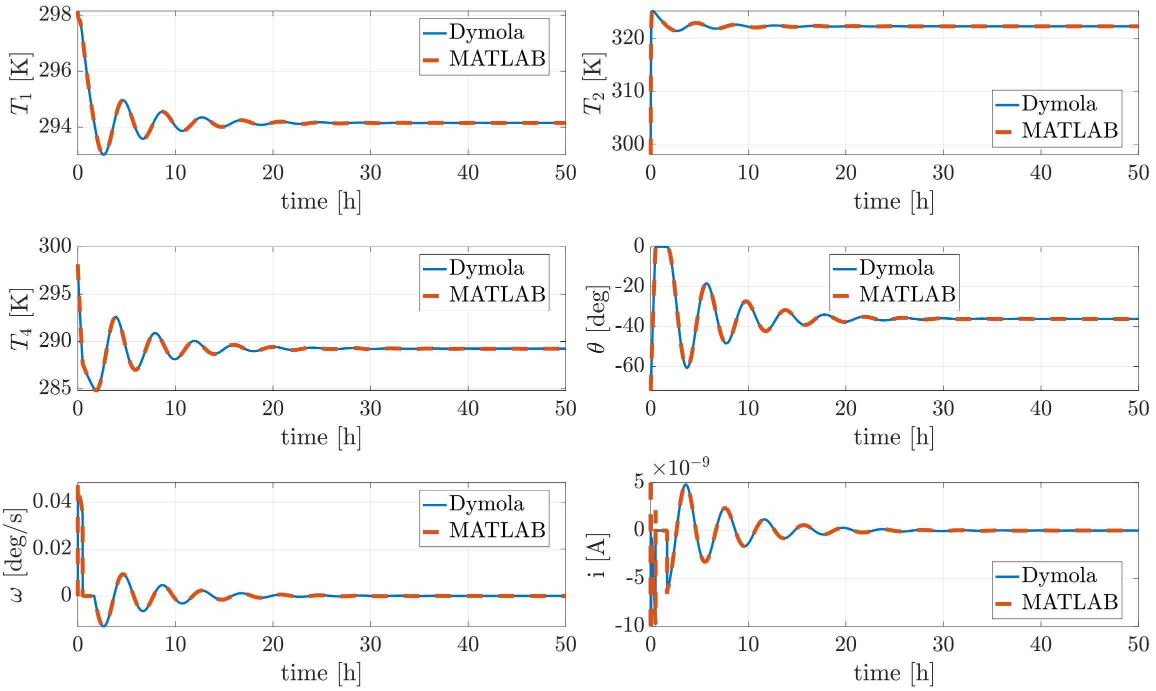

Figure 12: Electro-mechanical model in Dymola

Figure 12 shows an almost perfect overlap between the causal and acausal solutions, demonstrating the robustness and correctness of the implemented acausal model.

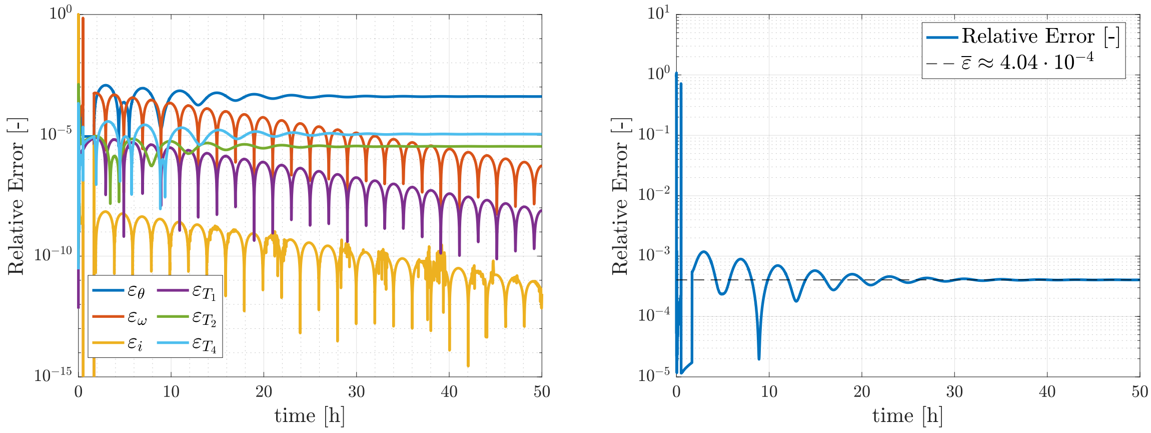

Figure 13: Time evolution of the relative error: component-wise (left) and overall norm (right).

A more detailed analysis was performed by examining the relative error between the two approaches, as shown in Figure 13. To make this comparison possible, it was necessary to evaluate the state variables obtained from the acausal simulation at the same time instants as those from the causal simulation. This was achieved by interpolating the *OpenModelica* output using MATLAB's `interp1` function, which computes a piecewise linear interpolating polynomial. The relative error for each component of the state vector was then computed using the formula in Equation 6:

$$\varepsilon_i = \frac{|\mathbf{x}_i\,(t) - \overline{\mathbf{x}}_i\,(t)\,|}{\max\left(|\mathbf{x}_i\,(t)\,|\right)} \tag{6}$$

where $\mathbf{x}_i(t)$ denotes the i-th component of the state vector computed in MATLAB, and $\overline{\mathbf{x}}_i(t)$ is its *OpenModelica* counterpart. The denominator was chosen as $\max\left(|\mathbf{x}_i(t)|\right)$ rather than $|\mathbf{x}_i(t)|$ to avoid divisions by zero, since some components may be zero at specific time instants.

As shown in Figure 13, the relative error is highest during the initial transient phase, when the state derivatives are largest. As the system approaches steady state, the relative error also stabilizes, settling at approximately $4.04 \cdot 10^{-4}$. The component-wise analysis reveals that the steady-state error is primarily dominated by the contribution from θ, while the errors in other components are significantly smaller. Conversely, during the initial transient, the error is mainly driven by the current, which exhibits the highest rate of change due to the strong initial control action caused by the large difference between T_1 and T_{ref}.

The results demonstrate that the two approaches yield nearly identical solutions, as evidenced by the almost perfect overlap of the graphs in Figure 12 and the small relative error. However, modeling the system in Dymola proved particularly straightforward, especially when implementing the active control of the radiator. The control logic was efficiently realized using logical switch blocks, underscoring the simplicity and effectiveness of the acausal modeling approach in this environment.

Exercise 2

A simplified conceptual map of the Attitude Control System (ACS) of a satellite is depicted in Figure 14. For the accuracy of some measurements that the spacecraft has to obtain, the probe has to fly in an orbit very close to the Earth's surface. The spacecraft is therefore subjected to a high atmospheric drag. The ACS has to compensate for the deceleration caused by this drag continuously. In general, ACSs are made of 3 sensors in orthogonal directions and 3 thrusters to detect and compensate for linear and angular accelerations on the probe in each direction, but for the sake of this project, only compensation in the tangential direction will be considered.

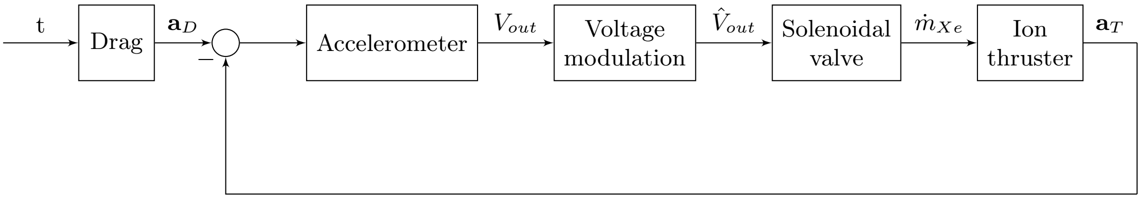

Figure 14: Block diagram of the ACS.

The ACS is modeled with several modules: an **accelerometer** measuring the accelerations acting on the spacecraft, a **voltage modulation block** controlling the output voltage, the **control valve** modifying the aperture of the thruster valve, and the **ion thruster**.

The accelerometer, kept aligned with the direction of velocity, presents inside a seismic mass that is subject to an external acceleration $\mathbf{a_D}$ and $\mathbf{a_T}$ given only by the drag D and the thrust T, respectively. The mass is in between the stators of a condenser and, moving for the acceleration given by these two forces changes the output voltage V_{out} of the circuit of the accelerometer. The dynamics of the seismic mass itself can be modeled as a mass-spring-damper system as follows

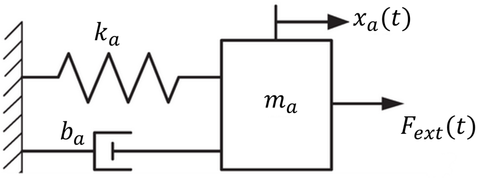

Figure 15: Model of the accelerometer.

$$\dot{x}_a = v_a \tag{7}$$

$$F_{ext}(t) = \frac{T - D}{M_{SC}} m_a = m_a \dot{v}_a + b_a v_a + k_a x_a \tag{8}$$

where M_{SC} is the mass of the spacecraft, m_a is the seismic mass value, and b_a and k_a are respectively the damper and the spring terms present in the accelerometer. For the causal modeling, consider the V_{out} as directly proportional to the velocity of the seismic mass, as $V_{out} = K_{acc} v_a$ [1].

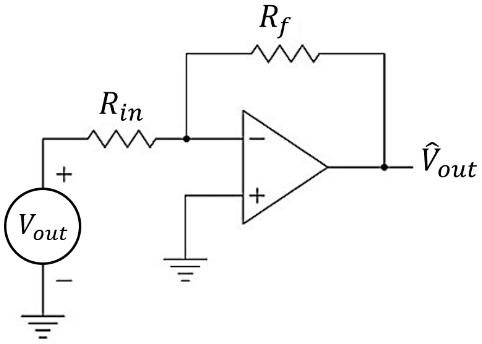

Figure 16: Inverting operational amplifier.

The output of the accelerometer V_{out} needs to be modulated into $\hat{V}_{out}$ to adjust the control for the solenoidal valve. This modulation is performed through an operational amplifier, displayed in Figure 16. The operational amplifier in inverting configuration modifies V_{out} into $\hat{V}_{out}$ accordingly to:

$$\hat{V}_{out} = -\frac{R_f}{R_{in}} V_{out} \tag{9}$$

where R_f and R_{in} are the two resistances modifying the input voltage.

The voltage $\hat{V}_{out}$ is itself the input of the solenoidal valve, shown in Figure 17, and creates a current I commanding the flow control valve. The current passes through a circuit modeled with only a solenoid with a variable inductance $L(x_v)$. As the current variably flows, a magnetic field is generated. A resulting force f_v acts on the spool, which as a result moves.

The armature-spool arrangement is then modeled again as a lumped parameter spring-mass-damper system

$$\dot{x}_v = v_v \tag{10}$$

$$m_v \dot{v}_v = -k_v x_v - b_v v_v + \frac{1}{2} I^2 \frac{dL}{dx_v} \tag{11}$$

$$\dot{I} = \frac{1}{L(x_v)} \hat{V}_{out} \tag{12}$$

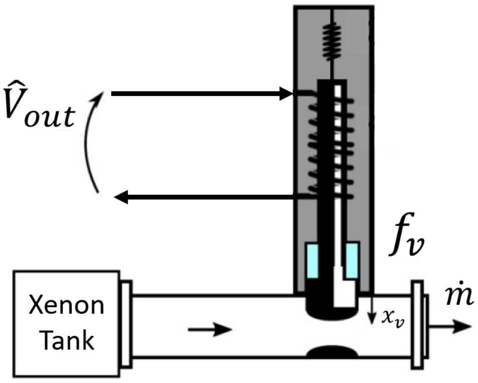

Figure 17: Model of the solenoidal valve.

where x_v and v_v represent the position and velocity of the spool, and b_v and k_v are respectively the damper and the spring terms, and m_v is the mass of the spool. The dynamics of the current I is generated by $\hat{V}_{out}$ through the variable inductance $L(x_v) = \frac{1}{\alpha+\beta x_v}$, depending on the position of the valve itself.

As the spool moves, the area $A_v(x_v)$ of the duct through which a Xenon flow is passing changes. The resulting mass flow rate $\dot{m}_{Xe}$ of Xenon is ejected into the ion thruster, shown in Figure 18. In particular, the area of the valve can be modeled as

$$A_v(x_v) = A_0 + \ell(x_{v,\max} - x_v) \tag{13}$$

In particular, the orifice area $A_v(x_v)$ has been modeled as linearly varying with x_v. The tool controlling the area is a flap of width ℓ and maximum extension $x_{v,\max} = \ell$, for simplicity. The value A_0 is the minimum area of the orifice, such that:

$$0 \leq x_v \leq x_{v,\max}$$

$$\underbrace{A_0 = A_v(x_{v,\max})}_{\text{minimum area}} \leq A_v \leq \underbrace{A_v(0) = \; A_0 + \ell^2}_{\text{maximum area}} \tag{14}$$

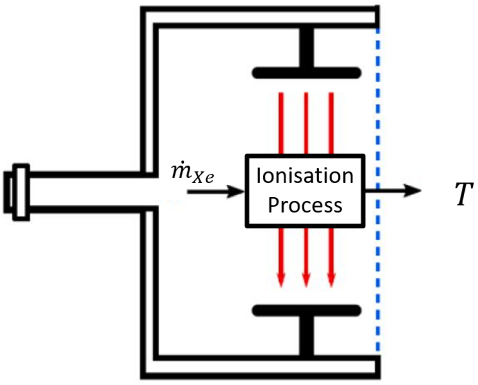

Figure 18: Model of the ion thruster.

The ion thruster comprises an ionization chamber that utilizes a strong magnetic field, ionizing the Xenon flux. The generated ions are moved towards the acceleration grid, which accelerates the ions using a strong electric field, producing a thrust on the spacecraft as the final effect. The flux of Xenon $\dot{m}_{Xe}$ can be expressed as follows

$$\dot{m}_{Xe} = A_v(x_v)\sqrt{k\rho_{Xe}p_T(\frac{2}{k+1})^{\frac{k+1}{k-1}}} \tag{15}$$

where $\rho_{Xe} = \dfrac{p_T}{\bar{R}T_T}$, k and $\bar{R}$ being respectively Xenon specific heat ratio and gas constant. In the tank, the gas is at a total pressure of p_T and temperature of T_T. The Xenon flow enters the thruster and is here ionized into ions with mass m_i and charge q. After the acceleration imposed by the acceleration grid with a voltage ΔV, the ions exit from the nozzle with a velocity v_{exit}, producing the thrust T, as follows

$$T = \dot{m}_{Xe} v_{ext} = \dot{m}_{Xe} \sqrt{\frac{2q\Delta V}{m_i}} \tag{16}$$

Table 4 reports the values for the parameters to simulate the ACS system, with symbol and unit of measure. The drag $D(t)$ can be modeled as a function of time only, as it follows

$$D(t) = 2.2 - \cos(\omega_s t) + 1.2 \sin(\omega_o t) \cos(\omega_o t) \tag{17}$$

where t enters in seconds, and $D(t)$ is in mN.

Component	Parameter	Symbol	Value	Unit
Accelerometer	Spacecraft mass	M_{SC}	300	kg
	Seismic mass	m_a	0.32	kg
	Accelerometer damper	b_a	$[1.5{\cdot}10^3$–$2{\cdot}10^4]$	Ns/m
	Accelerometer spring	k_a	$[5{\cdot}10^{-5}$–$3{\cdot}10^{-3}]$	N/m
	Acc. proportional coefficient	K_{acc}	1	Vs/m
Amplifier	Inverting resistance	R_{in}	[0.1–10]	Ω
	Feedback resistance	R_f	$[1{\cdot}10^4$–$8{\cdot}10^4]$	Ω
Solenoidal Valve	Spool mass	m_v	0.1	kg
	Valve spring	k_v	$1{\cdot}10^3$	N/m
	Valve damper	b_v	$1{\cdot}10^3$	Ns/m
	Solenoid constant	α	$2.1{\cdot}10^{-2}$	1/H
	Solenoid gain	β	-60	1/Hm
	Minimum area	A_0	$4.7{\cdot}10^{-12}$	m^2
	Maximum extension	$x_{v,\max}$	$1{\cdot}10^{-5}$	m
Thruster	Heat ratio	k	1.66	-
	Tank pressure	p_T	$2{\cdot}10^5$	Pa
	Tank temperature	T_T	240	K
	Gas constant	$\bar{R}$	63.32754	J/kg K
	Charge	q	$1.6{\cdot}10^{-19}$	C
	Voltage	ΔV	2000	V
	Ion mass	m_i	$2.188{\cdot}10^{-25}$	kg
Drag	Secular pulsation	ω_s	$1.658226{\cdot}10^{-6}$	rad/s
	Orbital pulsation	ω_o	$1.160758{\cdot}10^{-3}$	rad/s

Table 4: Parameters for the simulation of the ACS.

Part 1: causal modeling (9 points)

Reproduce in Matlab the physical model of the ACS, in particular

1. Formulate the full system of non-linear ODEs making explicit the state variables of the whole simulation. Tune the values of the parameters not explicitly stated in Table 4 to have the compensating thrust T the most similar to the disturbing drag D.

2. Choose the appropriate initial conditions and simulate for three orbital periods T_o, where $T_o = 2\pi/\omega_o$.

3. Discuss the selection of the ODE integration scheme.

Part 2: acausal modeling (6 points)

Reproduce in Simscape the physical model of the ACS. Choose the appropriate initial conditions and parameters, simulate for three orbital periods T_o, and compare the results with those obtained in the prior point.

Note: when modeling the solenoid valve in Simscape, please consider the nominal values shown in Table 5.

Parameter	**Value**	**Unit**
Pull-in forces [F_1 - F_2]	[9000 - 12]	N
Stroke [x_1 - x_2]	[0 - 0.1]	mm
Maximum Stroke	0.1	mm
Rated voltage	0.6	mV
Rated current	0.1	A
Contact stiffness	0.12×10^6	N/m
Contact damping	10^4	Ns/m

Table 5: Solenoid parameters in Simscape.

(15 points)

3 Part 3: Causal Modeling

3.1 Mathematical modeling

References

[1] R. Mukhiya, M. Garg, P. Gaikwad, et al. *Electrical equivalent modeling of MEMS differential capacitive accelerometer*. In: Microelectronics Journal. Vol. 99, pp. 104770, 2020. `https://doi.org/10.1016/j.mejo.2020.104770`