MSAS – Assignment #2: Modeling

Tarun Singh, 10894156

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.

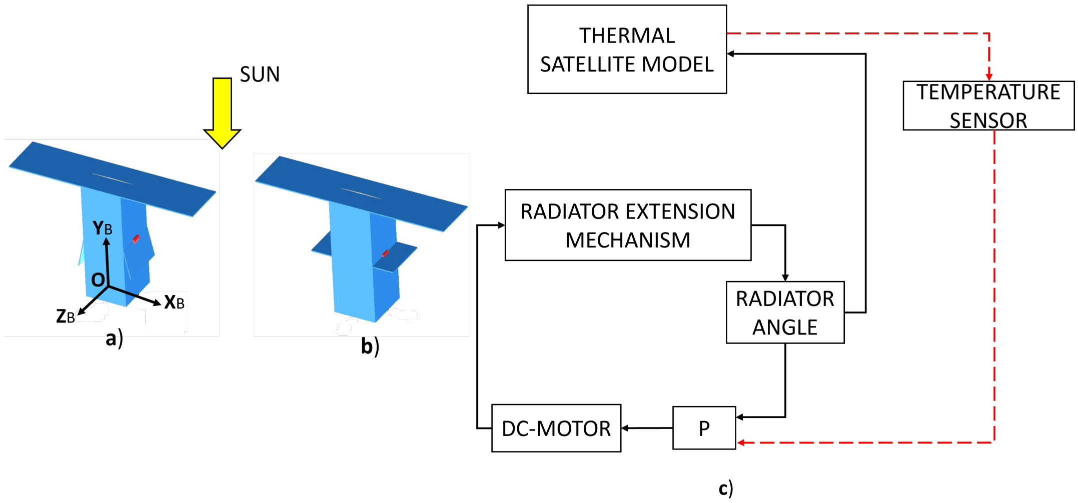


Figure 1: Satellite configuration: **a)** Closed radiator; **b)** Open radiator; **c)** Thermal control logic overview

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

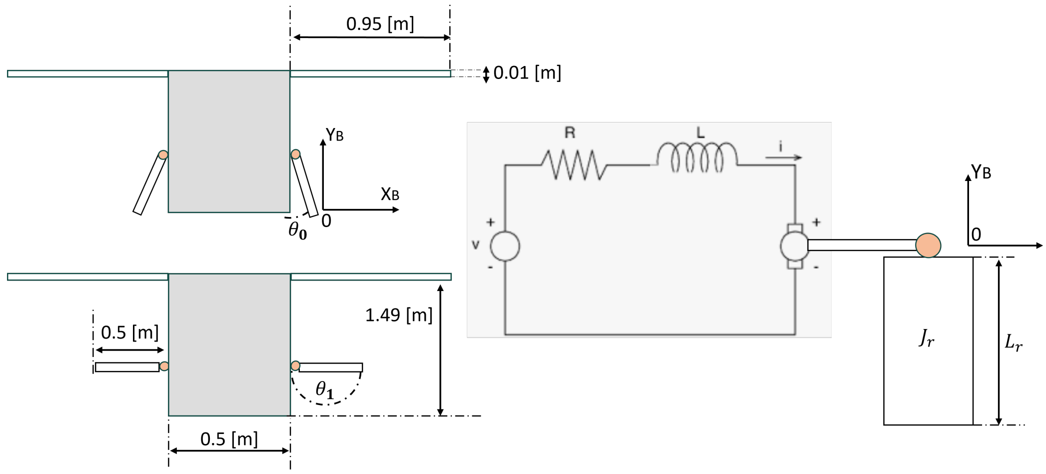


Figure 2: Schematics of the radiator extension mechanism

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).

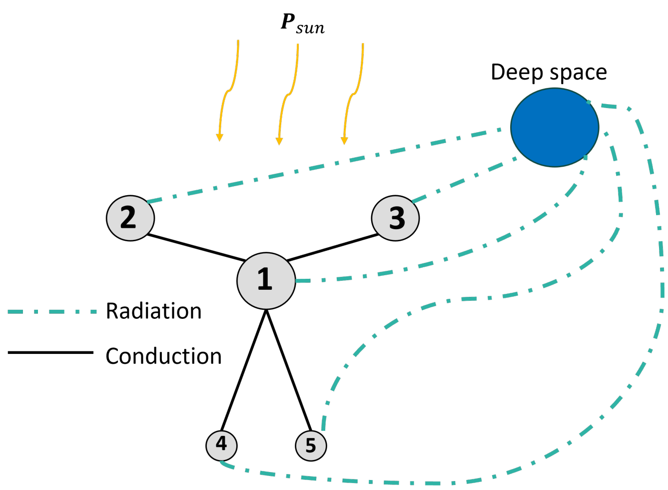


Figure 3: The thermal lumped nodes model

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.

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.

Part 1 Answer: Causal modeling

1. The following formulation of equations represents the complete system of non-linear ODEs for the given model, along with its state variables:

   - Thermal model formulation:

$$Q_{sun_i} = P_{sun} \cdot \alpha_i \cdot A_{s_i} \qquad i = \{1, 2, 3\} \tag{5}$$

$$Q_{rad_i} = \epsilon_i \cdot \sigma \cdot A_{r_i} \cdot (T_i^4 - T_{ds}^4) \qquad i = \{1, ..., 5\} \tag{6}$$

$$Q_{c_i} = G_{1i} \cdot (T_i - T_1) \qquad i = \{2, ..., 5\} \tag{7}$$

$$\frac{dT_1}{dt} = \frac{1}{C_1}(\sum_{i=2}^{5} Q_{c_i} + Q_{sun_1} + Q_{rad_1}) \tag{8}$$

$$\frac{dT_i}{dt} = \frac{1}{C_i}(-Q_{c_i} + Q_{sun_i} + Q_{rad_i}) \qquad i = \{2, 3\} \tag{9}$$

$$\frac{dT_i}{dt} = \frac{1}{C_i}(-Q_{c_i} - Q_{rad_i}) \qquad i = \{4, 5\} \tag{10}$$

     Here A_s represents the area exposed to the sun, A_r denotes the radiative area exposed to deep space, and T_i is the temperature of node i.

   - Control model formulation:

$$\frac{d\theta}{dt} = \dot{\theta} \tag{11}$$

$$\frac{d\dot{\theta}}{dt} = \frac{1}{J_r}(k_m \cdot i) \tag{12}$$

$$\frac{di}{dt} = \frac{1}{L}(V_{in} - i \cdot R - k_m \cdot \dot{\theta}) \tag{13}$$

   - The thermal model and control model together represent the satellite's cooling system. Eq. (8) to Eq. (13) collectively represent the non-linear ODE formulation for the entire simulation model. The state vector for the complete simulation model is $\{T_1, T_2, T_3, T_4, T_5, \theta, \dot{\theta}, i\}$, where i represents the current in the electrical circuit of the control system.

2. For $k_p = 4 \times 10^{-5}$, the main body temperature (T_1) converges to the target value $T_1^{ref} = 294.15$ $[K]$. After $t = 10$ $[h]$, the maximum oscillations of T_1 remain within 0.1% of T_1^{ref}. The value $k_p = 4 \times 10^{-5}$ was determined through iterative tuning to minimize overshoot and ensure convergence within the required oscillation bounds. Fig. 4 shows the temperature evolution of all the nodes. Run MATLAB script **Singh10894156_Assign2_EX1** corresponding to exercise 1 from zipped folder to obtain the simulation results illustrated from Fig. 4 to Fig. 6.

3. Fig. 5 illustrates the control of the radiator panels to maintain the main body temperature within the required limits. Initially, T_1 is significantly higher than T_1^{ref}. To achieve the desired temperature range, the radiators quickly transition from the stowed configuration to the fully open configuration. The large temperature deviation at the start triggers the control system to fully open the radiators, maximizing radiative cooling. As T_1 approaches T_1^{ref}, the control system gradually reduces the radiator angle to stabilize the temperature.

   Thermal dynamics evolve very slowly due to heat transfer processes involving large thermal capacities C_i of the nodes, which result in small eigenvalues in the thermal model.

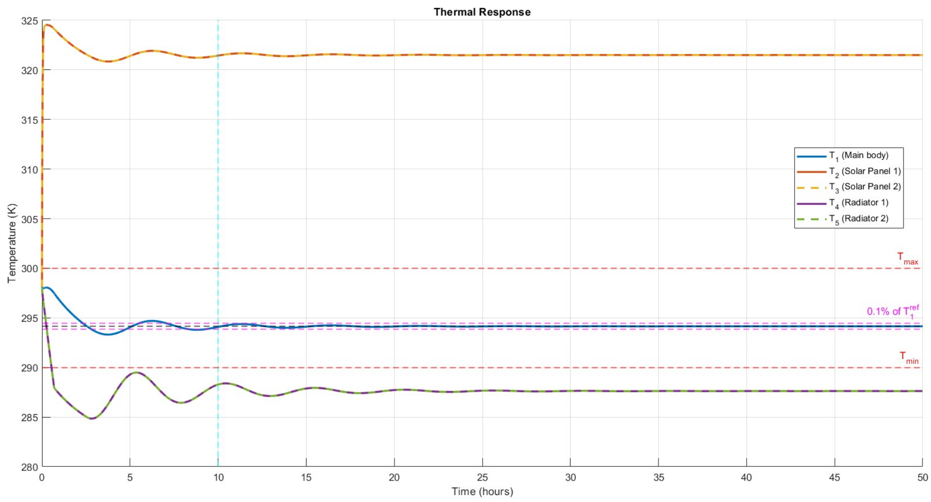
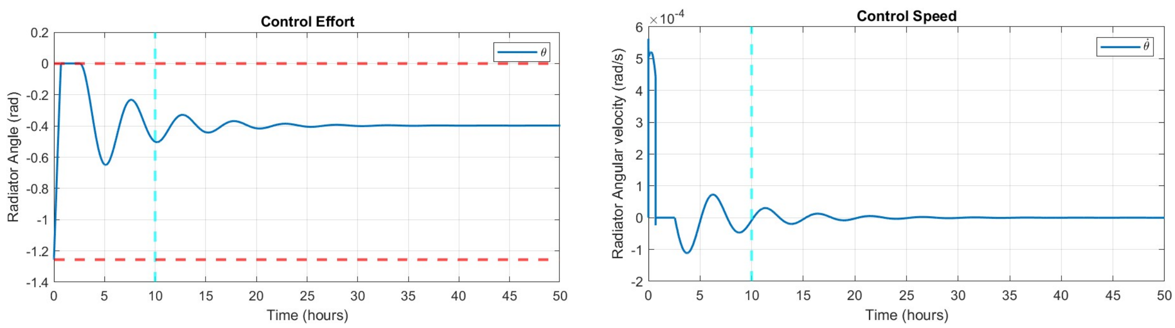


Figure 4: Evolution of the temperature of all the nodes (MATLAB).

Conversely, electrical and mechanical dynamics evolve rapidly because of the small inductance L and resistance R, leading to a fast time constant ($\tau = \frac{L}{R}$) and consequently large control model eigenvalues. This significant disparity in eigenvalues across the complete model makes the system of non-linear ODEs inherently stiff.

In MATLAB, `ode15s` is the most suitable solver for this stiff system for several advantages. It is a variable-order solver (1 to 5) that efficiently handles the strong non-linearity introduced by the T^4 term in radiative heat transfer. Additionally, it is well-suited for managing stiffness over a long integration span of 50 hours without compromising accuracy.

Fig. 6 illustrates the temperature evolution of the main body and current in the electro-mechanical control system, respectively.



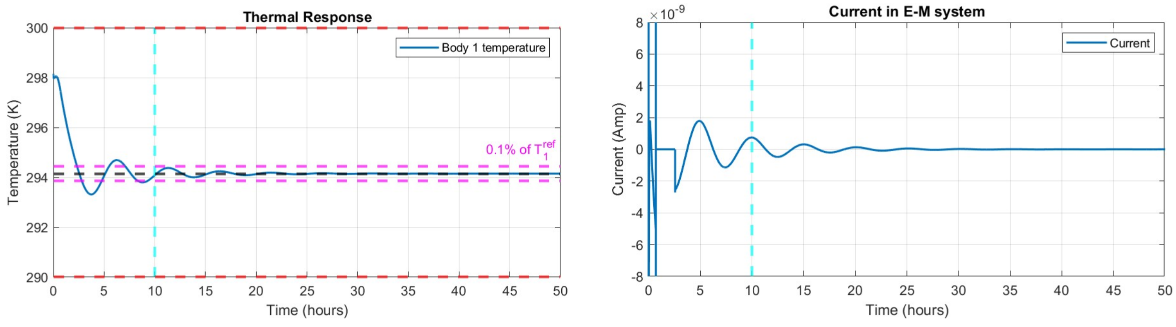
(a) Radiator angle. (b) Radiator angular speed.

Figure 5: Transient response of the radiator control system (MATLAB).

Part 2 Answer: Acausal modeling

Fig. 7 to Fig. 9 represent the results obtained from the OpenModelica simulation, which can be obtained from running the OpenModelica file **Singh10894156_Assign2_EX1** from the zipped folder for 180000 seconds.

Note: OpenModelica does not save the plots in the file itself upon closing the software.



(a)

(b)

Figure 6: (a) Main body temperature evolution; (b) Current in the electric circuit (MATLAB).

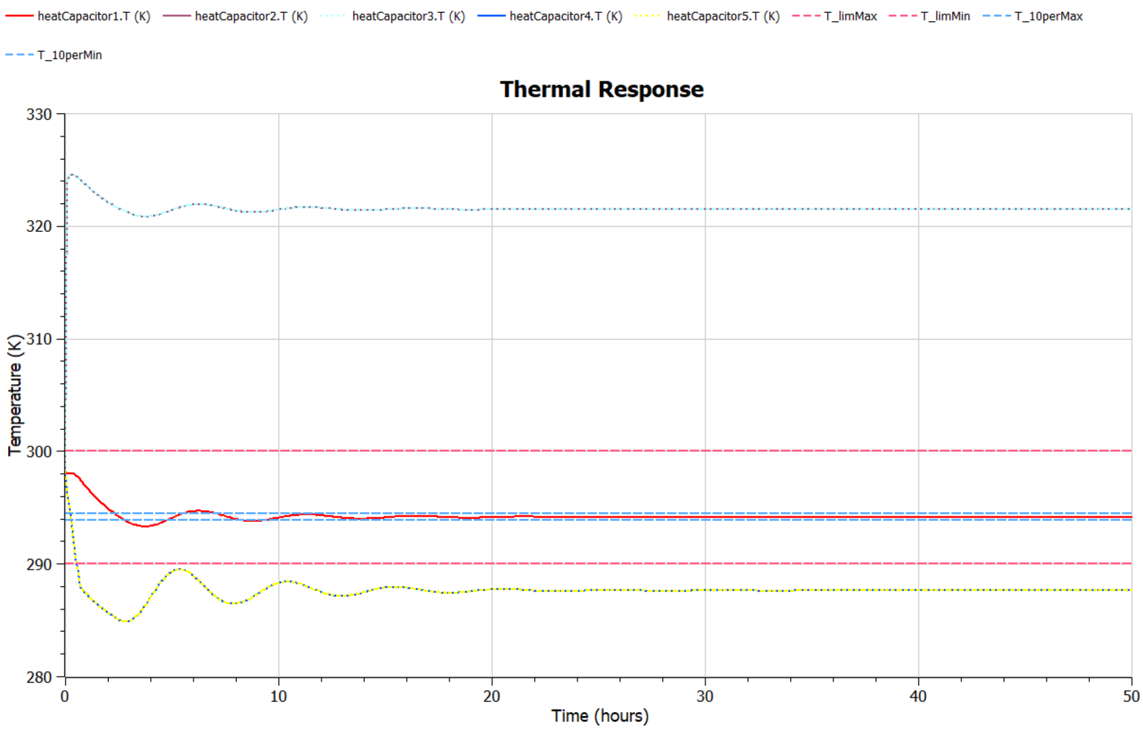
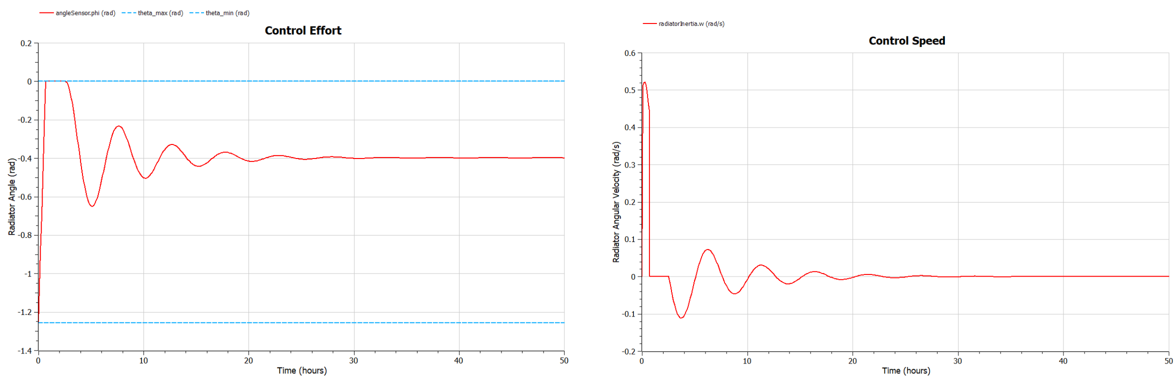


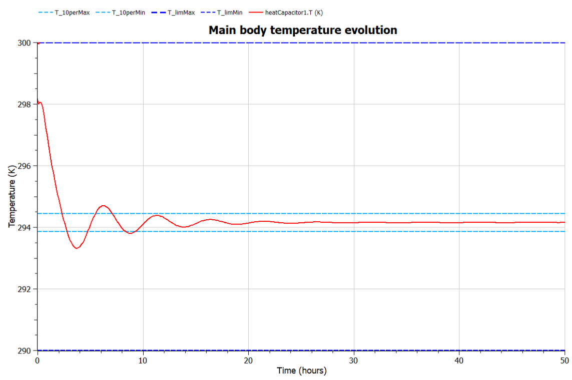
Figure 7: Evolution of the temperature of all the nodes (OpenModelica).



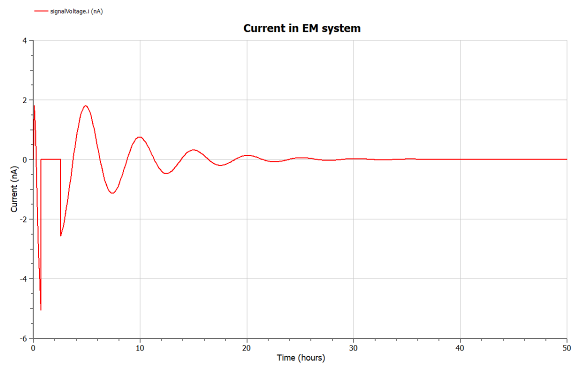
(a) Radiator angle.

(b) Radiator angular speed.

Figure 8: Transient response of the radiator control system (OpenModelica).



(a)



(b)

Figure 9: (a) Main body temperature evolution; (b) Current in the electric circuit (OpenModelica).

Exercise 2

A simplified conceptual map of the Attitude Control System (ACS) of a satellite is depicted in Figure 10. 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.

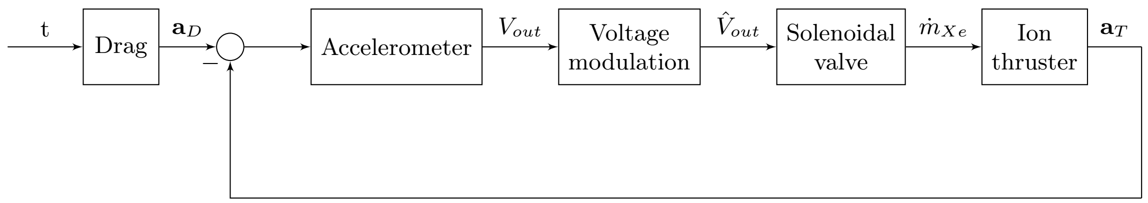


Figure 10: Block diagram of the ACS.

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

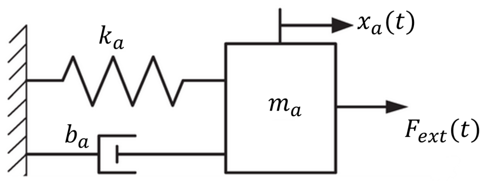


Figure 11: Model of the accelerometer.

$$\dot{x}_a = v_a \tag{14}$$

$$F_{ext}(t) = \frac{T - D}{M_{SC}} m_a = m_a \dot{v}_a + b_a v_a + k_a x_a \tag{15}$$

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].

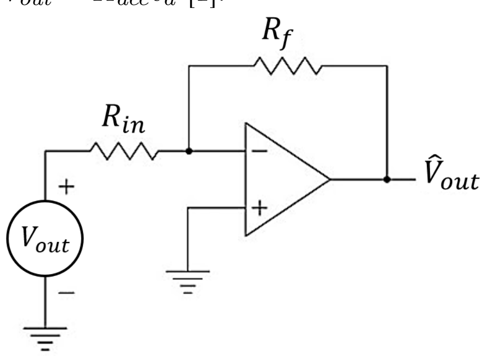


Figure 12: Inverting operational amplifier.

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 12. The operational amplifier in inverting configuration modifies V_{out} into $\hat{V}_{out}$ accordingly to:

$$\hat{V}_{out} = -\frac{R_f}{R_{in}} V_{out} \tag{16}$$

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 13, 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

$$\dot{x}_v = v_v \tag{17}$$

$$m_v \dot{v}_v = -k_v x_v - b_v v_v + \frac{1}{2} I^2 \frac{dL}{dx_v} \tag{18}$$

$$\dot{I} = \frac{1}{L(x_v)} \hat{V}_{out} \tag{19}$$

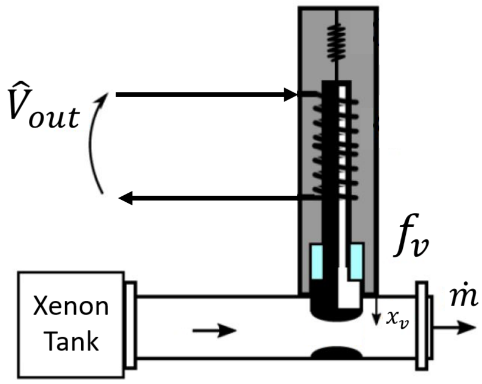


Figure 13: Model of the solenoidal valve.

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 14. In particular, the area of the valve can be modeled as

$$A_v(x_v) = A_0 + \ell(x_{v,\max} - x_v) \tag{20}$$

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:

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

$$\underbrace{A_0 = A_v(x_{v,\max})}_{\text{minimum area}} \leq A_v \leq \underbrace{A_v(0) = A_0 + \ell^2}_{\text{maximum area}} \tag{21}$$

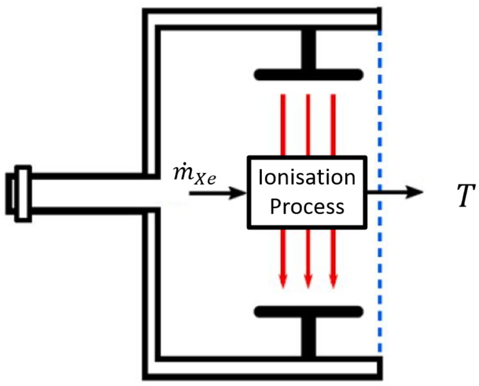


Figure 14: Model of the ion thruster.

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

$$\dot{m}_{Xe} = A_v(x_v) \sqrt{k \rho_{Xe} p_T \left(\frac{2}{k+1}\right)^{\frac{k+1}{k-1}}} \tag{22}$$

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

$$T = \dot{m}_{Xe} v_{ext} = \dot{m}_{Xe} \sqrt{\frac{2q\Delta V}{m_i}} \tag{23}$$

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

$$D(t) = 2.2 - \cos(\omega_s t) + 1.2 \sin(\omega_o t) \cos(\omega_o t) \tag{24}$$

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.

Part 1 Answer: Causal modeling

1. The following formulation of equations represents the complete system of non-linear ODEs for the given model, along with its state variables:

$$\frac{dx_a}{dt} = v_a \tag{25}$$

$$\frac{dv_a}{dt} = \frac{T - D}{M_{SC}} - \frac{1}{m_a}(b_a v_a + k_a x_a) \tag{26}$$

$$\frac{dx_v}{dt} = v_v \tag{27}$$

$$\frac{dv_v}{dt} = \frac{1}{m_v}[-k_v x_v - b_v v_v + \frac{1}{2}I^2(-\frac{\beta}{(\alpha + \beta x_v)^2})] \tag{28}$$

$$\frac{dI}{dt} = \hat{V}_{out}(\alpha + \beta x_v) \tag{29}$$

Eq. (25) to Eq. (29) collectively represent the non-linear ODE formulation for the simulation, capturing the coupled dynamics of the ACS. Thrust T and Drag D can be calculated from Eq. (23) and Eq. (24), respectively. The state vector for the complete simulation model is $\{x_a, v_a, x_v, v_v, I\}$, where I represents the current in the electrical circuit of the solenoid valve. Table 6 presents the values of parameters not explicitly stated in Table 4, tuned to make the compensating thrust T as similar as possible to the disturbing drag D.

Parameter	Symbol	Value	Unit
Accelerometer damper	b_a	$1.5 \cdot 10^3$	Ns/m
Accelerometer spring	k_a	$5 \cdot 10^{-5}$	N/m
Inverting resistance	R_{in}	0.1	Ω
Feedback resistance	R_f	$8 \cdot 10^4$	Ω

Table 6: Tuned parameters.

2. The selected initial conditions are shown in Table 7 and ensure the system starts from a physically realistic state.

Variable	Symbol	Initial Value	Unit
Accelerometer position	x_a	0	m
Accelerometer velocity	v_a	0	m/s
Spool position	x_v	0	m
Spool velocity	v_v	0	m/s
Solenoid current	I	0	Amp

Table 7: Initial condition of the state vector.

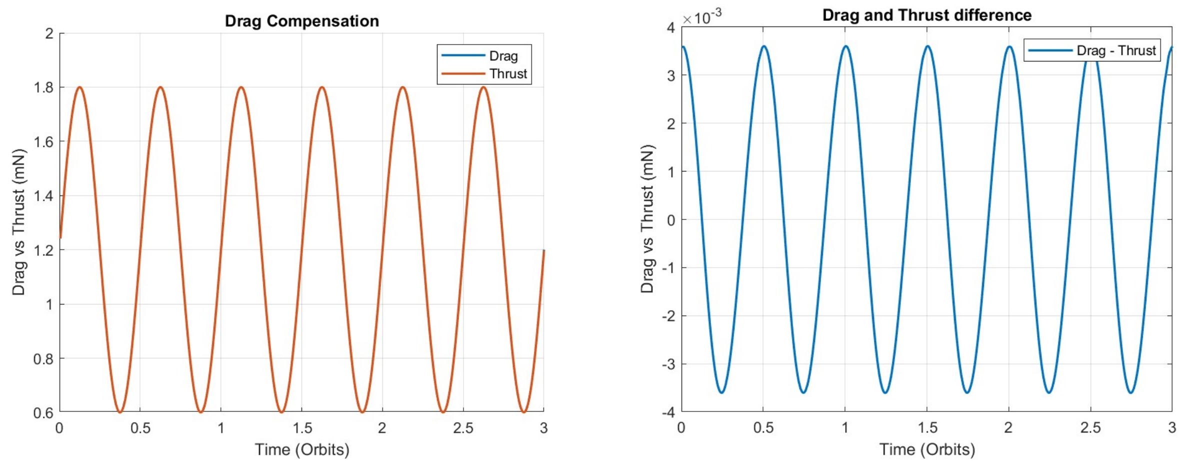
These values represent a stationary accelerometer mass and a realistic starting state for the solenoid at the beginning of the simulation.

Run MATLAB script **Singh10894156_Assign2_EX2** corresponding to exercise 2 from zipped folder to obtain the simulation results illustrated from Fig. 15 to Fig. 18.

3. Components such as the accelerometer and solenoid valve may exhibit rapid oscillations, resulting in a stiff system. The problem also includes a feedback loop between drag and thrust compensation. Furthermore, the simulation spans multiple orbital periods,

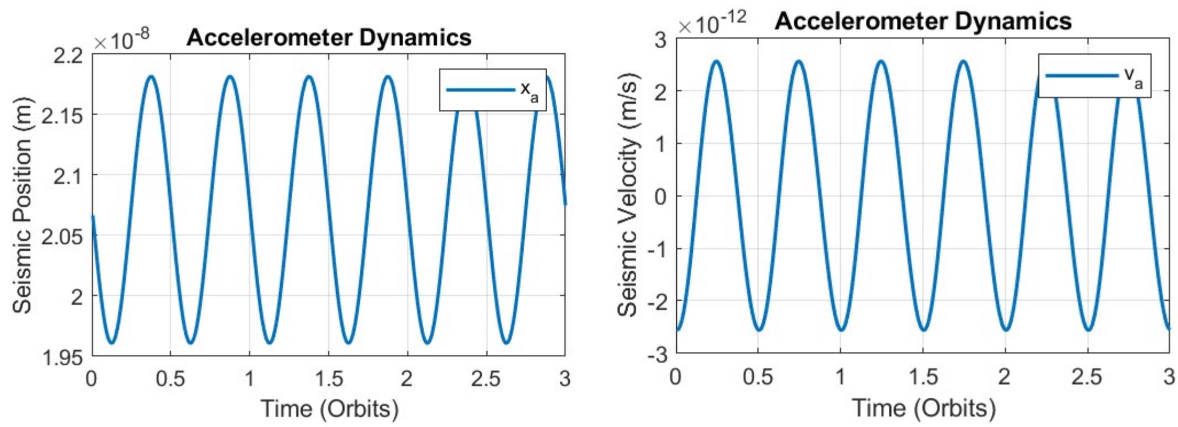
involving both fast dynamics (seismic mass and spool movements) and slower dynamics (overall thrust evolution).

`ode15s` is the ideal choice for this stiff problem, as it efficiently handles feedback loops and fast transient responses. It manages stiffness by dynamically adapting the step size and order of the integration method. Additionally, `ode15s` reduces the computational effort for stiff problems by utilizing backward differentiation formulas (BDFs).



(a) Drag and Thrust. (b) Drag compensation residual.

Figure 15: Drag and Thrust results from MATLAB for 3 orbital transient runs.



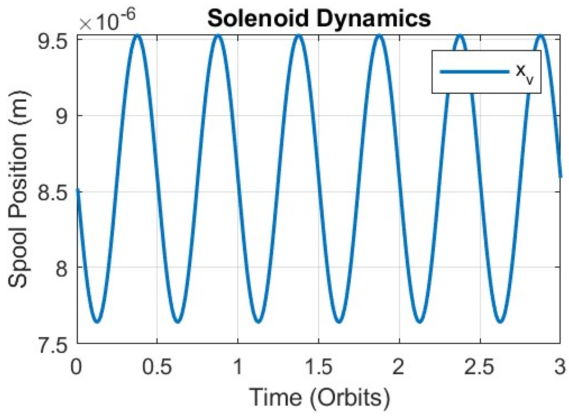
(a) Position (m). (b) Velocity (m/s).

Figure 16: Accelerometer Dynamics (MATLAB).

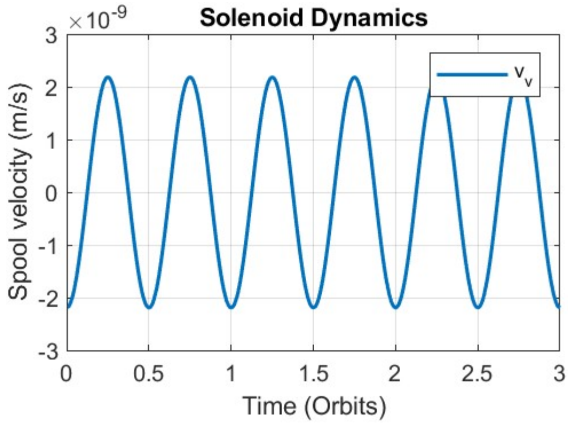
Part 2 Answer: Acausal modeling

Fig. 19 to Fig. 22 represent the results obtained from the Simscape simulation, which can be obtained from running the Simscape file **Singh10894156_Assign2_EX2** and then **Simulink_plotting** from the zipped folder.

Note: Simscape plots can also be inspected using the **Scope** block provided in the model. In some cases, it may be necessary to use the zoom cursor to examine the results, as they might initially appear as a straight line due to the initial spike in the values..

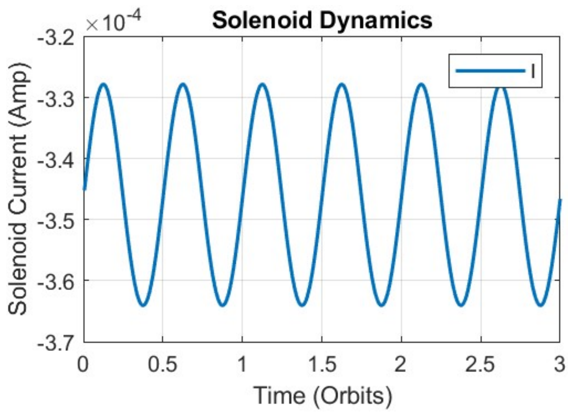


(a) Position (m).

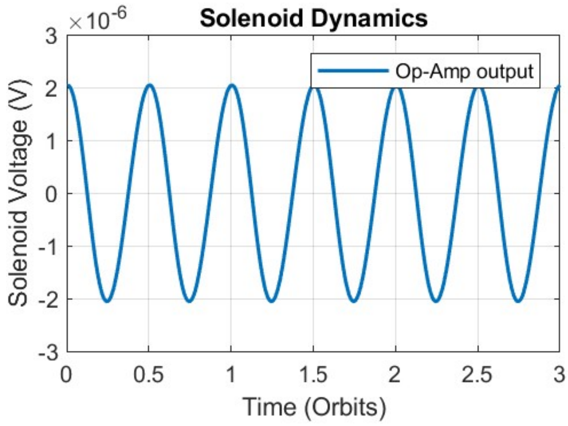


(b) Velocity (m/s).

Figure 17: Solenoid Spool Dynamics (MATLAB).

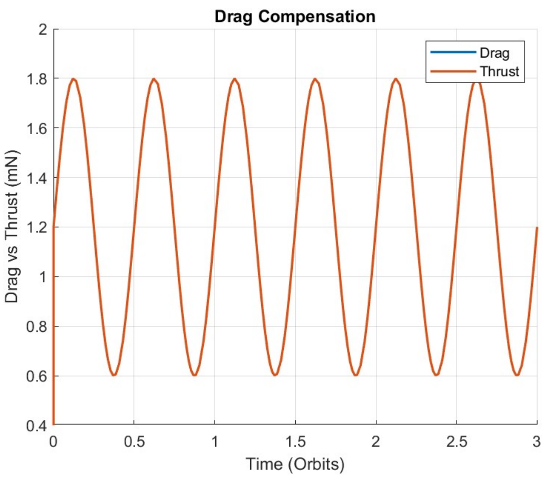


(a) Current (Amp).

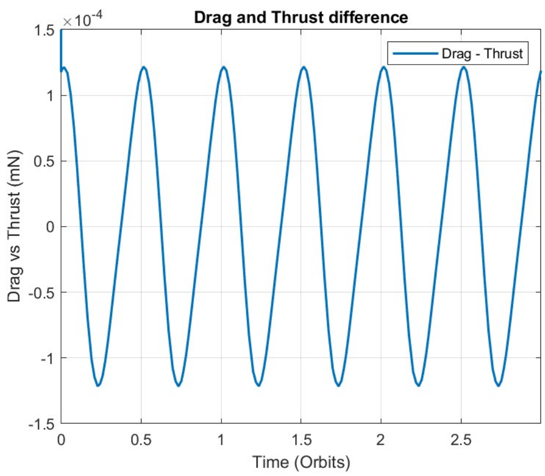


(b) Voltage (V).

Figure 18: Solenoid Armature Dynamics (MATLAB).

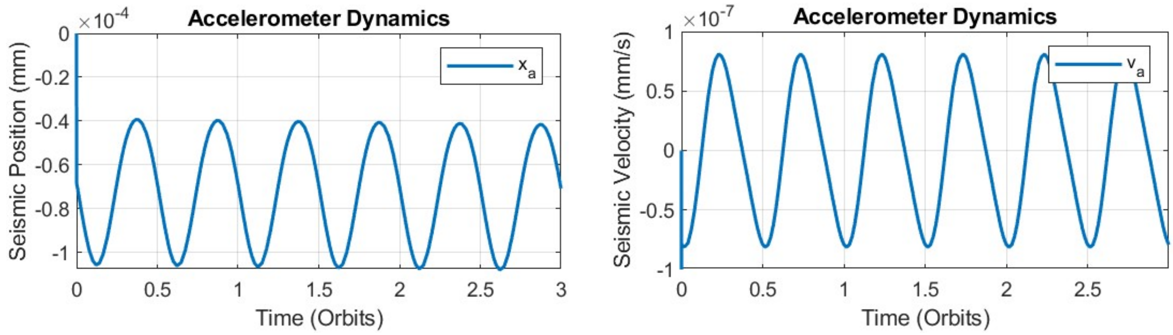


(a) Drag and Thrust.



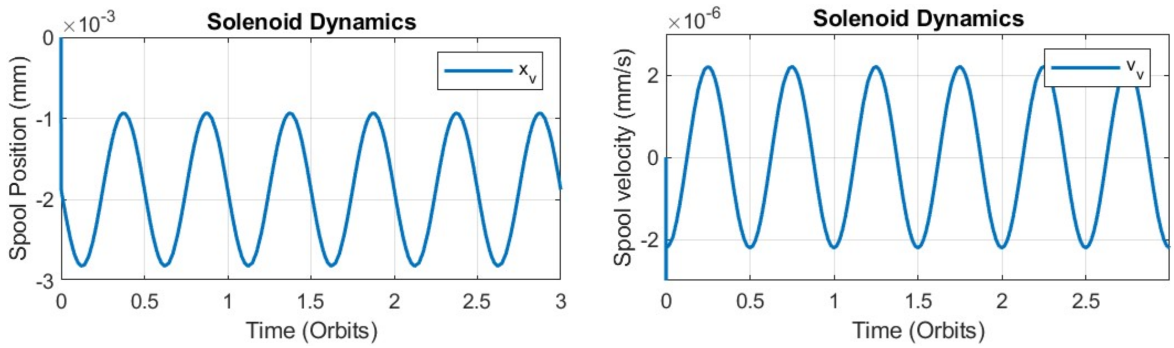
(b) Drag compensation residual.

Figure 19: Drag and Thrust results from MATLAB for 3 orbital transient runs.



(a) Position (mm). (b) Velocity (mm/s).

Figure 20: Accelerometer Dynamics (Simscape).



(a) Position (mm). (b) Velocity (mm/s).

Figure 21: Solenoid Spool Dynamics (Simscape).

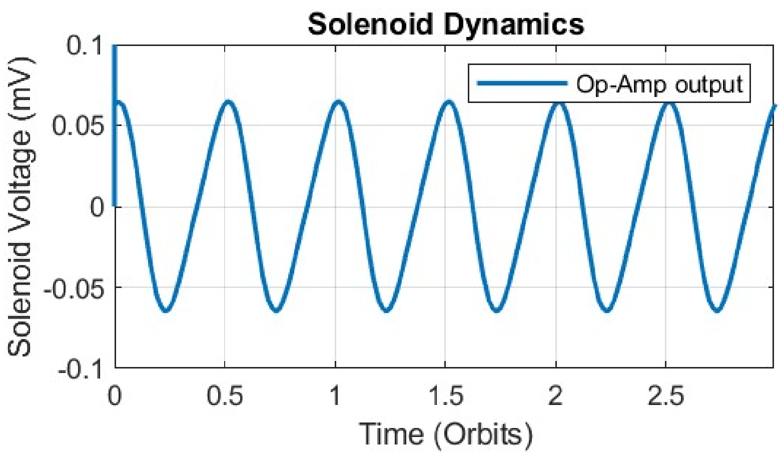


Figure 22: Solenoid Armature Dynamics (Simscape)

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`