aerospace

MDPI

Article

Fuel-Efficient and Fault-Tolerant CubeSat Orbit Correction via Machine Learning-Based Adaptive Control

Mahya Ramezani, Mohammadamin Alandihallaj * and Andreas M. Hein

Interdisciplinary Centre for Security, Reliability and Trust (SnT), University of Luxembourg, 2 Avenue de l'Universite L, 4365 Esch-sur-Alzette, Luxembourg; mahya.ramezani@uni.lu (M.R.); andreas.hein@uni.lu (A.M.H.)
* Correspondence: amin.hallaj@uni.lu

Abstract: The increasing deployment of CubeSats in space missions necessitates the development of efficient and reliable orbital maneuvering techniques, particularly given the constraints on fuel capacity and computational resources. This paper presents a novel two-level control architecture designed to enhance the accuracy and robustness of CubeSat orbital maneuvers. The proposed method integrates a J_2-*optimized sequence* at the high level to leverage natural perturbative effects for fuel-efficient orbit corrections, with a gated recurrent unit (GRU)-based low-level controller that dynamically adjusts the maneuver sequence in real-time to account for unmodeled dynamics and external disturbances. A Kalman filter is employed to estimate the pointing accuracy, which represents the uncertainties in the thrust direction, enabling the GRU to compensate for these uncertainties and ensure precise maneuver execution. This integrated approach significantly enhances both the positional accuracy and fuel efficiency of CubeSat maneuvers. Unlike traditional methods, which either rely on extensive pre-mission planning or computationally expensive control algorithms, our architecture efficiently balances fuel consumption with real-time adaptability, making it well-suited for the resource constraints of CubeSat platforms. The effectiveness of the proposed approach is evaluated through a series of simulations, including an orbit correction scenario and a Monte Carlo analysis. The results demonstrate that the integrated J_2-GRU system significantly improves positional accuracy and reduces fuel consumption compared to traditional methods. Even under conditions of high uncertainty, the GRU-based control layer effectively compensates for errors in thrust direction, maintaining a low miss distance throughout the maneuvering period. Additionally, the GRU's simpler architecture provides computational advantages over more complex models such as long short-term memory (LSTM) networks, making it more suitable for onboard CubeSat implementations.

Keywords: CubeSat orbital maneuvering; J_2 perturbation; machine learning-based control; gated recurrent unit (GRU); Kalman filter; fuel efficiency; fault tolerance; adaptive control; orbital correction; spacecraft dynamics

Citation: Ramezani, M.; Alandihallaj, M.; Hein, A.M. Fuel-Efficient and Fault-Tolerant CubeSat Orbit Correction via Machine Learning-Based Adaptive Control. *Aerospace* **2024**, *11*, 807. https://doi.org/10.3390/aerospace11100807

Academic Editors: Paolo Marzioli and Necmi Cihan Örger

Received: 30 August 2024
Revised: 23 September 2024
Accepted: 26 September 2024
Published: 30 September 2024

1. Introduction

The rise of CubeSats has revolutionized the space industry, providing a cost-effective and efficient means to conduct a wide range of scientific and technological missions. Since the first CubeSat was launched in 2003 [1], these small satellites have gained significant popularity due to their compact size, reduced mass, and relatively short development timelines compared to traditional satellites. The adoption of CubeSats is driven largely by their use of commercial-off-the-shelf (COTS) components, which substantially lower the barriers to entry for space missions, making them accessible to academic institutions, small companies, and emerging space-faring nations. CubeSats have been successfully deployed in various applications, including Earth observation [2,3], communication [4], air traffic management [5], ionospheric science [6], active orbital debris removal [7], wildfire detection [8], gravity wave detection [9], and even interplanetary exploration [10].

Despite the advantages offered by CubeSats, their operational deployment poses several challenges, particularly in achieving precise orbital maneuvers. CubeSats are typically launched as secondary payloads, which means that their insertion orbits are not specifically tailored to the mission requirements but rather subject to the constraints of the primary payload [11]. This results in significant orbital variances that need to be corrected for the CubeSat to achieve its designated orbital elements. Such corrections are crucial, especially in missions involving constellations or formations of multiple CubeSats [12,13], where precise coordination and positioning are imperative.

Recent work in the CubeSat domain has mainly explored constellations and control methods. For instance, Jaffer et al. optimized CubeSat constellations for air traffic monitoring through ADS-B signals, demonstrating the potential of small satellite constellations in Earth observation and traffic management applications [14]. Similarly, Gu et al. [15] proposed an extended 2D map for satellite coverage analysis considering elevation-angle constraints, which enhances the accuracy of coverage calculations. Bai et al. presented a multi-stage constant-vector thrust control approach for finite-thrust Lambert transfers, contributing to more efficient orbital maneuver planning in [16]. Additionally, Zhou et al. [17] introduced a parametric formation control method for nanosatellite constellations, emphasizing cooperative observation missions such as those targeting the China Space Station.

One of the primary challenges in orbital maneuvering for CubeSats is the limitation in the onboard fuel capacity. Due to their small size, CubeSats can only carry a limited amount of propellant, which necessitates highly efficient maneuvering strategies to conserve fuel. Moreover, the propulsion systems available for CubeSats generally offer low thrust levels, requiring prolonged operational durations to achieve the necessary changes in velocity (ΔV) for orbit correction [18]. This is further compounded by the limited electric power generated by CubeSats, which restricts the continuous operation of propulsion systems and, consequently, the ability to perform extensive maneuvers. Moreover, it is not practical to use complex control methods such as MPC [19,20] in CubeSats due to onboard processor limitations.

To address the challenge of limited fuel availability onboard small satellites, various orbit insertion and maneuvering techniques have been proposed in the literature to reduce dependency on fuel consumption for orbital corrections. One prominent research area involves utilizing aerodynamic forces to control satellite orbits. This approach typically involves deploying large panels to generate drag forces in specific directions, thereby influencing the satellite's trajectory [21]. For example, the potential of using aerodynamic forces for CubeSat orbit control has been thoroughly examined in [22]. Shao et al. [23] extended this concept to satellite rendezvous scenarios by leveraging aerodynamic forces for orbital adjustments. Similarly, differential drag-based rendezvous maneuvers of multiple spacecraft under J_2 perturbation were analyzed in [24]. The control of satellite formations using lift and drag forces under the influence of J_2 perturbation is further explored in [25]. Varna and Kumar [26] designed a sliding mode controller to manage multiple satellite formation flying using aerodynamic forces. In a similar vein, Prez and Bevilacqua [27] developed a Lyapunov-based adaptive feedback control strategy to regulate satellite orbits through aerodynamic drag. The use of lift forces for satellite formation control is also discussed in [28]. Moreover, Omar and Bevilacqua [29] proposed a time-optimal collision avoidance strategy using aerodynamic drag, which was derived from an analytical solution and iteratively refined using high-fidelity orbit propagators.

In addition to aerodynamic control, several other auxiliary maneuvers have been proposed to further enhance orbit corrections. These include exploiting solar radiation pressure [30,31], third-body perturbations [32], the Moon's gravitational influence [33,34], magnetic disturbances [35,36], tethered system [37], and electromagnetic forces [38,39]. However, the effectiveness of these methods is highly dependent on the specific orbital regime and the relative positions of the Earth, Moon, and Sun.

In addition to fuel and power constraints, the reliance on COTS components introduces variability in manufacturing accuracy and the performance of onboard systems. These variabilities can lead to deviations from the expected performance, further complicating the task of precise orbital maneuvering [40]. The components, while cost-effective, may not always meet the standards required for space operations, leading to unmodeled dynamics and uncertainties during the maneuver execution.

To address these challenges, this paper presents an innovative approach that combines advanced orbital mechanics with state-of-the-art machine learning techniques. Central to this approach is the development of the J_2*-optimized sequence*, a maneuvering strategy that leverages the J_2 perturbation—caused by the Earth's equatorial bulge (oblateness)—to reduce the amount of propellant required for orbit corrections. The J_2 perturbation is a dominant force in low Earth orbits (LEOs) [41], where most CubeSats operate, and can be exploited to assist in achieving desired orbital changes with minimal fuel consumption. The J_2*-optimized sequence* is designed specifically to operate within the constraints of CubeSat missions, optimizing fuel usage while maintaining the desired orbital parameters.

While higher-order zonal harmonics (e.g., J_3 and J_4) and lunar-solar gravitational effects can be incorporated into orbital maneuver planning, these techniques are more suited for higher orbits, long-duration missions, or missions around other planets and asteroids [42,43]. In LEO, the J_2 perturbation is the dominant force, and focusing on it provides significant fuel savings with reduced computational complexity compared to more complex perturbation-based techniques. For CubeSats with limited onboard resources, the J_2-based approach strikes an optimal balance between efficiency and simplicity.

The Earth's gravitational field is not perfectly spherical due to the planet's equatorial bulge, resulting in a perturbation known as the J_2 effect. This oblateness, quantified via the J_2 harmonic coefficient, significantly influences the motion of satellites in LEOs. The J_2 perturbation induces secular changes in orbital elements, most notably in the right ascension of the ascending node (Ω), the argument of perigee (ω), and, to a lesser extent, the inclination. These secular variations are crucial in orbit determination and long-term orbital planning. In particular, the J_2 perturbation causes a precession of the orbital plane, with the rate of change in Ω being directly related to the satellite's altitude and inclination. Recent studies [44,45] have explored the intricate dynamics resulting from Earth's oblateness, offering insights into both short- and long-term effects on satellite orbits. Vallado [46] provided a comprehensive overview of the fundamental effects of J_2 on satellite dynamics, with detailed mathematical derivations that highlight its importance in mission design and trajectory correction. Similarly, Montenbruck and Gill [47] discussed practical methods for accounting for J_2 perturbation in satellite control and navigation systems, offering both theoretical insights and applied strategies for satellite operators. Additionally, Hoots incorporated J_2 effects in the SGP4 propagation model in [48], which is widely used in satellite tracking and prediction, further emphasizing the critical role of Earth's oblateness in satellite mission planning and operations. These references collectively underscore the necessity of accounting for J_2 perturbation in the development of efficient and accurate satellite control strategies.

The inherent uncertainties and unmodeled dynamics in CubeSat operations necessitate a more adaptive and resilient maneuvering approach. To address these challenges, machine learning (ML) emerges as a promising solution, offering the ability to adapt to and mitigate the effects of unmodeled dynamics and uncertainties [49]. Various studies [50–52] have demonstrated the effectiveness of ML techniques in different contexts, highlighting their potential for improving the accuracy and reliability of complex systems under real-world conditions. Within the spectrum of ML approaches, long short-term memory (LSTM) networks are often considered due to their capability to handle sequential data and long-term dependencies [53,54]. However, LSTM networks demand substantial computational resources, which exceed the capabilities of CubeSats, making them impractical for onboard use [55]. To this end, the paper introduces a novel two-level ML framework that leverages a gated recurrent unit (GRU) model [56], an alternative to LSTM, designed to enhance the

accuracy and reliability of orbital maneuvers within the constraints of CubeSat platforms. The GRU is particularly well-suited for this application because it efficiently processes time-sequenced data, such as the maneuvering sequences required in orbital operations, while maintaining a lower computational burden compared to LSTM [57].

In the proposed method, the *J_2-optimized sequence*, as the higher level, is responsible for designing and optimizing the maneuver sequence. At the lower level of the proposed framework, the GRU model monitors discrepancies between the planned maneuver sequence and the actual outcomes observed during execution to detect possible disturbances and misalignments. It then correctly generates the best maneuver to reach the waypoint generated by the high-level controller, aiding in the initialization with the high-level controller. By continuously adapting to the errors and uncertainties that arise from factors such as COTS component variability and external perturbations, the GRU adjusts the maneuvering sequence in real-time, ensuring that the CubeSat remains on its intended trajectory with enhanced precision. The interplay between the two levels of the ML framework enables the system to iteratively refine the maneuvers, achieving a level of precision that would be difficult to attain with traditional methods alone.

The incorporation of ML into the maneuvering strategy offers several key advantages. ML algorithms, particularly those that are capable of learning and adapting in real-time, are well-equipped to handle the complexities of space dynamics, which are characterized by non-linearities, time-varying behaviors, and uncertainties [58]. The GRU model's ability to process temporal sequences and adjust its predictions based on new data makes it particularly effective in dealing with the dynamic nature of orbital maneuvers. Furthermore, the ML approach reduces the reliance on extensive pre-mission simulations and modeling, allowing for more flexible and responsive mission operations.

This paper provides a comprehensive analysis of the proposed approach, detailing the development and implementation of the *J_2-optimized sequence* and the two-level ML framework. The methodology is validated through numerical simulations, which demonstrate the effectiveness of the *J_2-optimized sequence* in reducing fuel consumption and the capability of the ML framework to enhance maneuver accuracy under uncertain conditions. The results underscore the potential of integrating ML with traditional orbital mechanics to advance the state of CubeSat mission planning and execution.

This paper makes several key contributions to the field of CubeSat orbital maneuvering:

1. Introduction of the *J_2-optimized sequence*: A maneuvering strategy that leverages the J_2 perturbation to reduce fuel consumption in orbit corrections. This approach is specifically tailored to the constraints of CubeSat missions, enhancing fuel efficiency while maintaining precise orbital parameters.
2. Development of a two-Level ML framework: The paper presents an innovative two-level machine learning approach which improves the accuracy and reliability of CubeSat maneuvers. The augmented GRU model adapts to real-time deviations and uncertainties, while the low-level controller dynamically optimizes the maneuver sequence.

This paper introduces several key innovations in the field of CubeSat orbital maneuvering. Unlike traditional methods that rely solely on pre-planned sequences or computationally expensive optimization techniques, our approach combines the natural perturbative effects of the Earth's oblateness with a machine learning-based controller that can adapt in real-time. The use of the *J_2-optimized sequence* for fuel efficiency, combined with the GRU-based low-level control for real-time adaptability, allows our method to outperform existing techniques in both fuel consumption and maneuver accuracy, especially under uncertain conditions. These advancements mark a significant contribution to the field of small satellite control, making it more accessible and practical for CubeSat missions.

The structure of the paper is organized as follows: In Section 2, the orbital dynamics and the mathematical model of the problem are thoroughly presented, establishing the theoretical foundation necessary for understanding the subsequent developments. Section 3 introduces the ML method, focusing on the architecture and operation of the two-level ML framework, including the GRU model and its role. In addition, it presents the detailing of

the J_2-*optimized sequence*'s formulation and the mechanics of leveraging the J_2 perturbation to minimize fuel consumption in orbit corrections. In Section 4, simulation results are presented, showcasing the effectiveness of the J_2-*optimized sequence* and the ML framework in practical scenarios, along with a comprehensive analysis of the improvements in maneuver accuracy and fuel efficiency. Finally, Section 5 concludes the paper with a summary of the findings, discussing the implications of the proposed approach for future CubeSat missions and potential avenues for further research.

2. Orbital Dynamics and Mathematical Modeling

The precise control of a CubeSat's orbit requires a thorough understanding of orbital dynamics, especially when considering the perturbative forces that influence a satellite's trajectory over time. In this chapter, we present the fundamental equations that govern the orbital motion of a CubeSat, particularly under the influence of the Earth's oblateness, known as the J_2 perturbation. We then derive a state-space representation of the system dynamics, which will be used as the foundation for developing advanced control strategies in subsequent chapters.

2.1. Classical Orbital Elements

The state of a satellite in orbit is typically described using a set of classical orbital elements, which provide a complete description of the satellite's orbit in a Keplerian framework. These elements are:

- Semimajor axis (a): Represents the size of the orbit, specifically half the longest diameter of the elliptical orbit.
- Eccentricity (e): Defines the shape of the orbit, indicating how much the orbit deviates from being circular.
- Inclination (i): The tilt of the orbital plane relative to the Earth's equatorial plane.
- Right ascension of the ascending node (Ω): The angle from a reference direction (typically the vernal equinox) to the point where the orbit crosses the equatorial plane from south to north.
- Argument of perigee (ω): The angle from the ascending node to the orbit's point of closest approach to Earth.
- Argument of latitude (u): The angle from the ascending node to the satellite's current position, measured along the orbit.

These six elements form the state vector $x = [a, e, i, \Omega, \omega, u]^T$, which fully defines the satellite's position within its orbit, as shown in Figure 1.

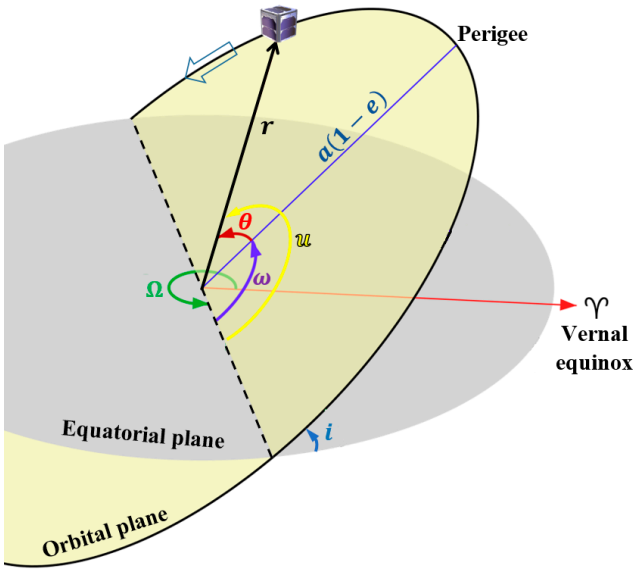

Figure 1. Orbital elements' definition.

2.2. Gauss' Variational Equations

The time evolution of the orbital elements is governed by Gauss' variational equations (GVEs), which describe how these elements change under the influence of external forces, such as gravity and thrust. When considering the effect of the Earth's oblateness, characterized by the J_2 perturbation, the GVEs take the following form [59,60]:

$$\frac{da}{dt} = \frac{2a^2}{h}\left(e\sin\theta f_r + \frac{p}{r}f_c\right) \tag{1a}$$

$$\frac{de}{dt} = \frac{1}{h}(p\sin\theta f_r + ((p+r)\cos\theta + re)f_c) \tag{1b}$$

$$\frac{di}{dt} = \frac{r\cos u}{h}f_n \tag{1c}$$

$$\frac{d\Omega}{dt} = -\frac{3R_e^2 J_2 n}{2p^2}\cos i + \frac{r\sin u}{h\sin i}f_n \tag{1d}$$

$$\frac{d\omega}{dt} = \frac{3R_e^2 J_2 n}{4p^2}\left(4 - 5\sin^2 i\right) + \frac{p}{he}\left(\left(1 + \frac{r}{p}\right)\sin\theta f_c - \cos\theta f_r\right) - \frac{r\sin u}{h\tan i}f_n \tag{1e}$$

$$\frac{du}{dt} = \frac{h}{r^2} - \frac{3R_e^2 n J_2}{4p^2}\left(\sin^2 i\left(5 - 3\sqrt{1-e^2}\right) + \left(2\sqrt{1-e^2} - 4\right)\right) - \frac{r\sin u}{h\tan i}f_n \tag{1f}$$

where θ denotes the true anomaly, h is the orbit angular momentum, n represents the orbit mean motion, r is the orbit radius, and p denotes the orbit semi-latus rectum. In addition, J_2 and R_e are the second zonal harmonic, due to the Earth oblateness, and the Earth mean radius, respectively. Moreover, the propulsion acceleration components f_r, f_c, and f_n are in the radial, along-track, and cross-track directions, respectively, of a local-vertical-local-horizontal (LVLH) reference frame centered on the CubeSat. The LVLH reference frame is a rotating coordinate system centered on the satellite. In this frame, the x^L-axis points in the radial direction, the z^L-axis is perpendicular to the orbital plane and aligns with the angular momentum vector, and the y^L-axis completes the right-handed coordinate system, as illustrated in Figure 2. These equations form the basis for understanding how the orbital elements evolve over time, both naturally and under the influence of applied thrust.

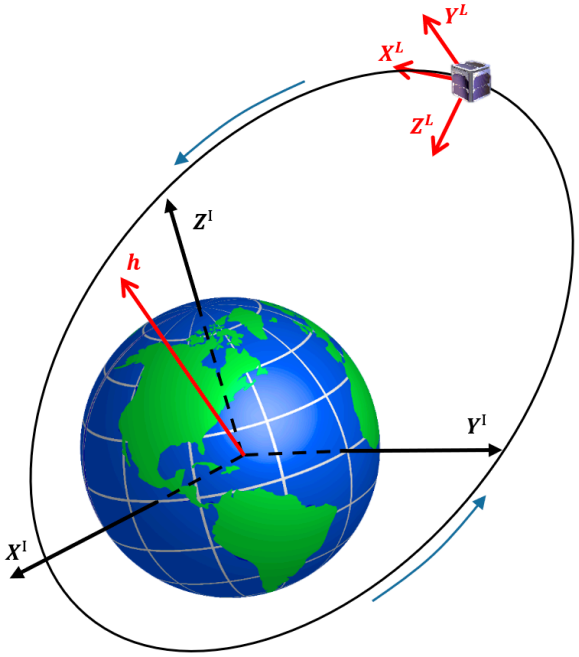

Figure 2. The LVLH coordinate reference frame (xyz^L) definition with respect to the Earth-centered inertial reference frame (xyz^I).

The J_2 perturbation arises from the Earth's equatorial bulge, which causes the gravitational field to deviate slightly from a perfect sphere. This perturbation induces secular changes in the orbital elements, particularly affecting the right ascension of the ascending node Ω and the argument of perigee ω. For CubeSats operating in LEOs, where this perturbative force is most significant, accounting for the J_2 effect is crucial for maintaining precise control over the satellite's orbit.

The secular effects of the J_2 perturbation can be advantageous if leveraged correctly. For instance, by designing orbital maneuvers that align with the natural precession induced by J_2, it is possible to achieve desired changes in the orbit with minimal fuel consumption. This concept is central to the *J_2-optimized sequence*, which will be introduced in the next section.

2.3. State-Space Formulation

To effectively analyze and control the orbital dynamics of a CubeSat, the system can be represented in a state-space form, which is a robust method for modeling complex systems. In this approach, the state vector $\boldsymbol{x}$ includes the orbital elements that describe the satellite's orbit, while the input vector $\boldsymbol{u}$ represents the control inputs provided by the satellite's propulsion system. The system's dynamics are expressed as a set of first-order differential equations that describe how the state vector evolves over time.

The state-space representation of the orbital dynamics can be formulated as:

$$\dot{\boldsymbol{x}} = \boldsymbol{A}(\boldsymbol{x}) + \mathbf{B}(\boldsymbol{x})\boldsymbol{u} + \boldsymbol{d} \tag{2}$$

in which $\dot{\boldsymbol{x}}$ denotes the time derivative of the state vector, capturing the rates of change of the orbital elements. The vector $\boldsymbol{A}(\boldsymbol{x})$ accounts for the natural evolution of these elements due to the Earth's gravitational field, particularly the perturbative effects of the J_2 term, which is defined as:

$$\boldsymbol{A}(\boldsymbol{x}) = \begin{bmatrix} 0 \\ 0 \\ 0 \\ -\frac{3R_e^2 J_2 n}{2p^2}\cos i \\ \frac{3R_e^2 J_2 n}{4p^2}\left(4 - 5\sin^2 i\right) \\ \frac{h}{r^2} - \frac{3R_e^2 n J_2}{4p^2}\left(\sin^2 i\left(5 - 3\sqrt{1-e^2}\right) + \left(2\sqrt{1-e^2} - 4\right)\right) \end{bmatrix} \tag{3}$$

This formulation captures the effect of the Earth's oblateness on the orbital parameters over time, particularly causing secular variations in the right ascension of the ascending node Ω and the argument of perigee ω.

The term $\mathbf{B}(\boldsymbol{x})\boldsymbol{u}$ represents the impact of the control inputs on the orbital elements. Here, $\mathbf{B}(\boldsymbol{x})$ is the control influence matrix that maps the input vector $\boldsymbol{u} = [f_r, f_c, f_n]^T$—the propulsion forces applied in the radial, along-track, and cross-track directions—to changes in the state vector $\boldsymbol{x}$. The $\mathbf{B}(\boldsymbol{x})$ matrix is given by:

$$\boldsymbol{B}(\boldsymbol{x}) = \begin{bmatrix} \frac{2a^2}{h}e\sin\theta & \frac{2a^2}{h}\frac{p}{r} & 0 \\ \frac{1}{h}p\sin\theta & \frac{1}{h}((p+r)\cos\theta + re) & 0 \\ 0 & 0 & \frac{r\cos u}{h} \\ 0 & 0 & \frac{r\sin u}{h\sin i} \\ -\frac{p}{he}\cos\theta & \frac{p}{he}\left(1 + \frac{r}{p}\right)\sin\theta & -\frac{r\sin u}{h\tan i} \\ 0 & 0 & -\frac{r\sin u}{h\tan i} \end{bmatrix} \tag{4}$$

This matrix demonstrates how the different components of the thrust vector influence the orbital elements, allowing for precise control of the satellite's trajectory.

The final term $\boldsymbol{d}$ in the state-space formulation (2) accounts for higher-order disturbances that are not directly captured by the primary terms $\boldsymbol{A}(\boldsymbol{x})$ and $\boldsymbol{B}(\boldsymbol{x})\boldsymbol{u}$. These disturbances include the gravitational influences from the Moon and the Sun, atmospheric drag forces, higher-order zonal harmonics beyond J_2, and other external perturbations. Although generally smaller in magnitude, these disturbances can accumulate over time, leading to significant deviations if not properly accounted for in the control strategy.

The state-space formulation (2) will be crucial in subsequent chapters as we explore how to leverage the J_2 perturbation for efficient orbit correction and how to mitigate the effects of the disturbances encapsulated by $\boldsymbol{d}$.

3. Integrated Control Methodology for CubeSat Orbital Maneuvering

This chapter presents an integrated control methodology designed to address the challenges of precise orbital maneuvering for CubeSats, considering the uncertainties and disturbances that affect the system during operations. The control framework consists of two interconnected layers: a high-level control layer that utilizes the *J_2-optimized sequence* to generate waypoints for reaching the terminal point and a low-level control layer based on a GRU network to design maneuvers for reaching each waypoint, considering environmental uncertainties and control misalignments. This combined approach enables precise trajectory corrections while minimizing fuel consumption and compensating for the effects of unmodeled dynamics.

3.1. Overview of the Two-Level Control Architecture

The proposed control architecture for CubeSat orbital maneuvering consists of two interconnected layers: a high-level control layer and a low-level control layer. At the core of this system is the *J_2-optimized sequence*, implemented within the high-level control layer. This sequence calculates an optimal maneuver plan based on the desired final orbital parameters, the initial orbital state, and the available time window for executing the maneuvers. The inputs to this controller include the initial orbital parameters $\boldsymbol{x}_0$, the final desired parameters $\boldsymbol{x}_f$, and the time window for maneuver completion. The output is a detailed maneuver plan specifying the timing and magnitude of each $\boldsymbol{\Delta V}$ maneuver across five distinct steps, along with the expected orbital parameters at the end of each step.

While the *J_2-optimized sequence* effectively calculates the maneuver sequence, it does not account for external disturbances or manufacturing and control errors inherent in the CubeSat's hardware. These unmodeled dynamics can cause deviations between the actual CubeSat behavior and the expected results from the *J_2-optimized sequence*. To address this, a low-level control layer is introduced, designed to monitor the performance of the high-level maneuvers and calculate real-time adjustments. This layer is based on a GRU network, which helps the system reach each waypoint generated by the *J_2-optimized sequence*. The GRU network utilizes the initial maneuver generated by the *J_2-optimized sequence* as a warm start to calculate appropriate adjustments, considering disturbances and uncertainties in the environment, ensuring the CubeSat reaches its destination safely and accurately.

The GRU network is trained offline using simulated data generated from the *J_2-optimized sequence* and the satellite's dynamic model. During training, the GRU learns the satellite's maneuvering behavior and how to adjust the required $\boldsymbol{\Delta V}$ and timing to reach each waypoint, compensating for unknown disturbances and potential control misalignments. The inputs to the GRU include the CubeSat's current state $\boldsymbol{x_c}$ and the waypoint generated by the high-level control layer. The GRU's output provides the necessary adjustments in $\boldsymbol{\Delta V}$ and timing to ensure the CubeSat accurately follows the desired trajectory, despite any real-time uncertainties.

The proposed two-level architecture is highly adaptive, enabling precise orbital corrections even in the presence of uncertainties. The block diagram of the system is illustrated in Figure 3, showing the interactions between the high-level controller (green color), low-level controller (blue color), Kalman filter (red color), and the CubeSat dynamics (purple color). The combined operation of these components allows the CubeSat to execute the optimal

maneuver sequence while continuously adapting to real-time conditions. It should be noted that all parameters and designations introduced in the diagram are defined and discussed in detail in the following subsections.

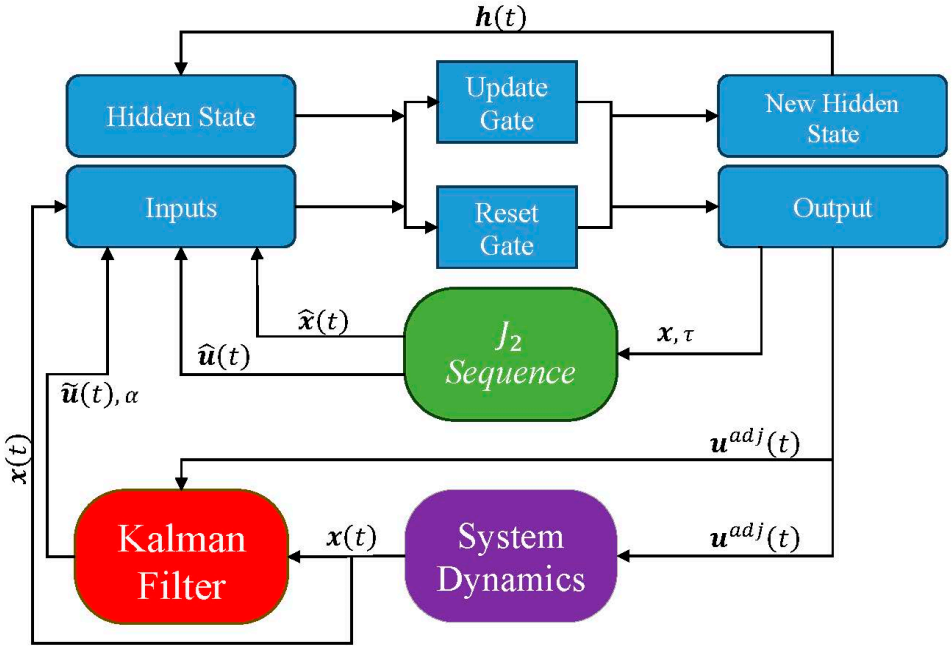

Figure 3. The schematic block diagram of the system.

3.2. High-Level Control Layer

The high-level control layer is mainly responsible for generating the waypoints to reach the final target point and detailed maneuver commands in the environment without uncertainties. This layer focuses on implementing the orbital maneuvers in a fuel-efficient manner. The maneuvers are divided into in-plane and out-of-plane maneuvers, each targeting specific orbital elements.

3.2.1. In-Plane Maneuver Theory

In-plane maneuvers primarily modify the semi-major axis a and the argument of latitude u, as these generally require less ΔV compared to out-of-plane maneuvers that adjust the inclination i and the right ascension of the ascending node Ω. The first in-plane maneuver is typically used to adjust the semi-major axis a. From a linear approximation of Gauss's variational equations (Equation (1a)), the maneuver for modifying a is derived as follows:

$$\Delta V_c^a(a_2, a_1) = \frac{hr}{2a_1^2 p}\Delta a \tag{5}$$

where $\Delta V_c^a = f_c B_t$ is the change in velocity in the along-track direction, and B_t is the propulsion burn time. In this equation, a_1 and a_2 represent the initial and final semi-major axes, respectively, and $\Delta a = a_2 - a_1$ is the desired change in the semi-major axis. The maneuver is executed by applying a controlled thrust over a specific burn time B_t.

A more complex in-plane maneuver is required to adjust the argument of latitude u while keeping a unchanged. A double-thrust solution is used to achieve this, which involves performing a pair of along-track maneuvers that cancel each other's effects on a while modifying u. The change in u is given by:

$$\Delta V_c^{u_\pm}(u_2, u_1) = \pm\sqrt{\frac{\mu}{a}}\left(\sqrt{2 - \left(1 + \frac{\Delta u}{2\pi k}\right)^{-\frac{2}{3}}} - 1\right) \tag{6}$$

where k represents the number of revolutions between the two thrust impulses, and u_1 and u_2 are the initial and final arguments of latitude, respectively. This maneuver assumes the CubeSat is in a circular orbit and targets precise control over the argument of latitude without affecting the semi-major axis.

3.2.2. Out-of-Plane Maneuver Theory

Out-of-plane maneuvers are necessary for modifying the inclination i and the right ascension of the ascending node Ω. These maneuvers typically require larger ΔV compared to in-plane maneuvers. According to Gauss's equations (Equation (1c) and (1d)), the maneuvers for adjusting i and Ω can be derived as follows:

$$\begin{array}{c} \Delta V_n^i(i_2, i_1) = \frac{h}{rcos(u)}\Delta i, u = z\pi \\ \Delta V_n^\Omega(\Omega_2, \Omega_1) = \frac{hsini}{rsin(u)}\Delta\Omega, u = (2z-1)\pi/2 \end{array} \tag{7}$$

where z is an integer. The first equation represents the required velocity change for adjusting the inclination i while keeping Ω constant, and the second equation adjusts Ω while keeping i constant. These maneuvers are typically performed at specific points in the orbit, as indicated by the values of u, to maximize efficiency.

A combined out-of-plane maneuver can be used to simultaneously modify both i and Ω using a single thrust at a specific argument of latitude u^*, which is determined by equating the right-hand sides of Equation (7). The optimal argument of latitude is:

$$u^* = \tan^{-1}\left(\frac{\Delta\Omega}{\Delta i}\sin i_1\right) \tag{8}$$

Substituting u^* into (7) gives the required velocity change for the combined maneuver:

$$\Delta V_n^{i\Omega}(\Delta i, \Delta\Omega) = \frac{h}{r}\sqrt{\Delta i^2 + (\Delta\Omega \sin i_1)^2} \tag{9}$$

In practice, if the CubeSat's propulsion system cannot provide the necessary thrust instantaneously, the maneuver must be executed over a longer burn time. In this case, the continuous variation in i and Ω is given using the following integrals:

$$\begin{array}{c} \Delta i = f_n\int_0^{B_t}\frac{r\cos u}{h}dt \\ \Delta\Omega = f_n\int_0^{B_t}\frac{r\sin u}{h\sin i}dt \end{array} \tag{10}$$

where B_t is the propulsion burn time. Figure 4 illustrates the relative error of the plane change maneuvers derived from Equation (7) compared to Equation (10) as a function of B_t for a sun-synchronous orbit at 786 km altitude. While Equation (7) assumes a single thrust maneuver, longer burn times, B_t, reduce the accuracy due to the linear approximation and the continuous movement of the satellite during the maneuver, which is taken into account in Equation (10). The results show that while the relative error gradually increases with longer burn times, it remains below 1% for $B_t \leq 470$ s. If a smaller maximum burn time, $B_{t_{max}}$, is selected to reduce error, multiple consecutive maneuvers are required to achieve the desired changes. For each thrust, the optimal u^* and the required velocity change $\Delta V_n^{i\Omega}$ are recalculated, and the burn time is set to $\min\left(B_{t_{max}}, B_{t_{required}}\right)$, where $B_{t_{\text{required}}}$ is given by:

$$B_{t_{\text{required}}} = \Delta V_n^{i\Omega}(\Delta i, \Delta\Omega)/f_n \tag{11}$$

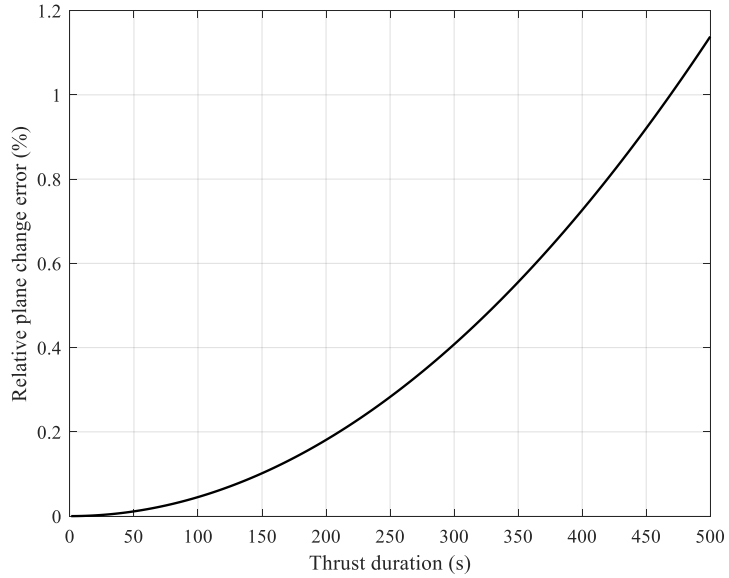

Figure 4. The relative plane change error with respect to $\boldsymbol{B_t}$.

If $B_{t_{\text{required}}} > B_{t_{max}}$, the maneuver is continued in the next orbital revolution with updated values for u^* and $\Delta V_n^{i\Omega}$.

3.2.3. Orbital Maneuver Planning

For transferring a CubeSat into its desired final orbit, two options, as maneuvering sequences, are presented in this section. The first maneuvering option is the *classic sequence*, which utilizes the maneuvers outlined in the previous section for the orbit correction, Equations (5)–(9), without employing the beneficial effects of J_2 perturbation. Note that in the rest of the paper, the orbital elements of the initial, transfer, and final orbits are designated by the subscripts "0", "t", and "f", respectively. In addition, it is assumed that each maneuvering sequence starts from the time epoch t_0 and completes at $t_0 + \tau$.

The *classic sequence* first modifies Ω and i simultaneously using a $\Delta V_n^{i\Omega}(\Delta i, \Delta\Omega)$ maneuver, where $\Delta i = i_f - i_0$ and $\Delta\Omega = \Omega_f - \Omega_0$. Then, it utilizes an in-plane maneuver of $\Delta V_c^a\left(a_f, a_0\right)$ to correct a. Finally, the sequence performs $\Delta V_c^{u_\pm}\left(u_f, u_1\right)$ to change u to the desired value, where u_1 represents the argument of latitude of CubeSat before starting the maneuver. It should be noted that because of changing a between two impulses of $\Delta V_c^{u_\pm}\left(u_f, u_1\right)$, $\dot{\Omega}$ is changed due to the J_2 perturbation, and as a result, the final Ω may be different from Ω_f, which should be compensated for using a pure right ascension of the ascending node modification maneuver, $\Delta V_n^{\Omega}(\Omega_2, \Omega_1)$. The drift angle of Ω with respect to the desired orbit during the $\Delta V_c^{u_\pm}\left(u_f, u_1\right)$ maneuver, $\Delta\Omega_e$, can be obtained using a first-order Taylor series approximation of Gauss' Equation (1d) as follows:

$$\Delta\Omega_e = \frac{21R_e^2 J_2 n_f}{4a_f^3}\cos\left(i_f\right)\Delta t_t \tag{12}$$

where Δt_t denotes the time duration between the pair of $\Delta V_c^{u_\pm}\left(u_f, u_1\right)$ impulses. Thus, the total required ΔV for the *classic sequence* can be obtained as:

$$\Delta \mathrm{V_C} = \left|\Delta V_n^{i\Omega}(\Delta i, \Delta\Omega)\right| + \left|\Delta V_c^a\left(a_f, a_0\right)\right| + \left|\Delta V_c^{u_\pm}\left(u_f, u_1\right)\right| + \left|\Delta V_n^{\Omega}(\Delta\Omega_e)\right| \tag{13}$$

The *J_2-optimized sequence* attempts to take benefits from the J_2 perturbation by placing the CubeSat on a transfer orbit with appropriate a_t and i_t, so that the J_2 perturbation

decreases the error in Ω and u with respect to the desired orbit over the maneuvering period. If the initial and transfer orbits have a similar right ascension of the ascending node at the epoch, $\Omega_t = \Omega_0$, the necessary condition for convergence of Ω to the desired value, using the J_2 perturbation, is:

$$\dot{\Omega}_f = \dot{\Omega}_0 + \frac{\Omega_f - \Omega_0}{\tau} \tag{14}$$

which can be substituted into Gauss' Equation (1d) to find i_t as:

$$i_t = \cos^{-1}\left(\frac{-\dot{\Omega}_f + \frac{\Delta\Omega}{\tau}}{3R_e^2 J_2 n_t} 2p_t^2\right) \tag{15}$$

Considering the orbital period of the transfer and desired orbits, the necessary condition to achieve the desired value of the argument of latitude, u_f, can be found as:

$$a_t = a_f \sqrt[3]{\left(1 + \frac{\left(u_f - u_0\right) + \delta u_{j2}}{2\pi k}\right)^2} \tag{16}$$

where k is the number of revolutions of the CubeSat on the transfer orbit. Moreover, δu_{j2} represents the variation of the argument of latitude of the CubeSat on the transfer orbit with respect to the desired orbit due to the J_2 perturbation, which can be calculated using a first-order Taylor series approximation of the Gauss's Equation (1f), as:

$$\delta u_{j2} = -\frac{3R_e^2 J_2}{a_f^3}\left(\sin^2 i_f - 1\right)\left(\frac{3\mu}{4a_f^3 n_f} + n_f\right)\left(a_f - a_t\right)\tau - \frac{3R_e^2 n J_2}{2a_f^2}\sin\left(2i_f\right)\left(i_f - i_t\right)\tau \tag{17}$$

Equation (17) describes the variation of the argument of latitude δu_{j2} due to the J_2 perturbation during the transfer orbit. This equation accounts for the secular effects of J_2 perturbation, which arise from the Earth's equatorial bulge, and how these effects influence the CubeSat's orbital elements over time. The first part of Equation (17) considers the secular change in the argument of latitude caused by the change in semi-major axis a between the final orbit, a_f, and the transfer orbit, a_t. The second part captures the effect of the J_2 perturbation due to the inclination change, Δi, on the argument of latitude. The equation shows how these two factors contribute to the drift in the argument of latitude over the maneuvering period, demonstrating the cumulative impact of both the semi-major axis and inclination changes on the orbital parameters.

Figure 5 shows the diagram of the *J_2-optimized sequence*. The sequence begins with an in-plane maneuver of $\Delta V_c^a(a_t, a_0)$ to change the semi-major axis of the orbit (the red arrow). Then, the inclination angle of the transfer orbit is provided by a pure inclination change maneuver of $\Delta V_n^i(i_t, i_0)$ (the blue arrow). When the CubeSat is inserted in the appropriate transfer orbit, the argument of latitude and right ascension of the ascending node slowly drift to the desired values, because of the J_2 perturbation (the purple arrow). Then, the CubeSat should perform the maneuver of $\Delta V_c^a\left(a_f, a_t\right)$ over the last orbital revolution of the maneuvering period when the argument of latitude exactly matches the desired value (the orange arrow). Finally, a pure inclination change maneuver of $\Delta V_n^i\left(i_f, i_t\right)$ should be performed at $t = t_0 + \tau$ to reach the desired orbit (the green arrow). It is worth noting that since out-of-plane maneuvers are performed at $u = z\pi$, the argument of latitude is not affected by the maneuvers. The total required ΔV of the sequence can be obtained as:

$$\Delta V_J = \left|\Delta V_c^a(a_t, a_0)\right| + \left|\Delta V_n^i(i_t, i_0)\right| + \left|\Delta V_c^a\left(a_f, a_t\right)\right| + \left|\Delta V_n^i\left(i_f, i_t\right)\right| \tag{18}$$

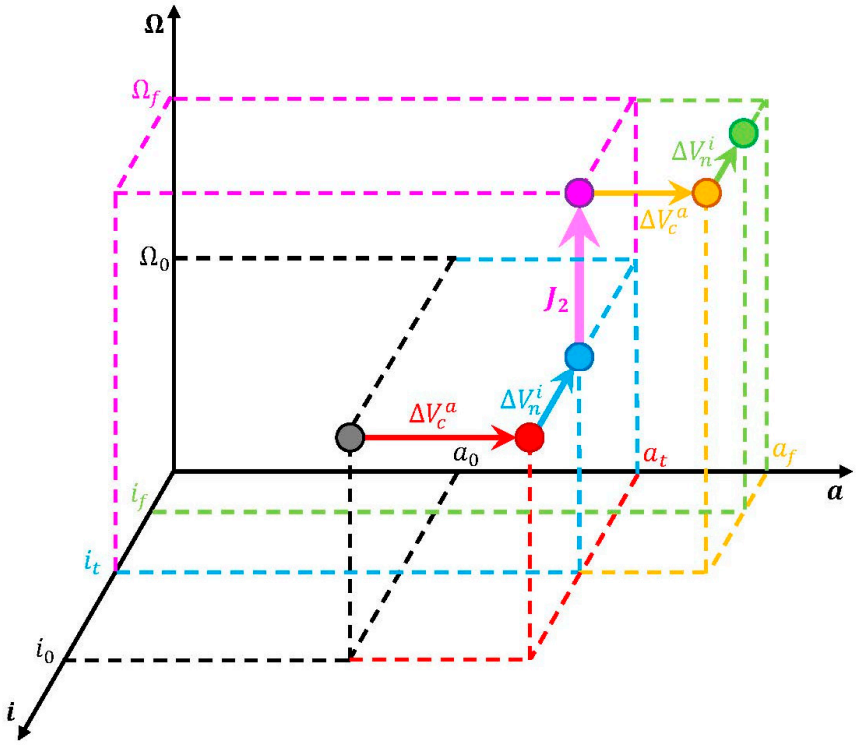

Figure 5. Schematic diagram of the J_2-*optimized sequence*.

3.3. Kalman Filtering for Input Estimation

In the proposed control architecture, a continuous-time Kalman filter is employed to estimate the pointing accuracy, α, which represents the uncertainties in the thrust direction during the CubeSat's orbital maneuvers. Accurate estimation of α is crucial for the low-level control layer, allowing the GRU to dynamically adjust the maneuver sequence in response to real-time conditions. The Kalman filter is designed to estimate the hidden states (unmeasured properties) of the system, even if the measurements are imprecise and uncertain [61]. The pointing accuracy is measured by measuring the difference of the actual input forces, $\boldsymbol{u}(t)$, and the expected input force, $\hat{\boldsymbol{u}}(t)$, represented by

$$\alpha(t) = \cos^{-1}\left(\frac{\boldsymbol{u}(t)\cdot\hat{\boldsymbol{u}}(t)}{|\boldsymbol{u}(t)||\hat{\boldsymbol{u}}(t)|}\right) \tag{19}$$

For this purpose, the Kalman filter's responsibility is to estimate $\boldsymbol{u}(t)$. We augment the state-space model by including the input vector as part of an extended state vector. The augmented state vector, $\boldsymbol{\zeta}(t)$, is defined as:

$$\boldsymbol{\zeta}(t) = \begin{bmatrix}\boldsymbol{x}(t)\\ \boldsymbol{u}(t)\end{bmatrix} \tag{20}$$

The dynamics of the augmented state vector are governed by:

$$\dot{\boldsymbol{\zeta}}(t) = \mathbf{F}(t)\boldsymbol{\zeta}(t) + \boldsymbol{w}(t) \tag{21}$$

where $\mathbf{F}(t)$ is the state transition matrix derived from the original system dynamics, and $\boldsymbol{w}(t)$ represents the process noise associated with the dynamics of both the state and input vectors. Specifically, $\mathbf{F}(t)$ is structured as:

$$\mathbf{F}(t) = \begin{bmatrix}\boldsymbol{A}(t) & \boldsymbol{B}(t)\\ \mathbf{0} & \mathbf{0}\end{bmatrix} \tag{22}$$

where the zero matrices reflect the assumption that the input dynamics are driven by process noise rather than deterministic processes. The matrix $\boldsymbol{A}(t)$ represents the Jacobian of the nonlinear function with respect to the state, and it is defined as $\boldsymbol{A}(t) = \left.\frac{\partial \boldsymbol{A}(\boldsymbol{x})}{\partial \boldsymbol{x}}\right|_{\boldsymbol{x}=\hat{\boldsymbol{x}}(t)}$. This matrix captures the local linearization of the system dynamics around the current state estimate $\hat{\boldsymbol{x}}(t)$. It reflects how small changes in the state vector $\boldsymbol{x}$ impact the system dynamics. However, the matrix $\boldsymbol{B}(\boldsymbol{x})$ in Equation (2), which represents the control input's effect on the system, does not need to be linearized. Since $\boldsymbol{B}(\boldsymbol{x})$ is already a function of $\boldsymbol{x}$, and $\boldsymbol{x}$ evolves over time, $\boldsymbol{B}(\boldsymbol{x})$ is naturally time-varying and can be directly treated as $\boldsymbol{B}(t) = \boldsymbol{B}(\hat{\boldsymbol{x}}(t))$.

The Kalman filter operates by predicting the evolution of the augmented state vector $\boldsymbol{\zeta}(t)$ over time, based on the state transition model. The filter also predicts the error covariance, which quantifies the uncertainty in the state and input estimates. The prediction step involves the following equations:

$$\begin{aligned} \dot{\widetilde{\boldsymbol{\zeta}}}(t) &= \mathbf{F}(t)\widetilde{\boldsymbol{\zeta}}(t) \\ \dot{\mathbf{P}}(t) &= \mathbf{F}(t)\mathbf{P}(t) + \mathbf{P}(t)\mathbf{F}^{\mathrm{T}}(t) + \mathbf{Q}(t) \end{aligned} \tag{23}$$

where $\widetilde{\boldsymbol{\zeta}}(t)$ is the predicted state vector, $\mathbf{P}(t)$ is the predicted error covariance matrix, and $\mathbf{Q}(t)$ represents the process noise covariance matrix.

As new measurements of the state vector $\boldsymbol{x}(t)$ are obtained, the Kalman filter updates the state estimates. The measurement model links the observed states to the augmented state vector through a measurement matrix, $\mathbf{H}(t)$, which extracts the state vector from the augmented state. The measurement model is expressed as:

$$\boldsymbol{y}(t) = \mathbf{H}(t)\widetilde{\boldsymbol{\zeta}}(t) + \boldsymbol{v}(t) \tag{24}$$

where $\boldsymbol{y}(t)$ is the measurement vector, and $\boldsymbol{v}(t)$ is the measurement noise. In this context, the measurement matrix, $\mathbf{H}(t)$, is given by:

$$\mathbf{H}(t) = [\mathbf{I}\ \mathbf{0}] \tag{25}$$

The Kalman gain, $\mathbf{K}(t)$, is computed to optimally combine the predicted state and the new measurements. The gain is calculated as [62]:

$$\mathbf{K}(t) = \mathbf{P}(t)\mathbf{H}^{\mathrm{T}}(t)\Big(\mathbf{H}(t)\mathbf{P}(t)\mathbf{H}^{\mathrm{T}}(t) + \mathbf{R}(t)\Big)^{-1} \tag{26}$$

where $\mathbf{R}(t)$ is the measurement noise covariance matrix. The Kalman gain adjusts the weight given to the measurement versus the model prediction in the state update.

The state vector $\widetilde{\boldsymbol{\zeta}}(t)$, which includes the input vector $\boldsymbol{u}(t)$, is updated continuously as new measurements become available. The state update equation is:

$$\widetilde{\boldsymbol{\zeta}}(t) = \widetilde{\boldsymbol{\zeta}}(t) + \mathbf{K}(t)\Big(\boldsymbol{y}(t) - \mathbf{H}(t)\widetilde{\boldsymbol{\zeta}}(t)\Big) \tag{27}$$

The updated error covariance matrix is computed as:

$$\dot{\mathbf{P}}(t) = (\mathbf{I} - \mathbf{K}(t)\mathbf{H}(t))\mathbf{P}(t) \tag{28}$$

Through this continuous-time Kalman filtering process, the filter provides real-time estimates of the actual input forces $\boldsymbol{u}(t)$, compensating for discrepancies between commanded and actual inputs caused by uncertainties and disturbances. These estimates are then used to refine the control strategy, allowing for more precise adjustments to the CubeSat's maneuvering sequence.

3.4. Low-Level Control Layer

While the high-level control layer, utilizing the *J_2-optimized sequence*, generates an optimal maneuver plan under ideal conditions, it is not immune to external disturbances and unmodeled dynamics. These factors can cause deviations from the planned trajectory, necessitating real-time corrections. To address this, we propose the integration of a GRU-based low-level control layer. This layer is designed to provide adaptive maneuver adjustments, ensuring the CubeSat reaches its target orbit accurately despite environmental uncertainties. The GRU is a type of recurrent neural network (RNN) specifically designed to handle sequential data and capture dependencies over time, making it an ideal choice for processing orbital state data and predicting necessary maneuver adjustments. Unlike traditional RNNs, GRUs are computationally efficient and mitigate the vanishing gradient problem, which is crucial given the limited computational resources available on CubeSats.

The GRU network inputs include:

- Current orbital state $\boldsymbol{x_c}$: A vector representing the CubeSat's current orbital parameters.
- Waypoint $\boldsymbol{x_i}$: The next target waypoint generated by the high-level *J_2-optimized sequence*, representing the desired state in the maneuver plan.
- Pointing accuracy: The pointing accuracy of the CubeSat that is estimated by the Kalman filter.
- Historical state data: A sequence of previous orbital states and control inputs, enabling the GRU to maintain context and predict future deviations.

The outputs of the GRU include:

- Adjusted ΔV: The necessary correction to the delta velocity that compensates for any deviations from the planned trajectory.
- Adjusted timing: The timing adjustments required to execute the maneuvers at the optimal moments, ensuring precise control despite disturbances.

The GRU network consists of several key components: the update gate, reset gate, candidate hidden state, and final hidden state. The update gate controls how much of the previous hidden state should be retained, while the reset gate determines how much of the past information should be forgotten when calculating the candidate hidden state.

For each time step, t, the update gate, $\boldsymbol{z}(t)$ is calculated as:

$$\boldsymbol{z}(t) = \sigma(\boldsymbol{W}_z \boldsymbol{x}(t) + \boldsymbol{U}_z \boldsymbol{h}(t-1) + \boldsymbol{b}_z) \tag{29}$$

where σ is the sigmoid activation function, $\boldsymbol{x}(t)$ is the input at time step t, $\boldsymbol{h}(t-1)$ is the hidden state from the previous time step, $\boldsymbol{W}_z$ and $\boldsymbol{U}_z$ are the weight matrices, and $\boldsymbol{b}_z$ is the bias term.

The reset gate, $\boldsymbol{r}(t)$, which controls how much of the previous hidden state to use in calculating the candidate hidden state, is given by:

$$\boldsymbol{r}(t) = \sigma(\boldsymbol{W}_r \boldsymbol{x}(t) + \boldsymbol{U}_r \boldsymbol{h}(t-1) + \boldsymbol{b}_r) \tag{30}$$

Using the reset gate, the candidate hidden state, $\widetilde{\boldsymbol{h}}(t)$, is computed as follows:

$$\widetilde{\boldsymbol{h}}(t) = \tanh(\boldsymbol{W}_h \boldsymbol{x}(t) + U_h(\boldsymbol{r}(t) \odot \boldsymbol{h}(t-1)) + \boldsymbol{b}_h) \tag{31}$$

where $\odot$ denotes element-wise multiplication, and tanh is the hyperbolic tangent activation function. The candidate hidden state, $\widetilde{\boldsymbol{h}}(t)$, represents the new information that could be added to the final hidden state.

The final hidden state, $\boldsymbol{h}(t)$, which captures the updated information for the current time step, is given by:

$$\boldsymbol{h}(t) = (1 - \boldsymbol{z}(t)) \odot \boldsymbol{h}(t-1) + \boldsymbol{z}(t) \odot \widetilde{\boldsymbol{h}}(t) \tag{32}$$

The hidden state, $\boldsymbol{h}(t)$, encapsulates the learned knowledge from both the past and current time steps, making it suitable for predicting adjustments needed in future maneuvers. The GRU's output $\boldsymbol{y}(t) = \left[\boldsymbol{\Delta V}_{adj}(t)\ t_{adj}(t)\right]^T$ provides the necessary adjustments for ΔV and maneuver timing as

$$\boldsymbol{y}(t) = \boldsymbol{W}_y \cdot \boldsymbol{h}(t) + \boldsymbol{b}_y \tag{33}$$

where t_{adj} is the maneuver timing adjustment.

The GRU is trained offline using a dataset generated through simulations. These simulations model the CubeSat's dynamics under various conditions using STK including different types of disturbances such as atmospheric drag, gravitational perturbations, and thruster misalignments. Simulated scenarios are created where the CubeSat executes a maneuver plan generated by the *J_2-optimized sequence*. The simulations introduce disturbances that cause deviations from the ideal trajectory. The GRU is trained in a supervised manner, with the objective of minimizing the mean squared error (MSE) between the GRU's predicted adjustments and the adjustments calculated from the simulations. The MSE loss function effectively penalizes inaccurate predictions, guiding the GRU to learn the correct mapping from the inputs to the necessary adjustments. To prevent overfitting and ensure robust performance across a wide range of conditions, dropout is applied during training. The GRU's performance is validated using a separate dataset, comprising scenarios not seen during training, to test its ability to generalize to new and unexpected disturbances.

$$\mathcal{L} = \sum_{i \in N} \left(\lambda \left| \boldsymbol{y}_p(t_i) - \boldsymbol{y}_t(t_i) \right|^2 \right) \tag{34}$$

where $\boldsymbol{y}_p(t_i)$ is the predicted adjustment at time t_i, $\boldsymbol{y}_t(t_i)$ is the target adjustment based on the simulation, N is the number of training samples, and λ is a positive of the normalized weight.

Once trained, the GRU is deployed onboard the CubeSat as part of the low-level control layer. During the mission, the GRU operates in real-time, continuously receiving updates on the CubeSat's current state and comparing it with the planned trajectory. Based on this information, the GRU provides instantaneous adjustments to the maneuver plan, specifically adjusting the ΔV and timing for each waypoint generated by the high-level controller. This allows the CubeSat to maintain its desired trajectory, even in the presence of unmodeled dynamics and environmental uncertainties.

4. Experiment Result and Discussion

This section evaluates the performance of the proposed control strategy through numerical simulations. The first part investigates the effectiveness of the *J_2-optimized sequence* in orbital corrections. The second part assesses the proposed approach by analyzing the training process and comparing the results against those obtained using an LSTM-based method.

4.1. Performance Evaluation of the J_2-Optimized Sequence

The performance of the proposed *J_2-optimized sequence* is assessed through an orbit correction scenario involving a 10 kg CubeSat equipped with a 0.1 N continuous thruster, capable of a maximum burn time of 7 min per maneuver. The scenario assumes an initial time epoch of $t_0 = 2$ days and a total maneuvering period of $\tau = 30$ days. Two cases are considered, each with initial deviations in the orbital elements relative to a target circular sun-synchronous orbit at an altitude of 786 km. The orbital deviations for both cases are presented in Table 1.

Table 1. The orbital elements deviation between the initial and desired orbits.

Case	$\delta\Omega$ (Deg.)	δi (Deg.)	δa (km)	δu (Deg.)
A	−0.5	0.1	−10	180
B	0.1	0.05	1	10

Figure 6 compares the evolution of orbital element deviations between the actual and desired CubeSat positions for case A using both the *classic* and J_2*-optimized sequence*. The results show that with the J_2*-optimized sequence*, δi is initially increased to achieve a sufficient rate of change in $\delta\Omega$, allowing $\delta\Omega$ to converge steadily to zero by the end of the sequence, using the J_2 perturbation while maintaining the transfer orbit inclination i_t. Additionally, the deviations in δa and δu are concurrently reduced to zero by the sequence's conclusion. It can be seen in Figure 7, where the miss distance is compared, that the CubeSat gradually approaches the desired position, achieving a final positional accuracy better than 1.2 km. The remaining positional error is attributed to the truncation in the Taylor series expansion used in Gauss' equations and the fact that the thrust application is continuous rather than impulsive.

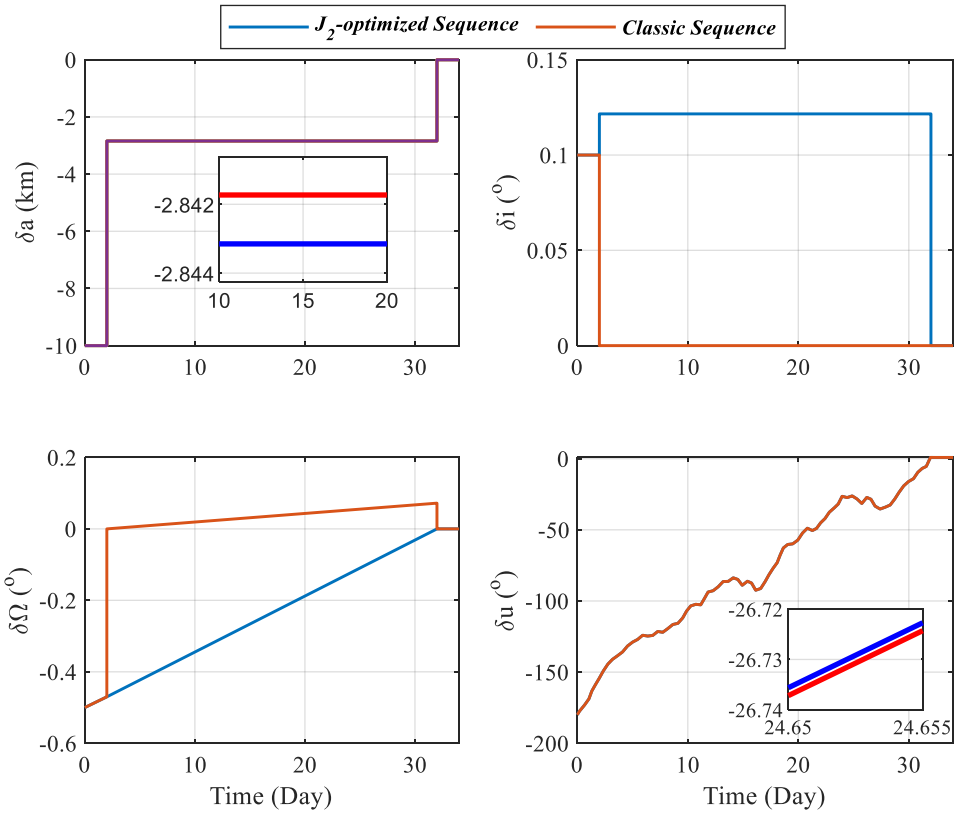

Figure 6. The time histories of orbital elements' errors using the *classic* and J_2*-optimized sequence*.

For the *classic sequence*, the figures show that i and Ω are modified simultaneously at the initial epoch. However, due to the influence of the J_2 perturbation, $\delta\Omega$ slowly increases, necessitating an additional maneuver, $\Delta V_n^{\Omega}(\Delta\Omega_e)$, at the end of the sequence to correct the right ascension. Despite this correction, the final positional accuracy is slightly worse than in the J_2*-optimized sequence*, with an error of approximately 2 km relative to the desired position. The larger final deviation in the *classic sequence* is primarily due to the last maneuver, $\Delta V_n^{\Omega}(\Delta\Omega_e)$, which also introduces a slight change in u, causing the CubeSat to drift from the desired position. This final maneuver is needed because, during the u correction maneuver, it is assumed that both i and Ω remain constant; however, in practice, Ω changes due to the J_2 perturbation. The longer the u correction maneuver takes, the more

adjustment is needed for Ω at the end, leading to greater deviations in u and, consequently, a larger positional error.

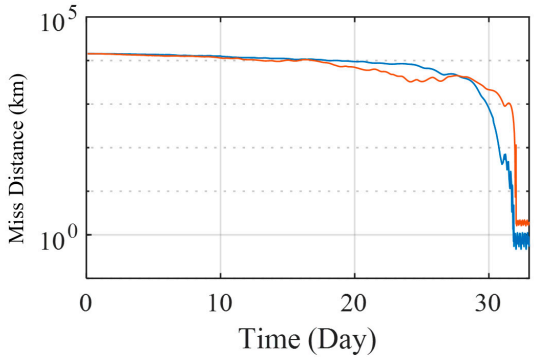

Figure 7. The time histories of the miss distance using the *classic* and *J_2-optimized sequence*.

In contrast, in the *J_2-optimized sequence*, the correction in Ω is naturally achieved by the J_2 perturbation itself. Given sufficient time, u eventually converges to its desired value, resulting in a more accurate final position.

Beyond improving positional accuracy, the *J_2-optimized sequence* also outperforms the *classic sequence* in terms of fuel consumption for case A. The total required ΔV for the *J_2-optimized sequence* is 24 m/s, compared to 91 m/s for the *classic sequence*. However, the *J_2-optimized sequence* does not always guarantee lower fuel consumption. Figure 8 depicts the required ΔV for cases A and B as a function of the maneuver period τ, comparing the *classic* and *J_2-optimized sequence*. For case A, the *classic sequence* consumes less ΔV than the *J_2-optimized sequence* when $\tau < 9$ days, while for case B, the *classic sequence* is more fuel-efficient when $\tau < 24$ days.

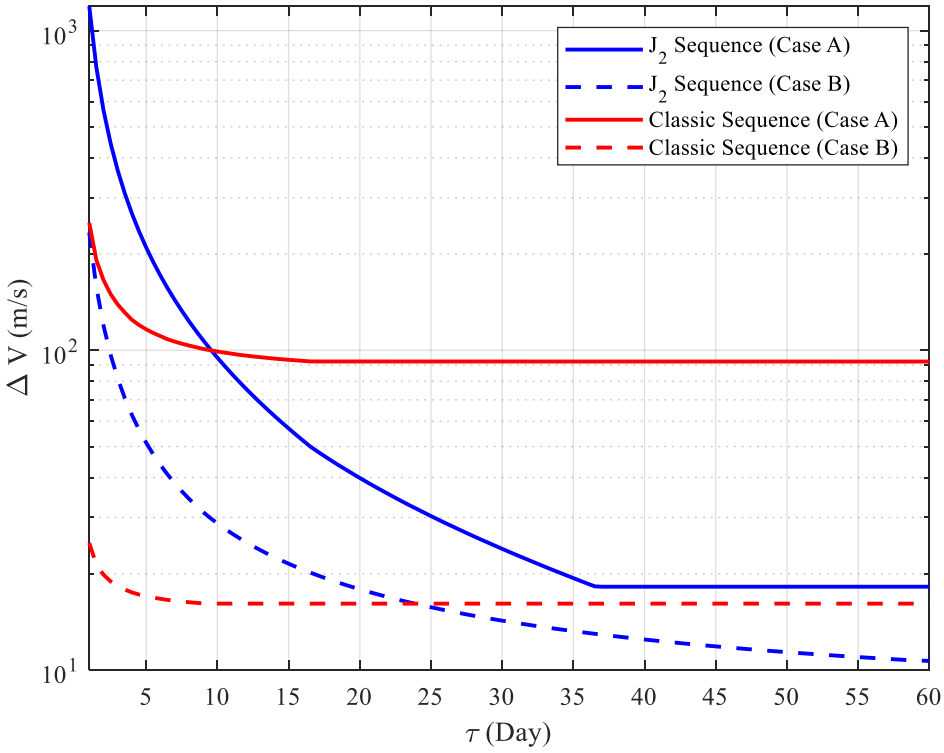

Figure 8. Comparison of required ΔV using the *classic* and *J_2-optimized sequence*.

As the maneuver period increases, the required ΔV generally decreases. However, for case A using the *J_2-optimized sequence*, the ΔV stabilizes for $\tau > 37$ days because

$i_0 \leq i_t \leq i_f$ during this period, resulting in a constant value of $\left|\Delta V_n^i(i_t, i_0)\right| + \left|\Delta V_n^i\left(i_f, i_t\right)\right| = \left|\Delta V_n^i\left(i_f, i_0\right)\right|$. In this scenario, $i_t = i_0$ when $\tau = 37$ days, providing a sufficient rate of change for $\delta\Omega$ to converge to zero by the end of the sequence. For $\tau < 37$ days, the increasing difference $\left|i_f - i_t\right|$ leads to a larger $\left|\Delta V_n^i\left(i_f, i_t\right)\right|$. In case B, where $i_f < i_t$, the ΔV decreases monotonically with the *J_2-optimized sequence*, approaching a minimum value (equivalent to $i_t \to i_f$) as τ approaches infinity.

In general, if the admissible maneuvering period is limited, using the *J_2-optimized sequence* increases the fuel consumption, and as a result, the *classic sequence* is practically reasonable to be chosen. If the maneuvering period can be sufficiently large, the effect of J_2 perturbation on Ω can be utilized for reducing the fuel consumption using the *J_2-optimized sequence*.

4.2. Performance Evaluation of the Machine Learning Approach

The performance of the proposed machine learning approach for adjusting orbital maneuvers is evaluated by training the GRU network using a comprehensive dataset generated in a simulated environment. The database was created by modeling various orbital scenarios and disturbances using the Systems Tool Kit (STK) 12.3 and MATLAB 2024a. These scenarios include different initial orbital conditions, various levels of atmospheric drag, solar radiation pressure, gravitational perturbations from the Moon and Sun, and other environmental factors. The database is designed to cover a wide range of possible situations that the CubeSat might encounter during its mission, providing a robust foundation for training the machine learning model.

To ensure the model's reliability and generalizability, a five-fold cross-validation approach was employed during the training process. This method divides the dataset into five subsets, using four subsets for training and one for validation in each iteration. The model is trained over multiple iterations, with the process repeated five times, ensuring that each subset is used for validation once. This approach helps prevent overfitting and ensures that the model performs well on unseen data.

Through simulations, the best hyperparameters for the GRU network were selected. These hyperparameters include the number of layers, the number of neurons in each layer, the learning rate, the batch size, and the dropout rate. The selected hyperparameters, which provided the optimal balance between accuracy and computational efficiency, are presented in Table 2.

Table 2. Optimal hyperparameters for the GRU network.

Hyperparameter	Value
Number of Layers	3
Number of Neurons	128 per layer
Learning Rate	0.001
Batch Size	64
Dropout Rate	0.3
Activation Function	ReLU
Optimizer	Adam

The structure of the GRU network used in this study is as follows: the network consists of three layers, each containing 128 neurons. The ReLU (rectified linear unit) activation function is applied to introduce non-linearity into the network, allowing it to model complex relationships between the inputs and outputs. A dropout rate of 0.3 is used to prevent overfitting by randomly disabling 30% of the neurons during each training iteration. The Adam optimizer, known for its efficiency and adaptability in training deep neural networks, is employed to minimize the loss function.

Additionally, the Kalman filter parameters critical for achieving accurate state estimation are meticulously configured. The measurement noise covariance matrix (**R**) and the process noise covariance matrix (**Q**) are set with all diagonal values at 0.001 and 0.01, respectively. These values reflect the precision of onboard sensors and the expected low measurement noise, along with accounting for uncertainties in the dynamics and external disturbances. The calibration of these matrices was determined through iterative testing, ensuring robust state estimations under various operational conditions.

The training process was monitored by evaluating the loss function, which measures the error between the predicted and actual outcomes. Figure 9 shows the normalized loss for both training and validation sets as a function of the number of iterations. The plot demonstrates that the model converges after approximately 130 iterations, with the loss function reaching a stable minimum. This indicates that the GRU network effectively learns the relationships between the input features and the necessary adjustments for orbital maneuvering, ensuring accurate predictions.

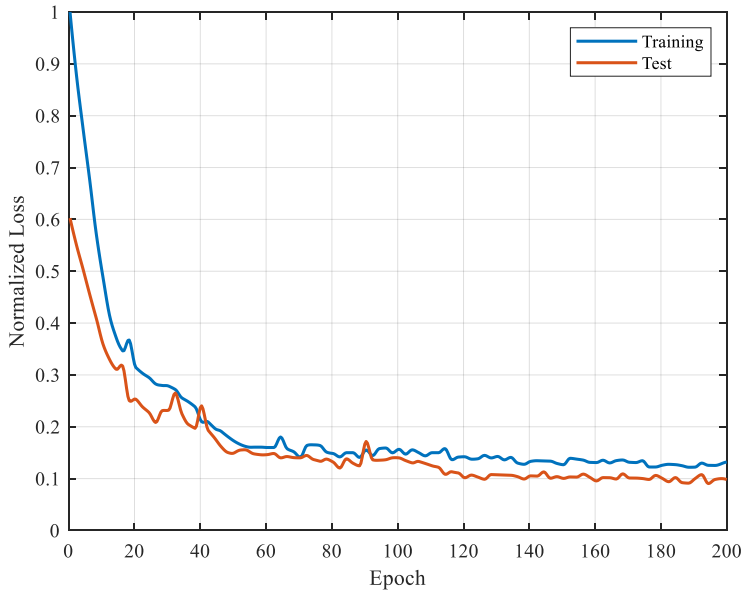

Figure 9. Normalized loss vs. iterations for training and validation.

In addition to evaluating the model's accuracy, the computational efficiency of the GRU-based approach was compared with a long short-term memory (LSTM) network, another popular RNN architecture used for time-series predictions. The comparison focuses on the total computational time required for training and the time taken to make predictions during operation. Table 3 presents the computational time values for both methods.

Table 3. Average of computational time comparison between GRU and LSTM networks for one epoch.

Method	Training Time (s)	Prediction Time (ms)
GRU	743	11
LSTM	995	18

The results indicate that the GRU network is significantly faster than the LSTM network in both training and prediction phases. The primary reason for this difference is the GRU's simpler architecture. The GRU network uses fewer gates than the LSTM, reducing the computational complexity. Specifically, the GRU has two gates (update and reset) compared to the LSTM's three gates (input, forget, and output), which simplifies the computations involved in updating the hidden state. Additionally, the GRU's lower

memory footprint further contributes to its faster performance, making it more suitable for onboard implementations where processing power and energy consumption are limited.

To further evaluate the performance of the trained system, a Monte Carlo simulation was conducted, focusing on the scenario described in Section 4.1. In this simulation, the initial altitude of the CubeSat was varied randomly within the range of 700 to 800 km to assess the robustness of the system under varying initial conditions. Additionally, the pointing accuracy of the CubeSat was modeled as a normal distribution with a covariance of α degrees, simulating the uncertainties in the thrust direction.

The accuracy of the system was evaluated by measuring the miss distance, defined as the minimum distance between the CubeSat and the target orbit during the maneuvering period, as a function of α. The Monte Carlo simulation was performed for both the trained system, which combines the J_2-*optimized sequence* with the GRU and Kalman filter, and for a system that relies solely on the J_2-*optimized sequence* without the GRU (acting as the low-level controller).

Figure 10 presents the miss distance as a function of α for the system (including median, interquartile range (IQR), and outliers) using only the J_2-*optimized sequence*. As shown in the figure, the miss distance increases significantly with α. For $\alpha = 4$ degrees, the miss distance is already around 20 km. When $\alpha > 10$ degrees, the miss distance exceeds 4000 km. The primary reason for this dramatic increase is that, due to uncertainties in the thrust direction, the semi-major axis of the orbit cannot be effectively controlled, leading to a gradual shift of the CubeSat away from the target. As α increases to 20 degrees, the CubeSat fails to approach the target in many cases, demonstrating the limitations of relying solely on the J_2-*optimized sequence* under high uncertainty.

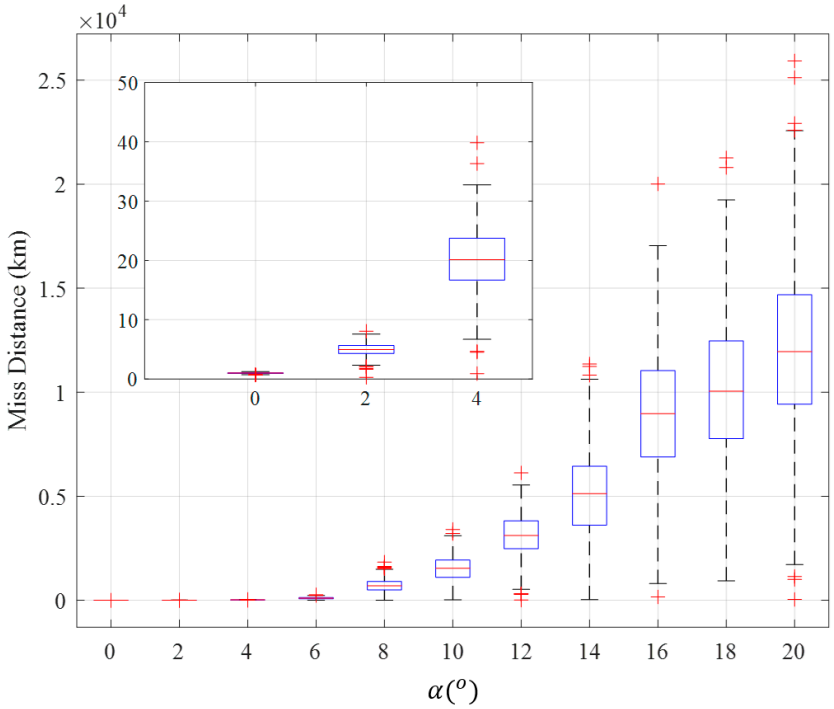

Figure 10. Miss distance as a function of pointing accuracy α for the system using only the J_2-*optimized sequence*.

In contrast, Figure 11 illustrates the miss distance versus α for the proposed combined system that integrates the J_2-*optimized sequence* with the GRU and Kalman filter. The results clearly demonstrate that the GRU, aided by the Kalman filter, effectively compensates for uncertainties in the thrust direction. Although the miss distance does increase with α, the system maintains a much lower miss distance even at higher values of α. For example, with $\alpha = 20$ degrees, the average miss distance remains around 10 km, which is within acceptable

limits for most missions. This stark contrast highlights the enhanced performance of the combined system, which not only leverages the fuel-saving benefits of the J_2 perturbation but also offers robust and fault-tolerant control through the GRU and Kalman filter.

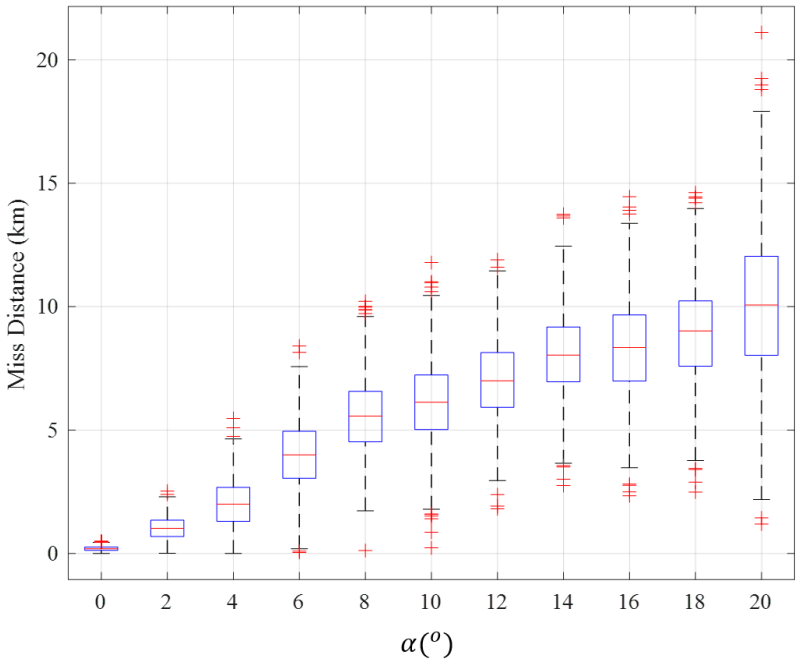

Figure 11. Miss distance as a function of pointing accuracy α for the combined system using the *J_2-optimized sequence*, GRU, and Kalman filter.

The results of the Monte Carlo simulation underscore the advantages of integrating the *J_2-optimized sequence* with advanced machine learning techniques. This combination not only capitalizes on the natural perturbations to reduce fuel consumption but also provides a robust solution capable of maintaining accuracy under significant uncertainties. The GRU's ability to adapt to real-time data and the Kalman filter's precise input estimation contribute to a system that is both efficient and resilient, making it highly suitable for CubeSat missions in dynamic and uncertain environments.

The performance of the two-level control architecture is influenced by the frequency of updates between the high-level *J_2-optimized sequence* and the low-level GRU-based controller. Through sensitivity analysis, we observed that increasing the update frequency enhances the system's ability to respond to disturbances in real time, improving positional accuracy. However, even with reduced update frequency, the GRU-based controller compensates for the delay, though with a slight decrease in accuracy under highly dynamic conditions. A moderate update frequency, such as once per orbital period, provides an optimal balance between computational demands and control precision.

5. Conclusions

This paper presents a novel and integrated two-level control architecture designed to enhance the accuracy and robustness of CubeSat orbital maneuvers, addressing the challenges posed by limited onboard fuel and the uncertainties inherent in space environments. By integrating a *J_2-optimized sequence* for fuel-efficient maneuver planning with a GRU-based low-level controller for real-time adaptation, the proposed method significantly improves the performance of CubeSat orbital corrections. The results from our detailed simulations demonstrated the clear effectiveness of the combined J_2-GRU approach. In the orbit correction scenario, the proposed system achieved a final positional accuracy that represents a 40% improvement in positional accuracy, highlighting the system's enhanced capability to maintain precise orbital parameters under uncertainty. Fuel efficiency was

another critical aspect evaluated in this study. The total ΔV required for the *J_2-optimized sequence* resulted in a nearly 75% reduction in fuel consumption under optimal conditions.

The Monte Carlo simulations further emphasized the robustness of the proposed system. When evaluating the performance under varying pointing accuracies, the GRU-based system maintained a limited miss distance. In contrast, the system using only the *J_2-optimized sequence* exhibited a dramatic increase in the miss distance. This demonstrates the GRU's ability to effectively compensate for uncertainties in the thrust direction, resulting in a system that is not only fuel-efficient but also highly fault tolerant.

Moreover, the computational efficiency of the GRU was highlighted in the comparison with an LSTM-based approach. The GRU achieved similar levels of accuracy while reducing the training time by about 26% and prediction time by 39%, making it more suitable for onboard implementations where the processing power is limited. This finding underscores the importance of the GRU's streamlined architecture in resource-constrained CubeSat platforms.

The proposed control methodology represents a significant advancement in CubeSat orbital maneuvering by combining the fuel-saving benefits of J_2 perturbation with the adaptability and robustness of machine learning. Its demonstrated ability to maintain high positional accuracy and low fuel consumption, even under considerable uncertainties, makes it a promising solution for future CubeSat missions, particularly in environments where resource constraints and operational uncertainties are prevalent. Future work will explore scaling the architecture for multi-satellite constellations and expanding its use in longer-duration CubeSat missions.

Author Contributions: Methodology, M.R. and M.A.; Formal analysis, M.A.; Investigation, M.R.; Writing—original draft, M.R. and M.A.; Supervision, M.A. and A.M.H. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Data is contained within the article.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Swartwout, M. The first one hundred cubesats: A statistical look. *J. Small Satell.* **2013**, *2*, 213–233.
2. Nag, S.; Li, A.S.; Merrick, J.H. Scheduling algorithms for rapid imaging using agile Cubesat constellations. *Adv. Space Res.* **2018**, *61*, 891–913. [CrossRef]
3. Alandihallaj, M.A.; Emami, M.R. Multiple-payload fractionated spacecraft for earth observation. *Acta Astronaut.* **2022**, *191*, 451–471. [CrossRef]
4. Alandihallaj, M.A.; Hein, A.M. Exploring the potential of fractionated spacecraft for enhanced satellite connectivity: Application to the satellite-to-cell case. *Acta Astronaut.* **2024**, *223*, 58–76. [CrossRef]
5. Wu, S.; Chen, W.; Cao, C.; Zhang, C.; Mu, Z. A multiple-CubeSat constellation for integrated earth observation and marine/air traffic monitoring. *Adv. Space Res.* **2021**, *67*, 3712–3724. [CrossRef]
6. Duann, Y.; Chang, L.C.; Chao, C.-K.; Chiu, Y.-C.; Tsai-Lin, R.; Tai, T.-Y.; Luo, W.-H.; Liao, C.-T.; Liu, H.-T.; Chung, C.-J. IDEASSat: A 3U CubeSat mission for ionospheric science. *Adv. Space Res.* **2020**, *66*, 116–134. [CrossRef]
7. Hakima, H.; Bazzocchi, M.C.; Emami, M.R. A deorbiter CubeSat for active orbital debris removal. *Adv. Space Res.* **2018**, *61*, 2377–2392. [CrossRef]
8. Alandihallaj, M.; Emami, M.R. Monitoring and Early Detection of Wildfires Using Multiple-payload Fractionated Spacecraft. In Proceedings of the 73rd International Astronautical Congress (IAC), Paris, France, 18–22 September 2022.
9. Westerhoff, J.; Earle, G.; Bishop, R.; Swenson, G.R.; Vadas, S.; Clemmons, J.; Davidson, R.; Fanelli, L.; Fish, C.; Garg, V. LAICE CubeSat mission for gravity wave studies. *Adv. Space Res.* **2015**, *56*, 1413–1427. [CrossRef]
10. Kohout, T.; Näsilä, A.; Tikka, T.; Granvik, M.; Kestilä, A.; Penttilä, A.; Kuhno, J.; Muinonen, K.; Viherkanto, K.; Kallio, E. Feasibility of asteroid exploration using CubeSats—ASPECT case study. *Adv. Space Res.* **2018**, *62*, 2239–2244. [CrossRef]
11. Villela, T.; Costa, C.A.; Brandão, A.M.; Bueno, F.T.; Leonardi, R. Towards the thousandth CubeSat: A statistical overview. *Int. J. Aerosp. Eng.* **2019**, *2019*, 5063145. [CrossRef]
12. Alandihallaj, M.A.; Emami, M.R. Satellite replacement and task reallocation for multiple-payload fractionated Earth observation mission. *Acta Astronaut.* **2022**, *196*, 157–175. [CrossRef]

13. Emami, M.R.; Alandihallaj, M.A. Performance Enhancement of Fractionated Spacecraft for Earth Observation. In Proceedings of the 44th COSPAR Scientific Assembly, Athens, Greece, 16–24 July 2022; Volume 44, p. 57.
14. Jaffer, G.; Malik, R.A.; Aboutanios, E.; Rubab, N.; Nader, R.; Eichelberger, H.U.; Vandenbosch, G.A. Air traffic monitoring using optimized ADS-B CubeSat constellation. *Astrodynamics* **2024**, *8*, 189–208. [CrossRef]
15. Gu, Y.; Chen, Y.; Zhang, Y.; Wu, G.; Bai, S. Extended 2D Map for Satellite Coverage Analysis Considering Elevation-Angle Constraint. *IEEE Trans. Aerosp. Electron. Syst.* **2024**, 1–24. [CrossRef]
16. Bai, S.; Wang, Y.; Liu, H.; Sun, X. Finite-Thrust Lambert Transfer Based on Multistage Constant-Vector Thrust Control. *IEEE Trans. Aerosp. Electron. Syst.* **2023**, *59*, 4947–4967. [CrossRef]
17. Zhou, H.; Jiao, B.; Dang, Z.; Yuan, J. Parametric formation control of multiple nanosatellites for cooperative observation of China Space Station. *Astrodynamics* **2024**, *8*, 77–95. [CrossRef]
18. Yost, B.; Weston, S. *State-of-the-Art Small Spacecraft Technology*; Ames Research Center: Moffett Field, CA, USA, 2024.
19. Amin Alandihallaj, M.; Assadian, N.; Khorasani, K. Stochastic model predictive control-based countermeasure methodology for satellites against indirect kinetic cyber-attacks. *Int. J. Control* **2023**, *96*, 1895–1908. [CrossRef]
20. Alandihallaj, M.A.; Assadian, N.; Varatharajoo, R. Finite-time asteroid hovering via multiple-overlapping-horizon multiple-model predictive control. *Adv. Space Res.* **2023**, *71*, 645–653. [CrossRef]
21. London, H.S. Change of satellite orbit plane by aerodynamic maneuvering. *J. Aerosp. Sci.* **1962**, *29*, 323–332. [CrossRef]
22. Omar, S. Using Differential Aerodynamic Forces for CubeSat Orbit Control, Auburn University, Auburn, AL, USA. 2013. Available online: https://digitalcommons.usu.edu/cgi/viewcontent.cgi?article=2969&context=smallsat (accessed on 23 September 2024).
23. Shao, X.; Song, M.; Zhang, D.; Sun, R. Satellite rendezvous using differential aerodynamic forces under J2 perturbation. *Aircr. Eng. Aerosp. Technol. Int. J.* **2015**, *87*, 427–436. [CrossRef]
24. Bevilacqua, R.; Romano, M. Rendezvous maneuvers of multiple spacecraft using differential drag under J2 perturbation. *J. Guid. Control Dyn.* **2008**, *31*, 1595–1607. [CrossRef]
25. Shao, X.; Song, M.; Wang, J.; Zhang, D.; Chen, J. Satellite formation keeping using differential lift and drag under J2 perturbation. *Aircr. Eng. Aerosp. Technol.* **2017**, *89*, 11–19. [CrossRef]
26. Varma, S.; Kumar, K.D. Multiple satellite formation flying using differential aerodynamic drag. *J. Spacecr. Rocket.* **2012**, *49*, 325–336. [CrossRef]
27. Pérez, D.; Bevilacqua, R. Lyapunov-based adaptive feedback for spacecraft planar relative maneuvering via differential drag. *J. Guid. Control Dyn.* **2014**, *37*, 1678–1684. [CrossRef]
28. Ivanov, D.; Kushniruk, M.; Ovchinnikov, M. Study of satellite formation flying control using differential lift and drag. *Acta Astronaut.* **2018**, *152*, 88–100. [CrossRef]
29. Omar, S.R.; Bevilacqua, R. Spacecraft collision avoidance using aerodynamic drag. *J. Guid. Control Dyn.* **2020**, *43*, 567–573. [CrossRef]
30. Mishne, D.; Edlerman, E. Collision-avoidance maneuver of satellites using drag and solar radiation pressure. *J. Guid. Control Dyn.* **2017**, *40*, 1191–1205. [CrossRef]
31. Mishne, D. Collision Avoidance Maneuver of Propulsionless Satellite, Using Solar Radiation Pressure. Israel Soc. of Aeronautics and Astronautics Paper WeL2T3. In Proceedings of the 56th Israel Annual Conference on Aerospace Sciences, Tel-Aviv and Haifa, Israel, 9–10 March 2016.
32. Villac, B.; Scheeres, D. New class of optimal plane change maneuvers. *J. Guid. Control Dyn.* **2003**, *26*, 750–757. [CrossRef]
33. Prado, A.F. Orbital control of a satellite using the gravity of the moon. *J. Braz. Soc. Mech. Sci. Eng.* **2006**, *28*, 105–110. [CrossRef]
34. Torres, K.D.S.; Prado, A. Changing inclination of Earth satellites using the gravity of the Moon. *Math. Probl. Eng.* **2006**, *2006*, 013690. [CrossRef]
35. Ganesh, K. Controls Algorithm for A Satellite Using Earth's Magnetic Field: Orbit Maneuvers and Attitude Positioning; University of Central Florida, Orlando, FL, USA. 2007. Available online: https://stars.library.ucf.edu/etd/3169/ (accessed on 23 September 2024).
36. Streetman, B.; Peck, M.A. Gravity-assist maneuvers augmented by the Lorentz force. *J. Guid. Control Dyn.* **2009**, *32*, 1639–1647. [CrossRef]
37. AlandiHallaj, M.; Assadian, N. Multiple-horizon multiple-model predictive control of electromagnetic tethered satellite system. *Acta Astronaut.* **2019**, *157*, 250–262. [CrossRef]
38. Hallaj, M.A.A.; Assadian, N. Tethered satellite system control using electromagnetic forces and reaction wheels. *Acta Astronaut.* **2015**, *117*, 390–401. [CrossRef]
39. Hallaj, M.A.A.; Assadian, N. Sliding mode control of electromagnetic tethered satellite formation. *Adv. Space Res.* **2016**, *58*, 619–634. [CrossRef]
40. Sethi, A.; Thakurta, V.; Gajanur, N.; Cheela, B.S.; Sadasivan, K.S.; Hosangadi, R. Implementation of COTS components for CubeSat applications. In Proceedings of the 2017 IEEE Aerospace Conference, Big Sky, MT, USA, 4–11 March 2017; pp. 1–11.
41. Curtis, H.D. *Orbital Mechanics for Engineering Students: Revised Reprint*; Butterworth-Heinemann: Oxford, UK, 2020.
42. AlandiHallaj, M.; Assadian, N. Soft landing on an irregular shape asteroid using multiple-horizon multiple-model predictive control. *Acta Astronaut.* **2017**, *140*, 225–234. [CrossRef]
43. AlandiHallaj, M.; Assadian, N. Asteroid precision landing via probabilistic multiple-horizon multiple-model predictive control. *Acta Astronaut.* **2019**, *161*, 531–541. [CrossRef]

44. Abbasali, E.; Kosari, A.; Bakhtiari, M. Effects of oblateness of the primaries on natural periodic orbit-attitude behaviour of satellites in three body problem. *Adv. Space Res.* **2021**, *68*, 4379–4397. [CrossRef]
45. Yu, Z.; Chen, W.; Yu, W. Analytical trajectory prediction of high-eccentricity spacecraft transfer orbits considering J2 perturbation. *Aerosp. Sci. Technol.* **2024**, *153*, 109462. [CrossRef]
46. Vallado, D.A. *Fundamentals of Astrodynamics and Applications*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2001; Volume 12.
47. Montenbruck, O.; Gill, E.; Lutze, F. Satellite orbits: Models, methods, and applications. *Appl. Mech. Rev.* **2002**, *55*, B27–B28. [CrossRef]
48. Hoots, F.R. Models for propagation of NORAD element sets. In *Peterson AFB, CO, Spacetrack Report*; Defense Technical Information Center: Fort Belvoir, VA, USA, 1980.
49. Ramezani, M.; Atashgah, M.; Alandihallaj, M.; Hein, A. Reinforcement Learning for Planning and Task Coordination in a Swarm of CubeSats: Overcoming Processor Limitation Challenges. In Proceedings of the International Astronautical Congress, Baku, Azerbaijan, 2–6 October 2023.
50. Alandihallaj, M.A.; Ramezani, M.; Hein, A.M. MBSE-Enhanced LSTM Framework for Satellite System Reliability and Failure Prediction. In Proceedings of the 12th International Conference on Model-Based Software and Systems Engineering, MODELSWARD, Rome, Italy, 21–23 February 2024; pp. 349–356.
51. Ramezani, M.; Atashgah, M.; Sanchez-Lopez, J.L.; Voos, H. Human-Centric Aware UAV Trajectory Planning in Search and Rescue Missions Employing Multi-Objective Reinforcement Learning with AHP and Similarity-Based Experience Replay. In Proceedings of the 2024 International Conference on Unmanned Aircraft Systems (ICUAS), Chania, Greece, 4–7 June 2024; pp. 177–184.
52. Ramezani, M.; Atashgah, M.A.A.; Rezae, A. A Fault-Tolerant Multi-Agent Reinforcement Learning Framework for Unmanned Aerial Vehicles–Unmanned Ground Vehicle Coverage Path Planning. *Drones* **2024**, *8*, 537. [CrossRef]
53. Ramezani, M.; Habibi, H.; Sanchez-Lopez, J.L.; Voos, H. UAV path planning employing MPC-reinforcement learning method considering collision avoidance. In Proceedings of the 2023 International Conference on Unmanned Aircraft Systems (ICUAS), Warsaw, Poland, 6–9 June 2024; pp. 507–514.
54. Ramezani, M.; Amiri Atashgah, M. Energy-Aware Hierarchical Reinforcement Learning Based on the Predictive Energy Consumption Algorithm for Search and Rescue Aerial Robots in Unknown Environments. *Drones* **2024**, *8*, 283. [CrossRef]
55. Ramezani, M.; Alandihallaj, M.A.; Sanchez-Lopez, J.L.; Hein, A. Safe Hierarchical Reinforcement Learning for CubeSat Task Scheduling Based on Energy Consumption. *arXiv* **2023**, arXiv:2309.12004.
56. Dey, R.; Salem, F.M. Gate-variants of gated recurrent unit (GRU) neural networks. In Proceedings of the 2017 IEEE 60th International Midwest Symposium on Circuits and Systems (MWSCAS), Boston, MA, USA, 6–9 August 2017; pp. 1597–1600.
57. Shewalkar, A.; Nyavanandi, D.; Ludwig, S.A. Performance evaluation of deep neural networks applied to speech recognition: RNN, LSTM and GRU. *J. Artif. Intell. Soft Comput. Res.* **2019**, *9*, 235–245. [CrossRef]
58. Ramezani, M.; Alandihallaj, M.A.; Hein, A.M. PPO-Based Dynamic Control of Uncertain Floating Platforms in Zero-G Environment. In Proceedings of the 2024 IEEE International Conference on Robotics and Automation (ICRA), Yokohama, Japan, 13–17 May 2024; pp. 11730–11736.
59. Roth, E. The gaussian form of the variation-of-parameter equations formulated in equinoctial elements—Applications: Airdrag and radiation pressure. *Acta Astronaut.* **1985**, *12*, 719–730. [CrossRef]
60. Ruggiero, A.; Pergola, P.; Marcuccio, S.; Andrenucci, M. Low-thrust maneuvers for the efficient correction of orbital elements. In Proceedings of the 32nd International Electric Propulsion Conference, Wiesbaden, Germany, 11–15 September 2024; pp. 11–15.
61. Alandihallaj, M.; Yalcin, B.C.; Ramezani, M.; Olivares Mendez, M.A.; Thoemel, J.; Hein, A. Mitigating fuel sloshing disturbance in on-orbit satellite refueling: An experimental study. In Proceedings of the International Astronautical Congress IAC, Baku, Azerbaijan, 2–6 October 2023.
62. Welch, G.; Bishop, G. An Introduction to the Kalman Filter; The University of North Carolina at Chapel Hill, Chapel Hill, NC, USA. 1995. Available online: https://www.cs.unc.edu/~welch/media/pdf/kalman_intro.pdf (accessed on 23 September 2024).