



Article

Adaptive Control of Mini Space Robot Based on Linear Separation of Inertial Parameters

Yuchen Liu ^{1,2} , Lai Teng ^{1,2,*}  and Zhonghe Jin ^{1,2}

¹ School of Aeronautics and Astronautics, Zhejiang University, Hangzhou 310058, China; 12024005@zju.edu.cn (Y.L.); jinzh@zju.edu.cn (Z.J.)

² Zhejiang Provincial Key Laboratory of Micro-Nano Satellite Research, Hangzhou 310058, China

* Correspondence: tenglai@zju.edu.cn

Abstract: Space robots exhibit a strong dynamic coupling between the manipulator and the base spacecraft, with this phenomenon being particularly pronounced in mini space robots. The uncertainty surrounding the inertial parameters of space robots often renders dynamics-based controllers ineffective, and identifying these parameters in an on-orbit environment poses significant challenges. In this paper, we propose an adaptive controller for dynamic approximation that is specifically designed for mini space robots. This controller employs a linear separation of inertial parameters and utilizes recursive least-squares and Lyapunov methods to update the inertial parameter vectors. Simulation results validate the effectiveness of this adaptive controller in enabling mini space robots to accurately track predefined trajectories. Additionally, we compare the effects of the two parameter update methods on the controller stability under varying prior inertial parameter errors. The proposed inertial parameter separation adaptive controller significantly approximates the dynamics of mini space robots and facilitates precise on-orbit control, thereby offering considerable potential for advancing space exploration, satellite missions, and robotic operations.

Keywords: minisatellite; inertial parameters; attitude control



Citation: Liu, Y.; Teng, L.; Jin, Z. Adaptive Control of Mini Space Robot Based on Linear Separation of Inertial Parameters. *Aerospace* **2023**, *10*, 679. <https://doi.org/10.3390/aerospace10080679>

Academic Editor: Hyun-Ung OH

Received: 25 June 2023

Revised: 26 July 2023

Accepted: 27 July 2023

Published: 30 July 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

As the complexity and diversity of spacecraft missions in space continue to increase, the structure of the spacecraft has become more complicated and free-flying robots are poised to play a vital role in future space exploration. The recent rapid advancements in miniature component technologies and their accessibility have elevated the importance of minisatellites as tools for space development and utilization [1,2]. As a typical comprehensive intelligent operating system, the development of mini space robots is changing the traditional modes of space transportation, space missions, planetary exploration, and making them a crucial enabling technology for future space missions [3]. The growing interest and adoption of mini space robots has stimulated multiple mission designs that feature robotic manipulators mounted on minispacecraft, which result in highly coupled systems. In addition to the strong coupling problem of the space chain manipulator structure, the center-of-mass offset and changes in inertial parameters of mini space robots significantly impact dynamic modeling and control [4,5]. Therefore, designing a dynamics-based controller for spacecraft with complex chain structures holds great significance.

Early complex-structured spacecraft did not utilize dynamics-based control methods [6–11]. The primary reason for this was that the International Space Station, as a spacecraft initially equipped with a space manipulator, possessed relatively large mass and inertial parameters, rendering the disturbance and coupling effects of manipulator operations on the spacecraft negligible [6–8]. Furthermore, the initial space manipulators were manually operated in space, and the level of automation was relatively simple. Consequently, the mathematical models for space manipulators in these cases resembled those used for ground-based

manipulators [12]. However, with the emergence of minisatellites, dynamic modeling has gained a greater importance, and satellites with such complex structures are also referred to as mini space robots [13].

The earliest studies, [14,15], employed the generalized Jacobian matrix (GJM) to describe the kinematic model when a single-chain manipulator was attached to a satellite. Building upon this, Ref. [16] proposed the virtual manipulator (VM) method to determine the offset of the spacecraft body, which is a virtual concept, by considering the change in the center of mass of the space robot system. Subsequently, Ref. [17] introduced the dynamically equivalent manipulator (DEM) method for passive spherical hinges based on the VM method. The DEM method was theoretically proven to be equivalent to the suspension problem as a fixed base. In terms of dual-arm space robots, Ref. [18] proposed a mathematical model that made significant contributions to the theoretical basis of dual-arm spacecraft's space grasping tasks. Furthermore, Refs. [19,20] extended the mathematical model of the free-flying spacecraft with two arms to encompass the dynamic modeling of multiarm space robots (MASRs).

When designing dynamics-based controllers for space robots, obtaining precise inertial parameters poses a significant challenge, particularly due to the uncertainties arising from complex space environments (such as fuel exhaustion and component damage or the complex structure of the satellite itself). These uncertainties make it difficult to acquire accurate inertial parameters, rendering the controllers ineffective [13]. To address this issue, two main approaches are commonly employed: designing adaptive controllers capable of updating parameters or performing on-orbit identification to obtain precise inertial parameters. Researchers such as [21–23] developed adaptive controllers for space robots that did not rely on assumptions about inertial parameters or initial estimates of the system momentum. They demonstrated the effectiveness of adaptive control in dealing with uncertain dynamical systems. In the study conducted by [24], adaptive laws and the minimum learning parameter (MLP) algorithm were employed. While these studies can estimate unknown dynamical systems with adaptive controllers, the slow convergence speed and extensive computational load associated with such controllers make them impractical for engineering implementation. To overcome these limitations, Ref. [25] proposed a gain-parameter-adaptive control algorithm capable of achieving set-point control tasks with a fast settling time and high accuracy, even in the presence of unknown dynamic systems. However, a drawback of that algorithm is that the update of the adaptive gain parameter can induce system oscillations.

Other scholars focus on how to identify inertial parameters in orbit. Xu solved the dynamic equation by unlocking the joints of the space robot in sequence and gradually identified the inertial parameters of each component [26]. However, that method did not take into account the specific installation direction of the joint, which made some inertial parameters unidentifiable, and that method had a long identification period. Nabavi-Chashmi developed a specific regression matrix in order to speed up the process of inertial parameter identification [27]. The author's theory was based on the planar body assumption and could not be used in generalized space robots, yet. Ma used a robotic arm to change the inertial distribution of the spacecraft, and that redistribution was used to solve the inertial parameters of the spacecraft base [28]. Some studies identify the inertial parameters of single-rigid-body spacecraft through feedback information from sensors and actuators, but these methods cannot be used in multi-rigid-body space robots [29–32]. Lei designed an adaptive fault-tolerant control algorithm for the free-floating space robot system in the case of parameter uncertainty and actuator failure, which could compensate for the unknown inertial parameters of the system and the actuator failure [33]. The downside was that similar to the study of Nabavi-Chashmi, the algorithm only worked on the plane hypothesis.

Taking the mini space robot with a large inertia ratio between the manipulators and the satellite base as the research object, this paper establishes the dynamic model by using a VM method and the Newton–Euler equation. Then, the adaptive controller based on a

linear separation of inertial parameters is used for the dynamic approximation control of mini space robots. Then, based on the recursive least-squares and the Lyapunov methods, two different parameter update methods for the controller are proposed, respectively. In addition, the influence of the prior inertial parameter error and related control parameters on the control results is illustrated by simulation. The two parameter update methods are also simulated to illustrate their suitable usage scenarios and their respective defects.

This paper is organized as follows: In Section 2, the kinematics and dynamics model of the mini space robot is established by the method of the virtual manipulator and Newton–Euler’s recursive dynamics. Then, the linear separation of inertial parameters, the controller design, and the associated stability proofs are given in Section 3. In order to prove the feasibility of the controller and the applicable scenarios of the two parameter updates, the relevant simulations are carried out in Section 4. Finally, the conclusions are stated in Section 5.

2. Modeling of Mini Space Robotic System

2.1. Problem Description

Similar to papers [20,34,35], the dynamic model of the mini space robot can be regarded as a number of different chain components connected to the satellite body. In this paper all components were considered as rigid bodies. A schematic diagram of the kinetic model is shown in Figure 1.

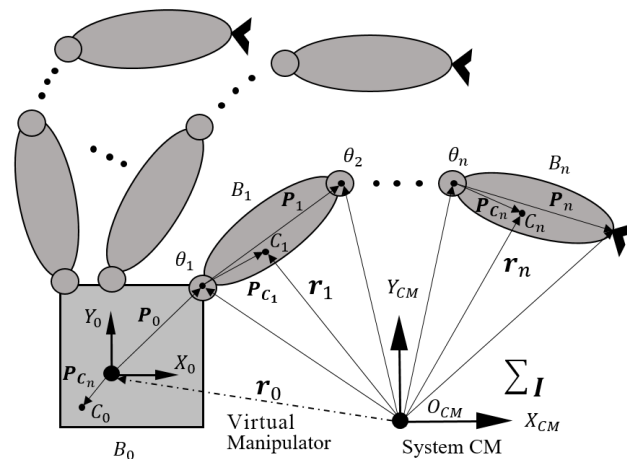


Figure 1. General model of a mini space robot.

2.2. Kinematics and Dynamics

The dynamic model of the mini space robot is divided into two parts: a spacecraft base and n manipulators connected to the base.

2.2.1. Calculation of System’s Centroid Position

The VM method establishes a connection between the center of mass of the system and the satellite base using a virtual manipulator. The virtual manipulator is characterized by its kinematic properties rather than its dynamic properties. By determining the positions of each rigid body component of the mini space robot, the position of the system’s centroid is also determined. The position of the centroid can be mathematically expressed using Equation (1):

$$r_0 = \left\{ \sum_{i=1}^n [m_i (\sum_{i=0}^{n-1} P_i + P_{C_i})] + m_0 P_{C_0} \right\} / \sum_{i=0}^n m_i \tag{1}$$

In Equation (1), P_i denotes the vector representing the position of the i -th component body coordinate pointing to the $(i+1)$ -th component. P_{C_i} represents the coordinates of the center

of mass in the body coordinate system of the i -th component. By using this formulation, the total inertia of the system can be expressed as shown in Equation (2):

$$I_{CM} = \sum_{i=0}^n [{}^c I_i + m_i(\mathbf{r}_i \mathbf{r}_i^T \mathbf{E} - \mathbf{r}_i^T \mathbf{r}_i)] \tag{2}$$

In Equation (2), ${}^c I_i$ represents the inertia tensor of the centroid position in the body coordinate system of each component.

2.2.2. Kinematics of Spacecraft Base

Euler angles are commonly used to express the attitude of a rigid body [26]. In this paper, the specific sequence of rotations $Z(\alpha) - Y(\beta) - X(\gamma)$ is employed, where the attitude is described by the Euler angles $\Phi = [\alpha, \beta, \gamma]^T$. The relationship between the time derivative of the Euler angles $\dot{\Phi}$ and the angular velocity ω is given by:

$$\omega = \begin{bmatrix} 0 & -\sin\alpha & \cos\alpha\cos\beta \\ 0 & \cos\alpha & \sin\alpha\cos\beta \\ 1 & 0 & -\sin\beta \end{bmatrix} \begin{bmatrix} \dot{\alpha} \\ \dot{\beta} \\ \dot{\gamma} \end{bmatrix} = N_\phi \dot{\Phi} \tag{3}$$

However, the singularity of Euler angles causes the rotational speed $\dot{\Phi}$ to be unrepresented by ω , because N_ϕ is irreversible when $\cos\alpha$ is equal to zero. To avoid singularity problems, quaternions are usually used to express the attitude of a rigid body [36], which can be expressed as Equation (4):

$$\begin{aligned} \mathbf{Q} &= [q_0 \ q_1 \ q_2 \ q_3]^T = [q_0 \ \mathbf{q}^T]^T \\ &= \left[\cos\frac{\psi}{2} \ \sin\frac{\psi}{2} \mathbf{e}^T \right]^T \end{aligned} \tag{4}$$

In Equation (4), \mathbf{e} represents the unit rotation axis, and ψ represents the rotation angle along \mathbf{e} . The equation satisfies the condition $q_0^2 + \mathbf{q}^T \mathbf{q} = 1$. It is important to note that any rigid body attitude can be represented by both \mathbf{Q} and $-\mathbf{Q}$. However, when $-\pi < \psi \leq \pi$ (in radians) and $q_0 > 0$ are satisfied, the expressed attitude is unique. The time derivative of \mathbf{Q} can be expressed as shown in Equation (5):

$$\begin{bmatrix} \dot{q}_0 \\ \dot{\mathbf{q}} \end{bmatrix} = \frac{1}{2} \begin{bmatrix} -\mathbf{q}^T \\ q_0 \mathbf{E}_n + \mathbf{q}^\times \end{bmatrix} \omega \tag{5}$$

where \mathbf{E}_n represents an identity matrix of size $n \times n$, and \mathbf{q}^\times is defined as:

$$\mathbf{q}^\times = \begin{bmatrix} 0 & -q_3 & q_2 \\ q_3 & 0 & -q_1 \\ -q_2 & q_1 & 0 \end{bmatrix}$$

ω can also be represented by \mathbf{Q} as shown in Equation (6):

$$\omega = 2 \left(\frac{q_0^2 \mathbf{E}_3 - q_0 \mathbf{q}^\times + \mathbf{q} \mathbf{q}^T}{q_0} \right) \dot{\mathbf{q}} \tag{6}$$

Then, a 3×3 attitude matrix denoted as ${}^i_B \mathbf{R}$ can be used to represent the conversion between the coordinates of the inertial system and the base coordinate system. The linear velocity of the spacecraft base, deviating from the center of mass of the inertial system, can be expressed as shown in Equation (7):

$${}^B v_B = \dot{r}_0 \tag{7}$$

2.2.3. Manipulator Chain Dynamics

By applying the kinematics of the spacecraft base, the dynamics of each manipulator can be modeled using the Newton–Euler method [13,18,24,37–40]. The angular and linear velocities of the $(i + 1)$ -th manipulator can be expressed in terms of the i -th manipulator, as shown in Equations (8) and (9):

$${}^{i+1}\boldsymbol{\omega}_{i+1} = {}^{i+1}\mathbf{R}^i \boldsymbol{\omega}_i + \dot{\boldsymbol{\theta}}_{i+1} {}^{i+1}\hat{\mathbf{Z}}_{i+1} \tag{8}$$

$${}^{i+1}\mathbf{v}_{i+1} = {}^{i+1}\mathbf{R} \left({}^i\mathbf{v}_i + {}^i\boldsymbol{\omega}_i \times {}^i\mathbf{P}_{i+1} \right) \tag{9}$$

where the superscript i denotes that the vector is represented in the i -th coordinate, ${}^i\hat{\mathbf{Z}}_i$ represents the unit vector of the rotation direction of the i -th axis, and ${}^{i+1}\mathbf{R}$ represents the rotation matrix from the $(i + 1)$ -th coordinate system to the i -th coordinate system. In Equation (9), ${}^i\mathbf{P}_{i+1}$ represents the position vector of the origin of the $(i + 1)$ -th coordinate system in the i -th coordinate system. The equations can also be expressed as follows:

$$\begin{bmatrix} \mathbf{v}_e \\ \boldsymbol{\omega}_e \end{bmatrix} = \mathbf{J}_b \begin{bmatrix} \mathbf{v}_0 \\ \boldsymbol{\omega}_0 \end{bmatrix} + \mathbf{J}_m \boldsymbol{\Theta} \tag{10}$$

where \mathbf{J}_b and \mathbf{J}_m are Jacobian matrices that map the velocities of the base and manipulator, respectively, to those of the end effector. They are defined as follows:

$$\mathbf{J}_b = \begin{bmatrix} \mathbf{E} & -(\mathbf{P}_e - \mathbf{P}_0)^\times \\ \mathbf{O} & \mathbf{E} \end{bmatrix} \in \mathbb{R}^{6 \times 6} \tag{11}$$

$$\mathbf{J}_m = \begin{bmatrix} \hat{\mathbf{Z}}_1 \times (\mathbf{P}_e - \mathbf{P}_1) & \cdots & \hat{\mathbf{Z}}_n \times (\mathbf{P}_e - \mathbf{P}_n) \\ \hat{\mathbf{Z}}_1 & \cdots & \hat{\mathbf{Z}}_n \end{bmatrix} \in \mathbb{R}^{6 \times n} \tag{12}$$

The angular and linear acceleration of each component can be obtained according to Equations (8) and (9)

$${}^{i+1}\dot{\boldsymbol{\omega}}_{i+1} = {}^{i+1}\mathbf{R}^i \dot{\boldsymbol{\omega}}_i + {}^{i+1}\mathbf{R}^i \boldsymbol{\omega}_i \times \dot{\boldsymbol{\theta}}_{i+1} {}^{i+1}\hat{\mathbf{Z}}_{i+1} + \ddot{\boldsymbol{\theta}}_{i+1} {}^{i+1}\hat{\mathbf{Z}}_{i+1} \tag{13}$$

$${}^{i+1}\dot{\mathbf{v}}_{i+1} = {}^{i+1}\mathbf{R}^i (\dot{\boldsymbol{\omega}}_i \times {}^i\mathbf{P}_{i+1} + {}^i\boldsymbol{\omega}_i \times ({}^i\boldsymbol{\omega}_i \times {}^i\mathbf{P}_{i+1}) + {}^i\dot{\mathbf{v}}_i) \tag{14}$$

Similarly, the linear acceleration at the center of mass of each component can be expressed as Equation (15)

$${}^{i+1}\dot{\mathbf{v}}_{C_{i+1}} = {}^{i+1}\dot{\boldsymbol{\omega}}_{i+1} \times {}^{i+1}\mathbf{P}_{C_{i+1}} + {}^{i+1}\boldsymbol{\omega}_{i+1} \times ({}^{i+1}\boldsymbol{\omega}_{i+1} \times {}^{i+1}\mathbf{P}_{C_{i+1}}) + {}^{i+1}\dot{\mathbf{v}}_{i+1} \tag{15}$$

Finally, the total external force and torque on the i -th component can be expressed according to Newton’s equation and Euler’s equation:

$${}^{i+1}\mathbf{F}_{i+1} = m_{i+1} {}^{i+1}\dot{\mathbf{v}}_{C_{i+1}} \tag{16}$$

$${}^{i+1}\mathbf{N}_{i+1} = {}^{C_{i+1}}\mathbf{I}_{i+1} {}^{i+1}\dot{\boldsymbol{\omega}}_{i+1} + {}^{i+1}\boldsymbol{\omega}_{i+1} \times {}^{C_{i+1}}\mathbf{I}_{i+1} {}^{i+1}\boldsymbol{\omega}_{i+1} \tag{17}$$

where m_i and ${}^C_i\mathbf{I}_i$ represent the mass and the inertia matrix at the center of mass of the i -th connecting rod in its own coordinate system. ${}^i\mathbf{F}_i$ and ${}^i\mathbf{N}_i$ represent the resultant force and moment of the i -th component, respectively. Finally, the force relationship between adjacent components can be expressed as:

$${}^i\mathbf{f}_i = {}^{i+1}\mathbf{R} {}^{i+1}\mathbf{f}_{i+1} + {}^i\mathbf{F}_i \tag{18}$$

$${}^i n_i = {}^i N_i + {}_{i+1}^i R^{i+1} n_{i+1} + {}^i P_{C_i} \times {}^i F_i + {}^i P_{i+1} \times {}_{i+1}^i R^{i+1} f_{i+1} \tag{19}$$

where ${}^i f_i$ and ${}^i n_i$ represent the force and moment exerted by connecting rod $i - 1$ on connecting rod i . The torque τ_i at each joint can be expressed as Equation (20):

$$\tau_i = {}^i n_i^T \hat{Z}_i \tag{20}$$

The conservation equations for the linear and angular momentum of the system can be written as follows:

$$P = m_0 v_0 + \sum_{i=1}^n m_i \dot{r}_i = P_0 \tag{21}$$

$$L = ({}^{CM} I_0 {}^{CM} \omega_0 + r_0 \times m_0 \dot{r}_0) + \sum_{i=1}^n ({}^{CM} I_i {}^{CM} \omega_i + r_i \times m_i \dot{r}_i) = L_0 \tag{22}$$

where the linear and angular momenta are represented by P and L , respectively, with initial values P_0 and L_0 . The terms ${}^{CM} I_i$ and ${}^{CM} \omega_i$ represent the inertial tensor and angular velocity of the i -th component in the inertial system. To summarize, the dynamic equation of the space robot can be expressed in the following state-space form (disregarding gravity effects):

$$\tau = M(\Theta) \ddot{\Theta} + B(\Theta) (\dot{\Theta} \dot{\Theta}) + C(\Theta) (\dot{\Theta}^2) \tag{23}$$

where $M(\Theta)$, $B(\Theta)$, and $C(\Theta)$ are all complex functions about joint space Θ . Among them, $M(\Theta) \in \mathbb{R}^{n \times n}$ is the mass matrix of manipulators, $B(\Theta) \in \mathbb{R}^{n \times \frac{n(n-1)}{2}}$ is the Coriolis coefficient matrix, and $C(\Theta) \in \mathbb{R}^{n \times n}$ is the centrifugal force's coefficient matrix. In addition, $(\dot{\Theta} \dot{\Theta})$ and $(\dot{\Theta}^2)$ can be expressed by Equations (24) and (25):

$$(\dot{\Theta} \dot{\Theta}) = [\dot{\theta}_1 \dot{\theta}_2 \quad \dot{\theta}_1 \dot{\theta}_3 \quad \dots \quad \dot{\theta}_{n-1} \dot{\theta}_n]^T \tag{24}$$

$$(\dot{\Theta}^2) = [\dot{\theta}_1^2 \quad \dot{\theta}_2^2 \quad \dots \quad \dot{\theta}_n^2]^T \tag{25}$$

The dynamic equation of the mini space robot can be expressed using an alternative formulation, as shown in Equation (26):

$$\begin{bmatrix} M_b & M_{bm} \\ M_{mb} & M_m \end{bmatrix} \begin{bmatrix} \ddot{\Theta}_b \\ \ddot{\Theta}_m \end{bmatrix} + \begin{bmatrix} C_b & C_{bm} \\ C_{mb} & C_m \end{bmatrix} \begin{bmatrix} \dot{\Theta}_b \\ \dot{\Theta}_m \end{bmatrix} = \begin{bmatrix} B_b & 0 \\ 0 & B_m \end{bmatrix} \begin{bmatrix} \tau_b \\ \tau_m \end{bmatrix} \tag{26}$$

Here, M_b is the matrix corresponding to the overall spacecraft–manipulator system at a system configuration Θ . M_m is the generalized mass matrix corresponding to the manipulator, and $M_{bm} = M_{mb}^T$ is the matrix representing the coupled inertia between the base and the manipulator.

If

$$\tau' = B(\Theta) (\dot{\Theta} \dot{\Theta}) + C(\Theta) (\dot{\Theta}^2), \tag{27}$$

then there is

$$\ddot{\Theta} = M^{-1}(\Theta) (\tau - \tau') \tag{28}$$

Thus, the forward dynamic system of the mini space robot is obtained. The dynamic simulation model of the mini space robot is illustrated in Figure 2.

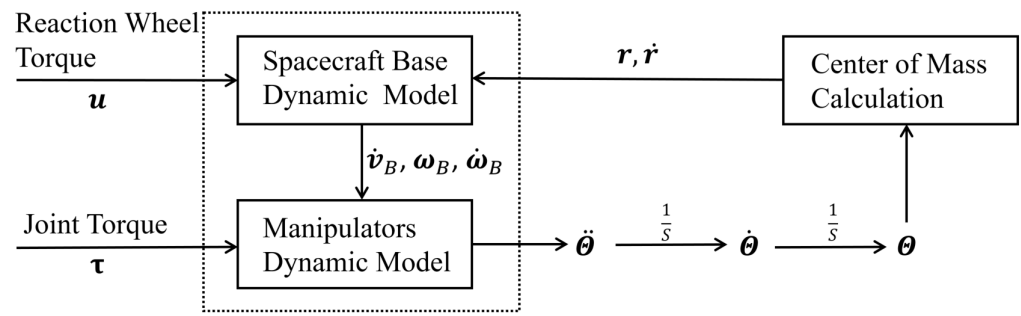


Figure 2. Schematic diagram of the dynamic model of the mini space robot.

3. Design of Parameter Adaptive Controller

In this section, the adaptive controller is designed based on the dynamics of the mini space robot. The reference model in the controller is assumed to be the same as the real physical model but the specific mass characteristic parameters cannot be determined. In order to enable the controller to update the dynamic parameters online, the linear separation of the parameters of the dynamic model is necessary.

3.1. Adaptive Controller Design

If the dynamic parameters are accurately known, the linearized equation represented by $Y(\Theta, \dot{\Theta}, \ddot{\Theta})\pi = u$ can be used to replace the inverse dynamics of the space robot, which facilitates the design of the controller. The feasibility of its inertial parameter separation is demonstrated in Appendix A. On the other hand, the linear separation expression is also convenient for a computer to iterate and update the parameters. Therefore, the nonlinear equation of motion can be expressed in a linear separation form with a suitable set of constant dynamic parameters such as Equation (29):

$$M(\Theta)\ddot{\Theta} + D(\Theta)\dot{\Theta} = Y(\Theta, \dot{\Theta}, \ddot{\Theta})\pi = u \tag{29}$$

where

$$D(\Theta)\dot{\Theta} = B(\Theta)(\dot{\Theta}\dot{\Theta}) + C(\Theta)(\dot{\Theta}^2) \tag{30}$$

Assuming that the computational model and the kinetic model are consistent, the control law can be considered:

$$u = M(\Theta)\ddot{\Theta}_r + D(\Theta)\dot{\Theta}_r + K_D\sigma \tag{31}$$

where K_D is a positive definite matrix; then, choose

$$\dot{\Theta}_r = \dot{\Theta}_d + \Lambda\tilde{\Theta} \quad \ddot{\Theta}_r = \ddot{\Theta}_d + \Lambda\dot{\tilde{\Theta}} \tag{32}$$

where Λ is a positive definite matrix, and $\dot{\Theta}_d$ and $\ddot{\Theta}_d$ represent the desired angular velocity and angular acceleration generated by the target trajectory. $\dot{\Theta}_r$ represents an intermediate reference variable generated by tracking the target angular velocity and angular acceleration, and the value of that variable is based on both the expected velocity and the position tracking error, and the representation of $\ddot{\Theta}_r$ is the same. Equation (32) expresses the nonlinear compensation and coupling terms as functions of the desired velocity and acceleration, modified by the current state of the space robot. If

$$\sigma = \dot{\Theta}_r - \dot{\Theta} = \dot{\tilde{\Theta}} + \Lambda\tilde{\Theta}, \tag{33}$$

then substitute Equations (31) into (29) to obtain:

$$M(\Theta)\ddot{\sigma} + D(\Theta)\dot{\sigma} + K_D\sigma = 0 \tag{34}$$

The Lyapunov function can be written as:

$$V(\sigma, \tilde{\Theta}) = \frac{1}{2}\sigma^T M(\Theta)\sigma + \frac{1}{2}\tilde{\Theta}^T W \tilde{\Theta} > 0 \quad \forall \sigma, \tilde{\Theta} \neq 0 \tag{35}$$

where $W \in \mathbb{R}^{n \times n}$ is a symmetric positive definite matrix. The time derivative of V along the trajectory of Equation (34) is:

$$\begin{aligned} \dot{V} &= \sigma^T M(\Theta)\dot{\sigma} + \frac{1}{2}\sigma^T \dot{M}(\Theta)\sigma + \tilde{\Theta}^T W \dot{\tilde{\Theta}} \\ &= -\sigma^T K_D \sigma + \tilde{\Theta}^T W \dot{\tilde{\Theta}} \end{aligned} \tag{36}$$

If $W = 2\Lambda K_D$ is chosen, then:

$$\dot{V} = -\dot{\tilde{\Theta}}^T K_D \dot{\tilde{\Theta}} - \tilde{\Theta}^T \Lambda K_D \tilde{\Theta} \tag{37}$$

The formula is 0 if and only if $\tilde{\Theta} = 0$ and $\dot{\tilde{\Theta}} = 0$, thus indicating that the time derivative is negative definite. It can be seen that the origin of the state space $[\tilde{\Theta}^T \quad \sigma^T]^T$ is globally asymptotically stable, so the control law can be adaptively established according to the parameter vector π . The control law expressed by Equation (31) can be expressed as:

$$\begin{aligned} u &= \hat{M}(\Theta)\ddot{\Theta}_r + \hat{D}(\Theta)\dot{\Theta}_r + K_D \sigma \\ &= Y(\Theta, \dot{\Theta}_r, \ddot{\Theta}_r)\hat{\pi} + K_D \sigma \end{aligned} \tag{38}$$

where $\hat{\pi}$ represents the available estimate of the inertial parameter, and the corresponding \hat{M} and \hat{D} represent the estimated terms in the dynamic model.

3.2. Correction and Update of Inertial Parameters

The available inertial parameter vector $\hat{\pi}$ cannot always accurately describe the mass characteristics of the system, and it is necessary to correct the parameters after obtaining the on-orbit control feedback information to improve the accuracy of the controller.

3.2.1. Recursive Least-Squares Method

On the premise that the state of the space robot system can be accurately feedback, the recursive least-squares method can identify the system parameters conveniently. After obtaining the torque and system state variables of each joint, the available inertial parameter vector $\hat{\pi}$ of the k -th control cycle can be calculated according to the iterative least-squares method, as shown in Equation (39), which can be derived by the basic least-squares method:

$$\hat{\pi}_k = \hat{\pi}_{k-1} + K_k[\tau_k - Y_k(\Theta_k, \dot{\Theta}_k, \ddot{\Theta}_k)\hat{\pi}_{k-1}] \tag{39}$$

where the gain matrix K_k is shown in Equation (40):

$$K_k = G_{k-1} Y_k^T [Y_k G_{k-1} Y_k^T + E]^{-1} \tag{40}$$

and the G_k is the intermediate variable matrix:

$$G_k = [E - K_k Y_k] G_{k-1} \tag{41}$$

Equation (39) shows that the parameter estimate $\hat{\pi}$ at time k is equal to the parameter estimate π at time $k - 1$ plus a correction term, which is proportional to the difference between the torque vector measured at time k and $Y_k(\Theta_k, \dot{\Theta}_k, \ddot{\Theta}_k)\hat{\pi}_{k-1}$. The schematic diagram of the control loop based on the recursive least-squares parameter-adaptive controller is shown in Figure 3:

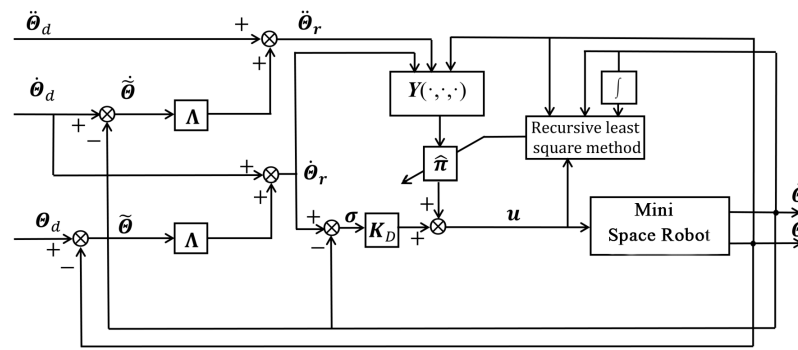


Figure 3. Recursive least-squares parameter-adaptive controller.

3.2.2. Parameter Update Based on Lyapunov’s Method

The key for the controller to identify inertial parameters is to provide an accurate parameter adaptation law, for which the Lyapunov method can provide an effective calculation method. According to Equation (38), the matrix Y does not depend on the actual value of the joint acceleration but on the expected value of the joint acceleration, so the problem of directly measuring the acceleration is avoided. Therefore, the Lyapunov function expressed by Equation (35) can be expressed as:

$$V(\sigma, \tilde{\Theta}, \tilde{\pi}) = \frac{1}{2} \sigma^T M(\Theta) \sigma + \frac{1}{2} \tilde{\Theta}^T \Lambda K_D \tilde{\Theta} + \frac{1}{2} \tilde{\pi}^T K_\pi \tilde{\pi} > 0 \quad \forall \sigma, \tilde{\Theta}, \tilde{\pi} \neq 0 \tag{42}$$

In Equation (42), the inertial parameter’s error vector term of the space robot is added to the Lyapunov function and K_π is a symmetric positive definite matrix. The time derivative of V along the trajectory of the system is:

$$\dot{V} = -\dot{\tilde{\Theta}}^T K_D \dot{\tilde{\Theta}} - \tilde{\Theta}^T \Lambda K_D \dot{\tilde{\Theta}} + \tilde{\pi}^T [K_\pi \dot{\tilde{\pi}} - Y_k^T(\Theta_k, \dot{\Theta}_k, \ddot{\Theta}_k) \sigma] \tag{43}$$

To make the last term of Equation (43) zero, we can make:

$$\dot{\tilde{\pi}} = K_\pi^{-1} Y_k^T(\Theta_k, \dot{\Theta}_k, \ddot{\Theta}_k) \sigma \tag{44}$$

Finally, the parameter adaptation law with Equation (38) as the control law in the mini space robot model represented by Equation (26) is obtained as:

$$\dot{\hat{\pi}} = K_\pi^{-1} Y_k^T(\Theta_k, \dot{\Theta}_k, \ddot{\Theta}_k) (\dot{\tilde{\Theta}} + \Lambda \tilde{\Theta}) \tag{45}$$

The schematic diagram of the control loop of the parameter-adaptive controller based on the Lyapunov method is shown in Figure 4.

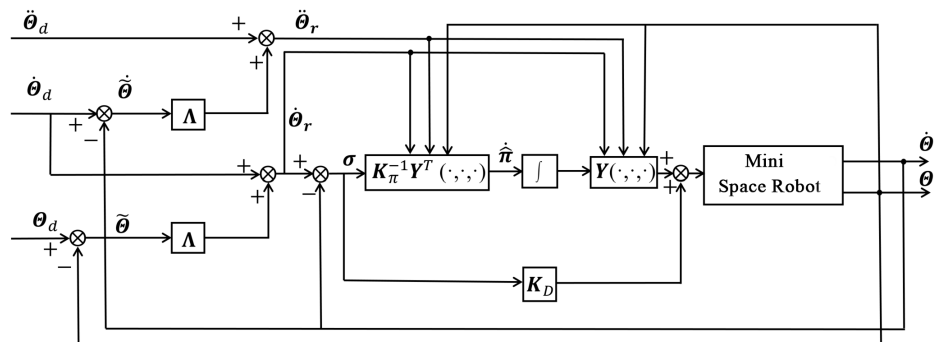


Figure 4. Lyapunov method based parameter-adaptive controller.

4. Numerical Simulation

In this section, an existing mini space robot with a four-DOF manipulator was modeled, as depicted in Figure 5. The satellite base and the antenna served as the primary objects of interest in the simulation presented in this paper. The satellite base requires precise positioning in a specific attitude, while the antenna's center needs to be accurately oriented in a specified direction to fulfill mission-specific requirements.

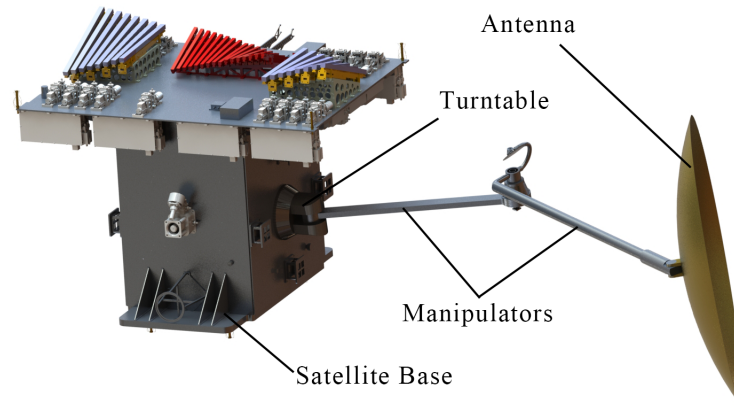


Figure 5. Mini space robot model.

Disregarding component interference, joints θ_1 , θ_3 , and θ_4 had a rotational range from -180° to 180° within their respective motion spaces, while joint 2 had a range from 0° to 180° . The installation positions and rotation ranges of the joints are illustrated in Figure 6. Table 1 presents the mass properties, centroid positions ${}^iP_{C_i}$, and Denavit–Hartenberg parameters of the bodies. Furthermore, for the sake of convenience, a simplified model and component labeling were utilized. The spacecraft base was denoted as B_0 , while the accessory parts were labeled as B_i (with subscripts). The coordinate positions of the mini space robot are depicted in Figure 7.

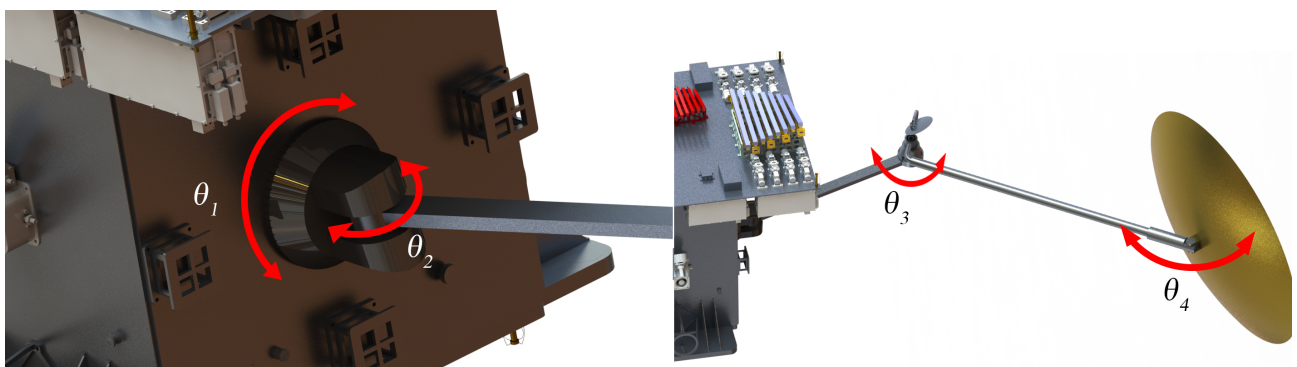


Figure 6. Schematic diagram of the joint's installation position and rotation.

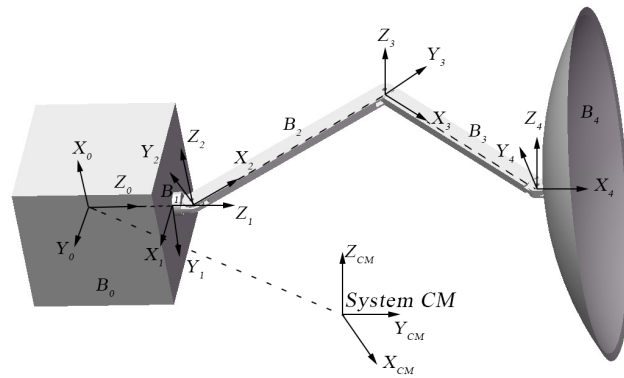


Figure 7. Simplified model of the space robot.

Table 1. The mass properties of the space robotic system.

Parameters		B ₀	B ₁	B ₂	B ₃	B ₄
Mass (kg)		131.734	2.082	24.569	25.280	134.078
ⁱ P _{C_i} (m)	ⁱ x	−0.045	0	0.496	0.500	0.149
	ⁱ y	0.101	0	0	0	0
	ⁱ z	−0.061	0.035	0	0	0
ⁱ I _{<i>i</i>} (kg · m ²)	I_{xx}	7.610	0.003	0.017	0.017	36.254
	I_{yy}	11.721	0.003	1.949	2.090	18.637
	I_{zz}	11.404	0.002	1.959	2.101	18.637
	I_{xy}	−1.773	0	0	0	0
	I_{xz}	1.799	0	0	0	0
	I_{yz}	0.652	0	0	0	0
Twist angle α _{<i>i</i>} (rad)		0	0	π/2	0	0
Length of links a _{<i>i</i>} (m)		0.320	0.056	1.000	1.000	0
Offset of links links d _{<i>i</i>} (m)		0	0	0	0	0
Joint angle θ _{<i>i</i>} (rad)		[α, β, γ] ^T	θ ₁ (t)	θ ₂ (t)	θ ₃ (t)	θ ₄ (t)

4.1. Influence of Prior Inertia Parameters on Controller Performance

The general orientation control task of a space robot is to perform attitude control while the manipulators are moving. In the first simulation, the mini space robot concurrently performed satellite attitude adjustment and end-effector antenna pointing tasks. This task exhibited generality in the on-orbit orientation of mini space robots, where other satellite-manipulator on-orbit maneuver modes can be regarded as submodes of this task. During this process, the controller utilized both the real inertial parameters and the prior inertial parameters with a certain level of error to control the same process. However, the controller did not update the parameters during the control process. Suppose the initial attitude of the mini space robot is $\Theta_0 = [0^\circ \ 0^\circ \ 0^\circ \ 90^\circ \ 45^\circ \ 45^\circ \ 0^\circ]$ and the target attitude is $\Theta_f = [10^\circ \ 0^\circ \ 0^\circ \ 90^\circ \ 55^\circ \ 55^\circ \ 10^\circ]$. Assume a certain ratio error exists between the prior inertial parameters and the actual parameters, following a normal distribution. In this paper, a three-time standard deviation was chosen as the error range, as shown below:

$$\pi_{priori} \sim N(\pi_{real}, \frac{1}{3}\eta\pi_{real}) \tag{46}$$

where π_{priori} and π_{real} represent the prior inertial parameters and real inertial parameters, respectively, and η represents the ratio of error in the prior inertial parameters. Initially, the controller carried out the orientation task using the real inertial parameters. The time-varying curves of the spacecraft base’s attitude and the angles of each joint are depicted in Figure 8.

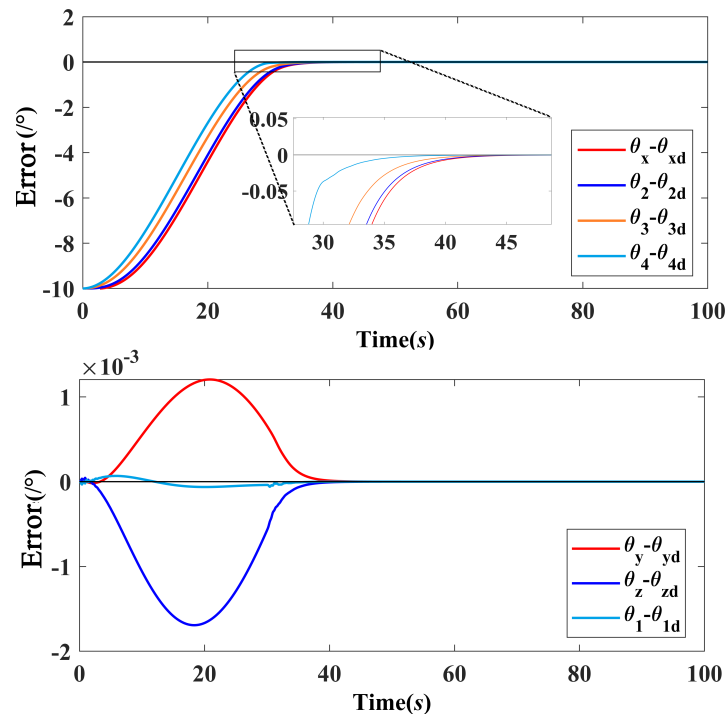


Figure 8. Orientation process under real inertial parameters.

The stability of the system under the control of the controller with real inertial parameters is demonstrated in Figure 8. Next, by setting η to 20%, four different sets of a priori inertial parameters were randomly generated. Using the satellite’s attitude error as an example, Figure 9 illustrates the performance of the controller under these different sets of inertial parameters.

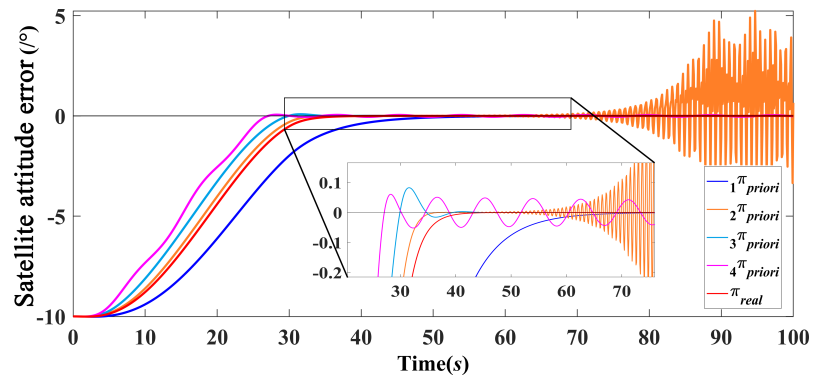


Figure 9. Attitude control error under different inertial parameters.

The red curve in Figure 9 represents the attitude error of the controller under the real inertial parameters, and the $i\pi_{priori}$'s, respectively, represent the corresponding control curves under the i -th group of prior inertia parameters. It can be seen from Figure 9 that without updating the prior parameters, inaccurate parameters lead to unstable control results. In addition, the inaccurate inertial parameters cause the controller to produce different degrees of overshoot, oscillation, slow convergence, and even divergence. It can be proved that the controller depends on the accuracy of the inertial parameters, and it is particularly important to obtain accurate inertial parameters or parameters that can be accurately equivalent to the system.

4.2. Influence of Weight Coefficient Λ on Controller Performance

The presence of Λ in Equation (32) allows the nonlinear compensation and coupling terms to be expressed as functions of desired velocity and acceleration, corrected by the current state of the spacecraft ($\dot{\Theta}$ and Θ). Λ represents a velocity-dependent weight based on the desired velocity and position tracking error. Therefore, the value of Λ has a significant influence on the controller performance. In the second simulation, the real inertial parameters were also used as the preset parameters of the controller to carry out the same orientation process as the first simulation. Then, taking the satellite's attitude error as an example, the control effect caused by different Λ 's is studied in Figure 10.

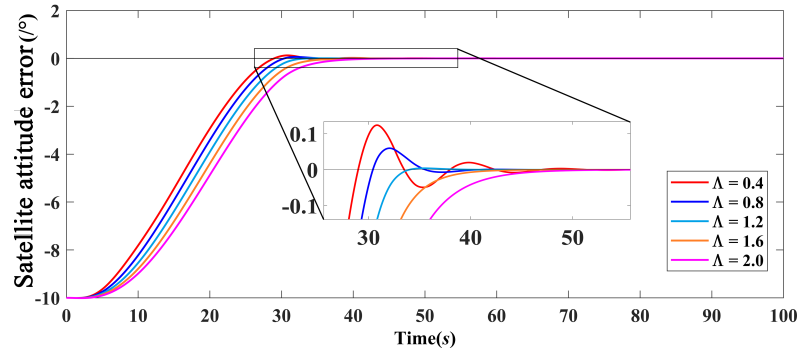


Figure 10. Influence of weight coefficient Λ on controller performance.

This paper examined the convergence curve of the system when Λ varied from 0.4 to 2.0. Figure 10 shows that the influence of parameter Λ on the control effect is mainly manifested in two aspects: overshoot and convergence speed. Within a certain range, the smaller the value of Λ , the lower the dependence on the tracking error. At that time, the system has a faster response speed and greater overshoot.

4.3. Inertial Parameter Update Using Recursive Least Square Method and Lyapunov Methods

Without loss of generality, the harmonic trajectories were tracked for each joint of the spacecraft in this part. The controller used recursive least-squares (RLS) and Lyapunov methods (LM) to adapt the inertial parameters during trajectory tracking, respectively. In addition, the stability of the two parameter adaptive methods was investigated by using the prior inertial parameters with different prior errors. Similar to the first part, we kept the y -axis and z -axis attitudes of the spacecraft base, and joint θ_1 was unchanged. At the same time, the attitude of the x -axis of the spacecraft base and the angles of θ_2 , θ_3 , and θ_4 tracked the sine-wave motion trajectory. The tracking process of both different parameter update methods is shown in Figure 11.

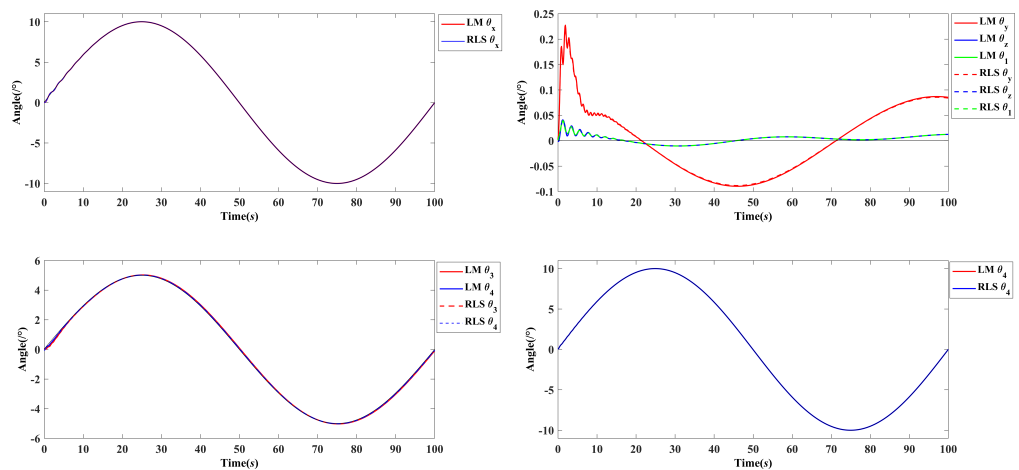


Figure 11. Tracking curves of the two methods.

Figure 11 shows the tracking curves of the two methods when the error of the prior inertial parameter vector is 20%. The black solid line in the figure represents the given tracking trajectory and the traces of the RLS and LM methods are distinguished using red and blue or solid and dashed lines, respectively. Both parameter update methods can track the given trajectory accurately, and the curves of the two methods have a high degree of coincidence. Since the most important difference between the RLS and the Lyapunov methods for updating parameters is whether there is torque feedback information, it was necessary to further investigate the influence of the accuracy of the prior parameters on the control results of the two different parameter updating methods. In this process, the change curves of the inertial parameters of the satellite base in the two control algorithms are shown in Figure 12. It can be seen from Figure 12 that the estimation accuracy of the inertial parameters of the LM method was worse than that of the RLS method because there was no torque feedback information. Of course, the rotation of the y -axis and z -axis of the satellite base was not involved in that process (only the attitude of the y -axis and z -axis was kept stable), so the correction of I_{yy} and I_{zz} by both methods was inaccurate. It can be further shown that that approximation can only correct the reference dynamic model in the direction of motion.

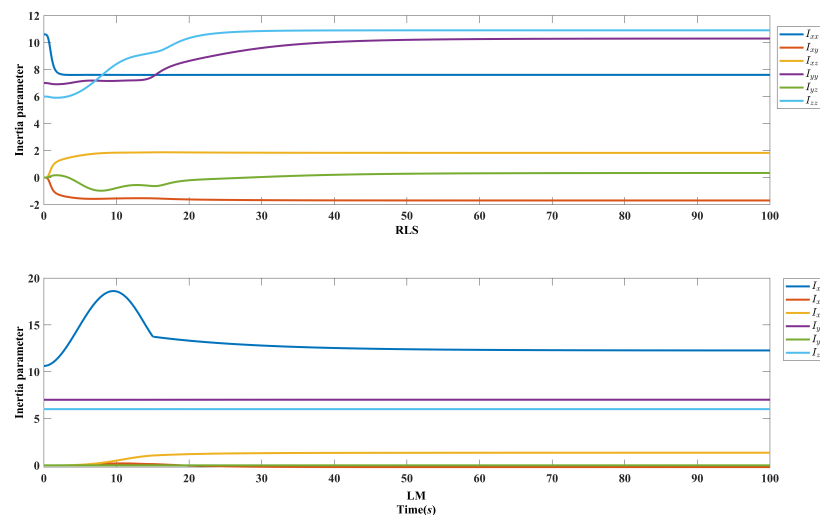


Figure 12. Inertia parameter's change curve.

Hence, building upon the first part, the error in the prior inertial parameters was incrementally increased, and the performance of both parameter update methods in tracking a harmonic trajectory was compared once again. Six sets of different a priori inertial parameters were randomly generated within the range of η from 10% to 40%. The recursive least-squares (RLS) and Levenberg–Marquardt (LM) algorithms were employed to simulate the harmonic trajectory tracking for each parameter group. Subsequently, the tracking errors are presented in Figure 13 for comparison.

In Figure 13, the solid and dashed lines represent the tracking error curves of the recursive least-squares (RLS) and Levenberg–Marquardt (LM) methods, respectively. The results in Figure 13 demonstrate that when the error in the prior inertial parameters is small, the two parameter update methods exhibit similar control outcomes. However, as η gradually increases, the Lyapunov parameter update method introduces fluctuations, and the tracking error gradually increases during the convergence process. Conversely, the RLS method is less affected by the error in the prior parameters and only exhibits slight oscillations when η approaches 30%.

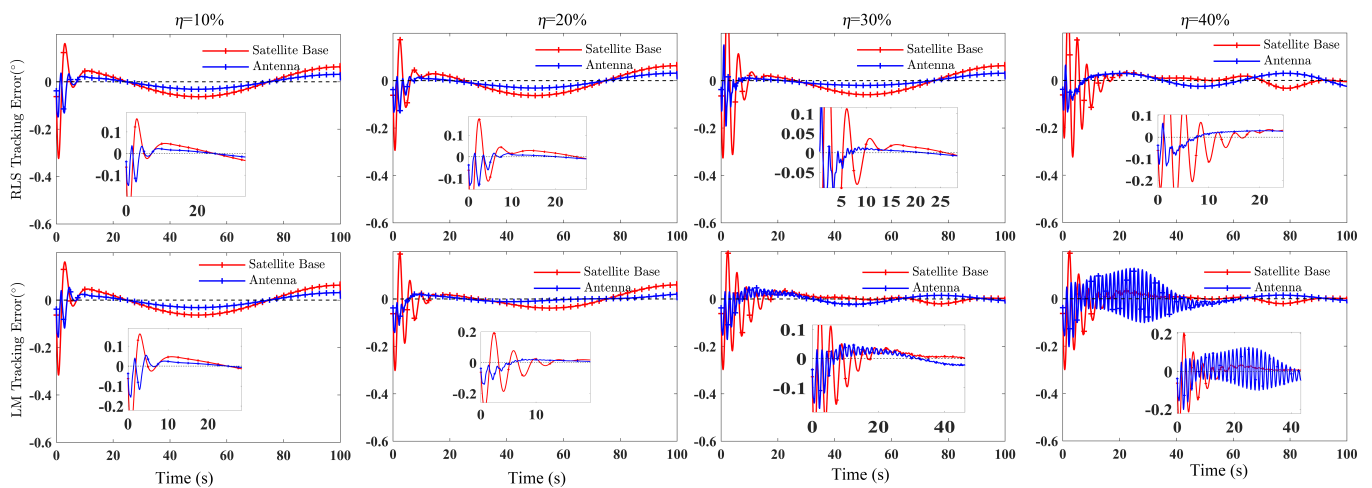


Figure 13. Comparison of the adaptability of the prior inertia parameters of the two methods to different initial errors.

Equations (39) and (44) show that neither the RLS nor the Lyapunov method can obtain the most accurate system inertial parameter vector in principle, which is different from the on-orbit inertial parameter identification. These methods only update the parameters to approximate the system dynamics captured by the $Y(\Theta, \dot{\Theta}, \ddot{\Theta})\pi$ term. From the simulation results, both the recursive least squares method and the Lyapunov method effectively adapted the inertial parameters, enabling the system to stably track the desired trajectory. The RLS-based parameter-adaptive controller maintained stable control performance even when the prior inertial parameters were highly inaccurate ($\eta = 40\%$), but this relied on accurate torque feedback for each joint. Additionally, in terms of computational resources, the Lyapunov method did not require matrix inversion operations during parameter updates. In practical applications, engineers typically strive to maintain the measurement error of inertial parameters for each component within the range of 5% to 10%. Within this range, controllers based on Lyapunov methods may offer certain advantages.

In addition, 10,000 repeated experiments were performed in Matlab to evaluate the computation time spent by the two algorithms in each control cycle, which is shown in Table 2. It can be seen that the average calculation time of the two algorithms was on the order of 10^{-4} s, which is far less than the control calculation period of the satellite on-orbit operation (the calculation period of the satellite on-orbit control algorithm is usually 0.25–1 s).

Table 2. Algorithm calculation time comparison

Calculation Method	Calculation Time(s)			
	Mean	Var	Max	Min
RLS	3.202×10^{-4}	7.736×10^{-8}	22.63×10^{-3}	2.187×10^{-4}
LM	2.440×10^{-4}	1.325×10^{-8}	6.405×10^{-3}	1.832×10^{-4}

5. Conclusions

In this paper, a dynamic-approximation-based inertial-parameter-adaptive controller was studied for mini space robots. The dynamic model established by the combination of the VM method and Newton–Euler equations was proved to be able to play an important role in the numerical simulation and dynamics-based controller design. Two methods based on recursive least squares and Lyapunov were designed for updating the parameters of the adaptive controllers. It was verified by simulation that the accuracy of the inertial parameters had a significant impact on the performance of the controller. Moreover, the parameter Λ in the controller that expresses the degree of dependence on the current

state of the system also had an important influence on the convergence characteristics of the system. Finally, the parameter update methods based on iterative least-squares and Lyapunov methods both showed a strong tracking ability in harmonic-tracking simulations. The difference was that the Lyapunov method had a poor adaptive ability to inaccurate prior inertial parameters, while the recursive least-squares method required more hardware resources such as accurate joint torque feedback and more computing power. The inertial-parameter-adaptive controller proposed in this paper can accurately approximate the dynamics of mini space robots and has certain application prospects in this field.

Author Contributions: Conceptualization, Y.L. and L.T.; methodology, Y.L. and L.T.; software, Y.L. and L.T.; validation, Y.L. formal analysis, Y.L. and Z.J.; investigation, Y.L. and Z.J.; resources, Y.L.; data curation, Y.L. and L.T.; writing—original draft preparation, Y.L. and L.T.; writing—review and editing, Y.L. and Z.J.; visualization, Y.L.; supervision, L.T.; project administration, Z.J.; funding acquisition, Z.J. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Key Laboratory of Micro-Nano Satellite Research Zhejiang Province and Huanjiang Laboratory.

Acknowledgments: This research was supported by “Zhejiang Provincial Key Laboratory of Micro-nano Satellite Research”.

Conflicts of Interest: There is no conflict of interest.

Appendix A

The dynamic equations derived by the Newton–Euler method are not linear equations with respect to the inertial parameters. The existence of the $m_i {}^i P_{C_i}^2$ term in Equation (19) makes the inertial parameters and dynamics nonlinear, so in order to eliminate this term, Equation (19) can be transformed:

$$\begin{aligned} & {}^i N_i + {}^i P_{C_i} \times {}^i F_i \\ &= {}^{C_i} I_i \dot{\omega}_i + {}^i \omega_i \times ({}^{C_i} I_i \omega_i) + {}^i P_{C_i} \times (m_i \dot{v}_{C_i}) \\ &= {}^{C_i} I_i \dot{\omega}_i + {}^i \omega_i \times ({}^{C_i} I_i \omega_i) + m_i {}^i P_{C_i} \times [{}^i \dot{v}_i + {}^i \dot{\omega}_i \times {}^i P_{C_i} + {}^i \omega_i \times ({}^i \omega_i \times {}^i P_{C_i})] \end{aligned} \quad (A1)$$

The third term in Equation (A1) can be expanded to obtain:

$$\begin{aligned} & m_i {}^i P_{C_i} \times [{}^i \dot{v}_i + {}^i \dot{\omega}_i \times {}^i P_{C_i} + {}^i \omega_i \times ({}^i \omega_i \times {}^i P_{C_i})] \\ &= m_i {}^i P_{C_i} \times {}^i \dot{v}_i + m_i {}^i P_{C_i} \times ({}^i \dot{\omega}_i \times {}^i P_{C_i}) + m_i {}^i P_{C_i} \times [{}^i \omega_i \times ({}^i \omega_i \times {}^i P_{C_i})] \end{aligned} \quad (A2)$$

The terms $m_i {}^i P_{C_i} \times ({}^i \dot{\omega}_i \times {}^i P_{C_i})$ and $m_i {}^i P_{C_i} \times [{}^i \omega_i \times ({}^i \omega_i \times {}^i P_{C_i})]$ in Equation (A2) can be written as:

$$\begin{aligned} & m_i {}^i P_{C_i} \times ({}^i \dot{\omega}_i \times {}^i P_{C_i}) \\ &= {}^i P_{C_i} \times ({}^i \dot{\omega}_i \times m_i {}^i P_{C_i}) \\ &= m_i ({}^i P_{C_i}^T {}^i P_{C_i} E - {}^i P_{C_i} {}^i P_{C_i}^T) {}^i \dot{\omega}_i \end{aligned} \quad (A3)$$

and:

$$\begin{aligned} & m_i {}^i P_{C_i} \times [{}^i \omega_i \times ({}^i \omega_i \times {}^i P_{C_i})] \\ &= {}^i \omega_i \times [{}^i P_{C_i} \times ({}^i \omega_i \times m_i {}^i P_{C_i})] + ({}^i \omega_i \times m_i {}^i P_{C_i}) \times ({}^i \omega_i \times {}^i P_{C_i}) \\ &= {}^i \omega_i \times [m_i ({}^i P_{C_i}^T {}^i P_{C_i} E - {}^i P_{C_i} {}^i P_{C_i}^T) {}^i \dot{\omega}_i] \end{aligned} \quad (A4)$$

Applying the parallel axis theorem, Equation (29) can be expressed as:

$$\begin{aligned}
 & {}^iN_i + {}^iP_{C_i} \times {}^iF_i \\
 &= {}^{C_i}I_i {}^i\dot{\omega}_i + {}^i\omega_i \times ({}^{C_i}I_i {}^i\omega_i) + m_i {}^iP_{C_i} \times {}^i\dot{v}_i + m_i ({}^iP_{C_i}^T {}^iP_{C_i} E - {}^iP_{C_i} {}^iP_{C_i}^T) {}^i\dot{\omega}_i \\
 &\quad + {}^i\omega_i \times [m_i ({}^iP_{C_i}^T {}^iP_{C_i} E - {}^iP_{C_i} {}^iP_{C_i}^T) {}^i\dot{\omega}_i] \\
 &= {}^{O_i}I_i {}^i\dot{\omega}_i + {}^i\omega_i \times ({}^{O_i}I_i {}^i\omega_i) + m_i {}^iP_{C_i} \times {}^i\dot{v}_i
 \end{aligned} \tag{A5}$$

Thus, Equation (19) can be expressed in the form of Equation (A6):

$${}^i n_i = {}^{O_i}I_i {}^i\dot{\omega}_i + {}^i\omega_i \times ({}^{O_i}I_i {}^i\omega_i) + m_i {}^iP_{C_i} \times {}^i\dot{v}_i + {}_{i+1}R^{i+1} n_{i+1} + {}^iP_{i+1} \times {}_{i+1}R^{i+1} f_{i+1} \tag{A6}$$

Then, the inertia parameter of the i -th component can be represented by a vector:

$$\pi_i = [{}^{O_i}I_{xx} \quad {}^{O_i}I_{xy} \quad {}^{O_i}I_{xz} \quad {}^{O_i}I_{yy} \quad {}^{O_i}I_{yz} \quad {}^{O_i}I_{zz} \quad m_i {}^iP_{C_{ix}} \quad m_i {}^iP_{C_{iy}} \quad m_i {}^iP_{C_{iz}} \quad m_i]^T \tag{A7}$$

Now, the equation does not contain the $m_i {}^iP_{C_i}^2$ term. The dynamic equation of the end part can be linearly separated as:

$${}^n n_n = [K({}^n\dot{\omega}_n) + S({}^n\omega_n)K({}^n\omega_n) \quad -S({}^n\dot{v}_n) \quad \mathbf{0}] \pi_n \tag{A8}$$

where the functions $K(a)$ and $S(a)$ are expressed as:

$$K(a) = \begin{bmatrix} a_x & a_y & a_z & 0 & 0 & 0 \\ 0 & a_x & 0 & a_y & a_z & 0 \\ 0 & 0 & a_x & 0 & a_y & a_z \end{bmatrix} \quad S(a) = \begin{bmatrix} 0 & -a_z & a_y \\ a_z & 0 & -a_x \\ -a_y & a_x & 0 \end{bmatrix}$$

The torque of each joint of the space robot can be expressed as Equation (A9), and Y_{ij} in the coefficient matrix is summarized as Equation (A10):

$$\begin{bmatrix} {}^b n_b \\ {}^1 n_1 \\ {}^2 n_2 \\ \vdots \\ {}^n n_n \end{bmatrix} = \begin{bmatrix} Y_{bb} & Y_{b1} & Y_{b2} & \cdots & Y_{bn} \\ \mathbf{0} & Y_{11} & Y_{12} & \cdots & Y_{1n} \\ \mathbf{0} & \mathbf{0} & Y_{22} & \cdots & Y_{2n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & Y_{nn} \end{bmatrix} \begin{bmatrix} \pi_b \\ \pi_1 \\ \pi_2 \\ \vdots \\ \pi_n \end{bmatrix} \tag{A9}$$

$$Y_{ij} = \begin{cases} [K({}^i\dot{\omega}_i) + S({}^i\omega_i)K({}^i\omega_i) \quad -S({}^i\dot{v}_i) \quad \mathbf{0}] & i = j \\ [K({}^i\dot{\omega}_j) + S({}^i\omega_j)K({}^i\omega_j) \quad S({}^i\dot{\omega}_j) + S({}^i\omega_j)S({}^i\omega_j) - S({}^i\dot{v}_j) \quad {}^i\dot{v}_j] & i \neq j \end{cases} \tag{A10}$$

Equation (A9) is abbreviated as:

$$\tau = Y(\Theta, \dot{\Theta}, \ddot{\Theta}) \pi \tag{A11}$$

where Y is a function of the mini space robots Θ , $\dot{\Theta}$, and $\ddot{\Theta}$, and only the kinetic parameters are included in π .

References

1. Nakasuka, S.; Miyata, K.; Tsuruda, Y.; Aoyanagi, Y.; Matsumoto, T. Discussions on attitude determination and control system for micro/nano/pico-satellites considering survivability based on Hodoyoshi-3 and 4 experiences. *Acta Astronaut.* **2018**, *145*, 75–174. [\[CrossRef\]](#)
2. Zhang, X.; Liu, J.; Tong, Y.; Liu, Y.; Ju, Z. Attitude Decoupling Control of Semifloating Space Robots Using Time-Delay Estimation and Supertwisting Control. *IEEE Trans. Aerosp. Electron. Syst.* **2021**, *57*, 4280–4295. [\[CrossRef\]](#)
3. Jin, R.; Rocco, P.; Chen, X.; Geng, Y. LPV-Based Offline Model Predictive Control for Free-Floating Space Robots. *IEEE Trans. Aerosp. Electron. Syst.* **2021**, *57*, 3896–3904. [\[CrossRef\]](#)

4. Chu, X.; Hu, Q.; Zhang, J. Path Planning and Collision Avoidance for a Multi-Arm Space Maneuverable Robot. *IEEE Trans. Aerosp. Electron. Syst.* **2021**, *54*, 217–232. [CrossRef]
5. Virgili-Llop, J.; Drew, J.V.; Zappulla, I.R.; Romano, M. Laboratory experiments of resident space object capture by a spacecraft–manipulator system-ScienceDirect. *Aerosp. Sci. Technol.* **2017**, *71*, 530–545. [CrossRef]
6. Jorgensen, G.; Bains, E. SRMS History, Evolution and Lessons Learned. In Proceedings of the AIAA SPACE 2011 Conference & Exposition, Long Beach, CA, USA, 27–29 September 2011.
7. Landzettel, K. MSS Ground Control Demo with MARCO. In Proceedings of the 6th International Symposium on Artificial Intelligence, Robotics and Automation in Space: iSAIRAS 2001, Montreal, QC, Canada, 18–21 June 2001.
8. Stieber, E.M.; Hunter, G.D.; Abramovici, A. Overview of the Mobile Servicing System for the International Space Station. *Artif. Intell. Robot. Autom. Space* **1999**, *440*, 37–42.
9. Stieber, M.F.; Trudel, C.P.; Hunter, D.G. Robotic systems for the International Space Station. In Proceedings of the International Conference on Robotics and Automation, Albuquerque, NM, USA, 25 April 1997; pp. 3068–3073.
10. Sasiadek, J.Z. Space robotics and manipulators—The past and the future. *Control Eng. Pract.* **1994**, *2*, 491–497. [CrossRef]
11. Piedboeuf, J.C.; Carufel, J.D.; Aghili, F.; Dupuis, E. Task verification facility for the Canadian special purpose dextrous manipulator. In Proceedings of the IEEE International Conference on Robotics and Automation, Detroit, MI, USA, 10–15 May 2002; pp. 1077–1083.
12. Wanga, X.; Shi, L.; Katupitiya, J. Coordinated Control of a Dual-arm Space Robot to Approach and Synchronise with the Motion of a Spinning Target in 3D Space. *Acta Astronaut.* **2020**, *176*, 99–110. [CrossRef]
13. Moghaddam, B.M.; Chhabra, R. On the guidance, navigation and control of in-orbit space robotic missions: A survey and prospective vision. *Acta Astronaut.* **2021**, *184*, 70–100. [CrossRef]
14. Yoji, U.; Yoshida, K. Resolved motion rate control of space manipulators with generalized Jacobian matrix. *IEEE Trans. Robot. Autom.* **1989**, *5*, 303–314.
15. Nenchev, D.; Umetani, Y.; Yoshida, K. Analysis of a Redundant Free-Flying Spacecraft/Manipulator System. *IEEE Trans. Robot. Autom.* **1992**, *8*, 1–6. [CrossRef]
16. Dubowsky, S.; Papadopoulos, E. The kinematics, dynamics, and control of free-flying and free-floating space robotic systems. *Robot. Autom. IEEE Trans.* **1993**, *9*, 531–542. [CrossRef]
17. Liang, B.; Xu, Y.; Bergerman, M. Mapping a space manipulator to a dynamically equivalent manipulator. *J. Dyn. Syst. Meas. Control* **1998**, *120*, 1–7. [CrossRef]
18. Yoshida, K.; Kurazume, R.; Umetani, Y. Dual arm coordination in space free-flying robot. In Proceedings of the ICRA, Sacramento, CA, USA, 9–11 April 1991.
19. Papadopoulos, E.; Moosavian, S. Dynamics and control of multi-arm space robots during chase and capture operations. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, Munich, Germany, 12–16 September 1994.
20. Yang, S.; Zhang, Y.; Chen, T.; Wen, H.; Jin, D. Assembly Strategy for Modular Components Using a Dual-Arm Space Robot with Flexible Appendages. *Aerospace* **2022**, *9*, 819. [CrossRef]
21. Walker, M.W.; Wee, L.B. Adaptive control of space-based robot manipulators. *Robot. Autom. IEEE Trans.* **1991**, *7*, 828–835. [CrossRef]
22. Wee, L.B.; Walker, M.W.; Mcclamroch, N.H. An articulated-body model for a free-flying robot and its use for adaptive motion control. *Robot. Autom. IEEE Trans.* **1997**, *13*, 264–277.
23. Liu, D.; Chen, L. Space Robot On-Orbit Operation of Insertion and Extraction Impedance Control Based on Adaptive Neural Network. *Aerospace* **2023**, *10*, 466. [CrossRef]
24. Wu, X.; Zhao, H.; Huang, B.; Li, J.; Liu, R. Minimum-learning-parameter-based anti-unwinding attitude tracking control for spacecraft with unknown inertia parameters. *Acta Astronaut.* **2021**, *179*, 498–508. [CrossRef]
25. Ls, A.; He, Y.B.; Ms, C.; Qg, D.; Xin, J.A. Robust control of a space robot based on an optimized adaptive variable structure control method. *Aerosp. Sci. Technol.* **2022**, *120*, 107267.
26. Xu, W.; Hu, Z.; Zhang, Y.; Liang, B. On-orbit identifying the inertia parameters of space robotic systems using simple equivalent dynamics. *Acta Astronaut.* **2017**, *132*, 131–142. [CrossRef]
27. Nabavi-Chashmi, S.Y.; Malaek, M.B. On the identifiability of inertia parameters of planar Multi-Body Space Systems. *Acta Astronaut.* **2018**, *145*, 199–215. [CrossRef]
28. Ma, O. On-Orbit Identification of Inertia Properties of Spacecraft Using Robotics Technology. 2006. Available online: https://www.researchgate.net/publication/235063328_On-Orbit_Identification_of_Inertia_Properties_of_Spacecraft_using_Robotics_Technology (accessed on 24 June 2023).
29. Nainer, C.; Garnier, H.; Gilson, M.; Evain, H.; Pittet, C. Parameter estimation of a gyroless micro-satellite from telemetry data. *Control Eng. Pract.* **2022**, *123*, 105–134. [CrossRef]
30. Nuthi, P.; Subbarao, K. Computational adaptive optimal control of spacecraft attitude dynamics with inertia matrix identification. In Proceedings of the American Control Conference, Boston, MA, USA, 6–8 July 2016.
31. Chang, H.; Huang, P.; Lu, Z.; Zhang, Y.; Meng, Z.; Liu, Z. Inertia Parameters Identification for Cellular Space Robot through Interaction. *Aerosp. Sci. Technol.* **2017**, *71*, 464–474. [CrossRef]

32. Yao, X.; Gang, T.; Ma, Y.; Qi, R. An adaptive actuator failure compensation scheme for spacecraft with unknown inertia parameters. In Proceedings of the Decision & Control, Maui, HI, USA, 10–13 December 2012.
33. Lei, R.H.; Chen, L. Adaptive fault-tolerant control based on boundary estimation for space robot under joint actuator faults and uncertain parameters. *Def. Technol.* **2019**, *5*, 964–971. [[CrossRef](#)]
34. Huang, P.; Wang, M.; Meng, Z.; Zhang, F.; Liu, Z. Attitude takeover control for post-capture of target spacecraft using space robot. *Aerosp. Sci. Technol.* **2016**, *51*, 171–180. [[CrossRef](#)]
35. Palma, P.; Seweryn, K.; Rybus, T. Impedance Control Using Selected Compliant Prismatic Joint in a Free-Floating Space Manipulator. *Aerospace* **2022**, *9*, 406. [[CrossRef](#)]
36. Ulrich, S.; Sasiadek, J.Z.; Barkana, I. Nonlinear adaptive output feedback control of flexible-joint space manipulators with joint stiffness uncertainties. *J. Guid. Control Dyn.* **2014**, *37*, 1961–1975. [[CrossRef](#)]
37. Marchionne, C.; Sabatini, M.; Gasbarri, P. GNC architecture solutions for robust operations of a free-floating space manipulator via image based visual servoing. *Acta Astronaut.* **2021**, *180*, 218–231. [[CrossRef](#)]
38. Liu, Y.; Jin, Z.; Teng, L. PSO-Based Time Optimal Rapid Orientation for Micronano Space Robot. *IEEE Trans. Aerosp. Electron. Syst.* **2023**, *59*, 1921–1934. [[CrossRef](#)]
39. Zhang, X.; Ling, K.; Lu, Z.; Zhang, X.; Liao, W.; Lim, W. Piece-wise affine MPC-based attitude control for a CubeSat during orbital manoeuvres. *Aerosp. Sci. Technol.* **2021**, *118*, 106997. [[CrossRef](#)]
40. Oestreich, C.E.; Linares, R.; Gondhalekar, R. Tube-Based Model Predictive Control with Uncertainty Identification for Autonomous Spacecraft Maneuvers. *J. Guid. Control Dyn.* **2023**, *46*, 6–20. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.