



Intelligent and Robust Control of Space Manipulator for Sustainable Removal of Space Debris

Shabadini Sampath^a , Dr Jinglang Feng^{a*} 

^a *Department of Mechanical and Aerospace Engineering, University of Strathclyde, 16 Richmond St, Glasgow G1 1XQ, shabadini.sampath@strath.ac.uk, jinglang.feng@strath.ac.uk*

* Corresponding Author

Abstract

This study focuses on enhancing the control precision and efficiency of a two-degree-of-freedom (2-DOF) space manipulator used for active space debris removal. The unpredictable space environment introduces large uncertainties, which introduces unique challenges beyond the capabilities of a standalone computed torque controller and degrades control performance. To address this problem, a robust controller is developed, integrating traditional techniques such as sliding mode and computed torque control with a Neural Network framework. This synergy leverages both methods' strengths—conventional controls' accuracy and Neural Network's adaptability. The integration of Neural Network-based sliding mode control complements the robustness of computed torque control by actively mitigating uncertainties and disturbances inherent in the space environment. The 2-DOF manipulator's state variables model the system dynamics, necessitating accurate relative motion estimation between the manipulator and debris. The global asymptotic stability of the developed algorithm is demonstrated through the Lyapunov theorem, guaranteeing error convergence to zero. The convergence, stability, precision, tracking errors, and responsiveness of the controller have been analysed and validated by the MATLAB Simulink simulations. The novel approach's performance effectiveness is substantiated by numerical simulations and a comparative analysis with conventional computed torque control. Outcomes highlight the superior precision and efficiency in manipulator tracking the trajectory, validating the integrated controller's potential for successful active space debris removal.

Keywords: Space Manipulator, Active Debris Removal, Intelligent Controllers, Neural Network, Sliding Mode Controller, Computed Torque Controller.

1. Introduction

The term "space debris" refers to fragments of human-made objects, including defunct satellites, discarded rocket stages, and particles resulting from collisions with other debris. These elements pose a threat to satellites in Earth's orbit. Furthermore, these particles have the ability to damage satellite and spacecraft components, leading to the creation of additional fragments over time. The ongoing launch of more satellites and spacecraft for diverse space missions is contributing to the steady growth of the space industry.

Spacecraft equipped with space manipulators play a pivotal role not only in clearing space debris but also in various on-orbit service missions, as well as the assembly, maintenance, and repair of spacecraft. The success of these missions relies on efficient control of space manipulator systems. In the context of on-orbit servicing, a space manipulator is tasked with executing intricate operations that necessitate real-time intelligent trajectory planning. However, the application of space manipulators in active space debris removal faces challenges.

When dealing with a non-controlled and tumbling target object from a distance, such an object does not convey any information to the spacecraft or chaser satellite. This lack of information makes it difficult to effectively employ space manipulators in missions

focused on active space debris removal. Consequently, this area remains an ongoing and open field of research.

Conventional spacecraft Guidance, Navigation, and Control (GNC) systems are primarily designed to follow commands from ground control with minimal onboard autonomy (Hao et al., 2021). Intelligent control techniques leverage artificial intelligence (AI) approaches, such as neural networks, fuzzy logic, and reinforcement learning to make real-time decisions, particularly when system information integration into traditional methods involving ground control becomes challenging. However, a complete transition to AI-based systems requires overcoming the challenges including adaptability in dynamic space environments and the need for extensive testing and validation to ensure safety and reliability in space environments. Therefore, an intermediate solution called an intelligent GNC system with AI components is being explored to replace conventional components for autonomous operations of free-floating space manipulators.

The intelligent GNC system faces two primary technical challenges: the limited computational power of hardware in space's radiation environment and constrained power supply, and the robustness of pre-trained neural networks for real space missions (Hao et al., 2021). Ensuring safety in manipulator operations requires constraining the joint angles within a designated

safe region, which adds another layer of complexity to controller design. The literature reviews various space robotics technologies for on-orbit services. For instance, DLR has developed a method for capturing non-cooperative, tumbling targets (Shan, Guo, and Gill, 2016), focusing on a single manipulator arm due to its practicality for ground testing and higher technology readiness level (TRL), despite drawbacks like collision risks and rendezvous requirements. Pathak et al. (2008) discuss diverse robust control strategies for free-floating space manipulators, highlighting the challenges of trajectory control for single-degree-of-freedom (DOF) manipulators. Liu et al., (2018) present a robust decentralised control strategy based on signal compensation and back-stepping to reduce tracking errors and demonstrate robustness. Zhang et al., (2022) demonstrate a strong robustness in the control strategy developed for controlling the motions of the space manipulator using Time-Delay Estimation (TDE). The combination of SMC and TDE improves tracking accuracy and robustness.

Survey papers also provide insight into existing work on intelligent robots. He, Li, and Chen (2017) survey human-centric intelligent robots' challenges, while Moosavian and Papadopoulos (2007) delve into the dynamics, modelling, planning, and control of free-floating robots in space, explaining various control algorithms. The following studies applied artificial intelligence techniques to control the space manipulator motions. Ertugrul and Kaynak, (1997a) demonstrate the efficiency of a neural network-based adaptive sliding mode controller to control the SCARA robot manipulator. Fuzzy-neural-network inherited sliding mode controllers were developed by Rong-Jong Wai and Muthusamy, (2013), Hui Hu and Peng-Yung Woo, (2006), and Sun et al., (2019) and (Tian and Collins, 2005) demonstrated the fuzzy and neuro-fuzzy-based control techniques for flexible manipulators. However, these papers didn't address uncertainties as a way to improve the control performance of space manipulator. This research will focus on developing an intelligent and also robust controller to accurately control the motion of the manipulator with uncertainties.

In on-orbit tasks, achieving precise trajectory tracking using space manipulators encounters difficulties due to uncertainties present in the space environment. These uncertainties can be categorized into two main types: aleatory and epistemic (Benke et al., 2018). Distinguishing epistemic uncertainties from aleatoric ones can be achieved by assessing whether additional knowledge can eliminate the uncertainty. The term "epistemic" originates from the Greek word "επιστήμη (epistēmē)," which roughly translates to knowledge (Hüllermeier and Waegeman, 2021). Epistemic uncertainty within a classical neural network typically refers to the uncertainty associated with neural network

parameters that could be diminished with more essential information provided by the dataset. On the other hand, aleatoric uncertainty emerges from the probabilistic prediction itself, as revealed through maximum likelihood inference. In the context of nonlinear systems, the design of manipulators and the selection of controllers are significantly influenced by the presence of these uncertainties. To tackle these uncertainties, intelligent and robust controllers are essential, and the advancement in this domain is still a challenge. Robustness, denoting the ability of a closed-loop system to remain unaffected by parameter variations, is one of the most valuable properties of a controller (Pathak et al., 2008). Hence, the study aims to develop an intelligent and robust controller for precise and robust control of space manipulators amidst disturbances like modelling errors, noise, and variations in system behaviour. The proposed controller, aided by Neural Networks, targets the mitigation of uncertainties to enhance manipulator control performance. It strives for rapid responsiveness and global asymptotic stability, ultimately advancing intelligent and robust control in space manipulators, crucial for on-orbit tasks. Neural networks learn from system behaviour and adapt controller parameters based on observed uncertainties. They can approximate nonlinear relationships between system inputs and outputs, allowing the controller to adjust in real time to varying conditions or uncertainties. By continuously learning and updating the controller gains, neural networks enhance the system's robustness against uncertainties that might otherwise disrupt the performance of traditional fixed-gain controllers.

The structure of this paper is organized as follows. Section II presents the modelling of a space manipulator by deriving its dynamic equations. Section III explains the theoretical design of the proposed control scheme, ensuring high-precision control in joint motions. In Section IV, the simulation results of the proposed controller are demonstrated, and a comparison is made with the conventional controller. Finally, Section V provides the concluding remarks of the paper.

2. Dynamical Modelling of a Space Manipulator

A spatial free floating 2-DOF space manipulator system is considered as it is convenient to test the effectiveness of the proposed controller, and it is illustrated in Fig. 1. It consists of a spacecraft base, two revolute joints, and an end effector used for capturing target objects. The spacecraft body frame $\{x, y\}$ and the end effector's body frame $\{x_e, y_e\}$ are attached to the spacecraft and end effector aligned with the inertial principal axes of the spacecraft and end effector, respectively. The revolute joints have angle variables denoted as q_1 and q_2 , which are perpendicular to the principal axes of the links.

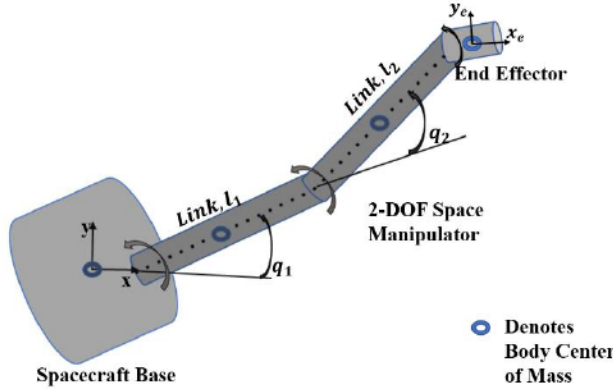


Fig. 1. Two-Degree-of-Freedom Spatial Free-Floating Space Manipulator System

2.1 Dynamical Equations of Motion for a Space Manipulator

The equations of motion's dynamics are obtained through the utilization of Lagrange's equation, formulated as:

$$Q_i = \frac{d}{dt} \left(\frac{\partial L}{\partial \dot{q}_i} \right) - \frac{\partial L}{\partial q_i} \quad (1)$$

In this equation, L denotes the Lagrangian of the system, representing the difference between kinetic and potential energy. Q_i represents non-conservative forces acting on the system, t symbolizes time, and q_i represents the vector of generalized coordinates. For the dynamic models used in this study, Q is considered a vector of applied torques, represented as τ . Thus, the dynamic equation is simplified into matrix form:

$$M(q)\ddot{q} + h(q, \dot{q}) + T_d = \tau \quad (2)$$

Here, $M(q)$ is the 2*2 inertia matrix, and $h(q, \dot{q})$ with 2*2 matrix represents the centrifugal and Coriolis terms:

$$h(q, \dot{q}) = C(q, \dot{q})\dot{q} + G(q) \quad (3)$$

The Coriolis term $C(q, \dot{q})$ and the gravity term $G(q)$ are included in the control system to develop a more accurate dynamical model. The term T_d represents a disturbance in the system dynamics and is depicted as a sine wave function in the MATLAB-Simulink simulation. Sine waves functions are often used to simulate external disturbances in the control theory as they represent the oscillatory disturbances, which are suitable for modelling various types of disturbances encountered in a real-world system (Cordero et al., 2022 and Kim et al., 2013). Additionally, $\tau(t)$ corresponds to the control torque applied to the space manipulator's joints. Several properties of the space manipulator are outlined as follows:

- Property 1: By appropriately defining the matrix $C(q, \dot{q})$, $\dot{M}(q) - 2C(q, \dot{q})$ is skew-symmetric and fulfils the condition:
 - For any vector x , $x^T [\dot{M}(q) - 2C(q, \dot{q})] x = 0, \forall x \in R^n$.
 - $C(q, \dot{q}) \dot{q}$ is quadratic with respect to \dot{q} and bounded as to $\|C(q, \dot{q}) \dot{q}\| \leq \mu_3(q) \|\dot{q}\|^2$, is a positive constant for revolute links.
- Property 2: The inertia matrix M is symmetric, positive definite, and satisfies the condition:
 - $M^T = M$.
 - Bounded as $\alpha_2 I_n \leq M(q) \leq \alpha_3 I_n$, $\forall q \in R^n$ where α_2 and α_3 are positive constants. For revolute links, they remain constants. $I_n \in R^{n \times n}$ denotes the identity matrix.
- Property 3: The gravity term G is bounded as $\|G(q)\| \leq \alpha_4(q)$, where $\alpha_4(q)$ is a positive constant for revolute links. It is independent of q .

3. Design of Controllers

The traditional control methodologies explored in this study comprise sliding mode control (SMC), and computed torque control (CTC). A novel strategy is developed, by merging both sliding mode control and computed control with a neural network, to derive a combined torque responsible for governing the joint movements of the space manipulator. The complex nonlinear dynamical model of space manipulators presents a challenge in effectively controlling them using conventional methods. Despite the competence of classic controllers, the inherent uncertainties in manipulators remain substantial. To address this, an intelligent approach involving a neural network is applied. This integration aims to mitigate uncertainties, sustain the controllers' robustness, enhance convergence speed, and elevate overall performance through uncertainty reduction. The workflow diagram illustrating the integration of neural network-based sliding mode control and computed torque control, termed NSMCTC, is presented in Fig. 2. The input for the trajectory in terms of q, \dot{q} and \ddot{q} is given to the controllers. The combined torque of the neural network and sliding mode controller is added to the computed torque from CTC to achieve a 100% convergence rate. The overall output of this proposed controller is given to the space manipulator to control its motions.

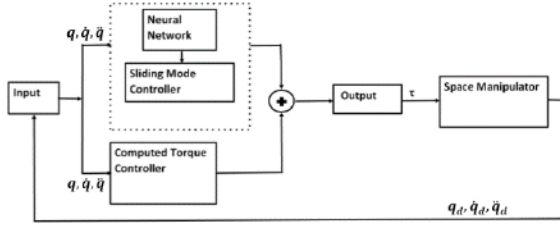


Fig. 2. Workflow Diagram of Proposed Controller, NSMCTC

3.1 Sliding Mode Control (SMC)

The sliding mode control methodology proves to be a potent approach for creating robust controllers suited for high-order nonlinear dynamic models operating within uncertain conditions. This technique originated in the former Soviet Union approximately four decades ago and has gained substantial international recognition over the past two decades. SMC offers distinct advantages including robustness, finite-time convergence, and reduced-order compensated dynamics. The fundamental principle of SMC involves constraining the manipulator's trajectory onto a hyperplane, skilfully guiding it to an asymptotic equilibrium point through controlled sliding. To enhance robust control performance, it is theoretically feasible to employ a high switching gain within the discontinuous function (Liu et al., 2021). However, this technique can lead to chattering issues characterized by undesirable oscillations with finite frequency and amplitude centered around a predefined switching manifold.

Chattering arises due to two primary reasons (Young, Utkin, and Ozguner, 1999):

- The presence of parasitic dynamics in the control system leads to high-frequency oscillations with limited amplitude. Consequently, this type of chattering can often be ignored during controller design.
- Switching non-idealities, such as delays in sampling time and transmission delays in networked control systems, can introduce high-frequency oscillations. These non-idealities need to be addressed as they contribute to the controller's excessive high-frequency oscillations.
- The unmodeled system dynamics present in the control system might trigger high-frequency chattering within the SMC. Therefore, the effectiveness of SMC is closely tied to a comprehensive understanding of system uncertainties and dynamics.

Consider the design of a nonlinear sliding variable using simple error dynamics. The tracking error, known as the filtered tracking error, can be defined as follows:

$$e = q - q_d \quad (4)$$

Upon taking the first and second derivatives, the following expressions emerge:

$$\dot{e} = \dot{q} - \dot{q}_d \quad (5)$$

$$\ddot{e} = \ddot{q} - \ddot{q}_d \quad (6)$$

where e represents the error between the desired position and the measured position. e and \dot{e} are assumed to remain bounded. In this context, q_d , \dot{q}_d and \ddot{q}_d denote the vectors corresponding to the desired position, velocity, and acceleration of the joint.

The pivotal aspect of SMC design revolves around creating a sliding manifold, sliding surface, or hyperplane. The formulation for the sliding surface is given by:

$$s = \Lambda e + \dot{e} \quad (7)$$

Here, Λ is a constant positive definite diagonal matrix, denoted as $\Lambda = \text{diag}(\Lambda_1, \dots, \Lambda_n)$ $\Lambda_i > 0$. The reference state can be expressed as:

$$\dot{q}_r = \dot{q} - s = \dot{q} - \Lambda e - \dot{e} = \dot{q}_d - \Lambda e \quad (8)$$

$$\ddot{q}_r = \ddot{q} - \dot{s} = \ddot{q} - \Lambda \dot{e} - \ddot{e} = \ddot{q}_d - \Lambda \dot{e} \quad (9)$$

Equations (4) and (5) are functions reliant on the manipulator dynamics, involving tracking position error and velocity error. The time derivative of the sliding surface is denoted by (Hui Hu and Peng-Yung Woo, 2006):

$$\dot{s} = \Lambda(\dot{q} - \dot{q}_d) - \ddot{q}_d + M^{-1}(q)[\tau - h(q, \dot{q}) - T_d] \quad (10)$$

Here, T_d stands for unmodeled uncertainty, assumed to be bounded to ensure adequate control performance.

From **property 1**, $\dot{M}(q) - 2C(q, \dot{q})\dot{q}$ constitutes a skew matrix satisfying $x^T[\dot{M}(q) - 2C(q, \dot{q})\dot{q}]x = 0$. Consequently, by multiplying both sides of the equation by M , the following equation is derived:

$$M\dot{s} = \tau - h(q, \dot{q}) - M(q)\ddot{q}_d + M(q)\Lambda\dot{e} - T_d \quad (11)$$

Using equation (3) in (11), the following equation is derived:

$$\dot{s} = \tau - C(q, \dot{q})\dot{q} - G(q) - M(q)\ddot{q}_d + M(q)\Lambda\dot{e} - T_d \quad (12)$$

Substituting equation (12) into the relation between \dot{q} , \ddot{q}_d and s from equation (8), the following equation is obtained:

$$M\dot{s} = \tau - C(q, \dot{q})\dot{q}_d - G(q) - M(q)\ddot{q}_d + M(q)\Lambda\dot{e} - T_d \quad (13)$$

When the state trajectory approaches the sliding surface, $s = 0$. This leads to the equation:

$$\tau - C(q, \dot{q})\dot{q}_d - G(q) - M(q)\ddot{q}_d + M(q)\Lambda\dot{e} - T_d = 0 \quad (14)$$

Assuming no external interference τ_d , the modified control input τ is formulated as follows:

$$\tau = \tau_{SMC} + \alpha \sin(s) + As \quad (15)$$

In equation (15), the term As is introduced to expedite the approach of the state towards the switching manifolds and ensure the convergence of tracking error in finite time. The presence of noise outside the sliding manifold can result in increased tracking errors or inconsistencies between the desired and actual trajectories. This deviation might cause the system to exhibit undesired behaviours, leading to less precise control and potentially impacting overall system performance. The robust terms α and A is tuned to mitigate the effects of uncertainties by ensuring the trajectory remains within the sliding manifold. When the trajectory is within the sliding manifold, noise affects can be reduced (Hušek, 2016).

By substituting equation (14) into (15), the following equation derived:

$$C(q, \dot{q})\dot{q}_r + G(q) + M(q)(\ddot{q}_d - \Lambda\dot{e}) + T_d = \tau_{SMC} + \alpha \sin(s) + As \quad (16)$$

Consequently, the control signal for SMC is given as follows:

$$\tau_{SMC} = M(q)\ddot{q}_r + C(q, \dot{q})\dot{q}_r + G(q) + T_d - \alpha \sin(s) - As \quad (17)$$

where $\alpha = \text{diag}(\alpha_1, \dots, \alpha_n)$ is a positive diagonal definite matrix, bounded as per property 2. Similarly, $\sin(s) = [\sin(s_1), \dots, \sin(s_n)]^T$ and $A = \text{diag}(a_1, \dots, a_n)$ denotes a positive matrix where $s^T ks > 0 \forall s \neq 0$.

3.2 Computed Torque Control (CTC)

Computed torque control (CTC) is a well-established motion control approach ensuring global asymptotic stability for space manipulators. However, due to challenges in acquiring precise dynamical models in practical scenarios, this study innovatively integrates CTC with a neural network sliding mode controller, aiming to enhance joint motion regulation. This novel hybrid approach addresses the limitations of conventional CTC by leveraging neural networks for improved performance. The integration of neural

network (NN) based SMC with CTC stems from several reasons; CTC alone cannot accurately handle chattering arising in the system, NN excels in learning complex patterns and non-linear relationships, allowing CTC to adapt to varying conditions.

From equation (2) and (3), the expression for \ddot{q} is given as follows:

$$\ddot{q} = M(q)^{-1}(\tau - h(q, \dot{q})) \quad (18)$$

Upon substituting \ddot{q} from equation (18) into (6), the equation takes the following form:

$$\ddot{e} = \ddot{q}_d - M(q)^{-1}(\tau - h(q, \dot{q})) \quad (19)$$

The formulation of the control input function is established as follows (Lewis, Abdallah, and Dawson, 2018):

$$u = \ddot{e} = \ddot{q}_d - M(q)^{-1}(\tau - h(q, \dot{q})) \quad (20)$$

Derived from equation (20), the computed joint torque is expressed through the following equation:

$$\tau = M(q)(\ddot{q}_d - u) + h(q, \dot{q})\dot{q} \quad (21)$$

The control signal denoted as " u " embodies proportional-derivative feedback, and its formulation is given as follows:

$$u = -K_d\dot{e} - K_p e \quad (22)$$

By substituting equation (22) into equation (21), the following formulation for the computed torque control law is derived:

$$\tau_{CTC} = M(q)(\ddot{q}_d + K_d\dot{e} + K_p e) + h(q, \dot{q})\dot{q} \quad (23)$$

where the relationship between derivative gain (K_d) and proportional gain (K_p) is defined as $K_d = 2\sqrt{K_p}$ and $K_p = K_d^2/4$. Based on a trial-and-error method, the controllers' gains are initially tuned to achieve the desired control performance. However, neural network learns from the initial tuning process and, continuously adjust the gains to respond to external disturbances. Then these gains are automatically tuned to improve the response of the control system to the changing conditions and maintain stability, which indirectly contributes to reducing the tracking errors by minimising the oscillations and overshoots.

3.3 Neural Network

Intelligent control methods such as fuzzy control and neural networks have the capability to approximate nonlinear systems, making them suitable for adaptive

applications (Fei and Ding, 2012). Neural networks, in particular, are well-suited for adaptive tasks due to their capacity to learn and approximate a wide range of nonlinear functions. While conventional control techniques often rely on a deep understanding of a process's dynamics and its operational context, neural networks shine in nonlinear control domains such as space manipulator control. However, their effectiveness can be compromised when dealing with poorly modelled and unspecified processes or environments. The neural network architecture selected for this research is shown in Fig. 3. It consists of input layer, hidden layer and output layer. There are 10 neurons in the hidden layer and 2 neurons each in the input and output layers based on the number of inputs and outputs. The inputs to the neural network x_{i1} and x_{i2} , the output of the neural network τ_{NN} are derived in section 3.4. Different weights (\bar{W}_i and W_i) are added to each layer and they are also defined in the section 3.4.

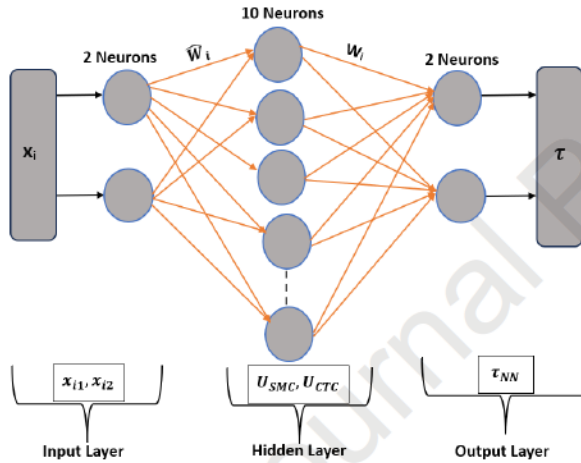


Fig. 3. Neural Network Architecture for the Proposed Controller

The novelty of the developed neural network approach lies in its utilization to improve robustness and controller performance. Specifically, the substitution of a switching function with a constant slew rate ($\alpha' = \alpha * A$) and constant amplitude (A), by an equivalent activation function (sine), addresses the "chattering" phenomenon. This enhancement method distinguishes and outperforms the proposed neural network approach from existing works, contributing significantly to the adaptability and stability of control in uncertain environments.

3.4 Design of NSMCTC

To address concerns about convergence and computational complexity, the proposed neural network approach incorporates a multi-layer architecture with adaptive weights. Initially, these weights adjust the sliding surface's slope, the gains associated with sliding mode control (SMC), and the control input, derived from

error and its rate of change. The inputs of the first input layer are given as follows:

$$x_{i1} = [e_i \ \dot{e}_i \ \ddot{e}_i]^T \quad (24)$$

$$\ddot{e}_i = \ddot{q}^* - \ddot{q} \quad (25)$$

$$\ddot{q}^* = \ddot{q} + K_{Nd}\dot{e} + K_{Np}e \quad (26)$$

The updated values of K_{Np} and K_{Nd} will be derived in the cost function section. The first input layer's output is expressed as:

$$y_i = \Lambda_1 e_i + \Lambda_2 \dot{e}_i = \Lambda_i \cdot x_i \quad (27)$$

Here, Λ_1 and Λ_2 denote the respective gains of e_i and \dot{e}_i . This modified sliding surface is computed as:

$$s_i = f_i(y_i) \quad (28)$$

where f_i represents the activation function of the respective neuron in the neural network.

The first hidden layer's input is formulated as:

$$x_{i2} = [s_{i1} \ s_{i2} \ K_{Np}K_{Nd}]^T \quad (29)$$

With $s_{i1} = s_i$, $s_{i2} = \sin(s_i)$, while K_{Np} and K_{Nd} serve as offset signals compensating the control input signal. The first hidden layer's output for SMC is defined as:

$$U_1 = -\Lambda_N s_{i1} - \Lambda_N K_{Np} \quad (30)$$

$$U_2 = -\Lambda_N s_{i2} - \Lambda_N K_{Nd} \quad (31)$$

$$U_{SMC} = [U_1 \ U_2] \quad (32)$$

The CTC relies on inverse dynamic control to determine the manipulator's real joint acceleration vector using the given torque, τ . This approach results in the system's dynamics being derived as:

$$\ddot{q} = \ddot{q}_d + K_d \dot{e} + K_p e \quad (33)$$

By substituting the expression for \ddot{q}_d from equation (33) into equation (6), the following is obtained:

$$\ddot{e} = \ddot{q} - \ddot{q}_d - K_d \dot{e} - K_p e \quad (34)$$

$$\ddot{e} + K_d \dot{e} + K_p e = 0 \quad (35)$$

Equation (35) signifies that in the absence of external disturbances, the control input is zero. When the proportional gain K_p and derivative gain K_d are suitably chosen, the error asymptotically approaches zero. This equation serves as the control system's error in the NSMCTC approach. This derived control input from equation (35) is reformulated as follows (Jung and Hsia, 1995):

$$d(t) = \ddot{q}_d + K_d(\dot{q}_d - \dot{q}) + K_p(q_d - q) \quad (36)$$

The presence of equation (36) makes CTC method invalid, leading to difficulties in achieving convergence and stability of the manipulator. Consequently, the control aim of this study is to establish an SMC-based control law to act as a compensator for CTC, thereby enabling the manipulator's joint motions, as described by equation (2), to align with desired trajectories. Equation (36)'s right-hand side is represented by SMC, adhering to the fundamental notion of SMC acting as the inverse of the manipulator under CTC. This arrangement ensures that the manipulator's motion closely follows the desired path while achieving minimal distortion and optimal control response.

By combining equations (2) and (36), the following equation is derived:

$$\ddot{q}_d + K_d(\dot{q}_d - \dot{q}) + K_p(q_d - q) = \hat{M}^{-1}(\Delta M \ddot{q} + \Delta h + T_d) \quad (37)$$

To analyse the overall performance of the NSMCTC controller, ideal outputs are represented as $E_d = q_d$, $\dot{E}_d = \dot{q}_d$ and $\ddot{E}_d = \ddot{q}_d$. By defining the tracking error, equation (37) is transformed into:

$$\ddot{q}_d + K_d(\dot{q}_d - \dot{q}) + K_p(q_d - q) = \delta \quad (38)$$

where $\hat{M}^{-1}(\Delta M \ddot{q} + \Delta h + T_d) = \delta$, signifying the collective contribution of NN outputs to the tracking error in equation (38). NSMCTC outputs are incorporated into desired trajectories, resulting $q_N = E_d + q_d$, $\dot{q}_N = \dot{E}_d + \dot{q}_d$ and $\ddot{q}_N = \ddot{E}_d + \ddot{q}_d$. Substituting this relation into equation (37), the following equation is derived:

$$\ddot{e} + K_d \dot{e} + K_p e = \delta \quad (39)$$

Upon controller convergence $e = 0$, ideal outputs satisfy the relationship:

$$\delta = 0 \quad (40)$$

From equation (40), it is clear that neural network must counteract the manipulator model's uncertainties.

The hidden layer's output for CTC is expressed as follows:

$$U_{CTC} = \sum_{n=1}^N x_i \hat{W} \quad (41)$$

An adaptation rule for the weighting factor is based on Lyapunov analyses:

$$\hat{W} = \alpha(k_1 e + k_2 \dot{e}) \quad (42)$$

k_1 and k_2 are the learning parameters. The hidden layer's weight is automatically adjusted to minimize the cost function:

$$W_i = \frac{1}{2}(q_{id} - q_i)^2 \quad (43)$$

The overall neural network output is given by:

$$\tau_{NN} = U_{SMC} * U_{CTC} * W_i \quad (44)$$

The absence of the signum function in the control input torque from the neural network eliminates chattering in the controllers.

For simulation purposes, the NSMCTC's overall inputs are $[e \ \dot{e} \ \ddot{e} \ U_{CTC}]$, where $\ddot{e} = \delta - \ddot{q}$. The neural network outputs, denoted as $[W_i * \text{input of NN}]$, incorporate the adjusted weight factor (W_i) crucial for minimising the cost function. Fig. 2 provides the visual representation of this integration. The neural network's outputs are seamlessly integrated into SMC, contributing to the NSMCTC control inputs. This integration strategically aims to counteract manipulator model uncertainties and eliminate chattering. The combined control output provided to the space manipulator for joint motion control is derived as follows:

$$\tau_{NSMC} = \tau_{SMC} + \tau_{NN} \quad (45)$$

$$\tau_O = \tau_{CTC} + \tau_{NSMC} \quad (46)$$

where τ_{CTC} represents the torque generated by CTC, and τ_{SMC} is the compensating torque produced by NSMCTC.

3.5 Cost Function

For a neural network, the cost function represents the accumulation of errors across each layer. Individual errors are computed at each layer and summed to determine the total error. The role of the cost function is to quantify how effectively the models are performing. Its purpose is to evaluate the overall error resulting from predictions. The objective is to minimize the cost function, as the smaller value of the cost function indicates a closer match between predicted and actual values (Meel, 2023).

In the study by (Ertugrul and Kaynak, 1997), the controller gains of Computed Torque Control (CTC) are treated as the weights of the neural network. These gains are automatically updated by the neural network to minimize the cost function. The adjusted weights are calculated as demonstrated in equation (43).

The updated gains are obtained using the following equations:

$$K_{pi}(t+1) = K_{pi}(t) + \eta_{K_p} \Delta K_{pi}(t) \quad (47)$$

$$K_{di}(t+1) = K_{di}(t) + \eta_{K_d} \Delta K_{di}(t) \quad (48)$$

where η_{K_p} and η_{K_d} are learning rates for the proportional and derivative gains, and $\Delta K_{pi}(t)$ and $\Delta K_{di}(t)$ represent changes in the gains calculated using a steepest descent method:

$$\Delta K_{pi}(t) = \frac{\partial W_i}{\partial K_{pi}} = \frac{\partial W_i}{\partial q_i} \cdot \frac{\partial q_i}{\partial \tau_i} \cdot \frac{\partial \tau_i}{\partial u_i} \cdot \frac{\partial u_i}{\partial K_{pi}} = -e_i \frac{\Delta q_i}{\Delta \tau_i} (-s_i) \quad (49)$$

$$\Delta K_{di}(t) = \frac{\partial W_i}{\partial K_{di}} = \frac{\partial W_i}{\partial q_i} \cdot \frac{\partial q_i}{\partial \tau_i} \cdot \frac{\partial \tau_i}{\partial u_i} \cdot \frac{\partial u_i}{\partial K_{di}} = -e_i \frac{\Delta q_i}{\Delta \tau_i} (-\sin(s_i)) \quad (50)$$

These equations involve the sliding surface parameters. Thus, equations (47) and (48) are redefined as:

$$K_{Npi}(t+1) = K_{pi}(t) + \eta_{K_p} e_i \frac{\Delta q_i}{\Delta \tau_i} s_i \quad (51)$$

$$K_{Ndi}(t+1) = K_{di}(t) + \eta_{K_d} e_i \dot{e}_i \frac{\Delta q_i}{\Delta \tau_i} \sin(s_i) \quad (52)$$

Similarly, changes in the slope of the sliding surface are calculated using:

$$\Lambda_{N1}(t+1) = \Lambda_1(t) + \eta_{\Lambda_1} \Delta \Lambda_{i1}(t) \quad (53)$$

$$\Lambda_{N2}(t+1) = \Lambda_2(t) + \eta_{\Lambda_2} \Delta \Lambda_{i2}(t) \quad (54)$$

$$\begin{aligned} \Delta \Lambda_1(t) &= \frac{\partial W_i}{\partial \Lambda_1} = \frac{\partial W_i}{\partial q_i} \cdot \frac{\partial q_i}{\partial \tau_i} \cdot \frac{\partial \tau_i}{\partial u_i} \cdot \frac{\partial s_i}{\partial \Lambda_1} \\ &= -e_i \frac{\Delta q_i}{\Delta \tau_i} \left(-K_{pi} - K_{di} \frac{\partial(\sin(s_i))}{\partial s_{pi}} \right) f'_{i1} e_i \end{aligned} \quad (55)$$

For convenience, consider $\frac{\partial q_i}{\partial \tau_i} \approx \frac{\Delta q_i}{\Delta \tau} = 1$

$$\frac{\partial W_i}{\partial \Lambda_1} = -e_i \frac{\Delta q_i}{\Delta \tau_i} \left(-K_{pi} - K_{di} \frac{4e^{-2s_i}}{(1+e^{-2s_i})^2} \right) e_i \quad (56)$$

$$\begin{aligned} \Delta \Lambda_2(t) &= \frac{\partial W_i}{\partial \Lambda_2} = \frac{\partial W_i}{\partial q_i} \cdot \frac{\partial q_i}{\partial \tau_i} \cdot \frac{\partial \tau_i}{\partial u_i} \cdot \frac{\partial s_i}{\partial \Lambda_2} \cdot \frac{\partial y_i}{\partial \Lambda_2} \\ &= -e_i \frac{\Delta q_i}{\Delta \tau_i} \left(-K_{pi} - K_{di} \frac{\partial(\sin(s_i))}{\partial s_{pi}} \right) f'_{i1} e_i \end{aligned} \quad (57)$$

$$\frac{\partial W_i}{\partial \Lambda_1} = -e_i \frac{\Delta q_i}{\Delta \tau_i} \left(-K_{pi} - K_{di} \frac{4e^{-2s_i}}{(1+e^{-2s_i})^2} \right) \dot{e}_i \quad (58)$$

These changes in slope are used to adjust the sliding surface parameters:

$$\Lambda_{N1}(t+1) = \Lambda_1(t) - \eta_{\Lambda_1} e_i^2 \frac{\Delta q_i}{\Delta \tau_i} \left(-K_{pi} - K_{di} \frac{4e^{-2s_i}}{(1+e^{-2s_i})^2} \right) \quad (59)$$

$$\Lambda_{N2}(t+1) = \Lambda_2(t) - \eta_{\Lambda_2} e_i \dot{e}_i \frac{\Delta q_i}{\Delta \tau_i} \left(-K_{pi} - K_{di} \frac{4e^{-2s_i}}{(1+e^{-2s_i})^2} \right) \quad (60)$$

$$\Lambda_N = \frac{\Lambda_{N1}(t+1)}{\Lambda_{N2}(t+1)} = \Lambda_{i-eqv} \quad (61)$$

Equation (61) represents the reaching law that accelerates convergence. The sliding surface slope is adjusted to increase the equivalent slope, promoting optimal convergence where errors are minimized.

3.6 Stability Analysis

Regarding stability analysis, the study examines the stability of SMC along with the neural network as a compensator for CTC. In the presence of uncertainties and disturbances, the system (joints) deviates from the sliding manifold. SMC intentionally modifies its structure using discontinuous control to steer the phase trajectory to a stable sliding surface. To ensure the manipulator's trajectory stays close to the sliding manifold, a Lyapunov function is introduced, which could demonstrate that the tracking error eventually converges to a neighbourhood of zero with its time derivative being negative. The Lyapunov function is represented as follows (Hui Hu and Peng-Yung Woo, 2006):

$$V = \frac{1}{2} s^T M s \quad (62)$$

where M is a symmetric and positive definite matrix, and V is a positive scalar function of the vector s and time t . The derivative of V is expressed as:

$$\dot{V} = s^T M \dot{s} \quad (63)$$

Substituting equation (10) into equation (63), the following is obtained:

$$\dot{V} = s^T [\tau - h(q, \dot{q}) - M(q)(\ddot{q}_d - \Lambda \dot{e}) - T_d] \quad (64)$$

By introducing the relationship $\ddot{q}_r = \ddot{q}_d - \Lambda \dot{e}$ from equation (9), equation (64) becomes:

$$\dot{V} = \frac{1}{2} s^T [\tau - h(q, \dot{q}) - M(q)\ddot{q}_r + T_d] \quad (65)$$

The system's stability is demonstrated if the controller's gains of the switching controller surpass the

upper bound of the uncertainties. By substituting equation (17) into equation (65), the following equation is obtained:

$$\dot{V} = -s^T \alpha \sin(s) - s^T k s + s^T T_d \quad (66)$$

Assuming bounded derivatives, i.e., $\|T_d\|_\infty \leq T_{d\text{bound}}$, the system's stability is ensured, and the derivative of the Lyapunov function is negative:

$$\begin{aligned} \dot{V} &\leq -s^T \alpha \sin(s) - s^T k s + s^T T_d \leq 0 \\ \rightarrow s^T \alpha \sin(s) + s^T k s &\geq s^T T_{d\text{bound}} \end{aligned} \quad (67)$$

where $T_{d\text{bound}} = f(\dot{x}) = f_1 \dot{x} + f_2 \sin(\dot{x})$ and, f_1 and f_2 are the positive constant matrices.

4. Results and Discussion

In this section, the controllers outlined in equations (17), (23), and (44) are assessed using MATLAB-Simulink to demonstrate the performance of the proposed approach. The sinusoidal desired or reference trajectories are defined as follows and then tested within the MATLAB Simulink environment:

$q_{1d} = \frac{55}{2} + \frac{55}{2} \sin\left(\frac{2\pi t}{10}\right) - \left(\frac{\pi}{2}\right)$ and $q_{2d} = \frac{85}{2} + \frac{85}{2} \sin\left(\frac{2\pi t}{10}\right) - \left(\frac{\pi}{2}\right)$. These trajectories represent the desired positions of the system over time in the subsequent plots. To assess the controllers' performance, the trajectories are tested within the MATLAB-Simulink environment. In the desired trajectory equations, the sine function is introduced to represent the unmodeled uncertainties, allowing for the investigation of the robustness of the combined controller. The outcomes obtained for each controller, SMC, CTC and NSMCTC, are displayed in Fig. 4, Fig. 5, and Fig. 6. The presented results illustrate that the controllers accurately track the provided trajectory. Fig. 5 particularly reveals that the proposed CTC has achieved convergence. However, Fig. 6 exhibits chattering, which stems from the uncertainties and noise inherent in the system.

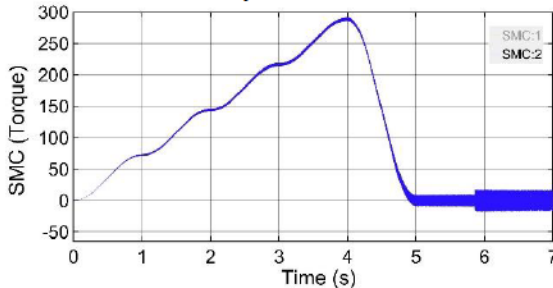


Fig. 4. Sliding Mode Control's Output Torque



Fig. 5. Computed Torque Control's Output Torque

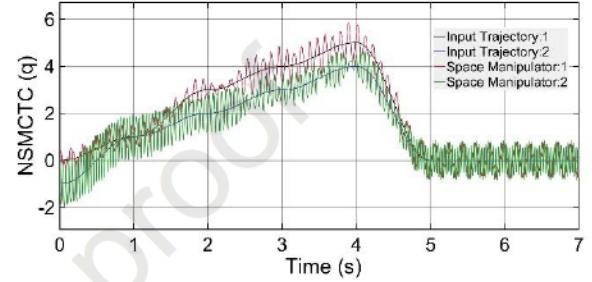


Fig. 6. Tracking Performance with Chattering

The Levenberg Marquardt (LM) algorithm is a widely used optimisation technique for training neural networks (Mystkowski et al., 2023). LM is known for its robustness, efficiency and ability to handle a wide range of optimisation problems. In this research, the objective of LM is to minimise the error between the network's predictions and the desired position provided in the training data. To address the issue of chattering and reduce the magnitude of error, the following parameters are fine tuned in the neural network:

- SMC
 - Slope of the sliding surface: $\Lambda_i = [100 \ 0.05]$ for $i = 1, 2$.
 - Robust terms: $A = [12.5, 15]$ and $\alpha = [0.05 \ 0.05]$.
 - Offset signal compensating the control input: $K_{i0} = [0.001, 0.001]$.
 - Disturbance: $T_d = [0.002, 0.2]$.
- CTC
 - Proportional and derivative gains: $K_p = 12.5$ and $K_d = 12.5$.
- NSMCTC
 - Learning rates: $k_1 = k_2 = [100, 100]$.
 - Learning rates for minimizing the cost function (explained in the next section): $\eta_{K_{pi}} = \eta_{K_{vi}} = \eta_{\Lambda_{i1}} = \eta_{\Lambda_{i2}} = [0.7 \ 0.7]$.

Neural network-based tuning allows for dynamical optimization of control parameters, by enabling the controller to adapt to changes in the system or environment, reducing the impact of uncertainties on system performance. With these carefully adjusted parameters, the enhanced tracking performance is

illustrated in Fig. 7. The outcomes clearly showcase that the manipulated space system attains a high level of control precision in terms of tracking, effectively eliminating chattering and reducing the impact of noise.

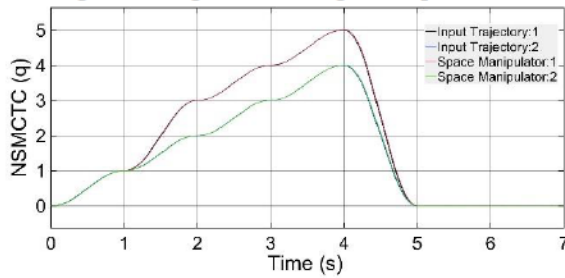


Fig. 7. Tracking Performance of NSMCTC Without Chattering

The overall control performance based on the position and velocity is demonstrated in Fig. 8 and Fig. 9. The desired trajectory has a good convergence rate of 4.999 seconds and has good responsiveness to the system.

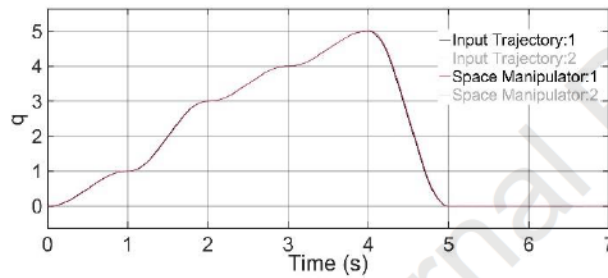


Fig. 8. The Performance Evaluation of Position Control q

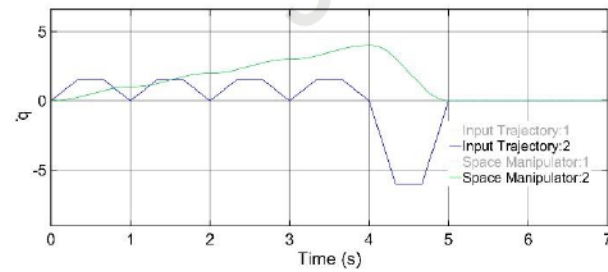


Fig. 9. The Performance Evaluation of Velocity Control \dot{q}

The error performance of NSMCTC is demonstrated in Fig. 10. The errors are significantly small and are converged to zero with a convergence rate of 4.994 s.

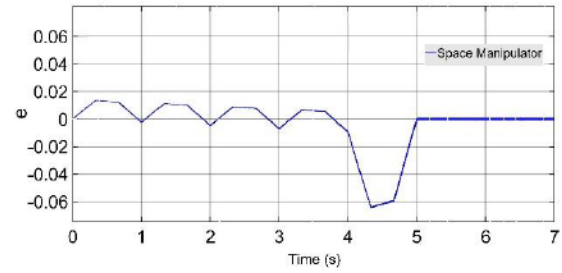


Fig. 10. The Performance evaluation of Tracking Error e .

The stability of the proposed controller is tested in the MATLAB-Simulink environment. The Lyapunov derivative \dot{V} is -7.957×10^{-6} , ensuring that the controller has achieved stability.

NSMCTC is also tested with polynomial trajectory in Simulink environment using the polynomial reference trajectory and the results are demonstrated in Fig. 11 and 12. The tracking performance is similar with that of the sinusoidal reference, as demonstrated in Fig. 7 and Fig. 11. However, the tracking error is slightly larger than that of the sinusoidal reference, as show in Fig. 10 and Fig. 12. The errors are converged at 5.002s and 4.994s for the polynomial and sinusoidal trajectory, respectively.

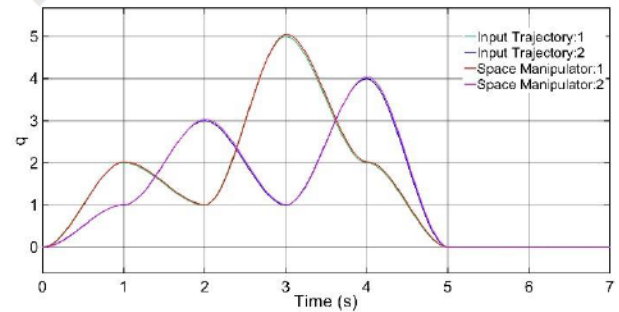


Fig. 11 Tracking Performance of NSMCTC with a polynomial input Trajectory

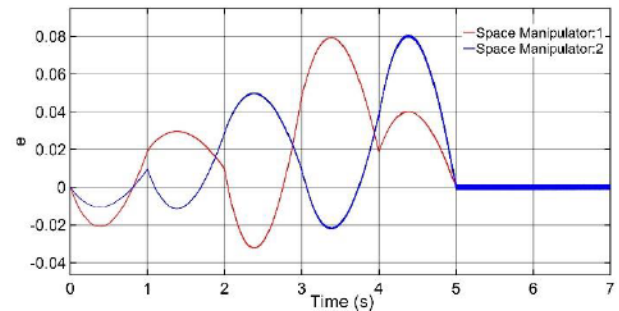


Fig. 12 Error Performance of NSMCTC with a polynomial input Trajectory

However, the controller accurately tracks the desired trajectory and has significantly smaller error in polynomial trajectory too. This indicates the efficiency of NSMCTC's control action generation and feedback correction mechanisms. The neural network automatically tunes the controller's parameters to achieve similar performance. Additionally, NSMCTC's adaptability allows it to adjust control actions based on specific trajectory characteristics, ensuring effective performance. The results demonstrate NSMCTC's robustness, adaptability, and generalization capability of accurately and predictably controlling of the space manipulator, which indicates its suitability for real-world applications.

4.1 Comparison

This study presents a comparison between the control outcomes of the traditional CTC and our proposed CTC, as shown in Fig. 13. In control systems dealing with uncertainties and assumed variables, the performance of the traditional CTC is notably degraded, exhibiting a reduction in control effectiveness when faced with unknown variables. However, the novel controller employs adaptive principles from Equation (42), allowing it to achieve the anticipated control performance even when variables are uncertain. The controller gains are adjusted to attain the desired control performance. Equation (37) provides a mathematical representation of how the control effectiveness of traditional CTC is anticipated to decrease when dealing with uncertainties. The comparison indicates that the proposed CTC surpasses the traditional CTC concerning convergence, accuracy in tracking, stability, and responsiveness. Fig. 14 and 15 visually demonstrate the enhanced convergence speed of the proposed CTC compared to the conventional CTC. Unlike the conventional approach, the proposed CTC maintains a steady state post-convergence. These outcomes were achieved through the self-learning capabilities of the neural network, confirming its effectiveness in enhancing the control performance of the space manipulator. Unlike traditional position-based computer torque control schemes, NSMCTC demonstrates superior control outcomes without imposing a heavy computational burden. This combination of effectiveness and computational efficiency positions NSMCTC as a promising approach for advanced control applications in space manipulators.

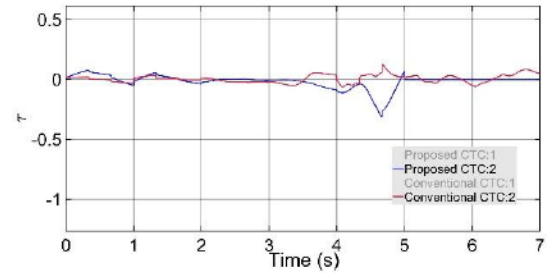


Fig. 13. The Performance Evaluation of Torque τ

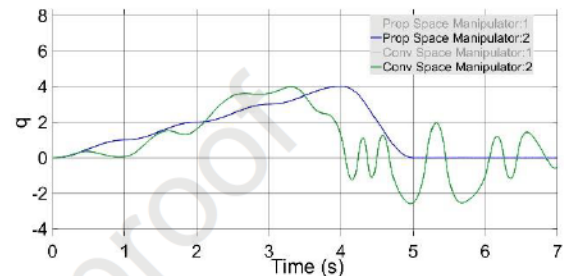


Fig. 14. The Performance Evaluation of Position Control q

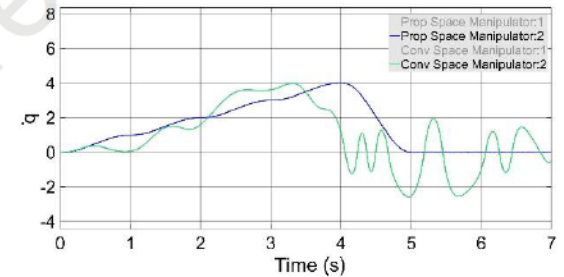


Fig. 15. The Performance Evaluation of Velocity Control \dot{q}

The performance of tracking error against the time is demonstrated in Fig. 16. The results show that the tracking error of the proposed CTC consistently converges to zero across all time intervals, confirming the attainment of global asymptotic stability. The behaviour of the proposed CTC closely resembles that of an ideal controller, as evidenced by the near absence of tracking error.

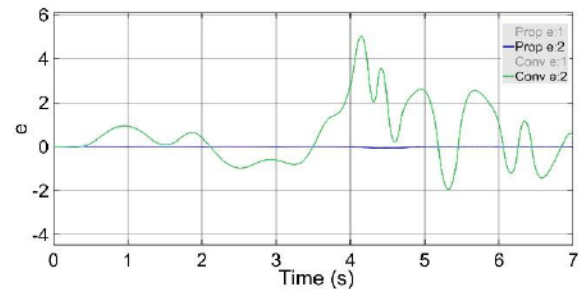


Fig. 16. The Performance Evaluation of Tracking Error e

In Fig. 17, a slight overshoot in the tracking error rate is observed due to the substantial learning rate, which aids rapid convergence. Notably, the NSMCTC controller achieves both zero tracking error convergence and enhanced responsiveness, illustrated in Fig. 16 and Fig. 17.

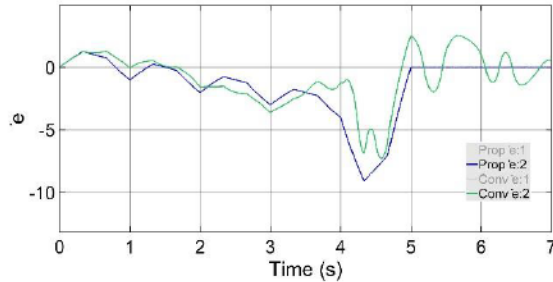


Fig. 17. The Performance Evaluation of Rate of Tracking Error \dot{e}

5. Conclusion

This study developed neural network based sliding mode computed torque control approach for robust position tracking of space manipulators. The primary goal was to address uncertainties stemming from disturbances and unmodelled dynamics. By combining sliding mode control, computed torque control, and neural networks, NSMCTC demonstrated enhanced performance in terms of precision in trajectory tracking, reduced chattering and rapid convergence compared to conventional methods. The Lyapunov stability theorem validated its uniform and ultimate boundedness and global asymptotic stability.

Extensive MATLAB-Simulink simulations verified the effectiveness of NSMCTC in accurately tracking space manipulator positions, achieving precise joint motion control and demonstrating improved responsiveness. The approach's key advantage lies in its self-learning capability, enabling adaptation to the system's dynamic model without prior knowledge. Additionally, NSMCTC offers simplicity and low computational complexity, setting it apart from traditional position-based computed torque control schemes.

This research sheds light on the NSMCTC's potential application in spacecraft manipulator attitude control, particularly for multidimensional dynamics and safe object capture with collision avoidance. By incorporating image recognition systems, the controller could enhance the capture of uncooperative objects, further expanding its utility in space missions.

Acknowledgements

I would like to express my sincere gratitude to Professor Massimiliano Vasile, the Director of the Aerospace Centre of Excellence, for his invaluable support and encouragement throughout the course of this

research. I am also immensely grateful to the International Astronautical Federation (IAF) and British Interplanetary Society (BIS) for awarding me the prize money to attend the International Astronautical Congress (IAC) 2023 conference. This opportunity provided me with valuable exposure to the latest advancements in the field and enriched my research experience. Last but certainly not least, thanks to the International Strategic Partner (ISP) of University of Strathclyde for their generous funding that supports my PhD studies.

References

- [1] Behera, L., Gopal, M. and Chaudhury, S. (1996). On adaptive trajectory tracking of a robot manipulator using inversion of its neural emulator. *IEEE Transactions on Neural Networks*, 7(6), pp.1401–1414. doi:<https://doi.org/10.1109/72.548168>.
- [2] Benke, K.K., Norng, S., Robinson, N.J., Benke, L.R. and Peterson, T.J. (2018). Error propagation in computer models: analytic approaches, advantages, disadvantages and constraints. *Stochastic Environmental Research and Risk Assessment*, 32(10), pp.2971–2985. doi: <https://doi.org/10.1007/s00477-018-1555-8>.
- [3] Shan, M., Guo, J. and Gill, E. (2016). Review and comparison of active space debris capturing and removal methods. *Progress in Aerospace Sciences*, 80, pp.18–32. doi: 10.1016/j.paerosci.2015.11.001.
- [4] Welandar, P. (2010, February 1). Understanding Derivative in PID Control. *Control Engineering*. <https://www.controleng.com/articles/understanding-derivative-in-pid-control/>
- [5] Pathak, P.M., Kumar, R.P., Mukherjee, A. and Dasgupta, A. (2008). A scheme for robust trajectory control of space robots. *Simulation Modelling Practice and Theory*, 16(9), pp.1337–1349. doi: 10.1016/j.simpat.2008.06.011.
- [6] Moosavian, S.A.A. and Papadopoulos, E. (2007). Free-flying robots in space: an overview of dynamics modeling, planning and control. *Robotica*, 25(5), pp.537–547. doi:10.1017/s0263574707003438.
- [7] Ali, S., Moosavian, S.A.A. and Papadopoulos, E. (1997). Coordinated Motion Control of Multiple Manipulator Space Free-Flyers. *core.ac.uk*. Available at: <https://core.ac.uk/display/24496335>.
- [8] Ertugrul, M. and Kaynak, O. (1997a). Neural network adaptive sliding mode control and its application to SCARA type robot manipulator. *IEEE Xplore*. doi: <https://doi.org/10.1109/ROBOT.1997.606732>.
- [9] Jung, S. and Hsia, T.C. (1995). A new neural network control technique for robot manipulators*. *Robotica*, 13(5), pp.477–484. doi: <https://doi.org/10.1017/S0263574700018312>.

- [10] Papadopoulos, E., Aghili, F., Ma, O. and Lampariello, R. (2021). Robotic Manipulation and Capture in Space: A Survey. *Frontiers in Robotics and AI*, 8. doi:10.3389/frobt.2021.686723.
- [11] Rong-Jong Wai and Muthusamy, R. (2013). Fuzzy-Neural-Network Inherited Sliding-Mode Control for Robot Manipulator Including Actuator Dynamics. *IEEE Transactions on Neural Networks and Learning Systems*, 24(2), pp.274–287. doi:10.1109/tnnls.2012.2228230.
- [12] Mystkowski, A., Wolniakowski, A., Kadri, N., Sewiolo, M. and Scalera, L. (2023). Neural Network Learning Algorithms for High-Precision Position Control and Drift Attenuation in Robotic Manipulators. *Applied Sciences*, [online] 13(19), p.10854. doi:https://doi.org/10.3390/app131910854.
- [13] Hui Hu and Peng-Yung Woo (2006). Fuzzy supervisory sliding-mode and neural-network control for robotic manipulators. *IEEE Transactions on Industrial Electronics*, 53(3), pp.929–940. doi: https://doi.org/10.1109/tie.2006.874261.
- [14] Hüllermeier, E. and Waegeman, W. (2021). Aleatoric and epistemic uncertainty in machine learning: an introduction to concepts and methods. *Machine Learning*, 110(3), pp.457–506. doi: https://doi.org/10.1007/s10994-021-05946-3.
- [15] Fei, J. and Ding, H. (2012). Adaptive sliding mode control of dynamic system using RBF neural network. *Nonlinear Dynamics*, 70(2), pp.1563–1573. doi: https://doi.org/10.1007/s11071-012-0556-2.
- [16] Young, K.D., Utkin, V.I. and Ozguner, U. (1999). A control engineer's guide to sliding mode control. *IEEE Transactions on Control Systems Technology*, 7(3), pp.328–342. doi: https://doi.org/10.1109/87.761053.
- [17] Wang, J. and Cui, Y. (2022). Adaptive Neural Tracking Control for a Two-Joint Robotic Manipulator with Unknown Time-Varying Delays. *Complexity*, 2022, pp.1–12. doi:10.1155/2022/7230853.
- [18] Wu, Y., Huang, R., Li, X. and Liu, S. (2019). Adaptive neural network control of uncertain robotic manipulators with external disturbance and time-varying output constraints. *Neurocomputing*, 323, pp.108–116. doi: 10.1016/j.neucom.2018.09.072.
- [19] Sun, W., Su, S.-F., Xia, J. and Nguyen, V.-T. (2019). Adaptive Fuzzy Tracking Control of Flexible-Joint Robots with Full-State Constraints. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 49(11), pp.2201–2209. doi:10.1109/TSMC.2018.2870642.
- [20] Tian, L. and Collins, C. (2005). Adaptive neuro-fuzzy control of a flexible manipulator. *Mechatronics*, 15(10), pp.1305–1320. doi: 10.1016/j.mechatronics.2005.02.001.
- [21] Hao, Z., Shyam, R.B.A., Rathinam, A. and Gao, Y. (2021). Intelligent Spacecraft Visual GNC Architecture with the State-Of-the-Art AI Components for On-Orbit Manipulation. *Frontiers in Robotics and AI*, 8. doi:10.3389/frobt.2021.639327.
- [22] Yoshida, K., Dimitrov, D. and Nakanishi, H. (2006). On the Capture of Tumbling Satellite by a Space Robot. *IEEE Xplore*. doi:10.1109/IROS.2006.281900.
- [23] Liu, Y., Yu, C., Sheng, J. and Zhang, T. (2018). Self-collision Avoidance Trajectory Planning and Robust Control of a Dual-arm Space Robot. *International Journal of Control, Automation and Systems*, 16(6), pp.2896–2905. doi:10.1007/s12555-017-0757-z.
- [24] Wu, Q., Lin, C.-M., Fang, W., Chao, F., Yang, L., Shang, C. and Zhou, C. (2018). Self-Organizing Brain Emotional Learning Controller Network for Intelligent Control System of Mobile Robots. *IEEE Access*, 6, pp.59096–59108. doi:10.1109/access.2018.2874426.
- [25] Kazuya, Y. and Yoji, U. (1990). Control of Space Free-Flying Robot. *Proceedings of the 29th conference on Decision and Control, (CH2917-3/90/000O-00S7\$1.00 @ 1990 IEEE)*. Department of Mechanical Engineering Science Tokyo Institute of Technology 2-12-1, 0-okayama, Meguro, Tokyo 152.
- [26] Song, Z., Yi, J., Zhao, D. and Li, X. (2005). A computed torque controller for uncertain robotic manipulator systems: Fuzzy approach. *Fuzzy Sets and Systems*, 154(2), pp.208–226. doi: https://doi.org/10.1016/j.fss.2005.03.007.
- [27] Sharkawy, A.-N. and Koustoumpardis, P. (2019). Dynamics and Computed-Torque Control of a 2-DOF manipulator: Mathematical Analysis. *International Journal of Advanced Science and Technology*, 28(12), pp.201–212. Available at: https://hal.science/hal-03598924/document.
- [28] Yoshida, K. (1994). Experimental study on the dynamics and control of a space robot with experimental free-floating robot satellite. *Advanced Robotics*, 9(6), pp.583–602. doi:10.1163/156855395x00319.
- [29] Kobayashi, T. and Tsuda, S. (2011). Sliding Mode Control of Space Robot for Unknown Target Capturing. *Engineering letters, EL_19_2_03(issues v19)*.
- [30] Vu, D.H., Huang, S. and Tran, T.D. (2019). A Novel Robust Adaptive Intelligent and Compound Control of an Adaptive Neural Network, SMC, FLC

- and PI for Robot Manipulators. IOP Conference Series: Materials Science and Engineering, 631(2), p.022041. doi:10.1088/1757-899x/631/2/022041.
- [31] Tejomurtula, S. and Kak, S. (1999). Inverse kinematics in robotics using neural networks. Information Sciences, 116(2-4), pp.147–164. doi:10.1016/s0020-0255(98)10098-1.
- [32] Liu, C., Wen, G., Zhao, Z. and Sedaghati, R. (2021). Neural-Network-Based Sliding-Mode Control of an Uncertain Robot Using Dynamic Model Approximated Switching Gain. IEEE Transactions on Cybernetics, 51(5), pp.2339–2346. doi:10.1109/TCYB.2020.2978003.
- [33] Yazdanpanah, M.J. and Ghafari, A. (2003). Modification of sliding mode controller by neural network with application to a flexible link. 2003 European Control Conference (ECC). doi: <https://doi.org/10.23919/ecc.2003.7085224>.
- [34] Zhang, Z., Li, X., Wang, X., Zhou, X., An, J. and Li, Y. (2022). TDE-Based Adaptive Integral Sliding Mode Control of Space Manipulator for Space-Debris Active Removal. Aerospace, 9(2), p.105. doi:<https://doi.org/10.3390/aerospace9020105>.
- [35] Cordero, R., Estrabis, T., Gentil, G., Caramalac, M., Suemitsu, W., Onofre, J., Brito, M. and dos Santos, J. (2022). Tracking and Rejection of Biased Sinusoidal Signals Using Generalized Predictive Controller. *Energies*, [online] 15(15), p.5664. doi: <https://doi.org/10.3390/en15155664>.
- [36] Kim, W., Shin, D., Won, D. and Chung Choo Chung (2013). Disturbance-Observer-Based Position Tracking Controller in the Presence of Biased Sinusoidal Disturbance for Electrohydraulic Actuators. *IEEE Transactions on Control Systems and Technology*, 21(6), pp.2290–2298. doi: <https://doi.org/10.1109/tcst.2013.2237909>.

Intelligent and Robust Control of Space Manipulator for Sustainable Removal of Space Debris

1. Neural network-based sliding mode computed torque control (NSMCTC) approach ensures precise space manipulator position tracking.
2. NSMCTC reduces chattering and improves convergence for enhanced trajectory control in dynamic space environments.
3. Lyapunov stability theorem validates uniform and ultimate boundedness, ensuring global asymptotic stability.
4. Self-learning capability and low computational complexity distinguish NSMCTC from traditional computed torque control methods.
5. Potential applications extend to spacecraft manipulator attitude control, offering adaptability for safe debris capture in multidimensional dynamics.

Journal Pre-proof

Declaration of interests

☒ The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

☐ The author is an Editorial Board Member/Editor-in-Chief/Associate Editor/Guest Editor for [Journal name] and was not involved in the editorial review or the decision to publish this article.

☐ The authors declare the following financial interests/personal relationships which may be considered as potential competing interests:

--