

Article

Nonlinear MPC-Based Deep Neural Network for Chaotic System Synchronization

Junhong Xie ^{1,*}, Alexandra Grancharova ² and Zhaojun Meng ¹

¹ Department of Electrical and Automation Engineering, Liaoning Institute of Science and Technology, Benxi 117004, China; mzj744@163.com

² Department of Industrial Automation, University of Chemical Technology and Metallurgy, 1756 Sofia, Bulgaria; grancharova@uctm.edu

* Correspondence: xiejunhong@lnist.edu.cn

Abstract

The synchronization of chaotic systems is a critical issue in nonlinear control theory, providing a foundation for various practical applications, such as fault detection, image encryption, and secure communication. However, chaotic systems are highly complex nonlinear dynamical systems. Therefore, achieving the required control accuracy of such systems represents a challenging task. In this paper, the nonlinear model predictive control (NMPC) methodology, combined with deep neural networks (DNNs), is applied for synchronization between master and slave chaotic systems. Such an approach aims to address the synchronization control problem under constraints, with an improved control performance and a minimal computational burden. An explicit NMPC controller is designed that is based on the idea to build offline a neural network approximation of the predictive controller. Then, online, the computation of control input consists of simple function evaluation as realized by the neural network. The performance of the NMPC-DNN is studied for third-order chaotic systems. The results show that the proposed control method leads to smaller synchronization error and stability of the master–slave chaotic systems.

Keywords: synchronization control; master–slave chaotic systems; deep learning; model predictive control; neural networks

1. Introduction

The discovery of the Lorenz chaos system revealed intricate patterns within seemingly random behavior [1]. As researchers delve deeper into nonlinear chaotic systems, the practice of achieving consistent behavior among the trajectories of two or more systems through control methods is termed synchronous control of chaotic systems [2]. Synchronous control of nonlinear chaotic systems provides the fundamental underpinning for a wide spectrum of practical applications, such as securing digital information through secure communications, enhancing diagnostic capabilities in fault detection, and ensuring data protection in image encryption [3–5]. This reveals the significance of synchronization in chaotic systems for practical applications.

However, the inherent characteristics of chaotic systems bring challenges for the design of synchronous controllers. Conventional control methodologies encounter several obstacles in practical implementations. Dynamic systems are typically characterized by nonlinearity and uncertainty, which inhibit the attainment of satisfactory control performance. Moreover, the constraints associated with system variables lead to increased complexity



Academic Editors: Cecilio Angulo, Mohammad Jafari and Mahmut Reyhanoglu

Received: 10 August 2026

Revised: 15 September 2026

Accepted: 23 September 2026

Published: 28 September 2026

Copyright: © 2026 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 \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

in control tasks. Therefore, conventional control methods struggle to achieve the required precision when synchronizing highly complex chaotic systems.

To overcome these challenges, intelligent control technology primarily addresses the control problems of nonlinear complex systems, which pose significant challenges to classical control methodologies [6–9]. Specifically, these approaches offer superior handling of highly nonlinear systems, effectively managing system complexities while enabling precise control of multivariable systems subject to constraints [10]. In addition, intelligent control systems possess both learning and optimization capabilities [11]. The learning capability allows the system to acquire knowledge from external information, thereby continuously enhancing its control performance.

In recent years, neural network-based intelligent control has attracted significant attention [12,13]. Neural networks were proposed in the 1940s [14]. They are technologies widely used in nonlinear system modelling and control [15,16]. The advantages of neural networks not only lie in their powerful learning ability but also their continuous nonlinear function approximation ability. This data-driven approach makes them highly suitable for the control of highly complex nonlinear chaotic systems. In [17], an adaptive quantum neural networks (QNN)-based controller was proposed for chaos synchronization, employing Lyapunov-based adaptive laws to estimate system uncertainties and achieve high-precision synchronization, with successful applications demonstrated in secure communication and cryptography. The QNN architecture leverages quantum superposition and entanglement properties, enabling more efficient information processing for complex chaotic systems. Parallel to the QNN, brain-inspired emotional learning controllers have attracted considerable attention for chaos synchronization applications by mimicking the interaction between the emotional network and the sensory network in the human brain. The brain emotional learning controller (BELC) provides adaptive weighting of sensory information based on stimulus significance, a mechanism particularly effective for robust learning in uncertain environments. Recent studies have extended this architecture to fuzzy brain emotional learning neural networks (FBELNNs), integrating fuzzy inference systems with emotional learning to achieve robust synchronization under external disturbances and parameter uncertainties [18,19]. An interval type-2 fuzzy brain emotional learning controller was developed for hyper-chaos synchronization, demonstrating superior tracking performance and computational efficiency compared to RBF neural networks and BP neural networks in [20]. Furthermore, hybrid approaches combining recurrent cerebellar model articulation controllers (RCMACs) with fuzzy BELCs have been successfully applied to audio and image secure communications.

In [21], a novel memristive synapse with multiple stable states and local activity was studied. A multi-area neural network was built based on this synapse, demonstrating complex chaotic behaviors and self-reinforcing attractor dynamics. Control mechanisms are implemented to achieve synchronized activity across the simulated brain regions. In [22], the authors introduced an RBF-DNN enhanced by Bayesian regularization to model a chaotic system. Model validation was performed through multiple performance, confirming its accuracy and robustness. In [23], a robust adaptive controller combining chaotic fuzzy logic and type-2 neural networks was developed for seismic response mitigation in structures, using multilayer perceptron estimation and Extended Kalman filter training. In [24], two high-dimensional chaotic systems were synchronized via a brain-inspired adaptive fuzzy neural controller, producing dynamic encryption keys in real time to protect transmitted signals. Moreover, common neural network models include CNN [25], RNN [26], long short-term memory network [27], DNN [28], and RBF neural network models [29].

Meanwhile, model predictive control (MPC) has garnered widespread attention due to its ability to handle multivariable systems, adapt to constraints, and optimize future control

performance [30,31]. In addition, achieving robust synchronization and dynamic consensus has become a hot topic in complex multi-agent architectures and network systems [32–34]. In this paper, a control approach is developed for the synchronization control of chaotic systems based on deep neural networks by using nonlinear MPC. The core objective of this work is to pursue desired control performance while minimizing the associated online computational burden. This intelligent control method is not recommended to replace any existing methods but rather serves as a supplementary method for synchronizing master–slave chaotic systems.

Although strategies for approximating explicit NMPC controllers using neural networks have been explored in the field of nonlinear control [35–40], their application to the synchronization of chaotic systems has not yet been fully investigated. In this paper, the method is developed specifically for the synchronization problem of master–slave chaotic systems. It should be noted that there exists a number of neural network-based explicit MPC methods [35–40], but they do not explicitly address the dynamics of synchronization errors between master–slave chaotic systems. Our NMPC-DNN framework is applied to a synchronization error model and achieved high synchronization control performance. For existing learning-based synchronization methods, such as QNN [17], FBELNN [18–20], and RBF-DNN [22], the focus is primarily on adaptive or brain-inspired control strategies, without explicit consideration of constraints or predictive optimization. Therefore, the main innovation of this study lies in the new synchronization results obtained for master–slave chaotic systems under constraints and external disturbances. This study offers the following innovations. First, it establishes a new synchronization performance benchmark for constrained chaotic systems using NMPC-DNN. Second, it provides a complete design pipeline from error dynamics to DNN training that ensures constraint satisfaction. Third, it offers stability analysis and quantitative comparisons that show the superiority of the proposed method over existing synchronization control approaches. Additionally, the proposed method reduces the computational burden compared to the NMPC based on online optimization.

This paper is structured as follows: Section 2 describes the problem for master and slave systems. Section 3 details the NMPC-DNN controller for the synchronization of systems. Section 4 presents the simulation results.

2. Description of Master–Slave Chaotic Systems

2.1. General Description

A generic representation of the master chaotic system takes the following form [41]:

$$\dot{z}_i = g_{m_i}(z(t)), i = 1, 2, \dots, n \quad (1)$$

where $z = [z_1, z_2, \dots, z_n] \in \mathbb{R}^n$ is the master state vector, and $g_{m_i} : \mathbb{R}^n \rightarrow \mathbb{R}$, $i = 1, 2, \dots, n$ are nonlinear continuous functions [31]. It is supposed that the corresponding slave system has the following dynamics [31]:

$$\dot{r}_i = g_{s_i}(r(t)) + \mu_i + d_i(t), i = 1, 2, \dots, n \quad (2)$$

where $r = [r_1, r_2, \dots, r_n] \in \mathbb{R}^n$ is the slave state vector, $g_{s_i} : \mathbb{R}^n \rightarrow \mathbb{R}$, $i = 1, 2, \dots, n$ are nonlinear continuous functions, $\mu_i \in \mathbb{R}$, $i = 1, 2, \dots, n$ are the control inputs, and the disturbances are denoted by $d_i(t) \in \mathbb{R}$, $i = 1, 2, \dots, n$ [41]. The synchronization error between systems (1) and (2) is as follows:

$$\varepsilon_i = r_i - z_i, i = 1, 2, \dots, n \quad (3)$$

By substituting (1) and (2) into (3) and taking the derivative, we obtain the following error dynamics [41]:

$$\dot{\varepsilon}_i = g_{s_i}(r(t)) - g_{m_i}(z(t)) + \mu_i + d_i(t), \quad i = 1, 2, \dots, n \quad (4)$$

Ideally, master–slave system synchronization is accomplished when the error trajectory $\varepsilon_i \rightarrow 0$ asymptotically.

2.2. Nonlinear Third-Order Chaotic System

The synchronization of nonlinear third-order master and slave chaotic systems is considered. The master system is characterized according to the following [41]:

$$\begin{cases} \dot{z}_1 = 1 - z_2 z_3 \\ \dot{z}_2 = -a z_3 + b z_2 \\ \dot{z}_3 = z_1 z_2 \end{cases} \quad (5)$$

The resulting space trajectory exhibits the complex chaotic attractor, as depicted in Figure 1.

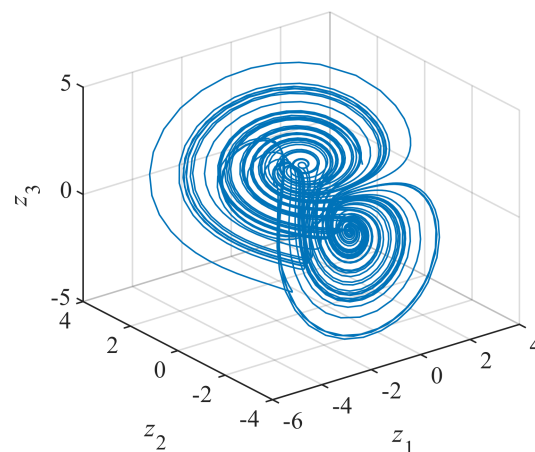


Figure 1. Trajectory of system (5).

The slave system dynamics are as follows [41]:

$$\begin{cases} \dot{r}_1 = 1 - r_2 r_3 + \mu_1 + d_1 \\ \dot{r}_2 = -c_1 r_3 + c_2 r_2 + \mu_2 + d_2 \\ \dot{r}_3 = r_1 r_2 + \mu_3 + d_3 \end{cases} \quad (6)$$

3. Nonlinear MPC-Based Deep Neural Network for Chaotic System Synchronization

3.1. NMPC Problem Formulation

Nonlinear Model Predictive Control (NMPC) has emerged as a powerful and flexible control strategy that is well suited to handling the complexities of nonlinear systems. NMPC is a form of control in which an optimization problem is solved at each time step to determine the control inputs that minimize a given cost function over a finite prediction horizon [30]. The drawback of NMPC is the computational complexity. The computational requirements for solving nonlinear optimization problems online are high. In contrast, Neural Networks (NNs) have demonstrated the ability to learn from data.

Here, an approach to design an explicit NMPC for synchronization between master and slave chaotic systems is applied that is based on the idea to build offline a neural

network controller approximation. Then, online, the computation of control input consists of simple function evaluation realized by the neural network. Explicit NMPC controllers have a simple software and hardware implementation, which makes them very useful in embedded control applications.

Based on the specific third-order master system (5) and slave system (6), the model (4) is rewritten:

$$\dot{\varepsilon}(t) = f(\zeta(t)) + \mu(t) + d(t) \quad (7)$$

The nonlinear term $f(\zeta(t)) = g_s(z(t) + \varepsilon(t)) - g_m(z(t))$ captures the synchronization error $\varepsilon(t)$, where $\zeta(t) = [\varepsilon^T(t), z^T(t)]^T$ is defined as an augmented state vector. $d(t)$ denotes bounded external perturbations, and t is the continuous-time. The evolution of synchronization error dynamics inherently depends on the master system, as the nonlinear expansion of $g_s(z(t) + \varepsilon(t))$ generates cross-coupling terms. Consequently, the master state trajectory continuously modulates the operating point of the error dynamics. Applying the forward Euler method with a sampling time of T_s , the model (7) is discretized to obtain the following:

$$\varepsilon(k+1) = f_d(\zeta(k)) + \Phi\mu(k) + d(k) \quad (8)$$

The forward Euler method is adopted to mitigate the computational complexity during the repeated evaluations of the NMPC optimization problem. As shown in Section 4.2, this discretization algorithm reduces the online computational complexity compared to the Runge–Kutta integration method. It is important to note that this compact representation is specifically valid for the particular chaotic systems defined by Equations (5) and (6). In (8), the sampled nonlinear map $f_d(\zeta(k)) = \varepsilon(k) + T_s f(\zeta(k))$ results from a Euler step applied to the continuous error equation, $\Phi = \text{diag}\{T_s, T_s \dots T_s\}$ is the input-gain matrix, and k denotes the discrete time. The following constraint is imposed on the control input:

$$\mu_{\min} \leq \mu(k) \leq \mu_{\max}, \quad \forall k \quad (9)$$

where $\mu_{\min}$ and $\mu_{\max}$ are the lower and upper bounds on $\mu(k)$.

The receding-horizon optimal control task for system (8) is to drive the synchronization error ε to the origin while satisfying the actuator limitation in (9). It is supposed that at each sampling instant k , the finite-horizon NMPC problem is solved:

Problem 1.

$$V^*(\varepsilon(k)) = \min_U J(U, \varepsilon(k)) \quad (10)$$

subject to $\varepsilon_{k|k} = \varepsilon(k)$, $z_{k|k} = z(k)$ and

$$\begin{aligned} \varepsilon_{k+i+1|k} &= f_d(\varepsilon_{k+i|k}, z_{k+i|k}) + \Phi\mu_{k+i|k}, \quad i = 0, 1, \dots, N_p - 1 \\ z_{k+i+1|k} &= z_{k+i|k} + T_s g_m(z_{k+i|k}), \quad i = 0, 1, \dots, N_p - 1 \end{aligned} \quad (11)$$

$$\mu_{\min} \leq \mu_{k+i} \leq \mu_{\max}, \quad i = 0, 1, \dots, N_p - 1 \quad (12)$$

Equation (11) is the nominal prediction model of the synchronization error dynamics and master system states over the prediction horizon N_p . In (10), $U = [\mu_k, \mu_{k+1}, \dots, \mu_{k+N_p-1}]$ is the control input sequence that is to be optimized. The cost function $J(U, \varepsilon(k))$ penalizes the synchronization error accumulation and the excessive control effort and is formulated as

$$J(U, \varepsilon(k)) = \sum_{i=0}^{N_p-1} [\|\varepsilon_{k+i|k}\|_Q^2 + \|\mu_{k+i|k}\|_R^2] \quad (13)$$

where $l(\varepsilon_{k+i|k}, \mu_{k+i|k}) = \|\varepsilon_{k+i|k}\|_Q^2 + \|\mu_{k+i|k}\|_R^2$ is the stage cost function. The weighting matrices $Q \succ 0$ and $R \succ 0$ are assumed to be symmetric positive definite matrices. Let $U^* = [\mu_k^*, \mu_{k+1}^*, \dots, \mu_{k+N_p-1}^*]$ denote the optimal solution to Problem 1. According to the receding horizon control strategy, the control input applied to the system at time k is given by $\mu(k) = \mu_k^*$ [30]. It is worth noting that the trajectory of the master system $z(k)$ is strictly bounded within the compact invariant set of the chaotic attractor $\Omega_z \subset \mathbb{R}^3$ (i.e., $\|z(k)\| \leq \bar{z}$). The compactness and invariance of Ω_z guarantee that all relevant stability properties hold uniformly with respect to $z(k) \in \Omega_z$.

3.2. NMPC Design by Deep Neural Networks

Here, the NMPC for synchronization of master and slave chaotic systems is designed by using deep neural networks, referred to as NMPC-DNN. For the purpose of the NMPC design, the following compact nominal model (without disturbances) is considered, which describes the dynamic behavior of an augmented state, comprising the synchronization error and master system state in the third-order master–slave system from Section 3.1:

$$\xi(k+1) = \varphi(\xi(k), \mu(k)) \quad (14)$$

where $\xi(t) = [\varepsilon^T(t), z^T(t)]^T$ is defined as an augmented state vector. $\varphi(\xi(k), \mu(k)) = [f_d(\xi(k)) + \Phi\mu(k)]^T, g_m^T(z(k))]^T$ with $g_m(z(k)) = z(k) + T_s g_m(z(k))$ denoting the discrete-time dynamics of the master system. The evolution of synchronization error dynamics inherently depends on the master system. The master state trajectory continuously modulates the operating point of the error dynamics. Consequently, $z(k)$ is considered as a bounded time-varying parameter vector. The augmented state vector $\xi(k) = [\varepsilon^T(k), z^T(k)]^T \in \mathbb{R}^{n_\xi}$ and the control input vector $\mu(k) \in \mathbb{R}^{n_\mu}$ represent the input and the output of the network, respectively. The DNN employed in the NMPC framework defines a functional mapping, with the form as $\mathcal{N} : \mathbb{R}^{n_\xi} \rightarrow \mathbb{R}^{n_\mu}$. The DNN architecture is as follows [42]:

$$\mathcal{N}(\xi; \theta, M, L) = \begin{cases} \omega_{L+1} \circ \varsigma_L \circ \omega_L \circ \dots \circ \varsigma_1 \circ \omega_1(\xi), & L \geq 2 \\ \omega_{L+1} \circ \varsigma_1 \circ \omega_1(\xi), & L = 1 \end{cases} \quad (15)$$

Each activation is expressed as the composition $\varsigma_l \circ \omega_l(\cdot) = \varsigma_l(\omega_l(\cdot))$, where M denotes the per-layer neuron count (network width) and L denotes the total number of hidden layers (network depth) [42]. The affine activation function for the l -th layer is

$$\omega_l(\zeta_{l-1}) = W_l \zeta_{l-1} + b_l \quad (16)$$

Here, $\zeta_{l-1} \in \mathbb{R}^M$ is the output with the $(l-1)$ layer, W_l represents the weights matrix, and b_l denotes biases [42]. The activation chosen for g_l is ReLU:

$$\varsigma_l(\omega_l) = \max(0, \omega_l) \quad (17)$$

The parameters of the neural network are denoted as $\theta = \{\theta_1, \dots, \theta_{L+1}\}, l = 1, \dots, L+1$. Here, the training data for the NMPC-DNN are generated by solving Problem 1 for a set of N_s initial augmented states $\xi_i = [\varepsilon_i^T, z_i^T]^T, i = 1, 2, \dots, N_s$ and obtaining the optimal control sequences $\mu^*(\xi_i), i = 1, \dots, N_s$. Thus, the objective of training a predefined network is to obtain the optimal parameter set θ^* that minimizes the loss function.

$$\theta^* = \arg \min_{\theta} \frac{1}{N_s} \sum_{i=1}^{N_s} \|\mu^*(\xi_i) - \mathcal{N}(\xi_i; \theta)\|^2 \quad (18)$$

where the optimal control input and network output are $\mu^*(\xi_i)$ and $\mathcal{N}(\xi_i; \theta)$, respectively. The resulting neural network control law is $\mu(\xi(k)) = \text{sat}(\mathcal{N}(\xi(k); \theta^*))$.

The block-diagrams of the relation between the DNN and the NMPC are shown in Figure 2. Thus, Figure 2a illustrates the offline training of the DNN by using the optimal trajectories obtained by solving the NMPC problem. Figure 2b shows the closed-loop system where the NMPC-DNN is the already trained controller.

To achieve the master–slave system synchronization, the NMPC-DNN strategy is employed. The synchronization error model is obtained from Equations (3) and (4) using the master system (5) and slave system (6). Additionally, the Problem 1 is then solved for the error states by using the CasADi toolbox [43,44]. Then, obtaining the optimal control inputs μ_k^* enables the generation of the optimal control pairs (ξ_k, μ_k^*) for neural network training.

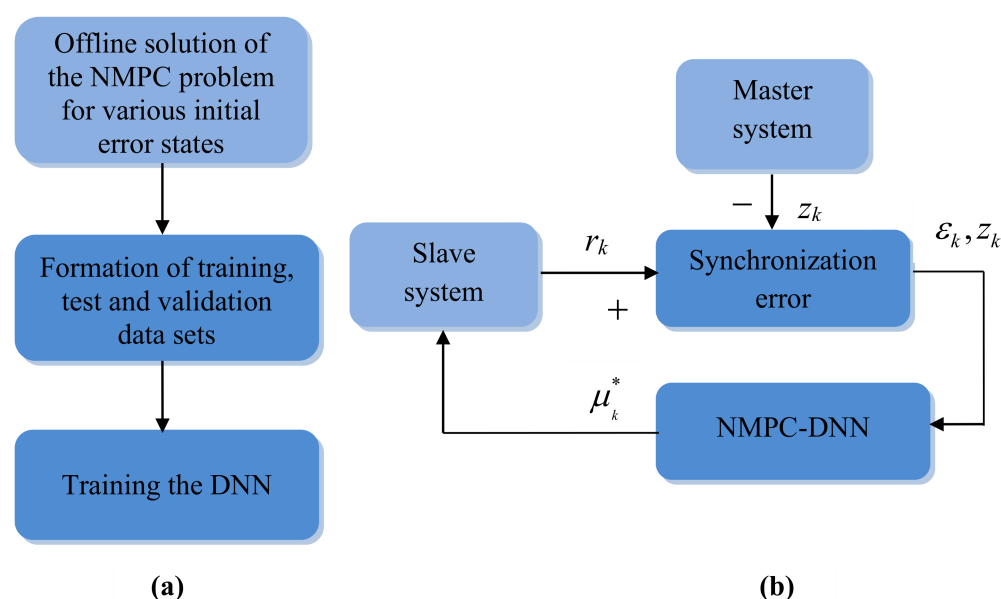


Figure 2. Illustration of the NMPC-DNN approach: (a) offline training of the DNN with the solution of the NMPC problem, (b) closed-loop control system with the trained NMPC-DNN controller.

3.3. Stability Analysis of the NMPC-DNN Scheme

The stability of the proposed NMPC-DNN control scheme is analyzed with respect to the synchronization error $\varepsilon(k)$.

Let the discrete-time dynamics of the closed-loop system be described by the following:

$$\begin{cases} \varepsilon(k+1) = F_\varepsilon(\varepsilon(k), z(k)) + \Phi\mu(\xi(k)) + d(k) \\ z(k+1) = z(k) + T_s g_m(z(k)) \end{cases} \quad (19)$$

where $F_\varepsilon(\cdot)$ represents discrete-time mapping of the synchronization error dynamics. The trajectory of the master chaotic system $z(k)$ is strictly bounded within a compact invariant set $\Omega_z \subset \mathbb{R}^3$, i.e., $\|z(k)\| \leq \bar{z}$ for all $k \geq 0$. It serves exclusively as a bounded time-varying parameter that modulates the error dynamics through the nonlinear coupling term in $F_\varepsilon(\cdot)$. The synchronization objective is to drive the error $\varepsilon(k)$ to a neighborhood of the origin, while $z(k)$ remains its bounded autonomous motion. The control input applied to the system is generated by the trained neural network control law $\mu(\xi(k)) = \text{sat}(\mathcal{N}(\xi(k); \theta^*))$, where $\xi(k) = [\varepsilon^T(k), z^T(k)]^T$ and $z(k)$ is treated as a measurable parameter vector. The

nominal optimal NMPC control law is denoted as $\mu^*(\xi(k))$. We define the neural network approximation error as $\delta(k)$.

$$\delta(k) = \text{sat}(\mathcal{N}(\xi(k); \theta^*)) - \mu^*(\xi(k)) \quad (20)$$

To analyze the robustness of the system, we introduce the Practical Input-to-State Stability (pISS) framework [45]. We establish the following standard assumptions for the nominal NMPC: first, we consider the stability of the nominal closed-loop system:

$$\begin{cases} \tilde{\varepsilon}(k+1) = F_{\varepsilon}(\tilde{\varepsilon}(k), z(k)) + \Phi \mu^*(\tilde{\xi}(k)) \\ z(k+1) = z(k) + T_s g_m(z(k)) \end{cases} \quad (21)$$

where $\tilde{\varepsilon}(k)$ is the synchronization error in the absence of disturbance, and $\tilde{\xi}(k) = [\tilde{\varepsilon}^T(k), z^T(k)]^T$. It is known that there are two groups of methods for guaranteeing the stability of the closed-loop systems with MPC controllers. The first types of approaches are based on using terminal constraint and/or terminal cost function in the MPC problem formulation [46,47]. The second group of methods is based on the relaxed dynamic programming concept [48,49], where no terminal constraint/cost is used. Here, we adopt the latter concept, since it is easily applied to tracking problems.

Let, at the current time k , the synchronization error be $\tilde{\varepsilon}(k)$ and the optimal control input be $\mu^*(\tilde{\xi}(k))$. In [49], stability and a certain degree of suboptimality is proved if the relaxed dynamic programming inequality (adapted to our problem setup)

$$V^*(\tilde{\varepsilon}(k)) \geq V^*(F_{\varepsilon}(\tilde{\varepsilon}(k), z(k)) + \Phi \mu^*(\tilde{\xi}(k))) + \beta l(\tilde{\varepsilon}(k), \mu^*(\tilde{\xi}(k))) \quad (22)$$

holds for some $\beta \in (0, 1)$ and for all time steps in the closed loop trajectory of the nominal system (21). Here, $V^*(F_{\varepsilon}(\tilde{\varepsilon}(k), z(k)) + \Phi \mu^*(\tilde{\xi}(k))) = V^*(\tilde{\varepsilon}(k+1))$, i.e., it is the optimal cost function value computed at the successor error state. Thus, we use the following assumption:

Assumption 1. Assume that for a pre-specified value of suboptimality degree $\beta \in (0, 1)$, a control horizon N_p is known such that Equation (22) holds.

Remark 1. The notation $V^*(\tilde{\varepsilon}(k))$ of the optimal cost function is adopted for brevity. Since the autonomous master trajectory $z(k)$ is restricted to the compact invariant set Ω_z , its one-step-ahead state remains bounded with respect to $z(k) \in \Omega_z$. Therefore, Assumption 1 serves as a foundation for establishing the robustness of the subsequent NMPC-DNN scheme.

Further, we consider the stability of the disturbed closed-loop system (19) with the NMPC law approximated by the DNN, i.e., $\mu(\xi(k)) = \text{sat}(\mathcal{N}(\xi(k); \theta^*))$. The following assumption is made.

Assumption 2. The optimal cost function $V^*(\varepsilon(k))$ is locally Lipschitz continuous. There exist Lipschitz constants $C_{\delta} > 0$ and $C_d > 0$ such that for any bounded approximation error $\delta(k)$ and bounded disturbance $d(k)$,

$$\|V^*(\varepsilon(k+1)) - V^*(\tilde{\varepsilon}(k+1))\| \leq C_{\delta} \|\delta(k)\| + C_d \|d(k)\| \quad (23)$$

where $\tilde{\varepsilon}(k+1)$ denotes the trajectory under the nominal optimal control law. Also, the stage cost function satisfies $l(\varepsilon(k), \mu^*(\xi(k))) \geq \alpha \lambda_{\min}(Q) \|\varepsilon(k)\|^2$, where $\alpha > 0$ and $\lambda_{\min}(Q) > 0$ is the minimum eigenvalue of Q .

Assumption 3. Through the offline DNN training process, where the loss function is minimized, the DNN approximation error is bounded over the operating domain. That is, there exists a small positive constant $\bar{\delta}$ such that $\|\delta(k)\| \leq \bar{\delta}$. The external disturbance is also bounded as $\|d(k)\| \leq \bar{d}$.

Theorem 1. Under Assumptions 1–3, the closed-loop system (19) controlled by the proposed NMPC-DNN control law $\mu(\xi(k)) = \text{sat}(\mathcal{N}(\xi(k); \theta^*))$ is Practical Input-to-State Stable (pISS) with respect to the disturbance $d(k)$ and approximation error $\delta(k)$.

Proof. By evaluating the difference of the Lyapunov function along the actual closed-loop trajectory with the DNN control law and disturbance, we obtain

$$\Delta V^* = V^*(\varepsilon(k+1)) - V^*(\varepsilon(k)) \quad (24)$$

Adding and subtracting the nominal trajectory term $V^*(\tilde{\varepsilon}(k+1))$ yields

$$\begin{aligned} \Delta V^* &= V^*(\varepsilon(k+1)) - V^*(\tilde{\varepsilon}(k+1)) \\ &\quad + V^*(\tilde{\varepsilon}(k+1)) - V^*(\varepsilon(k)) \end{aligned} \quad (25)$$

Applying Assumptions 1 and 2, we have

$$\Delta V^* \leq -\beta\alpha\lambda_{\min}(Q)\|\varepsilon(k)\|^2 + C_\delta\|\delta(k)\| + C_d\|d(k)\| \quad (26)$$

Substituting the uniform bounds from Assumption 3, this yields

$$\Delta V^* \leq -\beta\alpha\lambda_{\min}(Q)\|\varepsilon(k)\|^2 + C_\delta\bar{\delta} + C_d\bar{d} \quad (27)$$

Equation (27) shows that the Lyapunov function is strictly decreasing outside the compact invariant set $\Omega = \{\varepsilon \in \mathbb{R}^3 : \beta\alpha\lambda_{\min}(Q)\|\varepsilon(k)\|^2 \leq C_\delta\bar{\delta} + C_d\bar{d}\}$. The ultimate bound of the synchronization error trajectory satisfies $\limsup_{k \rightarrow \infty} \|\varepsilon(k)\| \leq \sqrt{(C_\delta\bar{\delta} + C_d\bar{d}) / \beta\alpha\lambda_{\min}(Q)}$. Thus, the closed-loop system trajectory is ultimately bounded within a region determined by the DNN approximation error bound $\bar{\delta}$ and the disturbance upper bound $\bar{d}$. The master system state $z(k)$ remains bounded within the chaotic attractor Ω_z due to its own autonomous dynamics.

In addition, to guarantee that the control input strictly satisfies constraints $\mu_{\min} \leq \mu(k) \leq \mu_{\max}$, a saturation layer is applied to the output of the DNN $\mu(k) = \text{sat}(\mathcal{N}(\xi(k); \theta^*)) = \min(\max(\mathcal{N}(\xi(k); \theta^*), \mu_{\min}), \mu_{\max})$. This mechanism ensures zero constraint violation regardless of the magnitude of the approximation error $\delta(k)$. The optimal control sequences in the offline dataset were generated by solving Problem 1, where the constraint $\mu_{\min} \leq \mu(k) \leq \mu_{\max}$ is strictly enforced. Thus, the DNN learns an empirical mapping that resides within the feasible domain. The proof is complete. $\square$

This stability analysis shows that as long as the DNN is trained sufficiently well, the master–slave chaotic systems will maintain high-precision synchronization. It is fully corroborated by the trajectories demonstrated in the numerical simulations in Section 4.

4. Numerical Results

Here, the performance of an NMPC-DNN designed for the nonlinear third-order chaotic system in Section 3.2 is evaluated and compared with other control methods.

4.1. Data for the Design of the NMPC-DNN

Example 1 In the numerical simulation, the parameter values of system (5) are $a = 1$, $b = 1$, and the initial condition is $z_1(0) = 1$, $z_2(0) = -0.5$, $z_3(0) = 1.5$. The initial condition of system (6) is $r_1(0) = -1$, $r_2(0) = 1.5$, $r_3(0) = 1$ and parameter values are $c_1 = 1$, $c_2 = 1$ [41]. Synchronization error dynamics are analyzed subject to disturbance $d_i = 0.5\sin(t)$, $i = 1, 2, 3$. Following the initial values set for the chaotic systems, the synchronization error at its initial state is identified as $\varepsilon_1(0) = -2$, $\varepsilon_2(0) = 2$, $\varepsilon_3(0) = -0.5$. In order to discretize the model (7), the forward Euler method is used with a sampling time of $T_s = 0.01$.

In the NMPC Problem 1, the prediction horizon is $N_p = 10$ and the weighting matrices are $Q = \begin{bmatrix} 200 & 0 & 0 \\ 0 & 200 & 0 \\ 0 & 0 & 200 \end{bmatrix}$ and $R = \begin{bmatrix} 0.01 & 0 & 0 \\ 0 & 0.01 & 0 \\ 0 & 0 & 0.01 \end{bmatrix}$. In constraints in Equations (9) and (12), the control input is restricted by the lower bound $\mu_{\min} = -60$ and the upper bound $\mu_{\max} = 60$. The DNN has depth of $L = 4$ hidden layers with $M = 3$ neurons per layer. The training dataset consists of 600 samples generated through closed-loop simulations of NMPC based on online optimization. The data collection is randomly sampled from the hyper-rectangular domain $\xi(k) = [\varepsilon^T(k), z^T(k)]^T$, where $\varepsilon(k) = [-3, 1] \times [-2, 3] \times [-1.5, 0.5]$ and $z(k) = [-4, 4] \times [-4, 5] \times [-4, 4]$. This domain is chosen to encompass the specific initial error condition used in numerical experiments while also covering a sufficiently broad region of the initial error state space to ensure the trained network generalizes well. Both the input states and output control sequences of the trained DNN are normalized using Min–Max scaling to improve convergence. The dataset is randomly divided into training 70%, validation 15%, and testing 15% subsets. The network is trained using the Levenberg-Marquardt backpropagation algorithm with full batch gradient descent. The training and validation errors are stabilized for 137 epochs, and the initial damping factor is configured to 0.001. The final Mean Squared Error for the training and validation sets converged to approximately 3.6×10^{-7} and 4.0×10^{-7} , respectively, indicating that the network accurately approximates the explicit NMPC control law without overfitting.

Example 2 The other set of initial values is used for the master–slave systems to verify robustness and disturbance for the NMPC-DNN strategy: parameter values of the system (5) $a = 1.5$, $b = 2.1$, and initial conditions $z_1(0) = 2$, $z_2(0) = -1$, $z_3(0) = 2$. The initial value of Equation (6) is $r_1(0) = -1.5$, $r_2(0) = 0.5$, $r_3(0) = -1$, and parameter values are $c_1 = c_2 = 1$. Meanwhile, synchronization error dynamics are analyzed subject to disturbance $d_i = 0.8\cos(t) + 0.3\sin(2t)$, $i = 1, 2, 3$. The synchronization error at its initial state is identified as $\varepsilon_1(0) = -3.5$, $\varepsilon_2(0) = 1.5$, $\varepsilon_3(0) = -3$. In the NMPC Problem 1, the prediction horizon is $N_p = 15$. The DNN has depth $L = 10$ hidden layers with $M = 6$ neurons per layer. The training dataset consists of 600 samples generated through closed-loop simulations. The data collection is randomly sampled from the hyper-rectangular domain $\varepsilon(k) = [-5, 1] \times [-2, 3] \times [-4, 2]$ and $z(k) = [-2, 4] \times [-6, 3] \times [-2, 4]$. This domain is chosen to encompass the specific initial error condition used in numerical experiments, while also covering a sufficiently broad region of the initial error state space to ensure the trained network generalizes well. The DNN is trained in 23 epochs, and the initial damping factor is configured to 0.001. The final Mean Squared Error for the training and validation sets converged to approximately 3.38×10^{-10} and 4.0×10^{-10} , respectively.

4.2. Performance of the NMPC-DNN Approach and Comparison with Other Control Methods

Here, the performance of the NMPC-DNN is studied and compared with that of the novel exponentially perturbed sliding mode neural network (NEPSM-NN) approach in [41].

The key of the NEPSM-NN (novel exponentially perturbed sliding mode neural network) approach was to design an exponentially convergent disturbance observer, to achieve high-precision synchronization of master–slave chaotic systems combining RBF-NN with sliding mode control (SMC) [41]. First, an exponential disturbance observer was designed. This allows the controller to effectively compensate for disturbances within the sliding mode control law, reducing chattering. Second, an RBF-NN was used to approximate the estimator function generated by the continuous function of the master–slave system. Finally, a new exponential perturbation sliding mode neural network was constructed, whose performance was proven to guarantee the stability of the master–slave system synchronization [41]. The proposed NEPSM-NN strategy is described in detail in [41].

Example 1 The control input trajectories obtained with the NMPC-DNN are shown in Figures 3 and 4a (the blue dashed curves). In addition, the synchronization error trajectories are displayed in Figure 4b. The Mean Squared Error of DNN is shown in Figure 5. In Figures 3 and 4, the red dashed lines represent the lower bound $\mu_{\min}$ and the upper bound $\mu_{\max}$ in the control input constraints, respectively. It can be observed that, despite aggressive transient control actions at the beginning of the synchronization process, the proposed NMPC-DNN controller strictly adheres to the boundary conditions, thereby successfully demonstrating its constraint-handling capability. At this point, both Equations (9) and (12) are satisfied.

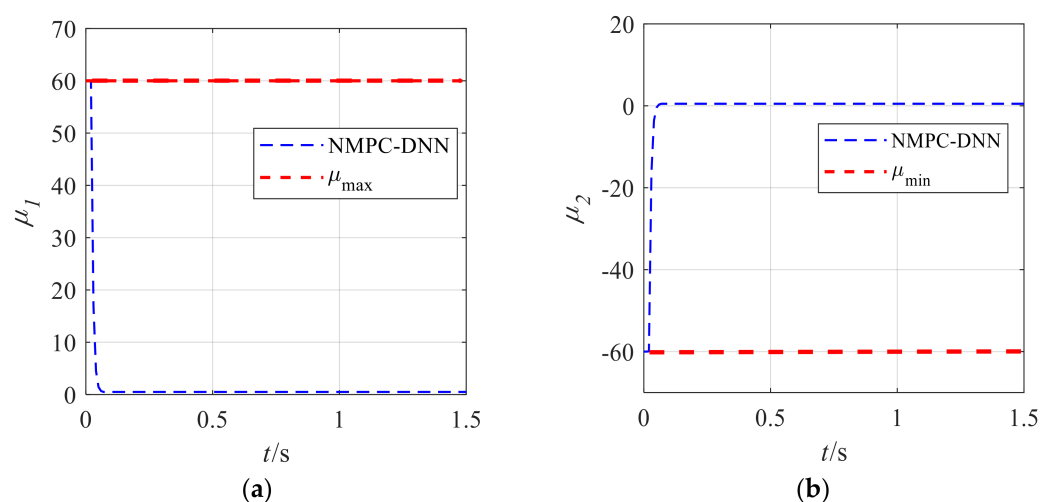


Figure 3. Trajectory of (a) control input μ_1 ; (b) control input μ_2 .

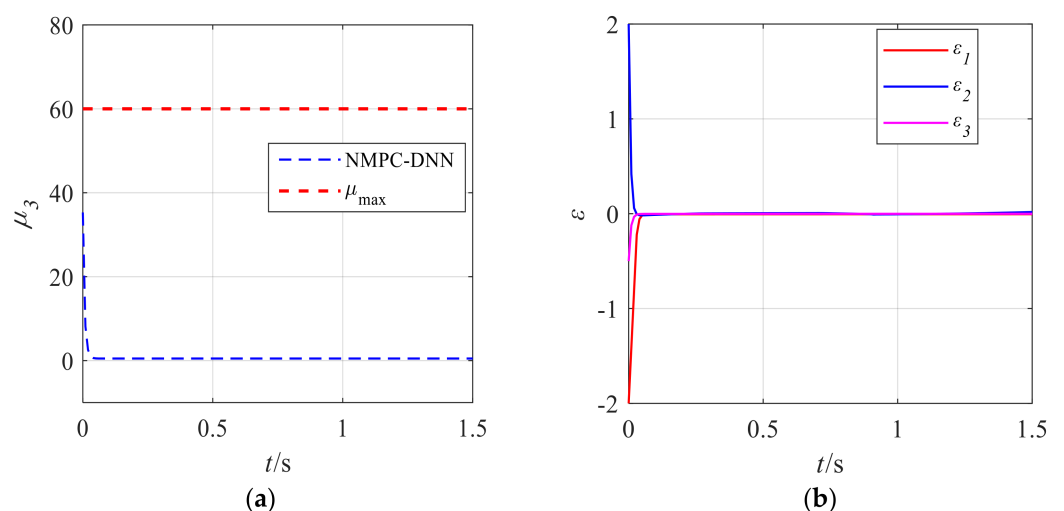


Figure 4. Trajectory of (a) control input μ_3 ; (b) synchronization errors ε_1 , ε_2 and ε_3 .

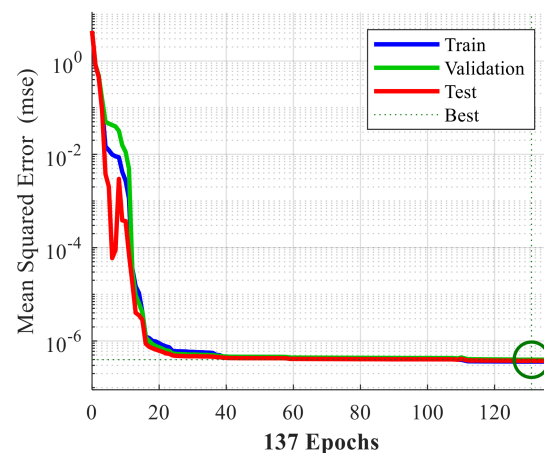


Figure 5. The Mean Squared Error of DNN.

Here, the use of the forward Euler method to discretize the continuous-time error dynamics in Equation (7) is justified, as well as the choice of the sampling time. The sampling time $T_s = 0.01$ s is chosen. It is sufficiently small to capture the fast transient dynamics of the third-order chaotic system, keeping the local truncation error of the Euler prediction model bounded over the finite horizon. A larger T_s would lead to constraint violations and predictive instability, whereas a much smaller T_s would unnecessarily increase the prediction horizon N_p , escalating the computational burden.

We implemented the fourth order Runge–Kutta (RK4) integration algorithm in our formulation to compare the control performance. The results are summarized in Table 1.

Table 1. Performance comparison of NMPC-DNN under different discretization algorithms.

Discretization Algorithm	Average CPU Time (s)	Example 1		
		ISE (ϵ_1)	ISE (ϵ_2)	ISE (ϵ_3)
Forward Euler	0.00023 s	0.049	0.051	0.002
Runge–Kutta4 (RK4)	0.00028 s	0.053	0.058	0.002

As shown in Table 1, the difference between the integral squared error obtained by using the RK4 discretization method and the forward Euler method is less than 0.1. Therefore, the NMPC-DNN performance using these two methods is similar in terms of the ISE indicator. However, the forward Euler discretization algorithm provides a lower average CPU time. This proves that the discretization method we used is not only reasonable but also reduces the computational burden to a certain extent.

It is worth noting that the RK4 method significantly increases the computational burden during the optimization process, primarily due to the increased depth of the computational graph. The time required for memory access and matrix operations increases significantly when the nonlinear programming solver computes gradients and search directions in each iteration.

Example 2 The control input trajectories obtained with the NMPC-DNN are shown in Figures 6 and 7a (the blue dashed curves), where the red dashed lines represent the lower bound $\mu_{\min}$ and the upper bound $\mu_{\max}$ in the control input constraints, respectively. The synchronization error trajectories are displayed in Figure 7b. The Mean Squared Error of DNN is shown in Figure 8.

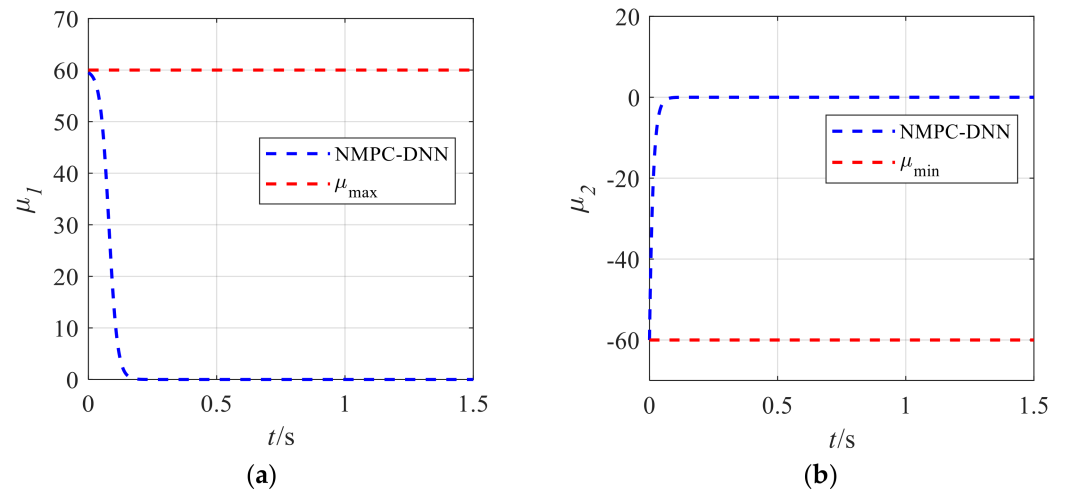


Figure 6. Trajectory of (a) control input μ_1 ; (b) control input μ_2 .

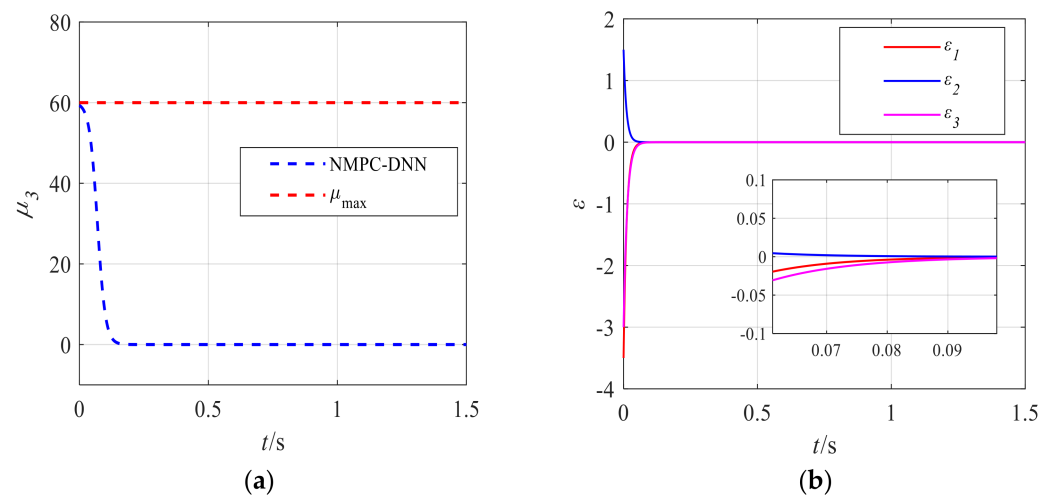


Figure 7. Trajectory of (a) control input μ_3 ; (b) synchronization errors ε_1 , ε_2 , and ε_3 .

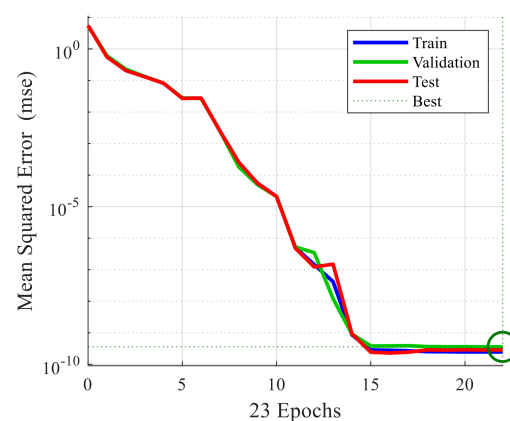


Figure 8. The Mean Squared Error of DNN.

4.2.1. Comparison of the NMPC-DNN with Other Control Methods

A performance comparison of the NMPC-DNN with the other controllers is provided in Table 2, based on two key control quality indicators: the settling time and the integral-squared error (ISE).

Table 2. Comparison between the proposed NMPC-DNN and other types of controllers.

Control Strategy	Error State	Settling Time	ISE
This paper (NMPC-DNN)	Example 1	ε_1	0.04 s
		ε_2	0.03 s
		ε_3	0.002
	Example 2	ε_1	0.08 s
		ε_2	0.07 s
		ε_3	0.016
[41]	Example 1	ε_1	0.17 s
		ε_2	0.26 s
		ε_3	0.002
	Example 2	ε_1	0.32 s
		ε_2	0.25 s
		ε_3	0.154

All the data in Table 2 demonstrate that the NMPC-DNN approach provides better control quality, marked by expedited stabilization and a diminished integral-squared error (ISE) for the synchronization error states. Conversely, the approaches in [41] yield a substantially slower stabilization and a higher ISE, reflecting its inferior control accuracy relative to the NMPC-DNN approach.

4.2.2. Online Computational Efficiency of NMPC-DNN

Furthermore, to substantiate the reduced online computational burden, a quantitative comparison of the computation time is recorded under identical software environments in MATLAB R2018, Intel Core i7-9750H CPU @ 2.60GHz. With the NMPC based on online optimization, at each sampling instant k , the finite-horizon optimization problem is solved with the CasADi/IPOPT solver. Computing the control inputs with the NMPC-DNN online requires only simple matrix multiplications and activation function evaluations. For Example 1, Table 3 summarizes the computational CPU time per sampling step for both controllers.

As shown in Table 3, the NMPC-DNN approach accelerates the execution speed by approximately 118 times compared to NMPC based on online optimization. This quantitative evidence explicitly demonstrates that the proposed method effectively reduces the online computational burden.

Table 3. Quantitative comparison of online computational efficiency.

Control Strategy	Average CPU Time (s)
NMPC based on online optimization	0.0271 s
Proposed NMPC-DNN	0.00023 s

5. Conclusions

To address the synchronization control challenges of chaotic systems with constraints and improve the control performance, this paper presents an effective control strategy by integrating NMPC with DNN. The neural network is trained offline to approximate the NMPC controller. Then, online, the computation of the control input consists of the function evaluation realized by the neural network. This design alleviates the online computational burden.

The performance of the NMPC-DNN controller was evaluated through numerical simulations and compared with the recent studied control methods. The proposed NMPC-DNN method achieves faster stabilization and lower integral-squared error for synchronization error states. The results demonstrate that the proposed method provides better control accuracy for the considered third chaotic system.

Several limitations of this study should be acknowledged. The proposed method is validated only on a third chaotic system. The offline training data are generated by solving the NMPC problem over set of initial error states; other alternative data generation strategies are not explored. Additionally, the present study focuses on numerical simulations; hardware implementation has not been carried out.

The future work will focus on extending the NMPC-DNN approach to other nonlinear systems, including higher-dimensional chaotic systems and systems with time delays. Systematic offline data generation methods will be investigated to improve training efficiency and controller generalization, such as adaptive sampling or active learning. Meanwhile, embedded applications of the NMPC-DNN approach to specific chaotic systems will be considered.

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

Funding: This research was supported by the Liaoning Institute of Science and Technology (No. 2606B47).

Data Availability Statement: The datasets used and analyzed during the current study are available from the corresponding author on reasonable request.

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

Abbreviations

The following abbreviations are used in this manuscript:

NMPC	Nonlinear Model Predictive Control
DNN	Deep Neural Network
QNN	Quantum Neural Network
BELC	Brain Emotional Learning Controller
FBELNN	Fuzzy Brain Emotional Learning Neural Network
RCMAC	Recurrent Cerebellar Model Articulation Controller
ARMPC	Active Robust Model Predictive Control
RC	Reservoir Computing
RO	Reservoir Observer
OC	Online Control
ZNN-FTSMC	Zeroing Neural Network-Based Fixed-Time Sliding Mode Control

References

1. Lorenz, E. Deterministic nonperiodic flow. *J. Atmos. Sci.* **1963**, *20*, 130–141. [[CrossRef](#)]
2. Pecora, L.M.; Carroll, T.L. Synchronization in chaotic systems. *Phys. Rev. Lett.* **1990**, *64*, 821. [[CrossRef](#)] [[PubMed](#)]
3. Leutcho, G.D.; Wang, H.; Fozin, T.F.; Sun, K.; Njitacke, Z.T.; Kengne, J. Dynamics of a new multistable 4D hyperchaotic Lorenz system and its applications. *Int. J. Bifurc. Chaos* **2022**, *32*, 2250001. [[CrossRef](#)]
4. Keerthana, N.; Elayabharath, V.T.; Sakthivel, R.; Monisha, S. Secure fault estimation and resilient fault-tolerant control for nonlinear chaotic systems based on fuzzy intermediate estimator. *Chaos Solitons Fractals* **2025**, *192*, 115954. [[CrossRef](#)]

5. Yan, M.; Xie, J. A conservative chaotic system with coexisting chaotic-like attractors and its application in image encryption. *J. Control Decis.* **2023**, *10*, 237–249. [[CrossRef](#)]
6. Jankowska, M.A.; Karageorghis, A.; Chen, C. Training RBF neural networks for solving nonlinear and inverse boundary value problems. *Comput. Math. Appl.* **2024**, *165*, 205–216. [[CrossRef](#)]
7. Takara, L.d.A.; Teixeira, A.C.; Yazdanpanah, H.; Mariani, V.C.; Coelho, L.S. Optimizing multi-step wind power forecasting: Integrating advanced deep neural networks with stacking-based probabilistic learning. *Appl. Energy* **2024**, *369*, 123487. [[CrossRef](#)]
8. Della Santina, C.; Duriez, C.; Rus, D. Model-based control of soft robots: A survey of the state of the art and open challenges. *IEEE Control Syst.* **2023**, *43*, 30–65. [[CrossRef](#)]
9. Grancharova, A.; Kocijan, J.; Johansen, T.A. Explicit stochastic predictive control of combustion plants based on Gaussian process models. *Automatica* **2008**, *44*, 1621–1631. [[CrossRef](#)]
10. Liang, H.; Liu, G.; Zhang, H.; Huang, T. Neural-network-based event-triggered adaptive control of nonaffine nonlinear multiagent systems with dynamic uncertainties. *IEEE Trans. Neural Netw. Learn. Syst.* **2020**, *32*, 2239–2250. [[CrossRef](#)] [[PubMed](#)]
11. Alloghani, M.; Al-Jumeily, D.; Mustafina, J.; Hussain, A.; Aljaaf, A.J. A systematic review on supervised and unsupervised machine learning algorithms for data science. In *Supervised and Unsupervised Learning for Data Science*; Springer: Cham, Switzerland, 2020; pp. 3–21.
12. Chen, Y.; Xin, R.; Cheng, M.; Gao, X.; Li, S.; Shao, W.; Deng, L.; Zhang, M.; Fu, S.; Liu, D. Unveil the time delay signature of optical chaos systems with a convolutional neural network. *Opt. Express* **2020**, *28*, 15221–15231. [[CrossRef](#)] [[PubMed](#)]
13. Aneja, S.; Sharma, A.; Gupta, R.; Yoo, D.Y. Bayesian regularized artificial neural network model to predict strength characteristics of fly-ash and bottom-ash based geopolymer concrete. *Materials* **2021**, *14*, 1729. [[CrossRef](#)] [[PubMed](#)]
14. McCulloch, W.S.; Pitts, W. A logical calculus of the ideas immanent in nervous activity. *Bull. Math. Biophys.* **1943**, *5*, 115–133. [[CrossRef](#)]
15. Hunt, K.J.; Sbarbaro, D.; Żbikowski, R.; Gawthrop, P.J. Neural networks for control systems—A survey. *Automatica* **1992**, *28*, 1083–1112. [[CrossRef](#)]
16. Kocijan, J. *Modelling Dynamic Systems with Artificial Neural Networks and Related Methods*; University of Nova Gorica Press: Nova Gorica, Slovenia, 2023.
17. Aliabadi, F.; Majidi, M.H.; Khorashadizadeh, S. Chaos synchronization using adaptive quantum neural networks and its application in secure communication and cryptography. *Neural Comput. Appl.* **2022**, *34*, 6521–6533. [[CrossRef](#)]
18. Lin, C.-M.; Pham, D.-H.; Huynh, T.-T. Synchronization of Chaotic System Using a Brain-Imitated Neural Network Controller and Its Applications for Secure Communications. *IEEE Access* **2021**, *9*, 75923–75944. [[CrossRef](#)]
19. Thi, H.C.P.; Dang, N.Q.; Giap, V.N. Secure Communication of Electric Drive System Using Chaotic Systems Base on Disturbance Observer and Fuzzy Brain Emotional Learning Neural Network. *Math. Comput. Appl.* **2025**, *30*, 73. Correction in *Math. Comput. Appl.* **2025**, *30*, 118. [[CrossRef](#)]
20. Sun, Y.; Ouyang, S.; Zeng, H.; Wang, Q.; Gao, J. Hyperchaos synchronization control of interval type-2 fuzzy brain emotional learning and its application in secure communication. *J. Chongqing Univ.* **2026**, *49*, 55–68.
21. Wang, E.; Hu, B.; Guan, Z.-H. Chaotic dynamics and synchronization of multi-region neural network based on locally active memristor. *Chaos Solitons Fractals* **2025**, *197*, 116437. [[CrossRef](#)]
22. Chen, Q.; Sabir, Z.; Mehmood, M.A.; Baskonus, H.M. A machine learning radial basis deep neural network for solving the fractional chaotic financial system. *J. Comput. Appl. Math.* **2025**, *474*, 116936. [[CrossRef](#)]
23. Sabetahd, R.; Jafarzadeh, O. Development of an adaptive chaotic fuzzy neural network controller for mitigating seismic response in a structure equipped with an active tuned mass damper. *Expert Syst. Appl.* **2025**, *267*, 126048. [[CrossRef](#)]
24. An, N.H.; Giap, V.N. 4-D chaotic system synchronization for secure wireless control of DC motor using fuzzy brain-inspired neural network controller and disturbance observer. *Nonlinear Sci.* **2025**, *5*, 100068. [[CrossRef](#)]
25. Seifouripour, Y.; Nobahari, H. A control architecture for fixed-wing aircraft based on the convolutional neural networks. *J. Frankl. Inst.* **2024**, *361*, 106664. [[CrossRef](#)]
26. Alhajeri, M.S.; Ren, Y.M.; Ou, F.; Abdullah, F.; Christofides, P.D. Model predictive control of nonlinear processes using transfer learning-based recurrent neural networks. *Chem. Eng. Res. Des.* **2024**, *205*, 1–12. [[CrossRef](#)]
27. Schwedersky, B.B.; Flesch, R.C. Nonlinear model predictive control algorithm with iterative nonlinear prediction and linearization for long short-term memory network models. *Eng. Appl. Artif. Intell.* **2022**, *115*, 105247. [[CrossRef](#)]
28. Zhang, C.; Xu, J.; Niu, Y. Event-triggered adaptive deep neural network sliding mode control design for unmanned aerial vehicle systems. *Frankl. Open* **2024**, *8*, 100120. [[CrossRef](#)]
29. Kapnopoulos, A.; Kazakidis, C.; Alexandridis, A. Quadrotor trajectory tracking based on backstepping control and radial basis function neural networks. *Results Control Optim.* **2024**, *14*, 100335. [[CrossRef](#)]
30. Bemporad, A.; Morari, M. Robust model predictive control: A survey. In *Robustness in Identification and Control*; Lecture Notes in Control and Information Sciences; Springer: London, UK, 1999; Volume 245, pp. 207–226.

31. Grancharova, A.; Olaru, S. Distributed Model Predictive Control of Time-Delay Systems. *IFAC-PapersOnLine* **2022**, *55*, 85–90. [[CrossRef](#)]
32. Shi, L.; Yan, S.; Li, W. Consensus and products of substochastic matrices: Convergence rate with communication delays. *IEEE Trans. Syst. Man Cybern. Syst.* **2025**, *55*, 4752–4761. [[CrossRef](#)]
33. Wang, Z.; Li, W. Co-Design of Measurement–Communication and Distributed Model Predictive Control for Multi-AUV Formation Tracking. *Appl. Math. Model.* **2026**, *159*, 117033. [[CrossRef](#)]
34. Li, W.; Yan, S.; Shi, L.; Yue, J.; Shi, M.; Lin, B.; Qin, K. Multiagent consensus tracking control over asynchronous cooperation–competition networks. *IEEE Trans. Cybern.* **2025**, *55*, 4347–4360. [[CrossRef](#)] [[PubMed](#)]
35. Parisini, T.; Zoppoli, R. A receding-horizon regulator for nonlinear systems and a neural approximation. *Automatica* **1995**, *31*, 1443–1451. [[CrossRef](#)]
36. Kardamaki, A.; Protoulis, T.; Alexandridis, A.; Sarimveis, H. An explicit model predictive control framework based on physics-informed neural networks. *J. Process Control* **2026**, *15*, 103634. [[CrossRef](#)]
37. Djouadi, H.; Ouari, K.; Belkhier, Y.; Lehouché, H.; Ibaouene, C.; Bajaj, M.; AboRas, K.M.; Khan, B.; Kamel, S. Nonlinear multivariable permanent magnet synchronous machine control: A robust nonlinear generalized predictive controller approach. *IET Control Theory Appl.* **2023**, *17*, 1688–1702. [[CrossRef](#)]
38. Maddalena, E.T.; Moraes, C.G.d.S.; Waltrich, G.; Jones, C.N. A neural network architecture to learn explicit MPC controllers from data. *IFAC-PapersOnLine* **2020**, *53*, 11362–11367. [[CrossRef](#)]
39. Ren, R.; Li, S. Knowledge-driven approximate model predictive controller with probabilistic performance validation. *Asian J. Control* **2026**, *28*, 1435–1447. [[CrossRef](#)]
40. Li, J.; He, D.; Wang, X.; Kang, Y. BP neural network-based explicit MPC of nonlinear boiler-turbine systems. *J. Process Control* **2025**, *145*, 103353. [[CrossRef](#)]
41. Xie, J.; Grancharova, A.; Fan, L. Synchronization of chaotic systems using novel sliding mode neural network controller. In Proceedings of the 2023 International Conference Automatics and Informatics (ICAI), Varna, Bulgaria, 5–7 October 2023; pp. 301–305.
42. Karg, B.; Alamo, T.; Lucia, S. Probabilistic performance validation of deep learning-based robust NMPC controllers. *Int. J. Robust Nonlinear Control* **2021**, *31*, 8855–8876. [[CrossRef](#)]
43. Andersson, J.A.E.; Gillis, J.; Horn, G.; Rawlings, J.B.; Diehl, M. CasADi: A software framework for nonlinear optimization and optimal control. *Math. Program. Comput.* **2019**, *11*, 1–36. [[CrossRef](#)]
44. Xie, J. Intelligent Model-Based Control of Nonlinear Systems. Doctoral Dissertation, University of Chemical Technology and Metallurgy, Sofia, Bulgaria, 2025.
45. Jiang, Z.P.; Wang, Y. Input-to-state stability for discrete-time nonlinear systems. *Automatica* **2001**, *37*, 857–869. [[CrossRef](#)]
46. Mayne, D.Q.; Rawlings, J.B.; Diehl, M. *Model Predictive Control: Theory and Design*, 2nd ed.; Nob Hill Publishing, LCC: Madison, WI, USA, 2009.
47. Mayne, D.Q.; Rawlings, J.B.; Rao, C.V.; Scokaert, P.O.M. Constrained model predictive control: Stability and optimality. *Automatica* **2000**, *36*, 789–814. [[CrossRef](#)]
48. Grüne, L.; Pannek, J. *Nonlinear Model Predictive Control: Theory and Algorithms*; Springer International Publishing: Cham, Switzerland, 2016; pp. 45–69.
49. Grüne, L.; Rantzer, A. On the infinite horizon performance of receding horizon controllers. *IEEE Trans. Autom. Control* **2008**, *53*, 2100–2111. [[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.