actuators

MDPI

Article

A Sparse Neural Network-Based Control Method for Saturated Nonlinear Affine Systems

Jing Zhang [1,2,*], Baoqun Yin [1], Jianwen Huo [2], Hongliang Guo [3] and Zhan Li [4]

[1] Department of Automation, University of Science and Technology of China, Hefei 230022, China; bqyin@ustc.edu.cn
[2] School of Information Engineering, Southwest University of Science and Technology, Mianyang 621002, China; huojianwen@swust.edu.cn
[3] College of Computer Science, Sichuan University, Chengdu 610044, China; guohongliang@scu.edu.cn
[4] Department of Computer Science, Swansea University, Swansea SA2 8PP, UK; zhan.li@swansea.ac.uk
* Correspondence: zhangjing@swust.edu.cn

Abstract: Saturated nonlinear affine systems are widely encountered in many engineering fields. Currently, most control methods on saturated nonlinear affine systems are not specifically designed based on sparsity-based control methodologies, and they might require sparse identification at the beginning stage and applying tracking control afterwards. In this paper, a sparse neural network (SNN)-based control method from an l_p-norm $(1 \leq p < 2)$ optimization perspective is proposed for saturated nonlinear affine systems by taking advantage of the nice properties of primal dual neural networks for optimization. In particular, when $p = 1$, a new alternative controller based on SNN is derived, encountering computational difficulties distinct from those of another solution set in the basic dual neural network. The convergence properties of such SNN-based controllers are investigated and analyzed to find a control solution. Five illustrative examples further are shown to demonstrate the efficiency of the proposed SNN-based control method for tracking the desired references of saturated nonlinear affine systems. In the practical application scenario involving the UR5 robot control, the trajectory's average errors are consistently confined to a minimal magnitude of 10^{-4} m. These findings substantiate the efficacy of the SNN-based control approach proposed for precise tracking control in saturated nonlinear affine systems.

Keywords: saturation system; nonlinear affine system; sparse neural network; redundant manipulator

Citation: Zhang, J.; Yin, B.; Huo, J.; Guo, H.; Li, Z. A Sparse Neural Network-Based Control Method for Saturated Nonlinear Affine Systems. *Actuators* **2024**, *13*, 204. https://doi.org/10.3390/act13060204

Academic Editors: Ioan Ursu, Xuerui Wang and Erik-Jan Van Kampen

Received: 10 April 2024
Revised: 19 May 2024
Accepted: 27 May 2024
Published: 29 May 2024

1. Introduction

Affine control systems have recently risen to prominence in both science and engineering, making the development of their control strategies a key research area. These nonlinear affine systems are prevalent across various domains, such as formation control of multi-agents [1], wind power transfer [2], motion planning, and control of kinematically redundant manipulators [3,4]. Addressing the control challenges of these systems has been central to nonlinear affine control research, leading to diverse strategies for different applications. Notable examples include self-triggering model predictive control [5], adaptive PID controllers for uncertain environments [6], and output regulation for switched affine systems in the circuit model [7].

However, in real physical systems and engineering, due to inherent physical properties or the establishment of safety thresholds, the system's response exhibits an upper or lower limit. The input or output of the system ceases to show any further increase or decrease upon reaching a specific threshold. This also causes the input of the nonlinear affine system to reach saturation [8,9], e.g., trajectory tracking control for surface ships [10], modulation control of power converters [11], spacecraft fly tracking control [12,13], vibration control of flexible strings [14]. Saturated nonlinear control systems have important research significance. However, they often have complex mathematical models. The utilization of

sparsity-based control techniques in saturated nonlinear affine systems can significantly enhance the performance of the control system. The main challenge is to address the issue of sparse identification in dynamic systems [15,16]. To achieve computational efficiency, Selesnick et al. introduced a technique for implementing sparse regularization using convex analysis [17]. Babazadeh et al. propose a new method to construct optimal sparse control structures that are guaranteed to have a prescribed number of zero elements [18]. The effectiveness of the approach is evaluated through synthetic and real-life case studies. Torres et al. proposed sparse controls for network spread processes [19]. Wensing et al. proposed the sparse control of dynamic movement primitive systems by introducing a sparsely inhibited rhythmic alternative [20]. Kaheman et al. developed SINDy-PI (sparse identification of nonlinear dynamics—parallel, implicit), which is an enhanced iteration of the SINDy algorithm [21]. This version has the ability to detect underlying dynamics and logical non-linearities. To address the trajectory tracking problem of industrial manipulator systems with modeling uncertainties, varying loads, and unknown dead-band characteristics, Zhao et al. proposed a compensation-based adaptive switching control solution [22].

At the same time, neural network control methods based on affine systems are also constantly developing. Meng et al. studied a group of high-order nonaffine nonlinear models whose dynamics are entirely unknown in relation to adaptive neural network (NN) regulation [23]. Lin et al. designed a data-fault-tolerant controller based on a particle swarm neural network for nonlinear affine systems [24]. Kim et al. achieved optimal control over nonlinear affine systems through the application of deep reinforcement learning methods [25]. Li et al. used adaptive reinforcement learning to improve the robustness of nonlinear affine systems [26]. Although complex, intelligent neural networks provide benefits, they also introduce new problems. These neural network methods are often large and consume excessive control resources. Researchers such as Louizos et al. [27] and Srinivas et al. [28] have been working on the problem of analyzing the sparsity of neural networks. This has been a topic of interest. Therefore, studying the sparsity of neural networks is highly significant. Tang et al. studied the automatic sparseness of neural networks [29]. Nonlinear affine systems often present intricate dynamics, with nonlinear and interconnected relationships between states and inputs. Traditional control methods may struggle to handle such complexity efficiently, requiring extensive parameter tuning and computational resources, particularly for large-scale systems. Sparse control methods offer a solution by leveraging the sparsity or locality properties of system dynamics. To address such a problem, Reiners et al. studied and verified the efficiency of sparse neural networks (SNNs) [30]. By focusing control inputs on key aspects of the system dynamics, sparse control methods achieve more efficient control. In nonlinear affine systems, sparsity may manifest in localized linearizations of system dynamics or the relevance of a few critical states/inputs. Thus, adopting sparse control methods can effectively exploit the system's structural properties, simplifying controller design while improving performance.

Among the many sparse control studies, few have combined sparse control with neural networks. However, we argue that combining SNN with control systems is of great significance for intelligent control methods. Tian et al. introduced a neural-network-based approach for solving large-scale sparse optimization problems with many objectives [31]. Zhu et al. introduced a method for collaboration between an SNN design and a hardware design [32]. Moreover, control methods based on SNN also have specific applications, e.g., application in hypersonic aircraft [33]; application in vehicle anti-lock braking systems, etc., [34].

In the realm of robotics, researchers have advocated using redundant robots to include sparse optimization into recurrent neural networks for kinematic control. This approach aims to reduce kinetic energy and address failures in joints with zero velocity. Examples of such work include studies by Li et al. [35] and Li et al. [36]. To tackle the challenge of saturated nonlinear control in redundant robotic manipulators, Zhang et al. introduced and applied a novel varying-parameter convergent-differential neural network (VP-CDNN) [37].

Zhao et al. developed a unique control strategy for managing saturated nonlinear systems, merging projection-based operational space control with a neural network-based adaptive controller, and implemented it for the manipulation and grasping tasks of mobile robotic arms [38]. Bilal et al. employed neural networks with sparse regression to enhance the precision of robot posture processing [39]. Shukla et al. used SNN for robot grasping motion [40]. Sayar et al. developed a two-timescale recurrent neural network (TNN) optimization strategy that employs the infinity norm as the objective function and uses slack variables to circumvent situations where optimization fails to meet equality constraints, thereby enhancing the smoothness and precision of motion for a 9-degree-of-freedom robot end-effector [41]. Pan et al. applied generalized SNN to force observations [42]. Utilizing SNN to control saturated nonlinear affine systems provides novel perspectives for tackling numerous challenges in nonlinear system control [43].

In this study, inspired by prior studies, we propose a control strategy for nonlinear affine systems based on sparse neural network (SNN). Our proposed sparse norm-based neurodynamic optimization control strategy utilizes neural networks and sparse norms to enforce sparsity constraints on control inputs, and it can also solve the saturation constraint problem of control inputs. This approach enables us to achieve control input sparsity while maintaining control performance, simplifying controller implementation and computation. Initially, we formulated optimal constraint conditions, converting the control issue of saturated nonlinear systems into an optimization problem. Subsequently, we developed an SNN control strategy with an l_p-norm $(1 \leq p < 2)$ optimization, which is solved using a novel primal dual neural network. The convergence properties of such an SNN-based control method are analyzed and substantiated. Furthermore, we engineered an l_p-norm, enhancing system input smoothness and reducing end-effector jitter by adjusting the p norm. Additionally, the efficacy of the proposed approach for a variety of saturated nonlinear affine systems was demonstrated through five examples.

The remainder of the paper is as follows. In Section 2, the basic principles and problem formulation of the nonlinear affine system are introduced. Section 3 discusses the suggested strategy for SNN and presents its theoretical results within the relevant optimization framework. The control results via five examples are demonstrated in Section 4. Section 5 presents the conclusion and observations.

2. Problem Formulation

This study focuses on analyzing a complex nonlinear system with multiple inputs and outputs, known as an affine-dynamic system.

$$\dot{x} = f(x) + g(x)u \tag{1}$$

where $x \in R^n$ is the nonlinear affine system's state variable vector, $f(x) \in R^n$ is the numerical array of nonlinear mapping functions with vector values, and $g(x) \in R^{n \times m}$ denotes the matrix-valued nonlinear mapping function array, and $u \in R^m$ denotes the control input variable vector with saturation $u^- \leq u \leq u^+$. The nonlinear affine system should be guaranteed to be globally stable under the suitable control input u within the boundary $[u^- u^+]$.

To obtain a suitable control input for the nonlinear affine system, we need to first rewrite (1) as:

$$g(x)u = \dot{x} - f(x) \tag{2}$$

Therefore, when $n = m$, the following statement can be used to analytically solve for the control input u.

$$u = g^{-1}(x)[\dot{x} - f(x)] \tag{3}$$

where $g(\cdot)$ is the original mapping function array, $g^{-1}(\cdot) \in R^{n \times m}$ denotes the inverse mapping function array of $g(\cdot)$. However, it is usually not necessary or convenient to find the inverse mapping of $g(\cdot)$ with input saturation directly in an analytical manner, and

it may yield additional computational cost to formulate a suitable control input u as the control law for the nonlinear affine system in real time.

To achieve the required global and exponential convergence of the state vector x to the desired state x_d, in this work we propose to make the following equation hold between the state variable and the desired one during the control process at the first-order time derivative level:

$$\dot{x} - \dot{x}_d = -k(x - x_d) \tag{4}$$

where the control parameter $k > 0$ denotes the convergence scaling parameter that can be used to scale or accelerate the convergence of the controller, which is reflected in the exponential decay rate. A larger k results in faster convergence. From (4), one can evidently obtain:

$$x - x_d = e^{-kt}[x(0) - x_d(0)] \tag{5}$$

which suggests that x has the potential for exponential convergence towards the target state x_d from their initial conditions at $x(0)$ and $x_d(0)$. As a result, the reformulation of Equation (1) into Equation (6) becomes feasible.

$$g(x)u = -k(x - x_d) + \dot{x}_d - f(x) \tag{6}$$

As u is difficult to solve in an algebraic manner and cannot contain any sparsity information, solving u within a dynamic equation might be another feasible routine. Our proposed SNN-based control strategy is developed based on the specific nonlinear affine system to achieve exponential convergence for the state variable, which will be described in detail in subsequent sections.

3. Methodology

To achieve the tracking control task, we propose an SNN control paradigm for the aforementioned saturated nonlinear affine system from an $l_p(1 \leq p < 2)$ norm-based optimization perspective, as follows:

$$\begin{aligned} &\text{minimize } \|u\|_p \\ &\text{subject to } g(x)u = -k(x - x_d) + \dot{x}_d - f(x), u^- \leq u \leq u^+ \end{aligned} \tag{7}$$

where $\|u\|_p = \left(\sum_{i=1}^{m} u_i^p\right)^{\frac{1}{p}}$. The rationale for such an optimization is to utilize the constrained convex paradigm to formulate the saturated control input variable. Employing the $l_p(1 \leq p < 2)$ norm, we ensure convexity in the objective function, relying solely on the engagement of the control input u and the preservation of sparsity. Other types of convex variants of the objective function on the control input u may be possible, but $||u||_p^p$ can be a concise form. To further develop the SNN-based controller according to the l_p-norm based optimization paradigm, it is necessary to formulate the Lagrange function in the following manner:

$$L(u, \lambda) = \|u\|_p / p + \lambda^T \left[g(x)u + k(x - x_d) - \dot{x}_d + f(x)\right] \tag{8}$$

where $\lambda \in R^n$ refers to the Lagrangian coefficient vector. Inspired by the construction methodology of neural networks following primal-dual principles and feasible solution sets [44], the subsequent SNN-based controller with the l_p optimization paradigm is further developed:

$$\begin{cases} \epsilon \dot{u} = -u + P_{\Omega_p}\left(u - \frac{\partial L(u,\lambda)}{\partial u}\right) \\ \epsilon \dot{\lambda} = g(x)u + k(x - x_d) - \dot{x}_d + f(x) \end{cases} \tag{9}$$

where $\epsilon > 0$ represents the parameter scaling the convergence of dynamics, along with the array operator $P_{\Omega_p}(\cdot)$ denotes a piecewise linear projection with a feasible solution set Ω_p and an input vector z, which is:

$$P_{\Omega_p}(z) = \begin{cases} z^+, z > z^+ \\ z, z^- \leq z \leq z^+ \\ z^-, z < z^- \end{cases} \tag{10}$$

where z^- and z^+ constrain the input z as the lower and upper bounds, respectively.

Figure 1 shows the block diagram of the proposed SNN controller. Unlike traditional artificial neural networks (ANNs), which typically rely on dense layers of neurons and require extensive data for training, our SNN is designed to exploit sparsity in both its structure and operation. As illustrated in Figure 1a, SNN is implemented using a dual-neuron configuration. In contrast, the control framework based on ANN, as shown in Figure 1b, comprises an input layer, hidden layers, and an output layer composed of multiple neurons. This approach significantly reduces the network's complexity and the need for training data, making it particularly suited for control tasks where computational efficiency and adaptability are critical. The SNN operates based on a principle that selectively activates a minimal number of neurons for any given input, thereby achieving a balance between control performance and computational load. The process of solving the parameters of the SNN is analogous to that of the zeroing neural network (ZNN), both of which are referred to as neural dynamic solving. Initially, it is necessary to design an error function to measure the solving error. Subsequently, a derivative of this time-based error function with respect to time is taken, ensuring that all terms within the error function converge to zero. This error function is then substituted into the differential equation to solve for the neural network parameters. Unlike traditional ANNs (artificial neural networks) and CNNs (convolutional neural networks), which require extensive data for parameter training, the SNN does not necessitate such extensive data training.

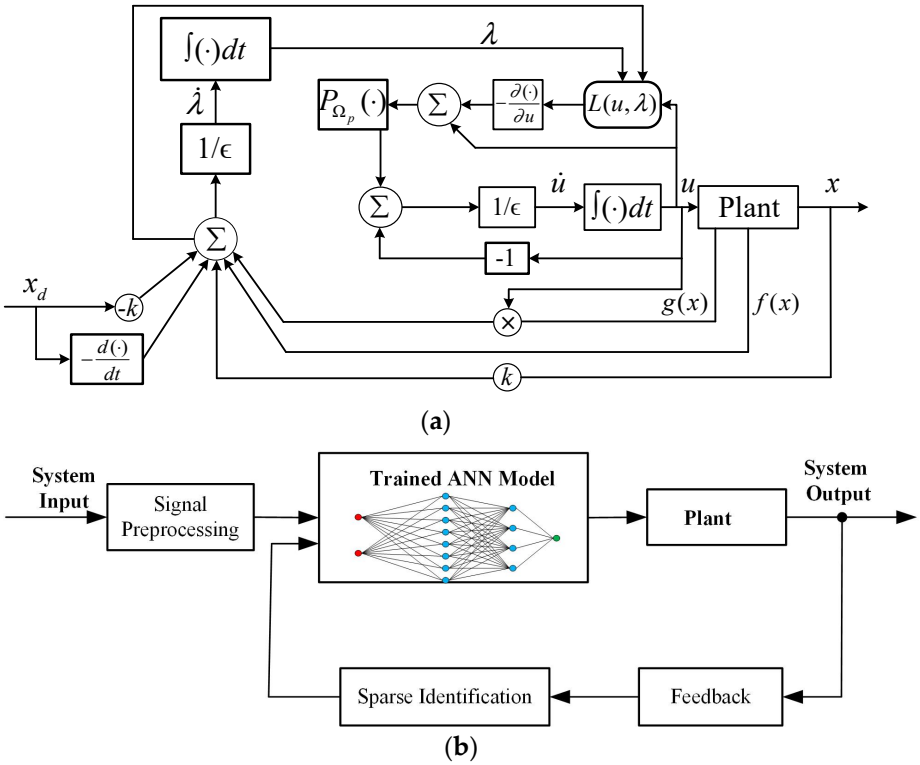


Figure 1. The block diagram of controller. (**a**) The block diagram of SNN-based controller. (**b**) The block diagram of ANN based controller.

The designed controller, employing SNN within the optimization paradigm, can be segmented into the following two conditions:

Condition 1. *For* $1 < p < 2$, *The derivative of the Lagrange function's control input u can be expressed as:*

$$\begin{aligned}\frac{\partial L(u,\lambda)}{\partial u} &= \frac{\partial \|u\|_p / p}{\partial u} + \frac{\partial \lambda^T [g(x)u + k(x - x_d) - \dot{x}_d + f(x)]}{\partial u} \\ &= p\mathrm{sgn}^{(p-1)}(u) + g^T(x)\lambda n\end{aligned} \tag{11}$$

where $sgn^{(p-1)}(\cdot)$ *denotes the sign function array* $R^n \to R^n$ *with its form being:*

$$\mathrm{sgn}^{(p-1)}(u) = \begin{bmatrix} |u_1|^{(p-1)}\mathrm{sgn}(u_1) \\ |u_2|^{(p-1)}\mathrm{sgn}(u_2) \\ \vdots \\ |u_i|^{(p-1)}\mathrm{sgn}(u_2) \\ \vdots \\ |u_n|^{(p-1)}\mathrm{sgn}(u_n) \end{bmatrix} \tag{12}$$

and its each entry is

$$\mathrm{sgn}(u_i) = \begin{cases} 1, & \text{if } u_i > 0 \\ 0, & \text{if } u_i = 0 \\ -1, & \text{if } u_i < 0 \end{cases} \tag{13}$$

Thus, the based controller (9) further becomes

$$\begin{cases} \epsilon \dot{u} = -u + P_{\Omega_p}\left(u - p\mathrm{sgn}^{(p-1)}(u) + g^T(x)\lambda\right) \\ \epsilon \dot{\lambda} = g(x)u + k(x - x_d) - \dot{x}_d + f(x) \end{cases} \tag{14}$$

Condition 2. *Especially, for* $p = 1$, *the SNN-based controller (9) is*

$$\begin{cases} \epsilon \dot{u} = -u + P_{\Omega_1}\left(u - \mathrm{sgn}^{(0)}(u) + g^T(x)\lambda\right) \\ \epsilon \dot{\lambda} = g(x)u + k(x - x_d) - \dot{x}_d + f(x) \end{cases} \tag{15}$$

However, the sign operator $sgn^{(0)}(u)$ *involves term* 0^0 *when* $u = 0$ *appears, which is meaningless or indefinite, the SNN-based controller may fall into certain unstable conditions. To remedy this drawback, we devise an alternative formulation for the SNN-based controller by leveraging the subsequent optimization paradigm,*

$$\begin{aligned} &\mathrm{minimize} a^T v \\ &\textit{subject to } g(x)u = -k(x - x_d) + \dot{x}_d - f(x) \end{aligned} \tag{16}$$

where $u^- \le u \le u^+$, $-u \le v \le u$, $a = [c, c, \cdots, c]^T \in R^n$ *denotes the coefficient vector with a constant entry* $c > 0$, *and* $v \in R^n$ *denotes the newly involved unknown variable for optimization. Under such circumstance, by redefining the Lagrange function:*

$$L(w, \lambda) = a^T v + \lambda^T \left[g(x)u + k(x - x_d) - \dot{x}_d + f(x)\right] \tag{17}$$

The new SNN-based controller becomes

$$\begin{cases} \epsilon \dot{w} = -w + P_{\widetilde{\Omega}_1}\left(w - \frac{\partial L(w,\lambda)}{\partial w}\right) \\ \epsilon \dot{\lambda} = g(x)u + k(x - x_d) - \dot{x}_d + f(x) \end{cases} \tag{18}$$

where $w = \left[u^T \; v^T\right]^T$ denotes the newly combined control input, $\widetilde{\Omega}_1$ denotes a new solution set for a new projector $P_{\widetilde{\Omega}_1}(w)$ that is composed as

$$P_{\widetilde{\Omega}_1}(w) = \cup P_{\widetilde{\Omega}_{1,i}}(w_i) \tag{19}$$

with

$$P_{\widetilde{\Omega}_{1,i}}(w_i) = \begin{cases} w_i, & |u_i| \leq u_i^+ \text{ and } |u_i| \leq v_i \\ \begin{bmatrix} u_i^+ \\ v_i \end{bmatrix}, & u_i \geq u_i^+ \text{ and } v_i \geq u_i^+ \\ \begin{bmatrix} u_i^- \\ v_i \end{bmatrix}, & u_i \leq u_i^- \text{ and } v_i \geq u_i^+ \\ \begin{bmatrix} u_i^+ \\ u_i^+ \end{bmatrix}, & v_i \leq u_i^+ \text{ and } u_i \geq -v_i + 2u_i^+ \\ \begin{bmatrix} u_i^- \\ u_i^+ \end{bmatrix}, & v_i \leq u_i^+ \text{ and } u_i \leq v_i + 2u_i^- \\ \begin{bmatrix} \frac{u_i+v_i}{2} \\ \frac{u_i+v_i}{2} \end{bmatrix}, & |v_i| \leq u_i \leq -v_i + 2u_i^+ \\ \begin{bmatrix} \frac{u_i-v_i}{2} \\ \frac{v_i-u_i}{2} \end{bmatrix}, & v_i + 2u_i^- \leq u_i \leq -|v_i| \\ 0, & |u_i| \leq -v_i \end{cases} \tag{20}$$

Therefore, the specific form of the new SNN-based controller with $p = 1$ is

$$\begin{cases} \epsilon \dot{w} = -w + P_{\widetilde{\Omega}_1}\left(w - \begin{bmatrix} g^T(x)\lambda \\ a \end{bmatrix}\right) \\ \epsilon \dot{\lambda} = g(x)u + k(x - x_d) - \dot{x}_d + f(x) \end{cases} \tag{21}$$

The newly derived SNN-based controller has new dynamic constraints v, and a compound control variable w is generated. Such a method of processing may increase the sparsity as the new dynamic constraints limit the original control input variable u, but might increase oscillations in its solution process.

Concerning the convergence properties of controllers based on SNN, we present the following theoretical analysis theorem.

Theorem 1. *To track the desired/reference state variable x_d for nonlinear affine system (1) with input saturation $u^- \leq u \leq u^+$. The nonlinear affine system synthesized through the utilization of the proposed controllers based on SNN guarantees that the tracking error $e = x - x_d$ globally converge to zero.*

Proof. Due to the convexity of the l_p $(1 \leq p < 2)$ norm, and according to the widely recognized Karush–Kuhn–Tucker (KKT) condition [45,46], we have

$$-\frac{\partial L(u,\lambda)}{\partial u} \in N_{\Omega_p}(u), \frac{\partial L(u,\lambda)}{\partial \lambda} = 0 \tag{22}$$

and

$$-\frac{\partial L(w,\lambda)}{\partial w} \in N_{\widetilde{\Omega}_1}(w), \frac{\partial L(w,\lambda)}{\partial \lambda} = 0 \tag{23}$$

where $N_{\Omega_p}(u)$ and $N_{\widetilde{\Omega}_1}(w)$ denote, respectively, the normal cones of solution sets Ω_p and $\widetilde{\Omega}_1$ for the sparse controllers. In the steady state of a nonlinear affine system using the SNN-based controllers that make projections to normal cones, we have

$$\begin{cases} P_{\Omega_p}\left(u - \frac{\partial L(u,\lambda)}{\partial u}\right) = u \\ P_{\widetilde{\Omega}_1}\left(w - \frac{\partial L(w,\lambda)}{\partial w}\right) = w \end{cases} \tag{24}$$

All these derivations indicate that $\dot{\lambda} \to 0$ can be achieved during the steady state as $\dot{u} \to 0$ and $\dot{w} \to 0$ are satisfied via the projection to normal cones. In this situation, $\dot{\lambda} = g(x)u + k(x - x_d) - \dot{x}_d + f(x) \to 0$ can be guaranteed to let x $\to x_d$.

Let us define tentative variables $\xi_1 = -u + P_{\Omega_p}\left(u - \frac{\partial L(u,\lambda)}{\partial u}\right)$ and $\xi_2 = -w + P_{\widetilde{\Omega}_1}\left(w - \frac{\partial L(w,\lambda)}{\partial w}\right)$ for the two sparse neural networks based controllers, and then we have $\epsilon\dot{u} = \xi_1$ and $\epsilon\dot{w} = \xi_2$.

Next, Lyapunov functions $V_1 = u^T u/2 \geq 0$ and $V_2 = w^T w/2 \geq 0$ for nonlinear affine system synthesized by the two sparse neural network-based controllers are defined and positive-definite, and their time derivatives are, respectively,

$$\dot{V}_1 = u^T\dot{u} = u^T\xi_1/\epsilon = u^T\left[-u + P_{\Omega_p}\left(u - \frac{\partial L(u,\lambda)}{\partial u}\right)\right]/\epsilon \tag{25}$$

and

$$\dot{V}_2 = w^T\dot{w} = w^T\xi_2/\epsilon = w^T\left[-w + P_{\widetilde{\Omega}_1}\left(w - \frac{\partial L(w,\lambda)}{\partial w}\right)\right]/\epsilon \tag{26}$$

Based on the properties of piecewise linear projection [47], we have

$$u^T\left(P_{\Omega_p}(v_1) - u\right) \leq -u^T u \tag{27}$$

and

$$w^T\left(P_{\widetilde{\Omega}_1}(v_2) - w\right) \leq -w^T w \tag{28}$$

where $v_1 = u - \frac{\partial L(u,\lambda)}{\partial u}$ and $v_2 = u - \frac{\partial L(w,\lambda)}{\partial w}$. As a result, we have

$$\dot{V}_1 = -u^T(u - P_\Omega(v_1))/\epsilon \leq -\frac{1}{\epsilon}u^T u \leq 0 \tag{29}$$

and

$$\dot{V}_2 = -w^T\left(w - P_{\widetilde{\Omega}_1}(v_2)\right)/\epsilon \leq -\frac{1}{\epsilon}w^T w \leq 0 \tag{30}$$

which shows that the time derivative of the Lyapunov functions V_1 and V_2 are negative-definite. This indicates that the nonlinear affine systems synthesized by SNN-based controllers are stable to guarantee global convergence with saturated control input solved via the l_p-norm optimization. All of these steps make the proof thus complete. □

4. Results

This section presents the validation of the proposed SNN-based method for saturated nonlinear affine systems through four representative use cases. Parameters $\epsilon = 0.01$, $k = 10$ and randomly generated initial states $x(0)$ are configured for the SNN-based controllers in all examples. Additionally, we evaluate the method's applicability in real-world use cases by deploying it to a real-world UR5 robot's motion control and then quantitatively analyzing the control error.

For underactuated nonlinear systems, the system model can be transformed using methods such as differential flatness. By applying appropriate state transformations or introducing virtual control inputs, we can increase the system's control freedom, ensuring that the dimension of the controlled state matches the dimension of the inputs [48,49]. For third-order and higher-order nonlinear underactuated control systems with non-decomposable input variables, we can transform the system model using approaches similar to higher-order decoupling sliding mode observers [50]. After transforming the system model and appropriately introducing virtual control inputs, we can then utilize the SNN controller to achieve control.

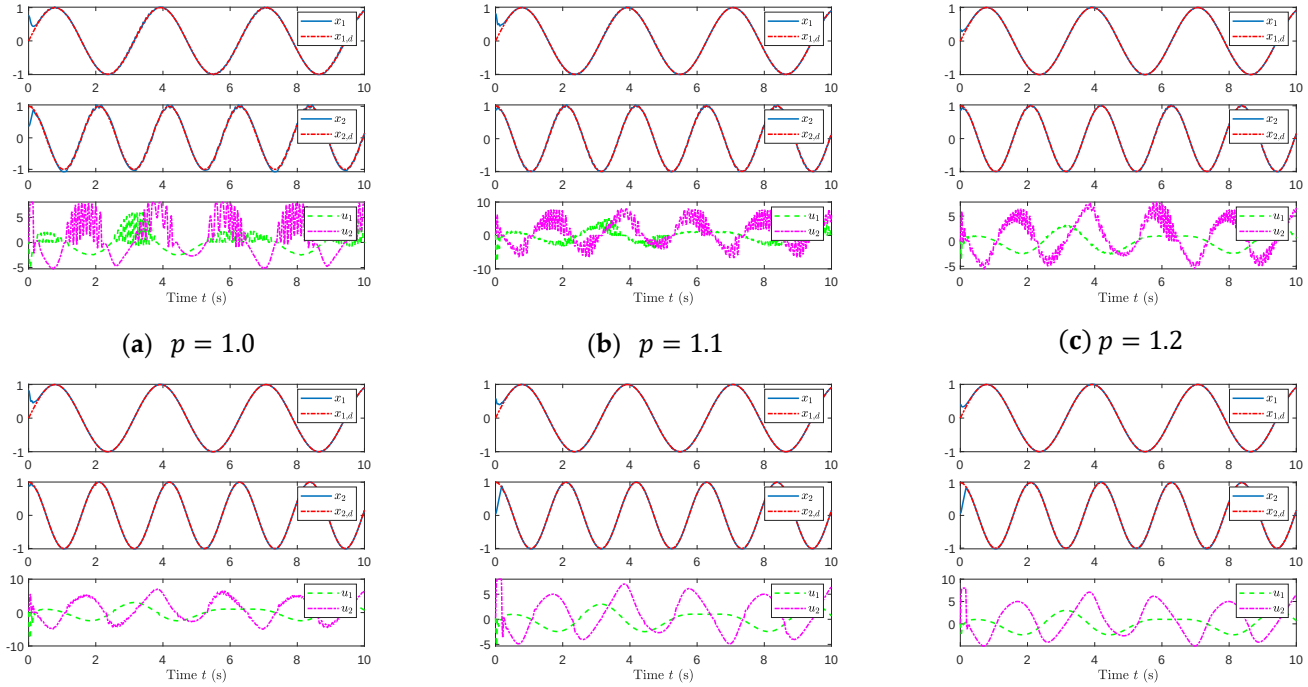
A. Use Case 1: A Second-Order Nonlinear Control System

The second-order nonlinear control system's dynamics is described as:

$$\begin{cases} \dot{x}_1 = x_2 + u_1 \\ \dot{x}_2 = -\cos x_1 - \sin x_2 + \sin(\cos x_1)u_2 \end{cases} \tag{31}$$

A nonlinear affine system has an equal number of state variables as the control input variables. Our goal of control is to eventually make the state variable $x = [x_1\ x_2]^T$ of the nonlinear control system in Equation (31) follow the specified trajectory $x_d = [x_{1,d}\ x_{2,d}]^T$, where $x_{1,d} = \sin 2t$ and $x_{2,d} = \cos 3t$ under the input saturation $u_1^- \leq u_1 \leq u_1^+$ and $u_2^- \leq u_2 \leq u_2^+$ where $u_1^- = u_2^- = -8$, and $u_1^+ = u_2^+ = 8$. The resultant SNN-based controller is then designed using Equation (10), which can be represented as dynamic equations instead of algebraic equations.

Figure 2 shows the control performance results based on the SNN controllers with different p values chosen (i.e., $p = 1.0, 1.1, 1.2, 1.3, 1.5, 1.8$). We can evidently observe that, the proposed SNN-based method is capable of making the nonlinear control system's state vector converge to the specified trajectory elegantly with different parameters p. When the parameter p increases, fewer oscillations occur in the controller curves, which may reveal that l_p optimization with a larger p can produce smoother control inputs as the convexity situation might be different. As p ranges from close to 1 to 2 within the l_p norm, where $1 \leq p \leq 2$, the convexity and sparsity of the norm undergo transformations. As p approaches 1, the convexity of the l_p norm becomes more ronounced. As p gradually increases towards 2, the norm function's convexity diminishes. The norm function becomes less inclined to promote sparsity as p approaches 2, indicating a decrease in sparsity. In summary, as p varies from close to 1 to 2 within the l_p norm, the convexity gradually diminishes while the sparsity decreases.



(**a**) $p = 1.0$ (**b**) $p = 1.1$ (**c**) $p = 1.2$

(**d**) $p = 1.3$ (**e**) $p = 1.5$ (**f**) $p = 1.8$

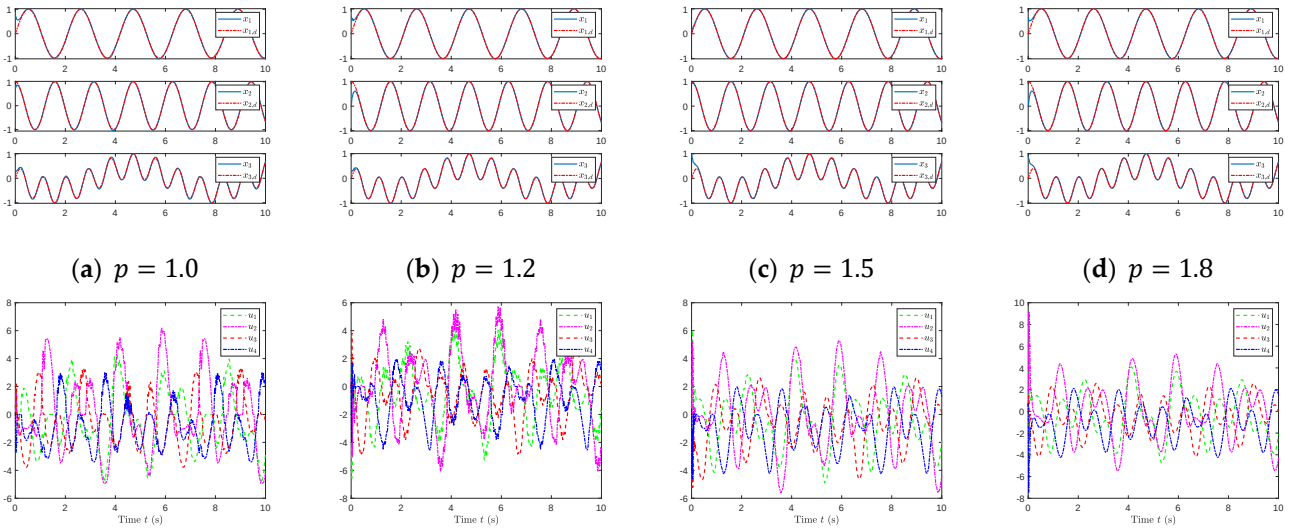
Figure 2. Convergence results of Equation (32) with the SNN control method: (**a**) $t = 10$ s; (**b**) $t = 10$ s; (**c**) $t = 10$ s; (**d**) $t = 10$ s; (**e**) $t = 10$ s; (**f**) $t = 10$ s.

B. Use Case 2: A Nonlinear Control System with Third-Order Over-Determination

This subsection examines the subsequent nonlinear control system with the third-order over-determination:

$$\begin{cases} \dot{x}_1 = \sin^2 x_2 + \cos^2 x_3 + u_1 + u_4 \\ \dot{x}_2 = x_1^4 + x_3^3 + u_2 + u_3 \\ \dot{x}_3 = \sin(\cos x_1) + \cos(\sin x_2) + u_3 + u_4 \end{cases} \tag{32}$$

The dynamic system has more control input variables than state variables, i.e., the number of control inputs is four, but the number of state variables is three. The specified trajectories of the state variables are set as $x_d = [x_{1,d}\ x_{2,d}\ x_{3,d}]^T$ with $x_{1,d} = \sin 3t$, $x_{2,d} = \cos 4t$ and $x_{3,d} = \sin t \cos 2t$, and the saturated control input is $u_1^- \le u_1 \le u_1^+$, $u_2^- \le u_2 \le u_2^+$ and $u_3^- \le u_3 \le u_3^+$ where $u_1^- = u_2^- = u_3^- = -10$ and $u_1^+ = u_2^+ = u_3^+ = 10$. Figure 3 shows that the results synthesized by the proposed SNN-based controllers with different parameters p can guarantee that the state of the plant converges to the reference as expected, and the controller with larger parameter p also has fewer oscillation profiles. A typical variant or extended version of such an over-determined nonlinear affine system can be the differential kinematic models of redundant robots when the number of active joints is greater than that of motion degree of freedom in the end-effector level [51]. Such over-determined nonlinear systems can be fault tolerant, while some input(s) fail to be the correct solution, and one canonical example of application scenarios involves addressing the challenge of fault-tolerant motion control in redundant robots. This indicates that such over-determined nonlinear affine systems could be good alternatives for objective plants in sparsity-based control schemes.



(**a**) $p = 1.0$ (**b**) $p = 1.2$ (**c**) $p = 1.5$ (**d**) $p = 1.8$

(**e**) $p = 1.0$ (**f**) $p = 1.2$ (**g**) $p = 1.5$ (**h**) $p = 1.8$

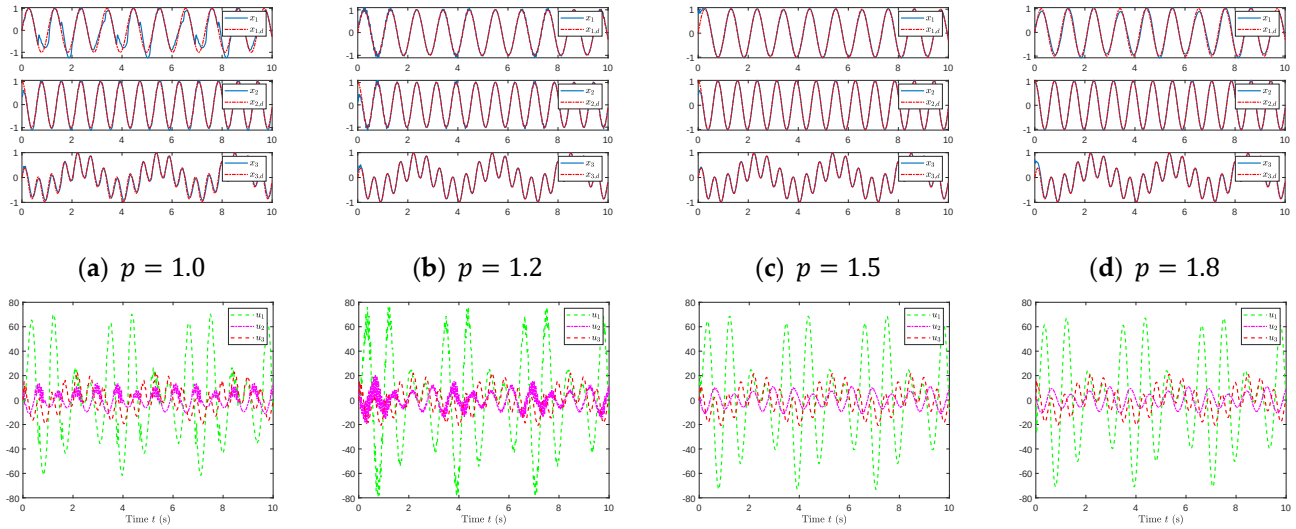
Figure 3. Convergence results of Equation (33) with the implemented SNN control approach: (**a**) $t = 10$ s; (**b**) $t = 10$ s; (**c**) $t = 10$ s; (**d**) $t = 10$ s; (**e**) $t = 10$ s; (**f**) $t = 10$ s; (**g**) $t = 10$ s; (**h**)$t = 10$ s.

C. Use Case 3: Lü's Attractor System

Consider the following famous Lü's attractor system [52] with three additional control input variables:

$$\begin{cases} \dot{x}_1 = \sigma x_2 - \sigma x_1 + u_1 \\ \dot{x}_2 = \rho x_1 - x_1 x_3 - x_2 + u_2 \\ \dot{x}_3 = x_1 x_2 - \beta x_3 + u_3 \end{cases} \tag{33}$$

with the parameters configured as $\sigma = 36$, $\rho = 3$ and $\beta = 20$. Such a nonlinear affine system is a chaotic dynamic system that can be reformulated in a nonlinear affine form. For the control task setting, the desired trajectories $x_d = [x_{1,d} \ x_{2,d} \ x_{3,d}]^T$ with $x_{1,d} = \sin 6t$, $x_{2,d} = \cos 8t$ and $x_{3,d} = \sin 5t \cos 7t$ were used for the state variable to track. The upper and lower bounds for the saturated inputs were specified as 80 and -80, respectively. Figure 4 shows the results synthesized by the proposed SNN-based controllers, and we can see that the state variables of the Lü's attractor system converge to the specified trajectory with input saturation satisfied. However, controllers with larger p exhibit small oscillations in the dynamic process. The control performance results in use cases 1 and 2, combined with those mentioned above, collectively validate the effectiveness and efficiency of the proposed SNN-based control method for tracking the states of nonlinear affine systems under control input saturation.



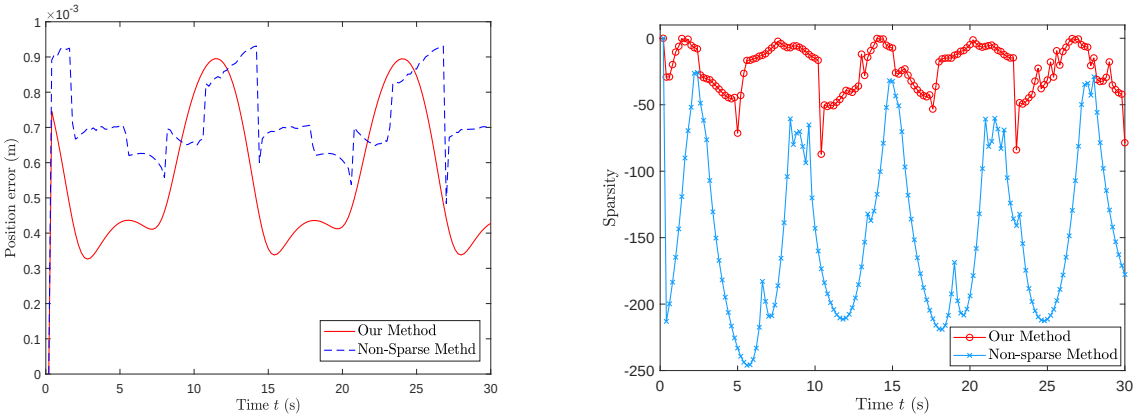
(**a**) $p = 1.0$ (**b**) $p = 1.2$ (**c**) $p = 1.5$ (**d**) $p = 1.8$

(**e**) $p = 1.0$ (**f**) $p = 1.2$ (**g**) $p = 1.5$ (**h**) $p = 1.8$

Figure 4. Convergence results of (34) synthesized by the proposed SNN-based control method: (**a**) $t = 10$ s; (**b**) $t = 10$ s; (**c**) $t = 10$ s; (**d**) $t = 10$ s; (**e**) $t = 10$ s; (**f**) $t = 10$ s; (**g**) $t = 10$ s; (**h**) $t = 10$ s.

D. Use Case 4: Manipulator Control Example

In this use case, the proposed SNN with $p = 1$ was further applied to the kinematic control of the manipulator, whose dynamics are expressed as a nonlinear affine system, i.e., $\dot{x} = J\dot{\theta}$ [53,54], where x represents the end-effector's position, J denotes the Jacobian matrix, and θ denotes the joint angle. A circular trajectory with a radius of 0.15 m is set as the specified path x_d for the end-effector to track. For comparative purposes, non-sparse neural network [55]-based control methods are adopted for the same tracking task with the same parameter $\kappa = 1000$ and the same initial joint angle $\theta(0)$. In Figure 5, we present a performance comparison between the SNN and the non-sparse neural network-based control method. Figure 5a shows that the tracking error magnitudes of the position by the two methods are within the same scale range of 10^{-3} m; however, the average sparsity $||\dot{x}||_l$ $(l = 0.5,\ 0.75,\ 1)$ of our proposed method is enhanced, as demonstrated by Figure 5b. The findings indicate that, at an equivalent level of control precision, the proposed method, SNN, attains more effective sparse solutions.



(**a**) Position error $||x - x_d||_2$ (**b**) Average sparsity $-||\dot{x}||_l$

Figure 5. Performance comparison between our method and non-sparse neural network-based control method: (**a**) $t = 30$ s; (**b**) $t = 30$ s.

E. Applications in real-world UR5 robotic control

In this example, our SNN method (with p = 1) was utilized to control the motion planning of a real UR5 robot. The joint coordinate system of the UR5 robot is defined, as depicted in Figure 6.

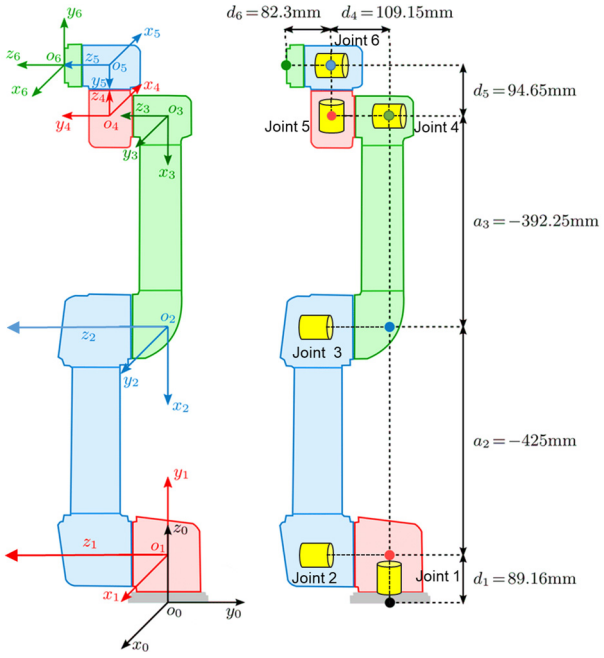


Figure 6. Diagram of UR5 robot joint coordinates.

We first define the position vector of the manipulator's end effector as $r(t) \in \boldsymbol{R^m}$ and the joint angle vector as $\theta(t) \in \boldsymbol{R}^n$. Then, the forward kinematics equation of the UR5 manipulator can be established as shown in Equation (34).

$$r(t) = f(\theta(t)) \tag{34}$$

where $f(\bullet) : \boldsymbol{R^n} \to \boldsymbol{R^m}$ is used to describe the forward kinematics. The UR5 is a 6-degree-of-freedom robotic arm that is controlled to operate in a three-dimensional task space. Therefore, the values of n and m are 6 and 3, respectively.

The inverse kinematics equation can be derived using Equation (34). By differentiating the inverse kinematics equation with respect to t, one can obtain:

$$J\dot{\theta}(t) = \dot{r}(t) \tag{35}$$

where $J = \partial f(\theta(t))/\partial\theta(t) \in \boldsymbol{R}^{\boldsymbol{m}\times\boldsymbol{n}}$ denotes the Jacobian matrix of the robotic arm. The vector $\dot{r}(t) = dr(t)/dt \in \boldsymbol{R}^{m}$ represents the velocity at the end of the robotic arm's trajectory, whereas $\dot{\theta}.(t) = d\theta(t)/dt \in \boldsymbol{R}^{n}$ represents the angular velocity of the robotic arm's joints. The D-H operating parameter information can be obtained from Table 1.

Table 1. D-H parameters of the UR5 robot.

Kinematics i	θ [rad]	a [m]	d [m]	α [rad]	Dynamics
1	θ_1	0	0.08916	$\pi/2$	Link 1
2	θ_2	-0.425	0	0	Link 2
3	θ_3	-0.39225	0	0	Link 3
4	θ_4	0	0.10915	$\pi/2$	Link 4
5	θ_5	0	0.09465	$-\pi/2$	Link 5
6	θ_6	0	0.0823	0	Link 6

Note: Table 1, which contains information on DH parameters, is credited to the Universal Robots official website at the following URL: https://www.universal-robots.com/articles/ur/application-installation/dh-parameters-for-calculations-of-kinematics-and-dynamics/ (accessed on 20 March 2024).

The D-H parameters are a set of four parameters used to describe the geometry and kinematics of a robotic manipulator. These parameters include the lengths of the links, the angles between the links, and the offsets along and around the axes of the joints. They provide a standardized framework for defining the relationship between consecutive rigid bodies in a robotic arm.

UR5 is a redundant manipulator, and its repetitive motion index can be expressed mathematically as: $s = \alpha(\theta(\mathrm{t}) - \theta(0))$, where $\alpha > 0$ is the scalar coefficient of the manipulator joint offset $\theta(\mathrm{t}) - \theta(0)$, $\theta(0)$ represents the starting state of the robot arm joint angle vector. The repetitive motion index can be represented as an optimization term in quadratic programming. To achieve feedback management of the robot's end position tracking inaccuracy, the feedback coefficient matrix K is introduced.

$$\begin{aligned} &\text{minimize} \quad \tfrac{1}{2}\|\dot{\theta}(t) + s\|_2^2 \\ &\text{subject to } J\dot{\theta}(t) = \dot{r}(t) + K(r(t) - f(\theta(t))) \end{aligned} \tag{36}$$

To guarantee that the generated joint angular and velocity values comply with the physical limit constraint, i.e.,

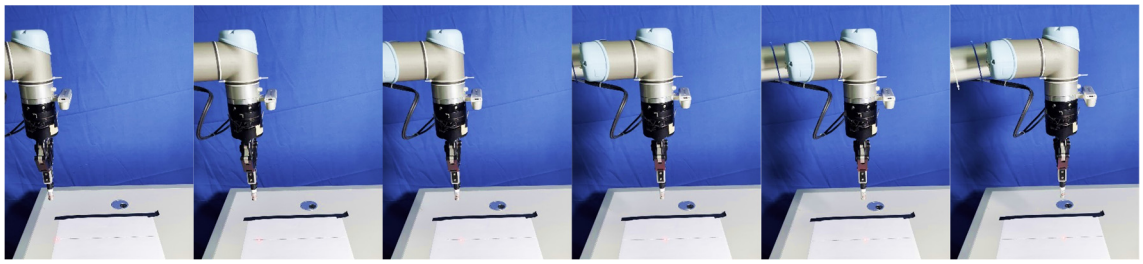
$$\begin{cases} \theta^- \le \theta \le \theta^+ \\ \dot{\theta}^- \le \dot{\theta} \le \dot{\theta}^+ \end{cases} \tag{37}$$

where θ^- and θ^+ are the joint angle θ's physical limit parameters.

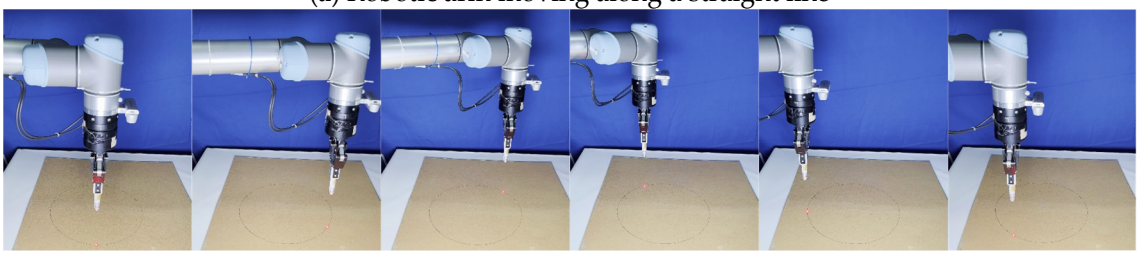
By utilizing the aforementioned approach, we can manipulate the UR5 robot to execute several types of trajectories, including linear motion control, circular trajectory motion control, and sinusoidal trajectory motion control.

(1) When guiding the robot to move in a straight line, we generate a random trajectory and direct the robot to follow this predetermined path. The initial position of the straight line trajectory in this experiment is $P_{start}(-0.07049, 0.350504, 0.22019)$, and the final position is $P_{end}(-0.07049,\ 0.75509,\ 0.22019)$. (2) We randomly select the center point $P_c(-0.07049, 0.487504, 0.22019)$ and set the radius as $r = 0.15$, when controlling the robot to move along a circular trajectory. (3) We randomly set the starting point $P_{\text{start}}(-0.07049, 0.487504, 0.22019)$, with an amplitude of 0.05 and a unit period of 0.1256, to drive the robot's end to travel along a sinusoidal trajectory. The end point of the sinusoidal trajectory is $P_{\text{end}}(-0.07049, 0.613104, 0.22019)$. The experiment imposed a maximum limit

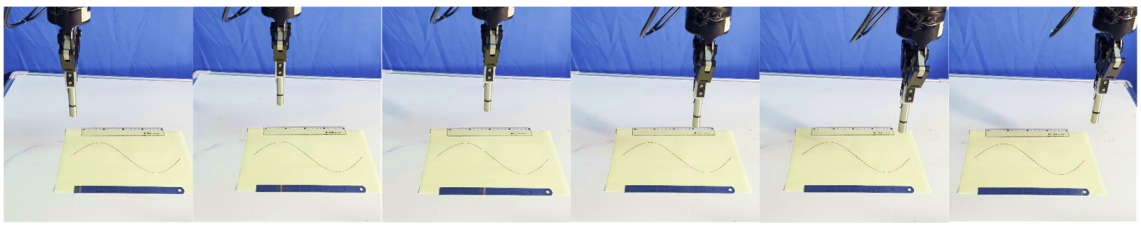
of 0.1 m/s for the robot's final velocity and 0.1 m/s^2 for its acceleration. Figure 7 shows the end movement of the robot.



(**a**) Robotic arm moving along a straight line



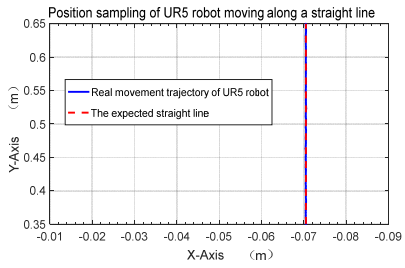
(**b**) Robotic arm moving in a circle



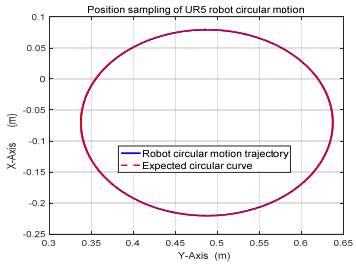
(**c**) Robotic arm moving along a sinusoidal curve

Figure 7. Applications in real-world UR5 robotic control.

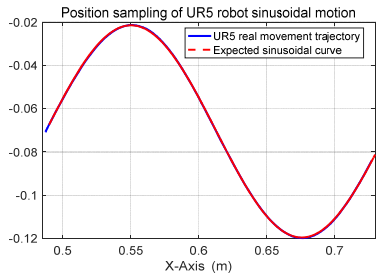
The robot end motion trajectory and trajectory error are shown in Figure 8.



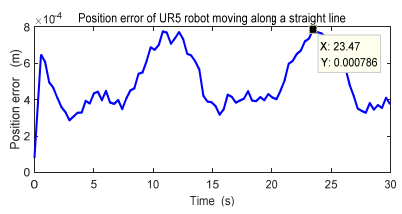
(**a**) The trajectory of linear motion



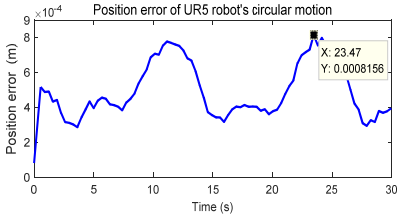
(**b**)The trajectory of circular motion



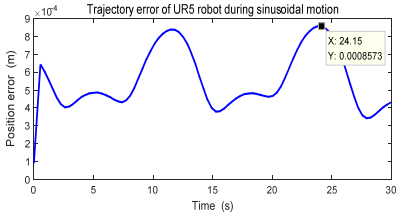
(**c**)The trajectory of sinusoidal motion



(**d**) Error in the linear trajectory



(**e**) Error in the circular trajectory



(**f**) Error in the sinusoidal trajectory

Figure 8. Position error in real-world UR5 robotic control: (**a**) the linear trajectory exhibits an average error of 5.76×10^{-4} m; (**b**) the circular trajectory demonstrates an average error of 5.98×10^{-4} m; (**c**) the sinusoidal trajectory showcases an average error of 6.15×10^{-4} m; (**d**) $t = 30$ s; (**e**) $t = 30$ s; (**f**) $t = 30$ s.

Figure 8a–f corresponds to each other and are consistent in their quantitative data representation. Based on the information provided in Figure 8a–f, it is evident that the redundant robot manipulator UR5 is able to accurately follow the desired path. Furthermore, the end effector consistently maintains the intended position throughout the task. Figure 8d–f shows the positional errors of the end effector. The linear trajectory exhibits an average error of 5.76×10^{-4} m, the circular trajectory demonstrates an average error of 5.98×10^{-4} m, and the sinusoidal trajectory showcases an average error of 6.15×10^{-4} m. They were maintained at a minimum magnitude of 10^{-4} m.

5. Conclusions

In this paper, an SNN-based control method from an l_p-norm$(1 \leq p < 2)$ optimization perspective is proposed for saturated nonlinear affine systems by taking advantage of the good properties of primal dual neural networks for optimization. We formulated optimal constraint conditions, converting the control issue of saturated nonlinear affine systems into an optimization problem. In particular, when $p = 1$, a new alternative controller based on SNN is derived, encountering computational difficulties distinct from those of another solution set in the basic dual neural network. An analysis is conducted on the theoretical findings about the control approach based on SNN. Additionally, four concrete instances (including over-determined nonlinear affine systems, the well-known Lü's attractor system, and the manipulator kinematics system) are provided to demonstrate the efficiency of our approach for the tracking control of saturated nonlinear affine systems. Furthermore, we evaluate the method's viability by applying it to real-world UR5 robot motion control and then quantitatively analyzing the control errors.

Our research primarily involved experimental validation using second-order nonlinear saturated affine systems, and the results have been promising. However, significant challenges remain for underactuated nonlinear systems and higher-order nonlinear drive control systems. We have identified that these systems require model transformation and reconstruction, along with the appropriate introduction of virtual control inputs, to effectively utilize SNN controllers. This insight underscores the need for improvements in current SNN control methods when dealing with complex nonlinear systems. Therefore, in future research, we will focus on developing customized solutions for underactuated systems and higher-order nonlinear underactuated systems. Specifically, we plan to optimize existing SNN control methods to better accommodate the unique characteristics of these complex systems, ensuring their effective operation across various nonlinear scenarios.

Author Contributions: Conceptualization, J.Z. and Z.L.; methodology, Z.L.; formal analysis, H.G. and Z.L.; investigation, J.H.; writing—original draft preparation, J.Z. and Z.L.; writing—review and editing, J.Z. and B.Y.; supervision, B.Y.; project administration, J.Z.; funding acquisition, J.Z. All authors have read and agreed to the published version of the manuscript.

Funding: The present research received financial support from the Sichuan Provincial Natural Science Youth Fund Project (Grant Number: 2023NSFSC1442) and the 2023 Sichuan Provincial Key Laboratory of Artificial Intelligence Open Fund Project (Grant Number: 2023RYY05).

Data Availability Statement: The article contains all the data used in this paper, and the experimental section provides an explanation. For further information, interested parties may contact the corresponding author of this paper for additional discussion and inquiries.

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

References

1. Zhi, H.; Chen, L.; Li, C.; Guo, Y. Leader–follower affine formation control of second-order nonlinear uncertain multiagent systems. *IEEE Trans. Circuits Syst. II Express Briefs* **2021**, *68*, 3547–3551.
2. Vaezi, M.; Izadian, A. Piecewise affine system identification of a hydraulic wind power transfer system. *IEEE Trans. Control Syst. Technol.* **2015**, *23*, 2077–2086. [CrossRef]
3. Xu, G.-H.; Qi, F.; Lai, Q.; Iu, H.H.-C. Fixed time synchronization control for bilateral teleoperation mobile manipulator with nonholonomic constraint and time delay. *IEEE Trans. Circuits Syst. II Express Briefs* **2020**, *67*, 3452–3456. [CrossRef]

4. Li, Z.; Li, S. A sparse optimization-based control method for manipulator with simultaneous potential energy minimization. *IEEE Trans. Circuits Syst. II Express Briefs* **2021**, *68*, 2062–2066. [CrossRef]
5. Li, P.; Kang, Y.; Zhao, Y.-B.; Wang, T. A novel self-triggered MPC scheme for constrained input-affine nonlinear systems. *IEEE Trans. Circuits Syst. II Express Briefs* **2021**, *68*, 306–310. [CrossRef]
6. Zhao, C.; Guo, L. Control of nonlinear uncertain systems by extended PID. *IEEE Trans. Autom. Control* **2021**, *66*, 3840–3847. [CrossRef]
7. Li, Z.; Zhao, J. Almost output regulation of switched affine systems and its application to a circuit model. *IEEE Trans. Circuits Syst. II Express Briefs* **2021**, *68*, 3256–3260. [CrossRef]
8. Zhou, Q.; Shi, P.; Tian, Y.; Wang, M. Approximation-based adaptive tracking control for MIMO nonlinear systems with input saturation. *IEEE Trans. Cybern.* **2015**, *45*, 2119–2128. [CrossRef] [PubMed]
9. Sun, L.; Wang, Y.; Feng, G. Control design for a class of affine nonlinear descriptor systems with actuator saturation. *IEEE Trans. Autom. Control* **2015**, *60*, 2195–2200. [CrossRef]
10. Zhu, G.; Du, J. Global robust adaptive trajectory tracking control for surface ships under input saturation. *IEEE J. Ocean. Eng.* **2018**, *45*, 442–450. [CrossRef]
11. Albea, C.; Sferlazza, A.; Gordillo, F.; Gomez-Estern, F. Control of power converters with hybrid affine models and pulse-width modulated inputs. *IEEE Trans. Circuits Syst. I Regul. Pap.* **2021**, *68*, 3485–3494. [CrossRef]
12. Huang, Y.; Jia, Y. Adaptive finite-time 6-dof tracking control for spacecraft fly around with input saturation and state constraints. *IEEE Trans. Aerosp. Electron. Syst.* **2019**, *55*, 3259–3272. [CrossRef]
13. Su, Y.; Shen, S. Adaptive global prescribed performance control for rigid spacecraft subject to angular velocity constraints and input saturation. *Nonlinear Dyn.* **2023**, *111*, 21691–21705. [CrossRef]
14. He, W.; Ge, S.S. Vibration control of a flexible string with both boundary input and output constraints. *IEEE Trans. Control Syst. Technol.* **2015**, *23*, 1245–1254. [CrossRef]
15. Brunton, S.L.; Proctor, J.L.; Kutz, J.N. Sparse identification of nonlinear dynamics with control (SINDYc). *IFAC-PapersOnLine* **2016**, *49*, 710–715. [CrossRef]
16. Yoo, J.; Shin, J.; Park, P. An improved NLMS algorithm in sparse systems against noisy input signals. *IEEE Trans. Circuits Syst. II Express Briefs* **2015**, *62*, 271–275. [CrossRef]
17. Selesnick, I. Sparse Regularization via Convex Analysis. *IEEE Trans. Signal Process.* **2017**, *65*, 4481–4494. [CrossRef]
18. Babazadeh, M. Regularization for Optimal Sparse Control Structures: A Primal-Dual Framework. In Proceedings of the 2021 American Control Conference (ACC), New Orleans, LA, USA, 25–28 May 2021; pp. 3850–3855.
19. Torres, J.A.; Roy, S.; Rausch, S. Sparse linear and nonlinear controls for network spread processes. In Proceedings of the 2016 American Control Conference (ACC), Boston, MA, USA, 6–8 July 2016; pp. 3372–3377.
20. Wensing, P.M.; Slotine, J.-J. Sparse control for dynamic movement primitives. *IFAC-PapersOnLine* **2017**, *50*, 10114–10121. [CrossRef]
21. Kaheman, K.; Kutz, J.N.; Brunton, S.L. SINDy-PI: A robust algorithm for parallel implicit sparse identification of nonlinear dynamics. *Proc. R. Soc. A* **2020**, *476*, 20200279. [CrossRef]
22. Zhao, X.; Liu, Z.; Zhu, Q. Neural network-based adaptive controller design for robotic manipulator subject to varying loads and unknown dead-zone. *Neurocomputing* **2023**, *546*, 126293. [CrossRef]
23. Meng, W.; Yang, Q.; Jagannathan, S.; Sun, Y. Adaptive neural control of high-order uncertain nonaffine systems: A transformation to affine systems approach. *Automatica* **2014**, *50*, 1473–1480. [CrossRef]
24. Lin, H.; Zhao, B.; Liu, D.; Alippi, C. Data-based fault tolerant control for affine nonlinear systems through particle swarm optimized neural networks. *IEEE/CAA J. Autom. Sin.* **2020**, *7*, 954–964. [CrossRef]
25. Kim, J.W.; Park, B.J.; Yoo, H.; Oh, T.H.; Lee, J.H.; Lee, J.M. A model-based deep reinforcement learning method applied to finite-horizon optimal control of nonlinear control-affine system. *J. Process Control* **2020**, *87*, 166–178. [CrossRef]
26. Li, J.; Ding, J.; Chai, T.; Lewis, F.L.; Jagannathan, S. Adaptive interleaved reinforcement learning: Robust stability of affine nonlinear systems with unknown uncertainty. *IEEE Trans. Neural Netw. Learn. Syst.* **2020**, *33*, 270–280. [CrossRef] [PubMed]
27. Louizos, C.; Welling, M.; Kingma, D.P. Learning sparse neural networks through L_0 regularization. *arXiv* **2017**, arXiv:1712.01312.
28. Srinivas, S.; Subramanya, A.; Venkatesh Babu, R. Training sparse neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, Honolulu, HI, USA, 21–26 July 2017; pp. 138–145.
29. Tang, Z.; Luo, L.; Xie, B.; Zhu, Y.; Zhao, R.; Bi, L.; Lu, C. Automatic Sparse Connectivity Learning for Neural Networks. *IEEE Trans. Neural Netw. Learn. Syst.* **2023**, *34*, 7350–7364. [CrossRef] [PubMed]
30. Reiners, M.; Klamroth, K.; Heldmann, F.; Stiglmayr, M. Efficient and sparse neural networks by pruning weights in a multiobjective learning approach. *Comput. Oper. Res.* **2022**, *141*, 105676. [CrossRef]
31. Tian, Y.; Lu, C.; Zhang, X.; Tan, K.C.; Jin, Y. Solving large-scale multiobjective optimization problems with sparse optimal solutions via unsupervised neural networks. *IEEE Trans. Cybern.* **2020**, *51*, 3115–3128. [CrossRef]
32. Zhu, M.; Zhang, T.; Gu, Z.; Xie, Y. Sparse tensor core: Algorithm and hardware co-design for vector-wise sparse neural networks on modern gpus. In Proceedings of the 52nd Annual IEEE/ACM International Symposium on Microarchitecture, Columbus, OH, USA, 12–16 October 2019; pp. 359–371.
33. Nivison, S.A.; Khargonekar, P. A sparse neural network approach to model reference adaptive control with hypersonic flight applications. In Proceedings of the 2018 AIAA Guidance, Navigation, and Control Conference, Kissimmee, FL, USA, 8–12 January 2018; pp. 1–25.

34. Ait Abbas, H. A new adaptive deep neural network controller based on sparse auto-encoder for the antilock bracking system systems subject to high constraints. *Asian J. Control* **2021**, *23*, 2145–2156. [CrossRef]
35. Li, Z.; Li, S. An l1-norm based optimization method for sparse redundancy resolution of robotic manipulators. *IEEE Trans. Circuits Syst. II Express Briefs* **2021**, *69*, 469–473. [CrossRef]
36. Li, Z.; Li, C.; Li, S.; Zhu, S.; Samani, H. A sparsity-based method for fault-tolerant manipulation of a redundant robot. *Robotica* **2022**, *40*, 3396–3414. [CrossRef]
37. Zhang, Z.; Fu, T.; Yan, Z.; Jin, L.; Xiao, L.; Sun, Y.; Yu, Z.; Li, Y. A Varying-Parameter Convergent-Differential Neural Network for Solving Joint-Angular-Drift Problems of Redundant Robot Manipulators. *IEEE/ASME Trans. Mechatron.* **2018**, *23*, 679–689. [CrossRef]
38. Zhao, T.; Liu, Y.; Li, Z.; Su, C.-Y.; Feng, Y. Adaptive Control and Optimization of Mobile Manipulation Subject to Input Saturation and Switching Constraints. *IEEE Trans. Autom. Sci. Eng.* **2019**, *16*, 1543–1555. [CrossRef]
39. Bilal, D.K.; Unel, M.; Tunc, L.T.; Gonul, B. Development of a vision based pose estimation system for robotic machining and improving its accuracy using LSTM neural networks and sparse regression. *Robot. Comput.-Integr. Manuf.* **2022**, *74*, 102262. [CrossRef]
40. Shukla, P.; Kushwaha, V.; Nandi, G.C. Vision-Based Intelligent Robot Grasping Using Sparse Neural Network. *arXiv* **2023**, arXiv:2308.11590.
41. Sayar, E.; Gao, X.; Hu, Y.; Chen, G.; Knoll, A. Toward coordinated planning and hierarchical optimization control for highly redundant mobile manipulator. *ISA Trans.* **2024**, *146*, 16–28. [CrossRef]
42. Pan, M.; Li, J.; Yang, Q.; Wang, Y.; Tang, Y.; Pan, L.; Jiang, X.; Lin, Y.; Liang, K. An adaptive sparse general regression neural network-based force observer for teleoperation system. *Eng. Appl. Artif. Intell.* **2023**, *118*, 105689. [CrossRef]
43. Zhao, L.; Sun, Z.; Liu, K.; Zhang, J. The dynamic relaxation form finding method aided with advanced recurrent neural network. *CAAI Trans. Intell. Technol.* **2023**, *8*, 635–644. [CrossRef]
44. Li, Z.; Li, S. Neural network model-based control for manipulator: An autoencoder perspective. *IEEE Trans. Neural Netw. Learn. Syst.* **2023**, *8*, 622–634. [CrossRef]
45. Goberna, M.A.; Guerra-Vazquez, F.; Todorov, M.I. Constraint qualifications in convex vector semi-infinite optimization. *Eur. J. Oper. Res.* **2016**, *249*, 32–40. [CrossRef]
46. Chieu, N.H.; Jeyakumar, V.; Li, G.; Mohebi, H. Constraint qualifications for convex optimization without convexity of constraints: New connections and applications to best approximation. *Eur. J. Oper. Res.* **2018**, *265*, 19–25. [CrossRef]
47. Haidar, I.; Chitour, Y.; Mason, P.; Sigalotti, M. Lyapunov characterization of uniform exponential stability for nonlinear infinite-dimensional systems. *IEEE Trans. Autom. Control* **2021**, *67*, 1685–1697. [CrossRef]
48. Tang, C.P.; Miller, P.T.; Krovi, V.N.; Ryu, J.C.; Agrawal, S.K. Differential-flatness-based planning and control of a wheeled mobile manipulator—Theory and experiment. *IEEE/ASME Trans. Mechatron.* **2010**, *16*, 768–773. [CrossRef]
49. Yuan, W.; Liu, Y.; Liu, Y.H.; Su, C.Y. Differential flatness-based adaptive robust tracking control for wheeled mobile robots with slippage disturbances. *ISA Trans.* **2024**, *144*, 482–489. [CrossRef]
50. Shen, H.; Iorio, J.; Li, N. Sliding mode control in backstepping framework for a class of nonlinear systems. *J. Mar. Sci. Eng.* **2019**, *7*, 452. [CrossRef]
51. Li, Z.; Li, S. Saturated pi control for nonlinear system with provable convergence: An optimization perspective. *IEEE Trans. Circuits Syst. II Express Briefs* **2021**, *68*, 742–746. [CrossRef]
52. Munteanu, F. Analyzing the Jacobi Stability of Lü's Circuit System. *Symmetry* **2022**, *14*, 1248. [CrossRef]
53. Li, Z.; Li, S. Recursive recurrent neural network: A novel model for manipulator control with different levels of physical constraints. *CAAI Trans. Intell. Technol.* **2023**, *8*, 622–634. [CrossRef]
54. Liu, Y.; Li, T.; Duan, J.; Wu, X.; Wang, H.; Fan, Q.; Lin, J.; Hu, Y. On a hierarchical adaptive and robust inverse dynamic control strategy with experiment for robot manipulators under uncertainties. *Control Eng. Pract.* **2023**, *138*, 105604. [CrossRef]
55. Yang, M.; Zhang, Y.; Zhou, X.; Hu, H. Pose control of constrained redundant arm using recurrent neural networks and one-iteration computing algorithm. *Appl. Soft Comput.* **2021**, *113*, 108007. [CrossRef]