

Simultaneous identification, tracking control and disturbance rejection of uncertain nonlinear dynamics systems: a unified neural approach[☆]

Dechao Chen¹, Shuai Li^{2,*}, Qing Wu^{1,**}, Liefu Liao³

¹*School of Computer Science and Technology, Hangzhou Dianzi University, Hangzhou 310018, P.R. China*

²*School of Engineering, Swansea University, Swansea, United Kingdom, postcode SA1 7EN*

³*School of Information Engineering, Jiangxi University of Science and Technology, Ganzhou 341000, P.R. China*

Abstract

Previous works of traditional zeroing neural networks (or termed Zhang neural networks, ZNN) show great success for solving specific time-variant problems of known systems in an ideal environment. However, it is still a challenging issue for the ZNN to effectively solve time-variant problems for uncertain systems without the prior knowledge. Simultaneously, the involvement of external disturbances in the neural network model makes it even hard for time-variant problem solving due to the intensively computational burden and low accuracy. In this paper, a unified neural approach of simultaneous identification, tracking control and disturbance rejection in the framework of the ZNN is proposed to address the time-variant tracking control of uncertain nonlinear dynamics systems (UNDS). The neural network model derived by

[☆]This work is supported by the National Natural Science Foundation of China (with numbers 61906054, 61401385 and 61702146), by Hong Kong Research Grants Council Early Career Scheme (with number 25214015), by Departmental General Research Fund of Hong Kong Polytechnic University (with number G.61.37.UA7L), and also by PolyU Central Research Grant (with number G-YBMU).

*Corresponding author.

E-mail address: shuaili@ieee.org (S. Li).

**Corresponding author.

E-mail address: wuqing@hdu.edu.cn (Q. Wu).

the proposed approach captures hidden relations between inputs and outputs of the UNDS. The proposed model shows outstanding tracking performance even under the influences of uncertainties and disturbances. Then, the continuous-time model is discretized via Euler forward formula (EFF). The corresponding discrete algorithm and block diagram are also presented for the convenience of implementation. Theoretical analyses on the convergence property and discretization accuracy are presented to verify the performance of the neural network model. Finally, numerical studies, robot applications, performance comparisons and tests demonstrate the effectiveness and advantages of the proposed neural network model for the time-variant tracking control of UNDS.

Keywords: Zhang neural networks (ZNN), Time-variant tracking control, Time-variant problems, Robustness, Identification
2010 MSC: 62G35, 92B20, 93B51

1. Introduction

Many practical systems in real world are nonlinear dynamics systems (NDS) [1], such as servo motor systems [2], autonomous underwater vehicles systems [3], and robot manipulator systems [4, 5]. It is well known that finding an effective solution to the NDS is a challenging issue due to their high nonlinearity and complicated dynamics nature [6]. Therefore, the investigation of NDS, including the time-variant problem solving of NDS, has been a hot scientific topic in recent years for the wide applications [7, 8]. For examples, Fu *et al.* [9] developed a new optimal control approach with an observer-critic structure for the nonlinear-singularly-perturbed system with unknown dynamics as well as input constraints. In [10], the authors obtained a general result that there exist global finite-time observers for an uniformly observable as well as globally Lipschitzian single output nonlinear system. Na *et al.* [11] introduced an effective adaptive control framework for the nonaffine pure feedback nonlinear systems with unknown dynamics. Such an alternative adaptive control framework in [11] avoided the use of the back-stepping control for pure feedback systems to address the explosion of complexity. Hu *et al.* [12] introduced a novel adaptive visual servo tracking control method to complete the asymptotic tracking of the design trajectory. Li *et al.* [13] developed a new model predictive control strategy combining the neural dynamic optimization for tracking control of mobile robots, which is a

typical kind of nonlinear affine system. In [14], a novel real-time adaptive approximate approach was investigated to handle the optimal tracking control of an uncertain nonlinear dynamics system (UNDS). Chen *et al.* [15] proposed a new control strategy on the basis of uncertainties-and-disturbances estimator. The control strategy mitigates the effect of hysteresis nonlinearity and also improves the performance of the tracking control.

Recent studies in [16, 17] show that recurrent neural networks (RNN) provide alternative solutions to the UNDS [18]. Hopfield and Tank [19] firstly proposed a new RNN for handling the quadratic optimization in 1985. This is an early work of the RNN. The corresponding RNN was also implemented on analog circuits in [20]. After that, numerous RNN have been introduced and investigated for handling various time-variant problems since such a seminal work [21, 22]. For examples, Xia *et al.* [23] proposed an RNN to handle the quadratic programming as well as the linear-piecewise equations. Shen [24] investigated the asymptotic stability issue for delayed RNN considering time-variant delays. Liu and Wang [25] developed an effective RNN possessing only one layer with the discontinuous activation function for the linear program. Zhang *et al.* [26] presented an RNN for handling Sylvester equation possessing the time-variant coefficient matrix. In addition, a novel RNN was developed for time-variant redundancy-resolution of redundant robot arms with joint angle constraints and velocity constraints in [27]. Being derived from the RNN, Zhang neural networks (ZNN) raised by Zhang *et al.* [28] possess the ability to handle problems with multiple-state dimensions. Those RNNs are able to zero out each element of the designed error function in terms of neural network models [28]. By exploiting a vector-valued or matrix-valued error function together with the time derivative coefficients of the involved systems, the related ZNN ensure the exponential convergence property to the solution [29, 30]. The ZNN design method has been deemed as an effective and systematic scheme for dealing with different time-variant problems [31, 32] (including time-variant tracking control problems) [33, 34]. For examples, Cai and Zhang [35] addressed the inverse kinematic issue of redundant robot arms by using the gradient neural network (GNN) as well as ZNN. In addition, Zhang *et al.* [36] proposed an effective discrete time ZNN model to handle time-variant quadratic minimization. For accelerating ZNN models to finite time convergence, Xiao [37] proposed an effective design formula. This new formula makes a breakthrough in the research of convergence performance of the ZNN. For the discretization and application of the ZNN, Guo *et al.* [38] newly introduced a discrete time ZNN (DTZNN) to

handle time-variant matrix inversion with application to the motion tracking of robot arms.

Despite the great success of traditional ZNN in time-variant problem solving, finding an effective solution to the UNDS is still a challenging issue. Because of the involvements of system uncertainties and external disturbances may heavily destroy the convergence performance, and also lead to heavy computational overhead [39]. Therefore, very few breakthroughs have been made on the robustness research on the ZNN concerning both the system uncertainties and external disturbances at the same time. It is also worth pointing out here that the model designed in a unified neural approach possesses the advantages compared with those via quadratic program based approaches as the following facts. Firstly, a neural network model has an explicit analytical solution. In contrast, the quadratic program based approaches may possess only the optimal solution, which should be found by an optimization solver [40]. Secondly, it is readily and directly to analyze the convergence performance of a neural network model since it has the explicit analytical solution. On the contrary, the theoretical analyses of the quadratic program based approaches are more complicated, which require the theoretical basis such as the duality theory and Karush-Kuhn-Tucker condition [41]. Last but not least, the optimization solution of the quadratic program based approaches may possess the high order nonlinearity [42]. This would make the tracking performance of the involved systems unstable in complex applications while a neural network model with an explicit analytical solution would possess more stable tracking performance.

The exploration of input-output response of the UNDS, early attempt was made by using identification approaches [43, 44]. Motivated by the inspiring works [43, 44], in this paper, a unified neural approach of simultaneous identification, tracking control and disturbance rejection in the framework of the ZNN is developed for designing neural network model to address the time-variant tracking control of UNDS. Note that the proposed neural network model is a typical kind of Hopfield neural network [26], which does not need offline learning in advance. In addition, the model designed in a unified neural approach can be readily implemented on circuits such as very large-scale integration [20]. With the superiorities of parallel processing distributed feature, high performances in large-scale real-time applications, as well as the convenience of hardware implementations, the dynamic model designed in a unified neural approach with the form of RNN has been focused intensively for time-variant linear or nonlinear dynamic systems handling [1, 45]. Such

a unified neural approach is now regarded as a powerful alternative for real-time computation as well as optimization. As for this specific work, to make progresses along the direction of neural network models for system control in complex applications, a novel neural network model is developed. Such a neural network model derived by the proposed approach captures hidden relations between inputs and outputs of the UNDS, and shows outstanding tracking performance even under the influence of system uncertainties and external disturbances, simultaneously. To the best of the authors' knowledge, the proposed neural network model designed by the novel approach that can elegantly and simultaneously handle the system uncertainties and external disturbances for the UNDS has not been reported and investigated in the existing literature.

The rest of the paper is structured as below. Section 2 presents preliminaries on time-variant tracking control of UNDS. The neural network model is designed by the proposed approach together with theoretical analyses presented in Section 3. In Section 4, simulation studies including two robot applications together with performance comparisons and tests are shown. Section 5 concludes the paper. The main contributions of the work are summarized as below.

- Unlike existing ZNN works for solving specific time-variant problems of known systems in an ideal environment, a unified neural approach of simultaneous identification, tracking control and disturbance rejection is proposed to address the time-variant tracking control of UNDS. This paper makes new progresses on the frontier of the robustness research on the ZNN by concerning both the system uncertainties and model disturbances, simultaneously.
- The proposed approach provides a complete and universal neural network model design process for the general UNDS. The neural network model derived by the proposed approach captures hidden relations between inputs and outputs of UNDS, and shows outstanding tracking performance even under system uncertainties and external disturbances.
- Theoretical analyses on convergence property and discretization accuracy are provided to guarantee the validity of the proposed neural network model for the time-variant tracking control of UNDS.

- Simulation studies including two practical applications, i.e., applications to serial robot manipulators and parallel robot manipulators, together with performance comparisons and tests substantiate the efficacy of the proposed neural network model.

2. Preliminaries

Consider a general form of continuous-time NDS described as follows [43]:

$$\begin{cases} \dot{\mathbf{x}}(t) = \mathbf{s}(\mathbf{x}(t), \mathbf{u}(t)), \\ \mathbf{y}(t) = \mathbf{h}(\mathbf{x}(t)), \end{cases} \quad (1)$$

where $\mathbf{x}(t) \in \mathbb{R}^n$, $\mathbf{u}(t) \in \mathbb{R}^m$, and $\mathbf{y}(t) \in \mathbb{R}^m$ are the system state vector, input vector, and output vector of NDS, respectively. In addition, functions $\mathbf{s} : \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}^n$ and $\mathbf{h} : \mathbb{R}^n \rightarrow \mathbb{R}^m$ are continuous and smooth nonlinear mappings.

Note that system (1) investigated in this work is a common form of continuous-time NDS [43]. System (1) can be deemed as a general description in mathematic. As for a specific application for such an NDS, such as the time-variant tracking control, NDS (1) can be reformulated with practical application meanings. The realization of time-variant tracking control of NDS (1) in practical applications is to obtain a neural network model with the prior knowledge from the real system. By constructing a dynamic mapping $\mathbf{u}(t) \rightarrow \mathbf{y}^*(t)$, it can identify the system dynamics of NDS (1) by using only the input-output measurements and minimizing $\|\mathbf{y}^*(t) - \mathbf{y}(t)\|_2$, where $\mathbf{y}^*(t)$ is the desired output vector of the NDS, and $\|\cdot\|_2$ is Euclidean norm of a vector [44]. With the above practical application objective in mind, a class of NDS is transformable to the following optimization problem:

$$\min \quad \|\mathbf{y}^*(t) - \mathbf{y}(t)\|_2, \quad (2)$$

$$\text{s.t.} \quad \dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t))\dot{\mathbf{y}}(t), \quad (3)$$

$$\mathbf{y}(t) = \mathbf{h}(\mathbf{x}(t)). \quad (4)$$

Note that mapping $\mathbf{f} : \mathbb{R}^n \rightarrow \mathbb{R}^{n \times m}$ is a nonlinear function corresponding to the system information, and mapping $\mathbf{h} : \mathbb{R}^n \rightarrow \mathbb{R}^m$ satisfies $\mathbf{f}(\partial\mathbf{h}/\partial\mathbf{x}) = I$ and is corresponding to a specific tracking effector with a certain physical structure and parameter to receive control signals and generate the real-time output trajectory $\mathbf{y}(t)$, which is fixed and explicit during the hold tracking process.

Under the influence unexpected impacts, e.g., the lack of prior knowledge from the real system, nonlinear mapping $\mathbf{f}(\cdot)$ would become an unknown mapping due to the system uncertainties. In addition, external disturbances usually occur during the tracking process of a given system in practical applications. Therefore, a practical form of time-variant tracking control of UNDS with external disturbances is described as follows:

$$\min \quad \|\mathbf{y}^*(t) - \mathbf{y}(t)\|_2, \quad (5)$$

$$\text{s.t.} \quad \dot{\mathbf{x}}(t) = \tilde{\mathbf{f}}(\mathbf{x}(t))(\dot{\mathbf{y}}(t) + \sigma(t)), \quad (6)$$

$$\mathbf{y}(t) = \mathbf{h}(\mathbf{x}(t)), \quad (7)$$

where $\tilde{\mathbf{f}} : \mathbb{R}^n \rightarrow \mathbb{R}^{n \times m}$ is the unknown mapping, and vector $\sigma(t) \in \mathbb{R}^m$ denotes the external disturbances for the UNDS. In practical applications, the objective of time-variant tracking control of UNDS (5)–(7) is to design an effective control law such that the output $\mathbf{y}(t)$ tracks a given path $\mathbf{y}^*(t)$. According to [39, 43] as well as to lay a basis for further investigation, two assumptions are presented.

Assumption 1: The real-time system state vector $\mathbf{x}(t)$, tracking output trajectory $\mathbf{y}(t)$ and its first order time derivative $\dot{\mathbf{y}}(t)$, second order time derivative $\ddot{\mathbf{y}}(t)$ are measurable during the whole tracking process. In addition, unknown nonlinear mapping $\tilde{\mathbf{f}}(\cdot)$ is continuously differentiable with respect to time t .

Assumption 2: The external disturbances $\sigma(t)$ are the continuous and bounded random-form time-variant disturbances during the whole tracking process $t \in [0, T_d]$ with T_d being the task duration. Time-variant disturbances could be considered to vanish after the tracking task duration, i.e., $\sigma(t) = 0$ with $t > T_d$.

Remark 1: To handle the system uncertainties of the UNDS (5)–(7) in this work, a continuously differentiable nonlinear mapping $\tilde{\mathbf{f}}(\cdot)$ is required to calculate $\tilde{\mathbf{f}}(\mathbf{x}(t))$ for updating the nonlinear mapping $\tilde{\mathbf{f}}(\cdot)$ in real-time t . It is required to make an assumption that the nonlinear mapping $\tilde{\mathbf{f}}(\cdot)$ is continuously differentiable to formulate the proposed neural network model. It is worth mentioning here that such an assumption is commonly applied in practice such as the tracking control of serial and parallel robot manipulators [46, 47]. As for the special case that the nonlinear mapping $\tilde{\mathbf{f}}(\cdot)$ is discontinuous, the real-time $\tilde{\mathbf{f}}(\mathbf{x}(t))$ would not exist in the discontinuous point. Therefore, the whole neural network model would not be designed via the proposed unified neural approach in this case.

Remark 2: Due to the complexity of the tracking process, the system uncertainties and external disturbances are unavoidable for the dynamics systems in real-world applications. Many tracking control processes of industrial systems can be formulated as the UNDS (5)–(7), such as aircraft systems [2], underwater vehicle systems [3], and robot systems [46], which covers many common nonlinear systems [48].

3. Neural network model design

As preliminaries, the tracking control problem formulation of the UNDS is shown in Section 2. In this part, we propose a unified neural approach of simultaneous identification, tracking control and disturbance rejection and its associated neural network model to address time-variant tracking control of UNDS. Then, the continuous-time neural network model is discretized together with the algorithm description and block diagram shown for the convenience of implementation. Moreover, theoretical analyses are presented in detail to prove the effectiveness of the proposed approach and neural network model.

3.1. Tracking control

To solve the time-variant tracking control problem of UNDS (5)–(7) under the influence external disturbances, a continuous-time neural network model is designed by the following approach.

Firstly, to monitor the tracking process of the UNDS (5)–(7), an error function can be defined as below:

$$\mathbf{e}(t) = \mathbf{y}^*(t) - \mathbf{y}(t), \quad (8)$$

where $\mathbf{y}^*(t) \in \mathbb{R}^m$ is the predefined path for the UNDS effector. Then, a ZNN-based error function can be defined as below:

$$\mathbf{z}_1(t) = \int_0^t \mathbf{e}(\tau) d\tau. \quad (9)$$

To make each element $e_i(t)$ of the error function (8) converge to zero for $i = 1, 2, \dots, m$, by exploiting unified ZNN design formula $\dot{\mathbf{z}}_1(t) = -\gamma \mathbf{z}_1(t)$ with $\gamma \in \mathbb{R}$ denoting a predefined parameter to adjust the rate of convergence [26], one can obtain:

$$\mathbf{e}(t) = -\gamma \int_0^t \mathbf{e}(\tau) d\tau. \quad (10)$$

Define a second ZNN-based error function:

$$\mathbf{z}_2(t) = -\mathbf{e}(t) - \gamma \int_0^t \mathbf{e}(\tau) d\tau. \quad (11)$$

Substituting (11) into the ZNN design formula $\dot{\mathbf{z}}_2(t) = -\lambda \mathbf{z}_2(t)$ with $\lambda \in \mathbb{R}$ being a predefined parameter to adjust the convergence rate, it has the following dynamical equation:

$$\dot{\mathbf{e}}(t) = -(\gamma + \lambda)\mathbf{e}(t) - \gamma\lambda \int_0^t \mathbf{e}(\tau) d\tau. \quad (12)$$

Predefined design parameters $\gamma \in \mathbb{R}^+$ and $\lambda \in \mathbb{R}^+$ are set for the convergence as well as stability of the proposed neural network model. By substituting error function (8) into dynamical equation (12), the following equation is attained:

$$\begin{aligned} \dot{\mathbf{y}}^*(t) - \dot{\mathbf{y}}(t) = & -(\gamma + \lambda)(\mathbf{y}^*(t) - \mathbf{y}(t)) \\ & - \gamma\lambda \int_0^t \mathbf{y}^*(\tau) - \mathbf{y}(\tau) d\tau. \end{aligned} \quad (13)$$

On the basis of equation (6) of UNDS, one can further obtain:

$$\begin{aligned} \dot{\mathbf{x}}(t) = & \tilde{\mathbf{f}}(\mathbf{x}(t)) \left(\dot{\mathbf{y}}^*(t) + \rho_1(\mathbf{y}^*(t) - \mathbf{y}(t)) \right. \\ & \left. + \rho_2 \int_0^t \mathbf{y}^*(\tau) - \mathbf{y}(\tau) d\tau + \sigma(t) \right) \end{aligned} \quad (14)$$

with $\gamma + \lambda = \rho_1 \in \mathbb{R}^+$ and $\gamma\lambda = \rho_2 \in \mathbb{R}^+$.

3.2. Identification

For the real-time identification of the unknown mapping $\tilde{\mathbf{f}}(\mathbf{x}(t))$ in (14), a vector-valued error function can be defined:

$$\boldsymbol{\epsilon}(t) = \dot{\mathbf{x}}(t) - \tilde{\mathbf{f}}(\mathbf{x}(t))\dot{\mathbf{y}}(t) \quad (15)$$

with vector $\boldsymbol{\epsilon}(t) \in \mathbb{R}^m$. By utilizing the ZNN design formula again:

$$\dot{\boldsymbol{\epsilon}}(t) = -\nu\boldsymbol{\epsilon}(t), \quad (16)$$

where $\nu \in \mathbb{R}^+$ is another predefined parameter to adjust the rate of convergence for identification, the following dynamical equation for the real-time identification of UNDS is obtained:

$$\ddot{\mathbf{x}}(t) - \dot{\tilde{\mathbf{f}}}(\mathbf{x}(t))\dot{\mathbf{y}}(t) - \tilde{\mathbf{f}}(\mathbf{x}(t))\ddot{\mathbf{y}}(t) = -\nu(\dot{\mathbf{x}}(t) - \tilde{\mathbf{f}}(\mathbf{x}(t))\dot{\mathbf{y}}(t)). \quad (17)$$

Equation (17) can be explicitly rewritten as

$$\dot{\tilde{\mathbf{f}}}(\mathbf{x}(t)) = (\ddot{\mathbf{x}}(t) - \tilde{\mathbf{f}}(\mathbf{x}(t))\ddot{\mathbf{y}}(t) + \nu(\dot{\mathbf{x}}(t) - \tilde{\mathbf{f}}(\mathbf{x}(t))\dot{\mathbf{y}}(t)))\dot{\mathbf{y}}^\dagger(t), \quad (18)$$

where superscript \dagger denotes the pseudo-inverse of a vector or matrix. Note that equation (18) is for real-time identification of UNDS (5)–(7). Hence, the whole neural network model is attained as follows:

$$\left\{ \begin{array}{l} \dot{\mathbf{x}}(t) = \tilde{\mathbf{f}}(\mathbf{x}(t)) \left(\dot{\mathbf{y}}^*(t) + \rho_1(\mathbf{y}^*(t) - \mathbf{y}(t)) \right. \\ \quad \left. + \rho_2 \int_0^t \mathbf{y}^*(\tau) - \mathbf{y}(\tau) d\tau + \sigma(t) \right), \\ \dot{\tilde{\mathbf{f}}}(\mathbf{x}(t)) = (\ddot{\mathbf{x}}(t) - \tilde{\mathbf{f}}(\mathbf{x}(t))\ddot{\mathbf{y}}(t) \\ \quad + \nu(\dot{\mathbf{x}}(t) - \tilde{\mathbf{f}}(\mathbf{x}(t))\dot{\mathbf{y}}(t)))\dot{\mathbf{y}}^\dagger(t). \end{array} \right. \quad (19)$$

Definition 1. For solving a time-variant tracking control problem of UNDS (5)–(7) considering both the system uncertainties and external disturbances, initially starting from a neural network state $\mathbf{y}(0)$, the output trajectory $\mathbf{y}(t)$ synthesized by a neural network model is said to be convergent to the predefined path $\mathbf{y}^*(t)$ if it satisfies

$$\mathbf{y}(t) \rightarrow \mathbf{y}^*(t), \text{ as } t \rightarrow \infty.$$

Theorem 1. For a time-variant tracking control of UNDS (5)–(7) in consideration of both system uncertainties together with continuous-and-bounded random-form disturbances, initially starting from a neural network state $\mathbf{y}(0)$, the output trajectory $\mathbf{y}(t)$ synthesized by the continuous-time neural network model (19) is convergent to the desired output $\mathbf{y}^*(t)$ with each element of steady-state error satisfying $\lim_{t \rightarrow \infty} e_i(t) = 0$, where $i = 1, 2, \dots, m$.

PROOF. Note that the ZNN-based error function $\mathbf{e}(t) \in \mathbb{R}^m$ is a vector with its each element being $e_i(t)$. The Laplace transformation [49] can be utilized from the time domain with real variable t to frequency domain with complex

variable s for scalar form systems. So the theoretical analysis process is presented with the i th subsystem of the vector form neural network model (19). According to the definition of the error function, the i th subsystem (or termed element) of the continuous-time neural network model (19) can be written as follows:

$$\dot{e}_i(t) = -\rho_1 e_i(t) - \rho_2 \int_0^t e_i(\tau) d\tau + \sigma_i(t). \quad (20)$$

Then, within the region of convergence (ROC), one can readily make the Laplace transformation (represented by $\mathcal{F}(s)$) for both sides of equation (20) as below:

$$\mathcal{F}(s) = \mathcal{L}(\dot{e}_i(t)) = \mathcal{L}\left(-\rho_1 e_i(t) - \rho_2 \int_0^t e_i(\tau) d\tau + \sigma_i(t)\right), \quad (21)$$

where operator $\mathcal{L}(\cdot)$ is a Laplace transformation for index $i = 1, 2, \dots, m$. Therefore, the i th subsystem has the transformation form in frequency domain with complex variable s as

$$\mathcal{F}(s) = s e_i(s) - e_i(0) = -\rho_1 e_i(s) - \frac{\rho_2}{s} e_i(s) + \int_0^{+\infty} \sigma_i(t) \exp(-st) dt, \quad (22)$$

where $\int_0^{+\infty} \sigma_i(t) \exp(-st) dt$ is a Laplace transformation of $\sigma_i(t)$. Now we have finished the Laplace transformation of the i th subsystem of the proposed neural network model from the time domain with real variable t to frequency domain with complex variable s . Then, we continue the theoretical analysis process in frequency domain as follows. Equation (22) is rewritten as below:

$$(s^2 + s\rho_1 + \rho_2)e_i(s) = s e_i(0) + s \int_0^{+\infty} \sigma_i(t) \exp(-st) dt.$$

On the basis of the final-value theorem [49], for $|\sigma_i(t)| \leq \sigma_{\max}$ for $t \in [0, T_d]$,

and $\sigma_i(t) = 0$ for $t > T_d$ with $i = 0, 1, \dots, m$, it has:

$$\begin{aligned}
 \left| \lim_{t \rightarrow \infty} e_i(t) \right| &= \left| \lim_{s \rightarrow 0} s e_i(s) \right| = \left| \lim_{s \rightarrow 0} \frac{s^2 \left(e_i(0) + \int_0^{+\infty} \sigma_i(t) \exp(-st) dt \right)}{s^2 + s\rho_1 + \rho_2} \right| \\
 &= \left| \lim_{s \rightarrow 0} \frac{s^2 \left(e_i(0) + \int_0^{T_d} \sigma_i(t) \exp(-st) dt \right)}{s^2 + s\rho_1 + \rho_2} \right| \\
 &\leq \left| \lim_{s \rightarrow 0} \frac{|s^2| \left(|e_i(0)| + \int_0^{T_d} |\sigma_i(t)| \exp(-st) dt \right)}{|s^2 + s\rho_1 + \rho_2|} \right| \\
 &\leq \left| \lim_{s \rightarrow 0} \frac{|s^2| \left(|e_i(0)| + \sigma_{\max} \int_0^{T_d} \exp(-st) dt \right)}{|s^2 + s\rho_1 + \rho_2|} \right| \\
 &= \left| \lim_{s \rightarrow 0} \frac{|s^2| |e_i(0)| + \sigma_{\max} |s| |1 - \exp(-sT_d)|}{|s^2 + s\rho_1 + \rho_2|} \right| \\
 &= 0.
 \end{aligned}$$

Thus, it obtains $|\lim_{t \rightarrow \infty} e_i(t)| = \lim_{t \rightarrow \infty} e_i(t) = 0$ with $i = 1, 2, \dots, m$. Considering vector-valued error function (8) with its each element satisfying the above results, we finally obtain the following result:

$$\mathbf{y}(t) \rightarrow \mathbf{y}^*(t), \text{ as } t \rightarrow \infty. \quad (23)$$

According to Definition 1, one can readily obtain the conclusion that the system output trajectory $\mathbf{y}(t)$ via the proposed neural network model (19) is convergent to the predefined output $\mathbf{y}^*(t)$. This completes the proof. \square

3.3. Discretization

Note that the above neural network model (19) is a continuous-time model. For better implementation, the following EFF [50] is employed:

$$\dot{\mathbf{v}}_k \approx \frac{\mathbf{v}_{k+1} - \mathbf{v}_k}{\varsigma}, \quad (24)$$

where \mathbf{v} denotes the vector-valued or matrix-valued variable that needs to be discretized, and ς is a sampling period. In addition, k is an update index with $t = k\varsigma$.

Note that the EFF is the first one and also the simplest one of discretization formula that is widely applied in engineering for decades [50]. The discretization of the neural network model using the EFF would make the discrete-time model be simplification. Different from the EFF, the Euler backward formula (EBF) requires the priori knowledge of the state, i.e., \mathbf{v}_{k-1} , to calculate the current state \mathbf{v}_k . Such priori knowledge may be difficult to be obtained in advance for the EBF. In addition, other improved discretization formulas, such as the multiple step discretization formulas [33, 51], it is required to additionally design or choose the coefficients, which makes the discretization process be complicated. Therefore, the EFF is selected to discretize the continuous-time neural network model in this work to focus on the proposed neural network model itself. Accordingly, we have the following theoretical analyses on the accuracy of the discrete algorithm.

Theorem 2. *Assume that \mathbf{v} and its first two derivatives are continuous on $[a, b]$ and that $k\varsigma, (k+1)\varsigma \in [a, b]$. Given a sampling period ς , EFF (24) for neural network model (19) has a truncation error of $\mathbf{O}(\varsigma)$.*

PROOF. Using the Taylor expansion theorem [52] yields:

$$\dot{\mathbf{v}}_k = \frac{\mathbf{v}_{k+1} - \mathbf{v}_k}{\varsigma} + \mathbf{e}_{\text{tru}}(\varsigma), \quad (25)$$

where $\mathbf{e}_{\text{tru}}(\varsigma)$ denotes the truncation error of EFF (24). Then, it obtains:

$$\mathbf{e}_{\text{tru}}(\varsigma) = -\frac{\varsigma \mathbf{v}^{(2)}(c)}{2} = \mathbf{O}(\varsigma) \quad (26)$$

with c being a constant between \mathbf{v}_k and \mathbf{v}_{k+1} , and $\mathbf{O}(\varsigma)$ is a vector with each element being $O(\varsigma)$. The proof is thus completed. \square

Considering the integral term $\int_0^t \mathbf{y}^*(\tau) - \mathbf{y}(\tau) d\tau$ in neural network model (19), EFF (24) can not being used directly for the discretization. To discretize the neural network model (19) and integral term $\int_0^t \mathbf{y}^*(\tau) - \mathbf{y}(\tau) d\tau$ in the same way, we define $\mathbf{r}(t) = \int_0^t \mathbf{y}^*(\tau) - \mathbf{y}(\tau) d\tau$. Thus, neural network model (19) is rewritten in a matrix form as follows:

$$\begin{bmatrix} \dot{\mathbf{x}}(t) \\ \dot{\mathbf{r}}(t) \\ \dot{\mathbf{f}}(\mathbf{x}(t)) \end{bmatrix} = \begin{bmatrix} \mathbf{p}(t) + \rho_1 \tilde{\mathbf{f}}(\mathbf{x}(t))(\mathbf{y}^*(t) - \mathbf{y}(t)) \\ \mathbf{y}^*(t) - \mathbf{y}(t) \\ \mathbf{q}(t)\dot{\mathbf{y}}^\dagger(t) \end{bmatrix}, \quad (27)$$

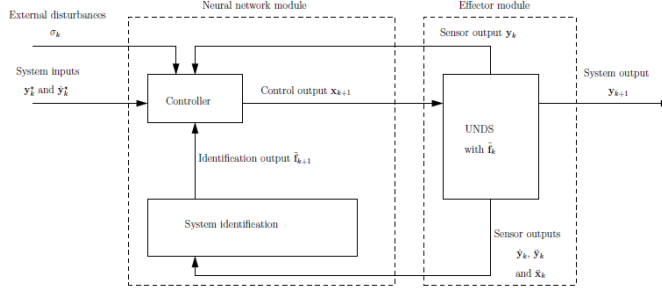


Figure 1: Block diagram of the UNDS (5)–(7) for the time-variant tracking control in the k th update index.

where $\mathbf{p}(t) = \tilde{\mathbf{f}}(\mathbf{x}(t))\dot{\mathbf{y}}^*(t) + \rho_2\tilde{\mathbf{f}}(\mathbf{x}(t))\mathbf{r}(t) + \tilde{\mathbf{f}}(\mathbf{x}(t))\sigma(t)$ and $\mathbf{q}(t) = \ddot{\mathbf{x}}(t) - \tilde{\mathbf{f}}(\mathbf{x}(t))\dot{\mathbf{y}}(t) + \nu(\dot{\mathbf{x}}(t) - \tilde{\mathbf{f}}(\mathbf{x}(t))\dot{\mathbf{y}}(t))$.

According to EFF (24), the continuous-time matrix-form model (27) can be discretized as follows:

$$\begin{bmatrix} \mathbf{x}_{k+1} \\ \mathbf{r}_{k+1} \\ \tilde{\mathbf{f}}_{k+1} \end{bmatrix} = \begin{bmatrix} \varsigma\mathbf{p}_k + \varsigma\rho_1\tilde{\mathbf{f}}_k(\mathbf{y}_k^* - \mathbf{y}_k) \\ \varsigma(\mathbf{y}_k^* - \mathbf{y}_k) \\ \varsigma\mathbf{q}_k\dot{\mathbf{y}}_k^\dagger \end{bmatrix} + \begin{bmatrix} \mathbf{x}_k \\ \mathbf{r}_k \\ \tilde{\mathbf{f}}_k \end{bmatrix}, \quad (28)$$

where $h = \varsigma\rho_1 \in \mathbb{R}^+$ denotes the step size. Besides, $\mathbf{p}_k = \tilde{\mathbf{f}}_k\dot{\mathbf{y}}_k^* + \rho_2\tilde{\mathbf{f}}_k\mathbf{r}_k + \tilde{\mathbf{f}}_k\sigma_k$ and $\mathbf{q}_k = \ddot{\mathbf{x}}_k - \tilde{\mathbf{f}}_k\dot{\mathbf{y}}_k + \nu(\dot{\mathbf{x}}_k - \tilde{\mathbf{f}}_k\dot{\mathbf{y}}_k)$.

Corollary 1. *For a time-variant tracking control problem of UNDS (5)–(7) considering system uncertainties together with continuous and bounded random-form disturbances, initially starting from a neural network state \mathbf{y}_0 , the discrete-time neural network model (28) converges to the theoretical solution of UNDS (5)–(7). The steady-state residual error satisfies $\lim_{k \rightarrow \infty} \|\mathbf{e}_k\|_2$ of the discrete-time neural network model (28) discretized via EFF (24) with step size being $h \in (0, 2)$ being of order $O(\varsigma)$.*

PROOF. It can be generalized from the proofs of Theorems 1 and 2. \square

For better understanding, the corresponding algorithm description about the implementation of the discrete-time neural network model (28) is presented in the algorithm part. Besides, the block diagram of UNDS (5)–(7) for the real-time tracking control in the k th update index is also presented in Fig. 1. Note that the whole UNDS with the proposed neural network model

Algorithm 1 Implementation of Neural Network Model (28)

- 1 **Initialize:** The update index $k = 0$ with initial state vector \mathbf{x}_0 and initial unknown mapping $\tilde{\mathbf{f}}_0$
 - 2 **Set:** The tracking task duration T_d , the sampling period ς , the design parameters γ , λ and ν
 - 3 **Input:** The desired path \mathbf{y}_k^* and its first-order derivative $\dot{\mathbf{y}}_k^*$
 - 4 **Read:** The real-time output trajectory \mathbf{y}_k , and its first-order and second-order derivatives $\dot{\mathbf{y}}_k$ and $\ddot{\mathbf{y}}_k$, and the second-order derivative of state vector $\ddot{\mathbf{x}}_k$
 - 5 **Calculate:** The time derivative of present state vector in the presence of disturbances via $\dot{\mathbf{x}}_k = \mathbf{p}_k + \rho_1 \tilde{\mathbf{f}}_k(\mathbf{y}_k^* - \mathbf{y}_k)$
 - 6 **Calculate:** The time derivative of the unknown mapping via $\dot{\tilde{\mathbf{f}}}_k = \mathbf{q}_k \dot{\mathbf{y}}_k^\dagger$
 - 7 **Update:** The unknown mapping via $\tilde{\mathbf{f}}_{k+1} = \tilde{\mathbf{f}}_k + \varsigma \dot{\tilde{\mathbf{f}}}_k$
 - 8 **Update:** The state vector via $\mathbf{x}_{k+1} = \mathbf{x}_k + \varsigma \dot{\mathbf{x}}_k$
 - 9 **Output:** The real-time \mathbf{x}_{k+1} as control signals for tracking effector
 - 10 **if** $k > T_d/\varsigma$ **then**
 - 11 Stop updating
 - 12 **else**
 - 13 $k \leftarrow k + 1$ and go to step 3
 - 14 **end**
-

is a typical closed-loop system that can take full advantage of the feedback information of the task execution. In addition, for the identification of unknown mapping $\tilde{\mathbf{f}}_k$ with system uncertainties, the model input given by the users and the system effector outputs measured by the corresponding sensors are also fully utilized in real time (see also Fig. 1). The block diagram clearly illustrates the main principle and control strategy of the whole UNDS with the proposed discrete-time neural network model (28).

4. Simulation studies

To validate the efficacy of the proposed neural network model, we successively consider three tracking examples for a general UNDS, a serial robot manipulator, and a parallel robot manipulator, respectively. Without losing generality, dimensions $n = 6$ and $m = 3$ are considered in the simulations.

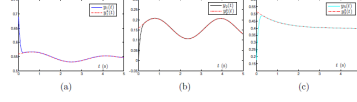


Figure 2: Synthesized tracking outputs by the proposed neural network model (28) for the effector of general UNDS tracking the desired paths. (a) Trajectories of $y_1(t)$ and $y_1^*(t)$. (b) Trajectories of $y_2(t)$ and $y_2^*(t)$. (c) Trajectories of $y_3(t)$ and $y_3^*(t)$.

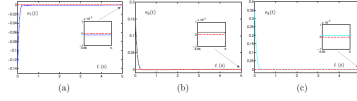


Figure 3: Synthesized tracking errors by the proposed neural network model (28) for the effector of general UNDS tracking the desired paths. (a) Profile of $e_1(t)$. (b) Profile of $e_2(t)$. (c) Profile of $e_3(t)$.

We conduct the tracking process based on an inaccurate mapping $\tilde{\mathbf{f}} = \mathbf{f} + v$ containing the unknown biases v caused by system uncertainties. Moreover, to further investigate the tracking performance of discrete-time neural network model (28), comparisons as well as tests are illustrated in terms of the tracking accuracy and computational complexity. The simulation studies are carried out in MATLAB R2012b environment implemented on a personal-digital computer with the CPU being Inter(R) Core(TM) i5-7200U @ 2.50 GHz, 4.00 GB memory as well as the Windows 10 Ultimate operating system.

4.1. Real-time tracking control of general UNDS

In this part, the time-variant tracking control of a general UNDS (5)–(7) with external disturbances is considered, of which the real-time desired paths are set as follows:

$$\mathbf{y}^*(t) = \begin{bmatrix} y_1^*(t) \\ y_2^*(t) \\ y_3^*(t) \end{bmatrix} = \begin{bmatrix} 0.01(\sin(2t) + \cos(t)) \\ 0.1 \sin(t) \cos(t) \\ 0.12 \exp(-t) \end{bmatrix}.$$

In addition, the external disturbances are set to be

$$\sigma(t) = \begin{bmatrix} 0.05 \sin(0.5t) \\ 0.06 \sin(0.8t) \\ 0.03 \sin(t) \end{bmatrix} + \begin{bmatrix} 0.02 \exp(-2t) \\ 0.03 \exp(-5t) \\ 0.01 \exp(-t) \end{bmatrix}.$$

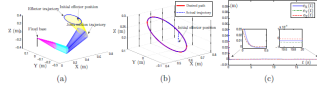


Figure 4: Synthesized motion results by the proposed neural network model (28) for the effector of a serial robot manipulator tracking a circular path. (a) Motion trajectories of the serial robot manipulator. (b) Profiles of the actual trajectory and the desired path. (c) Profiles of the position error.

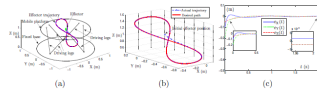


Figure 5: Synthesized motion results by the proposed neural network model (28) for the effector of a parallel robot manipulator tracking an infinity-sign path. (a) Motion trajectories of the parallel robot manipulator. (b) Profiles of the actual trajectory and the desired path. (c) Profiles of the position error.

The tracking task duration is set to be $T_d = 5$ s. The designed parameters are set to be $\gamma = 5$, $\lambda = 5$ and $\nu = 50$ in this tracking task.

The corresponding simulation results of the general UNDS to track the desired path $\mathbf{y}^*(t)$ synthesized by the neural network model (28) are presented in Fig. 2 as well as Fig. 3. Figure 2 illustrates the real-time tracking outputs synthesized by neural network model (28). All the outputs $y_1(t)$, $y_2(t)$ and $y_3(t)$ track the desired paths $y_1^*(t)$, $y_2^*(t)$ and $y_3^*(t)$ precisely within a short time even under the influences of system uncertainties and external disturbances. From Fig. 3, the tracking errors quickly converge to zero with the steady-state tracking error being small values, which illustrates the great tracking performance of the proposed neural network model for the UNDS. The above results also verify that the system's effector tracking task is completed effectively and well.

4.2. Application to serial robot manipulators

In the above subsection, the effectiveness of the proposed neural network model for time-variant tracking control of a general UNDS is substantiated. In this subsection, we present a practical application about the time-variant tracking control of an uncertain serial robot manipulator with external disturbances to further verify the efficacy of the proposed model. The PUMA560 is a conventional kind of serial robot manipulators with six independent control joints, which has been broadly applied in different robotic

fields [46]. In this practical application, the effector of the serial robot manipulator is utilized to track a circular path in 3D space. The detailed kinematics system of the PUMA 560 can be found in [46]. According to the forward kinematics $\mathbf{y}(t) = \mathbf{h}(\Theta(t))$ of serial robot manipulators with $\Theta(t) \in \mathbb{R}^6 = \mathbf{x}(t)$ being the robot joint-angle state, the dynamical equation of the serial robot system at velocity level is depicted as $\dot{\Theta}(t) = \tilde{J}^\dagger(\Theta(t))\dot{\mathbf{y}}(t)$, where $\tilde{J}^\dagger(\Theta(t)) \in \mathbb{R}^{6 \times 3} = \tilde{\mathbf{f}}(\mathbf{x}(t))$ is an unknown pseudo-inverse of Jacobian matrix with uncertain robot system information such as physical parameter biases. Without losing generality, the tracking task duration is set to be $T_d = 20$ s. Predefined parameters are set to be $\gamma = 5$, $\lambda = 5$ and $\nu = 50$ in this tracking task. The initial value of joint angle vector is set to be $\Theta(0) = [0, -\pi/4, 0, 2\pi/3, -\pi/4, 0]^T$ rad, and the initial value of joint velocity vector is set to be $\dot{\Theta}(0) = [0, 0, 0, 0, 0, 0]^T$ rad/s. In addition, desired circular path for the effector is depicted as below:

$$\mathbf{y}^*(t) = \begin{bmatrix} \iota \cos(2\pi \sin^2(0.5\pi t/T_d)) - \iota \\ \iota \cos(\pi/6) \sin(2\pi \sin^2(0.5\pi t/T_d)) \\ \iota \sin(\pi/6) \sin(2\pi \sin^2(0.5\pi t/T_d)) \end{bmatrix},$$

with geometry parameter being set to be $\iota = 0.1$ m in the serial robot manipulators tracking task. In addition, the external disturbances are set to be

$$\sigma(t) = \begin{bmatrix} 0.03 \cos(0.3t) \\ 0.05 \sin(0.7t) \\ 0.01 \sin(t) \end{bmatrix} + \begin{bmatrix} 0.02 \exp(-4t) \\ 0.03 \exp(-3t) \\ 0.01 \exp(-t) \end{bmatrix}.$$

The corresponding simulation results of serial robot manipulator tracking the circular path synthesized by the proposed neural network model (28) are illustrated in Fig. 4. Specifically, the motion result of serial robot manipulator in 3D space during the tracking process is shown in Fig. 4(a). It can be readily found that the actual trajectory of the effector and the desired path are both circular paths. The effector of the serial robot manipulator successfully tracks the desired circular path in 3D space. As illustrated in Fig. 4(b), the actual trajectory of the effector of the serial robot manipulator is sufficiently close to the desired circular path. From Fig. 4(c), the maximal absolute value of effector position error $\mathbf{e} = [e_X, e_Y, e_Z]^T$ (i.e., the difference between the desired path and the actual trajectory in X-, Y-, and Z-axes) is less than 5×10^{-3} m (i.e., less than 1.25% with the whole motion diameter of the task space being about 0.4 m), which illustrates the high accuracy

of tracking process. The above results illustrate that the effector's path-tracking task is completed well synthesized by the proposed neural network model (28).

4.3. Application to parallel robot manipulators

In the application, the time-variant tracking control of parallel robot manipulators is considered. Note that parallel robot manipulators are widely used in many engineering applications because of the inherent high carrying capacity and quick respond speciality [47]. As a typical parallel robot manipulator, Stewart platform contains a mobile platform as well as a fixed base connected together through six independently controlled prismatic legs. The effector locates on the top and center of the mobile platform which can execute the time-variant tracking task. The effector of an uncertain parallel robot manipulator with disturbances is utilized to track an infinity-sign path. According to forward kinematics $\mathbf{y}(t) = \mathbf{h}(\mathbf{l}(t))$ of parallel robot manipulators with $\mathbf{l}(t) \in \mathbb{R}^n = \mathbf{x}(t)$ being the robot leg-length state, the dynamical equation of the parallel robot system is described as $\dot{\mathbf{l}} = \tilde{C}(\mathbf{l}(t))\dot{\mathbf{y}}(t)$, where $\tilde{C}(\mathbf{l}(t)) \in \mathbb{R}^{n \times m} = \tilde{\mathbf{f}}(\mathbf{x}(t))$ is an unknown coefficient matrix with uncertain robot system information. Without losing of generality, the tracking-task duration is set to be $T_d = 2$ s. The designed parameters are set to be $\gamma = 10$, $\lambda = 10$ and $\nu = 50$ in this tracking task. The value of leg length vector is initially set as $\mathbf{l}(0) = [1.184, 1.184, 1.184, 1.184, 1.184, 1.184]^T$ m, and the value of leg velocity vector is initially set as $\dot{\mathbf{l}}(0) = [0, 0, 0, 0, 0, 0]^T$ m/s. In addition, desired circular path for the effector is described as

$$\mathbf{y}^*(t) = \begin{bmatrix} \cos(\pi/2), 1, -\sin(\pi/2) \\ \sin(\pi/2), \cos(\pi/2), 0 \\ \sin(\pi/2), 0, \cos(\pi/2) \end{bmatrix} \times \begin{bmatrix} \eta \cos(\pi(t + 0.5)) \\ \eta \\ -(\eta/2) \sin(2\pi(t + 0.5)) \end{bmatrix},$$

with the geometry parameter being set as $\eta = 0.24$ m in the parallel robot manipulators tracking task. In addition, the external disturbances are set to be

$$\sigma(t) = \begin{bmatrix} 0.05 \sin(0.5t) \\ 0.04 \sin(0.2t) \\ 0.03 \cos(0.6t) \end{bmatrix} + \begin{bmatrix} 0.02 \exp(-2t) \\ 0.03 \exp(-5t) \\ 0.01 \exp(-t) \end{bmatrix}.$$

The related results of the parallel robot manipulator tracking the infinity-sign path synthesized by the proposed neural network model (28) are shown in Fig. 5. Specifically, Fig. 5(a) presents the tracking motion of the parallel

robot manipulator in the 3D space during the tracking process. It can be readily found that the actual trajectory of the effector and the desired path are both infinity-sign paths. The effector for the parallel robot manipulator successfully tracks the desired infinity-sign path. As detailedly zoomed in Fig. 5(b), the actual trajectory of the effector of the parallel robot manipulator in the application is also sufficiently close to the desired infinity-sign path. Besides, the maximal absolute value of effector position error $\mathbf{e} = [e_x, e_y, e_z]^T$ shown in Fig. 5(c) is less than 5×10^{-3} m (i.e., less than 0.625% with the whole motion diameter of the task space being about 0.8 m). The results indicate that the predefined infinity-sign path tracking objective is conducted well even under the system uncertainties and external disturbances.

Note that external disturbances are all considered in three tracking control examples, i.e., tracking control problems of general UNDS, serial robot manipulator, and parallel robot manipulator. Different external disturbances are investigated in different tracking control examples with specific descriptions. Therefore, different specific external disturbances in different specific examples exhibit different specific tracking errors in Fig. 3, Fig. 4(c) and Fig. 4(c), respectively. Three tracking control examples are run independently. In addition, the simulation results keep stable and reproducible for the same simulation condition and the same external disturbances. Each tracking error synthesized by the proposed neural network model remains almost the same in a specific tracking control example with specific external disturbances. So we do not need to calculate mean tracking errors for each specific tracking control example.

Remark 3: Generally, there unavoidably exist various forms of external time-variant disturbances and system uncertainties for the tracking control of UNDS, such as serial and parallel robot manipulators. Generally, different forms of external time-variant disturbances can be described as different forms of mathematical expressions. For examples, the offset errors in robot control system hardware implementation can be deemed as the linear-form time-variant disturbances. During the robot signal processing and transmission, the external signals interference, e.g., electromagnetic interference, can be deemed as the superposition of sine-form time-variant disturbances with unknown frequency and magnitude. The instantaneous decline of power sources equipped in the robot control module can be described as the exponential-decay-form time-variant disturbances. In this work, without losing of generality, we consider the superposition of sine-form and exponential-decay-form time-variant disturbances in simulation studies

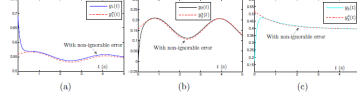


Figure 6: Comparisons on tracking outputs by the CRNN model (29) for the effector of general UNDS tracking the desired paths. (a) Trajectories of $y_1(t)$ and $y_1^*(t)$. (b) Trajectories of $y_2(t)$ and $y_2^*(t)$. (c) Trajectories of $y_3(t)$ and $y_3^*(t)$.

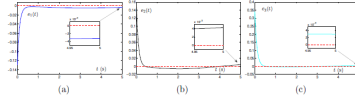


Figure 7: Comparisons on tracking errors by the CRNN model (29) for the effector of general UNDS tracking the desired paths. (a) Profile of $e_1(t)$. (b) Profile of $e_2(t)$. (c) Profile of $e_3(t)$.

to testify the robustness of the neural network model. Moreover, internal system uncertainties such as parameter deviation usually occur in robot control systems due to the diversity and complexity of the control environment, which can also be deemed as the offset errors in robot control system with linear-form mathematical expression. The above discussions are the rationale of considering external time-variant disturbances and system uncertainties in this work.

4.4. Comparisons with other models

To highlight the advantages on robustness of the proposed neural network model, comparisons on the proposed neural network model (28) with existing conventional recurrent neural network (CRNN) model are performed in this part. Note that the CRNN model is also a typical kind of Hopfield neural network [26], which can be readily implemented on circuits such as very large-scale integration. The CRNN model for solving the time-variant tracking control problem of the UNDS (5)–(7) under the influence external disturbances can be described as follows:

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t))(\dot{\mathbf{y}}^*(t) + \delta(\mathbf{y}^*(t) - \mathbf{y}(t)) + \sigma(t)) \quad (29)$$

with $\delta > 0 \in \mathbb{R}$ denoting a predefined parameter to adjust the convergence rate related to the CRNN model (29). Specifically, the comparison results on the proposed neural network model (28) with CRNN model (29) are presented in Fig. 6 through Fig. 9. All the comparative studies are performed

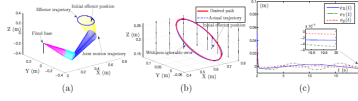


Figure 8: Comparisons on motion results by the CRNN model (29) for the effector of a serial robot manipulator tracking a circular path. (a) Motion trajectories of the serial robot manipulator. (b) Profiles of the actual trajectory and the desired path. (c) Profiles of the position error.

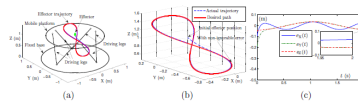


Figure 9: Comparisons on motion results by the CRNN model (29) for the effector of a parallel robot manipulator tracking an infinity-sign path. (a) Motion trajectories of the parallel robot manipulator. (b) Profiles of the actual trajectory and the desired path. (c) Profiles of the position error.

with the same simulation conditions. The corresponding comparison results of the general UNDS to track the desired path $\mathbf{y}^*(t)$ synthesized by the CRNN model (29) are presented in Fig. 6 as well as Fig. 7. Figure 6 comparatively shows the real-time tracking outputs synthesized by the CRNN model (29). All the outputs $y_1(t)$, $y_2(t)$ and $y_3(t)$ can not track the desired paths $y_1^*(t)$, $y_2^*(t)$ and $y_3^*(t)$ precisely with non-ignorable errors under the influences of system uncertainties and external disturbances. From Fig. 7, the tracking errors also can not converge to zero with the steady-state tracking error having an error bound. Such comparative results are different from the results of the outstanding tracking performance of the proposed neural network model (28) for the UNDS.

The corresponding comparison results of serial and parallel robot manipulators tracking the circular path and the infinity-sign path synthesized by the CRNN model (29) are illustrated in Fig. 8 and Fig. 9. Specifically, the motion results of serial and parallel robot manipulators in 3D space during the tracking process are shown in Fig. 8(a) and Fig. 9(a). One can readily find that the actual trajectories of effectors are not the exact circular and infinity-sign paths. The effectors of robot manipulators can not successfully track the desired paths in 3D space. As magnified in Fig. 8(b) and Fig. 9(b), the actual trajectories of the effector of robot manipulators can not track the predefined path. From Fig. 8(c) and Fig. 9(c), the maximal abso-

Table 1: Performance comparisons among different neural network models for solving the time-variant tracking control of UNDS.

Model	Anti-disturbance	Uncertainty rejection	Convergence performance	Tracking accuracy \ddagger
Proposed (28)	Yes	Yes	Convergence to zero	High
CRNN (29)	No	No	With error bound	Low
[1]	Yes	No	With error bound	Low
[5]	No	No	With error bound	Low
[53]	No	Yes	With error bound	Low
[54]	No	No	With error bound	Low

Note: \ddagger The tracking control tasks for UNDS are under the influence of both system uncertainties and external disturbances.

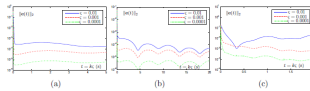


Figure 10: Synthesized residual errors by the proposed neural network model (28) with different values of sampling period ζ and step size $h = 0.2$ for the effector of different UNDS tracking different desired paths shown in Section 4.1 through Section 4.3. (a) Residual errors of tracking task in Section 4.1. (b) Residual errors of tracking task in Section 4.2. (c) Residual errors of tracking task in Section 4.3.

lute value of effector position errors can not converge to zero with relatively big error bounds, which illustrates the low accuracy of tracking process under the influences of system uncertainties and external disturbances. All the above comparison results illustrate that the advantages on robustness of the proposed neural network model (28).

Moreover, to fully verify the advantages of the proposed neural network model (28), performance comparisons among different existing neural network models for tracking control of UNDS with both system uncertainties and external disturbances are illustrated in Table 1. As compared in the table, the proposed neural network model (28) can achieve the tracking control task and simultaneously handle both system uncertainties and external disturbances. The robustness together with outstanding convergence performance and high tracking control accuracy have been illustrated via three tracking control examples in the above simulation studies, which are superior to other neural network models in existing literatures [1, 5, 53, 54] (see also Table 1).

4.5. Performance Tests

To deeply study the tacking performance of the proposed neural network model, extensive tests are conducted. Firstly, we conduct the simulation tests

Table 2: Operational complexity of the proposed model (28) for time-variant tracking control of UNDS (5)–(7) with $\tilde{\mathbf{f}}_k \in \mathbb{R}^{n \times m}$ and $\mathbf{x}_k \in \mathbb{R}^n$.

Component	Number	Complexity
Multiplications/divisions	$7mn + 6m + 1$	$O(mn)$
Additions/subtractions	$7mn + 8m$	$O(mn)$

in terms of the tracking accuracy under different values of sampling period ς . The residual errors $\|\mathbf{e}_k\|_2$ of three tracking tasks in Section 4.1 through Section 4.3 with different values of sampling period ς are shown in Fig. 10. Specifically, as shown in Fig. 10(a), the maximal steady-state residual errors (MSSREs) decrease 10 times (from about 5×10^{-4} m to 5×10^{-5} m and further to 5×10^{-6} m) as the values of sampling period ς reduce by 10 times (from 1×10^{-2} s to 1×10^{-3} s and further to 1×10^{-4} s). That is to say, the MSSREs synthesized by the discrete-time neural network model (28) change in an $O(\varsigma)$ pattern. Such graphical results are consistent with the theoretical results in Section 3.3. Thus, by choosing appropriate values of sampling period ς , the proposed neural network model (28) potentially possesses high enough computational accuracy and tracking performance.

Then, the computational complexity of neural network model is further investigated. The numbers of matrix operations, i.e., multiplications/divisions and additions/subtractions, required in applying the proposed model (28) are listed in Table 2. In general, the operational complexity of model (28) for time-variant tracking control of UNDS (5)–(7) per update is of order $O(mn)$. Specifically, as for the serial robot manipulator (with $m = 3$ and $n = 6$) in Section 4.2, model (28) performs 145 multiplications/divisions and 150 additions/subtractions per update. Such an operational complexity is acceptable in most practical applications. Moreover, the time complexity of model (28) is also illustrated in Table 3. Restricted by the computer operating environment, the total update duration (TUD) of model (28) for handling different tracking tasks is much smaller than the whole task duration T_d . Furthermore, the average-computing-time per updating (ACTPU) (being of order 10^{-4} s) is also smaller than the shortest sampling period $\varsigma = 0.001$ s. Such a computational speed fulfills the requirements of the real-time applications [26].

Table 3: Time complexity of the proposed model (28) for time-variant tracking control of different UNDS in terms of TUD and ACTPU.

UNDS	T_d	TUD	ACTPU
General UNDS	5 s	0.1293 s	2.5875×10^{-4} s
Serial robot manipulator	20 s	0.5637 s	2.8189×10^{-4} s
Parallel robot manipulator	2 s	0.4947 s	2.4739×10^{-4} s

5. Conclusion

In this paper, a unified neural approach of simultaneous identification, tracking control and disturbance rejection in the framework of the ZNN has been proposed for designing neural network model to address the time-variant tracking control issue of the UNDS. By following this approach, the proposed neural network model has featured the full utilization of the information from the input and output measurements of the UNDS, and has shown an outstanding tracking performance even simultaneously restricted by system uncertainties and external disturbances. Then, the discrete-time model has been further derived via the EFF. The algorithm description and its block diagram representation have been presented for the implementation convenience of practitioners. Besides, theoretical analyses have been presented to guarantee the validity of such model for the time-variant tracking of UNDS. Moreover, simulation results of a general tracking example, two robot applications as well as performance comparisons and tests have demonstrated the effectiveness and advantages of the proposed neural network model.

References

- [1] W. Li, B. Liao, L. Xiao, R. Lu, A recurrent neural network with predefined-time convergence and improved noise tolerance for dynamic matrix square root finding, *Neurocomputing* 337 (2019) 262–273.
- [2] J. Na, X. Ren, D. Zheng, Adaptive control for nonlinear pure-feedback systems with high-order sliding mode observer, *IEEE Trans. Neural Netw. Learn. Syst.* 24 (3) 2013 370–382.

- [3] P. Millán, L. Orihuela, I. Jurado, F. R. Rubio, Formation control of autonomous underwater vehicles subject to communication delays, *IEEE Trans. Control Syst. Technol* 22 (2) (2014) 770–777.
- [4] Z. Zhang, A. Beck, N. Magnenat-Thalmann, Human-like behavior generation based on head-arms model for tracking external targets and body parts, *IEEE Trans. Cybern.* 45 (8) (2015) 1390–1400.
- [5] Z. Zhang, Z. Li, Y. Zhang, Y. Luo, Y. Li, Neural-dynamic-method-based dual-arm CMG scheme with time-varying constraints applied to humanoid robots, *IEEE Trans. Neural Netw. Learning Syst.* 26 (12) (2015) 3251–3262.
- [6] P. Miao, Y. Shen, Y. Li, L. Bao, Finite-time recurrent neural networks for solving nonlinear optimization problems and their application, *Neurocomputing* 177 (2016) 120–129.
- [7] S. Dutta, P. K. Patchaikani, L. Behera, Near-optimal controller for nonlinear continuous-time systems with unknown dynamics using policy iteration, *IEEE Trans. Neural Netw. Learning Syst.* 27 (7) (2016) 1537–1549.
- [8] D. Liu, Q. Wei, Policy iteration adaptive dynamic programming algorithm for discrete-time nonlinear systems, *IEEE Trans. Neural Netw. Learning Syst.* 25 (3) (2014) 621–634.
- [9] Z. Fu, W. Xie, S. Rakheja, J. Na, Observer-based adaptive optimal control for unknown singularly perturbed nonlinear systems with input constraints, *IEEE/CAA J. Autom. Sin.* 4 (1) (2017) 48–57.
- [10] Y. Shen, Y. Huang, Uniformly observable and globally Lipschitzian nonlinear systems admit global finite-time observers, *IEEE Trans. Autom. Control* 54 (11) (2009) 2621–2625.
- [11] J. Na, X. Ren, D. Zheng, Adaptive control for nonlinear pure-feedback systems with high-order sliding mode observer, *IEEE Trans. Neural Netw. Learning Syst.* 24 (3) (2013) 370–382.
- [12] G. Hu, N. Gans, N. Fitz-Coy, W. Dixon, Adaptive homography-based visual servo tracking control via a quaternion formulation, *IEEE Trans. Control Syst. Technol.* 18 (1) (2010) 128–135.

- [13] Z. Li, J. Deng, R. Lu, Y. Xu, J. Bai, C.-Y. Su, Trajectory-tracking control of mobile robot systems incorporating neural-dynamic optimized model predictive approach, *IEEE Trans. Syst., Man, Cybern., Syst.* 46 (6) (2016) 740–749.
- [14] J. Na, G. Herrmann, Online adaptive approximate optimal tracking control with simplified dual approximation structure for continuous-time unknown nonlinear, *IEEE/CAA J. Autom. Sin.* 1 (4) (2014) 412–422.
- [15] J. Chen, B. Ren, Q.-C. Zhong, UDE-based trajectory tracking control of piezoelectric stages, *IEEE Trans. Ind. Electron.* 63 (10) (2016) 6450–6459.
- [16] Y. Pan, J. Wang, Model predictive control of unknown nonlinear dynamical systems based on recurrent neural networks, *IEEE Trans. Ind. Electron.* 59 (8) (2012) 3089–3101.
- [17] L. Xiao, Y. Zhang, K. Li, B. Liao, Z. Tan, A novel recurrent neural network and its finite-time solution to time-varying complex matrix inversion. *Neurocomputing* 331 (2019) 483–492.
- [18] D. Chen, S. Li, Q. Wu, Rejecting chaotic disturbances using a super-exponential-zeroing neurodynamic approach for synchronization of chaotic sensor systems, *Sensors* 19 (1) (2019) 74.
- [19] J. J. Hopfield, D. W. Tank, ‘Neural’ computation of decisions in optimization problems, *Biol. Cybern.* 52 (3) (1985) 141–152.
- [20] J. Hopfield, D. Tank, Simple ‘neural’ optimization networks: An A/D converter, signal decision circuit, and a linear programming circuit, *IEEE Trans. Circuits Syst. CAS-33* (5) (1986) 533–541.
- [21] D. Chen, Y. Zhang, S. Li, Zeroing neural-dynamics approach and its robust and rapid solution for parallel robot manipulators against superposition of multiple disturbances *Neurocomputing* 275 (2018) 845–858.
- [22] S. Li, J. He, Y. Li, M. U. Rafique, Distributed recurrent neural networks for cooperative control of manipulators: a game-theoretic perspective, *IEEE Trans. Neural Netw. Learning Syst.* 28 (2) (2017) 415–426.
- [23] Y. Xia, G. Feng, J. Wang, A recurrent neural network with exponential convergence for solving convex quadratic program and related linear piecewise equations, *Neural Netw.* 17 (7) (2004) 1003–1015.

- [24] Y. Shen, LMI-based stability criteria with auxiliary matrices for delayed recurrent neural networks, *IEEE Trans. Circuits Syst. Express Briefs* 55 (8) (2008) 811–815.
- [25] Q. Liu, J. Wang, A one-layer recurrent neural network with a discontinuous activation function for linear programming, *Neural Comput.* 20 (5) (2008) 1366–1383.
- [26] Y. Zhang, D. Jiang, J. Wang, A recurrent neural network for solving Sylvester equation with time-varying coefficients, *IEEE Trans. Neural Netw.* 13 (5) (2002) 1053–1063.
- [27] Y. Zhang, J. Wang, Y. Xia, A dual neural network for redundancy resolution of kinematically redundant manipulators subject to joint limits and joint velocity limits, *IEEE Trans. Neural Netw.* 14 (3) (2003) 658–667.
- [28] Y. Zhang, J. Wang, Global exponential stability of recurrent neural networks for synthesizing linear feedback control systems via pole assignment, *IEEE Trans. Neural Netw.* 13 (3) (2002) 633–644.
- [29] M. Di Marco, M. Forti, P. Nistri, L. Pancioni, Nonsmooth neural network for convex time-dependent constraint satisfaction problems, *IEEE Trans. Neural Netw. Learning Syst.* 27, (2) (2016) 295–307.
- [30] M. Di Marco, M. Forti, P. Nistri, L. Pancioni, Discontinuous neural networks for finite-time solution of time-dependent linear equations, *IEEE Trans. Cybern.* 46 (11) (2016) 2509–2520.
- [31] L. Xiao, A nonlinearly activated neural dynamics and its finite-time solution to time-varying nonlinear equation, *Neurocomputing* 173 (2016) 1983–1988.
- [32] D. Chen, S. Li, F.-J. Lin, Q. Wu, New super-twisting zeroing neural-dynamics model for tracking control of parallel robots: a finite-time and robust solution, *IEEE Trans. Cybern.*, to be published, doi: 10.1109/TCYB.2019.2930662.
- [33] L. Jin, Y. Zhang, Discrete-time Zhang neural network for online time-varying nonlinear optimization with application to manipulator motion

- generation, *IEEE Trans. Neural Netw. Learning Syst.* 26 (7) (2015) 1525–1531.
- [34] D. Guo, Y. Zhang, Zhang neural network for online solution of time-varying linear matrix inequality aided with an equality conversion, *IEEE Trans. Neural Netw. Learning Syst.* 25 (2) (2014) 370–382.
- [35] B. Cai, Y. Zhang, Different-level redundancy-resolution and its equivalent relationship analysis for robot manipulators using gradient-descent and Zhang *et al.*'s neural-dynamic methods, *IEEE Trans. Ind. Electron.* 59 (8) (2012) 3146–3155.
- [36] Y. Zhang, B. Mu, H. Zheng, Link between and comparison and combination of Zhang neural network and Quasi-Newton BFGS method for time-varying quadratic minimization, *IEEE Trans. Cybern.* 43 (2) (2013) 490–453.
- [37] L. Xiao, A new design formula exploited for accelerating Zhang neural network and its application to time-varying matrix inversion, *Theor. Comput. Sci.* 647 (2016) 50–58.
- [38] D. Guo, Z. Nie, L. Yan, Novel discrete-time Zhang neural network for time-varying matrix inversion, *IEEE Trans. Syst., Man, Cybern., Syst.* 47 (8) (2017) 2301–2310.
- [39] D. Chen, Y. Zhang, Robust zeroing neural-dynamics and its time-varying disturbances suppression model applied to mobile robot manipulators, *IEEE Trans. Neural Netw. Learning Syst.* 29 (9) (2018) 4385–4397.
- [40] L. Jin, Y. Zhang, S. Li, Y. Zhang, Modified ZNN for time-varying quadratic programming with inherent tolerance to noises and its application to kinematic redundancy resolution of robot manipulators, *IEEE Trans. Ind. Electron.* 63 (11) (2016) 6978–6988.
- [41] Y. Zhang, J. Wang, Y. Xu. A dual neural network for bi-criteria kinematic control of redundant manipulators, *IEEE Trans. Robot. Autom.* 18 (6) (2002) 923–931.
- [42] D. Chen, S. Li, Q. Wu, X. Luo, New disturbance rejection constraint for redundant robot manipulators: an optimization perspective, *IEEE Trans. Ind. Informat.* to be published, doi: 10.1109/TII.2019.2930685.

- [43] D. Zheng, W. F. Xie, X. Ren, J. Na, Identification and control for singularly perturbed systems using multitime-scale neural networks, *IEEE Trans. Neural Netw. Learning Syst.* 28 (2) (2017) 321–333.
- [44] M. A. González-Olvera, Y. Tang, Black-box identification of a class of nonlinear systems by a recurrent neurofuzzy network, *IEEE Trans. Neural Netw.* 21 (4) (2010) 672–679.
- [45] L. Jin, S. Li, B. Hu, M. Liu, J. Yu, A noise-suppressing neural algorithm for solving the time-varying system of linear equations: a control-based approach, *IEEE Trans. Ind. Informat.* 15 (1) (2019) 236–246.
- [46] D. Chen, Y. Zhang, S. Li, Tracking control of robot manipulators with unknown models: a Jacobian-matrix-adaption method, *IEEE Trans. Ind. Informat.* 14 (7) (2018) 3044–3053.
- [47] A. M. Mohammed, S. Li, Dynamic neural networks for kinematic redundancy resolution of parallel Stewart platforms, *IEEE Trans. Cybern.* 17 (3) (2015) 1400–1410.
- [48] C. Yang, Z. Li, R. Cui, B. Xu, Neural network based motion control of underactuated wheeled inverted pendulum models, *IEEE Trans. Neural Netw. Learn. Syst.* 25 (11) (2014) 2004–2016.
- [49] A. V. Oppenheim, A. S. Willsky, *Signals and Systems*. Englewood Cliffs, NJ, USA: Prentice-Hall, 1997.
- [50] J. C. Butcher, *Numerical methods for ordinary differential equations*. New Jersey: John Wiley & Sons, 2003.
- [51] D. Guo, Z.-Y. Nie, L. Yan, Theoretical analysis, numerical verification and geometrical representation of new three-step DTZD algorithm for time-varying nonlinear equations solving, *Neurocomputing* 214 (2016) 516–526.
- [52] J. H. Mathews, K. D. Fink, *Numerical Methods Using MATLAB*. New Jersey: Prentice Hall, 2004.
- [53] Z. Xu, S. Li, X. Zhou, W. Yan, T. Cheng, D. Huang, Dynamic neural networks based kinematic control for redundant manipulators with model uncertainties, *Neurocomputing* 329 (2019) 255–266.

- [54] L. Jin, S. Li, H. M. La, X. Luo, Manipulability optimization of redundant manipulators using dynamic neural networks, *IEEE Trans. Ind. Electron.* 64 (6) (2017) 4710–4720.

Journal Pre-proof



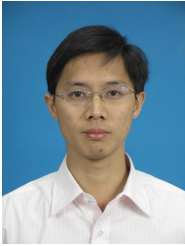
Dechao Chen received the B.S. degree in Electronic Information Science and Technology from Guangdong University of Technology, Guangzhou, China, in 2013, and the Ph.D. degree in Information and Communication Engineering from Sun Yat-sen University, Guangzhou, China, in 2018. He was a Postdoctoral Fellow at the Department of Computing, The Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong. He is currently an Associate Professor with School of Computer Science and Technology, Hangzhou Dianzi University, Hangzhou, China. His research interests include robotics, neural networks, dynamics systems, control systems, optimization, and machine learning.



Shuai Li received B.E. degree in precision mechanical engineering from Hefei University of Technology, Hefei, China, in 2005, M.E. degree in automatic control engineering from University of Science and Technology of China, Hefei, in 2008, and Ph.D. degree in electrical and computer engineering from Stevens Institute of Technology, Hoboken, NJ, USA, in 2014. He is currently an Associate Professor (Reader) at Swansea University, Wales, UK, leading the Robotic Lab, conducting research on robot manipulation and impedance control, multi-robot coordination, distributed control, intelligent optimization and control, and legged robots. Dr. Li is the founding Editor-in-Chief of International Journal of Robotics and Control and the General Co-Chair of 2018 International Conference on Advanced Robotics and Intelligent Control.



Qing Wu received his Ph.D. degree in computer science from Zhejiang University in 2006. He is a professor with the School of Computer Science and Technology, Hangzhou Dianzi University, Hangzhou, China. His current research interests include machine learning, data mining, adaptive software and ubiquitous computing.



Liefan Liao received the B.E. degree in computer science and technology from the Central South University of Technology, Changsha, China, in 1997, and the Ph.D. degree in management science and engineering from Xi'an Jiaotong University, Xi'an, China, in 2009.

Dr. Liao is currently with the School of Information Engineering, Jiangxi University of Science and Technology, Ganzhou, China, as a Professor and the Dean. His current research interests include machine learning, neural networks, and organizational learning.

Conflict of Interest

We declare that there is no conflict of interest for this manuscript.

Journal Pre-proof