ARTICLE IN PRESS

Neurocomputing ∎ (∎∎∎∎) ∎∎∎–∎∎∎

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

# Terminal neural computing: Finite-time convergence and its applications

Ying Kong [a,*], Hui-juan Lu [b], Yu Xue [d], Hai-xia Xia [c]

[a] School of Information and Electronic Engineering, Zhejiang University of Science and Technology, China
[b] School of Information Engineering, China Jiliang University, China
[c] College of Informatics, Zhejiang Sci-Tech University, China
[d] Nanjing University of Information Science and Technology, China

ARTICLE INFO

*Article history:*
Received 23 October 2015
Received in revised form
11 May 2016
Accepted 15 May 2016

*Keywords:*
Finite-time convergence
Neural networks
Time-varying matrices
Redundant manipulators
Trajectory planning

ABSTRACT

This paper will discuss the function of terminal neural networks (TNN) to enhance the convergent behaviors of asymptotic ones. The terminal attraction of the matrix differential equations is analyzed, and the results show that the method can assure the networks of converging to zero during a limited period. The terminal neural networks can also be used to account for the time-varying matrix inversion as well as the trajectory planning of redundant manipulators. The typical example for a planar is the manipulator in which the end-effector appeared as a closed path, and the joint variables can return to the initial values, making the motion repeatable. The simulation results certify for the validity and superiority of the terminal neural method.



## 1. Introduction

Sylvester equation is commonly encountered in mathematics and control theory [1] and finds applications in linear least squares regression [2], disturbance decoupling [3], Eigen-structure assignment [4,5], etc. And much more efforts have been applied to the solution algorithms of matrix square roots due to its basic roles. A lot of parallel-processing computational themes including ample active recurrent neural networks (RNN) have been fully generated and analyzed. RNN method is treated as an effective alternative to real-time computation and optimization for solving the redundancy resolution problems, owning to its parallel-processing nature and convenience of hardware implementation. Besides, robots always tend to have good profits in lots of fields including science and engineering which will activate the robot developers to enhance the functionality and flexibility of robot manipulators. As a robot manipulator is redundant when the degrees of freedom (DOF) are much more available than the minimum of DOF which is always required to work as a given end-effect main task [6,7]. Redundant manipulators are more likely to have a wider operational space as well as extra DOF caters to the number of functional constraints. RNN have been studied for real-time solution to those redundancy-resolution problems [8,9]. In recent years, Zhang neural network (ZNN) has been newly proposed and applied to robot kinematic planning [10].

Recurrent neural network has been thoroughly and widely studied in numerous scientific fields, especially after the discovery of the famous Hopfield neural network [11] which was initially designed for real-time process while the recurrent neural networks are getting more and more popular. Various gradient-based recurrent neural networks are proposed for solving Sylvester equation. Those methods adopt the norm of error matrix as the index and are applied to the gradient-based neural networks to ensure the error norm and could be vanished to zero with time in time-invariant case. However, in most time-varying problems, it intrinsically exists in scientific areas, in which situation, the error norm cannot be zeroed anymore even after such an infinite time. But fortunately, a novel neural network named Zhang neural network has been already proposed, and the result avoids the lagging errors thus can guarantee exact convergent performance to time-varying solution of time-varying problem in the error-free manner. ZNN may also be called an asymptotic neural network (ANN). Zak regarded terminal attracts as motivation function and applied finite-time neural network to time-varying matrix inversion [12]. We know that the normal activation function has a wider application background. So a terminal neural network (TNN) proposed is of finite-time characteristics when the time tends to be finite. To the best of our knowledge, it is a brilliant neural-solution to time-varying Sylvester equation as well as the reverse kinematic problem of redundant robot manipulators [13–15].

* Corresponding author.
*E-mail address:* kongying-888@163.com (Y. Kong).

http://dx.doi.org/10.1016/j.neucom.2016.05.091
0925-2312/© 2016 Elsevier B.V. All rights reserved.

Up to now, few works are reported to provide a finite-time neuro-solution to the time-varying Sylvester equation problem. The main contributions of this paper may lie in the following: (i) To solve the time-varying quadratic problem, an finite-time convergent neural networks is designed based on a vector-valued error function, which gives the exact solution after the convergent instant.

(ii) A novel repeatable robot manipulator program based on joint-velocity level scheme is presented and studied to reduce the joint-angle drift phenomenon. To the best of our knowledge, this terminal repeatable kinematic scheme which makes the joint angle of manipulator to its initial position in finite time has not been investigated by other researchers.

(iii) The comparison between TNN and ANN is analyzed and tested in this paper. Computer simulation results show good results on planar three-link redundant manipulator in velocity scheme.

An illustrative comparison example is presented, where TNN models are used to solve the time-varying problems. The novel velocity minimization using terminal attractors of the proposed TNN is shown effectively.

The remainder contents of this paper is organized as follows. In Section 2, the terminal neural network is put up. Section 3 will continue the analysis with terminal network in theory. In Section 4, we adopt the instance of time-varying matrix inversion to different the performance of the asymptotic neural network (ANN) with the TNN. In Section 5, terminal neural network is adopted to find the answers for the joint-angle drift problems of three-link planar redundant robot arms. Section 6 is the conclusion of this paper.

## 2. Terminal neural network

Firstly, for the time-varying matrix, $A(t)=(A_{ij}(t))\in R^{n\times n}$, $A_{ij}(\cdot)$: $R^{+}\to R$, $A_{ij}(\cdot)$, is the every entry of matrix $A(t)$, $1\le i,j\le n$. Throughout this paper, the following notation is used, a positive constant $\alpha$, $A^{\alpha}(t)=(A_{ij}^{\alpha}(t))$. The matrix function derivative $dA(t)/dt$ is the derivative of matrix $A(t)$, That is $dA(t)/dt=(\dot{A}_{ij}(t))$. For matrices, it is defined entry-wisely, i.e., for matrix $\int_a^b A(t)\mathrm{d}t=\left(\int_a^b A_{ij}(t)\mathrm{d}t\right)$.

Consider the following differential equations governed by

$$\frac{\mathrm{d}}{\mathrm{d}t}E(t)=-\gamma S(E(t)) \tag{1}$$

a where $\gamma>0$ is a positive constant, $E(t)\in R^{n\times n}$ is a time-varying matrix, $S(\cdot)$: $R^{n\times n}\to R^{n\times n}$ is an activation function, which is defined for matrix $S(E(t))=(S(E_{ij}(t)))$.

We choose the Lyapunov candidate $V(t)=\frac{1}{2}E^2(t)$. The derivative of $V(t)$ can be calculated as:

$$\frac{\mathrm{d}}{\mathrm{d}t}V(t)=E(t)\odot\frac{\mathrm{d}}{\mathrm{d}t}E(t)=-\gamma E(t)\odot S(E(t))=-\gamma E_{ij}(t)S(E_{ij}(t))$$

where symbol $\odot$ denotes the Hadamard product of two matrices. $E(t)\odot S(E(t))$ should be positive to guarantee the convergence of $E(t)$. Usually $S(\cdot)$ is chosen a monotonically increasing and odd function, satisfying $S(-\cdot)=-S(\cdot)$. Besides, different choices of activation function $S(\cdot)$ lead to different convergence performance of Formula (1).

Consider the following TNN governed by

$$\frac{\mathrm{d}}{\mathrm{d}t}E(t)=-\gamma S(E^{\alpha}(t)) \tag{2}$$

a where $\alpha>0$, then $\alpha=1$ which implies that Formula (2) is equivalently written as (1). Different from the sign-bi-power activation function which make $(\cdot)^{\alpha}$ as a kind of activation functions, the neural network function (2) is a kind of dynamic system (in fact, it is a terminal attraction dynamic system) [14].

Generally speaking, in the model (2), any monotonically increasing odd activation function $S(\cdot)$ can be adopted to construct the neural network. However, different choices for the activation function will lead to different convergence processes. Three types of functions are usually used:

(1) a linear function: $S(u)=u$;
(2) a power function $S(u)=u^p$ with integer $p\ge 3$;
(3) a bipolar sigmoid function.

$S(u)=(1-\exp(-\xi u))/(1+\exp(-\xi u))$ with $\xi\ge 1$. The linear activation function, $S(\cdot)=\cdot$ is usually used.

The detailed theoretical analysis of TNN model (2) is presented here. We also choose the Lyapunov candidate $V(t)=\frac{1}{2}E^2(t)$. The derivative of $V(t)$ can be expressed as:

$$\frac{\mathrm{d}}{\mathrm{d}t}V(t)=E(t)\odot\frac{\mathrm{d}}{\mathrm{d}t}E(t)=-\gamma E(t)\odot S(E^{\alpha}(t))$$

where $E(t)\odot S(E^{\alpha}(t))$ should be positive matrix, and $S(\cdot)$ is chosen as a monotonically increasing and odd function. So $\alpha$ can be set as:

(1) $E(t)\ge\Lambda$, $\alpha=q_1/p_1$, $q_1$ and $p_1$ are positive odd number respectively satisfying $q_1\ge p_1$.
(2) $E(t)<\Lambda$, $\alpha=q_2/p_2$, $q_2$ and $p_2$ are positive odd number respectively satisfying $q_2<p_2$.

$\Lambda$denotes an approximately dimensioned identity-matrix. The common type for TNN is given as:

$$\frac{\mathrm{d}}{\mathrm{d}t}E(t)=-\gamma S\left(\alpha_1E^{\frac{q_1}{p_1}}(t)+\alpha_2E^{\frac{q_2}{p_2}}(t)\right) \tag{3}$$

where $\gamma$, $\alpha_1$, $\alpha_2>0$, $q_1$, $p_1$, $q_2$, $p_2$ are positive odd number requiring $q_1\ge p_1$, $q_2<p_2$. In this paper, we adopt the linear activation function $S(E)=E$, $q_1=p_1$, $q_2=q$, $p_2=p$.

## 3. Finite-time convergence analysis

When the activation function $S(E)=E$ is given, the dynamic neural network is applied

$$E(t)=E(0)e^{-\gamma t} \tag{4}$$

which means that, as $t\to\infty$, $E(t)\to 0$ globally and exponentially as long as time goes infinity. The design formula (4) can be called asymptotically convergent neural network (ANN), with constant $\gamma$ denoting a positive design-parameter used to scale the convergence rate.

In order to improve the convergent rate, we introduce terminal attraction and propose a terminal neural network (TNN), which makes the error $E(t)$ to converge to zero in a finite time. Consider the following matrix differential equation governed by:

$$\dot{E}(t)=-\gamma E^{q/p}(t) \tag{5}$$

where $\gamma>0$, $q$ and $p$ are positive odd number satisfying $q<p$. Eq. (5) can be rewritten as

$$\mathrm{d}t=-\frac{1}{\gamma}E(t)^{-q/p}\odot\mathrm{d}E(t)$$

Integrating both side yields

$$\int_0^t\mathrm{d}t=-\frac{1}{\gamma}\int_{E(0)}^{0}E(t)^{-q/p}\odot\mathrm{d}E(t)$$

Hence, starting from any initial value $E(0)>0$, $E(t)$, converges to the equilibrium, $(E(t)=0)$, the convergence instant can be calculated by