

# 有限值终态递归神经网络计算

孙明轩 翁丁恩 张 钰

浙江工业大学信息工程学院 杭州 310023



**摘要** 通常的递归神经网络计算方法采用渐近收敛的网络模型,误差函数渐近收敛于零,理论上需经过无穷长的计算时间才能获得被求解问题的精确解。文中提出了一种终态递归神经网络模型,该网络形式新颖,具有有限时间收敛特性,用于解决时变矩阵计算问题时可使得计算过程快速收敛,且计算精度高。该网络的另一特点是动态方程右端函数值有限,易于实现。首先,分析渐近收敛网络模型在时变计算问题求解方面的缺陷,说明引入终态网络模型的必要性;然后,给出终态网络动态方程,推导出该网络收敛时间的具体表达式。对于时变矩阵逆和广义逆求解,定义一个误差函数,并依据误差函数构造终态递归神经网络进行求解,使计算过程在有限时间内收敛便能得到精确解。在将任意初始位置下的冗余机械臂轨迹规划任务转换为二次规划问题后,利用所提出的神经网络进行计算,得出的关节角轨迹导致末端执行器完成封闭轨迹跟踪,且关节角严格返回初始位置,以实现可重复运动。使用 MATLAB/SIMULINK 对时变矩阵计算问题和机器人轨迹规划任务分别进行仿真,通过比较分别采用渐近网络模型和终态网络模型时的计算过程与结果可以看出,使用终态网络模型的计算过程收敛快且显著提高了计算精度。对不同时变计算问题的求解体现了所提神经网络的应用背景。

**关键词**: 终态神经网络; 矩阵逆; 广义逆; 轨迹规划; 工业机器人

**中图分类号** TP18

## Time-variant Neurocomputing with Finite-value Terminal Recurrent Neural Networks

SUN Ming-xuan, WENG Ding-en and ZHANG Yu

College of Information Engineering, Zhejiang University of Technology, Hangzhou 310023, China

**Abstract** Conventional computing methods, by using recurrent neural networks, ensure the asymptotic convergence of the computing error such that the error converges to zero and the exact solution can be obtained as time approaches infinity. In this paper, a novel model of terminal recurrent neural networks was presented to address online computation problems arising from time-varying matrices. Such kind of network model is of the characteristics of the limited values of the right-hand side function and the finite settling time. Firstly, the shortcoming of asymptotically convergent network models in solving time-varying computational problems is analyzed, and the necessity of introducing the terminal network models is given. Then, the dynamics of the terminal network is characterized with the derivation for the expression of the settling time. For solving the problems of inverse and generalized inverses of time-varying matrices, an error function is defined, a terminal recurrent neural network is constructed based on the error function, so that the accurate solution can be achieved. For the path planning of industrial manipulators, the end effector tracks the closed trajectory by applying the terminal neural network, the joint angle returns to the initial position, and the repetitive motion is conducted in the presence of arbitrary initial position. MATLAB/SIMULINK is used for simulation of solving time-varying matrix computing problems and trajectory planning tasks of manipulators. By comparing the results obtained by the asymptotic network and the terminal network, it can be seen that the computing process using the terminal network converges in finite time and the computing accuracy is improved significantly. The presented solutions for different time-varying computing problems exhibit the applicability of the proposed terminal networks.

**Keywords** Terminal neural networks, Inverse matrix, Generalized inverses, Path planning, Industrial manipulators

## 1 引言

递归神经网络计算具有并行计算的特点,能够高效、高精度地获得计算结果。Hopfield 于 20 世纪 80 年代提出了递归

神经网络模型<sup>[1]</sup>,这种神经网络被广泛用于解决实时优化问题。例如,文献[2]将非线性规划归结为递归神经网络的稳态解,文献[3]设计了动态梯度系统求解优化问题,文献[4]采用了对偶递归神经网络模型求解二次规划问题。

收稿日期:2018-10-11 返修日期:2019-04-06 本文已加入开放科学计划(OSID),请扫描上方二维码获取补充信息。

基金项目:国家自然科学基金(61573320)

This work was supported by the National Natural Science Foundation of China (61573320).

通信作者:孙明轩(mxsun@zjut.edu.cn)

在线矩阵计算是工程场合经常会遇到的问题<sup>[5]</sup>,矩阵分解与稀疏矩阵处理是常用的处理技巧<sup>[6-8]</sup>。时变矩阵计算是其中具有挑战性的矩阵计算问题,采用数值计算会占用大量的计算机资源。目前,研究主要集中于应用渐近收敛递归神经网络。对于时变矩阵求逆问题,文献[9]提出了全局渐近收敛递归神经网络模型,其最终结果渐近收敛于被求解问题的精确解,文献[10]提出了具有误差指数收敛的递归神经网络模型。基于一阶导数的有限差分公式,文献[11]推导了离散递归神经网络模型。

已发表的大量研究对各类时变矩阵计算问题给出了有效解决方案。当矩阵行满秩或列满秩时,需考虑矩阵广义逆的计算问题<sup>[12-13]</sup>。文献[14-15]证明了递归神经网络在线计算时变矩阵广义逆是有效的,而采用渐近收敛网络模型的求解过程收敛至精确解需无限长的时间,因此有限时间收敛神经网络模型更适用于求解时变矩阵计算问题。文献[16-18]提出了递归神经网络计算模型,其计算过程能够在有限时间内收敛,给出了时变矩阵计算问题更有效的解决方案。文献[19-20]给出了应用于求解 Lyapunov 方程以及一般线性矩阵方程的有限时间收敛递归神经网络模型。

冗余机械臂是指:对于给定的轨迹规划任务,机械臂所具有的自由度多于所需的自由度。对于冗余机械臂,给定末端执行器位置,关节向量有无穷多种组合,轨迹规划任务即求解每个时刻的关节向量,伪逆法是机械臂轨迹规划的传统方法<sup>[21-22]</sup>。文献[23]提出了将轨迹规划问题转换为二次规划,通过拉格朗日乘子(Lagrange Multiplier)法将二次规划转换为矩阵求逆问题,以递归神经网络求解。文献[24]给出了将不等式约束转换为等式约束的方法。文献[25]放宽了约束项的凸性要求。最近,噪声影响下的神经计算问题引起了人们的关注<sup>[26]</sup>,文献[27]考虑了噪声影响下的机械臂轨迹规划。

基于递归神经网络的神经计算具有十分广泛的应用背景。对于时变矩阵求逆、时变矩阵广义逆求解和冗余机械臂重复运动轨迹规划,目前提出的神经网络模型多具有渐近收敛特性,而有限时间收敛网络模型多具有无限值特性,且不易于实现。本文首先分析渐近网络在时变矩阵计算方面所能达到的性能,阐明引入终态神经网络模型的必要性,从而提出了一种新颖的有限值终态神经网络模型用于时变矩阵计算,并仿真验证了有限终态神经网络模型在时变矩阵计算方面的有效性,最后将这种神经网络计算方法应用于工业机器人轨迹规划。

## 2 有限值神经网络模型

考虑由下述动态特性方程描述的递归神经网络模型:

$$\dot{E} = -\rho S(E) \quad (1)$$

其中, $E$ 为 $n \times n$ 维误差矩阵; $\rho(\rho > 0)$ 为可调节的增益系数,用于调节网络的收敛速度; $S(\cdot)$ 为具有有限值的激活函数,是严格单调递增的奇函数,满足 $S(\cdot) = -S(-\cdot)$ 。

选取 Lyapunov 函数  $V(t) = E(t) \odot E(t)$ 。根据式(1),可将  $V(t)$  导数表示为:

$$\dot{V}(t) = -2\rho E(t) \odot S(E(t)) \quad (2)$$

其中, $\odot$ 表示 Hadamard 积。

由激活函数性质易知,该递归神经网络是渐近稳定的。为了便于分析其渐近收敛性能,定义  $t = kT + t_1$ ,其中, $T > 0$ , $k = 0, 1, 2, \dots, t_1 \in [0, T]$ 。因此, $E(t)$ 可表示为  $E_k(t_1)$ , $V(t)$ 可表示为  $V_k(t_1)$ 。

**定理 1** 对于由动态方程(1)描述的递归神经网络,选取激活函数为严格单调递增奇函数,可使得当  $k \rightarrow \infty$ 时, $E_k(t_1)$ 在  $[0, T]$ 上一致收敛。即:

$$\lim_{k \rightarrow \infty} E_k(t_1) = 0, \forall t_1 \in [0, T] \quad (3)$$

证明:由式(2)得:

$$\begin{aligned} V_{ij,k}(t_1) &= V_{ij,k}(0) - 2\rho \int_0^T E_{ij,0}(s) S(E_{ij,0}(s)) ds - \\ & 2\rho \int_0^T E_{ij,1}(s) S(E_{ij,1}(s)) ds - \dots - \\ & 2\rho \int_0^{t_1} E_{ij,k}(s) S(E_{ij,k}(s)) ds \end{aligned} \quad (4)$$

其中,令  $t_1 = T$ 。

$$V_{ij,K}(T) - V_{ij,0}(0) = -2\rho \sum_{k=0}^{K-1} \int_0^T E_{ij,k}(s) S(E_{ij,k}(s)) ds \quad (5)$$

经移项整理可得:

$$\begin{aligned} \sum_{k=0}^{K-1} \int_0^T E_{ij,k}(s) S(E_{ij,k}(s)) ds &= \frac{1}{2\rho} (V_{ij,0}(0) - V_{ij,K}(T)) \\ &\leq \frac{1}{2\rho} V_{ij,0}(0) \end{aligned} \quad (6)$$

由于  $V_{ij,0}(0)$  有界,利用级数收敛性必要条件可知:

$$\lim_{k \rightarrow \infty} \int_0^T E_{ij,k}(s) S(E_{ij,k}(s)) ds = 0 \quad (7)$$

为了证明  $E_{ij,k}(t_1)$  的收敛性,采用了反证法。假设存在  $t_1^* \in (0, T)$ ,当  $k \rightarrow \infty$ 时, $E_{ij,k}(t_1^*)$ 不收敛于零,则存在一个子序列  $\{k_i\}$ 和常数  $\epsilon > 0$ ,使得:

$$|E_{ij,k_i}(t_1^*)| \geq \sqrt{2\epsilon} \quad (8)$$

其中, $I$ 为单位矩阵。由式(2)可知, $E_{ij,k}(t_1)$ 在  $[0, T]$ 上一致有界,继而由式(1)推知  $\dot{E}_{ij,k}(t_1)$ 的一致有界性。因此, $E_k(t_1)$ 是等度连续的,即对于给定的  $\epsilon$ ,存在  $\delta \leq \min\{|t_1^*|, |T - t_1^*|\}$ ,且  $|t_1 - t_1^*| \leq \delta$ ,使得:

$$|E_{ij,k_i}(t_1) - E_{ij,k_i}(t_1^*)| \leq \sqrt{\frac{\epsilon}{2}} \quad (9)$$

因而,

$$|E_{ij,k_i}(t_1)| \geq |E_{ij,k_i}(t_1^*)| - |E_{ij,k_i}(t_1) - E_{ij,k_i}(t_1^*)| \geq \sqrt{\frac{\epsilon}{2}} \quad (10)$$

并且,

$$\begin{aligned} &\int_0^T E_{ij,k_i}(s) S(E_{ij,k_i}(s)) ds \\ &\geq \int_{t_1^* - \delta}^{t_1^* + \delta} E_{ij,k_i}(s) S(E_{ij,k_i}(s)) ds \\ &\geq \frac{\epsilon}{2} \cdot 2\delta = \epsilon\delta \end{aligned} \quad (11)$$

其中,当  $k \rightarrow \infty$ 时,  $\int_0^T E_{ij,k_i}(s) S(E_{ij,k_i}(s)) ds$ 不能收敛于零,这与式(7)矛盾。由此证明了  $E_{ij,k_i}(s)$ 的一致收敛性。对于  $t_1^* = 0$ 和  $t_1^* = T$ 的证明可以类似地给出。

定理 1 表明,对于动态方程(1),当  $k \rightarrow \infty$ 时,误差变量  $E_k(t_1)$ 在  $[0, T]$ 上是一致收敛至零的。这类具有渐稳性质的动态特性刻画的递归神经网络模型被称为渐近收敛网络。具

体地,网络的渐近收敛性质意味着:

P1:只有当  $k$  足够大时,  $E_k(t_1)$  才能充分接近于零;

P2:当  $k$  较小,且  $E_0(0) \gg 0$  时,  $E_k(t_1) \gg 0$ 。

由 P1 可以看出,具有渐近收敛的网络在解决时变计算问题时,只有当  $k$  足够大时,才能得到较准确的结果;但由 P2 可知,渐近收敛网络无法给出当  $k$  较小时、较精确的解。因此,渐近收敛网络适于求解时不变计算问题,但对于时变计算问题,仍存在收敛时间无限、初始阶段计算精度不高等诸多问题。

为了有效求解时变计算问题,本文提出了一种终态神经网络模型,其动态特性方程为:

$$\dot{E}_{ij} = -\kappa \sqrt{\frac{|E_{ij}|}{|E_{ij}|+1/\delta}} \operatorname{sgn}(E_{ij}) \quad (12)$$

其中,  $\kappa > 0, 0 < \delta < 1$ 。

式(12)对应的激活函数可表示为:

$$S(E_{ij}) = \sqrt{\frac{|E_{ij}|}{|E_{ij}|+1/\delta}} \operatorname{sgn}(E_{ij}) \quad (13)$$

该函数严格单调递增,且满足  $S(-\cdot) = -S(\cdot)$ ,则式(12)所描述的网络具有渐近收敛性。进一步地,容易证明其有限时间收敛性。当  $E_{ij} > 0$  时,对式(12)两边积分可得:

$$\frac{1}{\kappa} \int_{E_{ij}(0)}^{E_{ij}(t)} \sqrt{\frac{|E_{ij}|+1/\delta}{|E_{ij}|}} dE_{ij} = - \int_0^t ds \quad (14)$$

令  $x = \frac{1}{\delta |E_{ij}|}$ ,式(14)可写成:

$$\frac{1}{\delta \kappa} \int_{x(0)}^{x(t)} \frac{\sqrt{1+x}}{x^2} dx = \int_0^t dt \quad (15)$$

两端求积分:

$$t = -\frac{1}{\delta \kappa} \left[ \frac{\sqrt{1+x}}{x} - \frac{1}{2} \ln \left| \frac{\sqrt{1+x}-1}{\sqrt{1+x}+1} \right| \right] \Bigg|_{x(0)}^{x(t)} \quad (16)$$

注意到:

$$\lim_{x \rightarrow \infty} \frac{\sqrt{1+x}}{x} - \frac{1}{2} \ln \left| \frac{\sqrt{1+x}-1}{\sqrt{1+x}+1} \right| = 0 \quad (17)$$

对于式(16),令  $t = t_s, x(t_s) = \infty$ ,可将  $E_{ij}(t)$  的收敛时间表示为:

$$t_s = \frac{1}{\delta \kappa} \left[ \frac{\sqrt{1+\mu}}{\mu} + \frac{1}{2} \ln \left| \frac{\sqrt{1+\mu}+1}{\sqrt{1+\mu}-1} \right| \right] \quad (18)$$

其中,  $\mu = 1/(\delta |E_{ij}(0)|)$ 。

同理,易得  $E_{ij}(t) < 0$  时的相同结果。由于  $\mu = 1/(\delta |E_{ij}(0)|)$ ,而  $E_{ij}(0)$  有限,因此式(18)是有限的。应该指出的是选择不同的参数,其收敛速率也不同。设置的参数  $\kappa$  越大,收敛时间就越小;通过调整  $\delta$ ,可有效改变激活函数的形状(见图1)。

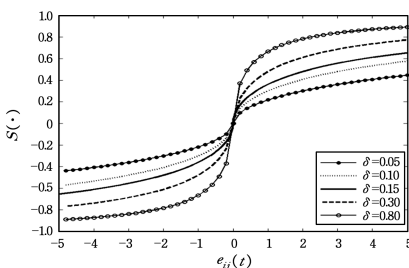


图1 不同  $\delta$  取值时的激活函数  $S(\cdot)$

Fig. 1 Activation function  $S(\cdot)$  with different  $\delta$

由于这种有限时间的收敛性,我们称动态方程(12)所描述的网络为终态网络。值得说明的是,与通常的终态网络不同,式(13)中的激活函数  $S(E_{ij})$  取有限值,在实际应用中易于实现。

### 3 神经网络计算

相对渐近网络而言,终态网络适于求解时变矩阵计算问题。本节将其用于时变矩阵逆以及时变矩阵广义逆的求解问题,以便进一步说明终态网络的计算性能。

#### 3.1 时变矩阵逆求解

时变矩阵逆可描述为求解如下方程:

$$A(t)X(t) = I \quad (19)$$

其中,  $A(t)$  是  $n \times n$  维光滑非奇异时变矩阵,  $X(t)$  是待求解的  $n \times n$  维时变矩阵。由于  $A(t)$  光滑非奇异,不失一般性,假定  $A(t)$  以及  $\dot{A}(t)$  已知或者可量测,则存在  $A(t)$  的逆矩阵  $A^{-1}(t)$ 。为求解  $X(t)$ ,将误差函数定义为:

$$E(t) = A(t)X(t) - I \quad (20)$$

根据该误差函数的定义,依照如式(1)所示递归神经网络的动态特性方程,可获得如下递归神经网络模型:

$$\dot{A}(t)X(t) + A(t)\dot{X}(t) = -\kappa S(E(t)) \quad (21)$$

其方框图如图2所示。经计算,得到  $t_s$  时刻之后的  $X(t)$  即为矩阵  $A(t)$  逆的精确解,即:

$$X(t) = X^*(t) (= A^{-1}(t)), t \geq t_s \quad (22)$$

利用终态网络获得的解是  $t_s$  时刻之后的精确解,而渐近网络需要足够大的计算时间,才能获得更精确的解。

#### 3.2 时变矩阵广义逆求解

对于  $m \times n$  维时变矩阵  $A(t), m \neq n$ ,其广义逆满足条件:

$$\begin{cases} A(t)X(t)A(t) = A(t) \\ X(t)A(t)X(t) = X(t) \\ X(t)A(t) \text{ 与 } A(t)X(t) \text{ 互为对称矩阵} \end{cases} \quad (23)$$

其中,  $X(t)$  被称为时变矩阵  $A(t)$  的广义逆矩阵,记  $X(t) = A^+(t)$ 。时变矩阵的广义逆矩阵总是存在且唯一,对于  $t \in [0, +\infty)$ ,如果  $A(t)$  总是满秩,即  $\operatorname{rank}(A(t)) = \min(m, n)$ ,矩阵  $A(t)$  的广义逆可表示为:

$$A^+(t) = \begin{cases} (A^T(t)A(t))^{-1}A^T(t), & m > n \\ A^T(t)(A(t)A^T(t))^{-1}, & m < n \end{cases} \quad (24)$$

其中,当  $m > n$  时,  $A^+(t)$  称为左逆矩阵;当  $m < n$  时,  $A^+(t)$  称为右逆矩阵。

为求解广义逆矩阵,根据式(24)可定义误差函数:

(1) 当  $m > n$  时,  $X(t) = (A^T(t)A(t))^{-1}A^T(t)$  两端左乘  $A^T(t)A(t)$ ,得  $A^T(t)A(t)X(t) = A^T(t)$ 。

这样,可定义:

$$E(t) = A^T(t)A(t)X(t) - A^T(t) \quad (25)$$

(2) 当  $m < n$  时,  $X(t) = A^T(t)(A(t)A^T(t))^{-1}$  两端右乘  $A(t)A^T(t)$ ,得  $X(t)A(t)A^T(t) = A^T(t)$ 。

这样,可定义:

$$E(t) = X(t)A(t)A^T(t) - A^T(t) \quad (26)$$

依照动态特性方程(1)可构造终态递归神经网络进行广义逆求解。具体地,根据式(25)和式(26)定义的误差函数分两种情况讨论。

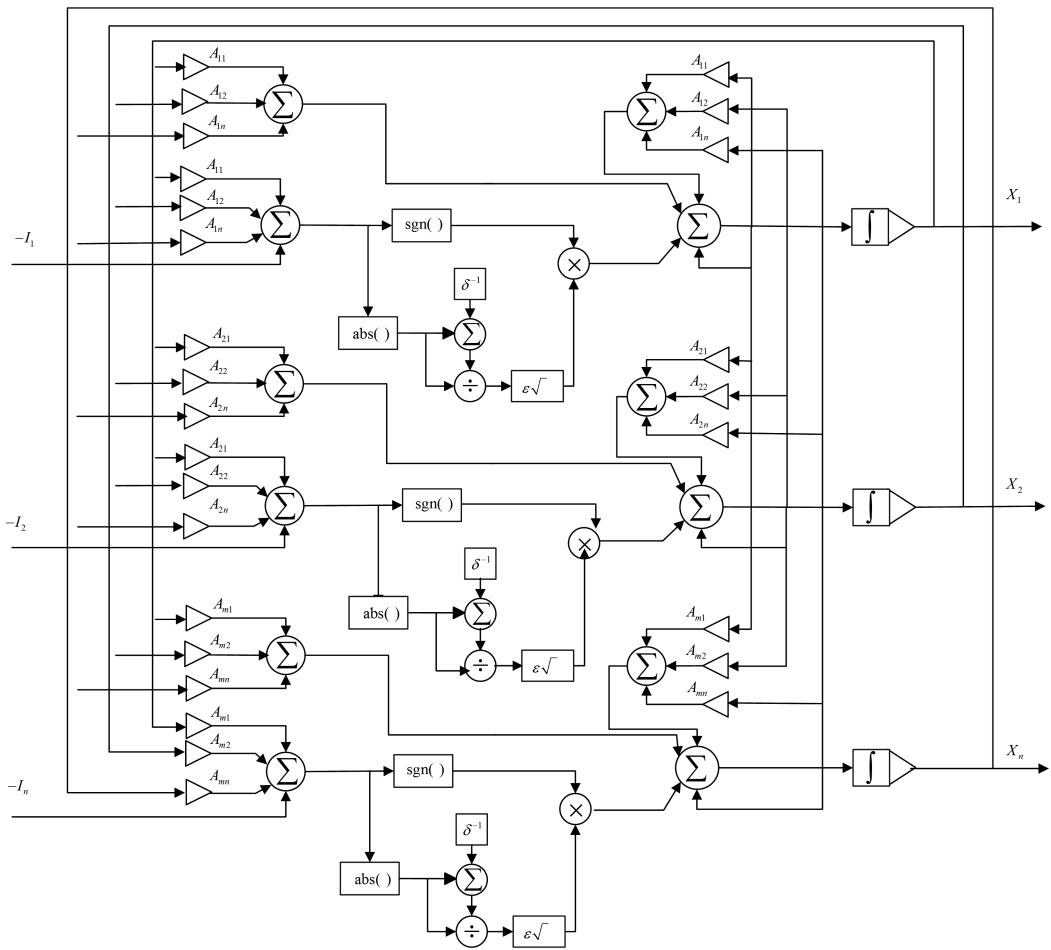


图2 递归神经网络模型

Fig.2 Recurrent neural network model

(1)对左逆矩阵求解,将式(25)代入动态方程(1),可得递归神经网络模型:

$$\begin{aligned} \dot{A}^T(t)A(t)X(t) + A^T(t)\dot{A}(t)X(t) + A^T(t)A(t)\dot{X}(t) - \\ \dot{A}^T(t) = -\kappa S(E(t)) \end{aligned} \quad (27)$$

(2)对右逆矩阵求解,将式(26)代入动态方程(1),可得递归神经网络模型:

$$\begin{aligned} \dot{X}(t)A(t)A^T(t) + X(t)\dot{A}(t)A^T(t) + X(t)A(t)\dot{A}^T(t) - \\ \dot{A}^T(t) = -\kappa S(E(t)) \end{aligned} \quad (28)$$

给定误差初值,运行网络(27)和网络(28), $E(t)$ 在 $t_s$ 时刻之后收敛于零,即得到矩阵 $A(t)$ 的广义逆 $X(t)$ 。

利用终态网络计算获得广义逆矩阵的解,所需时间 $t_s$ 是有限的,而渐近网络却需要足够长的计算时间。

### 3.3 数值算例

下面将完成矩阵求逆和矩阵广义逆求解的数值算例,给出有限值终态神经网络和渐近神经网络在计算过程中收敛速度的对比结果。

式(19)中的时变矩阵为:

$$A(t) = \begin{bmatrix} \sin(3t) & \cos(3t) \\ -\cos(3t) & \sin(3t) \end{bmatrix}$$

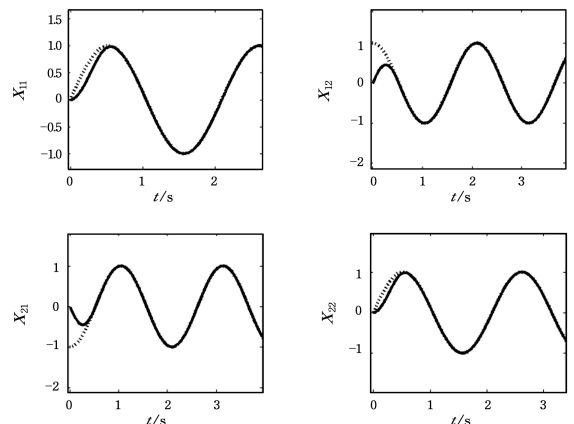
容易推导其逆矩阵,以便检验计算结果:

$$X^*(t) = A^{-1}(t) = \begin{bmatrix} \sin(3t) & -\cos(3t) \\ \cos(3t) & \sin(3t) \end{bmatrix}$$

初始值可选任意值,这里取:

$$X(0) = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$$

终态神经网络中的参数取为 $\kappa=5, \delta=0.5$ ,计算结果如图3所示,计算值 $X(t)$ 收敛于理想值 $X^*(t)$ 。



注:虚线为真值,实线为计算值

图3 矩阵求逆真值与计算值的对比

Fig.3 Comparison between true and resultant values of matrix inversion

为了便于比较,我们考虑如下渐近网络动态方程:

$$\dot{E}(t) = -\rho E(t) \quad (29)$$

其中,分别取参数  $\rho=5,8,10$ 。引入指标函数  $J_E(t) = \|E(t)\|_F$ , 以便对照。图 4 给出了分别采用渐近网络和终态网络计算时获得的指标函数。根据式(18)可计算出收敛时刻  $t_s = 0.609\text{ s}$ , 此收敛时刻与图 4 所示的收敛时刻的计算值是一致的。图中渐近网络的指标值  $J_E$  尚存在误差,而在终态网络中  $J_E$  在  $t_s$  内收敛至零,且具有较快的收敛速度。增大参数  $\rho$ ,渐近网络的收敛性能无显著改善。

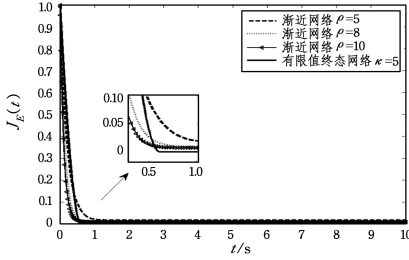


图 4 矩阵求逆误差

Fig. 4 Errors of matrix inversion

对于矩阵广义逆解算例,由于左逆求解与右逆求解的过程雷同,本文只考虑  $m > n$  的情况。考虑下述时变矩阵:

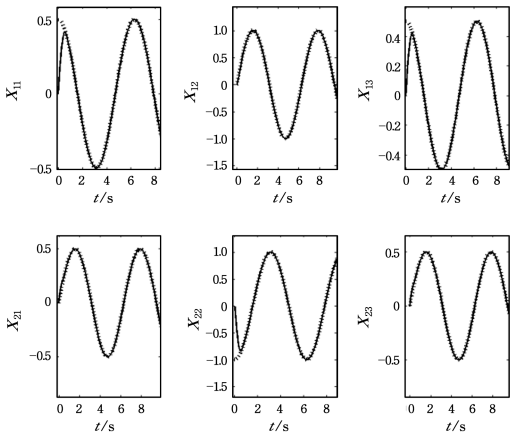
$$A(t) = \begin{bmatrix} \cos(t) & \sin(t) \\ \sin(t) & -\cos(t) \\ \cos(t) & \sin(t) \end{bmatrix}$$

为了方便验证,给出其左逆矩阵:

$$A^+(t) = \begin{bmatrix} 0.5\cos(t) & \sin(t) & 0.5\cos(t) \\ 0.5\sin(t) & -\cos(t) & 0.5\sin(t) \end{bmatrix}$$

选取初值  $X(0) = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$ ,取网络动态方程中的参数

$\kappa=5, \delta=0.5$ ,图 5 给出了运行有限值神经网络模型(27)得到的矩阵  $A(t)$  广义逆的解。可以看出,计算值  $X(t)$  收敛于理想值  $A^+(t)$ 。



注:虚线为真值,实线为计算值

图 5 矩阵左逆真值与计算值的对比

Fig. 5 Comparison between true and resultant values of left inverse

为了比较终态网络和渐近网络的性能,考虑式(29)所示的渐近网络动态方程,其中分别取参数  $\rho=5,8,10$ 。图 6 给出了分别运行渐近网络和终态网络获得的指标值。根据式(18)可计算出收敛时刻  $t_s = 0.609\text{ s}$ ,此收敛时刻与图 6 所示的收

敛时刻的计算值是一致的。图中渐近网络的指标值  $J_E$  尚存在误差,而终态网络的指标值  $J_E$  在有限时间内收敛至零,具有较快的收敛速度。同样,增大参数  $\rho$ ,渐近网络的收敛性能仍无显著改善。

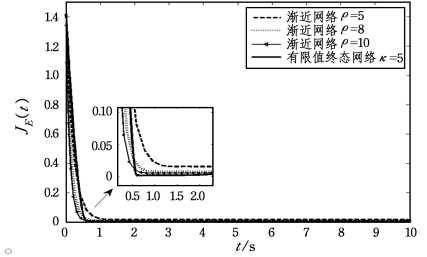


图 6 矩阵左逆误差

Fig. 6 Errors of left inverse

### 4 冗余机械臂重复轨迹规划

工业机械臂的笛卡尔空间终端轨迹与关节空间向量之间的关系为:

$$r(t) = f(\theta(t)) \quad (30)$$

其中,  $r(t) \in R^m$  是末端执行器的位置向量,  $\theta(t) \in R^n$  是关节向量,  $f(\theta(t))$  是关于  $\theta(t)$  的光滑非线性映射。对于冗余机械臂 ( $n > m$ ),通常转换到速度层进行轨迹规划。对式(30)两端关于  $t$  进行求导,得:

$$\dot{r}(t) = J(\theta)\dot{\theta}(t) \quad (31)$$

其中,  $J(\theta(t)) = \partial f(\theta(t))/\partial \theta$  是  $m \times n$  维雅克比矩阵;  $\dot{r}(t)$  是笛卡尔空间末端执行器的速度向量;  $\dot{\theta}(t)$  是关节速度向量。设计了一种冗余机械臂重复运动规划方案,该方案可描述为二次规划问题:

$$\min_{\dot{\theta}(t)} (\dot{\theta}(t) + z)^T (\dot{\theta}(t) + z) / 2 \quad (32)$$

$$\text{s. t. } J(\theta(t))\dot{\theta}(t) = \dot{r}^*(t) + \kappa_v(r^*(t) - f(\theta(t))) \quad (33)$$

其中,  $r^*(t)$  为期望终端轨迹;  $z = \kappa_s S(\theta(t) - \theta^*(0))$ ,  $\kappa_s > 0$  为可调参数;由于允许终端执行器初始位置偏离期望轨迹,引入附加项  $r^*(t) - f(\theta(t))$ ,  $\kappa_v > 0$  为可调增益。依照拉格朗日乘子法(Lagrange Multiplier)求解二次规划,定义如下拉格朗日函数:

$$L(\dot{\theta}(t), \lambda(t), t) = z^T \dot{\theta}(t) + \dot{\theta}(t)^T \dot{\theta}(t) / 2 + z^T z + \lambda^T (J(\theta(t))\dot{\theta}(t) - \dot{r}^*(t) + \kappa_v(r^*(t) - f(\theta(t)))) \quad (34)$$

其中,  $\lambda(t)$  为  $m+n$  维拉格朗日乘子向量。对该函数关于  $\dot{\theta}(t)$  和  $\lambda(t)$  分别求偏导,并令偏导为零,可得:

$$W(t)Y(t) = v \quad (35)$$

其中,

$$W = \begin{pmatrix} I & J(\theta(t))^T \\ J(\theta(t)) & 0 \end{pmatrix}, Y = \begin{pmatrix} \dot{\theta}(t) \\ \lambda(t) \end{pmatrix}$$

$$v = \begin{pmatrix} -z \\ \dot{r}^*(t) + \kappa_v(r^*(t) - f(\theta(t))) \end{pmatrix}$$

因此,二次规划问题求解转换成了式(35)所示的方程求解。

运用本文提出的有限值终态神经网络求解,应首先定义误差变量:

$$E = W(t)Y(t) - v \quad (36)$$

当  $E=0$  时,矩阵方程(35)成立,利用此时的解  $Y$  可以进一步得到机械臂关节角。根据误差定义(36),由动态方程(12)可建立终态网络模型:

$$\dot{Y}(t) = \dot{W}(t)Y(t) + W(t)\dot{Y}(t) - \dot{v}(t) + \kappa S(E(t)) + \dot{Y}(t) \quad (37)$$

本文以 PUMA560 为仿真对象,该机械臂连杆长度为  $L = [0.4318, 0.0203, 0.4318, 0.25625]^T$  m。期望轨迹为边长为 0.4m 的等边三角形,运行周期  $T = 6$  s,其初始点为原点。机械臂期望回绕的关节角度为:

$$\theta^*(0) = [0, -\frac{\pi}{4}, 0, \frac{\pi}{2}, -\frac{\pi}{4}, 0]^T$$

末端执行器初始点不在三角形的轨迹上,取实际末端执行器的初始位置为:

$$\theta(0) = [0, -0.9, 0, 1.5, -0.7, 0]^T$$

机械臂末端执行器的运动轨迹如图 7 所示。可以看出,虽然末端执行器的初始位置不在期望轨迹上,但回绕时的实际轨迹与期望轨迹吻合;关节角和角速度轨迹如图 8 和图 9 所示。末端执行器的终值位置误差精度在 3 个方向(XYZ 轴)上达到了  $10^{-7}$ ,如图 10 所示。

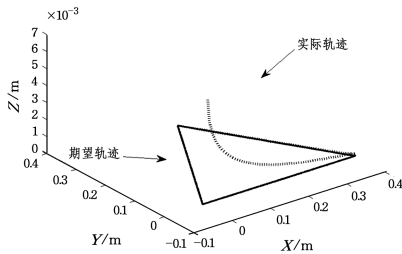


图 7 机械臂末端执行器的运动轨迹  
Fig. 7 Trajectory of end effector

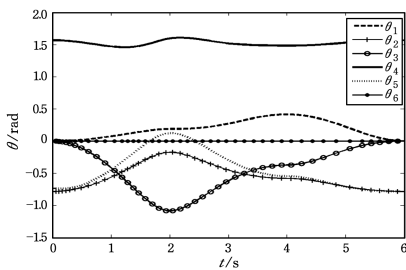


图 8 机械臂各个关节角的轨迹  
Fig. 8 Joint angle trajectories

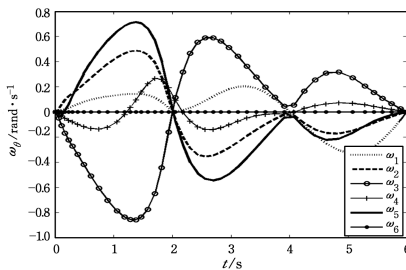


图 9 机械臂各个关节角的速度轨迹  
Fig. 9 Joint velocity trajectories

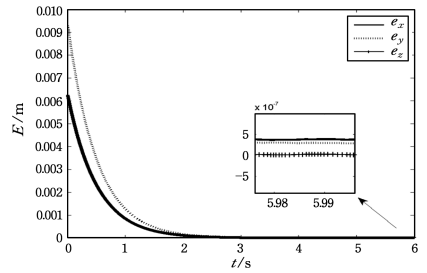


图 10 机械臂末端执行器位置的误差轨迹  
Fig. 10 Position error trajectory of end effector

图 11 给出了有限值终态网络和渐近网络的误差指标值 ( $J_E(t) = \|W(t)Y(t) - v(t)\|_2$ )。表 1 列出了  $T = 6$  s 时的关节角与期望关节角的偏差,应用终态网络时最大偏差为  $3.4 \times 10^{-5}$  rad;应用渐近网络时,分别取  $\rho$  为 5, 8, 10 时,最大偏差分别为  $1.2 \times 10^{-3}$  rad,  $5.7 \times 10^{-4}$  rad 和  $3.6 \times 10^{-4}$  rad。

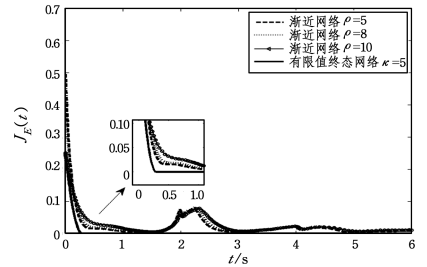


图 11 机械臂误差轨迹  
Fig. 11 Error trajectories of manipulator

表 1 关节角的回绕偏差

Table 1 Deviations of joint angles

网络求解器/ rad	渐近网络			终态网络
	$\rho=5$	$\rho=8$	$\rho=10$	$\kappa=5$
$\theta_1(6) - \theta_{d1}(0)$	$-1.2 \times 10^{-3}$	$-5.7 \times 10^{-4}$	$-3.6 \times 10^{-4}$	$8.6 \times 10^{-6}$
$\theta_2(6) - \theta_{d2}(0)$	$-2.0 \times 10^{-4}$	$-1.0 \times 10^{-4}$	$-5.5 \times 10^{-5}$	$-1.5 \times 10^{-5}$
$\theta_3(6) - \theta_{d3}(0)$	$2.6 \times 10^{-4}$	$1.3 \times 10^{-4}$	$-8.1 \times 10^{-5}$	$-2.5 \times 10^{-6}$
$\theta_4(6) - \theta_{d4}(0)$	$2.1 \times 10^{-4}$	$6.4 \times 10^{-5}$	$3.4 \times 10^{-5}$	$7.0 \times 10^{-6}$
$\theta_5(6) - \theta_{d5}(0)$	$1.0 \times 10^{-4}$	$2.6 \times 10^{-5}$	$1.7 \times 10^{-5}$	$3.4 \times 10^{-5}$
$\theta_6(6) - \theta_{d6}(0)$	0	0	0	0

**结束语** 本文提出了一种根式终态神经网络,从现有的有限时间系统理论来看,这种网络是新颖的。它具有有限值与有限时间收敛特性,本文推导了收敛时间的具体表达式,函数值有界的网络模型在实际中易实现。该网络能有效求解时变矩阵逆和广义逆问题,与渐近网络相比,终态网络的收敛时间和收敛精度均得以明显的改善。对于冗余机械臂轨迹规划,本文提出了终态吸引性能指标,并利用终态网络模型求解规划方案。与渐近性能指标及渐近网络求解方案相比,本文给出的终态网络方案具有收敛速度快和精度高的特点。

参考文献

[1] HOPFIELD J J. Neurons with graded response have collective computational properties like those of two-state neurons[J]. Proceedings of the National Academy of Sciences, 1984, 81(10): 3088-3092.  
[2] KENNEDY M P, CHUA L O. Neural networks for nonlinear programming [J]. IEEE Transactions on Circuits and Systems, 1988, 35(5): 554-562.

- [3] RODRIGUEZ-VAZQUEZ A, DOMINGUEZ-CASTRO R, RUEDA A, et al. Nonlinear switched capacitor neural networks for optimization problems [J]. *IEEE Transactions on Circuits and Systems*, 1990, 37(3):384-398.
- [4] LIU S, WANG J. A simplified dual neural network for quadratic programming with its KWTA application [J]. *IEEE Transactions on Neural Networks*, 2006, 17(6):1500-1510.
- [5] ROBERT H, STURGES. Analog matrix inversion (robot kinematics) [J]. *IEEE Journal on Robotics & Automation*, 2002, 4(2):157-162.
- [6] YEUNG K S, KUMBI F. Symbolic matrix inversion with application to electronic circuits [J]. *IEEE Transactions on Circuits & Systems*, 1988, 35(2):235-238.
- [7] EI-AMAWY A. A systolic architecture for fast dense matrix inversion [J]. *IEEE Transactions on Computers*, 1989, 38(3):449-445.
- [8] WANG Y Q, GOOI JH B. New ordering methods for space matrix inversion via diagonalization [J]. *IEEE Transactions on Power Systems*, 1997, 12(3):1298-1305.
- [9] ZHANG Y N, JIANG D, WANG J. A recurrent neural network for solving Sylvester equation with time-varying coefficients [J]. *IEEE Transactions on Neural Networks*, 2002, 13(5):1053-1063.
- [10] ZHANG Y N, GE S Z. Design and analysis of a general recurrent neural network model for time-varying matrix inversion [J]. *IEEE Transactions on Neural Networks*, 2005, 16(6):1477-1490.
- [11] SHI Y, QIU B, CHEN D, et al. Proposing and validation of new four-point finite-difference formula with manipulator application [J]. *IEEE Transactions on Industrial Informatics*, 2018, 14(4):1323-1333.
- [12] COURRIEU P. Fast computation of moore-penrose inverse matrices [J]. *Neural Information Processing Letters and Reviews*, 2005, 8(2):25-29.
- [13] GUO W B, HUANG T. Method of elementary transformation to compute Moore-Penrose inverse [J]. *Applied Mathematics and Computation*, 2010, 216(5):1614-1617.
- [14] WANG J. Recurrent neural network for computing pseudoinverses of rank-deficient matrices [J]. *Siam Journal on Scientific Computing*, 1997, 18(5):1479-1493.
- [15] WU G, WANG J, HOOTMAN J. A recurrent neural network for computing pseudoinverse matrices [J]. *Mathematical and Computer Modelling*, 1994, 20(1):13-21.
- [16] LI S, LI Y M, WANG Z. A class of finite-time dual neural networks for solving quadratic programming problems and its winners-take-all application [J]. *Neural Networks*, 2013, 39(39):27-39.
- [17] LI S, LI Y M. Nonlinearly activated neural network for solving time-varying complex Sylvester equation [J]. *IEEE Transactions on Cybernetics*, 2014, 44(8):1397-1407.
- [18] SUN M X, YU X F, KONG Y. Terminal neural computing; finite-time convergence and the related application [J]. *Journal of Zhejiang University of Technology*, 2015, 43(3):311-317.
- [19] XIAO L, LIAO B, LI S, et al. Design and analysis of FTZNN applied to the real-time solution of a nonstationary Lyapunov equation and tracking control of a wheeled mobile manipulator [J]. *IEEE Transactions on Industrial Informatics*, 2018, 14(1):98-105.
- [20] XIAO L, LIAO B L, LI S, et al. Nonlinear recurrent neural networks for finite-time solution of general time-varying linear matrix equations [J]. *Neural Networks*, 2018, 98:102-113.
- [21] HOLLERBACH J, SUH K. Redundancy resolution of manipulators through torque optimization [J]. *IEEE Journal of Robotics and Automation*, 1987, 3(4):308-316.
- [22] TCHON K, JAKUBIAK J. A repeatable inverse kinematics algorithm with linear invariant subspaces for mobile manipulators [J]. *IEEE Transactions on Systems Man and Cybernetics Part B*, 2005, 35(5):1051-1057.
- [23] ZHANG Y N, WANG J, XIA Y S. A dual neural network for redundancy resolution of kinematically redundant manipulators subject to joint limits and joint velocity limits [J]. *IEEE Transactions on Neural Networks*, 2003, 14(3):658-667.
- [24] GUO D S, ZHANG Y N. Acceleration-level inequality-based MAN scheme for obstacle avoidance of redundant robot manipulators [J]. *IEEE Transactions on Industrial Electronics*, 2014, 61(12):6903-6914.
- [25] LI S, ZHANG Y N, LONG J. Kinematic control of redundant manipulators using neural networks [J]. *IEEE Transactions on Neural Networks and Learning Systems*, 2017, 28(10):2243-2254.
- [26] JIN L, ZHANG Y, LI S. Integration-enhanced zhang neural network for real-time-varying matrix inversion in the presence of various kinds of noises [J]. *IEEE Transactions on Neural Networks and Learning Systems*, 2016, 27(12):2615-2627.
- [27] LI S, WANG H Q, RAFIQUE U M. A novel recurrent neural network for manipulator control with improved noise tolerance [J]. *IEEE Transactions on Neural Networks and Learning Systems*, 2018, 29(5):1908-1918.



**SUN Ming-xuan**, born in 1961, Ph. D., professor, Ph. D supervisor. His main research interests include learning systems and neural computing.