



计算机科学

COMPUTER SCIENCE

用于时变矩阵计算的固定时间递归神经网络及其在重复运动规划中的应用

李杏, 仲国民

引用本文

李杏, 仲国民. 用于时变矩阵计算的固定时间递归神经网络及其在重复运动规划中的应用[J]. 计算机科学, 2024, 51(8): 324-332.

LI Xing, ZHONG Guomin. Fixed-time Recurrent Neural Networks for Time-variant Matrix Computing and Its Application in Repeatability Motion Planning [J]. Computer Science, 2024, 51(8): 324-332.

相似文章推荐 (请使用火狐或 IE 浏览器查看文章)

Similar articles recommended (Please use Firefox or IE to view the article)

[面向铣削机器人的低成本精确标定及刀路拟合方法](#)

Low Cost Accurate Calibration and Tool Path Fitting Method for Milling Robot

计算机科学, 2022, 49(7): 187-195. <https://doi.org/10.11896/jsjcx.210500135>

[有限值终态递归神经网络计算](#)

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

计算机科学, 2020, 47(1): 212-218. <https://doi.org/10.11896/jsjcx.181001898>

[基于终态神经网络的冗余机械臂重复运动规划](#)

Repeatability Motion Planning of Redundant Manipulators Based on Terminal Neural Networks

计算机科学, 2018, 45(12): 201-205. <https://doi.org/10.11896/j.issn.1002-137X.2018.12.033>

[Sylvester时变矩阵方程求解的终态神经网络算法](#)

Terminal Neural Network Algorithm for Solution of Time-varying Sylvester Matrix Equations

计算机科学, 2018, 45(10): 207-211. <https://doi.org/10.11896/j.issn.1002-137X.2018.10.038>

用于时变矩阵计算的固定时间递归神经网络及其在重复运动规划中的应用

李杏 仲国民

浙江工业大学信息工程学院 杭州 310023

(xingli@zjut.edu.cn)

摘要 文中提出了具有对数调节时间的固定时间收敛递归神经网络(Recurrent Neural Network, RNN)模型,用于求解时变矩阵计算问题。设计并详细分析了两个新颖的RNN模型,推导出在给定初始条件下模型调节时间函数的精确表达式;并给出任意初始条件下调节时间函数的上界。相比现有的固定时间收敛的RNN模型,两个新颖的模型具有对数调节时间,其调节时间上界更小,收敛速度更快。考虑到初始误差实际上在一个有界的区域内,给出RNN模型半全局对数调节时间函数,并由此推导出半全局意义上的调节时间函数的上界。采用RNN模型半全局调节时间上界的倒数,提出半全局预定时间收敛到精确解的改进RNN模型,其预定时间是一个可调参数。给出了所提RNN模型对时变Lyapunov方程和时变Sylvester方程求解的仿真结果,并将其应用于具有初始误差的冗余机械臂的重复运动规划,进一步验证了所提RNN模型的有效性。

关键词: 时变神经计算;对数调节时间;固定/预定时间收敛;工业机器人;重复运动规划

中图分类号 TP241

Fixed-time Recurrent Neural Networks for Time-variant Matrix Computing and Its Application in Repeatable Motion Planning

LI Xing and ZHONG Guomin

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

Abstract Fixed-time recurrent neural network(RNN) models with logarithmic settling time are proposed for solving time-variant neural computing problems. Two novel RNN models are designed and analyzed in detail, deriving the explicit expressions of settling time functions and providing the upper bounds of the settling times under any initial condition. Compared with the existing RNN models with fixed-time convergence, the two novel models with logarithmic settling time have a smaller upper bound on the settling time and faster convergence speeds. Taking into account initial conditions located within a region with a definite finite radius, the settling time functions of the RNN models with logarithmic settling time are given, and the upper bounds on the settling time functions in the semi-global sense are derived. Modified RNN models adopt the inverse of the bound to ensure that the semi-global predefined time converges to the exact solution, and its prescribed time is an adjustable parameter. Simulation results of the proposed RNN model for solving time-variant Lyapunov and Sylvester equations are given. The proposed RNNs are applied to the repetitive motion planning of a redundant manipulator with initial errors, and numerical results are presented to verify the effectiveness of the proposed RNN models.

Keywords Time-variant neural computing, Logarithmic settling time, Fixed/predefined-time convergence, Industrial manipulators, Repetitive motion planning

1 引言

矩阵方程的求解问题广泛存在于许多科学研究及工程领域中。针对定常矩阵的计算问题,模拟数值的方法可以有效地求解矩阵方程的数值解^[1]。然而,实际应用中很多问题是时变的,数值算法的串行处理特性决定了它适用于小规模的不变问题的求解,并不能很好地解决大规模的时变矩阵问题。与数值算法不同,递归神经网络(RNN)是一种全连接

状态的神经网络动力学模型,能够满足大规模实时计算的需求,已经成为在线计算的一个有效工具^[2]。文献[3]提出了一种单隐含层相互连接的RNN模型,该模型被广泛应用于各种问题的求解中,如线性方程^[4-5]、非线性规划问题^[6]和线性和二次规划问题^[7]。基于递归神经网络的神经计算作为主要的计算方法被广泛应用于机械臂的运动规划中。冗余机械臂具有更多的自由度,因此其运动规划容易满足各种操作需求,然而其冗余特性会使得机械臂的同一个逆运动学问题可能

到稿日期:2023-05-09 返修日期:2023-10-12

基金项目:国家自然科学基金(62073291)

This work was supported by the National Natural Science Foundation of China(62073291).

通信作者:仲国民(zgm@zjut.edu.cn)

存在无穷多的解。文献[8]采用伪逆法解决冗余度解析问题。当机械臂末端执行器在其工作空间中沿封闭路径运动时,基于伪逆的方法可能导致所谓的关节角漂移问题(亦称为非重复问题)[9]。对于解决冗余度解析问题和非重复运动问题,在线优化方法是非常有效的。文献[10]将冗余度解析问题进一步表述为二次规划问题,通过拉格朗日乘子法转化为求解时变矩阵问题。在线求解时变矩阵问题时涉及大量的动态变量,理论解随时间变化,因此可能带来两个挑战:1)如何在每一个时刻都提供高精度解?2)如何在保证精度的基础上处理大规模的计算问题?为了有效地解决时变计算问题,研究者们提出了一种隐式误差动力学 RNN 模型,通过零化相应的误差函数来求解各种时变矩阵问题,如 Sylvester 方程[11]、冗余解析问题[12]以及矩阵反演[13]。隐式误差动力学保证了神经网络计算解的存在性和全局指数收敛性,这种连续时间神经网络计算方法对于求解时变优化问题是有效的。

收敛速度是时变矩阵计算问题的一个关键指标。渐近收敛的 RNN 模型收敛至精确解需无限长的时间,且无法得到初始区间上的解。因此需要设计一个在实时计算中更高效准确的求解方案,以打破计算时间浪费和无法得到初始区间的解等局限性。在已发表的文献中,为了进一步提升神经网络求解时变问题的性能,有限时间收敛的 RNN 模型受到关注。优化 RNN 模型的结构、选择合适的激活函数(Activation Functions, AFs)是 RNN 模型能够在有限时间内得到精确解的关键。文献[14]提出了一种带符号双幂次激活函数(Sign-bi-power AF)来提高 RNN 收敛速度,并且给出了相应的调节时间函数。随后,多种基于单调递增奇函数的激活函数被提出,相应设计的有限时间收敛 RNN 模型被广泛用于求解时变线性矩阵方程[15]、Lyapunov 方程[16-17]、时变矩阵反演[18],以及时变二次优化问题[19]等。目前提出的有限时间收敛 RNN 模型大多具有无限值特性。文献[20]设计了新颖的有限值终态 RNN 模型;文献[21]总结了各种类型的 RNN 模型,并分析了有限值激活函数的有限时间收敛性,证明了有限值终态 RNN 模型求解时变问题的有效性。有限时间收敛 RNN 模型的特点是其稳定时间依赖于初始条件。

对于时变矩阵问题的求解,若收敛时间能预先确定,则可以大大提高计算精度。文献[22]引入了固定时间稳定性的概念,其收敛时间函数的上界与初始条件无关,因此特别适用于时变计算问题的求解。文献[23]给出了精确估计调节时间的界,同时,提出的神经网络模型可以实现固定时间稳定性。文献[24]提出了预定时间稳定性的概念,建立了精确的调节时间函数,并推导了调节时间函数与系统参数有显式关系的最小上界。为了进一步提高收敛速度并且改善系统性能,文献[25-26]提出了收敛时间与 RNN 的初始状态无关的固定时间收敛模型,并通过在激活函数中添加符号项和比例项来解决鲁棒性问题。文献[27]构造各种激活函数,提出了预定时间收敛模型,显著提高了时变问题求解的计算精度。现有的分析结果是基于初始条件在无限区域内变化,从而保证模型的全局收敛性。然而,RNN 模型的初始条件取值并不会无穷大,一般限制在一个有界区域内。文献[28]分析了基于 Lyapunov 半全局预定时间稳定函数的标量系统,并设计了具有

鲁棒性的预定时间稳定控制器。而针对时变计算的相关研究很少,因此考虑设计半全局预定时间 RNN 模型求解时变矩阵方程是有意义的,其调节时间的界可以通过设计参数来预先调整,从而避免了对收敛时间过高的估计。

针对时变矩阵方程求解,提出了两种具有对数调节时间的 RNN 模型。文中详细分析了采用对数调节时间的 RNN 模型的收敛性能,推导了调节时间函数的具体的表达式,进一步提高了收敛速度和计算精度。在半全局意义上保证了 RNN 模型的固定时间稳定性,给出了所设计的 RNN 模型收敛时间更接近实际值的估计。此外,在初始条件有界的情况下,也保证了模型的预定时间收敛性。将所提出的 RNN 模型应用于时变 Lyapunov 方程求解和冗余机械臂的重复运动规划,从计算机仿真角度证明了所提模型在求解对应问题上的有效性。

2 问题描述

考虑以下时变矩阵方程的一般形式:

$$\sum_{k=1}^N \mathbf{A}_k(t) \mathbf{X}(t) \mathbf{B}_k(t) = \mathbf{C}(t) \quad (1)$$

其中, t 表示时间满足 $[0, \infty)$, $\mathbf{X}(t) \in \mathbf{R}^{n \times m}$ 表示待求解的未知矩阵, $\mathbf{A}(t) \in \mathbf{R}^{n \times n}$, $\mathbf{B}(t) \in \mathbf{R}^{m \times m}$, $\mathbf{C}(t) \in \mathbf{R}^{n \times m}$ 表示时变系数矩阵,并且已知时间导数 $\dot{\mathbf{A}}(t)$, $\dot{\mathbf{B}}(t)$ 。RNN 模型求解时变方程的设计方法是通过定义矢量形式的误差函数,使得该函数的所有元素都能收敛于零,从而得到解。为了实时求解方程(1),时变误差函数设计为:

$$\mathbf{E}(t) = \sum_{k=1}^N \mathbf{A}_k(t) \mathbf{X}(t) \mathbf{B}_k(t) - \mathbf{C}(t) \quad (2)$$

为保证时变误差函数(2)的每一项收敛于零, $\mathbf{E}(t)$ 的时间导数设计为:

$$\dot{\mathbf{E}}(t) = -\gamma \Phi(\mathbf{E}(t)) \quad (3)$$

其中, $\mathbf{E} = \{e_{ij}\}_{n \times m}$, $\Phi(\mathbf{E}) = \{\phi(e_{ij})\}_{n \times m}$; $\phi(\cdot)$ 表示激活函数为奇函数,并且是单调递增的。与神经网络模型的收敛性能密切相关的参数 $\gamma > 0$ 。将误差函数(2)的时间导数代入(3),得到 RNN 模型:

$$\sum_{k=1}^N \mathbf{A}_k(t) \dot{\mathbf{X}}(t) \mathbf{B}_k(t) = -\sum_{k=1}^N \dot{\mathbf{A}}_k(t) \mathbf{X}(t) \mathbf{B}_k(t) + \dot{\mathbf{C}}(t) - \sum_{k=1}^N \mathbf{A}_k(t) \mathbf{X}(t) \dot{\mathbf{B}}_k(t) - \gamma \Phi(\mathbf{E}(t)) \quad (4)$$

在求解时变矩阵问题时,设计参数 γ 和激活函数的合理选取对于模型获得更好的收敛性能起着至关重要的作用。设计参数对应一个电容参数的倒数,应设置为实际工程硬件允许的最大值,或根据仿真需要适当选择。适当增大设计参数 γ 时, $\mathbf{X}(t)$ 收敛到理论解的速度将加快。激活函数负责将神经元的输入映射到输出端,同样会影响模型的求解速度。为进一步提升模型收敛性能,本文将提出新的激活函数作用于模型来加快模型收敛的速度,进而提高计算精度。

3 具有对数调节时间的固定时间收敛 RNN 模型

在求解时变矩阵方程问题时,通常希望神经网络模型能具有优越的收敛性能,能更快地收敛到理论解。为了提高收敛速度与计算精度,本章提出了一种新的具有对数调节时间的模型来更有效地求解时变问题。

3.1 固定时间收敛 RNN 模型 I

为了解决时变矩阵方程的计算问题,本节提出具有对数调节时间的固定时间收敛 RNN 模型,并分析计算了这种模型的收敛时间。

提出以下固定时间收敛激活函数 I:

$$\phi_1(e_{ij}) = \frac{1}{1-\alpha} \left(1 + \mu \left(\frac{|e_{ij}|}{|e_{ij}|+a} \right)^{1-\alpha} \right) \left(\frac{|e_{ij}|}{|e_{ij}|+a} \right)^\alpha \frac{(|e_{ij}|+a)^2}{a} \operatorname{sgn}(e_{ij}) \quad (5)$$

其中, $\mu > 0, a > 0, 0 \leq \alpha < 1$ 。

我们将分析和讨论 RNN 模型(4)采用激活函数(5)时的收敛时间及稳定性。

定理 1 具有激活函数(5)的 RNN 模型(4)的状态 $\mathbf{X}(t)$ 收敛于它的理论解 $\mathbf{X}^*(t)$, 且收敛时间满足:

$$T_{s1} \leq \max\{T_s(e^+(0)), T_s(e^-(0))\} \quad (6)$$

$$T_s(e_{ij}(0)) = \frac{1}{\gamma\mu} \ln \left(1 + \mu \left(\frac{|e_{ij}(0)|}{|e_{ij}(0)|+a} \right)^{1-\alpha} \right)$$

其中, $e^+(0)$ 是初始误差 $e_{ij}(0)$ 的最大值, $e^-(0)$ 是初始误差 $e_{ij}(0)$ 的最小值。并且 RNN 模型(4)在激活函数(5)的作用下全局固定时间收敛于它的理论解 $\mathbf{X}^*(t)$ 。对于任意 $e_{ij}(0)$, 收敛时间的上界为:

$$T_s(e_{ij}(0)) \leq t_1 = \frac{1}{\gamma\mu} \ln(1+\mu) \quad (7)$$

证明:构造 Lyapunov 函数为 $V_{ij}(t) = e_{ij}^2(t)$, 其关于时间的导数 $\dot{V}_{ij}(t)$ 可以计算为:

$$\dot{V}_{ij} = 2e_{ij} \dot{e}_{ij} = -2\gamma e_{ij} \phi_1(e_{ij}) \quad (8)$$

考虑到激活函数 $\phi_1(e_{ij})$ 是单调递增的奇函数, 因此激活函数 $\phi_1(e_{ij})$ 的正负性始终与 $e_{ij}(t)$ 相同, 故 $\dot{V}_{ij}(0) < 0$ 在 $e_{ij}(t) \neq 0$ 时成立。当且仅当 $e_{ij}(t) = 0$ 时 $\dot{V}_{ij}(0) = 0$ 。当采用激活函数(5)时式(8)可表示为:

$$dt = - \frac{(1-\alpha) \left(\frac{|e_{ij}|}{|e_{ij}|+a} \right)^{-\alpha} dV_i}{2\gamma |e_{ij}| \left(1 + \mu \left(\frac{|e_{ij}|}{|e_{ij}|+a} \right)^{1-\alpha} \right) \frac{(|e_{ij}|+a)^2}{a}}$$

对上式两端分别进行积分(右边从 0 到 t_s , 左边从 $V_{ij}(0)$ 到 0), 得到如下的收敛时间函数:

$$T_s(e_{ij}(0)) = \int_0^{V_{ij}(0)} \frac{(1-\alpha) \left(\frac{\sqrt{V_{ij}}}{\sqrt{V_{ij}}+a} \right)^{-\alpha} \frac{a}{(\sqrt{V_{ij}}+a)^2}}{2\gamma \sqrt{V_{ij}} \left(1 + \mu \left(\frac{\sqrt{V_{ij}}}{\sqrt{V_{ij}}+a} \right)^{1-\alpha} \right)} dV_i$$

$$= \int_0^{V_{ij}(0)} \frac{(1-\alpha) \left(\frac{\sqrt{V_{ij}}}{\sqrt{V_{ij}}+a} \right)^{-\alpha} \frac{a}{(\sqrt{V_{ij}}+a)^2}}{\gamma \left(1 + \mu \left(\frac{\sqrt{V_{ij}}}{\sqrt{V_{ij}}+a} \right)^{1-\alpha} \right)} d \left(\frac{\sqrt{V_{ij}}}{\sqrt{V_{ij}}+a} \right)$$

$$= \int_0^{V_{ij}(0)} \frac{1}{\gamma \left(1 + \mu \left(\frac{\sqrt{V_{ij}}}{\sqrt{V_{ij}}+a} \right)^{1-\alpha} \right)} d \left(\frac{\sqrt{V_{ij}}}{\sqrt{V_{ij}}+a} \right)^{1-\alpha}$$

$$= \frac{1}{\gamma\mu} \ln \left(1 + \mu \left(\frac{\sqrt{V_{ij}(0)}}{\sqrt{V_{ij}(0)}+a} \right)^{1-\alpha} \right)$$

因此误差 $e_{ij}(t)$ 从 $e_{ij}(0)$ 收敛到原点所需的时间为 $T_s(e_{ij}(0))$ 。由于 $0 < \frac{e_{ij}(0)}{e_{ij}(0)+a} \leq 1$, 对于 $e_{ij}(0) > 0$, 收敛时间 $T_s(e_{ij}(0))$ 满足以下关系:

$$T_s(e_{ij}(0)) \leq t_1 = \frac{1}{\gamma\mu} \ln(1+\mu) \quad (9)$$

因此, 激活函数(5)作用下的 RNN 模型(4)具有固定时间收敛特性。证毕。

在激活函数(5)作用下 RNN 模型(4)的调节时间函数与函数 $\frac{e_{ij}(0)}{e_{ij}(0)+a}$ 有关。又由于 $0 < \frac{e_{ij}(0)}{e_{ij}(0)+a} \leq 1$, 因此 RNN 模型(4)的收敛上界 t_1 可以很容易得到, 并且它与 RNN 模型(4)的初始状态无关。同时, 我们注意到 t_1 的大小只与设计参数 γ, μ 和函数 $\frac{e_{ij}(0)}{e_{ij}(0)+a}$ 的上界有关。

根据激活函数式(5)中函数的构造形式, 去除其中的一项, 我们注意到它的另一种特殊形式:

$$\phi_2(e_{ij}) = \frac{1}{a(1-\alpha)} (|e_{ij}|)^\alpha (|e_{ij}|+a)^{2-\alpha} \operatorname{sgn}(e_{ij}) \quad (10)$$

根据对定理 1 的类似推导, RNN 模型在激活函数(10)下的固定时间收敛性建立在以下推论中。

推论 1 给定一个初值 $\mathbf{X}(0) \in \mathbf{R}^{n \times m}$, 如果采用激活函数(10), 所提出的 RNN 模型(4)的状态 $\mathbf{X}(t)$ 在时间上界 t_2 内收敛于理论解 $\mathbf{X}^*(t)$:

$$T_s(e_{ij}(0)) \leq t_2 = \frac{1}{\gamma} \quad (11)$$

证明:当使用激活函数(10)时, 根据式(8)可以得到:

$$\dot{V}_{ij}(t) = - \frac{2\gamma}{1-\alpha} \left(\frac{\sqrt{V_{ij}}}{\sqrt{V_{ij}}+a} \right)^\alpha \frac{(\sqrt{V_{ij}}+a)^2}{a} \sqrt{V_{ij}} \quad (12)$$

因此, 收敛时间函数可以计算为:

$$T_s(V_{ij}(0)) = \int_0^{V_{ij}(0)} \frac{(1-\alpha) \frac{a}{(\sqrt{V_{ij}}+a)^2} dV_{ij}}{2\gamma \left(\frac{\sqrt{V_{ij}}}{\sqrt{V_{ij}}+a} \right)^\alpha \sqrt{V_{ij}}}$$

$$= \frac{1}{\gamma} \left(\frac{\sqrt{V_{ij}(0)}}{\sqrt{V_{ij}(0)}+a} \right)^{1-\alpha}$$

其中, 由于 $0 < \frac{e_{ij}(0)}{e_{ij}(0)+a} \leq 1$, 可给出收敛时间函数的最小上界。因此, 式(11)得证。

与式(10)相比, 所提出的激活函数式(5)的收敛时间函数为对数形式, 其调节时间函数更小。这意味着当激活函数式(5)应用于 RNN 模型时具有更好的收敛性能。

3.2 固定时间收敛 RNN 模型 II

为有效求解时变矩阵方程, 进一步说明激活函数对神经网络收敛性的重要性, 我们利用指数函数, 构造了以下固定时间收敛激活函数 II:

$$\phi_3(e_{ij}) = \frac{1}{\beta} e^{(|e_{ij}|^\beta)} |e_{ij}|^{1-\beta} (1 + \mu(1 - e^{(-|e_{ij}|^\beta)})) \operatorname{sgn}(e_{ij}) \quad (13)$$

其中, $\mu > 0, 0 < \beta < 1$ 。

定理 2 具有激活函数(13)的 RNN 模型(4)的状态 $\mathbf{X}(t)$ 收敛于它的理论解 $\mathbf{X}^*(t)$, 且收敛时间满足:

$$T_{s2} \leq \max\{T_s(e^+(0)), T_s(e^-(0))\}$$

$$T_s(e_{ij}(0)) = \frac{1}{\gamma\mu} \ln(1 + \mu(1 - \exp(-|e_{ij}(0)|^\beta))) \quad (14)$$

RNN模型(4)在激活函数(13)作用下全局固定时间收敛,对于任意 $e_i(0)$,收敛时间的上界为:

$$T_s(e_i(0)) \leq t_3 = \frac{1}{\gamma\mu} \ln(1 + \mu) \quad (15)$$

证明:与定理1一样,定义一个李雅普诺夫候选函数 $V_{ij}(t) = e_{ij}^2(t)$ 。当采用激活函数(13)时,可以得到

$$dt = \frac{\beta dV_{ij}}{2\gamma \sqrt{V_{ij}} (1 + \mu(1 - \exp(-(\sqrt{V_{ij}})^\beta)) \exp((\sqrt{V_{ij}})^\beta) (\sqrt{V_{ij}})^{1-\beta})}$$

对上式两边进行积分并整理可得:

$$T_s(V_{ij}(0)) = \int_0^{V_{ij}(0)} \frac{\exp((\sqrt{V_{ij}})^\beta) (\sqrt{V_{ij}})^{1-\beta} dV_{ij}}{2\gamma\beta \sqrt{V_{ij}} (1 + \mu(1 - \exp(-(\sqrt{V_{ij}})^\beta))}$$

$$= \frac{1}{\gamma\mu} \ln(1 + \mu(1 - \exp(-\beta))) \quad (16)$$

从初始误差 $e_{ij}(0)$ 收敛到原点所需的时间如式(14)所示。又因 $\exp(-|e_{ij}(0)|^\beta) \in (0, 1]$,因此,可以得到调节时间的上界为一个常数。RNN-II模型的收敛时间上界满足:

$$T_s(e_{ij}(0)) \leq t_3 = \frac{1}{\gamma\mu} \ln(1 + \mu)$$

因此,在激活函数II下,RNN模型能够在固定时间收敛到理论解。证毕。

在之前的工作^[15,25-26]中,利用指数函数构造了以下被广泛研究的激活函数形式:

$$\phi_1(e_{ij}) = \frac{1}{\beta} \exp(|e_{ij}|^\beta) |e_{ij}|^{1-\beta} \operatorname{sgn}(e_{ij}) \quad (17)$$

其中,参数 $0 < \beta < 1$ 。当采用激活函数式(17)时,RNN模型可以实现固定时间收敛,其收敛时间的上界为^[15,25-26]:

$$T_s(e_{ij}(0)) \leq t_4 = \frac{1}{\gamma}$$

相比激活函数式(17),采用激活函数式(13)的RNN模型实现了对数调节时间,因此收敛速度更快。

在给定的初始条件下,可以得到两种RNN模型调节时间函数的精确表达式。相比已有的固定时间收敛的RNN模型,所提出的两个RNN模型具有对数调节时间,并实现了更快的收敛速度,其调节时间上界更小。激活函数I中包含多项式项而不是指数项。当在控制器设计中采用时,其优势显著,因为远离原点的多项式项意味着比指数项的控制能量更少。

对于激活函数(5)和(13),RNN模型(4)的调节时间函数相对于初始误差是有界的,如式(7)和式(15)所示,这意味着对于 $e_{ij}(0)$ 所有可能的取值,RNN模型(4)是全局固定时间稳定的。获得精确的初始误差值可以极大地提高计算精度,但收敛时间函数依赖于初始误差 $e_{ij}(0)$,如式(6)和式(14)所示。在实际应用中,由于测量噪声通常无法获得精确的初始误差值,因此综合考虑在已知的有界范围 $e_{ij}(0)$ 内可能的所有取值,这样可以获得RNN模型更接近实际收敛时间的上界。

4 半全局稳定RNN模型

为了满足时变计算的要求,考虑到计算解要求更完整、更准确,进一步缩短调节时间的估计就显得尤为重要。RNN

模型的调节时间实际上取决于它的初始条件。由于测量噪声的存在,很难给出 $\mathbf{E}(0)$ 的准确值。实际上,初始误差并不是无穷大的,通常在一个有界区域内变化。因此,我们主要考虑半全局固定/预定时间稳定的RNN模型,在 $\mathbf{E}(0)$ 有界的情况下,给出RNN模型的调节时间更准确的估计。本章给出了RNN模型的全局固定时间稳定性结果,以提高模型的收敛速度和计算精度。

4.1 半全局固定时间收敛RNN模型

设 $S = [-\Delta, \Delta]$,对于给定的 $\Delta > 0$ 。假设第3章中给出的RNN模型的初值满足 $|e_{ij}(0)| \leq \Delta$ 。下面的定理总结了半全局固定时间收敛性。

定理3 对于在 $|e_{ij}(0)| \leq \Delta$ 条件下给定的初始状态 $\mathbf{X}(0)$,RNN模型(4)的状态 $\mathbf{X}(t)$ 收敛于其理论值 $\mathbf{X}^*(t)$,并且RNN模型具有半全局固定时间稳定性,调节时间函数有最小上界。

当采用激活函数(5)时,

$$t_{g1}(\Delta) = \frac{1}{\gamma\mu} \ln\left(1 + \mu \left(\frac{\Delta}{\Delta+a}\right)^{1-\alpha}\right) \quad (18)$$

当采用激活函数(13)时,

$$t_{g2}(\Delta) = \frac{1}{\gamma\mu} \ln(1 + \mu(1 - \exp(-\Delta^\beta))) \quad (19)$$

证明:根据定理1和定理2,对 $T_s(e_{ij}(0))$ 进行上界估计,即当 $|e_{ij}(0)| \leq \Delta$ 时,依次由式(6)和式(14)得到半全局调节时间上界式(18)和半全局调节时间上界式(19)。证毕。

4.2 半全局预定收敛RNN模型

当我们采用以下动态误差函数时,可以得到预定时间稳定的RNN模型:

$$\dot{\mathbf{E}}(t) = -\gamma \Phi_f(\mathbf{E}(t)) \quad (20)$$

其中, $\Phi_f(\mathbf{E}) = (t_\Delta/t_f)(\mathbf{E})$ 。这里 $\Phi(\mathbf{E}(t))$ 是式(3)中使用的激活函数,我们采用激活函数式(5)和式(13)。可以看出,式(20)与式(3)不同,因为引入了增益 t_f/t_Δ 。 t_f 与所使用的激活函数中的参数无关,它是单独引入的参数,通过调节这个参数可以任意确定系统的收敛时间。 t_Δ 为定理3中提供的半全局调节时间函数的上界。求解一般时变线性方程(1)的预定时间收敛RNN模型可表示为:

$$\sum_{k=1}^N \mathbf{A}_k(t) \dot{\mathbf{X}}(t) \mathbf{B}_k(t) = - \sum_{k=1}^N \dot{\mathbf{A}}_k(t) \mathbf{X}(t) \mathbf{B}_k(t) + \dot{\mathbf{C}}(t) - \sum_{k=1}^N \mathbf{A}_k(t) \mathbf{X}(t) \dot{\mathbf{B}}_k(t) - \gamma \Phi_f(\mathbf{E}(t)) \quad (21)$$

定理4 当给定初始状态 $\mathbf{X}(0)$ 满足误差函数 $|e_{ij}(0)| \leq \Delta$ 时,RNN模型(21)在提出的激活函数下的状态变量 $\mathbf{X}(t)$ 是预定时间稳定的,并且在预定时间 t_f 内收敛到理论解 $\mathbf{X}^*(t)$ 。

证明:当 t_Δ 设为相应采用的激活函数在半全局意义上的最小上界时,RNN模型(21)的收敛时间函数得到满足。

$$\max_{|e_{ij}(0)| \leq \Delta} T_s(e_{ij}(0)) = t_f \quad (22)$$

因此,RNN模型(21)是预定时间稳定的。

当采用激活函数(5)和(13)时,RNN模型(21)中对应的参数 t_Δ 分别对应 $t_{g1}(\Delta)$ 和 $t_{g2}(\Delta)$ 。

RNN模型(21)用于求解时变矩阵方程时,可以在预定时间内得到理论解。对于预定时间稳定的RNN模型,可以

通过适当调整参数 t_f 来任意调节模型的收敛时间。结果表明,在不同的激活函数下,RNN 模型的预定收敛时间 t_f 可以是相同的,两者的区别在于表征的收敛时间的上界 t_Δ 。需要注意的是,预定时间 t_f 和固定上界 t_Δ 取决于不同的参数。

5 数值算例

在本章中,考虑时变矩阵方程式(1)中不同系数矩阵的形式,可以给出具体形式的时变矩阵方程。将所提出的 RNN 模型应用于具体时变矩阵问题的求解,进一步比较分析具有对数调节时间的神经网络的计算性能。

5.1 时变矩阵方程求解

考虑将一般时变线性方程式(1)中的参数设置为 $N=2, m=n, \mathbf{B}_1(t)=\mathbf{A}_2(t)=\mathbf{I}$, 并且 $\mathbf{B}_2(t)\mathbf{A}_1^\top(t)$, 则方程变为以下时变 Lyapunov 方程:

$$\mathbf{A}_1(t)\mathbf{X}(t)+\mathbf{X}(t)\mathbf{A}_1^\top(t)=\mathbf{C}(t) \tag{23}$$

根据 RNN 模型(4),可以得到以下具体的固定时间收敛神经网络模型:

$$\mathbf{A}_1(t)\dot{\mathbf{X}}(t)+\dot{\mathbf{X}}(t)\mathbf{A}_1^\top(t)=-\dot{\mathbf{A}}_1(t)\mathbf{X}(t)-\mathbf{X}(t)\dot{\mathbf{A}}_1^\top(t)+\dot{\mathbf{C}}(t)-\gamma\Phi_{1,2,3,4}(\mathbf{E}(t)) \tag{24}$$

若考虑将一般时变线性方程(1)中的参数设置为 $N=2, m=n, \mathbf{B}_1(t)=\mathbf{A}_2(t)=\mathbf{I}$, 则方程变为以下时变 Sylvester 方程:

$$\mathbf{A}_1(t)\mathbf{X}(t)+\mathbf{X}(t)\mathbf{B}_2(t)=\mathbf{C}(t) \tag{25}$$

根据 RNN 模型式(4),可以得到以下具体的固定时间收敛神经网络模型:

$$\mathbf{A}_1(t)\dot{\mathbf{X}}(t)+\dot{\mathbf{X}}(t)\mathbf{B}_2(t)=-\dot{\mathbf{A}}_1(t)\mathbf{X}(t)-\mathbf{X}(t)\dot{\mathbf{B}}_2(t)+\dot{\mathbf{C}}(t)-\gamma\Phi_{1,2,3,4}(\mathbf{E}(t)) \tag{26}$$

例1 考虑时变 Lyapunov 方程(23)中以下时变参数矩阵:

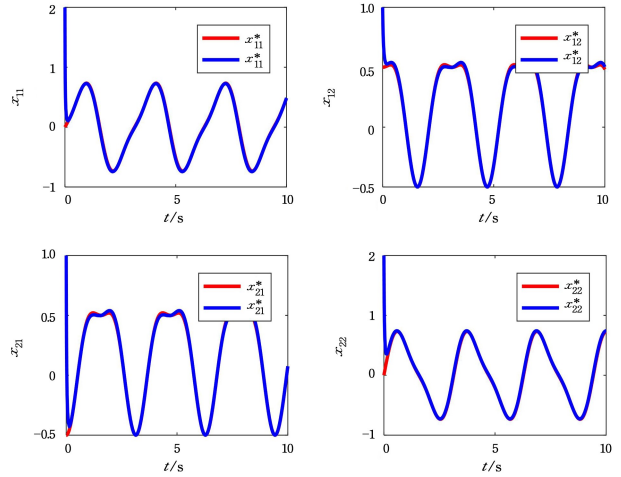
$$\mathbf{A}_1 = \begin{bmatrix} -1-0.5\cos(2t) & 0.5\sin(2t) \\ 0\sin(2t) & -1+0.5\cos(2t) \end{bmatrix}$$

$$\mathbf{C}(t) = \begin{bmatrix} \sin(2t) & \cos(2t) \\ -\cos(2t) & \sin(2t) \end{bmatrix}$$

分别考虑在激活函数(5)和(13)的作用下,RNN 模型(24)对时变 Lyapunov 方程进行求解。RNN 模型(24)的初始值设置为 $\mathbf{X}(0)=[X_{11}(0), X_{12}(0); X_{21}(0), X_{22}(0)]=[2, 1; 1, 2]$ 。神经网络模型的激活函数参数设置为 $\gamma=2, \mu=2, \alpha=3/5, \beta=1/3, a=1$ 。根据定理 3,可计算出 $t_{g1}(\Delta)=0.268, t_{g2}(\Delta)=0.255$,其中 $\Delta=10$ 。对于收敛时间的上界,分别计算出 $t_1=t_3=0.274, t_2=t_4=0.5$ 。

RNN 模型(24)使用激活函数式(5)和式(13)下得到的状态轨迹分别如图 1 和图 2 所示。可以观察到,RNN 模型式(24)的状态变量 $\mathbf{X}(t)$ 收敛于理论解 $\mathbf{X}^*(t)$ 。为了评估神经网络模型误差变量的收敛性能,引入误差指标 $J_E = \|\mathbf{E}\|_F$ 。图 3 和图 4 分别描述了固定时间收敛 RNN 模型 I 和 II 的瞬态行为,其中图 3 展示了应用激活函数式(5)和式(10)时的性能比较,同时给出了 RNN 模型(24)分别在激活函数式(5)和式(10)下收敛时间上界的估计。图 4 展示了使用激活函数式(13)和式(17)下模型误差指标的比较。可以看出,具有对数调节时间的神经网络具有更快的收敛速度,在

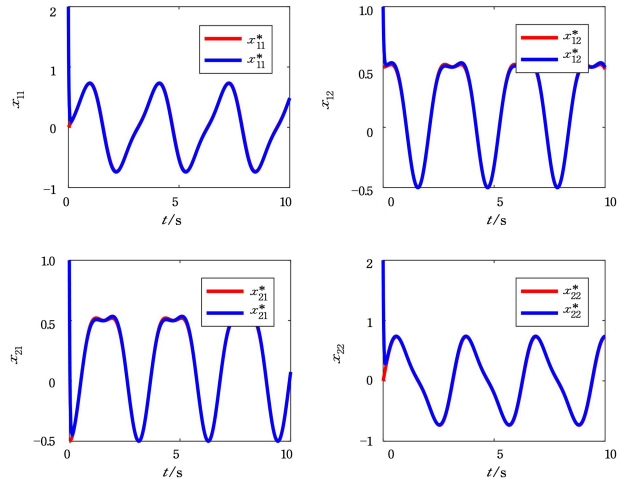
半全局意义下估计收敛时间的界更接近实际收敛时间。图 5 和图 6 展示了当分别应用激活函数式(5)和式(13)时神经网络模型式(21)的收敛结果。模型参数 t_{1f} 分别设置为 0.268, 0.15 和 0.1; t_{3f} 分别设置为 0.255, 0.15 和 0.1。可以观察到,RNN 模型的状态在时刻 t_f 收敛到理论解,收敛性通过可预定时间 t_f 来实现。这证实了 RNN 模型的预定时间收敛性。



注:红线为理论解,蓝线为计算状态解。

图 1 RNN 模型(24)使用激活函数(5)的求解结果 (电子版为彩图)

Fig. 1 Neural solutions of RNN model(24) with AF (5)



注:红线为理论解,蓝线为计算状态解。

图 2 RNN 模型(24)使用激活函数(13)的求解结果 (电子版为彩图)

Fig. 2 Neural solutions of RNN model(24) with AF (13)

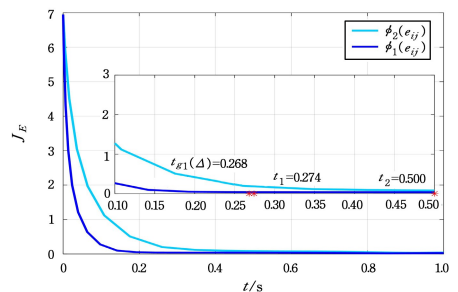


图 3 应用具有激活函数(5)和(10)的 RNN 模型(24)时的误差指标 Fig. 3 Error indices when applying RNN (24) with AFs (5) and (10) respectively

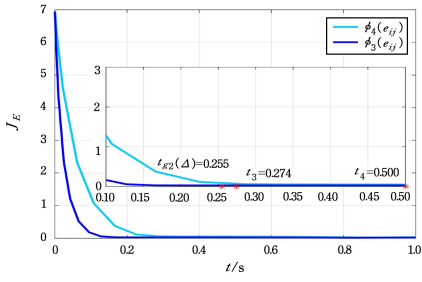


图4 应用具有激活函数(13)和(17)的 RNN 模型(24)时的误差指标
Fig. 4 Error indices when applying RNN (24) with AFs (13) and (17) respectively

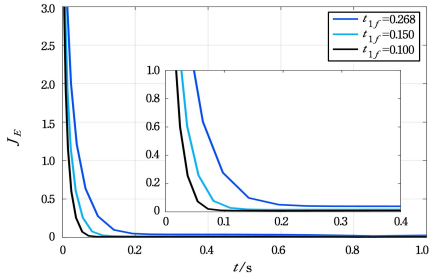


图5 在不同 t_f 下应用具有激活函数(5)的神经网络模型(21)求解例 1 时的误差指标

Fig. 5 Error indices of solving example 1 when applying neural network model (21) with AF (5) under different t_f

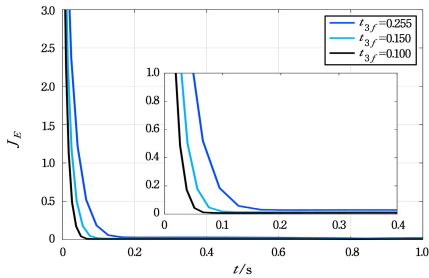


图6 在不同 t_f 下应用具有激活函数(13)的神经网络模型(21)求解例 1 时的误差指标

Fig. 6 Error indices of solving example 1 when applying neural network model (21) with AF (13) under different t_f

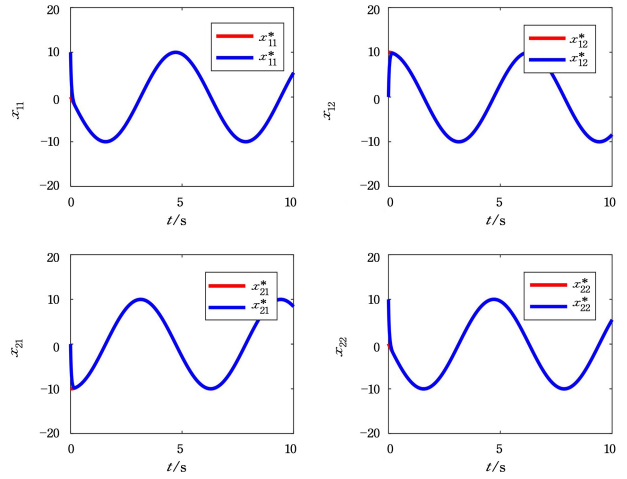
例 2 考虑时变 Sylvester 方程(25),其时变参数矩阵如下:

$$A_1 = \begin{bmatrix} -\frac{\sin t}{10} & -\frac{\cos t}{10} \\ \frac{\cos t}{10} & -\frac{\sin t}{10} \end{bmatrix}, B_2(t) = \begin{bmatrix} \frac{\sin t}{100} & 0 \\ 0 & \frac{\cos t}{50} \end{bmatrix}$$

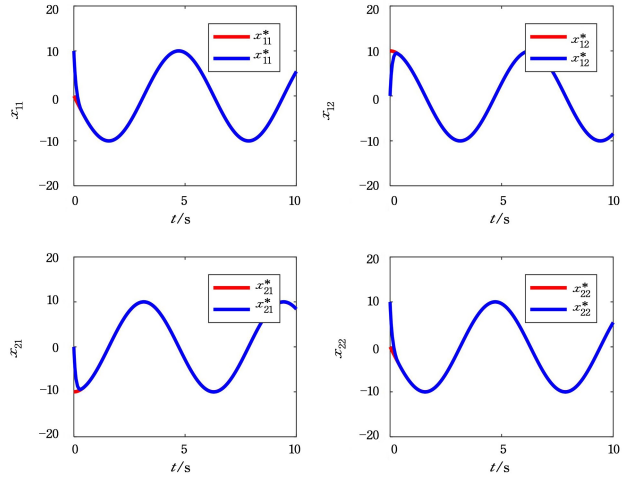
$$C(t) = \begin{bmatrix} \frac{\sin^2 t}{10} - 1 & -\frac{\cos^2 t}{5} \\ \frac{\sin t \cos t}{10} & \frac{\sin t \cos t}{5} - 1 \end{bmatrix}$$

利用提出的 RNN 模型来求解这个时变 Sylvester 方程问题。设置与例 1 相同的神经网络模型参数,选取初始值 $X(0) = [X_{11}(0), X_{12}(0); X_{21}(0), X_{22}(0)] = [10, 0; 0, 10]$ 。

用提出的激活函数(5)和(13)激活 RNN 模型(26)后,得到的求解时变 Sylvester 方程的结果如图 7 所示。可以看出,在激活函数(5)和(13)下 RNN 模型(26)生成的蓝色实线曲线(状态解)与红色虚线曲线(理论解)很快重合。



(a)采用激活函数(5)



(b)采用激活函数(13)

注:红线为理论解,蓝线为计算状态解。

图7 RNN 模型(26)使用不同激活函数的求解结果
(电子版为彩图)

Fig. 7 Neural solutions of RNN model(26) with different AFs

此外,为了达到比较的目的,图 8 给出了由激活函数(5)和(10)激活的 RNN 模型(26)求解时变 Sylvester 方程所对应的误差指标。图 9 同样展示了在激活函数(13)和(17)下模型的误差指标对比, J_E 在调节时间 t_Δ 内收敛至零。结果也验证了具有对数调节时间的 RNN 模型收敛速度较快,并且具有更小的调节时间上界。

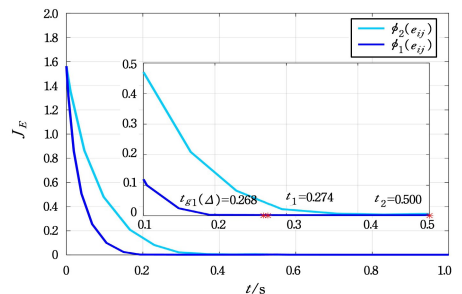


图8 应用具有激活函数(5)和(10)的 RNN 模型(26)时的误差指标

Fig. 8 Error indices when applying RNN (26) with AFs (5) and (10) respectively

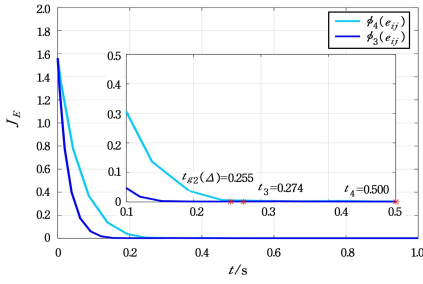


图9 应用具有激活函数(13)和(17)的RNN模型(26)时的误差指标

Fig. 9 Error indices when applying RNN (26) with AFs (13) and (17) respectively

图10和图11同样验证了预定时间收敛的RNN模型在求解时变Sylvester方程的有效性。从图中可以看出,调整 t_f 可以缩短收敛时间。同时,在不同的 t_f 和不同激活函数激活的神经网络模型下,最终的时间 t_f 可以是相同的。然而,它们的不同之处在于收敛特性 t_Δ 。

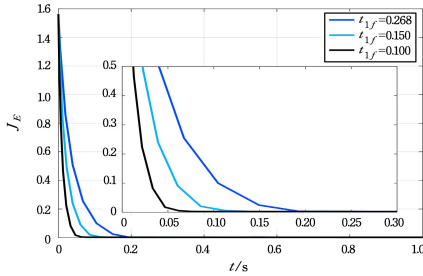


图10 在不同 t_f 下应用具有激活函数(5)的神经网络模型(21)求解例2时的误差指标

Fig. 10 Error indices of solving example 2 when applying neural network model (21) with AF (5) under different t_f

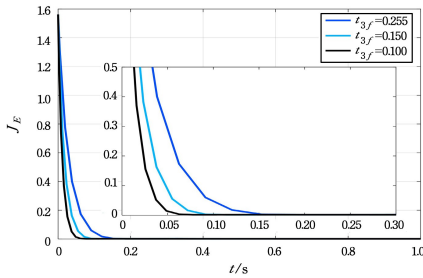


图11 在不同 t_f 下应用具有激活函数(13)的神经网络模型(21)求解例2时的误差指标

Fig. 11 Error indices of solving example 2 when applying neural network model (21) with AF (13) under different t_f

5.2 冗余机械臂重复运动规划

末端执行器位置和速度的运动学方程^[9]表示为 $\mathbf{r}(t) = \mathbf{f}(\boldsymbol{\theta}(t))$ 和 $\dot{\mathbf{r}}(t) = \mathbf{J}(\boldsymbol{\theta}(t))\dot{\boldsymbol{\theta}}(t)$,其中 $\mathbf{r}(t)$ 和 $\dot{\mathbf{r}}(t)$ 分别表示给定路径上期望的末端执行器位置向量和速度向量, $\boldsymbol{\theta}(t)$ 和 $\dot{\boldsymbol{\theta}}(t)$ 分别是关节角向量和关节速度向量, $\mathbf{J}(\boldsymbol{\theta}(t)) = \partial \mathbf{f}(\boldsymbol{\theta}(t)) / \partial \boldsymbol{\theta}(t)$ 是关节角的雅可比矩阵, $\mathbf{f}(\boldsymbol{\theta}(t))$ 是关于关节角 $\boldsymbol{\theta}(t)$ 的光滑非线性映射。

考虑到漂移问题,引入了以下重复运动规划方案^[12]:

$$\min \frac{1}{2} \dot{\boldsymbol{\theta}}(t)^T \boldsymbol{\theta}(t) + \mathbf{P}_o(t)^T \dot{\boldsymbol{\theta}}(t) \quad (27)$$

$$\text{s. t. } \mathbf{J}(\boldsymbol{\theta}(t)) \dot{\boldsymbol{\theta}}(t) = \dot{\mathbf{r}}(t) + \mathbf{P}_c(t)$$

其中, $\mathbf{P}_o(t) = \boldsymbol{\rho}(\boldsymbol{\theta}(t) - \boldsymbol{\theta}_d(0))$, $\mathbf{P}_c(t) = \boldsymbol{\kappa}_p(\mathbf{r}(t) - \mathbf{f}(\boldsymbol{\theta}(t)))$, $\boldsymbol{\rho} > 0$ 和 $\boldsymbol{\kappa}_p > 0$ 是调节参数, $\boldsymbol{\theta}_d(0)$ 表示期望的初始关节位置向量, \mathbf{r}_d 和 $\dot{\mathbf{r}}_d(t)$ 分别表示末端执行器的期望轨迹和速度向量。

为了解决QP问题,我们引入拉格朗日乘子 $\lambda(t)$,可以得到矩阵方程:

$$\mathbf{W}(t)\mathbf{z}(t) = \mathbf{u}(t)$$

其中

$$\mathbf{W}(t) = \begin{bmatrix} \mathbf{I} & \mathbf{J}(t)^T \\ \mathbf{J}(t) & 0 \end{bmatrix}, \mathbf{z}(t) = \begin{bmatrix} \dot{\boldsymbol{\theta}}(t) \\ \lambda(t) \end{bmatrix}$$

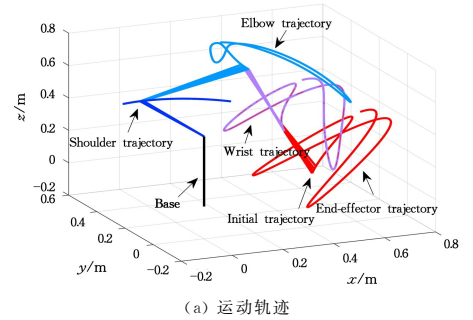
$$\mathbf{u}(t) = \begin{bmatrix} -\mathbf{P}_o(t) \\ \dot{\mathbf{r}}(t) + \mathbf{P}_c(t) \end{bmatrix}$$

根据模型(21),则求解QP问题的RNN模型可表示为:

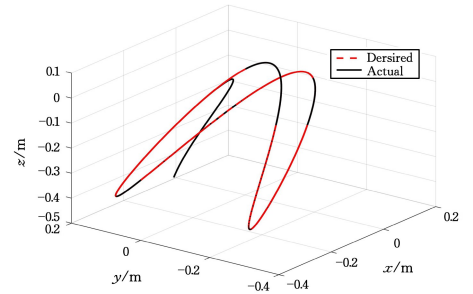
$$\mathbf{W}(t)\dot{\mathbf{z}}(t) = -\dot{\mathbf{W}}(t)\mathbf{z}(t) + \dot{\mathbf{u}}(t) - \gamma \boldsymbol{\Phi}_f(\mathbf{W}(t)\mathbf{z}(t) - \mathbf{u}(t)) \quad (28)$$

在RNN模型(28)中,考虑采用激活函数(5)和(13)来求解重复运动规划问题。本节以冗余机械臂PUMA560为对象进行仿真实验,设置机械臂末端执行器的期望初始角度为 $\boldsymbol{\theta}_d(0) = [0, -\frac{\pi}{4}, 0, \frac{\pi}{2}, -\frac{\pi}{4}, 0]^T$,考虑到机械臂末端执行器的初始位置不在期望的运动轨迹上,关节角的实际的初始值为 $\boldsymbol{\theta}(0) = [0, -\frac{\pi}{4} + \frac{1}{2}, 0, \frac{\pi}{2}, -\frac{\pi}{4} + \frac{1}{2}, 0]^T$ 。神经网络模型参数设置为 $k_p = 1, \rho = 2, \gamma = 2, \mu = 2, \alpha = 3/5, \alpha = 1, \beta = 1/3$ 。

图12(a)展示了末端执行器在三维工作空间中执行期望路径任务时的轨迹。图12(b)展示了末端执行器的期望轨迹和实际轨迹。



(a) 运动轨迹



(b) 末端执行器的期望路径和实际轨迹

图12 机械臂末端执行器跟踪圆形路径时,RNN模型(28)在激活函数(5)下的运动规划结果

Fig. 12 Motion planning results of RNN model (28) with AF(5) when the end effector of robotic arm tracks the circular path

从图 12(b)可以看出,RNN 模型保证了末端执行器的轨迹收敛到期望的轨迹,符合实际需求。在图 13(a)中,最终的关节角度返回到期望的初始位置。如图 13(a)一 图 13(b)所示,所有关节角度和关节速度都是连续且光滑的,再次验证了 RNN 模型(28)的有效性。图 13(c)分别给出了位置误差分量 e_x, e_y 和 e_z 。可以观察到,末端执行器的位置误差最大值很小,达到 10^{-4} 的数量级,表明计算目标已经实现。本文引入了指标 $J_E = \| \mathbf{W}(t)\mathbf{z}(t) - \mathbf{u}(t) \|_F$ 。图 14(a)和 图 14(b)分别展示了采用激活函数(5)和(10),(13)和(17)的 RNN 模型的误差指标对比结果。结果表明,具有对数调节时间的 RNN 模型的收敛速度比采用非对数调节时间的 RNN 模型更快。

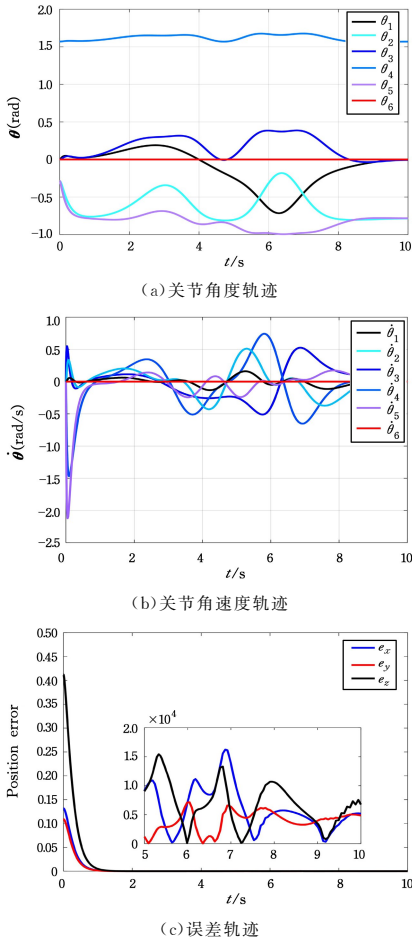


图 13 PUMA560 末端执行器跟踪圆形路径时,RNN 模型(28)在激活函数(5)下运动规划结果

Fig. 13 Motion planning results of RNN model (28) with AF (5), when the end effector of PUMA560 tracks the circular path

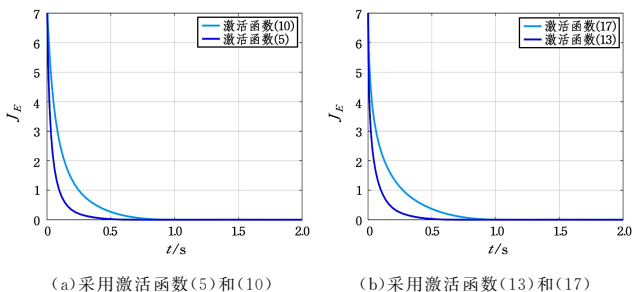


图 14 RNN 模型(28)在不同激活函数下的误差指标

Fig. 14 Error indices of RNN model(28) with different AFs

表 1 列出了在理论分析估计的调节时间上下界,RNN 模型(28)在不同激活函数下的收敛时间和相应的计算误差,结果验证了 RNN 模型在激活函数(5)和(13)下有较快的收敛速度和较小的计算误差。表 2 列出了当末端执行器完成轨迹规划时,使用由不同激活函数激活的 RNN 模型(28)时,机械臂的各关节角实际回拢角度与其期望角度之间的偏差对比。与已有的固定时间神经网络相比,具有对数调节时间的固定时间神经网络的收敛精度得到明显的改善。

表 1 在指定时间 t_f 下不同激活函数下模型(28)的计算误差

Table 1 Computing errors of model (28) with different AFs at specified time t_f

估计时间 t_f /s	激活函数(5)	激活函数(10)	激活函数(13)	激活函数(17)
0.255	2.77×10^{-3}	0.21	3.91×10^{-4}	0.29
0.268	1.98×10^{-3}	0.17	2.58×10^{-3}	0.25
0.274	1.69×10^{-3}	0.15	2.18×10^{-3}	0.24
0.500	2.17×10^{-4}	7.76×10^{-3}	1.90×10^{-4}	2.15×10^{-3}

表 2 使用具有不同激活函数的 RNN 模型时各个关节角偏差

Table 2 Deviations of each joint angles when using RNN model with different activation functions

网络求解器/ rad	激活函数 (5)	激活函数 (10)	激活函数 (13)	激活函数 (17)
$\theta_1(6) - \theta_{d1}(0)$	-7.13×10^{-4}	-6.76×10^{-3}	-6.11×10^{-5}	-6.79×10^{-3}
$\theta_2(6) - \theta_{d2}(0)$	-1.05×10^{-5}	-4.55×10^{-4}	-1.03×10^{-4}	-2.34×10^{-4}
$\theta_3(6) - \theta_{d3}(0)$	-5.74×10^{-4}	-4.71×10^{-3}	-4.13×10^{-4}	-4.85×10^{-3}
$\theta_4(6) - \theta_{d4}(0)$	-1.74×10^{-4}	-1.51×10^{-3}	-1.2×10^{-4}	-1.3×10^{-3}
$\theta_5(6) - \theta_{d5}(0)$	2.67×10^{-4}	1.89×10^{-3}	2.72×10^{-4}	1.9×10^{-3}
$\theta_6(6) - \theta_{d6}(0)$	0	0	0	0

结束语 本文针对时变矩阵方程求解,设计了两个新颖的激活函数,提出了实现对数调节时间的固定时间收敛 RNN 模型,从而进一步提高了时变计算的收敛速度和计算精度。给出了 RNN 模型半全局对数调节时间函数,并由此推导出半全局意义上的调节时间函数的上界。采用半全局调节时间上界来建立修正的 RNN 模型,不仅可以实现预定时间收敛,而且可以通过调节引入的参数来调整收敛时间。最后,给出了所提计算方案的仿真结果,并将其应用于冗余机械臂的重复运动规划,验证了所提模型的应用前景。仿真结果验证了所提模型的有效性。考虑到与实现相关的影响,未来的研究应侧重于在存在干扰的情况下解决时变计算问题,实现干扰抑制和鲁棒性。

参考文献

- [1] MANHERZ R, JORDAN B, HAKIMI S. Analog methods for computation of the generalized inverse[J]. IEEE Transactions on Automatic Control, 1968, 13(5): 582-585.
- [2] ZHANG Y N, YI C. Zhang neural networks and neural-dynamic method[M]. New York: Nova Science Publishers, 2011.
- [3] 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.
- [4] CICHOCKI A, UNBEHAUEN R. Neural networks for solving systems of linear equation and related problems [J]. *IEEE Transactions on Circuits and Systems*, 1992, 39(2):124-138.
- [5] WANG J. Recurrent neural networks for solving linear matrix equations[J]. *Computers and Mathematics with Applications*, 1993, 26(9):23-34.
- [6] KENNEDY M, CHUA L. Neural networks for nonlinear programming[J]. *IEEE Transactions on Circuits and Systems*, 1988, 35(5):554-562.
- [7] MAA C Y, SHANBLATT M A. Linear and quadratic programming neural network analysis[J]. *IEEE Transactions on Neural Networks*, 1992, 3(4):580-594.
- [8] CHENG F, CHEN T, SUN Y. Resolving manipulator redundancy under inequality constraints[J]. *IEEE Transactions on Robotics and Automation*, 1994, 10(1):65-71.
- [9] KLEIN C A, KEE K B. The nature of drift in pseudoinverse control of kinematically redundant manipulators [J]. *IEEE Transactions on Robotics and Automation*, 1989, 5(2):231-234.
- [10] ZHANG Y N, TAN Z G, YANG Z, et al. Seif motion of planar redundant manipulator based on quadratic programming [J]. *Chinese Journal of Robot*, 2008, 3(6):566-571.
- [11] 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.
- [12] ZHANG Y N, WANG J, XIA Y. 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.
- [13] ZHANG Y N, GE S. 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.
- [14] LI S, CHEN S, LIU B. Accelerating a recurrent neural network to finite-time convergence for solving time-varying Sylvester equation by using a sign-bi-power activation function[J]. *Neural Processing Letters*, 2013, 37:189-205.
- [15] LI W B. A recurrent neural network with explicitly definable convergence time for solving time-variant linear matrix equations[J]. *IEEE Transactions on Industrial Informatics*, 2018, 14(12):5289-5298.
- [16] 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(5):98-105.
- [17] XIAO L, ZHANG Y N, HU Z, et al. Performance benefits of robust nonlinear zeroing neural network for finding accurate solution of Lyapunov equation in presence of various noises[J]. *IEEE Transactions on Industrial Informatics*, 2019, 15(9):5161-5171.
- [18] LI W B. Design and analysis of a novel finite-time convergent and noise tolerant recurrent neural network for time-variant matrix inversion[J]. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2020, 50(11):4362-4376.
- [19] XIAO L, LI K I, DUAN M. Computing time-varying quadratic optimization with finite-time convergence and noise tolerance: a unified framework for zeroing neural network[J]. *IEEE Transactions on Neural Networks and Learning Systems*, 2019, 30(11):3360-3369.
- [20] SUN M X, WENG D E, ZHNG Y. Time-variant neurocomputing with finite-value terminal recurrent neural networks [J]. *Computer Science*, 2020, 47(1):212-218.
- [21] SUN M X, ZHANG Y, WU Y X, et al. On a finitely-activated terminal RNN approach to time-variant problem solving[J]. *IEEE Transactions on Neural Networks and Learning Systems*, 2022, 33(12):7289-7302.
- [22] POLYAKOV A. Nonlinear feedback design for fixed-time stabilization of linear control systems[J]. *IEEE Transactions on Automatic Control*, 2012, 57(8):2106-2110.
- [23] SUN M X, LI X, ZHONG G M. Semi-global fixed/predefined-time RNN models with comprehensive comparisons for time-variant neural computing[J]. *Neural Computing and Applications*, 2023, 35:1675-1693.
- [24] SANCHEZ-TORRES J D, SANCHEZ E N, LOUKIANOV A G. Predefined-time stability of dynamical systems with sliding modes[C]// *Proceedings of American Control Conference*. Chicago, IL, 2015:5842-5846.
- [25] XIAO L, ZHANG Y N, HU Z, et al. Performance benefits of robust nonlinear zeroing neural network for finding accurate solution of Lyapunovequation in presence of various noises[J]. *IEEE Transactions on Industrial Informatics*, 2019, 15(9):5161-5171.
- [26] XIAO L, CAO Y, DAI J H, et al. Finite-Time and predefined-time convergence design for zeroing neural network: theorem, method, and verification[J]. *IEEE Transactions on Industrial Informatics*, 2021, 17(7):4724-4732.
- [27] LI W B. Predefined-time convergent neural solution to cyclical motion planning of redundant robots under physical constraints [J]. *IEEE Transactions on Industrial Electronics*. 2020, 67(12):10732-10743.
- [28] JIMENEZ-RODRIGUEZ E, SANCHEZ-TORRES J D, LOUKIANOV A G. Semi-global predefined time stable vector systems [C]// *Proceedings of IEEE Conference on Decision and Control*. Miami Beach, FL, 2018:4809-4814.



LI Xing, born in 1994, Ph.D. Her main research interests include neural networks and robotics.



ZHONG Guomin, born in 1983, Ph.D, lecturer. His main research interests include iterative learning algorithms and neural networks.