

二次曲线型终态神经网络:时变神经计算与冗余机械臂重复运动规划^①

孙明轩^② 吴雨芯 张 钰

(浙江工业大学信息工程学院 杭州 310023)

摘要 提出二次曲线型终态神经网络,包括双曲线型、椭圆型和抛物线型 3 种终态神经网络,网络各变量取值有限,易于实现。详细分析了这类网络的有限时间收敛特性,给出了具体的收敛时间表达式,并以双曲线型终态神经网络为例,将其应用于时变线性矩阵方程求解以及机器人轨迹规划。对于一般时变线性矩阵方程的求解,讨论了时变 Lyapunov 方程和时变 Sylvester 方程,分别给出了终态神经网络用于求解这 2 类时变矩阵方程的计算结果,验证了该网络能在有限时间内精确收敛到理论解。对于冗余机械臂重复运动规划,文中将重复运动指标设计为终态收敛性能指标,在初始位置偏移的情况下,利用该终态神经网络进行求解,实现冗余机械臂有限时间收敛的重复运动规划任务。

关键词 终态神经网络; 时变线性矩阵方程; 冗余机械臂; 重复运动规划

0 引言

矩阵求逆与矩阵方程求解是广泛应用于科学研究所及工程等领域中的典型计算问题。Hopfield 于 20 世纪 80 年代提出的递归神经网络是常用的求解方法^[1,2]。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。这种神经网络也被广泛用于解决优化问题。例如,文献[3]将非线性规划归结为递归神经网络的稳态解,文献[4]设计动态梯度系统求解优化问题,文献[5]采用对偶递归神网络模型求解线性与二次规划问题。为解决传统递归神经网络在求解时变问题时的不足,文献[6]提出一种渐近收敛递归神经网络模型 (recurrent neural networks, RNN),将其用于求解时变线性矩阵方程,能够保证其计算解指数地收敛于理论解。文献[7]讨论了这种 RNN 在离散实现时的计算性能,并与牛顿迭代法进行了比较,其与梯度法的比较结果见文献[8]。文献[9]将这种方法应用于求

解线性矩阵不等式,文献[10]在误差方程中引入积分项,以提高干扰影响下的计算精度。文献[11]提出一种时变参数递归神经网络,并将其应用于时变 Sylvester 方程的在线求解。

与渐近稳定系统不同,终态吸引系统是一类具有有限时间收敛特性的动态系统^[12]。采用渐近收敛网络模型,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。将终态吸引系统理论用于递归神经网络,可以在 2 个方面改善计算性能:一方面提高收敛速度,另一方面提高计算精度。文献[13-15]提出的递归神经网络计算模型,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为有效的解决方案。这种终态神经网络已应用于求解线性矩阵方程^[16]、Lyapunov 方程^[17]以及 Sylvester 方程^[18,19]。

冗余机械臂是指末端执行器在执行给定任务时所具有自由度超出所需自由度的机械臂,超出的自由度可以使末端执行器在完成给定任务的同时完成

① 国家自然科学基金(61573320)资助项目。

② 男,1961 年生,博士,博士生导师,教授;研究方向:学习系统与神经计算;联系人,E-mail: mxsun@zjut.edu.cn
(收稿日期:2019-09-02)

其他各项性能指标。传统的冗余度解析方法是基于伪逆的方法。但是基于伪逆方法得到的关节运动可能不具有可重复性^[20,21]。文献[22]通过修正运动指标,形成重复运动规划(二次规划),进一步通过拉格朗日乘子法将二次规划转换为矩阵求逆问题,并以递归神经网络求解冗余度解析问题。近来,为了提高解算效率,基于终态神经网络求解运动规划引起人们的关注^[16-18]。从已发表的文献可以看出,目前提出的终态网络模型多具有无限值,不易于实现。

本文提出一类新颖的二次曲线型终态神经网络,包括双曲线型、椭圆型和抛物线型3种终态神经网络,其典型特征是网络各变量取值有限。文中详细分析了这类网络的有限时间收敛特性,并以双曲线型终态神经网络为例,检验其在时变矩阵计算与机器人轨迹规划方面的有效性。首先将其应用于一般时变线性矩阵方程的求解,它能够在有限时间内快速、准确地收敛到理论解。对于冗余机械臂重复规划问题,本文将重复运动指标取为终态收敛性能指标,通过将其转化为二次规划问题,在初始位置偏移的情况下,利用双曲线型终态神经网络进行求解,从而实现冗余机器人有限时间收敛的重复运动规划任务。

1 二次曲线型终态神经网络

为了解决时变神经计算问题,本节提出二次曲线型终态神经网络,并分析这种终态神经网络的有限时间收敛性、确定收敛时间。

1.1 双曲线型终态神经网络

提出2种双曲线型终态神经网络,用于时变问题求解。

双曲线型终态神经网络1的误差动态方程如下:

$$\dot{E}_{ij}(t) = -\varepsilon F_1(E_{ij}(t), a, b) \quad (1)$$

$$F_1(E_{ij}(t), a, b) =$$

$$\frac{b}{a} \sqrt{2a|E_{ij}(t)| + |E_{ij}(t)|^2} \operatorname{sgn}(E_{ij}(t))$$

其中, $E_{ij}(t)$ 为误差变量, $\varepsilon > 0$ 为用于调整收敛速

度的常值; $\delta, b > 0$ 分别表示双曲线半实轴与半虚轴长度。

定理1 由式(1)描述的双曲线型终态神经网络1全局有限时间收敛,其收敛时间为

$$T =$$

$$\frac{a}{\varepsilon b} \ln \left(1 + \frac{1}{a} (|E_{ij}(0)| + \sqrt{2a|E_{ij}(0)| + |E_{ij}(0)|^2}) \right) \quad (2)$$

证明 针对误差动态方程式(1),分2种情形讨论。

(1) 当 $E_{ij}(t) > 0$ 时, 定义 $Y = \sqrt{2aE_{ij}(t) + E_{ij}(t)^2}$, 将其代入式(1)得:

$$(E_{ij}(t) + a)^2 = Y^2 + a^2 \quad (3)$$

对其两端关于时间 t 求导:

$$(E_{ij}(t) + a)\dot{E}_{ij}(t) = Y\dot{Y} \quad (4)$$

这时,式(1)化简为

$$\dot{E}_{ij}(t) = -\frac{\varepsilon b}{a}Y \quad (5)$$

将式(3)与式(5)代入式(4),得:

$$\frac{dY}{dt} = -\frac{\varepsilon b}{a} \frac{\sqrt{Y^2 + a^2}}{a} \quad (6)$$

定义 $Y = a\tan Z$, 代入式(6)可得:

$$\sec Z \cdot dZ = -\frac{\varepsilon b}{a} dt \quad (7)$$

求取 $E_{ij}(t)$ 由 $E_{ij}(0)$ 收敛到 $E_{ij}(t) = 0$ 的时间:

$$\ln |\sec Z + \tan Z| \Big|_0^{\arctan \left(\frac{1}{a} (2aE_{ij}(0) + E_{ij}(0)^2) \right)} = -\frac{\varepsilon b}{a} T$$

解出 T 得:

$$T = \frac{a}{\varepsilon b} \ln \left(1 + \frac{1}{a} (E_{ij}(0) + \sqrt{2aE_{ij}(0) + E_{ij}(0)^2}) \right)$$

(2) 当 $E_{ij}(t) \leq 0$ 时, 同理可得:

$$T = \frac{a}{\varepsilon b} \ln \left(1 + \frac{1}{a} (-E_{ij}(0) + \sqrt{-2aE_{ij}(0) + E_{ij}(0)^2}) \right)$$

这样,误差从 $E_{ij}(0)$ 收敛到 $E_{ij}(t) = 0$ 所需时间满足式(2)。

双曲线型终态神经网络2的误差动态方程为

$$\dot{E}_{ij}(t) = -\varepsilon F_2(E_{ij}(t), \sigma) \quad (8)$$

$$F_2(E_{ij}(t), \sigma) =$$

$$\begin{cases} \frac{b}{a} \sqrt{2a|E_{ij}(t)| + |E_{ij}(t)|^2} \operatorname{sgn}(E_{ij}(t)) & |E_{ij}(t)| < \sigma \\ \operatorname{sgn}(E_{ij}(t)) & |E_{ij}(t)| \geq \sigma \end{cases}$$

其中 $\sigma = a\left(\frac{1}{b}\sqrt{b^2+1}-1\right) \geq 0$ 为 $F_2(E_{ij}(t), \sigma)$

的分段边界。

当 σ 取不同值时, 函数 $F_2(E_{ij}(t), \sigma)$ 的变化情形如图 1 所示。可以看出, 调整参数 σ 会改变该函数在原点附近的斜率; 当时间趋于 0 时, 该函数导数趋于无穷大, 从而使得网络有限时间收敛。

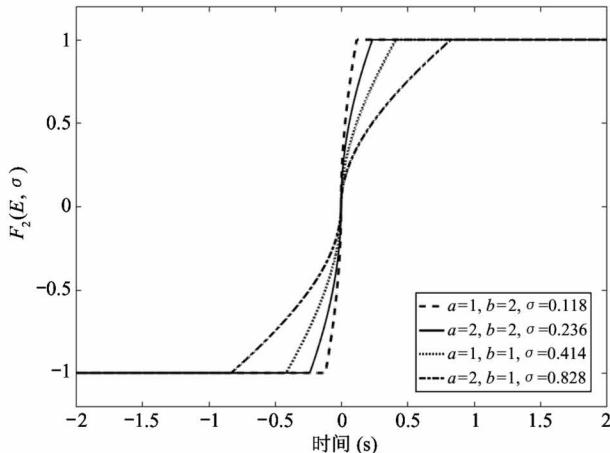


图 1 函数 $F_2(E_{ij}(t), \sigma)$

定理 2 由式(8)所描述的双曲线型神经网络 2 全局有限时间收敛, 其有限时间收敛性分 2 种情形。

(1) 当 $|E_{ij}(t)| < \sigma$ 时, 误差 $E_{ij}(t)$ 从 $E_{ij}(0)$ 收敛到原点所需时间为

$$T =$$

$$\frac{a}{\varepsilon b} \ln\left(1 + \frac{1}{a}(|E_{ij}(0)| + \sqrt{2a|E_{ij}(0)| + |E_{ij}(0)|^2})\right) \quad (9)$$

(2) 当 $|E_{ij}(t)| \geq \sigma$ 时, 该网络收敛时间为

$$T = \frac{|E_{ij}(0)| - \sigma}{\varepsilon} + \frac{a}{\varepsilon b} \ln\left(\frac{1 + \sqrt{1 + b^2}}{b}\right) \quad (10)$$

证明 针对误差方程式(8), 下面分 4 种情形分别讨论。

(1) 当 $0 \leq E_{ij}(t) < \delta$ 时, 类似定理 1 的证明,

求得的收敛时间为

$$T = \frac{a}{\varepsilon b} \ln\left(1 + \frac{1}{a}(E_{ij}(0) + \sqrt{2aE_{ij}(0) + E_{ij}(0)^2})\right)$$

(2) 当 $-\delta < E_{ij}(t) < 0$ 时, 同理可得:

$$T = \frac{a}{\varepsilon b} \ln\left(1 + \frac{1}{a}(-E_{ij}(0) + \sqrt{-2aE_{ij}(0) + E_{ij}(0)^2})\right)$$

这样, 当 $|E_{ij}(t)| < \delta$ 时, 收敛时间满足式(9)。

(3) 当 $E_{ij}(t) \geq \delta$ 时, 首先考虑误差从 $E_{ij}(0)$ 收敛到 σ 所需的时间 T_1 , 有:

$$\int_{E_{ij}(0)}^{\sigma} dE_{ij}(t) = - \int_0^{T_1} \varepsilon dt$$

解出 T_1 为

$$T_1 = \frac{E_{ij}(0) - \sigma}{\varepsilon}$$

误差由 $E_{ij}(t) = \sigma$ 收敛到 $E_{ij}(t) = 0$ 所需时间满足

$$\ln |\sec Z + \tan Z| \Bigg|_{\arcsin \frac{1}{b}}^0 = -\frac{\varepsilon b}{a} T_2$$

解得

$$T_2 = \frac{a}{\varepsilon b} \ln\left(\frac{1 + \sqrt{1 + b^2}}{b}\right)$$

因此, 当 $E_{ij}(t) \geq \sigma$ 时, $E_{ij}(t)$ 从 $E_{ij}(0)$ 收敛到 0 所需时间 T 为

$$T = T_1 + T_2 = \frac{E_{ij}(0) - \sigma}{\varepsilon} + \frac{a}{\varepsilon b} \ln\left(\frac{1 + \sqrt{1 + b^2}}{b}\right)$$

(4) 当 $E_{ij}(t) \leq -\sigma$ 时, 同理可得:

$$T = T_1 + T_2 = \frac{-E_{ij}(0) - \delta}{\varepsilon} + \frac{a}{\varepsilon b} \ln\left(\frac{1 + \sqrt{1 + b^2}}{b}\right)$$

这样, 当 $|E_{ij}(t)| \geq \delta$ 时, 该网络从 $E_{ij}(0)$ 收敛到 $E_{ij}(t) = 0$ 所需时间满足式(10)。

以上提出的是双曲线型网络, 除此以外, 二次曲线型终态神经网络还有 2 种形式, 即抛物线型与椭圆型终态网络。

1.2 抛物线型终态神经网络

抛物线型终态神经网络的误差动态方程如下:

$$\dot{E}_{ij}(t) = -\varepsilon F(E_{ij}(t), \sigma) \quad (11)$$

$$F(E_{ij}(t), \sigma) =$$

$$\begin{cases} \sqrt{\frac{|E_{ij}(t)|}{\sigma}} \operatorname{sgn}(E_{ij}(t)) & |E_{ij}(t)| < \sigma \\ \operatorname{sgn}(E_{ij}(t)) & |E_{ij}(t)| \geq \sigma \end{cases}$$

定理 3 由式(11)描述的抛物线型终态神经网络全局有限时间收敛, 其有限时间收敛性分以下 2 种情形。

(1) 当 $|E_{ij}(t)| < \sigma$ 时, 误差 $E_{ij}(t)$ 从初始误差 $E_{ij}(0)$ 收敛到 $E_{ij}(t) = 0$ 所需时间为

$$T = \frac{2\sqrt{|E_{ij}(0)| + \sigma}}{\varepsilon} \quad (12)$$

(2) 当 $|E_{ij}(t)| \geq \sigma$ 时, 该神经网络的收敛时间为

$$T = \frac{|E_{ij}(0)| + \sigma}{\varepsilon} \quad (13)$$

证明 依据误差动态方程式(1), 分 4 种情况分别讨论。

(1) 当 $0 \leq E_{ij}(t) < \sigma$ 时, 由式(11)可知:

$$\sqrt{E_{ij}(t)} \dot{E}_{ij}(t) = -\frac{\varepsilon}{\sqrt{\sigma}} \quad (14)$$

对式(14)两端积分:

$$\int_{E_{ij}(0)}^0 d\sqrt{E_{ij}(t)} = - \int_0^T \frac{\varepsilon}{2\sqrt{\sigma}} dt$$

误差 $E_{ij}(t)$ 由 $E_{ij}(0)$ 收敛至原点所需时间为

$$T = \frac{2\sqrt{|E_{ij}(0)|\sigma}}{\varepsilon}$$

(2) 当 $-\sigma < E_{ij}(t) < 0$ 时, 与情形(1)推导类似,

$$T = \frac{2\sqrt{-E_{ij}(0)\sigma}}{\varepsilon}$$

因此, 当 $|E_{ij}(t)| < \sigma$ 时, 收敛时间如式(12)所示。

(3) 当 $E_{ij}(t) \geq \sigma$ 时, 类似定理 2 的证明, 误差 $E_{ij}(t)$ 从 $E_{ij}(0)$ 收敛到 $E_{ij}(t) = \sigma$ 所需时间为 $T_1 = \frac{|E_{ij}(0)| - \sigma}{\varepsilon}$ 。误差从 $E_{ij}(t) = \sigma$ 收敛至原点的时间

T_2 满足:

$$\int_{\sigma}^0 d\sqrt{E_{ij}(t)} = - \int_0^{T_2} \frac{\varepsilon}{2\sqrt{\sigma}} dt$$

因此,

$$T_2 = \frac{2\sigma}{\varepsilon}$$

这样, 从初始误差 $E_{ij}(0)$ 收敛到原点的时间为

$$T = T_1 + T_2 = \frac{|E_{ij}(0)| + \sigma}{\varepsilon}$$

(4) 当 $E_{ij}(t) \leq -\sigma$ 时, 与情形(3)推导类似,

$$T = \frac{-E_{ij}(0) + \sigma}{\varepsilon}$$

故当 $|E_{ij}(t)| \geq \sigma$ 时, 收敛时间如式(13)所示。

1.3 椭圆型终态神经网络

椭圆型终态神经网络的误差动态方程如下:

$$\dot{E}_{ij}(t) = -\varepsilon F(E_{ij}(t), \sigma) \quad (15)$$

$$F(E_{ij}(t), \sigma) =$$

$$\begin{cases} b \sqrt{\frac{2|E_{ij}(t)|}{a} - \frac{|E_{ij}(t)|^2}{a^2}} \operatorname{sgn}(E_{ij}(t)) \\ \operatorname{sgn}(E_{ij}(t)) \end{cases} \begin{array}{l} |E_{ij}(t)| < \sigma \\ |E_{ij}(t)| \geq \sigma \end{array}$$

其中, $a > 0, b \geq 1, \sigma = a \left(1 - \sqrt{1 - \frac{1}{b^2}}\right) \geq 0$ 为函数 $F(E_{ij}(t), \sigma)$ 的分段边界。

定理 4 由式(15)描述的椭圆型终态神经网络全局有限时间收敛, 其有限时间收敛性分以下 2 种情形。

(1) 当 $|E_{ij}(t)| < \sigma$ 时, 该网络从初始误差 $E_{ij}(0)$ 收敛到 $E_{ij}(t) = 0$ 所需时间为

$$T = \frac{a}{\varepsilon b} \arcsin \left(\frac{\sqrt{2|E_{ij}(0)|a} - |E_{ij}(0)|^2}{a} \right) \quad (16)$$

(2) 当 $|E_{ij}(t)| \geq \sigma$ 时, 该网络的收敛时间为

$$T = \frac{|E_{ij}(0)| - \sigma}{\varepsilon} + \frac{a}{\varepsilon b} \arcsin \frac{1}{b} \quad (17)$$

证明 依据误差动态方程式(15), 分 4 种情形分别讨论。

(1) 当 $0 \leq E_{ij}(t) < \sigma$ 时, 定义 $Y = \sqrt{2E_{ij}(t)a - E_{ij}(t)^2}$, 将其代入式(15),

$$Y^2 + (E_{ij}(t) - a)^2 = a^2 \quad (18)$$

对式(18)两端关于时间求导:

$$\dot{Y} = -\frac{\varepsilon b}{a} \sqrt{a^2 - Y^2} \quad (19)$$

定义 $Z = \arcsin \frac{Y}{a}$, 代入式(19)可得:

$$\frac{dZ}{dt} = -\frac{\varepsilon b}{a} \quad (20)$$

对式(20)两端积分, 有:

$$\int_{\arcsin\left(\frac{\sqrt{2E_{ij}(0)a-E_{ij}(0)^2}}{a}\right)}^0 dZ = - \int_0^T \frac{\varepsilon b}{a} dt$$

可求得误差 $E_{ij}(t)$ 由 $E_{ij}(0)$ 收敛到原点的时间 T 满足:

$$T = \frac{a}{\varepsilon b} \arcsin\left(\frac{\sqrt{2E_{ij}(0)a-E_{ij}(0)^2}}{a}\right)$$

(2) 当 $-\sigma < E_{ij}(t) < 0$ 时, 同理可得:

$$T = \frac{a}{\varepsilon b} \arcsin\left(\frac{\sqrt{-2E_{ij}(0)a-E_{ij}(0)^2}}{a}\right)$$

故当 $|E_{ij}(t)| < \sigma$ 时, 收敛时间如式(16)所示。

(3) 当 $E_{ij}(t) \geq \sigma$ 时, 同定理 2 证明的步骤(3), 误差从 $E_{ij}(0)$ 收敛到 $E_{ij}(t) = \sigma$ 所需时间 T_1

$$= \frac{E_{ij}(0) - \sigma}{\varepsilon}。$$

误差从 $E_{ij}(t) = \sigma$ 收敛到 0 所需的时间 T_2 满足:

$$\int_{\arcsin\frac{1}{b}}^0 dZ = - \int_0^{T_2} \frac{\varepsilon b}{a} dt$$

解得,

$$T_2 = \frac{a}{\varepsilon b} \arcsin \frac{1}{b}$$

这样, 误差从初始误差 $E_{ij}(0)$ 收敛到 $E_{ij}(t) = 0$ 所需的时间为

$$T = T_1 + T_2 = \frac{E_{ij}(0) - \sigma}{\varepsilon} + \frac{a}{\varepsilon b} \arcsin \frac{1}{b}$$

(4) 当 $E_{ij}(t) \leq -\sigma$ 时, 同理可得:

$$T = \frac{-E_{ij}(0) - \sigma}{\varepsilon} + \frac{a}{\varepsilon b} \arcsin \frac{1}{b}$$

故当 $|E_{ij}(t)| \geq \sigma$ 时, 收敛时间如式(17)所示。

2 时变线性矩阵方程求解

考虑下述一般时变线性矩阵方程:

$$\sum_{k=1}^p \mathbf{A}_k(t) \mathbf{X}(t) \mathbf{B}_k(t) = \mathbf{C}(t) \quad (21)$$

其中, $\mathbf{A}_k(t) \in R^{n \times n}$, $\mathbf{B}_k(t) \in R^{m \times m}$, $\mathbf{C}(t) \in R^{n \times m}$ 为时变系数矩阵, $\mathbf{X}(t) \in R^{n \times m}$ 为待求解未知时变矩阵。

本文的计算目的是以双曲线型终态神经网络求解未知时变矩阵 $\mathbf{X}(t)$, 在有限时间收敛后, 获得时

变矩阵的解。为此目的, 依据式(21), 定义矩阵值误差函数为

$$\mathbf{E}(t) = \sum_{k=1}^p \mathbf{A}_k(t) \mathbf{X}(t) \mathbf{B}_k(t) - \mathbf{C}(t) \quad (22)$$

对式(22)左右两边同时求导, 可得到:

$$\begin{aligned} \dot{\mathbf{E}}(t) &= \sum_{k=1}^p \dot{\mathbf{A}}_k(t) \mathbf{X}(t) \mathbf{B}_k(t) + \sum_{k=1}^p \mathbf{A}_k(t) \dot{\mathbf{X}}(t) \mathbf{B}_k(t) \\ &\quad + \sum_{k=1}^p \mathbf{A}_k(t) \mathbf{X}(t) \dot{\mathbf{B}}_k(t) - \dot{\mathbf{C}}(t) \end{aligned} \quad (23)$$

为使 $\mathbf{E}(t)$ 全局有限时间收敛于 0, 利用终态神经网络的误差动态特性, 以便规定导数信号 $\dot{\mathbf{E}}_{ij}(t)$ 的变化性状。以双曲线型终态神经网络为例进行求解, 将式(23)代入其误差方程, 可得下述终态神经网络模型。

$$\begin{aligned} \sum_{k=1}^p \dot{\mathbf{A}}_k(t) \mathbf{X}(t) \mathbf{B}_k(t) + \sum_{k=1}^p \mathbf{A}_k(t) \dot{\mathbf{X}}(t) \mathbf{B}_k(t) - \dot{\mathbf{C}}(t) \\ + \sum_{k=1}^p \mathbf{A}_k(t) \mathbf{X}(t) \dot{\mathbf{B}}_k(t) + \varepsilon \mathbf{F}_{1,2}(\mathbf{E}(t), \sigma) = 0 \end{aligned} \quad (24)$$

为了验证所提出的双曲线型终态神经网络在计算时变线性矩阵方程式(21)方面的有效性, 本文设置时变线性矩阵方程式(21)不同的系数矩阵, 给出具体形式的时变矩阵方程, 包括时变 Lyapunov 方程和时变 Sylvester 方程。分别给出将双曲线型终态神经网络用于求解这 2 类时变矩阵方程的计算结果, 并与递归神经网络对于同一算例的结果进行比较, 以验证所提出的双曲线型终态神经网络的收敛性能。

2.1 时变 Lyapunov 方程

在时变线性矩阵方程式(21)中, 置 $k = 2, m = n$, $\mathbf{B}_1(t) = \mathbf{A}_2(t) = \mathbf{I}$ 且 $\mathbf{B}_2(t) = \mathbf{A}_1^\top(t)$, 该方程简化为下述时变 Lyapunov 方程:

$$\mathbf{A}_1(t) \mathbf{X}(t) + \mathbf{X}(t) \mathbf{A}_1^\top(t) = \mathbf{C}(t) \quad (25)$$

定义矩阵值误差函数为

$$\mathbf{E}(t) = \mathbf{A}_1(t) \mathbf{X}(t) + \mathbf{X}(t) \mathbf{A}_1^\top(t) - \mathbf{C}(t) \quad (26)$$

对式(26)左右两边同时求导:

$$\begin{aligned} \dot{\mathbf{E}}(t) &= \dot{\mathbf{A}}_1(t) \mathbf{X}(t) + \mathbf{A}_1(t) \dot{\mathbf{X}}(t) + \dot{\mathbf{X}}(t) \mathbf{A}_1^\top(t) \\ &\quad + \mathbf{X}(t) \dot{\mathbf{A}}_1^\top(t) - \dot{\mathbf{C}}(t) \end{aligned} \quad (27)$$

将式(27)代入网络误差动态方程式(1)和式(8), 可得如下终态神经网络模型。

$$\dot{\mathbf{A}}_1(t) \mathbf{X}(t) + \mathbf{A}_1(t) \dot{\mathbf{X}}(t) + \dot{\mathbf{X}}(t) \mathbf{A}_1^\top(t) + \mathbf{X}(t) \dot{\mathbf{A}}_1^\top(t)$$

$$-\dot{\mathbf{C}}(t) + \varepsilon \mathbf{F}_{1,2}(\mathbf{E}(t), \sigma) = 0 \quad (28)$$

例 1 利用双曲线型终态网络模型式(28)求解时变 Lyapunov 方程式(25), 其中时变矩阵为

$$\mathbf{A}_1(t) = \begin{pmatrix} -1 - \frac{1}{2}\cos(2t) & \frac{1}{2}\sin(2t) \\ \frac{1}{2}\sin(2t) & -1 + \frac{1}{2}\cos(2t) \end{pmatrix} \quad (29)$$

$$\mathbf{C}(t) = \begin{bmatrix} \sin(2t) & \cos(2t) \\ -\cos(2t) & \sin(2t) \end{bmatrix} \quad (30)$$

取双曲线型终态网络的误差吸引参数 $\varepsilon = 1$, 半轴长分别为 $a = 1$, $b = 1$; 因此, 分段边界 $\sigma = a\left(\frac{1}{b}\sqrt{b^2+1}-1\right) = \sqrt{2}-1$ 。仿真中解的初始值取为 $X(0) = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$ 。

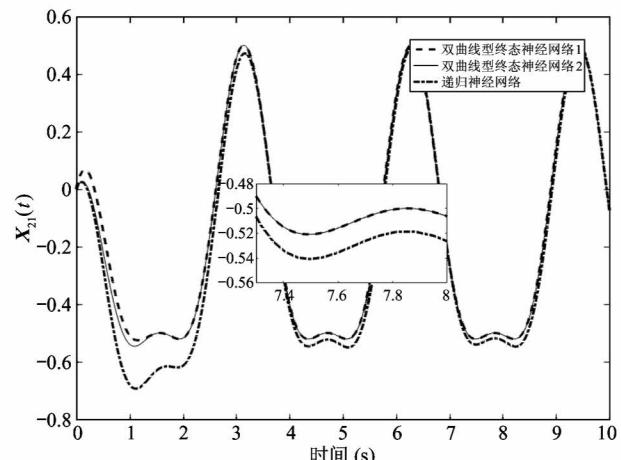
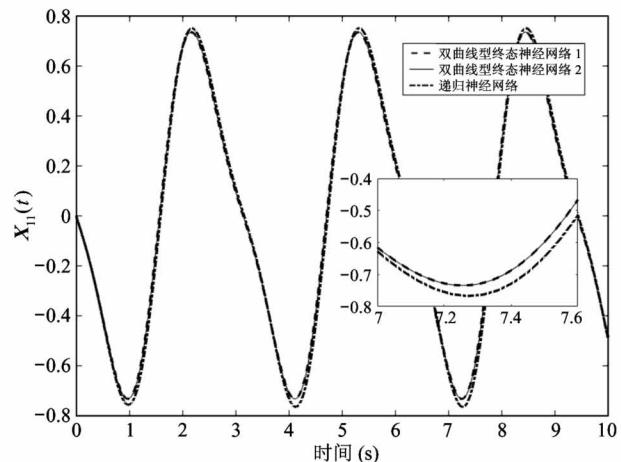
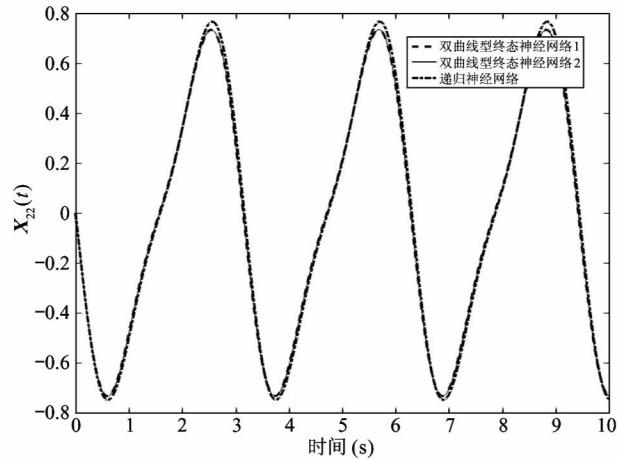
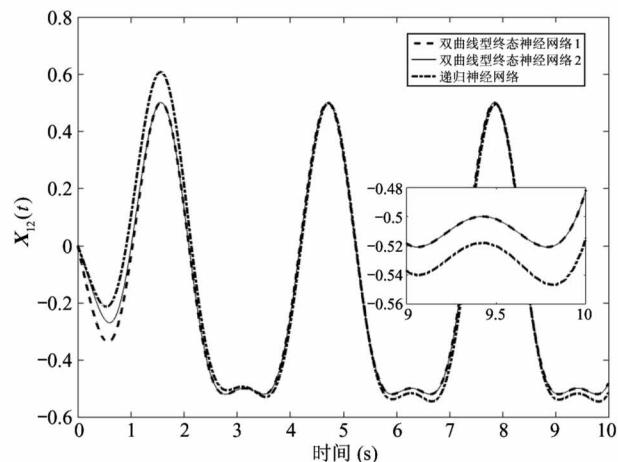


图 2 时变 Lyapunov 方程的求解结果

图 2 和图 3 分别给出了双曲线型终态网络及递归网络(其误差动态方程为 $\dot{\mathbf{E}}_{ij}(t) = -\varepsilon \mathbf{E}_{ij}(t)$, 参数 $\varepsilon = 1$)的求解结果及计算误差。可以看出, 递归网络求解误差呈指数收敛, 各分量收敛误差在 $[-0.05, 0.07]$ 范围内波动; 双曲线型终态网络计算得到的解由初始值有限时间收敛于理论解, 且各分量误差均小于 2×10^{-6} 。

图 4 给出了双曲线型终态网络与递归网络的计算误差对比, 其中, $\|\mathbf{E}\|_F = \sqrt{\text{tr}(\mathbf{E}^T \mathbf{E})} = \sqrt{\sum_{i=1}^2 \sum_{j=1}^2 \mathbf{E}_{ij}}$ 。通过定理 1 可得基于双曲线型终态网络 1 求解所需收敛时间为 1.3176 s。在仿真中, 当误差的 F 范数小于 2×10^{-6} 时, 求解所需时间为 1.3170 s; 通过定理 2 可得基于双曲线型终态网络 2 求解所需收敛时间为 1.4672 s。在仿真中, 当误差的



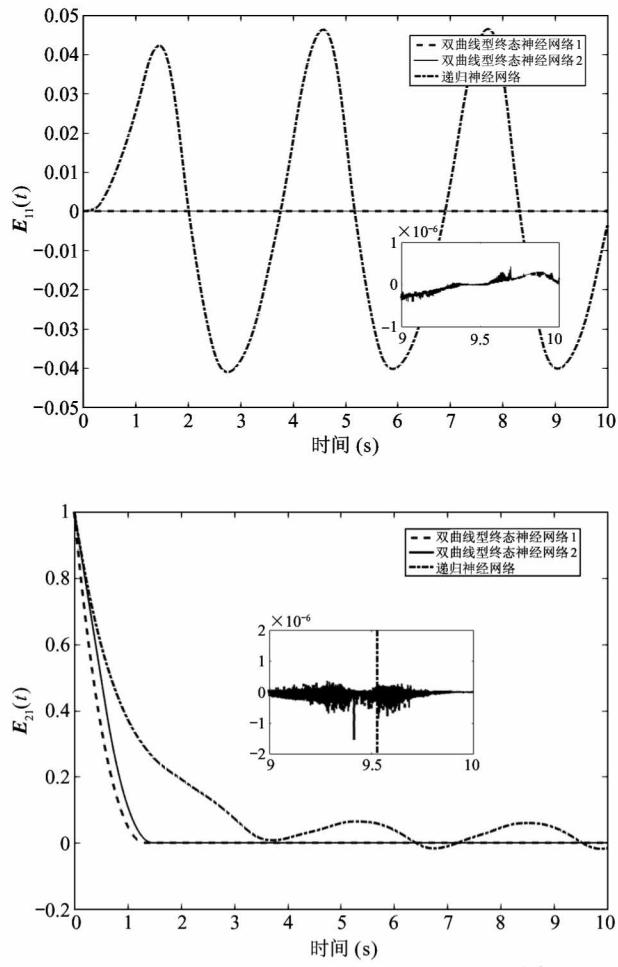


图3 时变 Lyapunov 方程计算误差轨迹

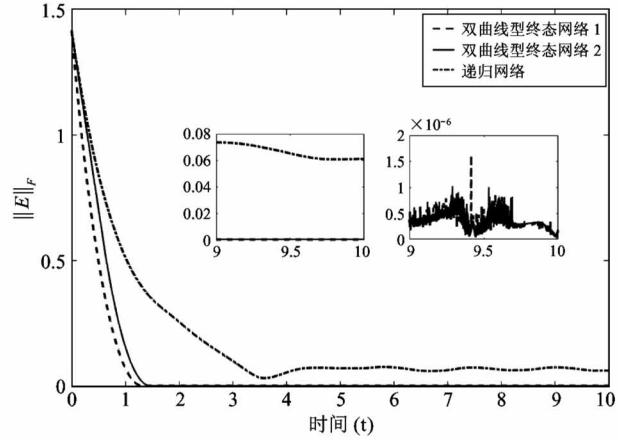
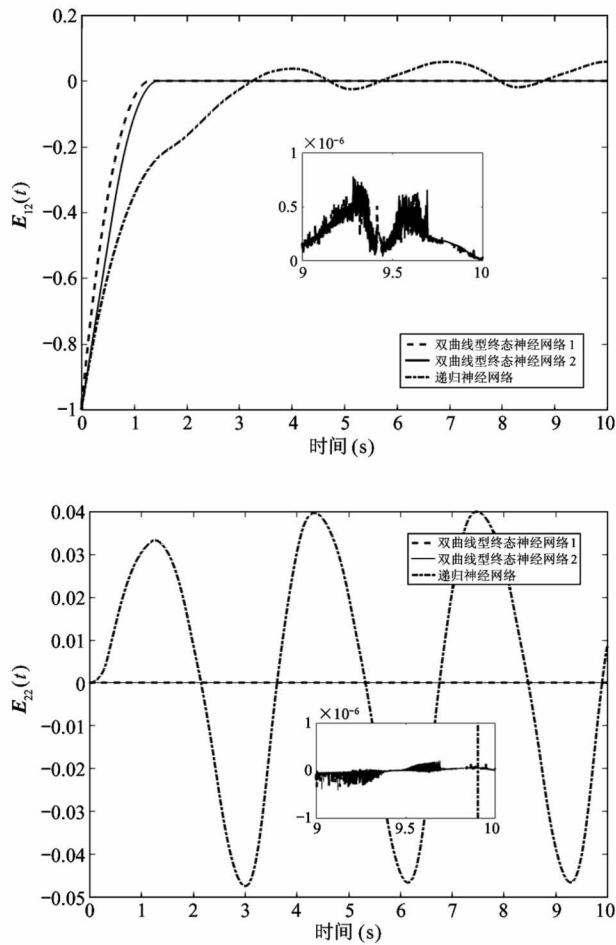


图4 时变 Lyapunov 方程计算误差对比

F 范数小于 2×10^{-6} 时, 求解所需时间为 1.4689 s。基于双曲线型终态网络求解得到的误差的 F 范数在 $T \in [9, 10]$ 内误差波动范围为 $[0, 2 \times 10^{-6}]$; 基于递归网络求解得到的误差的 F 范数在 $T \in [9, 10]$ 内误差波动范围为 $[0.05, 0.08]$ 。



2.2 时变 Sylvester 方程

在时变线性矩阵方程式(21)中, 置 $k = 2, m = n, \mathbf{B}_1(t) = \mathbf{A}_2(t) = \mathbf{I}$, 该方程简化为下述时变 Sylvester 方程:

$$\mathbf{A}_1(t)\mathbf{X}(t) + \mathbf{X}(t)\mathbf{B}_2(t) = \mathbf{C}(t) \quad (31)$$

定义矩阵值误差函数为

$$\mathbf{E}(t) = \mathbf{A}_1(t)\mathbf{X}(t) + \mathbf{X}(t)\mathbf{B}_2(t) - \mathbf{C}(t) \quad (32)$$

对式(32)左右两边同时求导, 可得:

$$\dot{\mathbf{E}}(t) = \dot{\mathbf{A}}_1(t)\mathbf{X}(t) + \mathbf{A}_1(t)\dot{\mathbf{X}}(t) + \dot{\mathbf{X}}(t)\mathbf{B}_2(t) + \mathbf{X}(t)\dot{\mathbf{B}}_2(t) - \dot{\mathbf{C}}(t) \quad (33)$$

将式(33)代入网络误差动态方程式(1)和式(8), 可得双曲线型终态神经网络模型。

$$\dot{\mathbf{A}}_1(t)\mathbf{X}(t) + \mathbf{A}_1(t)\dot{\mathbf{X}}(t) + \dot{\mathbf{X}}(t)\mathbf{B}_2(t) + \mathbf{X}(t)\dot{\mathbf{B}}_2(t) - \dot{\mathbf{C}}(t) + \varepsilon\mathbf{F}_{1,2}(\mathbf{E}_{ij}(t), \sigma) = 0 \quad (34)$$

例 2 利用双曲线型终态网络模型式(34)求解时变 Sylvester 方程式(31), 其中时变矩阵为

$$\mathbf{A}_1(t) = \begin{bmatrix} \sin(3t) & \cos(3t) \\ -\cos(3t) & \sin(3t) \end{bmatrix} \quad (35)$$

$$\mathbf{B}_2(t) = \begin{bmatrix} 2 & 0 \\ 0 & 3 \end{bmatrix} \quad (36)$$

$$\mathbf{C}(t) = \begin{bmatrix} \sin(2t) & \cos(2t) \\ -\cos(2t) & \sin(2t) \end{bmatrix} \quad (37)$$

取双曲线型终态网络误差吸引参数 $\varepsilon = 5$, 参

数 $a = 1$, $b = \frac{1}{\sqrt{3}}$, $\sigma = 1$ 。在仿真中解的初始值取为

$$\mathbf{X}(0) = \begin{bmatrix} 1 & 0 \\ -1 & 0 \end{bmatrix}。$$

图 5 和图 6 给出了双曲线型终态网络的求解结果及计算误差。当所有解元素的误差小于 5×10^{-3} 时,

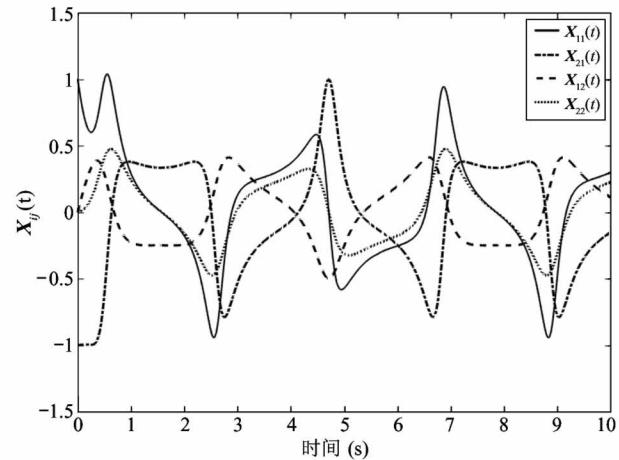


图 5 双曲线型终态网络 1 求解结果及计算误差

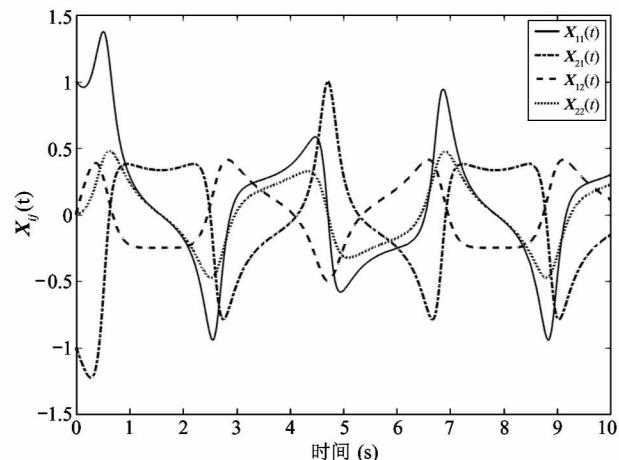
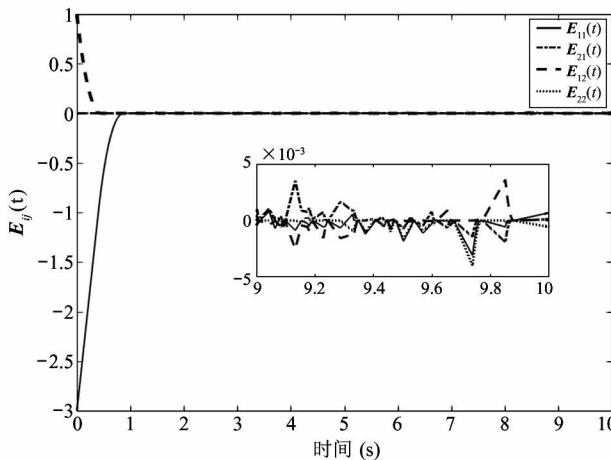
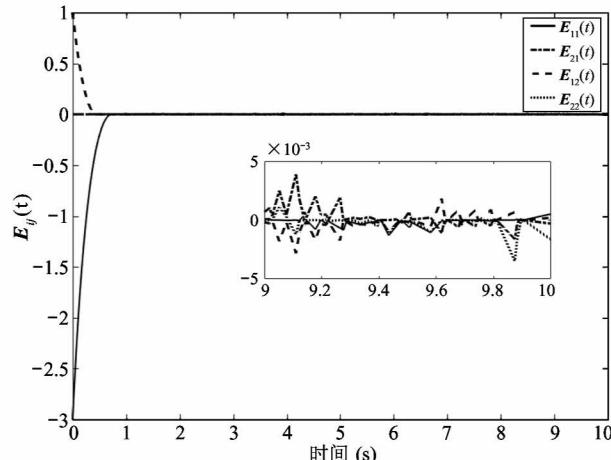


图 6 双曲线型终态网络 2 求解结果及计算误差

基于双曲线型终态网络 1 求解所需时间 $t = 0.699$ s; 基于双曲线型终态网络 2 求解所需时间 $t = 0.837$ s。当 $T \in [9, 10]$ 范围内, 基于双曲线型终态网络求解得到的误差波动范围为 $[-5 \times 10^{-3}, 5 \times 10^{-3}]$ 。图 7 为分段边界 σ 取不同值时, 利用双曲线型终态网络 2 求解时误差的 F 范数, 可以看出, 随着 σ 的减小, $\mathbf{X}(t)$ 收敛到理论解 $\mathbf{X}^*(t)$ 的速度加快。

上述算例表明, 所提出的双曲线型终态网络对于解决一般时变线性矩阵问题是有效的。相比于具有渐近收敛动态特性的递归网络, 双曲线型终态网络具有有限时间收敛性、收敛速度快、计算精度高的特点。



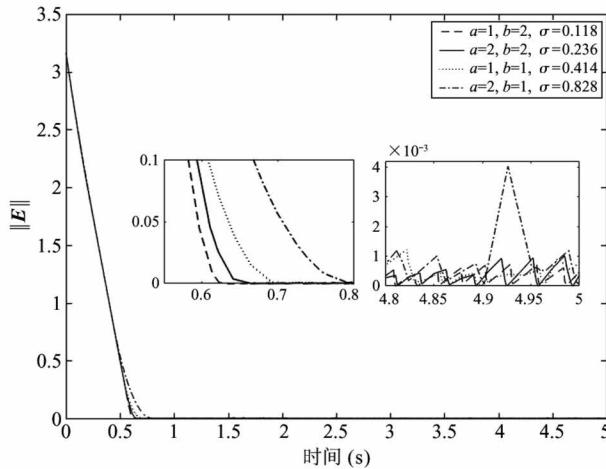


图 7 双曲线型终态神经网络 2 求解误差范数

3 冗余机械臂重复运动规划

本节将双曲线型终态网络应用于求解冗余机械臂重复运动规划,并以 PA10 机械臂为例验证所提出终态网络的适用性。

考虑 n 自由度机械臂的关节角度和末端执行器位移关系

$$\mathbf{r}(t) = \mathbf{f}(\boldsymbol{\theta}(t)) \quad (38)$$

其中, $\mathbf{r}(t)$ 表示末端执行器在笛卡尔坐标系下的位姿变量, $\boldsymbol{\theta}(t)$ 表示关节角。末端笛卡尔空间和关节空间的各变量微分之间的关系为

$$\dot{\mathbf{r}}(t) = \mathbf{J}(\boldsymbol{\theta})\dot{\boldsymbol{\theta}}(t) \quad (39)$$

其中, $\dot{\mathbf{r}}(t)$ 是 $\mathbf{r}(t)$ 的时间导数, $\dot{\boldsymbol{\theta}}(t)$ 是关节速度, $\mathbf{J}(\boldsymbol{\theta})$ 是机械臂的雅克比矩阵。

为了执行重复运动任务,可引入重复运动指标作为优化准则,将冗余机械臂运动规划描述为相应的二次规划问题,通过求解该优化问题形成重复运动规划方案。与文献[18]不同的是,本文将重复运动指标设计为终态收敛性能指标,其具体形式如下:

$$\frac{1}{2} \|\dot{\boldsymbol{\theta}}(t) + \mathbf{g}(\boldsymbol{\theta})\|^2 \quad (40)$$

其中,

$$\mathbf{g}(\boldsymbol{\theta}) =$$

$$-\beta_\theta \frac{b}{a} \sqrt{2a|\boldsymbol{\theta}(t) - \boldsymbol{\theta}_d(0)| + |\boldsymbol{\theta}(t) - \boldsymbol{\theta}_d(0)|^2}$$

$$\operatorname{sgn}(\boldsymbol{\theta}(t) - \boldsymbol{\theta}_d(0))$$

容易看出,式(40)最小解满足 $\dot{\boldsymbol{\theta}}(t) + \mathbf{g}(\boldsymbol{\theta}) = 0$, 此为本文提出的一个终态吸引系统。考虑到末端执行

器的约束关系,不能直接求解式(40),需形成相应的重复运动规划方案:

$$\begin{aligned} & \text{minimize} \quad \frac{1}{2} (\dot{\boldsymbol{\theta}}(t) + \mathbf{g}(\boldsymbol{\theta}))^\top (\dot{\boldsymbol{\theta}}(t) + \mathbf{g}(\boldsymbol{\theta})) \\ & \text{subject to} \quad \mathbf{J}(\boldsymbol{\theta})\dot{\boldsymbol{\theta}}(t) = \dot{\mathbf{r}}(t) + \beta_r(\mathbf{r}_d - \mathbf{f}(\boldsymbol{\theta})) \end{aligned} \quad (41)$$

其中, β_θ 是一个设计参数, $\boldsymbol{\theta}(t) - \boldsymbol{\theta}_d(0)$ 表示各关节角与初始期望关节角位移偏差, \mathbf{r}_d 、 $\dot{\mathbf{r}}_d$ 分别表示机械臂末端执行器期望轨迹和速度向量。考虑到机械臂的初始位置不位于期望轨迹上的情况,通过减小末端执行器期望路径与实际运动轨迹位置间的误差 $\mathbf{r}_d - \mathbf{f}(\boldsymbol{\theta})$ 以实现轨迹回归期望轨迹的目的。这里,增大增益 β_r 会提高收敛速率。

定义拉格朗日函数如下:

$$\begin{aligned} \mathbf{L}(\dot{\boldsymbol{\theta}}, \boldsymbol{\lambda}, t) = & \frac{1}{2} \dot{\boldsymbol{\theta}}^\top(t) \dot{\boldsymbol{\theta}}(t) + \mathbf{g}(\boldsymbol{\theta})^\top \dot{\boldsymbol{\theta}}(t) \\ & + \boldsymbol{\lambda}^\top(\mathbf{J}(\boldsymbol{\theta})\dot{\boldsymbol{\theta}}(t) - \dot{\mathbf{r}}_d(t) \\ & - \beta_r(\mathbf{r}_d - \mathbf{f}(\boldsymbol{\theta}))) \end{aligned} \quad (42)$$

其中, $\boldsymbol{\lambda}(t) \in R^m$ 为拉格朗日乘子向量。分别求取拉格朗日函数对 $\dot{\boldsymbol{\theta}}(t)$ 、 $\boldsymbol{\lambda}$ 的偏导,并令其为 0, 可得如下矩阵方程:

$$\mathbf{W}(t)\mathbf{Y}(t) = \mathbf{V}(t) \quad (43)$$

其中,

$$\mathbf{W}(t) = \begin{bmatrix} \mathbf{I} & \mathbf{J}^\top(\boldsymbol{\theta}) \\ \mathbf{J}(\boldsymbol{\theta}) & 0 \end{bmatrix}$$

$$\mathbf{Y}(t) = \begin{bmatrix} \dot{\boldsymbol{\theta}}(t) \\ \boldsymbol{\lambda}(t) \end{bmatrix}$$

$$\mathbf{V}(t) = \begin{bmatrix} -\mathbf{g}(\boldsymbol{\theta}) \\ \dot{\mathbf{r}}(t) + \beta_r(\mathbf{r}_d - \mathbf{f}(\boldsymbol{\theta})) \end{bmatrix}$$

为求解由式(41)所示的二次规划问题,定义误差:

$$\mathbf{E}(t) = \mathbf{W}(t)\mathbf{Y}(t) - \mathbf{V}(t) \quad (44)$$

将式(44)代入误差动态方程,可得终态神经网络模型。

$$\begin{aligned} \dot{\mathbf{Y}}(t) = & \dot{\mathbf{Y}}(t) + \dot{\mathbf{W}}(t)\mathbf{Y}(t) + \mathbf{W}(t)\dot{\mathbf{Y}}(t) \\ & - \dot{\mathbf{V}}(t) + \varepsilon \mathbf{F}(\mathbf{E}_{ij}(t), \sigma) \end{aligned} \quad (45)$$

据此模型完成求解过程,便可得到机械臂各个关节角轨迹。

给定 PA10 的期望末端轨迹为圆形,其圆心坐标为($x = 0.2$ m, $y = 0$ m, $z = 0$ m),半径为 0.2 m,

圆面与平面的夹角为 $\frac{\pi}{6}$ rad。机械臂期望回拢角度

$$\boldsymbol{\theta}_d(0) = \left[0 \quad \frac{\pi}{4} \quad 0 \quad \frac{2\pi}{3} \quad 0 \quad -\frac{\pi}{4} \quad 0 \right]^T \text{rad},$$

末端执行器完成轨迹所需时间 $T = 10$ s。考虑到机械臂的初始位置不在期望的运动轨迹上, 将机械臂的 7 个关节角度初值设为 $\boldsymbol{\theta}(0) = \left[0 \quad \frac{\pi}{4} + 0.5 \quad 0 \quad \frac{2\pi}{3} \quad 0 \quad -\frac{\pi}{4} \quad 0 \right]^T \text{rad}$ 。

图 8 分别给出了求解获得的 PA10 末端执行器在空间中的运动轨迹及各个关节运动轨迹。可以看出, 末端执行器的初始位置不在期望轨迹上, 但各个

关节的轨迹在运行一个周期后是闭合的, 实现了重复运动控制。为了说明该终端网络在重复运动规划中的有效性, 机械臂末端执行器完成圆轨迹过程中相应的关节角和关节角速度轨迹如图 9 所示。机械臂各关节角最终收敛于期望关节角位置, 关节角速度收敛于 0, 机械臂运动停止。末端执行器各个位置误差如图 10 所示, 随着时间增加 ($T = 10$ s), 末端执行器的终值位置误差精度在 XYZ 轴 3 个方向上达到 4×10^{-6} , 实际轨迹与期望轨迹吻合, 从而实现由初始位置收敛于期望轨迹。

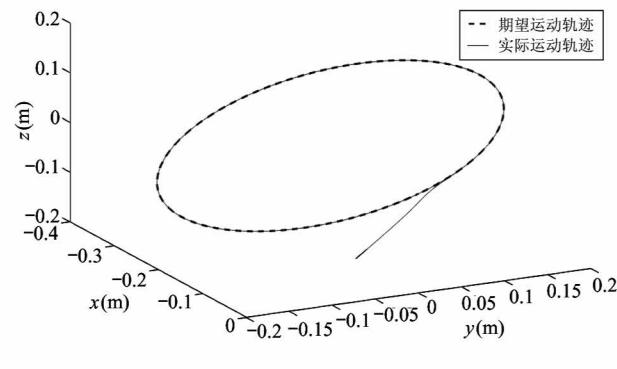


图 8 PA10 末端执行器运动轨迹及各关节运动轨迹

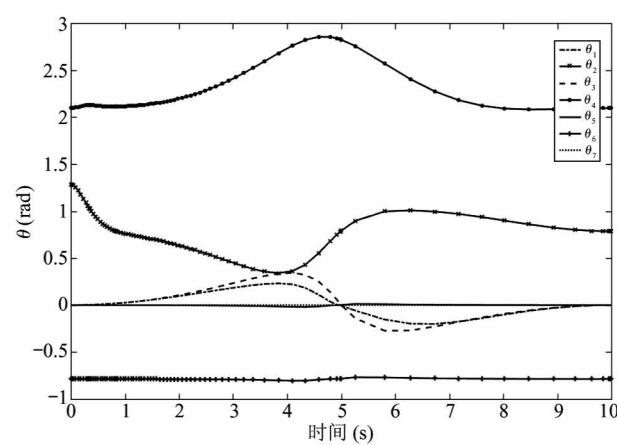
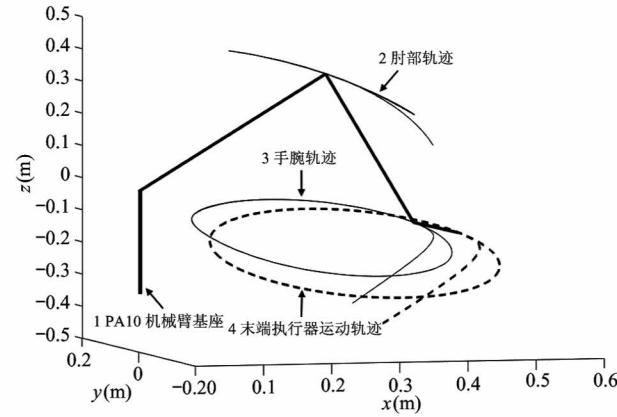


图 9 PA10 关节角及关节角速度轨迹

另外, 表 1 给出了当轨迹规划完成时的双曲线型终端网络及渐近收敛网络各关节角实际回拢角度与其期望角度之间的偏差对比, 其中, 取参数 $\varepsilon = 2$ 。

4 结 论

本文提出一种新的二次曲线型终端神经网络并

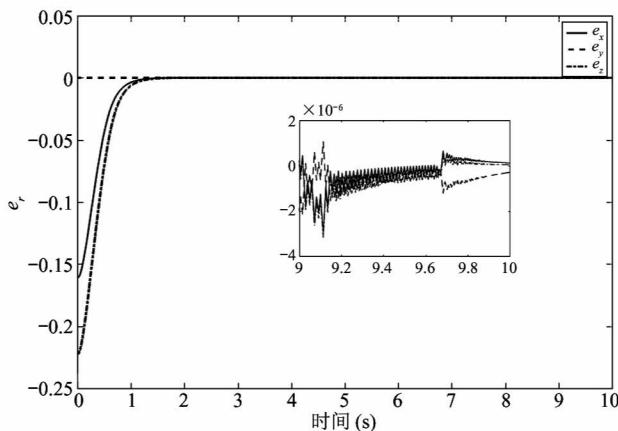


图 10 末端执行器各位置误差

表 1 各关节角实际回摆角度与期望角度偏差

网络求解器(rad)	双曲线型终态 神经网络 1	渐近神经网络
$\theta_1(10) - \theta_{d1}(0)$	-2.7961×10^{-7}	-3.8450×10^{-4}
$\theta_2(10) - \theta_{d2}(0)$	-1.9893×10^{-9}	3.1900×10^{-4}
$\theta_3(10) - \theta_{d3}(0)$	-2.7239×10^{-7}	-3.6540×10^{-4}
$\theta_4(10) - \theta_{d4}(0)$	-2.1408×10^{-7}	3.1379×10^{-4}
$\theta_5(10) - \theta_{d5}(0)$	1.3764×10^{-8}	6.5530×10^{-5}
$\theta_6(10) - \theta_{d6}(0)$	1.6807×10^{-8}	1.0427×10^{-4}
$\theta_7(10) - \theta_{d7}(0)$	0	0

证明其有限时间收敛性。以双曲线型终态神经网络为例,将其用于一般时变线性矩阵方程的求解,使时变线性矩阵方程解能够快速地在有限时间内收敛到其理论解。将双曲线型终态神经网络应用于冗余机械臂重复运动规划,并将重复运动指标设计为终态收敛性能指标,在初始位置偏移的情况下,实现冗余机械臂快速有限时间内收敛的重复运动规划任务,进一步说明了双曲线型终态神经网络的适用性。

参考文献

- [1] Wang J. Electronic realisation of recurrent neural work for solving simultaneous linear equations[J]. *Electronics Letters*, 1992, 28(5): 493-495
- [2] Wang J. Recurrent neural networks for computing pseudo-inverses of rank-deficient matrices[J]. *SIAM Journal on Scientific Computing*, 1997, 18(5): 1479-1493
- [3] Kennedy M P, Chua L O. Neural networks for nonlinear programming[J]. *IEEE Transactions on Circuits and Systems*, 1988, 35(5): 554-562
- [4] 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
- [5] Xia Y. A new neural network for solving linear and quadratic programming problems [J]. *IEEE Transactions on Neural Networks*, 1996, 7(6): 1544-1548
- [6] Zhang Y, 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
- [7] Zhang Y, Ma W, Cai B. From Zhang neural network to Newton iteration for matrix inversion[J]. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 2009, 56(7): 1405-1415
- [8] Xiao L, Zhang Y. Zhang neural network versus gradient neural network for solving time-varying linear inequalities [J]. *IEEE Transactions on Neural Networks*, 2011, 22(10): 1676-1684
- [9] Guo D, Zhang Y. Zhang neural network for online solution of time-varying linear matrix inequality aided with an equality conversion [J]. *IEEE Transactions on Neural Networks and Learning Systems*, 2014, 25(2): 370-382
- [10] Jin L, Zhang Y, Li S. Integration-enhanced Zhang neural network for real-time-varying matrix inversion in the presence of various kinds of noise[J]. *IEEE Transactions on Neural Networks and Learning Systems*, 2016, 27(12): 2615-2627
- [11] Zhang Z, Zheng L, Weng J, et al. A new varying-parameter recurrent neural-network for online solution of time-varying Sylvester equation[J]. *IEEE Transactions on Cybernetics*, 2018, 8(11): 3135-3148
- [12] Zak M. Terminal attractors for addressable memory in neural networks[J]. *Physics Letters A*, 1988, 133(1): 18-22
- [13] Li S, Li Y M, Wang Z. A class of finite-time dual neural networks for solving quadratic programming problems and its k-winners-take-all application [J]. *Neural Networks*, 2013, 39(39): 27-39
- [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(2):

189-205

- [15] 孙明轩, 余轩峰, 孔颖. 终态神经计算:有限时间收敛性与相关应用[J]. 浙江工业大学学报, 2015, 43(3): 311-317
- [16] 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
- [17] 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
- [18] Yan X G, Liu M, Jin L. New zeroing neural network models for solving nonstationary Sylvester equation with verifications on mobile manipulators[J]. *IEEE Transactions on Industrial Informatics*, 2019, 15(9): 5011-5022
- [19] 孔颖, 孙明轩. Sylvester 时变矩阵方程求解的终态神经网络算法[J]. 计算机科学, 2018, 45(10): 214-218, 246
- [20] Klein C, Huang C. Review of pseudoinverse control for use with kinematically redundant manipulators[J]. *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, 1983, 13(2): 245-250
- [21] 贾庆轩, 吴楚锋, 陈钢, 等. 考虑关节限位的冗余度机器人重复性路径规划研究[J]. 高技术通讯, 2013, 23(5): 476-483
- [22] Zhang Y, Lv X, Li Z, et al. Effective neural remedy for drift phenomenon of planar three-link robot arm using quadratic performance index [J]. *Electronics Letters*, 2008, 44(6): 436-437

Conic terminal neural networks: time-variant neural-computing and repeatable motion planning of redundant manipulators

Sun Mingxuan, Wu Yuxin, Zhang Yu

(School of Information Engineering, Zhejiang University of Technology, Hangzhou 310023)

Abstract

Conic terminal neural networks are proposed based on the geometric features of the curves, which include three categories: the hyperbolic, elliptic and parabolic types. The variables involved in the networks are all bounded, which makes such networks easily implementable. The finite-time convergence is established, with derivations for the explicit expressions for their settling time functions. The hyperbolic networks are applied to solve the time-variant linear matrix equations and for the motion planning of robotic manipulators, respectively. Lyapunov and Sylvester equations are taken as examples. The finite-time convergence is achieved due to the use of the terminal network, and the resultant solutions are presented with the expected accuracy. By using a terminal criterion, the repeatable motion planning of the manipulator undertaken is realized in the presence of the initial positional shift.

Key words: terminal neural network, time-varying linear matrix equation, redundant manipulator, repeatable motion planning