



(12) 发明专利

(10) 授权公告号 CN 116117825 B

(45) 授权公告日 2023.08.08

(21) 申请号 202310350455.5

(22) 申请日 2023.04.04

(65) 同一申请的已公布的文献号

申请公布号 CN 116117825 A

(43) 申请公布日 2023.05.16

(73) 专利权人 人工智能与数字经济广东省实验室(广州)

地址 510330 广东省广州市海珠区新港东路2429号首层自编051房

专利权人 华南理工大学

(72) 发明人 张智军 何昊天 罗亚梅 张铭洋

(74) 专利代理机构 广州科粤专利商标代理有限公司 44001

专利代理师 杨可维

(51) Int.Cl.

B25J 9/16 (2006.01)

(56) 对比文件

CN 114580627 A, 2022.06.03

CN 115139301 A, 2022.10.04

CN 114564009 A, 2022.05.31

US 2021347047 A1, 2021.11.11

CN 107891424 A, 2018.04.10

CN 106945041 A, 2017.07.14

陆荣秀等. 求解时变非线性不等式的新型动态学习网络. 控制工程. 2022, 第405-411页.

审查员 崔清雨

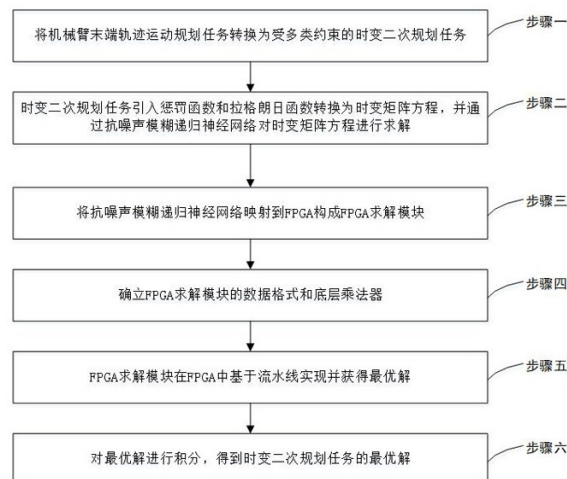
权利要求书2页 说明书8页 附图3页

(54) 发明名称

一种基于抗噪声模糊递归神经网络的FPGA实现方法

(57) 摘要

本发明公开一种基于抗噪声模糊递归神经网络的FPGA实现方法,包括以下步骤:将机械臂末端轨迹运动规划任务转换为受多类约束的时变二次规划任务;时变二次规划任务引入惩罚函数和拉格朗日函数转换为时变矩阵方程,并通过抗噪声模糊递归神经网络对时变矩阵方程进行求解;将抗噪声模糊递归神经网络映射到FPGA构成FPGA求解模块;确立FPGA求解模块的数据格式和底层乘法器;FPGA求解模块在FPGA中基于流水线实现并获得最优解;对最优解进行积分,得到时变二次规划任务的最优解。本发明的有益效果是:求解模块实时性好,计算速度快,效率更高,能够考虑多种约束条件。



1. 一种基于抗噪声模糊递归神经网络的FPGA实现方法,其特征在于,包括以下步骤:

将机械臂末端轨迹运动规划任务转换为受多类约束的时变二次规划任务;

所述时变二次规划任务引入惩罚函数和拉格朗日函数转换为时变矩阵方程,并通过抗噪声模糊递归神经网络对所述时变矩阵方程进行求解;

将所述抗噪声模糊递归神经网络映射到FPGA构成FPGA求解模块;

确立所述FPGA求解模块的数据格式和底层乘法器;

所述FPGA求解模块在FPGA中基于流水线实现并获得最优解;

对所述最优解进行积分,得到所述时变二次规划任务的最优解;

所述机械臂末端轨迹运动规划任务表示为:

$$\begin{cases} G(\theta(t)) = r(t) \\ J_M(\theta) \dot{\theta}(t) = \dot{r}(t) \end{cases} \quad (1);$$

式(1)中, G 为机械臂末端轨迹和关节角之间的非线性可微函数, J_M 为机械臂的雅克比矩阵, θ 为关节角, $\dot{\theta}$ 为关节角速度向量, t 为时间, r 为期待的跟踪路径, $\dot{r}$ 为机械臂末端速度向量;

所述将机械臂末端轨迹运动规划任务转换为受多类约束的时变二次规划任务包括:

$$\text{最小化} \|\dot{\theta}(t) + q(t)\|_2^2/2 \quad (2);$$

$$\text{受约束于 } J_M(\theta(t))\dot{\theta}(t) = \dot{r}(t) + F(r(t) - G(\theta)),$$

$$\max\{\mu(\theta^- - \theta), \dot{\theta}^-\} \leq \dot{\theta} \leq \min\{\mu(\theta^+ - \theta), \dot{\theta}^+\} \quad (3);$$

式(2)和(3)中, $q(t) = p(\theta(t) - \theta(0))$, p 为调节关节角偏移的正参数, F 为机械臂末端轨迹位置的反馈系数, μ 为强度系数, θ^- 为关节角约束下限值, θ^+ 为关节角约束上限值, $\dot{\theta}^-$ 为关节角速度向量约束。

2. 如权利要求1所述的基于抗噪声模糊递归神经网络的FPGA实现方法,其特征在于,所述抗噪声模糊递归神经网络对所述时变矩阵方程进行求解包括:

$$Q(t)y(t) = -S(t)y(t) - (\gamma + v)b^T\Phi(A(t)y(t) - g(t)) + u(t) \quad (4);$$

式(4)中, $Q(t)$ 、 $S(t)$ 、 $A(t)$ 、 $y(t)$ 、 $g(t)$ 、 $u(t)$ 为所述时变矩阵方程中与时间 t 相关的中间系数矩阵, $\dot{y}(t)$ 表示对矩阵 $y(t)$ 的导数, γ 为表示抗噪声模糊递归神经网络收敛速率的常数, v 表示模糊控制的值, b 表示可调节的参数, $\Phi(\cdot)$ 表示抗噪声模糊递归神经网络的激活函数。

3. 如权利要求2所述的基于抗噪声模糊递归神经网络的FPGA实现方法,其特征在于,所述中间系数矩阵包括:

$$A(t) = \begin{bmatrix} H & J^T(t) \\ J(t) & 0 \end{bmatrix}, y(t) = \begin{bmatrix} x(t) \\ \lambda \end{bmatrix} \quad (5);$$

$$g(t) = \begin{bmatrix} -b(t) - l\sigma \sum_{i=1}^m [a^{-\sigma[c_i(t) - J_i(t)x(t)]} \cdot K_i^T(t)] \\ b(t) \end{bmatrix} \quad (6);$$

$$Q(t) = \begin{bmatrix} H + l\sigma^2 \sum_{i=1}^m [a^{-\sigma[c_i(t) - J_i(t)x(t)]} \cdot K_i^T(t) \cdot K_i(t)] & J^T(t) \\ J(t) & 0 \end{bmatrix} \quad (7);$$

$$s(t) = \begin{bmatrix} \dot{H} + l\sigma^2 \sum_{i=1}^m [a^{-\sigma[c_i(t) - J_i(t)x(t)]} \cdot K_i^T(t) \cdot \dot{K}_i(t)] & j^T(t) \\ j(t) & 0 \end{bmatrix} \quad (8);$$

$$u(t) = \begin{bmatrix} -\dot{c}(t) - l\sigma \sum_{i=1}^m [a^{-\sigma[c_i(t) - J_i(t)x(t)]} (\dot{K}_i^T(t) - \sigma K_i^T(t) \dot{c}_i(t))] \\ \dot{b}(t) \end{bmatrix} \quad (9);$$

式(5) - (9)中,上标T表示矩阵的转置, λ 表示拉格朗日乘子, $l > 0$ 、 σ 、 a 是所述惩罚函数中可调的参数, $i = 1, \dots, m$ 表示机械臂末端执行器的运动维数, $\dot{H}$ 、 $\dot{K}(t)$ 、 $\dot{j}(t)$ 分别表示 H 、 $K(t)$ 、 $J(t)$ 对时间 t 的导数, $c_i(t)$ 、 $J_i(t)$ 、 $Z_i(t)$ 分别表示矩阵 $c(t)$ 、 $J(t)$ 、 $Z(t)$ 的第 i 项;由式(5) - (9)求得所述时变矩阵方程的最优解 y^* 。

4. 如权利要求3所述的基于抗噪声模糊递归神经网络的FPGA实现方法,其特征在于,所述FPGA求解模块的工作流程如下:

- 1) 输入三角函数的频率 ω 和时间 t ,使用坐标旋转矩阵生成系数矩阵 H, J, K, q, b, c ;
- 2) 系数矩阵 H, J, K, q, b, c 与惩罚函数系数矩阵组成中间系数矩阵 A, Q, S, g, u ;
- 3) 采用差分的方法来计算 $\dot{H}, \dot{K}, \dot{J}, \dot{c}, \dot{q}$,其中 $\dot{H}, \dot{K}, \dot{J}, \dot{c}, \dot{q}$ 表示对系数矩阵 H, K, J, c, q 的求导;
- 4) 通过6个乘法器阵列和4个加法器阵列,部署电路级的抗噪声模糊递归神经网络模块,得到一次迭代角度向量解;
- 5) 将获得的所述角度向量解输入机械臂位置生成器,获得机械臂的实际末端轨迹与期望末端轨迹之间的误差;所述误差被发送到抗噪声模糊递归神经网络模块;经过反复迭代计算,获得精确的机械臂末端轨迹。

一种基于抗噪声模糊递归神经网络的FPGA实现方法

技术领域

[0001] 本发明涉及神经网络的硬件实现技术领域,尤其涉及一种基于抗噪声模糊递归神经网络的FPGA实现方法。

背景技术

[0002] 机器人操纵器已被广泛应用于生物工程、医疗、智能制造、教育等领域。冗余机器人操纵器的运动规划已经吸引了大量的研究人员。

[0003] 冗余机器人操纵器有足够的自由度(DOF)来完成复杂的运动规划任务,就像仿人肢体一样,可以代替人类执行一些危险的任务或枯燥的重复性操作。在运动规划任务中,机器人操纵器的状态并不唯一,因此运动规划的过程可以被看作是一个优化问题。一些次要的性能指标,如关节角度漂移和关节运动极限,在原始程序中被表示为条件,传统的优化方法是基于数值法求解,在机器人操纵器的情况下,运动学建模是直接通过雅各布的伪逆来进行数值处理的。然而,数值法的缺点是迭代时间长,计算量大。利用并行计算的特点,递归神经网络(RNN)被广泛应用于一些实时运动场景,如机器人运动规划。实验证明,与数值方法相比,RNN规划机器人的收敛速度更快,控制误差更小。

[0004] 在过去的几年里,一些RNN被提出来用于解决机器人运动规划问题。在参考文献中,一个Hopfield类型的神经网络被用于解决机器人运动规划问题,其中,作者设计了一个Hopfield类型的神经网络,即双重神经网络(DNN),用于解决逆运动学问题。然而,DNN需要进行矩阵反转计算,这导致了较长的收敛时间成本。为了克服时间成本的困难,Xia提出了一个LVI原始-双神经网络(LVI-PDNN)方案,LVI-PDNN方案在机械手运动控制中的有效性已经在加速级和速度级得到了验证。随着运动轨迹复杂性的提高,机器人机械手的逆运动学应被视为一个时变的优化问题,而上述方法只能在静态情况下保持性能保证。受神经动力学方法和基于梯度的神经网络(GNN)的启发,Zhang提出了一种基于动力学的归零神经网络(ZNN),ZNN旨在使用固定参数来提高收敛率,然而,参数的扩大会导致冗余解决的问题。为了克服这个问题,Zhang提出了一个变化参数的收敛-差分神经网络(VP-CDNN)方案,并将其用于考虑干扰的机器人操纵器的速度和加速度控制。

[0005] 在实际的操纵器控制中,有许多潜在的问题,如关节运动限制、关节角度漂移和噪声容忍。其中,噪声无处不在,严重影响了运动规划方案,甚至导致失败。有几篇文献报道了如何提高耐噪声性能。Li等人考虑了在多项式噪声存在的情况下解决冗余的RNN方案。Guo等人研究了一种新的路径规划方案,该方案基于存在噪声的伪逆式公式,外部噪声被比例-积分-导数战术所抑制。然而,以前的文献都集中在外部干扰上,内部干扰在求解过程中非常重要。Zhang等人提出了一种自适应模糊RNN,用于冗余机器人操纵器控制的内部干扰抑制。

[0006] 现阶段,上述机械手控制方案都是在冯-诺依曼结构的基础上实现的,不能充分发挥神经网络的并行计算的优势。模糊系统在操纵器控制中具有以模糊输入调整响应幅度的优势,而在时变操纵器应用中的硬件实现是不存在的。还有其他致力于模糊系统的硬件实

现的先进工作,比如,Bicakci等人实现了一个基于FPGA的电磁搅拌器,以提高能源效率,其中模糊逻辑控制由ARM微控制器实现。由于ARM中的总线数量有限,处理速度也受到限制。此外,ARM和FPGA之间的额外传输时间是次要的。由于应用于模糊控制的成员函数形式是一个三角形函数,Gdaim等人提出了一个面向计算和简单实现的模糊逻辑电路。另一种实现方法是面向内存的,需要大量的LUT资源。

发明内容

[0007] 针对上述问题,本发明提出一种基于抗噪声模糊递归神经网络的FPGA实现方法,主要解决基于冯-诺依曼结构的计算芯片不能充分发挥神经网络的并行计算优势的问题。

[0008] 解决上述技术问题,本发明的技术方案如下:

[0009] 一种基于抗噪声模糊递归神经网络的FPGA实现方法,包括以下步骤:

[0010] 将机械臂末端轨迹运动规划任务转换为受多类约束的时变二次规划任务;

[0011] 所述时变二次规划任务引入惩罚函数和拉格朗日函数转换为时变矩阵方程,并通过抗噪声模糊递归神经网络对所述时变矩阵方程进行求解;

[0012] 将所述抗噪声模糊递归神经网络映射到FPGA构成FPGA求解模块;

[0013] 确立所述FPGA求解模块的数据格式和底层乘法器;

[0014] 所述FPGA求解模块在FPGA中基于流水线实现并获得最优解;

[0015] 对所述最优解进行积分,得到所述时变二次规划任务的最优解。

[0016] 在一些实施方式中,所述机械臂末端轨迹运动规划任务表示为:

$$\begin{cases} G(\theta(t)) = r(t) \\ J_M(\theta) \dot{\theta}(t) = \dot{r}(t) \end{cases} \quad (1); \text{式(1)中, } G \text{ 为机械臂末端轨迹和关节角之间的非线性可微函数,}$$

J_M 为机械臂的雅克比矩阵, θ 为关节角, $\dot{\theta}$ 为关节角速度向量, t 为时间, r 为期待的跟踪路径, $\dot{r}$ 为机械臂末端速度向量。

[0017] 在一些实施方式中,所述将机械臂末端轨迹运动规划任务转换为受多类约束的时变二次规划任务包括:最小化 $\|\dot{\theta}(t) + q(t)\|_2^2/2$ (2);

[0018] 受约束于 $J_M(\theta(t))\dot{\theta}(t) = \dot{r}(t) + F(r(t) - G(\theta))$,

$$\max\{\mu(\theta^- - \theta), \dot{\theta}^-\} \leq \dot{\theta} \leq \min\{\mu(\theta^+ - \theta), \dot{\theta}^+\} \quad (3);$$

[0019] 式(2)和(3)中, $q(t) = p(\theta(t) - \theta(0))$, p 为调节关节角偏移的正参数, F 为机械臂末端轨迹位置的反馈系数, μ 为强度系数, θ^- 为关节角约束下限值, θ^+ 为关节角约束上限值, $\dot{\theta}^-$ 为关节角速度向量约束。

[0020] 在一些实施方式中,所述抗噪声模糊递归神经网络对所述时变矩阵方程进行求解包括: $Q(t)y(t) = -S(t)y(t) - (\gamma + \nu)b^t\Phi(A(t)y(t) - g(t)) + u(t)$ (4);式(4)中, $Q(t)$ 、 $S(t)$ 、 $A(t)$ 、 $y(t)$ 、 $g(t)$ 、 $u(t)$ 为所述时变矩阵方程中与时间 t 相关的中间系数矩阵, $\dot{y}(t)$ 表示对矩阵 $y(t)$ 的导数, γ 为表示抗噪声模糊递归神经网络收敛速率的常数, ν

表示模糊控制的值, b 表示可调节的参数, $\Phi(\cdot)$ 表示抗噪声模糊递归神经网络的激活函数。

[0021] 在一些实施方式中, 所述中间系数矩阵包括:

$$A(t) = \begin{bmatrix} H & J^T(t) \\ J(t) & 0 \end{bmatrix}, y(t) = \begin{bmatrix} x(t) \\ \lambda \end{bmatrix} \quad (5);$$

$$g(t) = \begin{bmatrix} -b(t) - l\sigma \sum_{i=1}^m [a^{-\sigma[c_i(t)-J_i(t)x(t)]} \cdot K_i^T(t)] \\ b(t) \end{bmatrix} \quad (6);$$

$$Q(t) = \begin{bmatrix} H + l\sigma^2 \sum_{i=1}^m [a^{-\sigma[c_i(t)-J_i(t)x(t)]} \cdot K_i^T(t) \cdot K_i(t)] & J^T(t) \\ J(t) & 0 \end{bmatrix} \quad (7);$$

式(5)-(9)中, λ 表示

$$S(t) = \begin{bmatrix} H + l\sigma^2 \sum_{i=1}^m [a^{-\sigma[c_i(t)-J_i(t)x(t)]} \cdot K_i^T(t) \cdot K_i(t)] & J^T(t) \\ J(t) & 0 \end{bmatrix} \quad (8);$$

$$u(t) = \begin{bmatrix} -\dot{c}(t) - l\sigma \sum_{i=1}^m [a^{-\sigma[c_i(t)-J_i(t)x(t)]} (\dot{K}_i^T(t) - \sigma K_i^T(t) \dot{c}_i(t))] \\ \dot{b}(t) \end{bmatrix} \quad (9);$$

拉格朗日乘子, $l > 0$ 、 σ 、 a 是所述惩罚函数中可调的参数, $i = 1, \dots, m$ 表示机械臂末端执行器的运动维数, $\dot{H}$ 、 $\dot{K}(t)$ 、 $\dot{J}(t)$ 分别表示 H 、 $K(t)$ 、 $J(t)$ 对时间 t 的导数, $c_i(t)$ 、 $J_i(t)$ 、 $Z_i(t)$ 、 $J_i(t)$ 分别表示矩阵 $c(t)$ 、 $J(t)$ 、 $Z(t)$ 、 $J(t)$ 的第 i 项; 由式(5)-(9)求得所述时变矩阵方程的最优解 y^* 。

[0022] 在一些实施方式中, 所述FPGA求解模块的工作流程如下:

[0023] 1) 输入三角函数的频率 ω 和时间 t , 使用坐标旋转矩阵生成系数矩阵 H, J, K, q, b, c ;

[0024] 2) 系数矩阵 H, J, K, q, b, c 与惩罚函数系数矩阵组成中间系数矩阵 A, Q, S, g, u ;

[0025] 3) 采用差分的方法来计算 $\dot{H}, \dot{K}, \dot{J}, \dot{c}, \dot{q}$, 其中 $\dot{H}, \dot{K}, \dot{J}, \dot{c}, \dot{q}$ 表示对系数矩阵 H, K, J, c, q 的求导;

[0026] 4) 通过6个乘法器阵列和4个加法器阵列, 部署电路级的抗噪声模糊递归神经网络模块, 得到一次迭代角度向量解;

[0027] 5) 将获得的所述角度向量解输入机械臂位置生成器, 获得机械臂的实际末端轨迹与期望末端轨迹之间的误差; 所述误差被发送到抗噪声模糊递归神经网络模块; 经过反复迭代计算, 获得精确的机械臂末端轨迹。

[0028] 在一些实施方式中, 所述电路级的抗噪声模糊递归神经网络模块采用7*1多路复用器, 构建三阶多项式, $c_1 \eta x^3 + c_2 \eta x^2 + c_3 \eta x + c_4 \eta, \eta = 1, 2, \dots, 7$; c_1, c_2, c_3, c_4 表示常数系数参数; 所实现的电路结构中有四个端口, 分别为数据输入端口 X 、数据输出端口 Y 、使能端口 en 和时钟输入端口 clk 。

[0029] 本发明的有益效果为: 通过将机械臂末端轨迹运动规划任务转换为受多类约束的

时变二次规划任务转换为时变矩阵方程,并通过抗噪声模糊递归神经网络对时变矩阵方程进行求解,最后将抗噪声模糊递归神经网络映射到FPGA构成FPGA求解模块,实时性好,计算速度快,效率更高,能够考虑多种约束条件。

附图说明

[0030] 图1为本发明实施例公开的基于抗噪声模糊递归神经网络的FPGA实现方法的流程示意图;

[0031] 图2为本发明实施例公开的FPGA求解模块的工作流程示意图;

[0032] 图3为本发明实施例公开的流水线的结构示意图;

[0033] 图4为本发明实施例公开的电路级的抗噪声模糊递归神经网络模块的结构示意图。

具体实施方式

[0034] 为使本发明的目的、技术方案及优点更加清楚、明确,下面结合附图和具体实施方式对本发明的内容做进一步详细说明。可以理解的是,此处所描述的具体实施例仅仅用于解释本发明,而非对本发明的限定。另外还需要说明的是,为了便于描述,附图中仅示出了与本发明相关的部分而非全部内容。

[0035] 本实施例提出了一种基于抗噪声模糊递归神经网络的FPGA实现方法。与传统的基于冯-诺依曼结构的数字计算机实现相比,本发明通过FPGA实现,实时性好,能够考虑多种约束条件;

[0036] 与数值方法求解器相比,本发明属于神经网络求解器,计算速度快,效率更高;与经典的递归神经网络求解器相比,本发明采用新型的抗噪声模糊递归神经网络求解器,收敛速度更快,计算精度更高,鲁棒性更好。

[0037] 如图1所示,本方法包括以下步骤一至六:

[0038] 步骤一,将机械臂末端轨迹运动规划任务转换为受多类约束的时变二次规划任务。

[0039] 在步骤一中,将机械臂运动规划问题的标准形式描述为一个TV-QP形式,并转化为一般的TV-QP形式。机械臂末端轨迹运动规划任务表示为:
$$\begin{cases} G(\theta(t)) = r(t) \\ J_M(\theta) \dot{\theta}(t) = \dot{r}(t) \end{cases} \quad (1); \text{式(1)}$$

中, G 为机械臂末端轨迹和关节角之间的非线性可微函数, J_M 为机械臂的雅克比矩阵, θ 为关节角, $\dot{\theta}$ 为关节角速度向量, t 为时间, r 为期待的跟踪路径, $\dot{r}$ 为机械臂末端速度向量。

[0040] 将机械臂末端轨迹运动规划任务转换为受多类约束的时变二次规划任务包括:

[0041] 最小化 $\|\dot{\theta}(t) + q(t)\|_2^2/2 \quad (2);$

[0042] 受约束于 $J_M(\theta(t))\dot{\theta}(t) = \dot{r}(t) + F(r(t) - G(\theta)),$

$$\max\{\mu(\theta^- - \theta), \theta^-\} \leq \theta \leq \min\{\mu(\theta^+ - \theta), \theta^+\} \quad (3);$$

[0043] 式(2)和(3)中, $q(t) = p(\theta(t) - \theta(0))$, p 为调节关节角偏移的正参数, F 为机械臂末端轨迹位置的反馈系数, μ 为强度系数, θ^- 为关节角约束下限值, θ^+ 为关节角约束上限值, $\dot{\theta}^-$ 为关节角速度向量约束。

[0044] 式(2)和(3)可以改写为:

[0045] 最小化 $x(t)^T H x(t)/2 + q^T(t)x(t)$; 受约束于 $J(t)x(t) = b(t)$, $K(t)x(t) \leq c(t)$ 其中 H 为单位矩阵, $x(t) = \dot{\theta}(t)$, $J(t) = J_M(\theta(t))$, $b(t) = \dot{r}(t) + F(r(t) - G(\theta))$ 。

$K(t) = [E, -E]^T$, $c(t) = [\max\{\mu(\theta^- - \theta), \dot{\theta}^-\}, -\min\{\mu(\theta^+ - \theta), \dot{\theta}^+\}]^T$, 其中 E 表示单位矩阵, 符号 T 表示矩阵的转置。

[0046] 其中, $r(t) = 2\pi \cos^2(0.5\pi \cos(\pi t/20))^2 \in R^m$ 为期待的跟踪路径, $\theta(t) \in R^n$ 为由关节角组成的 n 维向量, $m = 2$ 为末端执行器运动空间的维数, $n = 3$ 为关节角空间的维数。

$\dot{r}(t) = dr(t)/dt = 0.25(-\pi^3 \sin(5\pi(-2t + \cos(\frac{\pi t}{10}))) / 10$

$+ \pi^3 \sin(0.5\pi(2t + 10 \cos(\pi t/10)) / 10) / 10$ 为跟踪速度向量, $\dot{\theta}(t) \in R^n$ 为关节角速度

向量。 $J_M(\theta(t)) = \partial g(\theta(t))/\partial \theta(t)$ 定义了机械臂的雅克比矩阵。 $\theta^- \leq \theta \leq \theta^+$ 和

$\dot{\theta}^- \leq \dot{\theta} \leq \dot{\theta}^+$ 分别表示电机关节角空间限制和运动角速度限制。 $\|\cdot\|$ 定义了向量的二范数。

F 是跟踪轨迹位置的反馈系数, $p = 0.05$ 是一个正参数, 用于调节关节角偏移

$\theta(t) - \theta(0)$ 反映。强度系数 $\mu = 0.5$ 用于将关节角 θ 约束需要转换成关节角速度 $\dot{\theta}$ 约束。

[0047] 步骤二, 时变二次规划任务引入惩罚函数和拉格朗日函数转换为时变矩阵方程, 并通过抗噪声模糊递归神经网络对时变矩阵方程进行求解。

[0048] 在步骤二中, 是获得TV-QP问题的系数矩阵, 即 H, J, K, q, b, c 。在实际的机器人规划问题中, 这些系数矩阵通常由三角函数和与时间输入 t 相关的系数组成。

[0049] 抗噪声模糊递归神经网络对时变矩阵方程进行求解包括:

$$Q(t)\dot{y}(t) = -S(t)y(t) - (\gamma + v)b^t \Phi(A(t)y(t) - g(t)) + u(t) \quad (4); \text{式(4)}$$

中, $Q(t)$ 、 $S(t)$ 、 $A(t)$ 、 $y(t)$ 、 $g(t)$ 、 $u(t)$ 为所述时变矩阵方程中与时间 t 相关的中间系数矩阵, $\dot{y}(t)$ 表示对矩阵 $y(t)$ 的导数, γ 为表示抗噪声模糊递归神经网络收敛速率的

常数, v 表示模糊控制的值, b 表示可调节的参数, $\Phi(\cdot)$ 表示抗噪声模糊递归神经网络的

激活函数。需要说明的是, Q 、 S 、 A 、 y 、 g 、 u 等是转换过程的中间系数, 本身只有数学含义, 并无具体的属性或者物理定义。

[0050] 中间系数矩阵包括:

$$A(t) = \begin{bmatrix} H & J^T(t) \\ J(t) & 0 \end{bmatrix}, y(t) = \begin{bmatrix} x(t) \\ \lambda \end{bmatrix} \quad (5);$$

$$g(t) = \begin{bmatrix} -b(t) - l\sigma \sum_{i=1}^m [a^{-\sigma[c_i(t)-J_i(t)x(t)]} \cdot K_i^T(t)] \\ b(t) \end{bmatrix} \quad (6);$$

$$Q(t) = \begin{bmatrix} H + l\sigma^2 \sum_{i=1}^m [a^{-\sigma[c_i(t)-J_i(t)x(t)]} \cdot K_i^T(t) \cdot K_i(t)] & J^T(t) \\ J(t) & 0 \end{bmatrix} \quad (7);$$

式(5) - (9)中,上

$$S(t) = \begin{bmatrix} \dot{H} + l\sigma^2 \sum_{i=1}^m [a^{-\sigma[c_i(t)-J_i(t)x(t)]} \cdot K_i^T(t) \cdot K_i(t)] & \dot{J}^T(t) \\ \dot{J}(t) & 0 \end{bmatrix} \quad (8);$$

$$u(t) = \begin{bmatrix} -\dot{c}(t) - l\sigma \sum_{i=1}^m [a^{-\sigma[c_i(t)-J_i(t)x(t)]} (\dot{K}_i^T(t) - \sigma K_i^T(t) \dot{c}_i(t))] \\ b(t) \end{bmatrix} \quad (9);$$

标 $\mathbf{T}$ 表示矩阵的转置, λ 表示拉格朗日乘子, $l > 0$ 、 σ 、 a 是惩罚函中可调的参数, $i = 1, \dots, m$ 表示机械臂末端执行器的运动维数, $\dot{H}$ 、 $\dot{K}(t)$ 、 $\dot{J}(t)$ 分别表示 H 、 $K(t)$ 、 $J(t)$ 对时间 t 的导数, $c_i(t)$ 、 $J_i(t)$ 、 $Z_i(t)$ 、 $J_i(t)$ 分别表示矩阵 $c(t)$ 、 $J(t)$ 、 $Z(t)$ 、 $J(t)$ 的第 i 项;由式(5) - (9)求得时变矩阵方程的最优解 $\mathbf{y}^*$ 。

[0051] 步骤二的主要目的是计算网络参数,坐标旋转阵列被部署来生成系数矩阵。作为三角函数的频率,固定参数 ω 存储在RAM中。与固定参数不同,时间输入 t 定义了连续的实时时间,需要被声明为网络模块的输入接口。根据抗噪声模糊递归神经网络的公式可知,中间系数矩阵,如 A, Q, S, g, u 是由第二步中提到的系数矩阵和惩罚函数系数组成。因此,系数模块 H, J, q, d 输出与惩罚函数模块的输入接口相连。在状态机的控制下,结合结尾部分的反馈,惩罚函数的结果经过多路复用器,与中间的系数模块相连。此外,由于网络中存在微分元素,需要采用差分的方法来计算 $\dot{H}, \dot{K}, \dot{J}, \dot{c}, \dot{q}$ 的值。综上所述,通过采用上述措施,通过三个因子,即步骤二中的矩阵系数因子、惩罚函数因子和微分因子,获得中间系数矩阵 A, Q, S, g, u 的值。

[0052] 步骤三,将抗噪声模糊递归神经网络映射到FPGA构成FPGA求解模块;

[0053] 在步骤三中,将获得的角度向量解空间输入机械臂位置生成器。机械臂位置生成器的作用是获得机械臂的实际末端执行器位置与期望轨迹之间的误差。然后,该误差被发送到模糊控制系统。网络计算部分的反馈被定义为 $\mathbf{Y}$,模糊控制系统的输出反馈被定义为 $\mathbf{v}$ 。最后,经过几轮迭代计算,网络计算部分反复工作,原始问题机械臂运动规划问题得到解决。

[0054] 步骤四,确立FPGA求解模块的数据格式和底层乘法器;

[0055] 步骤五,FPGA求解模块在FPGA中基于流水线实现并获得最优解;

[0056] FPGA求解模块的工作流程如图2所示,主要包括以下步骤1) - 5):

[0057] 1) 输入三角函数的频率 ω 和时间 t ,使用坐标旋转矩阵生成系数矩阵

H, J, K, q, b, c ;

[0058] 2) 系数矩阵 H, J, K, q, b, c 与惩罚函数系数矩阵组成中间系数矩阵 A, Q, S, g, u ;

[0059] 3) 采用差分的方法来计算 $\dot{H}, \dot{K}, \dot{J}, \dot{c}, \dot{q}$, 其中 $\dot{H}, \dot{K}, \dot{J}, \dot{c}, \dot{q}$ 表示对系数矩阵 H, K, J, c, q 的求导;

[0060] 4) 通过6个乘法器阵列和4个加法器阵列, 部署电路级的抗噪声模糊递归神经网络模块, 得到一次迭代角度向量解;

[0061] 5) 将获得的所述角度向量解输入机械臂位置生成器, 获得机械臂的实际末端轨迹与期望末端轨迹之间的误差; 所述误差被发送到抗噪声模糊递归神经网络模块; 经过反复迭代计算, 获得精确的机械臂末端轨迹。

[0062] 可选的, 通过6个乘法器阵列和4个加法器阵列, 部署了电路级的模糊系统模型, 得到了一次迭代角度向量解。考虑到高频率下的易变性, 在各模块中增加了寄存器缓冲区, 并尽可能地消除毛刺。

[0063] 在步骤五中, 如图3所示, 本实施例的流水线被划分为九个状态。首先, 当评估板上电时, 寄存器被重置并给出一个初始值。在这一点上, 状态机的下一个状态被设置为状态1。然后, 时钟被翻转, 状态机的当前状态被设置为状态1。坐标旋转阵列在 ST 和 P_1 中都有使用。 ST 定义了静态参数计算模块, 该模块在整个计算过程中只工作一次。与 ST 模块不同, P_1 定义了动态参数计算模块, 它需要在每次迭代中工作一次。 K 模块用于生成不含三角函数的系数。第三, 当当前状态被设置为状态2时, 原始系数模块开始工作, 下一个状态被设置为状态3。接下来, 时钟被翻转, 动态参数计算模块 P_2 开始工作。与 P_1 不同, P_2 的结果与状态2中产生的系数有关。 K 模块被部署来生成与 P_2 无关的系数。状态4被设定为生成中间系数 Q, S, g 和 u 。在状态5中, inv 模块被设定为计算 Q^{-1} , 即中间系数矩阵 Q 的逆。更重要的是, $nn1$ 被用来生成求逆前的网络参数 $h = -Sy - \gamma b^t \Psi(Ay - g) + u$ 。当FSM的状态变成状态6时, $nn1$ 开始工作, $Q^{-1}h$ 的值被得到。该值分别经过累加器和操纵器位置发生器后, 即模块 i 和 P_n , 得到一次迭代的结果和实际位置与期望位置之间的误差。最后, 误差经过模块 FZ , 即模糊控制系统, 并反馈到存储在RAM中的初始参数。然后, 第二次迭代开始工作。

[0064] 电路级的抗噪声模糊递归神经网络模块采用7*1多路复用器, 构建三阶多项式, $c_{1\eta}x^3 + c_{2\eta}x^2 + c_{3\eta}x + c_{4\eta}, \eta = 1, 2, \dots, 7$; c_1, c_2, c_3, c_4 表示常数系数参数; 所实现的电路结构中有四个端口, 分别为数据输入端口 X 、数据输出端口 Y 、使能端口 en 和时钟输入端口 clk 。

[0065] 如图4所示, 本实施例的电路级设计分为三个部分, 如逻辑组件、计算组件和存储组件。逻辑组件指的是流水线的子模块的状态机。首先, 将配置好的参数导入到逻辑组件

中。操作数对被依次存储到相应的队列中。然后,多路数据选择器遍历每个队列的输出,操作数对被转换为乘法单元,即计算组件。计算结果被存储到双端口RAM中。最后,在两个时钟周期后,结果被转换到逻辑组件,并构成新的操作数对。显然,上述的电路级设计与跨时钟域的数据处理有关。

[0066] 本实施例基于跨时钟域实现的网络的详细涉及四个时钟驱动,如 $CLK1$ 、 $CLK2$ 、 $CLK3$ 和 $CLK4$ 。 $CLK1 = 150Mhz$ 是流水线的主要驱动。 $CLK1$ 连接到全局路由资源,以实现信号的低偏斜分布。由锁相环产生的 $CLK2 = 300Mhz$ 与 $CLK1$ 相比有相位偏差。 $CLK2$ 于驱动子模块内部的复用器,目的是将当前状态的操作数按顺序写入队列中。为了保证数据传输的有效性,每个子模块都对应一个特定的队列,这与CPU上的实现不同。在实际的操纵器控制中,只要单次计算时间,即本文提出的协处理器的324个时钟周期,小于基本的迭代时间,即0.00005s。因此,在保证实时控制的情况下,应该说计算资源是有限的。根据经验,计算组件由20个乘法器组成。这个数字是由当前单一状态下的最大乘法数决定的。 $CLK3 = 350Mhz$ 是乘法器的最大时钟驱动。 $CLK4 = 200Mhz$ 用于控制双端口RAM的写入和读取。

[0067] 步骤六,对最优解进行积分,得到时变二次规划任务的最优解。

[0068] 上述实施例只是为了说明本发明的技术构思及特点,其目的是在于让本领域内的普通技术人员能够了解本发明的内容并据以实施,并不能以此限制本发明的保护范围。凡是根据本发明内容的实质所做出的等效的变化或修饰,都应涵盖在本发明的保护范围内。

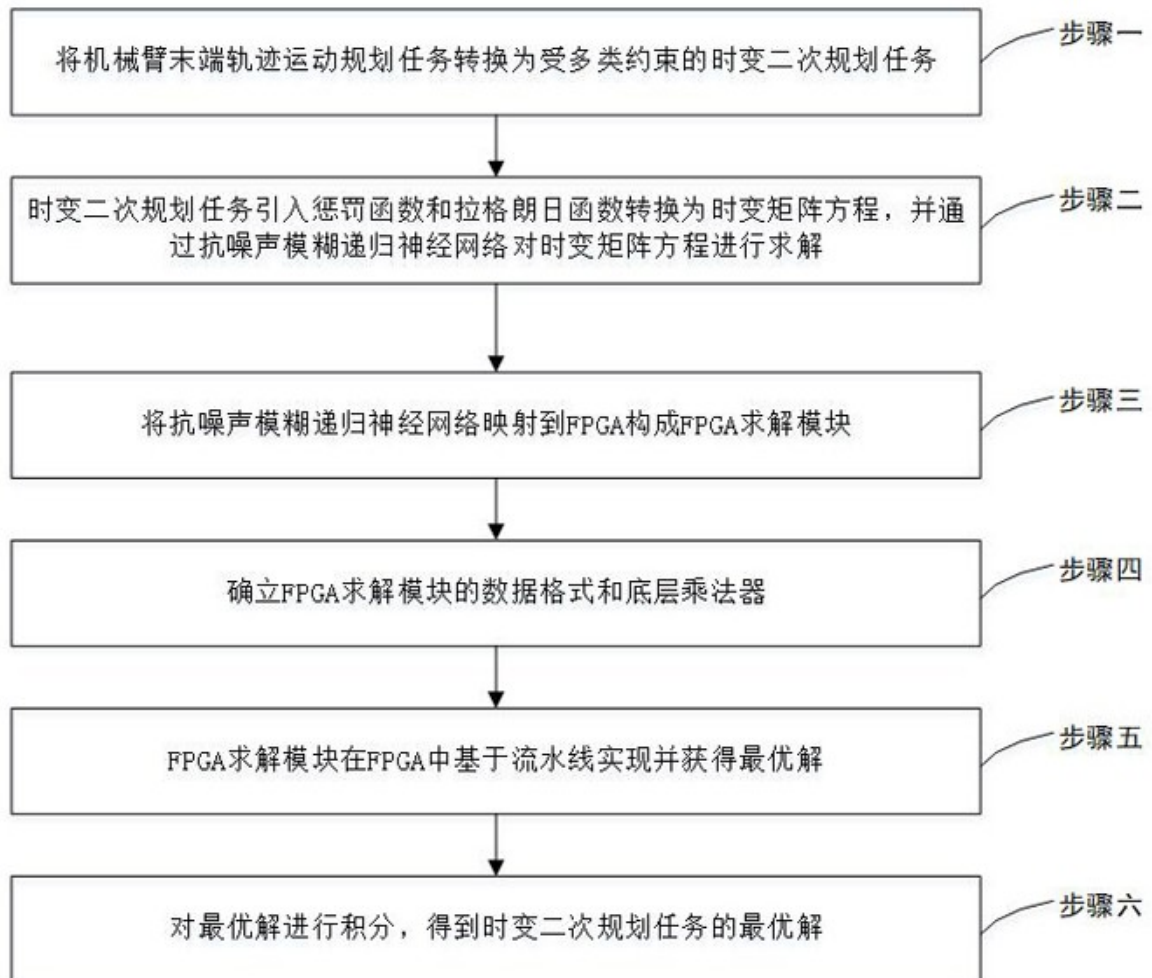


图 1

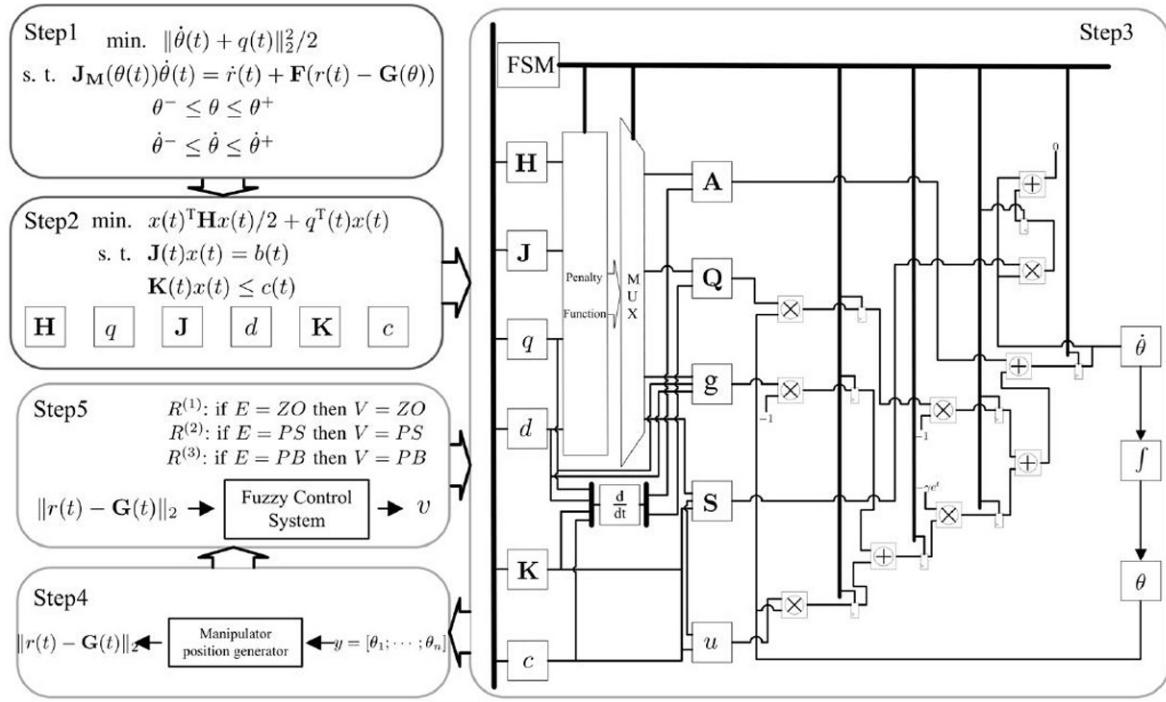


图 2

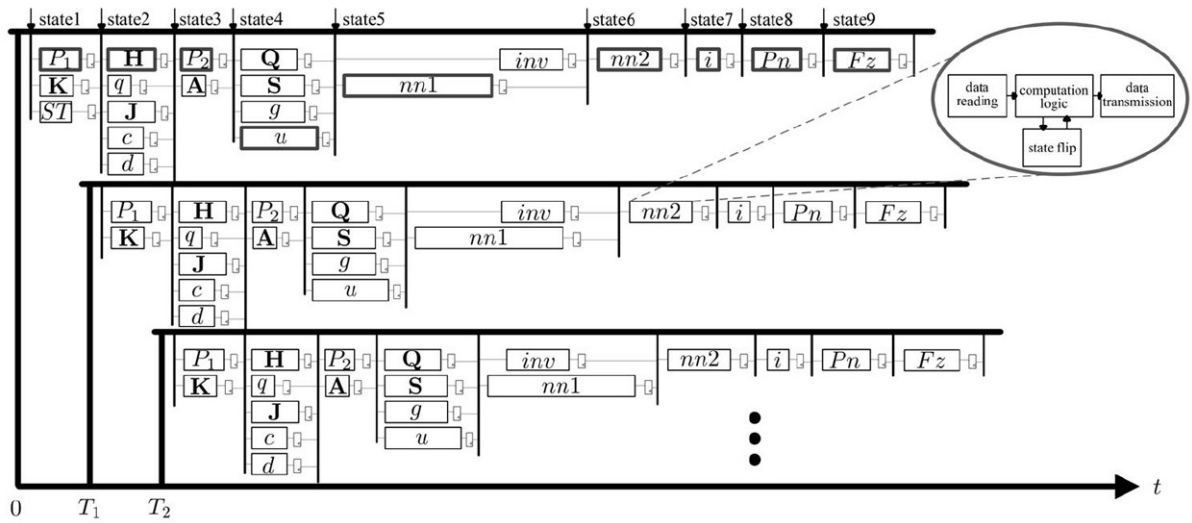


图 3

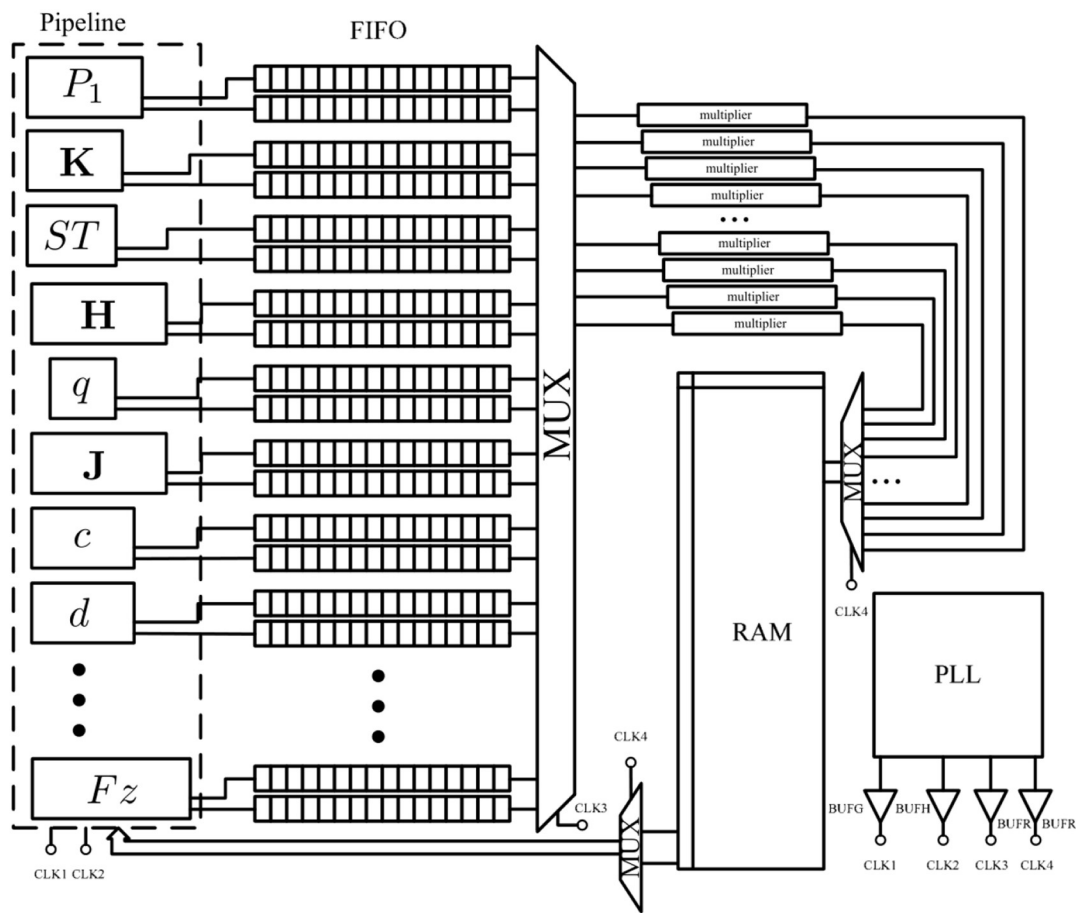


图 4