

面向FPGA便捷部署的智能模型预测控制

李星辰^{1,2}, 赵斐然^{1,2}, 孟庆辉³, 游科友^{1,2†}

(1. 清华大学自动化系, 北京 100084; 2. 北京信息科学技术国家研究中心, 北京 100084; 3. 潍柴动力股份有限公司, 山东 潍坊 261061)

摘要: 现场可编程逻辑门阵列(FPGA)具有可编程、易并行化的独特优势, 是实现一体化感知、决策、控制最具前景的人工智能芯片之一, 但其硬件描述语言(HDL)不易掌握. 本文提出了一种基于神经网络的智能MPC及其FPGA便捷部署方法, 使用高层次综合(HLS)生成HDL代码, 并通过MATLAB-Modelsim联合仿真验证代码功能, 可克服人工编写HDL代码的困难, 提高控制算法的部署效率. 该方法利用了深度神经网络的结构特点和FPGA的并行计算优势, 离线训练神经网络在线仅需硬件化正向传播, 在低资源占用的同时具有严格计算时间保证. 将所提方法分别应用于高速、高维控制系统中, FPGA在环测试验证了其有效性.

关键词: FPGA; 模型预测控制; 神经网络

引用格式: 李星辰, 赵斐然, 孟庆辉, 等. 面向FPGA便捷部署的智能模型预测控制. 控制理论与应用, 2023, 40(9): 1519 – 1528

DOI: 10.7641/CTA.2022.20336

Intelligent model predictive control with efficient FPGA implementation

LI Xing-chen^{1,2}, ZHAO Fei-ran^{1,2}, MENG Qing-hui³, YOU Ke-you^{1,2†}

(1. Department of Automation, Tsinghua University, Beijing 100084, China;

2. Beijing National Research Center for Information Science and Technology, Beijing 100084, China;

3. Weichai Power Co., Ltd., Weifang Shandong 261061, China)

Abstract: The field programmable gate array (FPGA) is one of the most promising artificial intelligence chips with unique advantages in programmability and parallelization to realize integrated perception, decision and control algorithms. However, the hardware description language (HDL) of FPGA is hard to manually write. This paper presents a convenient deployment method of intelligent MPC and FPGA based on the neural networks to overcome the difficulty of manually writing HDL code and improve the deployment efficiency of control algorithm, which generates the HDL code by high-level synthesis (HLS) and verifies the function of HDL code by the MATLAB-Modelsim co-simulation. This method takes advantage of the structural characteristics of deep neural networks and the parallel computing advantages of FPGA. A neural network is trained offline, and only the forward propagation on FPGA is required online, so a strict calculation time guarantee with low resource occupation is realized. The proposed method is applied to high-speed and high-dimensional control systems. The effectiveness of the method is verified by the FPGA-in-the-loop experiment.

Key words: FPGA; model predictive control; neural network

Citation: LI Xingchen, ZHAO Feiran, MENG Qinghui, et al. Intelligent model predictive control with efficient FPGA implementation. *Control Theory & Applications*, 2023, 40(9): 1519 – 1528

1 引言

现场可编程逻辑门阵列(field programmable gate array, FPGA)是一种可编程逻辑器件, 可根据需求改变芯片内存储单元、运算单元等电路单元的连接, 实现定制化计算电路^[1]. 与CPU, DSP等通用串行运算器件相比, 易于并行化计算的FPGA可实现更大数据吞吐量、更低的延迟. 与专用集成电路相比, FPGA具有更强的灵活性, 为定制化算法快速设计提供了硬件

基础^[2]. FPGA以其独特优势, 可融合计算机视觉^[3]、轨迹规划^[4]、底层控制算法^[5], 实现一体化感知、决策、控制的智能终端, 在无线通信^[6]、自动驾驶^[7]等领域具有广泛应用.

但FPGA使用较难掌握的硬件描述语言^[8](hardware description language, HDL)编程, 这限制了主要熟悉MATLAB, C++等高级语言的技术人员在FPGA上快速部署控制算法. 近年来, 直接通过高级语言代

收稿日期: 2022-05-01; 录用日期: 2022-11-17.

† 通信作者. E-mail: youky@tsinghua.edu.cn; Tel.: +86 10-62782721.

本文责任编辑: 龙离军.

科技创新2030-“新一代人工智能”重大项目(2022ZD0116700), 国家自然科学基金重点项目(62033006), 清华大学国强研究院项目资助.

Supported by the National Key R&D Program of China (2022ZD0116700), the National Natural Science Foundation of China (62033006) and the Grant from the Guoqiang Institute, Tsinghua University.

码生成HDL代码,即高层次综合(high level synthesis, HLS)备受关注^[9].使用HLS不仅能极大缩短开发周期,且其代码的执行效率可超越人工^[10].HDL代码的正确性是HLS的关键,但HLS生成的代码不具有可读性,无法人工验证其正确性,所以如何闭环测试生成的代码是FPGA控制算法的挑战之一.

此外,先进控制算法往往需要在线实时求解优化问题^[11],复杂的迭代求解方式并不能够完全发挥FPGA的并行计算优势.模型预测控制(model predictive control, MPC)随着采样时刻的推移进行滚动预测和优化,进而计算在性能指标下的最优控制律^[12],但这需要较大的计算成本^[13].使用FPGA可加速MPC优化问题的迭代求解,如在FPGA上定制化实现原对偶内点法^[14]、对偶积极集法^[15]、对数障碍内点法^[16].但在线迭代求解算法难以充分发挥FPGA的并行计算优势,且无法保证迭代次数,占用资源高,如文献[17]在控制变量为1维、控制时域为3的情况下,需要96个DSP单元和20k以上的查找表(lookup table, LUT)单元.

MPC控制律的求解本质是找到状态反馈和最优控制输入的映射关系,可使用神经网络^[18-21]或基函数组合^[22]高效求解MPC控制律.Karg和Lucia^[20]的研究表明,神经网络拟合显式MPC控制律将极大减少所需参数量.文献[23]使用神经网络求得MPC优化问题的“好初始解”,降低了平均迭代次数,但不能保证最坏情况下的加速效果.此外,现有研究^[18-23]未考虑被控对象的时变特性,不能应用于带有时变参数的系统.

本文利用了神经网络的结构特点和FPGA的并行计算优势,提出基于神经网络的智能MPC及其FPGA快速实现方法,具有严格求解时间保证.克服人工编写HDL代码开发周期长、开发难度大的困难,基于HLS将神经网络的高级语言代码转换为HDL代码,并通过MATLAB-Modelsim联合仿真验证HDL代码的正确性,实现了便捷高效的FPGA部署.

首先,将时变系统的预测控制问题描述为具有线性约束的二次规划问题.然后,设计了便于并行部署的快速前向传播深度神经网络,通过离散采样并反复求解MPC优化问题得到数据集,并离线学习数据近似控制问题的最优解.同时,设计了一种PI-MPC串级控制方法,以应对模型参数误差.以此智能MPC框架为基础,提出了基于HLS的高效HDL生成方法,适用于MATLAB的HDL软件级联合测试流程,及FPGA快速部署、测试验证方案.最后,将所提方法应用于控制频率高达10 kHz的永磁同步电机电流环以及四旋翼无人机轨迹跟踪,FPGA硬件在环实验验证了所提方法的快速性及鲁棒性.因此,本文的贡献主要有:1)提出了基于深度神经网络的智能PI-MPC串级控制方法及基于HLS的高效HDL生成方法和MATLAB-Mode-ls-

im联合HDL测试流程;2)在FPGA上提供了具有严格求解时间保证的智能MPC高控制频率便捷部署方案.

本文剩余部分结构如下:第2节讨论MPC及其二次规划(quadratic programming, QP)问题形式;第3节提出基于深度神经网络的智能MPC方法;第4节设计基于HLS的高效HDL生成方法、适用于MATLAB的HDL软件级联合测试流程;第5节将智能MPC应用在内置式永磁同步电机电流环(10 kHz高速控制系统)中;第6节将智能MPC应用在四旋翼无人机的轨迹跟踪控制(高维控制系统)中,并给出FPGA快速部署、测试验证方案,硬件在环(hardware in the loop, HiL)测试验证了所提方法的有效性;第7节给出结论.

2 时变系统的MPC及其QP形式

2.1 时变系统的MPC问题

考虑如下带约束的时变离散系统:

$$\begin{cases} x_{t+1} = A_t x_t + B_t u_t + e_t, \\ H_t u_t \leq h_t, G_t x_t \leq g_t, \end{cases} \quad (1)$$

其中: $x_t \in \mathbb{R}^n$ 为状态向量, $u_t \in \mathbb{R}^m$ 为控制向量, A_t, B_t 为相应维数的系统矩阵, e_t 为已知时变偏置向量, $H_t \in \mathbb{R}^{n_h \times m}, h_t \in \mathbb{R}^{n_h}, G_t \in \mathbb{R}^{n_g \times n}, g_t \in \mathbb{R}^{n_g}$ 描述了系统的状态和控制约束. 记时变参数向量 s_t 为 $A_t, B_t, e_t, H_t, h_t, G_t, g_t$ 的所有时变元素, $s_t \in \mathcal{S}, \mathcal{S}$ 为时变元素的变化范围.

在 t 时刻, MPC 利用状态反馈使状态-控制输入对 (x_t, u_t) 尽可能跟随给定状态-控制输入对 (x_t^r, u_t^r) , 即最小化如下二次型目标函数:

$$\begin{aligned} J(u_{0|t}, u_{1|t}, \dots, u_{N-1|t}) = & \|x_{N|t} - x_t^r\|_P^2 + \\ & \sum_{k=0}^{N-1} \|u_{k|t} - u_t^r\|_R^2 + \|x_{k|t} - x_t^r\|_Q^2, \end{aligned} \quad (2)$$

其中: $x_{k|t}$ 为 t 时刻通过模型预测 $t+k$ 时刻的状态向量, $u_{k|t}$ 为 $t+k$ 时刻的控制向量, N 为预测时长, $P \geq 0, Q \geq 0, R > 0, \|x\|_P^2 = x^T P x$. 由于难以获得未来的时变参数, s_t 在 t 时刻固定, 测试时不再变化. 固定的 s_t 在 N 较小时具有较好的近似效果, 因此适用于时变参数变化缓慢的系统.

求解问题(3)得到最优控制向量 $u_{0|t}^*, u_{1|t}^*, \dots, u_{N-1|t}^*$, 将 $u_{0|t}^*$ 作为系统的控制输入, 便完成了 t 时刻的模型预测控制.

$$\begin{aligned} \min_{u_{0|t}, u_{1|t}, \dots, u_{N-1|t}} & J(u_{0|t}, u_{1|t}, \dots, u_{N-1|t}), \\ \text{s.t. } & x_{k+1|t} = A_t x_{k|t} + B_t u_{k|t} + e_t, \\ & H_t u_{k|t} \leq h_t, G_t x_{k+1|t} \leq g_t, \\ & x_{0|t} = x_t, k = 0, 1, \dots, N-1. \end{aligned} \quad (3)$$

2.2 MPC的QP形式

问题(3)可转化为一个具有线性约束的二次规划问题(5). 根据式(1)得到

$$\underbrace{\begin{bmatrix} x_{1|t} \\ x_{2|t} \\ \vdots \\ x_{N|t} \end{bmatrix}}_x = \underbrace{\begin{bmatrix} B_t & 0 & \cdots & 0 \\ A_t B_t & B_t & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ A_t^{N-1} B_t & A_t^{N-2} B_t & \cdots & B_t \end{bmatrix}}_{M_t} \underbrace{\begin{bmatrix} u_{0|t} \\ u_{1|t} \\ \vdots \\ u_{N-1|t} \end{bmatrix}}_u + \underbrace{\begin{bmatrix} A_t x_t \\ A_t^2 x_t \\ \vdots \\ A_t^N x_t \end{bmatrix}}_{V_t} + \underbrace{\begin{bmatrix} I & 0 & \cdots & 0 \\ A_t & I & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ A_t^{N-1} & A_t^{N-2} & \cdots & I \end{bmatrix}}_{V_t} \underbrace{\begin{bmatrix} e_t \\ e_t \\ \vdots \\ e_t \end{bmatrix}}_{e_t} \quad (4)$$

进而, 优化问题(3)可以转化为

$$\begin{aligned} \min_u \quad & u^T Z_t u + c_t^T u, \\ \text{s.t.} \quad & \bar{H}_t u \leq \bar{h}_t, \quad \bar{G}_t (M_t u + V_t) \leq \bar{g}_t, \end{aligned} \quad (5)$$

其中:

$$\begin{aligned} Z_t &= (\bar{R} + M_t^T \bar{Q} M_t), \\ c_t^T &= 2(V_t^T \bar{Q} M_t - \bar{x}_t^T \bar{Q} M_t - \bar{u}_t^T \bar{R}), \\ \bar{H}_t &= \text{diag}\{H_t, \dots, H_t\}, \quad \bar{h}_t = [h_t; \dots; h_t], \\ \bar{G}_t &= \text{diag}\{G_t, \dots, G_t\}, \quad \bar{g}_t = [g_t; \dots; g_t], \\ \bar{x}_t &= [x_t^r; x_t^r; \dots; x_t^r], \quad \bar{u}_t = [u_t^r; u_t^r; \dots; u_t^r], \\ \bar{Q} &= \text{diag}\{Q, \dots, Q, P\}, \quad \bar{R} = \text{diag}\{R, R, \dots, R\}. \end{aligned}$$

3 基于神经网络的快速MPC

3.1 QP的快速求解

由于 $\bar{R} + M_t^T \bar{Q} M_t > 0$, 因此在式(5)中 t 时刻最优控制量 $u_{0|t}^*$ 是唯一的, 此时 $u_{0|t}^*$ 由状态 x_t , 参考状态 x_t^r , 参考控制输入 u_t^r 和时变参数 s_t 确定. 即

$$u_{0|t}^* = g_{\text{MPC}}(x_t, x_t^r, u_t^r, s_t). \quad (6)$$

求解优化问题(5)即计算函数 g_{MPC} , 但这需要较大计算成本. 直接在FPGA上部署迭代式QP求解算法^[14-17], 不仅占用资源高, 而且没有严格的求解时间保证.

本文离线采样状态反馈并反复求解MPC优化问题得到数据集, 并利用FPGA并行计算优势设计深度神经网络 $g_{\text{NN}}(x_t, x_t^r, u_t^r, s_t)$ 近似 $g_{\text{MPC}}(x_t, x_t^r, u_t^r, s_t)$, 进而离线对神经网络进行训练, 在线仅需进行神经网络正向传播即可得到控制输入, 提供了有严格求解时间保证、占用资源少的高控制频率FPGA部署方案.

3.2 神经网络的设计

为了在保证拟合效果的同时, 降低FPGA的资源占用, 需对神经网络进行结构设计. FPGA使用基本逻辑单元构建电路, 较难处理非线性函数的计算, 激活函数采用形式简单的分段线性函数, 如常用的ReLU^[24], LeakyReLU^[25]. 为了复用电路模块, 采用多层全连接网络实现神经网络, 且隐层的节点数目相同. g_{NN} 的定义为

$$\begin{aligned} g_{\text{NN}}(x_t, x_t^r, u_t^r, s_t; \theta) &= \\ (L_{\text{out}} \circ L_K \circ \sigma \circ \cdots \circ \sigma \circ L_1 \circ L_{\text{in}})([x_t; x_t^r; u_t^r; s_t]), \end{aligned} \quad (7)$$

其中: $\sigma(x) = \text{LeakyReLU}(x) = \max\{x, 0.01x\}$ 为非线性激活函数, $L_i(x) = W_i x + b_i$ 为线性函数, $\theta = \{W_1, \dots, W_K, b_1, \dots, b_K\}$ 为参数的集合. L_{in} 和 L_{out} 为缩放层 (scaling layer), 将输入进行归一化, 并将输出缩放到合适范围.

神经网络的整体结构如图1所示, 为保证神经网络输出都在可行集内, 在神经网络最后添加一投影层, 根据神经网络输入将神经网络输出投影到多面体可行集内. 由于神经网络输出仅为一个时刻的控制量 $u_{0|t}^*$, 投影的计算量小.

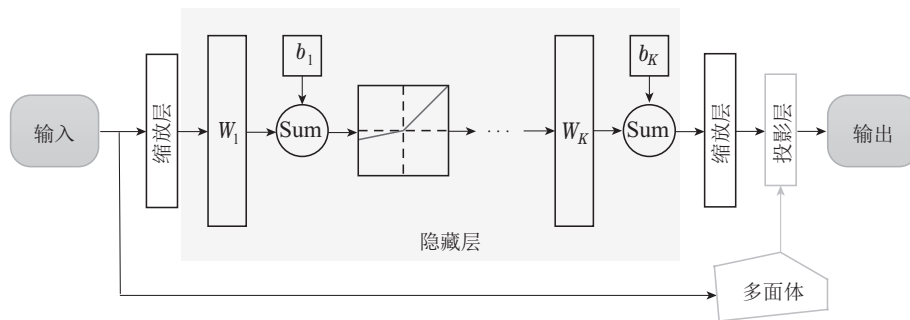


图 1 神经网络整体结构设计图
Fig. 1 Neural network structure design

3.3 神经网络的离线训练

神经网络的输入为 $[x_t; x_t^r; u_t^r; s_t]$, 即确定MPC优化问题的所需参数, 包括状态、参考状态、参考控制输

入和系统方程中的时变参数. 神经网络的输出为MPC优化问题的最优解 $u_{0|t}^*$. 神经网络训练由大量的输入输出对 $([x_t; x_t^r; u_t^r; s_t], u_{0|t}^*)$ 组成, 这可以通过反复求

解MPC优化问题得到。

求解输入输出对 $([x_t; x_t^r; u_t^r; s_t], u_{0t}^*)$ 时, 首先确定 $[x_t; x_t^r; u_t^r; s_t]$ 的变化范围(如 s_t 满足 $s_t \in \mathcal{S}$, x_t 满足 $G_t x_t \leq g_t$ 等), 样本点 z_i 从该范围中随机选取. 得到 z_i 后, 可离线求解QP问题以得到对应最优控制 $u_{0t}^* = g_{MPC}(z_i)$, 反复求解 M 次, 即得到样本数量为 M 的数据集.

神经网络训练目标为 g_{NN} 尽可能接近 g_{MPC} , 同时 g_{NN} 在可行集内, 构建如下带约束的监督学习问题:

$$\min_{\theta} \sum_{i=1}^M \|g_{NN}(z_i; \theta) - g_{MPC}(z_i)\|_2^2, \quad (8)$$

$$\text{s.t. } g_{NN}(z_i; \theta) \in \mathcal{U}(z_i), i = 1, 2, \dots, M,$$

其中: z_i 为一系列样本点, $\mathcal{U}(z_i) = \{u | \bar{H}_t u \leq \bar{h}_t, \bar{G}_t(M_t u + V_t) \leq \bar{g}_t, [x_t; x_t^r; u_t^r; s_t] = z_i\}$ 为样本点 z_i 对应的控制量的可行集.

3.4 PI-MPC串级控制

为了提高控制性能, 并应对系统建模时可能出现的模型偏差, 在神经网络MPC前添加积分环节或PI控制, 当存在静态误差时可以及时调整神经网络的目标值, 从而减少静态误差, 并提高控制鲁棒性. 实验表明, 该控制方案结合了PI控制的历史信息和MPC的未来信息, 实现了比单独PI控制或单独MPC更好的控制效果.

PI控制的输入为目标状态与状态反馈的差值, 控制框架如图2. 对于具有 n 维状态变量的系统, 对每一个维度均设置前置PI控制, 此时每个PI控制的作用为微调MPC某个维度的参考信号.

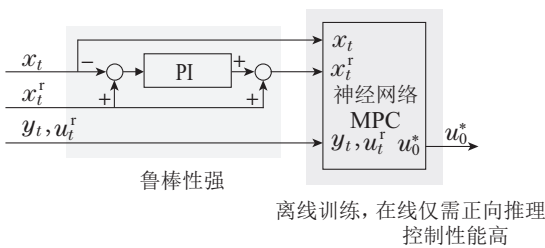


图2 控制设计原理框架

Fig. 2 Control design framework

4 MPC的FPGA快速实现与测试

4.1 基于HLS的HDL生成与Modelsim验证

为克服人工编写HDL代码开发周期长、开发难度大的困难, 本文使用Xilinx Vitis HLS工具, 将C++高级语言代码直接转换成HDL代码, 极大缩短了开发时间. MATLAB为功能强大的通用建模和控制仿真工具^[26], 本文提出了适用于MATLAB的HDL软件级联合测试流程, 被控对象和控制器原型设计均可以在MATLAB中完成, 不用使用其他语言重新编写.

为保证控制算法的硬件描述语言版本与原始版本功能相同, 本文提出一套基于MATLAB仿真模型和

Xilinx HLS的HDL生成与测试流程, 如图3. 使用MATLAB可以方便地进行控制算法设计, 但为了生成HDL代码, 需要编写与MATLAB代码功能相同的C++代码. 为验证C++代码和MATLAB代码的一致性, 在MATLAB R2020a/Simulink环境下通过Xilinx公司开发的Model Composer进行验证对比仿真.

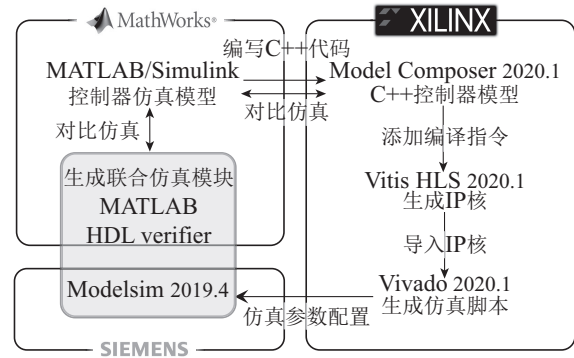


图3 HDL生成与测试流程图

Fig. 3 HDL Generation and testing flowchart

确认C++代码调试通过后, 在代码中添加Xilinx HLS相关的编译指令, 设置端口、控制协议、流水线、循环展开、存储结构等硬件属性, 通过Xilinx Vitis HLS 2020.1将带有编译指令的C++代码转换成硬件语言描述的IP核. 在MATLAB HDL Verifier中配置相关脚本和仿真属性, 并在Simulink中对比仿真HDL代码和MATLAB代码. 当两者功能一致时, IP核的功能与原始控制算法完全相同, 后续只需进行FPGA原理图的综合和部署.

该流程通过对比仿真、MATLAB-Modelsim联合仿真验证生成代码的正确性, 相比手动编写HDL代码, 极大提高了编程效率, 缩短开发时间.

4.2 FPGA测试系统实现

为测试智能MPC对应IP核在FPGA芯片上的真实运行情况, 本文使用Xilinx Zynq UltraScale+ MPSoC ZCU-102开发套件进行FPGA测试系统的搭建. ZCU 102开发板具有丰富的外设和高性能接口, 其ZYNQ芯片型号为XCZU9EG-FFVB1156-2-E.

FPGA测试系统实现方案如图4. 测试系统主要分为3个部分: PL系统、PS程序、PC上位机. PL系统为FPGA硬件电路构成, 主要运行控制器的IP核, 通过Vivado软件进行设计开发. PS为Arm核上的裸机程序, 进行与PL系统、PC上位机的通信, 控制测试流程及运行永磁同步电机(permanent magnet synchronous motor, PMSM)被控对象. PS通过Vitis软件进行设计开发, PL与PS通过AXI总线连接. PC上位机运行MATLAB程序, 使用波特率为115200 bps的COM端口向Arm核发送控制指令, Arm核上程序通过传输速率为1G bps的以太网口和以太网线向PC上位机发送运

行数据。

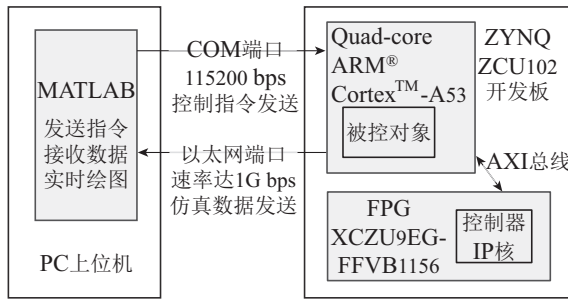


图4 FPGA测试系统实现方案
Fig. 4 FPGA test system implementation scheme

5 智能MPC在高控制频率系统中的应用

本节将智能MPC应用到PMSM的电流环控制上。由于PMSM机械时间常数远远大于电气时间常数, 通常采用如图5的双闭环控制(上标 r 表示变量的参考值), 本文的控制目标为电流环的跟踪控制, 由于其动态变化很快, 其控制频率要求高达10 kHz。

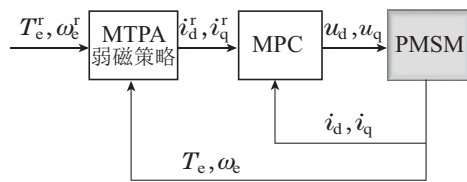


图5 PMSM的一种双闭环控制框图
Fig. 5 A double closed-loop control of PMSM

针对永磁同步电机电流环的控制, 文献[13]使用德州仪器F28335 Delfino DSP在线求解MPC优化问题, 把永磁同步电机电流环控制计算时间稳定在300 us以内。文献[27]使用具有双核2 GHz实时处理器的dSPACE, 调用开源求解器qpOASES在线求解MPC优化问题。实验表明, 在该硬件配置下最坏情况需要大概70 us的计算时间, 但无法给出严格的求解时间保证。而且文献[27]使用的硬件性能较强, 价格昂贵, 难以低成本部署以实现产品化。

算法特性和硬件环境的匹配是控制算法部署最重要的问题之一, 本文综合神经网络结构和FPGA硬件特性, 提出基于神经网络的智能模型预测方法并使用FPGA进行快速实现。由于FPGA定制硬件电路程序循环周期数、FPGA时钟频率是确定不变的, 因此具有严格求解时间保证(小于100 us, 可以实现10 kHz稳定控制频率), 且占用FPGA资源少, 可以使用价格低廉、存算资源有限的FPGA芯片部署。

5.1 永磁同步电机数学模型

内置式永磁同步电机的数学模型是非线性、时变的。在 d - q 旋转坐标系下, 其状态为 d 轴电流 i_d , q 轴电流 i_q , 转速 ω_e 。其数学模型微分方程表述如式(9)。

$$\begin{cases} \dot{i}_d = \frac{1}{L_d}(u_d - R_s i_d + \omega_e L_q i_q), \\ \dot{i}_q = \frac{1}{L_q}(u_q - R_s i_q - \omega_e L_d i_d - \omega_e \varphi_r), \\ \dot{\omega}_e = \frac{n_p}{J}(T_e - T_L), \\ T_e = \frac{3}{2}n_p(\varphi_r i_q + (L_d - L_q)i_d i_q), \end{cases} \quad (9)$$

其中: u_d, u_q 分别为 d 轴、 q 轴电压, L_d 为定子 d 轴电感, L_q 为定子 q 轴电感, $R_s = 0.0249 \Omega$ 为定子电阻, $\varphi_r = 0.02932 \text{ Wb}$ 为转子磁链, $n_p = 6$ 为极对数, $J = 1.036 \text{ Kg} \cdot \text{m}^2$ 为转动惯量, T_e 为输出转矩, T_L 为负载转矩。 L_d, L_q 的值取决于 i_d, i_q , 其函数关系是非线性的。

在本文中, 主要实现永磁同步电机电流环的控制, 电流环系统离散预测模型为

$$x_{k+1|t} = A_t x_{k|t} + B u_{k|t} + e_t, \quad (10)$$

其中: $T_s = 10^{-4} \text{ s}$,

$$x_{k|t} = \begin{bmatrix} i_d \\ i_q \end{bmatrix}_{k|t}, \quad u_{k|t} = \begin{bmatrix} u_d \\ u_q \end{bmatrix}_{k|t}, \quad e_t = \begin{bmatrix} 0 \\ -\frac{T_s \omega_e(t) \varphi_r}{L_q} \end{bmatrix},$$

$$A_t = \begin{bmatrix} 1 - \frac{T_s R_s}{L_d} & \frac{T_s \omega_e(t) L_q}{L_d} \\ -\frac{T_s \omega_e(t) L_d}{L_q} & 1 - \frac{T_s R_s}{L_q} \end{bmatrix}, \quad B = \begin{bmatrix} \frac{T_s}{L_d} & 0 \\ 0 & \frac{T_s}{L_q} \end{bmatrix}. \quad (11)$$

控制量 u_d, u_q 满足 $u_d^2 + u_q^2 \leq u_{\max}$ 。将约束条件进行线性化, 约束边界从圆变为圆内接正12边形, 线性化约束的表达式如式(12)。

$$\left[\cos \frac{(2j+1)\pi}{12}, \sin \frac{(2j+1)\pi}{12} \right] u_{k|t} \leq u_{\max} \cos \frac{\pi}{12}, \quad j = 1, \dots, 12, \quad k = 0, 1, 2. \quad (12)$$

5.2 基于神经网络的智能MPC

构造 $J(u) = \sum_{k=0}^{N-1} \|x_{k|t} - x_t^r\|_Q^2 + \|u_{k|t} - u_t^r\|_R^2$, 其中 $Q = \text{diag}\{1, 0.05\}$, $N = 3$, $R = 0.001 \times \mathbf{I}_{2 \times 2}$, 结合式(10)–(12)使用优化问题(5)的形式得到MPC。优化问题为典型的二次规划问题。

使用MALTAB R2020b自带函数quadprog进行求解, 指定方法 active-set。在CPU型号为i5-8400, 内存16 G, Windows 10系统计算机上求解10000次该优化问题, 平均单次用时1.3 ms, 不能满足 $T_s = 10^{-4} \text{ s}$ 的要求。

为了得到神经网络训练所需数据集, 首先确定状态、参考信号和时变参数的上下界, 如表1。在 $i_d, i_q, i_d^r, i_q^r, \omega_e, u_{\max}$ 的上下界范围内均匀采样变量的值, 可还原出优化问题(5)。求解该优化问题, 得到第一步对应的最优控制 $u_{0|t}^*$ 。反复随机采样变量的值, 并得到对

应最优控制量, 最终构成神经网络训练的数据集, 数据集样本数为 3×10^6 .

表 1 变量的上下界

Table 1 Upper and lower bounds of variables

	i_d/A	i_q/A	i_d^r/A	i_q^r/A	$\omega_e/(r \cdot \min^{-1})$	u_{max}/V
下界	-580	-580	-580	-580	0	230.94
上界	0	580	0	580	3000	433.01

选取数据集中5%的样本为验证集, 其余95%为训

练集. 选取隐层数量为5, 隐层宽度为50, 激活函数为LeakyReLU的神经网络. 设定 $\omega_e = 1591.54 \text{ r/min}$, $i_d = -100 \text{ A}$, $i_q = -100 \text{ A}$, $u_{max} = 346.41 \text{ V}$, 绘制 u_d 随着 i_d^r, i_q^r 变化的三维图如图6, MPC和神经网络两者非常接近.

对不同参数数量、深度、激活函数的神经网络进行训练, 最终选取隐层数量为5, 隐层宽度为50, 激活函数为LeakyReLU的神经网络, 有优越的训练效果. 其训练集均方平均损失为 6.18×10^{-5} , 验证集均方平均损失为 6.34×10^{-5} .

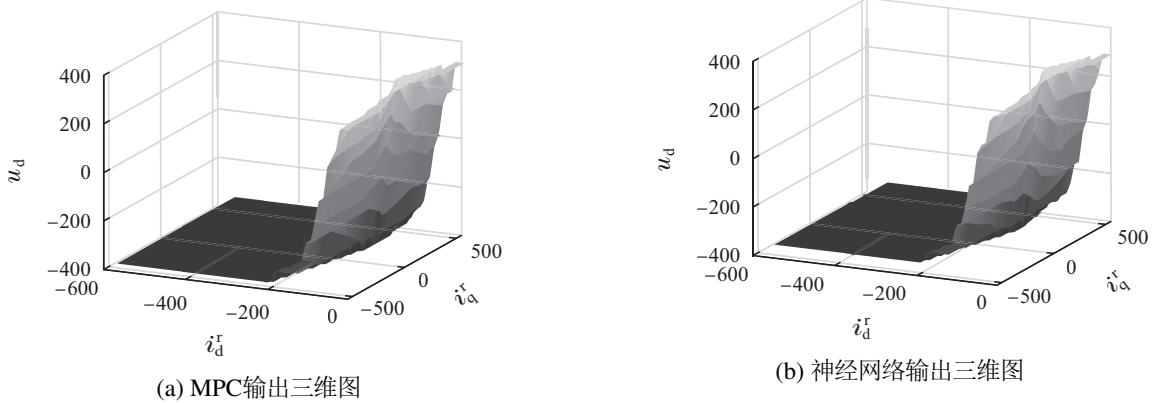


图 6 MPC和神经网络输出三维图

Fig. 6 MPC and neural network output 3D graph

5.3 HDL生成

使用不同的编译指令生成的硬件性能不同, 为保证控制频率的同时尽可能减低资源占用, 本文对HLS编译指令进行了优化设计.

运行时间和资源占用情况如表2和表3. 编号1到编号6使用了不同的HLS编译指令和计算顺序, 因此生成硬件的运行时间和资源占用不同. 编号1、编号2对程序整体流水线化. 计算神经网络矩阵运算时, 编号1, 3, 5先沿矩阵的第1维索引乘加, 编号2, 4, 6先沿矩阵的第2维索引乘加. 编号5, 6在矩阵乘向量运算的两层for循环处设置流水线.

表2中, 编号1, 2未能实现200 MHz(5 ns)的目标时钟, 只能以更低频率时钟运行, 但控制频率仍可达到50 kHz以上. 编号3, 4, 5, 6均能实现200 MHz的目标时钟, 但编号5未能实现10 kHz的控制频率. 表3中, 编号1, 2占用资源较多, 编号5, 6的BRAM和LUT使用量

均为3%以下, 数字信号处理单元(digital signal Processor, DSP)和触发器(flip-flop, FF)的使用量均为1%以下, 占用资源很少.

最终采用编号6的HLS生成方案, 该方案能够实现10 kHz的控制频率且资源占用很少, 其运行时间分析和资源占用分析示意图如图7, 其中隐层的正向传播占去86.5%的计算时间.

5.4 控制效果验证情况

FPGA控制器和原始MPC极为相近, 控制效果相当, 具有优越控制效果和高控制频率. 具体仿真设置为: $i_d^r = -213.77 \text{ A}$, $u_{max} = 346.41 \text{ V}$, 在恒定转速 $\omega_e = 900 \text{ r/min}$ 下进行测试. 在 $t = 0.3 \text{ s}$ 时, 设置 i_q^r 变为 218.92 A , 0.7 s 时 i_q^r 变为0. 并将FPGA控制器(实时测试)与原始MPC(在MATLAB仿真环境下进行测试)进行对比. i_q 跟随情况如图8.

表 2 不同优化方案对应IP核运行时间情况表

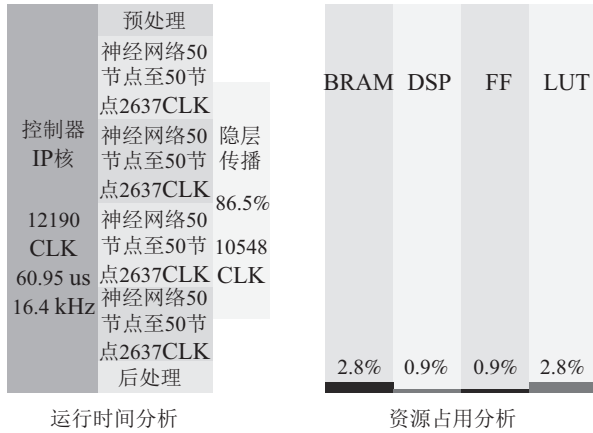
Table 2 IP core running time corresponding to different optimization schemes

编号	实现时钟/ms	运行时钟周期	运行时间/us	控制频率/kHz
1	5.037	2670	13.449	74.35
2	5.101	2415	12.318	81.18
3	4.549	3763	18.815	53.15
4	4.320	11607	58.035	17.23
5	4.264	86861	434.305	2.30
6	4.264	12190	60.950	16.41

表3 不同优化方案对应IP核资源占用情况表

Table 3 IP core resource occupancy corresponding to different optimization schemes

编号	BRAM	DSP	FF	LUT
1	14 (0.8%)	100 (4.0%)	22647 (4.1%)	109946 (40.1%)
2	14 (0.8%)	215 (8.5%)	41609 (7.6%)	103657 (37.8%)
3	225 (12.3%)	280 (11.1%)	37659 (6.9%)	34237 (12.5%)
4	223 (12.2%)	226 (9.0%)	25728 (4.7%)	22644 (8.3%)
5	51 (2.8%)	21 (0.8%)	4638 (0.8%)	7453 (2.7%)
6	51 (2.8%)	23 (0.9%)	5125 (0.9%)	7691 (2.8%)



运行时间分析

资源占用分析

图7 运行时间分析和资源占用分析示意图

Fig. 7 Run times and resource occupancy analysis

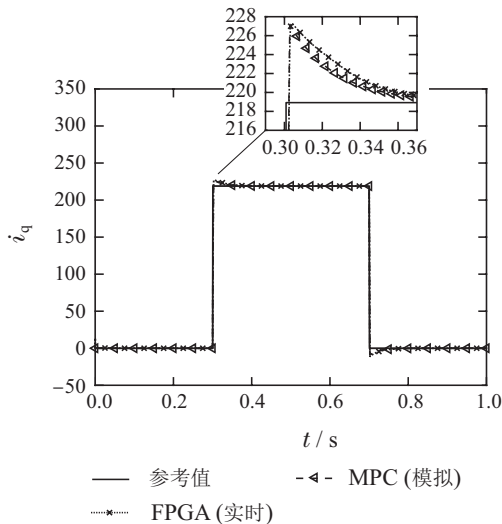


图8 q 轴电流跟随图

Fig. 8 q -axis current following diagram

FPGA控制器在各种参数误差工况下均保持了优越的控制性能. 具体仿真设置为: $i_q^r = -191.67$ A, $u_{\max} = 346.41$ V, 在恒定转速 $\omega_e = 300$ r/min 下进行测试. $t = 0.3$ s 时, 设置 i_q^r 变为 375.90 A, 0.7 s 时 i_q^r 变为 0 . 在PMSM模型的 R_s, L_d, L_q 参数添加20%的误差, 观察在有误差的PMSM模型下的控制跟随效果. i_q 跟随情况如图9.

在不同工况下, 智能MPC都能够满足最大电压约束, 如图10.

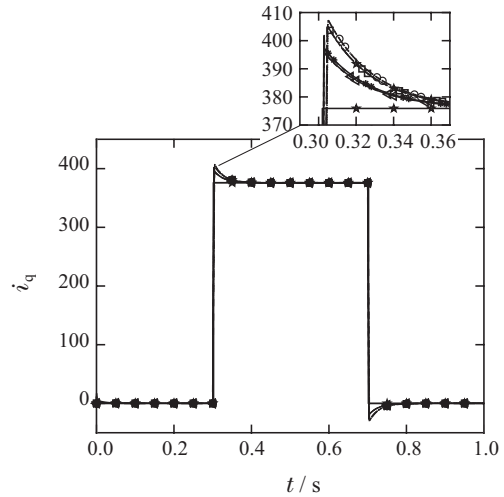


图9 具有模型参数误差时 q 轴电流跟随图

Fig. 9 q -axis current following diagram with model parameter errors

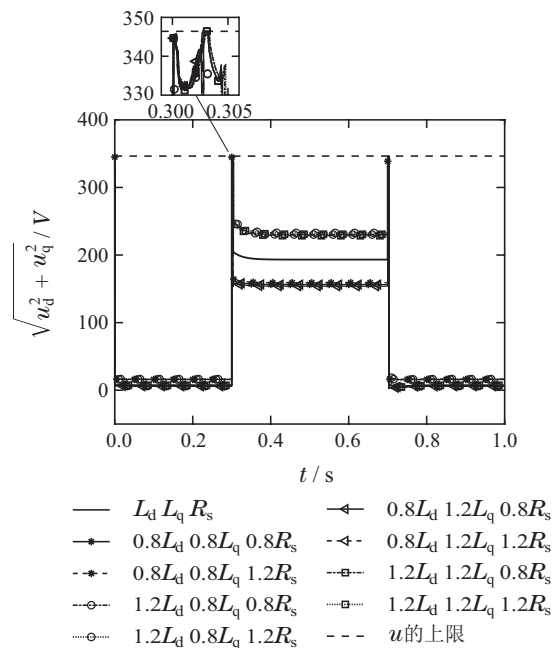


图10 具有模型参数误差时控制电压图

Fig. 10 Control input voltage diagram with model parameter errors

6 智能MPC在高维系统中的应用

本节将智能MPC应用到四旋翼无人机(unmanned aerial vehicle, UAV)的轨迹跟踪控制中. 在实际环境中, 无人机的工作空间通常具有障碍物, 因此要求四旋翼无人机能够在安全避障的情况下实现特定任务, 例如到达指定地点、以恒定速度环绕移动目标.

为验证智能MPC方法在高维、带约束系统中的有效性, 本节针对快速穿越障碍物区域的任务, 研究具有12维状态的四旋翼无人机的轨迹跟踪及避障问题.

6.1 四旋翼无人机数学模型

本文采用文献[28]中基于Newton-Euler方法给出的四旋翼无人机的数学模型

$$m\ddot{\boldsymbol{\xi}} = m\mathbf{g} + \mathbf{R}(\boldsymbol{\eta})\boldsymbol{\tau}_1 - \text{diag}\{d_x, d_y, d_z\}\dot{\boldsymbol{\xi}}, \quad (13)$$

$$\ddot{\boldsymbol{\eta}} = \mathbf{J}^{-1}(\boldsymbol{\eta})(\boldsymbol{\tau}_2 - \mathbf{C}(\boldsymbol{\eta}, \dot{\boldsymbol{\eta}})\dot{\boldsymbol{\eta}}), \quad (14)$$

其中: $\boldsymbol{\xi} = [x(t) \ y(t) \ z(t)]^T$ 为四旋翼无人机在世界坐标系中的三维空间位置, $\boldsymbol{\eta} = [\phi(t) \ \theta(t) \ \psi(t)]^T$ 为世界坐标系中的欧拉角, m 为无人机的质量, $\mathbf{g} = [0 \ 0 \ -9.81 \text{ m/s}^2]^T$ 为重力加速度, $\mathbf{R}(\boldsymbol{\eta})$ 为机身坐标系到世界坐标系的旋转矩阵, d_x, d_y, d_z 为速度阻尼系数, $\mathbf{J}^{-1}(\boldsymbol{\eta})$ 为雅可比矩阵的逆矩阵, $\mathbf{C}(\boldsymbol{\eta}, \dot{\boldsymbol{\eta}})$ 为科里奥利矩阵.

机身受到旋翼的力 $\boldsymbol{\tau}_1$ 和力矩 $\boldsymbol{\tau}_2$ 取决于4个旋翼的旋转速度

$$\boldsymbol{\tau}_1 = [0 \ 0 \ k_1 \sum_{i=1}^4 u_i(t)]^T, \quad (15)$$

$$\boldsymbol{\tau}_2 = \begin{bmatrix} k_2(-u_2(t) + u_4(t)) \\ k_2(-u_1(t) + u_3(t)) \\ k_3(-u_1(t) + u_2(t) - u_3(t) + u_4(t)) \end{bmatrix}, \quad (16)$$

其中: $u = [u_1(t) \ u_2(t) \ u_3(t) \ u_4(t)]$ 为与旋翼旋转速度相关的控制量, k_1, k_2, k_3 为常数.

令 $x = [\boldsymbol{\xi}; \boldsymbol{\eta}; \dot{\boldsymbol{\xi}}; \dot{\boldsymbol{\eta}}]$, 无人机模型为 $\dot{x} = f(x, u)$. 该模型为非线性模型, 使用显式欧拉法将该模型线性化, 并转化为LPV模型如下:

$$x_{t+1} = A_t x_t + B_t u_t. \quad (17)$$

取采样周期 $T_s = 0.1 \text{ s}$, 上式为

$$A_t = I + T_s \nabla_x f(x_t, u_t), \quad (18)$$

$$B_t = T_s \nabla_u f(x_t, u_t). \quad (19)$$

6.2 智能MPC设计与FPGA测试

给定状态参考轨迹 $\{x_0^r, x_1^r, \dots, x_T^r\}$, 在每个时刻, 轨迹跟踪的目标为最小化参考轨迹和预测轨迹的误差, 且最小化控制量与参考控制量 u^r 的误差. 目标函数为

$$\min \sum_{k=0}^{N-1} \|x_{k+1|t} - x_{k+1|t}^r\|_Q^2 + \|u_{k|t} - u^r\|_R^2, \quad (20)$$

为保证控制效果, 令 $N = 10$, 上式权重矩阵为

$$Q = \text{diag}\{10 \times \mathbf{I}_{3 \times 3}, 1 \times \mathbf{I}_{6 \times 6}, 0.1 \times \mathbf{I}_{3 \times 3}\}, \quad (21)$$

$$R = \text{diag}\{0.01, 0.01, 0.01, 0.01\}, \quad (22)$$

且控制量具有约束 $\mathbf{0}_{4 \times 1} \leq u \leq 12 \times \mathbf{1}_{4 \times 1}$.

此外, 为了避免与障碍物碰撞, 设置如图11所示的时变状态约束. 为保证无人机的安全, 距离无人机周围 $r = 0.3 \text{ m}$ 的位置不能出现障碍物, 即以无人机为中心、半径为 r 的圆与障碍物不相交. 在每个时刻, 将无人机与障碍物不相交简化为圆在某个半平面内, 该半平面由该时刻的参考位置确定. 该约束可写为 $G_t x_{N_p|t} \leq g_t$, 为时变约束. 当无人机的参考位置位于左上角时, 可行域在左边, 无人机的预测位置应该在这可行域1内. 当无人机来到下方时, 则应该在可行域2内.

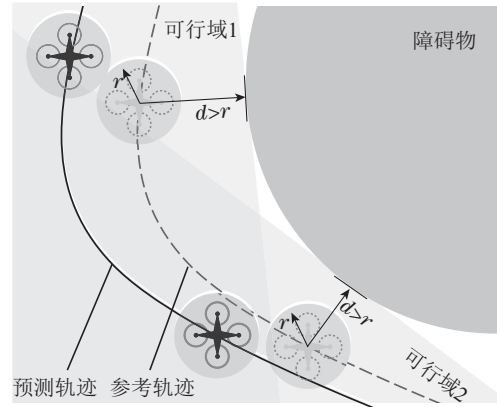


图 11 无人机避障约束示意图

Fig. 11 Diagram of obstacle avoidance constraint for UAV

为了得到神经网络训练所需数据集, 本文给定不同的随机参考轨迹, 分别使用上述MPC进行控制, 并收集状态参考轨迹 $\{x_0^r, x_1^r, \dots, x_T^r\}$, 状态轨迹 $\{x_0, x_1, \dots, x_T\}$ 和控制轨迹 $\{u_0, u_1, \dots, u_T\}$. 将参考轨迹的第1到3维抽出记为 y_k^r , 并将状态约束不等式两边的 G_t 和 g_t 的非零项也作为输入, 构造输入输出对 $([x_k; y_k^r; G_t; g_t], u_k)$, 输入为18维、输出为4维. 收集 2.8×10^6 个输出输出对, 得到训练数据集.

同样选择5%的数据集作为验证集, 其余95%为训练集. 选取隐层数量为4, 隐层宽度为80, 激活函数为LeakyReLU的神经网络使用Adam优化器进行训练. 其训练集均方损失为 3.13×10^{-4} , 验证集均方损失为 3.15×10^{-4} .

在HDL生成时, 使用第5.3节中的最优配置方案, 得到表4和表5. 控制算法在FPGA上运行仅需0.11 ms, 且占用FPGA资源少于4%.

同样地, 硬件电路程序循环周期数、FPGA时钟频率确定不变. 因此无人机智能MPC具有严格求解时间保证, 其求解频率能严格稳定在8.846 kHz, 远超采样

频率. 且该方案占用FPGA资源少, 这说明该方法应用
在更复杂的被控对象和控制任务上有很大潜力.

表 4 IP 核运行时间情况表
Table 4 IP core running time

实现时钟/ms	运行时钟周期	运行时间/ms	控制频率/kHz
3.848	22608	0.113	8.846

表 5 IP 核资源占用情况表
Table 5 IP core resource occupancy

BRAM	DSP	FF	LUT
64 (3.51%)	31 (1.23%)	7043 (1.28%)	7482 (2.73%)

6.3 控制效果验证情况

为验证算法完成快速穿越障碍物区域任务的能力, 人工指定一条参考轨迹, 并在每个时刻对控制量 u 添加服从正态分布 $\mathcal{N}(\mathbf{0}_{4 \times 1}, 0.2 \times \mathbf{I}_{4 \times 4})$ 的噪声. 将无状态约束的MPC, FPGA上的智能MPC, PID这3种控制律的控制效果进行对比. 轨迹跟踪情况如图12.

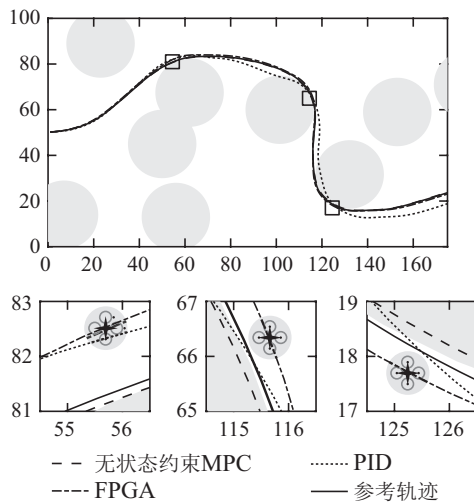


图 12 不同方法轨迹跟踪情况对比图
Fig. 12 Comparison diagram of trajectory tracking of different methods

可以看到, 添加了状态约束的FPGA控制器的能够始终保持与障碍物的安全距离, 无状态约束的MPC和PID将会撞向障碍物, 这验证了智能MPC处理约束的有效性.

在轨迹跟踪中, FPGA计算得到的控制量如图13, 可见其满足控制约束.

7 结论

本文以FPGA为实现平台, 研究了不精确时变系统MPC的FPGA快速实现问题. 为避免QP迭代求解资源占用高、求解时间不确定的问题, 使用适用于FPGA的深度神经网络求解MPC优化问题, 并考虑时变参

数. 为克服人工编写HDL代码开发难度大的困难, 本文使用HLS工具生成HDL代码, 并提出了MATLAB-Modelsim联合仿真测试流程, 以确保代码转换的一致性. 将所提方法应用于采样频率高达10 kHz的PMSM电流环控制和高维度的UAV轨迹跟踪控制, FPGA在环测试验证了有效性. 未来可进一步研究神经网络参数量与MPC问题复杂性的理论关系, 为神经网络结构设计提供理论参考. 此外, 为提高鲁棒性, 可探讨Tube-based MPC等方法的FPGA快速部署方案.

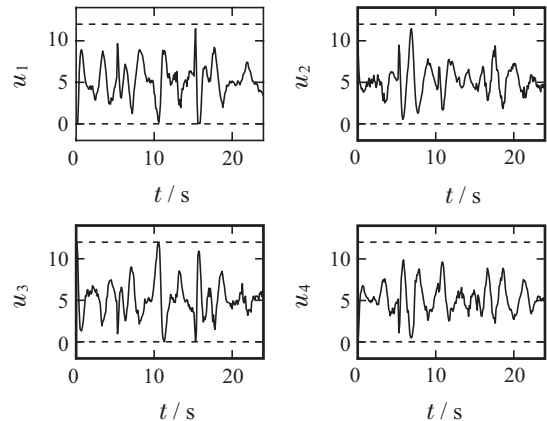


图 13 轨迹跟踪控制量变化图
Fig. 13 Control input diagram of trajectory tracking

参考文献:

- [1] MONMASSON E, CIRSTEA M N. FPGA design methodology for industrial control systems—A review. *IEEE Transactions on Industrial Electronics*, 2007, 54(4): 1824 – 1842.
- [2] ASANO S, MARUYAMA T, YAMAGUCHI Y. Performance comparison of FPGA, GPU and CPU in image processing. *International Conference on Field Programmable Logic and Applications*. Prague: IEEE, 2009: 126 – 131.
- [3] GUO K, SUI L, QIU J, et al. Angel-eye: A complete design flow for mapping cnn onto embedded fpga. *IEEE Transactions on Computer-aided Design of Integrated Circuits and Systems*, 2017, 37(1): 35 – 47.
- [4] GIRAU B, BOUMAZA A. Embedded harmonic control for dynamic trajectory planning on FPGA. *The IASTED International Conference on Artificial Intelligence and Applications*. Austria: ACTA Press, 2007, 1 – 7.
- [5] AGUIRRE M P, CALVINO L, VALLA M I. Multilevel current-source inverter with FPGA control. *IEEE Transactions on Industrial Electronics*, 2012, 60(1): 3 – 10.
- [6] BENGHERBIA B, ZMIRLI M O, TOUBAL A, et al. FPGA-based wireless sensor nodes for vibration monitoring system and fault diagnosis. *Measurement*, 2017, 101: 81 – 92.
- [7] DU S, HUANG T, HOU J, et al. FPGA based acceleration of game theory algorithm in edge computing for autonomous driving. *Journal of Systems Architecture*, 2019, 93: 33 – 39.
- [8] CASTILLO J, HUERTA P, MARTINEZ J I. An open-source tool for SystemC to Verilog automatic translation. *Latin American Applied Research*, 2007, 37(1): 53 – 58.
- [9] GAJSKI D D, DUTT N D, WU A C, et al. *High-Level Synthesis: Introduction to Chip and System Design*. Berlin: Springer Science & Business Media, 2012.

- [10] CONG J, LIU B, NEUENDORFFER S, et al. High-level synthesis for FPGAs: From prototyping to deployment. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2011, 30(4): 473 – 491.
- [11] LEWIS F L, VRABIE D, SYRMOS V L. *Optimal Control*. Hoboken: John Wiley Sons, 2012.
- [12] CAMACHO E F, ALBA C B. *Model Predictive Control*. Berlin: Springer Science & Business Media, 2013.
- [13] CIMINI G, BERNARDINI D, BEMPORAD A, et al. Online model predictive torque control for permanent magnet synchronous motors. *International Conference on Industrial Technology*. New York: IEEE, 2015: 2308 – 2313.
- [14] HARTLEY E, JEREZ J, SUARDI A, et al. Predictive control using an FPGA with application to aircraft control. *IEEE Transactions on Control Systems Technology*, 2013, 22(3): 1006 – 1017.
- [15] WILLS A G, KNAGGE G, NINNESS B. Fast linear model predictive control via custom integrated circuit architecture. *IEEE Transactions on Control Systems Technology*, 2011, 20(1): 59 – 71.
- [16] MILLS A, WILLS A G, WELLER S R, et al. Implementation of linear model predictive control using a field-programmable gate array. *IET Control Theory & Applications*, 2012, 6(8): 1042 – 1054.
- [17] BASTERRETXEA K, BENKRID K. Embedded high-speed model predictive controller on a FPGA. *NASA/ESA Conference on Adaptive Hardware and Systems*. San Diego: IEEE, 2011: 327 – 335.
- [18] PARISINI T, ZOPPOLI R. A receding-horizon regulator for nonlinear systems and a neural approximation. *Automatica*, 1995, 31(10): 1443 – 1451.
- [19] HERTNECK M, KÖHLER J, TRIMPE S, et al. Learning an approximate model predictive controller with guarantees. *IEEE Control Systems Letters*, 2018, 2(3): 543 – 548.
- [20] KARG B, LUCIA S. Efficient representation and approximation of model predictive control laws via deep learning. *IEEE Transactions on Cybernetics*, 2020, 50(9): 3866 – 3878.
- [21] CHEN S, SAULNIER K, ATANASOV N, et al. Approximating explicit model predictive control using constrained neural networks. *Annual American Control Conference*. Milwaukee: IEEE, 2018: 1520 – 1527.
- [22] BEMPORAD A, OLIVERI A, POGGI T, et al. Ultrafast stabilizing model predictive control via canonical piecewise affine approximations. *IEEE Transactions on Automatic Control*, 2011, 56(12): 2883 – 2897.
- [23] KLAUCO M, KALUZ M, KVASNICA M. Machine learning-based warm starting of active set methods in embedded model predictive control. *Engineering Applications of Artificial Intelligence*, 2019, 77: 1 – 8.
- [24] NAIR V, HINTON G E. Rectified linear units improve restricted boltzmann machines. *The 27th International Conference on Machine Learning*. Madison: Omnipress, 2010: 807 – 814.
- [25] MAAS A L, HANNUN A Y, NG A Y. Rectifier nonlinearities improve neural network acoustic models. *The 30th International Conference on Machine Learning*. Atlanta: Omnipress, 2013(30): 1 – 6.
- [26] WANG L. *Model Predictive Control System Design and Implementation Using MATLAB*. Berlin: Springer Science & Business Media, 2009.
- [27] FAVATO A, CARLET P, TOSO F, et al. Integral model predictive current control for synchronous motor drives. *IEEE Transactions on Power Electronics*, 2021, 36(11): 13293 – 13303.
- [28] LUUKKONEN T. Modelling and control of quadcopter. *Independent Research Project in Applied Mathematics*. Espoo, Finland: Aalto University, 2011(22): <https://sal.aalto.fi/en/publications/student-projects/>.

作者简介:

李星辰 博士研究生, 目前研究方向为智能控制及其应用, E-mail:

lixc21@mails.tsinghua.edu.cn;

赵斐然 博士研究生, 目前研究方向为强化学习、数据驱动的控制及其应用, E-mail: zhaofr18@mails.tsinghua.edu.cn;

孟庆辉 硕士, 目前研究方向为电机控制与应用, E-mail: mengqinghui@weichai.com;

游科友 博士, 长聘副教授, 国家优青, 博士生导师, 目前研究方向为网络控制系统、分布式优化、学习与应用, E-mail: youky@tsinghua.edu.cn.