

DOI:10.19651/j.cnki.emt.2417532

# 面向 RTC 的微秒级采样实时控制系统设计与实现<sup>\*</sup>

周杏伟<sup>1</sup> 胡文山<sup>1</sup> 刘国平<sup>2</sup> 雷忠诚<sup>1</sup>

(1. 武汉大学电气与自动化学院 武汉 430072; 2. 南方科技大学系统设计与智能制造学院 深圳 518055)

**摘要:** 随着现代控制系统对高实时性和高精度要求的不断提升,微秒级采样的实时控制技术已成为研究热点。针对现有技术高度依赖国外平台的问题,本研究基于 STM32 微控制器,设计并实现了一种具有自主知识产权的微秒级采样实时控制系统,重点解决了控制任务的实时性、通信任务的高效性以及自动代码生成的复杂性问题。系统采用 FreeRTOS 实时操作系统进行任务调度,通过硬件定时器实现控制任务的高优先级执行,同时引入 LwIP 协议以实现低延迟网络通信,支持实验数据的高效上传与指令接收。此外,本研究提出了一种基于图形化框图的控制算法解析与自动代码生成方法,通过服务器对用户前端搭建的控制框图进行解析,并生成适配 STM32 硬件的高效控制代码。以永磁同步电机的场向量控制为案例的实验验证表明,所设计系统在微秒级采样下具备优异的实时性和控制性能,有效突破了国外技术封锁对相关领域发展的限制,显著提升了自主控制系统的研发能力。该成果为解决关键技术“卡脖子”问题提供了一种可行的技术方案,并展现出广泛的应用前景与推广价值。

**关键词:** STM32; 微秒级控制; 实时操作系统; 自动代码生成; 永磁同步电机; 场向量控制

**中图分类号:** TP273; TN05   **文献标识码:** A   **国家标准学科分类代码:** 510.8020

## Design and implementation of a microsecond-level sampling real-time control system for RTC applications

Zhou Xingwei<sup>1</sup> Hu Wenshan<sup>1</sup> Liu Guoping<sup>2</sup> Lei Zhongcheng<sup>1</sup>

(1. School of Electrical Engineering and Automation, Wuhan University, Wuhan 430072, China; 2. School of Design and Intelligent Manufacturing, Southern University of Science and Technology, Shenzhen 518055, China)

**Abstract:** As modern control systems demand increasingly higher real-time performance and precision, microsecond-level sampling real-time control technology has become a research hotspot. To address the reliance on foreign platforms in existing solutions, this paper presents a microsecond-level sampling real-time control system based on the STM32 microcontroller, designed with independent intellectual property rights. The system addresses key challenges, including the real-time performance of control tasks, the efficiency of communication tasks, and the complexity of automatic code generation. It employs the FreeRTOS real-time operating system for task scheduling, hardware timers for high-priority execution of control tasks, and the LwIP protocol for low-latency network communication, enabling efficient data uploading and command reception. Furthermore, the paper proposes a graphical block diagram-based approach for control algorithm parsing and automatic code generation, where a server parses user-designed block diagrams and generates optimized control code tailored for STM32 hardware. Experimental validation using the field-oriented control of a permanent magnet synchronous motor demonstrates the system's excellent real-time performance and control capability under microsecond-level sampling. This system effectively overcomes the constraints imposed by foreign technological barriers, significantly enhancing the development capabilities of domestic control systems. The proposed solution offers a viable approach to addressing critical “bottleneck” issues in key technologies and holds great potential for wide-ranging applications and scalability.

**Keywords:** STM32; microsecond-level control; real-time operating system; automatic code generation; permanent magnet synchronous motor; field-oriented control

## 0 引言

近年来,线上教学的理念与方式在高校教育中得到了

广泛重视,尤其是在实验课程教学中,线上实验平台的构建已成为提升工程教育质量的重要方向。中华人民共和国教育部明确提出要加强线上教育教学,推进优质教育资源

收稿日期:2024-12-03

\*基金项目:国家自然科学基金(62073247,62173255)项目资助

共享,打造“智慧高校”平台<sup>[1]</sup>,并加快教育数字化进程以建设教育强国<sup>[2]</sup>。理论课程的线上授课较为成熟,但实验课程的线上教学仍存在显著挑战。实验教学不仅涉及理论知识的应用,还要求学生通过动手操作加深理解,这种过程在传统课堂中较难实现<sup>[3]</sup>。

目前,国内外已开发出多种线上实验平台,用于补充或替代传统实验室<sup>[4]</sup>,典型平台包括 SLD<sup>[5]</sup>、WebLab-Deusto<sup>[6]</sup>、UNILabs<sup>[7]</sup>及 NCSLab<sup>[8-9]</sup>等。这些平台通过远程连接真实实验设备或虚拟仿真实验设备,使用户能够实现与传统实验室相近的学习效果。然而,这些平台多以带操作系统的控制器(如树莓派、miniPC)为基础,控制周期通常为毫秒级(如 10 ms、40 ms 等)<sup>[10]</sup>,适合一般教学性实验的要求。然而,对于高速电机控制、工业级设备控制等复杂实验,毫秒级控制步长已难以满足实际需求,微秒级控制系统的实现成为研究热点<sup>[11]</sup>。

在此背景下,STM32 单片机以其高性能、低成本、低功耗以及对实时操作系统的支持<sup>[12]</sup>,成为实现微秒级控制的理想平台。相较于树莓派等控制器,STM32 不仅具备更高的实时性,还能通过定制化开发支持复杂的控制算法,从而满足高速电机和多机协同控制等工业级应用的需求<sup>[13]</sup>。特别是在永磁同步电机(permanent magnet synchronous motor, PMSM)的场向量控制(field-oriented control, FOC)等领域,微秒级的控制步长对于实现高动态响应至关重要<sup>[14]</sup>。

本研究基于 NCSLab (networked control system) 实验平台,提出了一种支持微秒级控制的 STM32 集成方案。用户可通过 NCSLab 平台搭建如 Simulink 般的图形化控制框图,平台将自动生成适配 STM32 的 C 代码并通过 Makefile 编译,最终使用 OpenOCD 工具实现代码烧录。此外,设计了一种高效的数据通信协议,支持实验状态的实时监控和参数在线调整,从而显著提升用户的调试效率与实验安全性。

在本研究之前,NCSLab 平台只支能实现对毫秒级控制步长的支持,本研究的研究不仅扩展了 NCSLab 平台的实验覆盖范围,使其能够支持工业级实验对象,还为高校实验教学提供了一种高效、安全且经济的解决方案。通过对 STM32 的软硬件架构优化与任务调度设计<sup>[15]</sup>,本研究实现了对微秒级控制系统的全面支持,为远程实验平台的发展提供了重要的技术支撑和实践参考。

此外,针对实时控制(real time control, RTC) 微秒级步长的控制,目前的主流解决方案主要依赖于 dSPACE 系统。然而,该方案仍然基于 Simulink 平台,且高度依赖国外技术,这在一定程度上限制了国产化进程。为此,本研究提出了一种完全基于自主知识产权的设计方案。该方案旨在突破国外技术壁垒,有效应对关键技术“卡脖子”问题,为推动相关领域的国产化发展提供了可能性。

## 1 系统总体架构设计

为实现对微秒级控制系统的支持,本研究基于 NCSLab 远程实验平台,设计了一套以 STM32 单片机为核心的系统架构。系统总体架构融合了硬件与软件的协同设计,覆盖从控制器硬件到代码生成、通信协议到实时监控的完整流程。通过模块化设计,系统具备高扩展性和灵活性,能够适应多种实验需求并支持复杂的控制任务。

### 1.1 NCSLab 实验平台架构

NCSLab 实验平台的整体架构设计实现了多层次分离与模块化的高效组合,以满足控制实验的实时性、可扩展性和跨平台需求。平台的前端采用 React 框架,提供高度互动和响应式的用户界面;后端(数据库)则基于 Laravel 框架,确保数据处理的稳定性和安全性;服务器端由 Tomcat 服务器承担核心业务逻辑和处理任务,确保在高并发条件下提供快速响应和负载均衡的性能。设备端则包含了与传统线下实验室类似的闭环控制要素,如控制器、执行器、被控对象、传感器等,确保用户在远程环境下获得完整的实验体验。NCSLab 的整体架构图如图 1 所示。

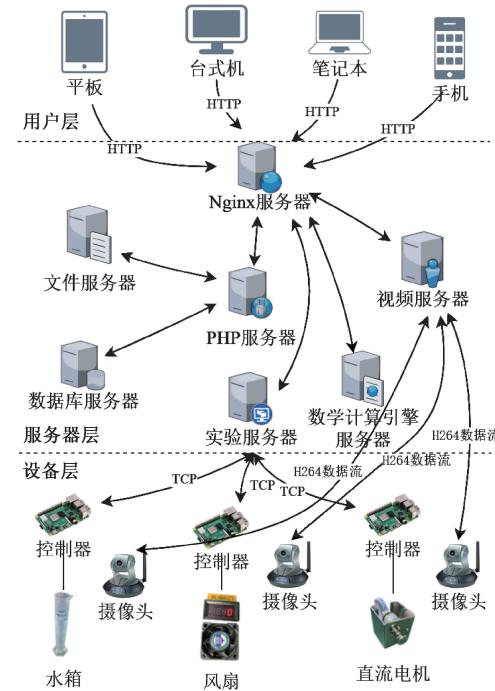


图 1 NCSLab 平台架构图

Fig. 1 Structure of the NCSLab platform

1) 客户端层。客户端层通过 Web 浏览器实现用户界面交互,利用 React 架构提供模块化、动态的页面呈现。用户可以通过拖拽模块的方式完成实验流程设置,例如控制算法框图的搭建和参数设置。该层采用前后端分离的架构,减少了对客户端硬件设备的依赖,提高了页面加载效率与用户操作的流畅性。此外,基于 React 的用户界面具有高度的跨平台兼容性,适用于 Windows、Linux、macOS、

Android 和 iOS 等操作系统,确保用户能够随时随地通过浏览器进行实验操作。

2) 服务器层。服务器层基于 Tomcat 服务器,作为中间层在客户端与设备端之间发挥数据处理和传输的关键作用。服务器层不仅承担了实验数据的实时传输,还提供自动代码生成、仿真计算、编译执行和数据缓存等功能。平台通过 Laravel 架构的数据库进行数据管理和用户权限控制,确保了实验数据的安全存储与高效管理。同时,服务器层具备负载均衡功能,能够在多用户并发访问时实现性能的动态调节,保障系统的稳定性和实时性。

3) 设备层。设备层包含控制实验所需的核心组件,类似于传统实验室的闭环控制系统,包括控制器、执行器、被控对象和传感器等。这些设备通过与服务器层的双向通信实时接收用户指令,并将实验数据反馈至客户端,实现了用户在远程环境中对真实物理设备的控制。设备层具备高度的模块化接口设计,支持多种实验装置的快速切换和灵活配置,使平台适用于不同实验对象的需求。

该架构不仅实现了实时数据传输与远程监控,还通过前后端分离和模块化设计提高了平台的扩展性和适应性。React 框架的客户端层提供了良好的用户交互体验; Laravel 框架管理的数据库提高了数据安全性和可控性; Tomcat 服务器的引入则优化了业务逻辑处理与并发性能。整体架构实现了在线实验的灵活、实时和跨平台操作。

## 1.2 STM32 控制器的设计

为了实现微秒级控制任务,STM32 控制器作为 NCSLab 平台的重要组成部分,其设计需要满足高性能控制任务和网络通信的双重需求,同时能够无缝集成到 NCSLab 平台的现有框架中。在此基础上,系统对 STM32 控制器进行了针对性的软件设计,确保控制任务的实时性和通信任务的稳定性,并通过模块化的设计使其具备高度扩展性和易用性。STM32 控制器的软硬件结构如图 2 所示。

在微秒级控制系统中,任务调度的高效性和实时性直接决定了系统整体性能的优劣。本研究采用 FreeRTOS (free real-time operating system) 作为实时操作系统,借助其灵活的任务管理机制、多任务调度能力以及资源共享管理功能,实现了控制任务与通信任务的有效协同,以满足 NCSLab 平台的微秒级控制需求。FreeRTOS 通过优先级调度、信号量、消息队列和任务通知等核心机制,为系统的微秒级控制和高频数据传输提供了基础架构支持。

控制任务是系统的核心,其主要负责执行用户通过图形化界面搭建的控制算法。该任务由 STM32 硬件定时器 (TIM) 周期性触发,定时器被配置为高精度微秒级定时,其优先级被设置为最高,确保在控制周期到来时任务能够立即抢占 CPU 资源进行执行。例如,在 20 μs 的控制周期内,控制任务需要采集传感器信号(如电流、电压)、计算控制输出(如 PWM 信号)、并将信号更新至执行器(如电机驱

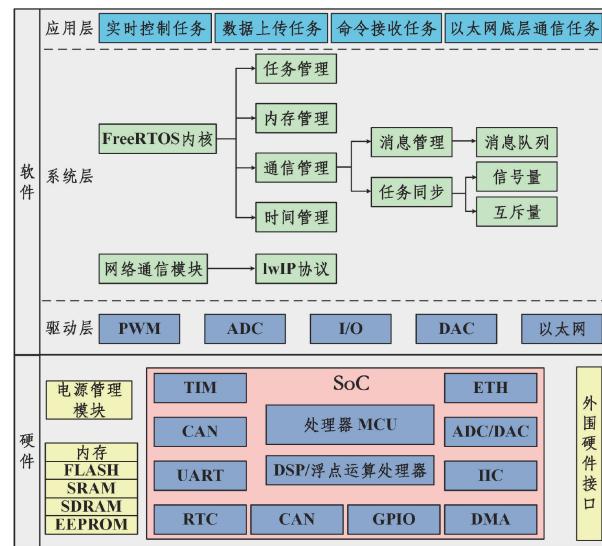


图 2 STM32 控制器的软硬件结构图

Fig. 2 Hardware and software architecture of the STM32 controller  
动器)。通过 FreeRTOS 的硬件中断处理机制,控制任务得以实现严格的时间约束,不受其他任务的干扰。

通信任务则包括数据上传任务和命令接收任务。为避免通信任务对控制任务的干扰,通信任务被赋予较低的优先级。数据上传任务通过 FreeRTOS 的软定时器定期唤醒,将缓存中的数据打包后发送至服务器;命令接收任务则监听服务器下发的控制指令,通过 FreeRTOS 的消息队列机制将指令存储至缓冲区,确保指令不会因网络波动而丢失。在任务切换过程中,FreeRTOS 通过任务通知机制快速完成任务间的上下文切换,例如在接收到新指令时,命令接收任务能够立即唤醒控制任务并触发参数更新,确保控制算法在下一控制周期内使用最新的控制参数。STM32 任务间的协同调度及实验数据传输过程和任务执行顺序分别如图 3 和图 4 所示。控制任务周期性执行不受通信任务的干扰。在控制任务完成后,数据上传任务周期性执行,而命令接收任务由于比数据上传任务有着更高的优先级,他会将数据上传任务打断,等该任务完成后,数据上传任务再接着运行。命令接受任务优先级比数据上传任务优先级高的原因在于参数的修改会立即影响控制系统的运行状态,而数据上传时丢失几个包并不会对控制系统的运行状态产生影响,数据上传仅供用户观察实验状态,因此即便小概率的丢包,也无关紧要。

信号量机制在任务间资源共享中发挥了关键作用。例如,当控制任务需要访问共享缓冲区写入传感器数据时,信号量会锁定该资源,防止通信任务在写入过程中读取未完成的数据。二进制信号量用于单次资源访问的保护,而计数信号量则管理多任务对同一资源的访问顺序。通过信号量的灵活运用,系统避免了任务间的资源争夺,提高了数据传输和处理的可靠性。

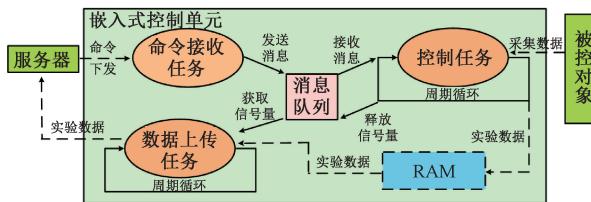


图 3 STM32 控制器任务调度及实验数据传输流程图

Fig. 3 Task scheduling and experimental data transmission flowchart of the STM32 controller

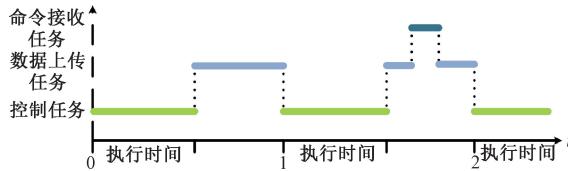


图 4 STM32 控制器内的任务执行顺序

Fig. 4 Task execution order in the STM32 controller

### 1.3 高频数据采集与实时数据处理机制

在微秒级控制系统中,高频数据采集和实时数据处理通过 FreeRTOS 的多任务调度机制得以高效实现,具体包括控制任务、数据上传任务和命令/参数接收任务的协同运行,其整体流程如图 3 所示。整个数据处理过程涵盖了从传感器信号采集、控制器执行实时算法、数据打包到网络传输的完整链路,确保了系统在微秒级实时性需求下的稳定性和高效性。

1) 数据采集与任务调度的实现。系统的核心控制任务由硬件定时器 TIM 周期性触发。该任务在每个控制周期(如 20 μs)内采集传感器信号(如电流和电压),并执行图形化框图生成的控制算法。以电流和电压的采集为例,采集信号通过 STM32 的模拟数字转换器(analog-to-digital converter, ADC)模块直接输入,利用直接内存存取(direct memory access, DMA)技术将数据从 ADC 传输到随机存取存储器(random access memory, RAM)中,避免了 CPU 参与数据搬运,提高了数据采集的效率。

在控制任务的执行过程中,ADC 采集的电流和电压信号会被传递到控制算法模块,计算输出控制信号(如 PWM 占空比)。计算完成后,任务会通过 PWM 生成模块更新硬件 PWM 信号,直接作用于执行器(如电机驱动器)以调节系统响应。控制任务通过硬件中断触发,其优先级设为最高,确保每个控制周期都能精准执行。

2) 数据传输的任务协同与优化。网络通信任务主要包括数据上传任务和命令接收任务。数据上传和命令接收通过 TCP 网络进行,基于轻量级 IP 协议栈(lightweight IP, LwIP)实现通信功能。任务间的协同通过 FreeRTOS 的优先级和信号量机制实现。

在数据上传阶段,数据上传任务会将 RAM 中的控制信号和传感器数据打包为 JSON 格式的数据包。数据包包

含时间戳、信号类型、信号值以及与控制器和服务器之间的通信标识。通过 FreeRTOS 的队列机制,数据上传任务将数据放入传输队列中,由 LwIP 协议栈负责通过以太网接口(ETH)发送至服务器。任务上传频率可以根据用户需求动态调整。当控制周期为 20 μs 时,为避免数据量过大,系统每隔 50 个控制周期采集一次数据并上传,即设置上传周期为 1 ms。数据上传过程通过 FreeRTOS 软定时器触发,确保数据传输与控制任务解耦。

在命令接收阶段,服务器向控制器下发参数调整或算法变更指令,命令接收任务通过 TCP 监听端口接收指令包。当服务器发送新指令时,LwIP 协议栈触发网络中断,将指令传递到任务队列。命令接收任务的优先级高于数据上传任务,确保新指令能够立即被处理。指令处理后,任务会更新控制器的内部参数,并通知控制任务在下一个控制周期内使用新的参数执行控制算法。

3) 数据同步与实时性保障。为确保数据处理与传输的时间一致性,系统采用 FreeRTOS 的任务通知和信号量机制。命令接受任务通过任务通知机制向控制任务传递参数更新信号,避免了任务之间的数据竞争。数据上传任务则使用二进制信号量来同步队列中数据的写入和读取操作,确保数据包的完整性。

在通信层面,系统利用 LwIP 协议栈的非阻塞式发送和接收模式,减少了数据传输对 CPU 资源的占用。通过优化 TCP 窗口大小和数据包长度,系统在保证数据传输稳定性的同时,降低了网络延迟。控制任务在执行过程中始终保持最高优先级,通信任务的执行仅在控制任务完成后占用 CPU 资源,从而实现了数据处理和传输的高效协同。

4) 数据处理与反馈的闭环机制。在整个数据处理闭环中,系统首先通过 ADC 采集传感器信号,并在控制任务中完成信号处理和算法计算。随后,计算结果通过 PWM 模块输出至执行器,同时采集的数据通过数据上传任务传输至服务器,形成完整的闭环控制系统。

## 2 STM32 代码的自动生成、编译和烧录

在 STM32 控制器中,由于其嵌入式性质和对实时性要求的极高需求,自动代码生成需直接面向硬件资源,如寄存器、ADC、PWM 等外设模块,并以 HAL 库为基础进行模块化封装,确保代码的兼容性和可移植性。STM32 的代码生成强调硬件定时器和外设资源的直接调用,通过解析用户在 NCSLab 平台上的图形化控制算法,系统将算法逻辑映射到 STM32 的底层硬件配置中,并生成控制器可直接执行的.hex 文件。同时,本研究对自动代码生成的关键步骤进行了详细描述,包括图形化算法解析、代码模块封装、交叉编译及烧录下载等,图 5 展示了 STM32 控制在 NCSLab 平台上的应用流程。

### 2.1 图形化控制算法的解析与代码的自动生成

在 NCSLab 平台上,STM32 控制器的嵌入式应用需要

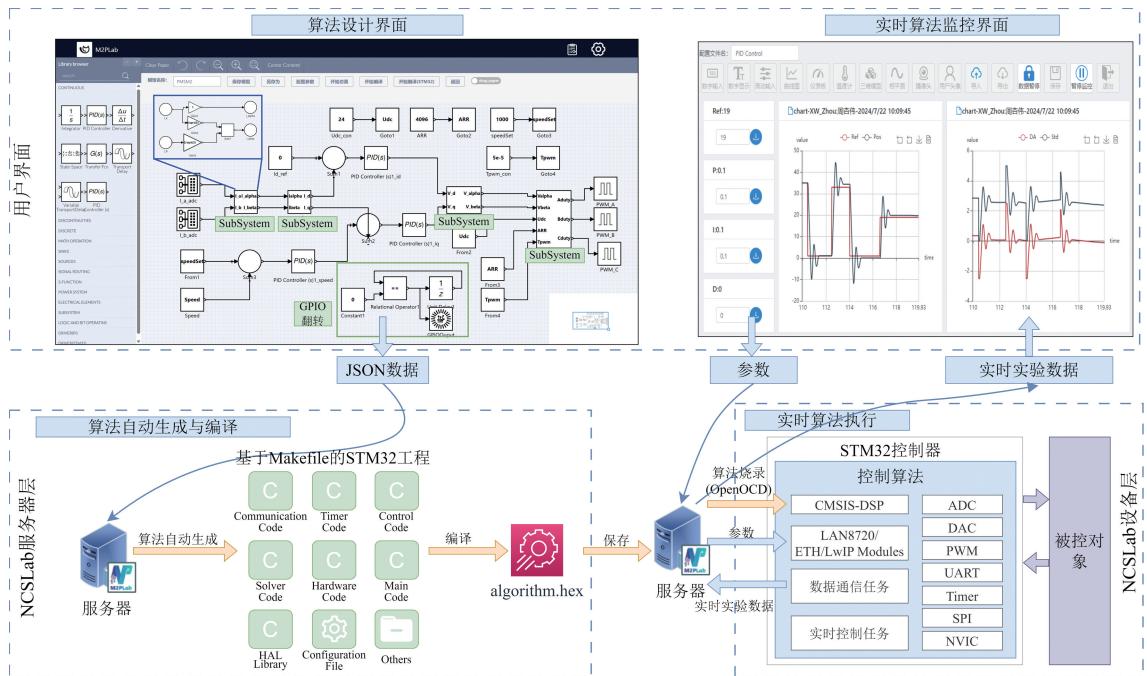


图 5 STM32 控制器在 NCSLab 平台上工作全流程

Fig. 5 The complete workflow of the STM32 controller in the NCSLab platform

同时实现控制任务和通信用任务。通信用任务基于 NCSLab 平台的固定协议进行设计,其功能模块是预先定义的,无需因用户搭建的控制算法变化而调整。因此,通信用任务的代码可以作为自动生成的模板代码固定写入,不随具体应用而更改。而控制任务则因用户在前端所搭建的控制框图而具有动态变化的需求。为此,系统需要通过解析图形化控制算法,并依据设定的规则自动生成控制代码,以满足不同实验的需求。

用户通过拖拽模块的方式完成控制框图的搭建后,平台前端会将框图的描述信息以 JSON 格式发送至服务器。服务器解析这些 JSON 数据,对 STM32 工程的代码进行自动生成。服务器的代码生成架构可划分为解析层、处理层、输出层和封装层 4 个部分,如图 6 所示。这一分层设计确保了从算法逻辑解析到代码输出的流程具有高效性和可扩展性。

1) 解析层。负责读取和解析前端传递的 JSON 数据,提取模块类型、连接关系及模块参数等关键信息,并生成模块的初步描述,为代码自动生成/组成做准备。

2) 处理层。中间的处理步骤包括配置求解器、构建输出链路和检测代数环。配置求解器是为了在生成代码时引入适当的求解算法,构建正确的输出链路是确保框图算法准确运行的基础,而代数环检测可以有效防止算法陷入死锁状态。

3) 输出层。根据处理层生成的输出链路,调用预定义的代码模板为每个模块生成对应的代码块。代码块包括初始化函数、更新函数、输出函数等。输出层主要负责代码生

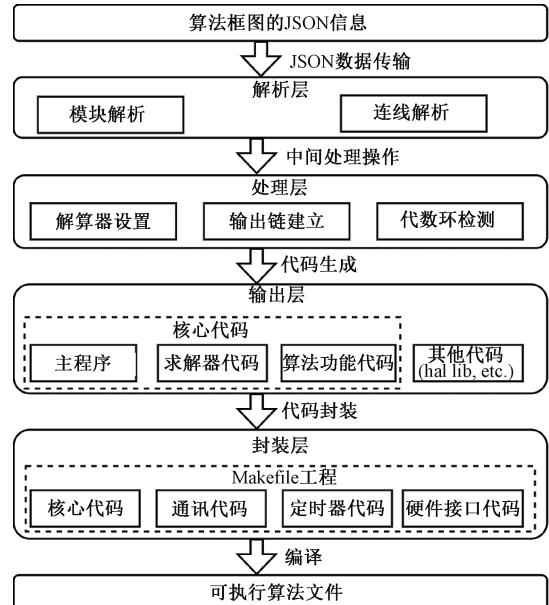


图 6 服务器层的算法设计及其自动生成流程

Fig. 6 Algorithm design and its automatic generation process at the server layer

成,生成的代码包括主程序、求解器代码、算法功能代码以及其他相关文件。

4) 封装层。在编译过程中,需要封装核心控制代码、通信代码、定时器代码和硬件接口代码,形成完整的 STM32 控制代码工程。最终,系统会进行编译,生成可执行文件。

通过上述 4 层结构的处理,服务器能够将前端搭建的

图形化控制算法框图自动转换为基于 STM32 工程的 C 代码工程。自动生成的代码结构包括以下主要函数模块：NCSLabInit()、NCSLabOutput()、NCSLabUpdate()、NCSLabDerivative() 和 NCSLabDerivativeUpdate()。这些函数分别用于初始化模块、计算和输出结果、更新连续模块状态、计算离散模块的导数值，以及更新离散模块的状态。这些函数为代码的核心逻辑提供了必要的功能支持。在具体实现中，前端的每个框图模块对应服务器中一组定义明确的代码块。这些代码块由上述核心函数组成，并在处理层中按照预设的输出链路进行组合。输出链路的构建是自动代码生成中的关键部分，它决定了各模块代码块的组合顺序。若输出链路的顺序不正确，生成的控制代码逻辑将出现错误，导致系统运行异常。因此，输出链路的准确性是自动代码生成流程中的核心保障。

输出链路的构建采用递归遍历方法，确保模块代码顺序与控制逻辑一致。如图 7 所示，系统从终端模块（如 Scope 模块）开始向前递归遍历，直至找到起始模块或已被遍历的模块。对于多输入端口的模块，按照输入端口顺序依次向前递归。在图 7 中，下方蓝色标号表示模块的遍历顺序，而上方红色标号表示生成代码块的排列顺序。通过这种遍历方式，服务器能够高效建立所有模块的输出程序。

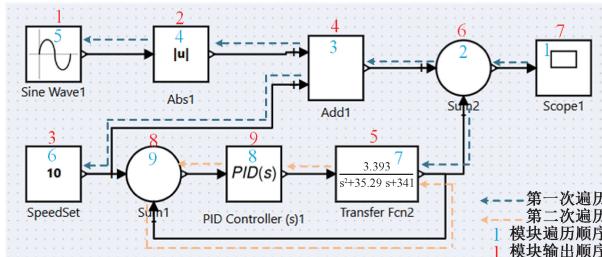


图 7 图形化控制框图代码生成输出链设计

Fig. 7 Design of the code generation output chain for the graphical control block diagram

块顺序，生成的代码逻辑准确无误。

在解算器的实现中，控制任务采用四阶龙格-库塔（Runge-Kutta, RK4）算法对积分、微分及传递函数等模块的状态值进行数值求解。RK4 方法是一种高精度的常微分方程数值解法，适用于对初值问题的精确计算。假设给定一个常微分方程的初值问题，其数学形式为：

$$y'(t) = f(t, y), y(t_0) = y_0 \quad (1)$$

RK4 方法以步长  $h$  前进，每一步的递推公式为：

$$y_{n+1} = y_n + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4) \quad (2)$$

其中， $k_1, k_2, k_3, k_4$  分别定义如下：

$$k_1 = hf(t_n, y_n) \quad (3)$$

$$k_2 = hf(t_n + \frac{h}{2}, y_n + \frac{k_1}{2}) \quad (4)$$

$$k_3 = hf(t_n + \frac{h}{2}, y_n + \frac{k_2}{2}) \quad (5)$$

$$k_4 = hf(t_n + h, y_n + k_3) \quad (6)$$

RK4 方法的高效性和精度使其特别适合控制任务中状态值的递推计算。在控制任务的每个时间步长中，RK4 算法以当前状态值为基础，结合系统动态方程计算出下一步的状态值。

以基础的 PID 闭环控制算法的代码生成为例，全面地对自动代码生成的实现进行阐述说明。在算法设计界面设计如图 8 所示的 PID 闭环控制框图，每一个模块都能够双击点开进行参数的设置。当用户在前端将算法框图搭建好后，即可发送编译指令。该指令会将算法框图的信息以 JSON 形式进行打包，该 PID 控制框图的 JSON 信息如图 8 所示。该 JSON 中包含了每一个模块及每一个连线的信息。模块信息中包含模块 id、模块名、模块类型、内部参数信息等；连线信息中包含了连线 id、连线起始接口、连线终止接口等信息。该 JSON 将整个框图的所有模块和连线的信息都进行了打包，接着被发送到服务器后进行解析。

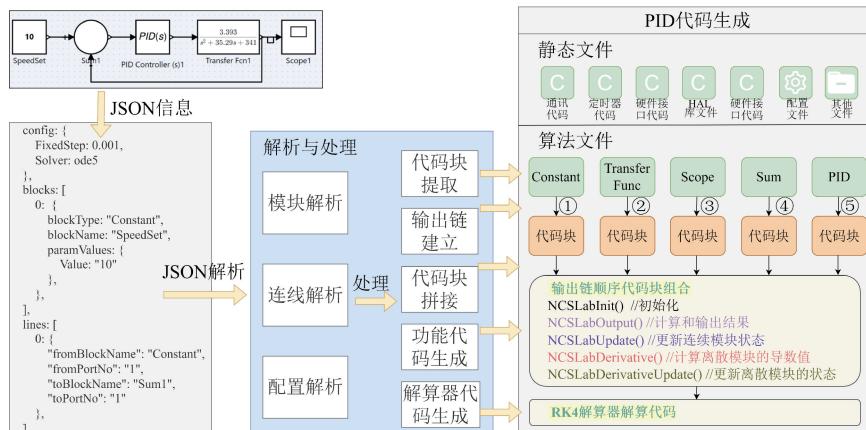


图 8 PID 控制框图自动代码生成流程

Fig. 8 Automatic code generation process for the PID control block diagram

服务器根据 JSON 中的各个模块的信息,对该模块的 C 语言代码块进行内部参数设置,并根据构建的输出链按照顺序将一个一个模块的代码块进行拼接。具体而言,每个模块都有其各自对应的函数模块(即:NCSLabInit()、NCSLabOutput()、NCSLabUpdate()、NCSLabDerivative() 和 NCSLabDerivativeUpdate()),通过输出链的顺序,将这些模块对应的函数模块进行拼接,最终生成出 C 代码工程的函数模块。在每个控制周期中,使用 RK4 求解器进行数值求解,其过程如图 8 所示。至此,完成了 PID 闭环控制自动代码生成的全部流程。

对该 PID 闭环控制框图使用同样的参数在 NCSLab 和 Simulink 中分别进行仿真,记录下示波器的输出,将 NCSLab 的输出结果与 Simulink 的输出结果进行对比,如表 1 所示,可以看出,NCSLab 平台基于 RK4 求解器的输出结果与 Simulink 的输出结果一致。证明了 NCSLab 中 RK4 求解器计算出的结果的正确性。

**表 1 NCSLab 与 Simulink 平台仿真输出结果对比**

**Table 1 Comparison of Simulation output results between NCSLab and simulink platforms**

时间/s	NCSLab	Simulink
0	0	0
0.01	0.015 097 207 802 137 498	0.015 097 207 802 138
0.02	0.058 287 181 001 213 904	0.058 287 181 001 214
0.03	0.128 439 183 756 887 37	0.128 439 183 756 887
0.04	0.223 882 203 348 654 14	0.223 882 203 348 654
0.05	0.342 594 004 115 903 2	0.342 594 004 115 903
0.10	1.205 444 762 317 752 9	1.205 444 762 317 753
0.50	8.097 674 713 675 588	8.097 674 713 675 590
1.00	9.794 965 848 655 332	9.794 965 848 655 334
5.00	9.999 999 996 587 064	9.999 999 996 587 064

## 2.2 控制代码的编译、烧录与执行

STM32 控制程序按照输出链逻辑自动生成后,在服务器端采用交叉编译工具链 arm-none-eabi-gcc 进行编译,该编译过程由系统在代码生成后自动触发,无需用户干预。当用户在 NCSLab 平台的前端点击“算法下载”按钮时,服务器将自动与 STM32 控制器建立通信,通过 CMSIS-DAP、J-Link 或 ST-Link 等编程器将生成的.hex 文件烧录至目标控制器中。烧录过程依赖开源调试工具 OpenOCD 完成,其指令流包括编程器连接初始化、目标文件加载和写入指令执行,确保代码能够正确写入 STM32 的非易失性存储器 FLASH 中。

为了支持 NCSLab 平台对大规模 STM32 控制器的管理和控制,系统在服务器与 STM32 控制器之间引入了中间节点设备——树莓派。树莓派作为中间通信桥梁,通过网络接收来自服务器端的编译后.hex 文件,并与 STM32

控制器建立物理连接,再通过 CMSIS-DAP 或 ST-Link 等烧录工具完成文件的写入操作。树莓派的引入解决了直接连接多台 STM32 控制器所面临的硬件接口限制问题,大幅提高了系统的扩展性和部署灵活性。这一过程如图 9 所示。

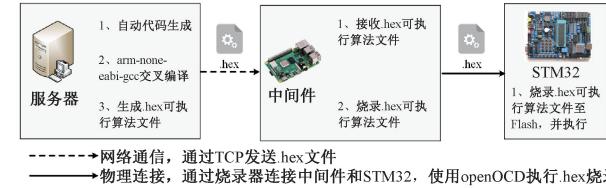


Fig. 9 Compilation, programming, and execution of the control code

这种中间节点的架构设计具有显著优势。服务器端无需为每个 STM32 控制器提供独立的物理接口,只需通过网络将烧录任务下发至树莓派,即可在分布式环境中实现对多控制器的管理。

针对 STM32 项目文件数量庞大且完全编译耗时较长的问题,系统通过优化编译流程显著降低了编译时间。具体而言,通用的且不需动态调整的 C 文件在代码模板创建阶段即完成预编译,其生成的目标文件(.o 文件)存储在 STM32 代码模板目录中。在后续的编译过程中,服务器仅需编译少量与用户自定义控制算法相关的 C 文件(通常为 15~16 个),并通过修改 Makefile 文件确保预编译的目标文件被正确链接到最终的项目中。这一优化设计有效避免了重复编译的冗余工作,极大地提高了编译效率,减少了用户等待时间。表 2 列出了采用预编译和完全编译的编译时长对比。

**表 2 预编译和完全编译的编译时长对比**

**Table 2 Comparison of compilation time between pre-compilation and full compilation**

编译形式	待编译.c 文件个数	预编译.c 文件个数	编译 时间/ms	文件工程 形式
预编译	15	95	1 019	Makefile
完全编译	110	0	7 642	Makefile

通过上述编译、烧录与执行流程,NCSLab 平台实现了从控制算法设计到硬件部署的全流程自动化。结合中间节点设备树莓派的引入,系统不仅克服了传统硬件接口的限制,还为大规模部署与管理提供了技术保障。

## 3 应用案例:永磁同步电机的场向量控制

永磁同步电机广泛应用于工业自动化、机器人及新能源汽车等领域,其高效性、精确性和快速动态响应能力使其成为电机驱动控制研究的重点。然而,PMSM 的高性能运行依赖于精确的控制策略,其中场向量控制是一种主流

的控制方法。FOC 能够在旋转坐标系下解耦电流分量,使电机获得与直流电机类似的控制特性,从而实现高动态性能的速度和转矩控制。

本节基于 NCSLab 平台的微秒级控制系统设计与实现,选取 PMSM 的 FOC 控制为实验案例,详细探讨 FOC 控制算法的理论基础、硬件和软件系统配置,以及实验数据的分析与性能评估。通过这一应用案例,验证 NCSLab 平台在高实时性、复杂控制算法实现中的有效性和适用性。

### 3.1 场向量控制算法的理论背景与控制要求

FOC 控制是一种基于旋转坐标系的矢量控制方法,通过坐标变换将 PMSM 的三相交流电流解耦为转矩电流和励磁电流两部分,分别控制电机的转矩输出和磁场强度。这种控制方法能够实现高动态性能的速度和转矩控制,其理论基础主要包括数学建模、坐标变换和解耦控制。

对 PMSM 的控制,实际上是对图 10 三个定子通电流进行控制,其本质就是通过控制 6 个桥臂的导通和闭合同时给 3 个定子供给变化的电流,以驱动 PMSM 转动。

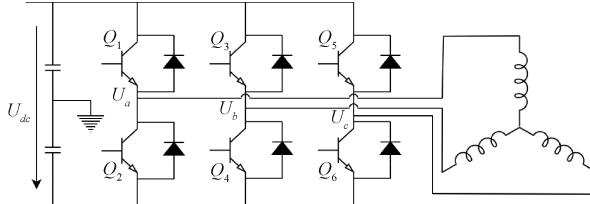


图 10 永磁同步电机的三相逆变器电路拓扑结构

Fig. 10 Three-phase inverter circuit topology for permanent magnet synchronous motor

在场向量控制系统中,采用双闭环 PID 控制结构以实现电流分量和速度的精确控制,其整个控制流程如图 11 (a) 所示。其控制结构由内层的电流环和外层的速度环组成。在双闭环控制中,速度环作为外环,产生电流环的参考值  $i_q^*$ ,而  $i_d^*$  通常设置为 0,以确保磁链分量最小化。电流环作为内环,直接控制电流分量,从而调节电机的电磁转矩和磁链,实现系统的动态响应和稳态性能。通过双闭环 PID 控制得到  $v_d$  和  $v_q$  后,需要通过 Park 和 Clarke 逆变换将其转换回三相电压信号  $v_a, v_b, v_c$ ,从而驱动逆变器生成 PWM 输出信号。

### 3.2 实验系统配置与控制算法实现

根据场向量控制算法的理论及图 11 (a) 所示的控制框图,在 NCSLab 平台中搭建了如图 11 (b) 所示的控制算法模型。为了提高控制框图的清晰度和模块化程度,不同功能模块通过子系统进行封装。同时,硬件接口模块(如 ADC、PWM)用于实现与实体永磁同步电机的物理连接,且通过配置这些模块的内部通道、频率及其他参数确保硬件接口的准确性和实时性。

在前端完成如图 11 (b) 所示控制算法的搭建后,将控制步长配置为  $25 \mu s$ ,以满足微秒级控制系统的实时性要

求。此外,还需配置 STM32 控制器的 IP 地址和端口信息,以便服务器能够与 STM32 控制器建立网络连接,从而实现实时监控与参数调整功能。通过 2.1 节的代码的自动生成后,图 11 (b) 所示的控制框图能够自动生成出基于 STM32 的 FOC 控制代码。并在服务器上通过交叉编译完成基于 STM32 的 ARM 架构的代码编译。编译完成后,通过服务器将生成的 hex 文件烧录至 STM32 控制器中,即可实现算法的执行。

PMSM 实物控制平台的搭建如图 12 所示。实验系统的硬件部分包括 PMSM、电机驱动板和 STM32 控制器。具体连接方式为:1) PMSM 与驱动板进行硬件连接,驱动板为 PMSM 提供电压和电流以驱动电机转动。2) 驱动板与 STM32 控制器之间通过信号线进行连接。驱动板将弱电信号(如电流和位置信息)通过 ADC 通道反馈至 STM32 控制器,用作控制算法的输入反馈信号;同时,STM32 控制器通过 PWM 通道发送控制信号至驱动板,用于调节电机运行状态。

当控制算法下载烧录进入 STM32 后,即可在 NCSLab 实验平台上搭建如图 11 (c) 所示的监控组态界面。在监控组态中,添加数字输入框和曲线图组件,并将关心的参数和信号与数字输入框和曲线图关联在一起。在本案例中,所关心的参数为设定转速、3 个 PID 模块的内部 PI 参数以及电机的实际转速。搭建如图 11 (c) 所示的监控组态界面后,即可在前端界面中实时修改参数的值以更改控制算法的运行状态,通过修改速度设定值和 3 个 PID 内部 PI 参数以观测 PID 算法的控制效果,即在曲线图中观测实际转速是否快速达到了转速设定值。

### 3.3 实验数据分析与系统性能评估

为验证控制系统是否能够以预设的微秒级控制步长正常工作,在控制框图中添加了一个 GPIO 模块,如图 11 (b) 所示,并配置为每隔一个控制步长( $25 \mu s$ )翻转一次。通过外接示波器实时观测该 GPIO 端口的输出信号波形,如图 13 所示。从示波器图像可以清晰地看到,GPIO 信号以预设的  $25 \mu s$  控制步长稳定翻转,证明控制任务能够按照设定的周期精确执行,满足 RTC 微秒级控制的实时性要求。

在控制效果评估方面,选择 MATLAB/Simulink 仿真工具作为基准进行对比分析。在 Simulink 中搭建与实物实验一致的控制框图模型,并将仿真程序的所有参数按照 PMSM 实际参数值进行设置,以确保仿真与实物实验的条件完全一致。图 14 展示了 Simulink 仿真得到的电机运行结果,而图 11 (c) 则为实物控制实验的实际运行结果。

通过对比仿真与实物的控制效果,可以发现两者在动态响应、稳态精度等方面的表现非常接近。仿真结果与实物实验结果的高度一致性验证了 NCSLab 平台微秒级控制系统的可靠性和有效性。这一结果不仅表明控制算法能够稳定运行于实际硬件环境中,也证明了通过 NCSLab

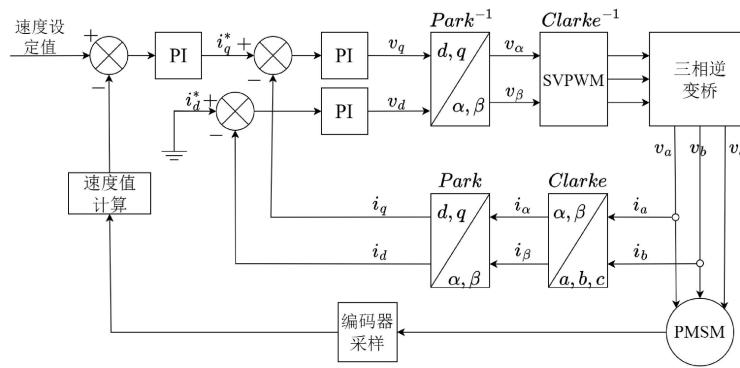
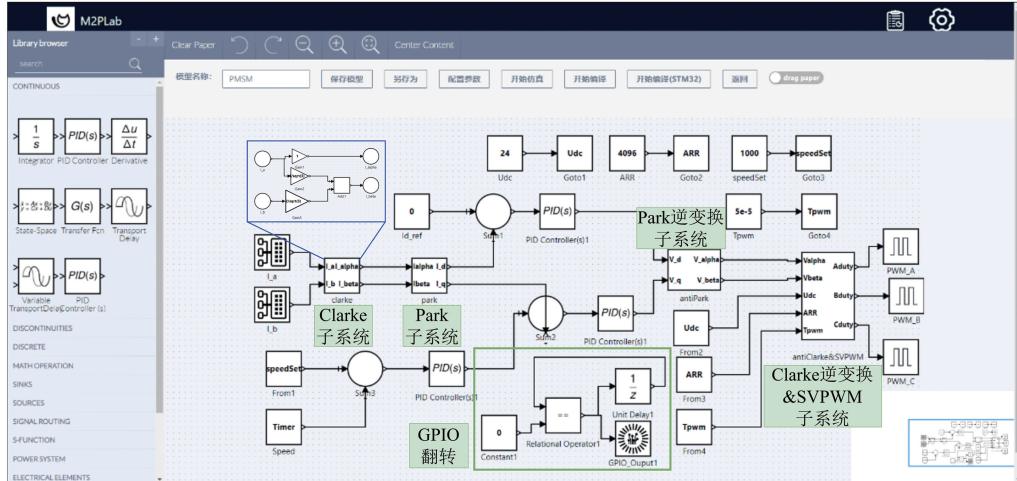
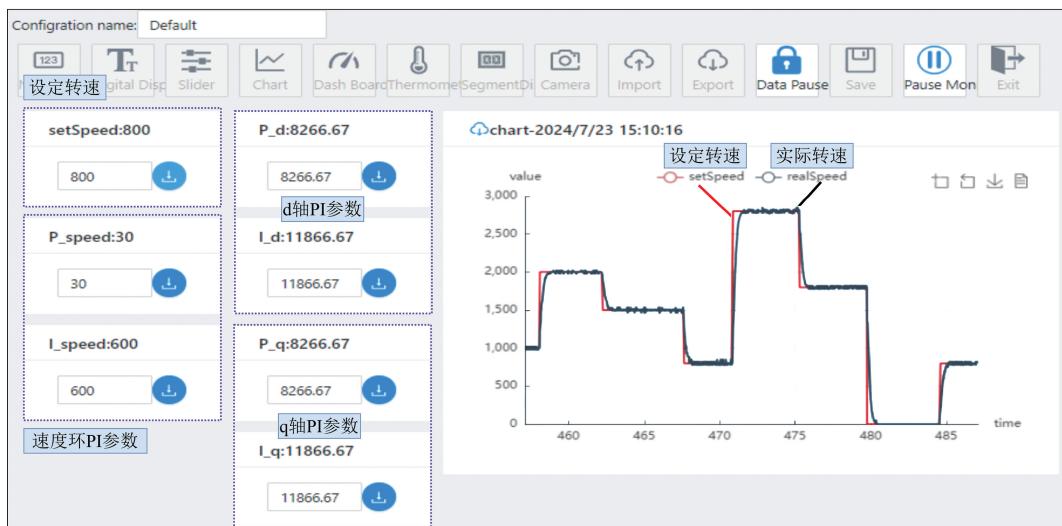
(a) 永磁同步电机的场向量控制原理框图  
(a) Block diagram of FOC for PMSM(b) 永磁同步电机在NCSLab平台的FOC算法搭建界面  
(b) FOC algorithm setup interface for PMSM in the NCSLab platform(c) 永磁同步电机在NCSLab平台的FOC算法的监控界面  
(c) Monitoring interface of the FOC algorithm for PMSM in the NCSLab platform

图 11 永磁同步电机在 NCSLab 中的算法搭建及实时监控

Fig. 11 Algorithm setup and real-time monitoring of the permanent magnet synchronous motor in NCSLab

平台生成的控制代码在算法实现与硬件执行上的严谨性和高效性。同时这也表明, NCSLab 能够对 RTC 设计开发

进行全流程覆盖, 从算法设计到实时监控都能够通过国产化自主研发的 NCSLab 平台得到实现。

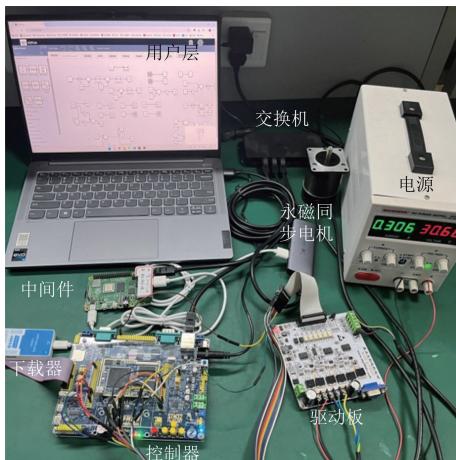


图 12 永磁同步电机实物实验平台

Fig. 12 Physical experimental platform of the permanent magnet synchronous motor

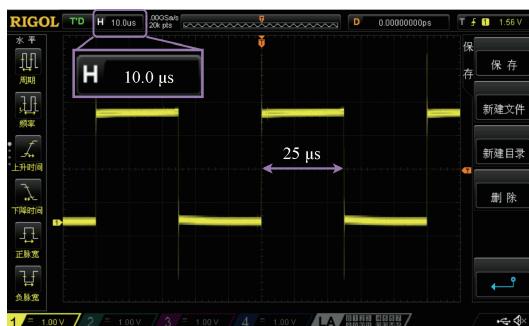


图 13 控制步长验证

Fig. 13 Verification of control step size

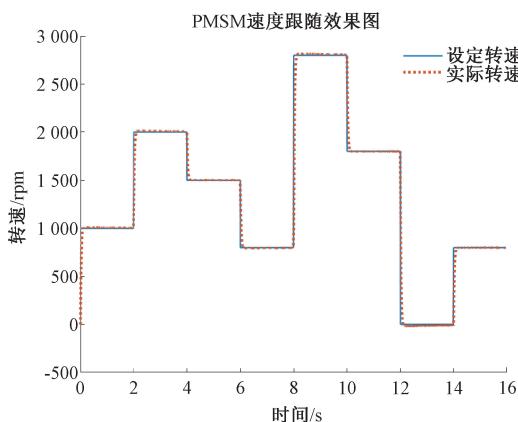


图 14 FOC 算法在 Simulink 中的仿真运行结果

Fig. 14 Simulation results of the foc algorithm in simulink

## 4 国内外平台对比

本研究提出的基于 STM32 微控制器的微秒级采样实时控制系统, 在控制性能上能够达到如 dSPACE、RTLab

等国外平台的微秒级控制步长和控制精度。除此之外, 具有显著的技术优势, 特别是在成本、灵活性、开发效率和自主可控性等方面, 相较于国外成熟平台, 展现出独特的优势。

相比于 dSPACE 和 RTLab 系统, STM32 的硬件成本低廉, 适合中小型企业和教育科研使用。dSPACE 和 RTLab 系统由于其高端硬件平台和全套集成解决方案, 其成本较高, 通常仅适用于大型工业应用或高性能需求的场景。STM32 系统能够在更低的成本下实现高精度控制, 尤其适用于国内市场的工业应用和教育领域。

STM32 基于开放式架构, 支持多种开发工具和操作系统, 提供更高的灵活性, 尤其在定制化和嵌入式控制方面表现优异。相比之下, dSPACE 和 RTLab 系统通常为封闭式硬件与软件集成方案, 虽然性能强大, 但在灵活性上较为局限, 难以快速响应不同应用需求。

本研究所提出的方案完全基于自主知识产权, 避免了对国外技术的依赖, 具有完全的自主可控性。dSPACE 和 RTLab 依赖国外技术, 尤其是在硬件和软件平台上, 面临技术封锁和更新周期的不确定性。因此, 本研究方案更适合国内产业自主化和国产化替代的需求。

STM32 系统通过 FreeRTOS、自动代码生成和远程实验平台等技术, 大幅提升了开发效率。dSPACE 和 RTLab 虽然提供高性能的开发工具, 但其集成环境较为复杂, 开发周期较长, 尤其在小规模和快速迭代的应用中, STM32 的开发效率更为突出。

通过上述对比分析, 可以看出 STM32 微控制器方案在多个维度上均表现出明显的优势。尤其是在成本、灵活性和自主可控性方面, 与国外高端平台如 dSPACE 相比, STM32 提供了更为经济、灵活且自主可控的解决方案。这使得本方案在国内产业化应用和国产化替代过程中具有强大的潜力和竞争力。

## 5 结论

本研究基于 STM32 微控制器设计并实现了一种微秒级采样实时控制系统, 解决了现代控制系统对高实时性和高精度的需求。通过引入 FreeRTOS 实时操作系统进行任务调度、硬件定时器实现控制任务的高优先级执行、LwIP 协议支持低延迟网络通信以及图形化框图的控制算法解析与自动代码生成, 本研究实现了高效、稳定的微秒级实时控制系统。

实验验证表明, 所提出的系统在微秒级采样条件下具有优异的实时性和控制性能, 特别是在永磁同步电机场向量控制的案例中, 表现出了高度的动态响应能力与控制精度。同时, 系统自动代码生成和远程实验平台的集成方案显著提高了用户的开发效率, 为工业级控制系统和高校实验教学提供了便捷、安全且高效的技术支撑。

本研究针对 RTC 领域中现有技术对国外平台和技术

的高度依赖问题,提出了一种完全基于自主知识产权的创新方案。该方案不仅实现了对微秒级步长的高精度控制,还为突破国外技术封锁提供了一种切实可行的解决路径。研究成果表明,本方案具有显著的技术优势和应用前景,不仅有助于解决关键技术“卡脖子”问题,也为相关领域的国产化进程注入了新的动力。然而,本研究仅通过永磁同步电机的场向量控制实验进行了验证,未来的研究可以进一步扩展至其他典型工业应用场景的验证,如伺服驱动、风力发电等,以进一步验证该系统的普适性与稳定性。未来,随着进一步的优化与推广,本方案有望在更广泛的工程实际中发挥重要作用。

## 参考文献

- [1] 朱继红. 大数据背景下高校智慧校园一站式平台建设[J]. 辽宁师专学报(自然科学版), 2024, 26(4): 31-36.
- ZHU J H. Exploration on the construction of a one-stop platform for smart campus in universities under the background of big data[J]. Journal of Liaoning Normal Colleges(Natural Science Edition), 2024, 26(4):31-36.
- [2] 赵俊芳, 范家宝. 高等教育治理数字化转型的目标指向、现实悖论与优化路径[J/OL]. 苏州大学学报(教育科学版), 1-9 [2024-12-25]. <http://kns.cnki.net/kcms/detail/32.1843.G4.20241224.1642.007.html>.
- ZHAO J F, FAN J B. The goals, practical paradoxes, and optimization paths of the digital transformation of higher education governance[J/OL]. Journal of Soochow University(Educational Science Edition), 1-9 [2024-12-25]. <http://kns.cnki.net/kcms/detail/32.1843.G4.20241224.1642.007.html>.
- [3] 薛莉玮, 刘国平, 胡文山. 考虑可切换拓扑的网络化多智能体快速原型开发系统[J]. 现代制造工程, 2024(10):41-48.
- XUE L W, LIU G P, HU W SH. Rapid prototyping development system for networked multi-agent systems considering switchable topologies[J]. Modern Manufacturing Engineering, 2024(10):41-48.
- [4] 陈爽, 尚和平. 一种码垛工艺方法及仿真实验平台设计[J]. 现代制造工程, 2019(9):54-61.
- CHEN SH, SHANG H P. A palletizing process and design of simulation experiment platform[J]. Modern Manufacturing Engineering, 2019(9):54-61.
- [5] RUBIO E, SANTANA I, ROHTEN J, et al. Remote laboratories for control education: Experience to classic PID control course [C]. 2018 IEEE International Conference on Automation/XXIII Congress of the Chilean Association of Automatic Control(ICA-ACCA). IEEE, 2018: 1-6.
- [6] PABLO O, GARCIA Z J, RODRIGUEZ G L, et al. The weblab-deusto remote laboratory management system architecture: Achieving scalability, interoperability, and federation of remote experimentation[J]. Cyber-Physical Laboratories in Engineering and Science Education, 2018: 17-42.
- [7] GALAN D, CHAOS D, DE L T L, et al. Customized online laboratory experiments: A general tool and its application to the furuta inverted pendulum[J]. IEEE Control Systems Magazine, 2019, 39(5): 75-87.
- [8] 夏志强, 胡文山, 刘国平, 等. 远程实验室仿真与算法生成的一体化软件设计[J]. 实验室研究与探索, 2023, 42(11):16-23.
- XIA ZH Q, HU W SH, LIU G P, et al. Integrated software design for remote laboratory simulation and algorithm generation[J]. Research and Exploration in Laboratory, 2023, 42(11): 16-23.
- [9] LEI ZH CH, ZHOU H, HU W SH, et al. Unified and flexible online experimental framework for control engineering education [J]. IEEE Transactions on Industrial Informatics, 2022, 69(1): 835-844.
- [10] 都业龙, 时维国. 一种变采样周期的网络控制系统控制与调度协同设计新方法[J]. 电子测量技术, 2022, 45(12): 31-39.
- DU Y L, SHI W G. A new co-design method of control and scheduling for networked control systems with variable sampling period [J]. Electronic Measurement Technology, 2022, 45(12): 31-39.
- [11] 孟庆富, 涂程旭, 杨洪波, 等. 微秒级温度传感器动态校准研究进展[J]. 电子测量技术, 2023, 46(24): 54-60.
- MENG Q F, TU CH X, YANG H B, et al. Progress on dynamic calibration of microsecond temperature sensors [J]. Electronic Measurement Technology, 2023, 46(24): 54-60.
- [12] 吴佳桓, 刘晨阳, 陈天森, 等. 基于物联网的电动自行车充电过程监测系统[J]. 电子测量技术, 2023, 46(24): 127-132.
- WU J H, LIU CH Y, CHEN T M, et al. Electric bicycle charging process monitoring system based on IOT[J]. Electronic Measurement Technology, 2023, 46(24): 127-132.
- [13] 胡小林, 张艳荣. 基于 STM32 和 CPLD 的永磁同步电机控制系统设计[J]. 电子测量技术, 2018, 41(19): 136-140.
- HU X L, ZHANG Y R. Design of permanent magnet

- synchronous motor control system based on STM32 and CPLD[J]. Electronic Measurement Technology, 2018, 41(19): 136-140.
- [14] 安锦锦. 永磁同步电机无位置传感器反步超螺旋滑模控制方法[J]. 现代制造工程, 2024(10): 138-145.  
AN J J. Position sensorless back-stepping hyperspiral sliding mode control method of permanent magnet synchronous motor [J]. Modern Manufacturing Engineering, 2024(10): 138-145.
- [15] 周浩, 万旺根. 边缘计算系统的任务调度策略[J]. 电子测量技术, 2020, 43(9): 99-103.  
ZHOU H, WAN W G. Task scheduling strategy of edge computing system[J]. Electronic Measurement Technology, 2020, 43(9): 99-103.

### 作者简介

周杏伟,博士研究生,主要研究方向为远程网络实验室、网络化控制。

E-mail:xingwei.zhou@whu.edu.cn

胡文山(通信作者),教授,博士研究生导师,主要研究方向为网络控制系统、远程网络实验室和智能电网。

E-mail:wenshan.hu@whu.edu.cn

刘国平,教授,博士研究生导师,主要研究方向为网络化控制、多智能体系统、非线性系统辨识与控制、多目标最优控制与决策理论以及工业先进控制和物联网系统控制等应用技术。

E-mail:gp.liu@sust.edu.cn

雷忠诚,副研究员,主要研究方向为网络化控制系统、远程和虚拟实验系统、数字孪生控制系统和电力系统。

E-mail:zhongcheng.lei@whu.edu.cn