

机电一体化系统中的软件开发与优化策略研究

郭鑫 吴兴国*

杭州伏特动力科技有限公司 浙江杭州 310000

摘要: 随着智能制造和工业自动化的不断推进,机电一体化系统作为现代工业设备的核心控制体系,发挥着至关重要的作用。本文围绕机电一体化系统中的软件开发与优化策略展开研究,旨在提高系统的运行效率、稳定性及智能化水平。针对当前软件开发中存在的开发周期长、系统冗余、响应不及时等问题,提出了基于模块化设计、嵌入式实时操作系统、智能算法引入等优化策略,从而为机电一体化系统的软件智能化开发和工业升级提供了理论依据和实践参考。

关键词: 机电一体化; 软件开发; 模块化设计

序言

机电一体化系统是将机械工程、电子技术、控制工程、计算机科学等多学科集成于一体的先进技术系统,广泛应用于数控机床、自动化生产线、智能机器人等现代工业领域。随着工业4.0和智能制造理念的深入发展,传统的控制系统已经难以满足复杂环境下的高效、智能、实时控制需求,软件系统作为机电一体化系统中的“大脑”,其开发水平直接关系到整个系统的性能与可靠性。因此,如何在保证系统稳定运行的基础上,通过优化软件架构和提升算法智能,实现更高效的控制逻辑与更便捷的人机交互,成为当前研究的重点。

一、机电一体化系统概述

(一) 硬件组成

机电一体化系统的硬件部分主要包括执行机构、传感器、控制器、动力装置以及通讯接口等模块,这些硬件之间协同配合,共同完成复杂的自动化控制任务。执行机构是系统中实际完成动作或操作的部件,如电机、液压缸、气缸等,需根据控制指令完成精准运动^[1]。传感器则负责采集位移、速度、温度、压力等信息,为系

统提供实时数据支持。控制器一般采用可编程逻辑控制器(PLC)、单片机或嵌入式处理器等,通过对传感器数据的分析与运算,发出控制信号^[2]。动力装置是系统运行的能量来源,包括电源、液压源和气源等;而通讯接口如CAN总线、Modbus、EtherCAT等,实现了不同模块之间以及与上位机之间的数据通信^[3]。一个完整的机电一体化系统需要硬件组件在结构布局、信号连接、电磁兼容性等方面进行合理设计,以确保系统的高效性与可靠性^[4]。在软硬件深度融合的背景下,硬件设计需兼顾软件运行要求,为实现智能化控制奠定坚实基础^[5]。

(二) 软件系统

机电一体化系统的软件系统作为整个系统的大脑和神经中枢,主要由控制逻辑程序、人机交互界面、实时数据处理模块、通讯接口程序以及诊断与维护模块等组成。控制逻辑程序基于具体工艺流程与任务需求,控制各类执行元件的动作与协作,通常采用梯形图、功能块、结构化文本等语言在PLC或嵌入式平台上编写;人机交互界面通过图形化的方式展示系统运行状态和数据,提升操作便捷性与安全性;实时数据处理模块用于采集并处理来自传感器的大量信息,实现故障判断、状态评估及预测控制,图1为设备状态监测系统的总体架构。通讯接口程序保证系统与上位机或其他控制节点之间的信息互联互通;而诊断与维护模块则用于系统自检、报警提示及远程运维。随着系统复杂度的提高,软件架构日益趋向模块化、可重构和智能化,常借助实时操作系统、人工智能算法以及大数据技术提升其鲁棒性与响应速度。优质的软件系统不仅决定了系统控制的精

作者简介:

郭鑫(1991.6-)男,蒙古族,籍贯:吉林省前郭尔罗斯蒙古族自治县,硕士研究生,研究方向为软件开发、机电结合。

吴兴国(1980.8-)男,汉族,籍贯:江苏省徐州市,本科,研究方向为软件开发、机电结合。

度和稳定性，更是未来机电一体化系统实现智能制造和自主运行的关键。

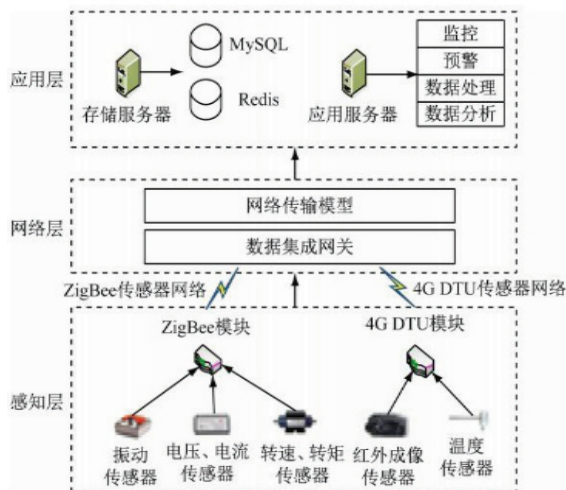


图1 设备状态监测系统的总体架构

二、机电一体化系统中的软件开发需求与关键技术

(一) 软件架构设计要点

在机电一体化系统中，软件架构设计是实现系统稳定性、可扩展性与高效协同控制的关键环节。良好的软件架构需充分结合机电系统的复杂性、多模块协同特性和实时性需求，采用模块化、分层式结构是目前主流的设计思路。模块化设计有助于将复杂系统划分为若干功能单元，如数据采集模块、控制算法模块、人机交互模块和通信模块等，实现功能分离、便于维护和升级。同时，分层结构（如感知层、控制层、执行层）可以提升系统的逻辑清晰度和软件重用率。在实际设计中，实时操作系统常用于保障控制任务的时间确定性；同时需重视软硬件接口设计的标准化，以确保嵌入式控制器、传感器、执行机构等间的数据交互准确高效。

(二) 关键技术分析

机电一体化系统的软件开发涉及多种关键技术的集成与优化，这些技术直接影响系统的运行效率与控制精度，嵌入式系统技术是核心基础，通过搭载高性能微处理器，实现实时数据处理与设备控制，其稳定性与处理能力对整体性能起决定性作用。实时操作系统的引入为多任务调度提供保障，使得传感器采集、运动控制和人机界面等任务可并发执行且响应迅速。通信技术也是关键一环，常见协议包括CAN、Modbus、EtherCAT等，要求高可靠性、低延时，适应多设备间复杂的数据交互场景。随着控制算法的嵌入与优化，如PID控制、自适应控制、模糊控制等，是实现高精度动态控制的核心，特

别是在机器人、数控设备等领域表现尤为重要。人机交互（HMI）技术也逐渐向图形化、智能化方向发展，通过图形界面与触摸交互提升操作体验与系统可视化管理水平。

三、机电一体化软件开发中的常见问题

(一) 程序实时性与稳定性不足

在机电一体化系统中，软件的实时性和稳定性直接关系到整个系统的运行效率与安全性。然而，当前许多机电一体化系统在软件开发过程中普遍存在实时性与稳定性不足的问题。一方面，机电设备对控制指令的响应要求极高，特别是在自动化生产、智能控制等应用场景中，任何延迟都可能导致设备动作失误，影响产品质量甚至引发安全事故。但由于开发人员缺乏嵌入式系统实时控制经验，或者系统架构设计不合理，导致程序响应滞后、调度机制不清晰，不能满足复杂工况下的高频控制要求。另一方面，软件稳定性往往受到多线程通信冲突、内存泄漏、外设接口异常等问题影响，长期运行时易发生系统死机或功能紊乱，给现场维护带来巨大挑战。

(二) 系统更新维护复杂性

随着机电一体化系统的功能逐步增强和智能化水平提升，系统的软件结构也日趋复杂，随之而来的更新维护工作也变得愈加繁琐和困难。许多系统在设计初期缺乏对软件可维护性的充分考虑，模块间耦合度高、结构松散，导致后期修改某一功能模块时容易影响整个系统的稳定性。部分机电设备部署于工业现场环境中，由于更新条件受限、缺乏远程更新机制，需人工现场操作，使得开发人员难以准确追溯系统变更，容易出现更新冲突或功能回退困难等问题。为了提升更新效率与维护便捷性，应在软件架构设计阶段注重模块化与接口标准化，采用面向服务的软件架构（如SOA），配合嵌入式远程升级机制（如OTA），同时引入自动化版本管理工具与维护日志系统，从而简化系统升级流程，降低维护成本，从而有利于提高软件生命周期内的可持续性和可靠性。

(三) 缺乏标准化开发流程与测试机制

当前机电一体化系统软件开发中普遍存在开发流程不规范、测试机制缺失的问题，严重制约了软件的质量和项目的交付效率。一方面，许多开发团队由于缺乏统一的标准化开发流程，开发方式较为随意，文档不完整，代码风格不统一，容易造成项目协作困难和后期维护难度大。特别是在跨专业团队协作的机电一体化项目中，软件、硬件、控制逻辑之间的接口若无明确规范，极易

引发系统集成时的问题。另一方面，测试机制的不完善也成为质量保障的短板。不少项目仅依赖人工功能测试，缺乏系统级的自动化测试、回归测试和压力测试流程，难以全面覆盖异常情况和边界场景，导致产品在实际使用中频繁出现故障。此外，测试数据管理混乱，测试结果缺乏量化指标和可追溯性，也不利于后期优化与问题溯源。

四、软件优化策略研究

(一) 优化系统架构与强化实时任务调度机制

在机电一体化系统中，软件的系统架构直接关系到系统的响应速度与运行稳定性，因此优化系统架构成为提升整体性能的关键路径。首先，应优先采用分层架构与松耦合设计，将控制层、通信层与执行层有效分离，增强系统的扩展性与可维护性。同时，引入面向服务的架构或微服务思想，能进一步将复杂功能拆分为独立模块，实现功能解耦与高效复用。在实时任务调度方面，强化调度机制是确保系统可靠运行的核心。可基于实时操作系统构建任务优先级调度策略，根据任务的重要性与执行时限动态调整资源分配，避免关键任务因资源竞争而延迟。引入优先级反转防止机制与时间片轮转结合抢占式调度算法，有助于确保系统的硬实时性需求，借助多核处理器架构进行任务并行处理，将高频次、低延时任务分配至专用核中执行，从而提升整体任务处理能力。

(二) 构建模块化设计与自动化部署体系

通过模块化设计可将复杂系统拆分为多个独立功能模块，如传感器数据采集模块、控制指令处理模块、人机交互模块等，不仅便于团队协同开发，还能提高代码复用率与系统灵活性。模块之间通过标准接口通信，有利于系统后期的升级与功能扩展。在此基础上，构建自动化部署体系可大大缩短软件发布周期，提高部署效率与可靠性。通过引入CI/CD（持续集成/持续交付）流程，将代码提交、构建、测试、部署等步骤自动化，实现快速迭代和快速交付。例如，可采用Jenkins或GitLab CI工具，配合Docker容器技术和Kubernetes进行部署管理，实现跨平台部署与资源动态调度。在部署过程中，加入自动化测试与回滚机制可确保系统在上线前已通过全面测试，从而能够降低部署风险。

(三) 建立统一开发规范与完善测试体系

统一的软件开发规范是保证机电一体化系统开发质

量与效率的前提条件。制定统一的编码规范、命名规则、文档模板和接口定义规范，不仅能提升团队协同效率，减少沟通成本，还能有效规避因代码风格混乱引发的系统故障。同时，借助版本控制系统（如Git）与代码审查机制，可强化源代码管理，提升代码可追溯性与安全性。在此基础上，建立完善的测试体系尤为重要。测试应贯穿软件开发全流程，包括单元测试、集成测试、系统测试与验收测试等多个阶段。对于实时性要求高的机电一体化系统，还应设置实时性能测试与压力测试，验证其在极端条件下的稳定性与响应能力。引入自动化测试框架（如Google Test、JUnit等）与持续测试机制，可显著提高测试效率与覆盖率，减少人为遗漏。在部署前阶段，应结合硬件环境进行联调测试，确保软件逻辑与硬件执行之间无缝对接。

结语

综上所述，在本文中，深入探讨了机电一体化系统中软件开发与优化的关键策略。通过分析当前系统的需求与技术挑战，结合多种优化方法和工具，提出了一系列切实可行的解决方案。通过探讨性能优化的技术路径，特别是在实时性和数据处理能力要求较高的环境下，如何通过算法优化、并行处理以及硬件加速等手段提高系统效率。针对系统的稳定性与安全性，提出了基于冗余设计、容错机制及加密技术的保障措施，确保系统能够在复杂与不确定的应用环境中稳定运行。

参考文献

- [1] 谢帅. 机电一体化系统设计与优化方法研究[J]. 机械与电子控制工程, 2024, 6(24).
- [2] 乔舒斐, 郝云晓, 权龙, 等. 机电液混合驱动直线执行器构型设计与性能测试[J]. 机械工程学报, 2022, 58(5): 212-222.
- [3] 赵江灵, 徐向阳, 祁宏钟, 等. 串并联式机电耦合系统方案设计及寻优分析[J]. 汽车技术, 2020(10): 5-10.
- [4] 赵志勇, 韩芳. 机电一体化技术的系统设计及在机械设计制造中的应用方向[J]. 造纸装备及材料, 2023, 52(6): 113-115.
- [5] 罗晶豪, 蓝永庭, 李俊明. 电动客车驱动系统机电耦合有限元建模及其响应[J]. 广西科技大学学报, 2023, 34(4): 1-11.