软件定义测发控系统架构分析与设计

田元元　薛建彬　车博山　寇海军
北方自动控制技术研究所　山西　太原　030006

摘　要：软件定义测发控系统架构以软件为核心，实现了对导弹、火箭等装备测试、发射、控制流程的高效管理。本文通过分析软件定义技术的原理与应用，设计了模块化、标准化、可扩展的系统架构，利用分层解耦、中间件技术及资源管理与调度策略，实现软硬件解耦与资源优化。该系统具有任务灵活度高、资源利用率高、系统可靠性强的特点，为测发控领域的技术创新与应用提供了有力支持，展现了显著的优势与应用潜力。

关键词：软件定义测发控系统；架构分析；设计

引言：随着信息技术的飞速发展，软件定义技术逐渐在航空航天领域展现出巨大潜力。软件定义测发控系统作为新一代测发控解决方案，通过以软件为核心，实现对测试、发射、控制流程的灵活定义与高效管理。本文旨在深入分析软件定义测发控系统的架构设计与实现方法，探索其在提升系统灵活性、资源利用率及可靠性方面的优势，为航空航天领域的测发控技术发展提供新的思路与参考，推动相关技术的创新与应用。

1 软件定义测发控系统基础

1.1 软件定义技术概述

（1）软件定义技术的起源与发展：源于信息技术领域对硬件功能灵活扩展的需求，早期在服务器虚拟化技术中萌芽，随着云计算、大数据技术的兴起快速发展。其核心是通过软件逻辑主导硬件资源调度，打破传统硬件功能固化的局限，近年来在工业控制、航空航天等领域加速渗透，成为技术升级的重要方向。（2）相关领域应用：在软件定义网络（SDN）中，通过分离控制平面与数据平面，实现网络拓扑动态调整，广泛应用于数据中心流量管理；软件定义无线电（SDR）则通过软件配置实现多频段、多制式信号处理，在通信、雷达等领域替代传统专用硬件，提升设备适应性与升级效率。

1.2 软件定义测发控系统内涵

（1）定义与特点：软件定义测发控系统是指以软件为核心驱动力，对导弹、火箭等装备的测试、发射、控制流程进行定义与管控的系统。它具备三大显著特点：软硬件解耦，硬件作为通用基础平台，标准化模块通过软件调用实现各类功能；功能可重构，依据不同任务需求，快速灵活地调整测发控逻辑；资源可复用，硬件资源可按需分配，有效降低设备冗余，提高资源利用效率。（2）与传统测发控系统的对比：相较于传统测发控系统，软件定义测发控系统优势明显。在任务准备阶段，能快速通过软件调整配置，大幅缩短准备时间；面对不同任务需求的改造，只需修改软件，成本远低于传统系统更换硬件的方式；且软件的容错机制能更好应对故障。不过，软件定义测发控系统也存在不足，对软件可靠性要求极高，一旦软件出现漏洞或故障，将影响整个系统运行；在复杂场景下，软件调度可能导致实时性受影响；同时，初期软件研发需投入大量人力、物力与时间成本[1]。

2 软件定义测发控系统架构设计

2.1 系统架构设计原则与目标

（1）模块化、标准化、可扩展性。模块化设计将系统拆分为独立功能模块，如测试模块、发射控制模块等，各模块通过标准化接口交互，降低模块间耦合度，便于单独升级与维护；标准化聚焦硬件接口、软件协议及数据格式统一，硬件采用通用总线接口，软件遵循行业通信协议，数据按固定格式传输，保障不同设备兼容；可扩展性通过预留硬件插槽与软件功能接口实现，硬件支持新增传感器、控制器接入，软件能快速集成新测发控算法，满足任务复杂度提升需求。（2）提高资源利用率、任务灵活度和系统可靠性。资源利用率提升依托动态资源分配机制，通过软件调度将硬件资源（如计算单元、存储模块）按需分配给不同任务，避免资源闲置；任务灵活度体现在软件定义任务流程，无需改动硬件，即可通过调整软件参数切换测发控模式，适配导弹、火箭等不同装备测试发射需求；系统可靠性通过硬件冗余设计与软件容错机制实现，关键硬件模块备份，软件具备故障检测与自动恢复功能，减少单点故障对系统的影响。

2.2 分层解耦架构

（1）层次划分。系统采用三层分层解耦架构，自下而上依次为硬件资源层、中间件层、应用层。硬件资源

层是系统基础，中间件层为衔接纽带，应用层直接面向用户需求，三层独立运行又协同工作，实现软硬件彻底解耦。（2）各层次功能与职责。硬件资源层由传感器、控制器、计算设备等物理硬件构成，负责采集测发控过程中的温度、压力等数据，执行控制指令，提供基础计算与存储能力，同时接受中间件层调度；中间件层承担硬件抽象与资源集成职责，通过硬件抽象接口将不同硬件的功能封装为标准化软件服务，屏蔽硬件差异，同时整合分散的硬件资源，形成统一资源池，为应用层提供调用接口；应用层依据实际测发控需求，开发测试管理、发射控制、数据可视化等应用程序，向用户提供具体服务，用户通过应用层操作即可完成装备测试、发射等任务[2]。

2.3 关键技术分析

（1）中间件技术。中间件技术通过构建硬件抽象层与资源集成框架，实现硬件资源的高效管理。硬件抽象层将硬件设备的驱动程序、功能接口封装为通用API，应用层调用API即可控制不同硬件，无需关注硬件细节；资源集成框架通过统一协议将分散的硬件资源接入资源池，实时监控资源状态，动态分配资源，保障硬件资源按需高效利用，是实现软硬件解耦的核心技术支撑。（2）资源管理与调度。资源管理与调度技术依托智能调度算法与资源监控系统，实时采集硬件资源使用情况，结合任务优先级与资源需求，制定最优资源分配方案。例如，高优先级的发射控制任务优先分配计算资源，低优先级的数据分析任务在资源空闲时执行；同时支持任务动态调整，当任务需求变化时，快速重新分配资源，既避免资源浪费，又确保任务高效执行，提升系统任务灵活度[3]。（3）安全与可靠性保障。安全保障方面，采用数据加密传输技术对测发控数据加密，防止数据泄露；通过访问控制机制限制用户操作权限，避免非法操作。可靠性保障则通过硬件故障检测、软件容错与系统备份实现，硬件实时监测运行状态，发现故障立即切换至备份模块；软件采用冗余设计，单一模块故障不影响整体功能；系统定期备份关键数据，故障后可快速恢复，确保测发控过程稳定可靠。

3 软件定义测发控系统实现

3.1 硬件资源集成与抽象

（1）硬件组件的模块化设计与实现。硬件组件采用模块化设计，将传感器（温度、压力传感器等）、控制器（伺服控制器、电源控制器等）、计算单元（嵌入式处理器、FPGA模块）等拆分为独立功能模块。各模块遵循统一机械尺寸与电气接口标准，例如采用标准化插槽式设计，可直接插拔更换；内部电路按功能分区布局，降低模块内部干扰。实现时通过仿真测试验证模块功能独立性，确保单一模块故障不影响其他组件，同时支持按需增减模块，适配不同测发控任务需求。（2）硬件资源的虚拟化与标准化。通过硬件抽象层（HAL）实现资源虚拟化，将物理硬件的功能封装为虚拟资源池，例如将多个传感器的采集功能虚拟为“通用数据采集服务”，隐藏硬件型号差异。标准化方面，制定统一资源描述协议，明确虚拟资源的参数格式（如数据精度、传输速率）与调用接口，确保不同硬件虚拟后可被统一调度。同时建立资源映射表，实时记录物理硬件与虚拟资源的对应关系，保障软件调用虚拟资源时能精准匹配物理设备。

3.2 中间件开发与部署

（1）中间件的功能实现与优化。中间件核心功能包括资源管理、任务调度与通信适配。资源管理模块通过扫描硬件抽象层获取虚拟资源信息，建立资源状态库；任务调度模块基于优先级算法分配资源，优先响应发射控制等关键任务。优化时采用轻量化设计，删减冗余代码降低运行开销，同时引入缓存机制，减少高频资源调用的延迟，通过压力测试验证中间件在高并发场景下的稳定性。（2）中间件与系统其他层次的接口设计。中间件与硬件资源层采用标准化硬件抽象接口，支持即插即用；与应用层通过RESTfulAPI接口交互，应用层可通过API调用中间件提供的资源服务，接口参数采用JSON格式，确保数据传输兼容性。接口设计时加入错误处理机制，当硬件故障或应用层请求异常时，返回明确错误码与提示信息，同时设置接口访问权限控制，防止非法调用，保障数据传输安全。

3.3 应用层设计与实现

（1）应用组件的模块化与动态加载。应用层按功能拆分为测试管理、发射控制、数据可视化等组件，各组件以插件形式开发，支持独立编译与更新。动态加载通过插件管理框架实现，系统启动时自动扫描插件目录，加载所需组件；运行中可通过软件指令卸载旧组件、加载新组件，无需重启系统，例如切换测发控任务时，动态加载对应任务的控制组件，提升系统灵活性。（2）用户界面设计与实现。用户界面采用分层设计，底层为数据交互模块，实时接收中间件传输的测发控数据；上层为可视化界面，通过图表（实时曲线、仪表盘）直观展示参数变化，关键数据（如燃料压力、点火信号）用高亮颜色标注。界面支持自定义布局，用户可按需调整模块位置；同时加入操作权限分级，管理员可配置参数，

普通操作员仅能查看数据，通过用户体验测试优化界面交互逻辑，降低操作复杂度[4]。

3.4 系统集成与测试

（1）各层次模块的集成与调试。采用自下而上的集成方式，先完成硬件资源层与中间件的集成，验证资源虚拟化与调度功能；再集成应用层组件，测试组件调用中间件服务的准确性。调试时使用示波器、逻辑分析仪监测硬件信号，通过日志分析工具追踪软件数据流向，定位模块间的接口冲突或数据传输错误，例如解决应用层与中间件的参数格式不匹配问题，确保各层次协同运行。（2）系统功能与性能测试。功能测试模拟实际测发控场景，验证测试数据采集、发射指令下发、故障报警等功能是否达标，例如测试传感器数据异常时系统是否能自动触发冗余传感器切换。性能测试通过压力测试工具模拟多任务并发场景，监测系统响应时间（要求关键任务响应 ≤ 100ms）、资源利用率（CPU利用率 ≤ 80%）等指标，同时进行可靠性测试，连续运行72小时验证系统稳定性，确保满足实际应用需求。

4 软件定义测发控系统应用与验证

4.1 应用场景分析

（1）航天器发射过程中的应用。在航天器发射中，系统可动态适配不同型号火箭的测发需求。发射前，通过软件快速配置推进剂检测、姿态校准等功能模块，无需更换硬件；发射中，实时调度硬件资源监控箭体温度、振动等参数，若某传感器故障，软件可立即切换至冗余设备，保障数据连续采集；发射后，通过软件回溯测发数据，辅助故障分析，大幅提升发射流程的灵活性与可靠性。（2）测控系统中的功能扩展与升级。传统测控系统升级需更换硬件，成本高、周期长。该系统通过软件即可扩展功能，如新增航天器轨道预测算法、远程操控模块；升级时仅需更新中间件或应用层软件，无需停机，例如为测控系统新增多目标协同监测功能，仅需加载对应应用插件，实现快速迭代，降低升级成本。

4.2 原型系统设计与实现

（1）原型系统的功能与性能要求。功能上需支持航天器基础测发（参数采集、指令下发）、资源动态调度；性能要求任务响应时间 ≤ 150ms，资源利用率 ≥ 85%，连续运行48小时无故障，同时兼容3种以上航天器测发协议。（2）原型系统的设计与实现过程。基于分层架构设计，硬件采用模块化传感器与嵌入式计算单元；中间件开发资源调度与接口适配模块；应用层开发测发控制、数据可视化组件。实现中先完成硬件集成与虚拟化，再开发部署中间件，最后集成应用组件，通过分步调试确保各模块协同，最终搭建出可模拟航天器测发流程的原型系统。

4.3 系统验证与评估

（1）验证方法与指标。采用场景模拟法验证，模拟航天器发射、测控升级场景。指标包括任务灵活性（切换测发任务耗时 ≤ 30分钟）、资源利用率（硬件空闲率 ≤ 15%）、系统可靠性（故障自愈时间 ≤ 10秒）。（2）验证结果与评估。验证显示，任务切换耗时22分钟，资源利用率88%，故障自愈时间8秒，均优于传统系统（任务切换需4小时、资源利用率65%、故障自愈需30分钟）。该系统在灵活性、资源效率、可靠性上显著领先，但复杂场景下实时性仍需优化。

结束语

综上所述，软件定义测发控系统架构以其模块化、标准化和可扩展性的设计理念，实现了软硬件的高效解耦与资源优化，显著提升了测发控流程的灵活性与系统可靠性。通过中间件技术、资源管理与调度策略的实施，该系统在实际应用中展现出强大的任务适应能力与资源利用效率。未来，随着技术的不断进步与应用的深入拓展，软件定义测发控系统将在航空航天领域发挥更加重要的作用，为装备测试、发射与控制提供更加智能、高效的技术支持。

参考文献

[1]王聪,王华茂.基于软件定义的运载火箭测发控系统架构研究[J].计算机测量与控制,2022,30(10):100-102.

[2]李军航天发射场软件定义测发控体系架构研究[J].宇航总体技术,2021,5(4):70-78.

[3]刘磊,丁一波.基于云原生的运载火箭测发控软件架构设计[J].航天控制,2023,41(1):80-87.

[4]郭思岩.面向多型号任务的通用测发控系统设计与实现[J].电子设计工程,2022,30(15):174-176.