

6-28-2024

Design of Real-time Simulation & Test Software Based on Windows/RTX

Yongbo Li

Xi'an Institute of Aerospace Propulsion Technology, Xi'an 710025, China, 765461890@qq.com

Runmei Tian

Xi'an Institute of Aerospace Propulsion Technology, Xi'an 710025, China

Hui Zhang

Xi'an Institute of Aerospace Propulsion Technology, Xi'an 710025, China

Shanpeng Guo

Xi'an Institute of Aerospace Propulsion Technology, Xi'an 710025, China

See next page for additional authors

Follow this and additional works at: <https://dc-china-simulation.researchcommons.org/journal>



Part of the Artificial Intelligence and Robotics Commons, Computer Engineering Commons, Numerical Analysis and Scientific Computing Commons, Operations Research, Systems Engineering and Industrial Engineering Commons, and the Systems Science Commons

This Paper is brought to you for free and open access by Journal of System Simulation. It has been accepted for inclusion in Journal of System Simulation by an authorized editor of Journal of System Simulation. For more information, please contact xtfzxb@126.com.

Design of Real-time Simulation & Test Software Based on Windows/RTX

Abstract

Abstract: Aiming at the limited real-time performance of traditional test software and the low generality of traditional simulation interface software, a real-time simulation software based on Windows/RTX is designed to meet the requirements of unit testing and control system simulation verification of semiphysical simulation software. Through modular design, GUI layer human-computer interface and RTX layer real-time operation program are developed. To ensure the real-time, the lock-free cyclic buffer plus dual-threading technology is used to solve the timeout problem of serial data transmission and reception when the simulation step size is 1 ms under RTX environment. A timeout detection algorithm is proposed to monitor the real-time status of simulation nodes. With cJSON, the test case configuration files are optimized and the users can edit the test cases more flexibly. RTW automatic code generation is used to integrate the projectile model compilation into RTX copycat software engineering. Test results show that the imitation software effectively considers the single test and the simulation, improves the versatility and secondary development efficiency, and reduces the development difficulty.

Keywords

Windows/RTX, real-time, equipment test, hardware in the loop simulation(HILS)

Authors

Yongbo Li, Runmei Tian, Hui Zhang, Shanpeng Guo, and Qi Li

Recommended Citation

Li Yongbo, Tian Runmei, Zhang Hui, et al. Design of Real-time Simulation & Test Software Based on Windows/RTX[J]. Journal of System Simulation, 2024, 36(6): 1468-1474.

基于 Windows/RTX 的实时仿测软件设计

李勇波, 田润梅, 张辉, 郭善鹏, 李琪

(西安航天动力技术研究所, 陕西 西安 710025)

摘要: 针对传统测试软件实时性有限、传统仿真接口软件通用性程度低的问题, 为满足半实物仿真软件需兼顾单元测试和控制系统仿真验证的需求, 设计了基于 Windows/RTX 的实时仿测软件。仿测软件采用模块化的设计原则, 开发了 GUI 层人机交互界面和 RTX 层实时运行程序。为保证实时性, 采用无锁循环缓冲区+多线程技术, 解决了 RTX 环境下仿真步长为 1 ms 时串口数据收发超时问题; 提出一种超时检测算法监测仿真节点的实时状态。借助 cJSON 优化了测试用例配置文件, 用户可以更灵活地编辑测试用例, 利用 RTW 自动代码生成将弹体模型编译集成到 RTX 仿测软件工程。实验结果表明: 该仿测软件有效兼顾了单测与仿真, 提高了仿测软件的通用性和二次开发效率, 降低了开发难度。

关键词: Windows/RTX; 实时; 部件单测; 半实物仿真

中图分类号: TP391.9 文献标志码: A 文章编号: 1004-731X(2024)06-1468-07

DOI: 10.16182/j.issn1004731x.joss.23-0120

引用格式: 李勇波, 田润梅, 张辉, 等. 基于 Windows/RTX 的实时仿测软件设计[J]. 系统仿真学报, 2024, 36(6): 1468-1474.

Reference format: Li Yongbo, Tian Runmei, Zhang Hui, et al. Design of Real-time Simulation & Test Software Based on Windows/RTX[J]. Journal of System Simulation, 2024, 36(6): 1468-1474.

Design of Real-time Simulation & Test Software Based on Windows/RTX

Li Yongbo, Tian Runmei, Zhang Hui, Guo Shanpeng, Li Qi

(Xi'an Institute of Aerospace Propulsion Technology, Xi'an 710025, China)

Abstract: Aiming at the limited real-time performance of traditional test software and the low generality of traditional simulation interface software, a real-time simulation software based on Windows/RTX is designed to meet the requirements of unit testing and control system simulation verification of semi-physical simulation software. Through modular design, *GUI layer human-computer interface and RTX layer real-time operation program* are developed. To ensure the real-time, *the lock-free cyclic buffer plus dual-threading technology* is used to solve the timeout problem of serial data transmission and reception when the simulation step size is 1 ms under RTX environment. *A timeout detection algorithm* is proposed to monitor the real-time status of simulation nodes. With cJSON, the test case configuration files are optimized and the users can edit the test cases more flexibly. RTW automatic code generation is used to integrate the projectile model compilation into RTX copycat software engineering. Test results show that the imitation software effectively considers the single test and the simulation, improves the versatility and secondary development efficiency, and reduces the development difficulty.

Keywords: Windows/RTX; real-time; equipment test; hardware in the loop simulation(HILS)

收稿日期: 2023-02-10 修回日期: 2023-04-22

第一作者: 李勇波(1995-), 男, 工程师, 硕士, 研究方向为火箭/导弹半实物仿真。E-mail: 765461890@qq.com

0 引言

随着精确制导武器的快速发展, 弹上部件单测和半实物仿真作为火箭飞行试验前必不可少的环节, 显得尤为重要。传统的导弹仿测软件存在测试体系单一且通用化程度不高的缺点^[1], 仿测软件应具有严格的实时响应能力、较强的通用性和良好的交互方式^[2-4]。随着承担项目的增多, 如何将部件单测与半实物仿真有效结合, 减少重复开发, 高效完成仿测任务成为一个亟待解决的问题。

目前, 实时仿真测试系统主要有两种, 一种是基于 PCI104 嵌入式平台, 运行 C 代码模拟测发控软件。另一种采用上下位机模式, 上位机运行 Windows 系统, 下位机运行 QNX 等实时操作系统。常晓飞等^[5]利用 RTW 完成了小型化便携式外场实时仿真测试系统的研制。王承军等^[6]采用 LabVIEW 程序搭建了软件架构, 实现了飞行控制系统实物仿真测试、产品综合仿真测试等功能。

与纯数学仿真不同, 有实物在环的测试与仿真要求仿真时钟必须与真实物理时钟严格同步^[6], 否则会出现数据解析超时引起错乱, 仿测实验无法正常进行。Windows 系统是非实时系统, 无法满足仿测软件的实时性要求, RTX^[7]是目前 Windows 平台唯一基于软件的硬实时扩展子系统,

具有开发周期短、成本低等优点。RTX 通过在硬件抽象层(HAL 层)增加实时 HAL 扩展实现抢占式实时任务的管理和调度, 克服了 Windows 系统实时性差的不足, 解决了仿测软件实时与非实时任务兼容处理难的问题。

本文基于 Windows/RTX 完成仿测软件设计, 以一套软件实现了部件单测与半实物仿真, 旨在提高仿测软件通用性、加快仿测软件开发效率、高效完成部件单体测试和系统半实物仿真实验。

1 仿测软件架构

实时仿测软件的重要功能是模拟测发控, 利用 Simulink 难以处理 RS422、CAN 总线等交互式信息流, 不适合模拟测发控等握手式界面, 而仅依靠 Windows 又不能满足实时性需求。为兼顾单测与仿真, 仿测软件的设计采用分层架构的模式, 以模块化的设计原则完成软件架构, 如图 1 所示。

仿测软件由硬件层、中间层、应用层 3 部分组成, 仅硬件层与硬件平台直接相关, 对上 2 层实现了硬件平台差异的隔离, 保证了多平台下软硬件的良好集成。中间层与硬件层几乎不需要更改, 后续主要开发工作应用层, 应用层的仿真和测试软件的部分功能函数是通用的, 且复用性强, 提高了软件的二次开发效率。

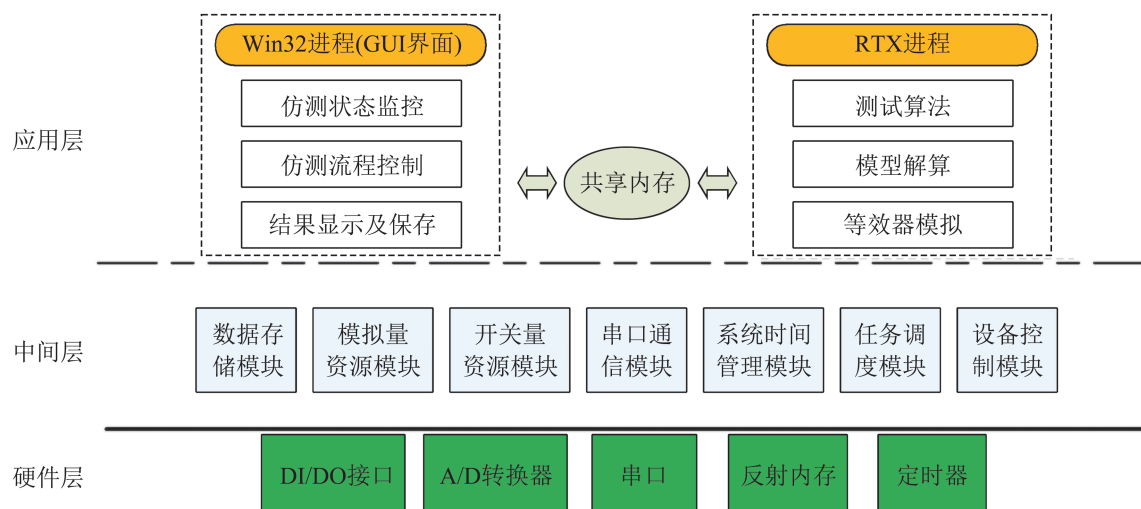


图1 仿测软件架构图

Fig. 1 Structure of simulation & test software

<http://www.china-simulation.com>

• 1469 •

1.1 硬件层

硬件层由各类硬件资源驱动模块组成，位于软件架构的最底层，由中间层调度运行，完成与硬件的实际交互操作，实现软硬件的信息交互。硬件层与硬件平台直接相关，使系统具备使用基本硬件资源的能力。采用面向对象技术，用 C 语言封装了 DI/O 接口、A/D 转换器、串口、反射内存等驱动，反射内存中的本地内存和 VMIC 分别支持多进程和多设备；DAQ 支持 6 种不同型号 PCI 扩展卡的操作；串口支持主机和 Moxa 卡操作。封装的驱动支持不同硬件功能扩展卡，解决了二次开发软件与不同 PCI 扩展卡的适配问题。

1.2 中间层

中间层位于应用层与硬件层之间，对系统资源通用部分进行抽离整合，实现了应用层与底层驱动的隔离，包括模拟量资源模块、开关量资源模块、串口通信模块、系统时间管理模块、任务调度模块、设备控制模块、数据存储模块等。为满足系统对同类型资源多次使用的需求，中间层对同类型资源进行统一管理，简化应用层的调用接口，使整个程序结构清晰，封装了任务调度、I/O 访问、数据存储、设备控制等实现细节，为仿测软件的开发提供了简单易用的通用功能。

1.3 应用层

应用层是分层结构的最顶层，按照具体型号功能需求设计，当需求有变化时，仅需修改应用层相应函数即可。应用层主要由上层人机交互界面(GUI层)和下层实时测控程序(RTX层)组成，如图 2 所示。二者通过共享内存可以实现简单和标准的通信和同步，在保证软件实时性的同时减少了耦合，降低了实现复杂度。

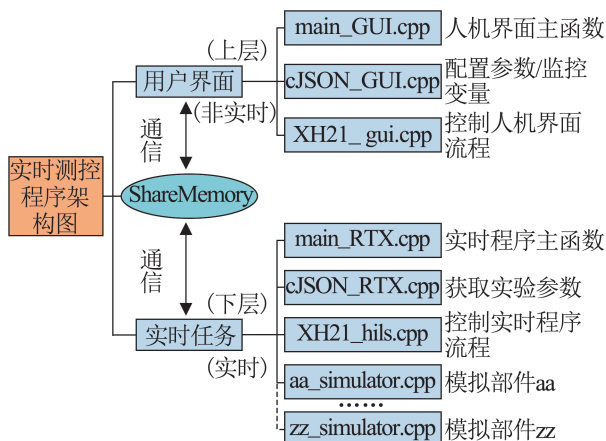
上层基于 QT 开发，是用户实际进行操作的界面，用户可在界面中配置参数并操作流程。下层基于 C 语言开发，由相关所需功能函数组成，主

要实现实验流程控制、等效器模拟、数据解析与记录、被测部件上下电等。

仿测软件启动后，应用层各模块按照仿测大纲中规定的系统工作流程有序运行，进而带动中间层、硬件层运行，实现仿测软件与参试设备的数据交互。在完成硬件层、中间层、应用层的开发后，遵循测试驱动开发思想，同步设计了 200 多个舵机、惯导、速陀的测试用例，对 API 函数的功能和性能进行了全面测试和优化，解决了以往软件运行不稳定、可靠性差等问题。



(a) GUI层界面



(b) 实时测控程序架构图

图2 实时仿测软件应用层

Fig. 2 Application of real-time simulation & test software

2 仿测软件关键技术

仿测软件的实时性直接影响仿测结果的置信度, 要保障实时性单步仿真耗时必须小于仿真步长(本文取 1 ms)。对于部件单测, 还需保障仿测软件与参试部件信息交互过程中串口数据解析处理的实时性; 对于半实物仿真, 还需保障仿真系统主节点与各工作节点的时钟一致。为了提高用户测试效率, 还需对测试用例进行配置。此外, 借助自动代码生成技术简化了模型编写方式, 将弹体模型与仿真软件运行的环境有效地匹配起来。

2.1 实时性保障

2.1.1 数据收发实时性

在进行部件单元测试和半实物仿真实验时, 遥测下传及仿真注入的数据量多达 300 多字节, 需要在一个仿真步长内更新一次, 由于仿真步长较小, 容易造成线程拥堵而引起超时。为了解决超时问题, 保证在单个仿真步长内串口数据收发的实时性, 采用无锁循环缓冲区+双线程的方案。

无锁循环缓冲区的作用是缓存串口收发的数据, 同时缓存写入的数据文件, 防止多线程进行时数据文件的丢失。双线程^[8]即只有一个线程读, 一个线程写, 不加互斥锁, 读线程维护读指针, 写线程维护写指针, 保持数据一致, 其中一个数据被用掉后, 其余数据元素不需要移动其存储位置, 从而减少拷贝, 提高数据处理效率。无锁循环缓冲区+双线程技术解决了 RTX 运行环境下仿真耗时小于 1 ms 时串口数据的发送、接收和解析, 以及大量实验数据存储时产生的超时问题。

2.1.2 仿真节点超时检测

针对仿真节点的超时问题, 提出了一种超时检测算法, 分为相对超时和绝对超时。记当前系统时钟为 SysClock, 当前节点时钟为 NodeClock, 当二者相对误差 $> 0.5\%$ 时, 为相对超时, 输出节点 ID; 记上一帧系统时钟为 LastSysClock, 上一帧节点时钟为 LastNodeClock, 若当前系统时钟与上一帧时

钟差 10 ms 以上且系统时钟与节点时钟之差减去前一帧两者之差的绝对值大于 3 ms 时, 为绝对超时。若存在超时工作节点, 则输出其 ID 号。该算法对仿真设备运行时钟的瞬时误差和累计误差进行监测和区分, 提高了仿真系统的时钟同步检测能力。

2.2 测试用例配置

传统仿测软件通过定义测试用例的结构体, 将变量和其对应的地址、类型、单位、注释等信息绑定起来, 通过地址访问变量。这种方式耦合较深, 变量信息是在实时程序里写定的, 用户难以修改。为了让用户能更灵活地编辑修改测试用例, 本文基于 cJSON 修改统一封装配置文件, 生成运行在 RTX 环境的文件, 包含变量名、地址等, 只需访问变量名即可获取地址等信息, 方便了用户自行编辑测试用例。在界面加载 .case 文件可直接读取测试用例, 将测试数据以图形化界面显示出来, 减轻了用户的工作量和出错率。主要函数有 cJSON_AddItemToObject() 函数, 在 Object 链表后面增加一个 item 结构体, 用来新增测试用例里的变量名, 如实验科目、控制指令等; cJSON_GetObjectItem() 函数, 查找 Object 链表里的 item 值, 用来获取测试用例里的变量。

2.3 自动代码生成

传统的仿真测试系统建模主要采用 C/C++ 语言编写, 导弹模型较为复杂, 开发周期长, 且与控制系统设计团队使用 Simulink 建模的现状不匹配。本文利用 RTW^[9] 将 Simulink 弹体模型转化为可在 RTX 环境下运行的实时代码, 兼顾了实时性与通用性, 也减少了设计师的工作量和沟通成本。

RTW 是一个基于 Simulink 的代码自动生成环境, 它能直接将 Simulink 模型转化为可移植的个性化代码, 并根据目标来配置自动生成多种环境下的程序。自动代码生成的过程如图 3 所示, 先通过 Simulink 快速建立弹体模型并验证, 然后利用 RTW 生成标准 C 代码, 再将 C 代码源文件集成

至 RTX 仿测软件工程，经编译、链接生成可在 RTX 环境下运行的 RTSS 程序。

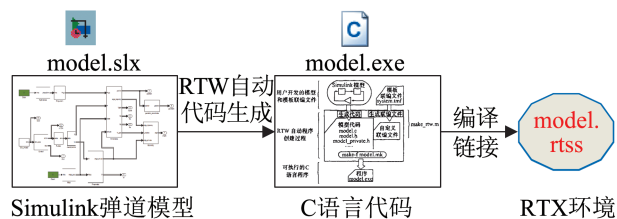


图3 自动代码生成过程
Fig. 3 Process of automatic code generation

3 部件单测与半实物仿真

3.1 实时性能测试

在半实物仿真实验中，对该仿测软件的实时性进行测试，仿测软件的实时性能参数^[10]主要由单步仿真耗时和仿真步长表征，测试结果如图 4 所示，该仿测软件与传统仿测软件的实时性能参数对比如表 1 所示。

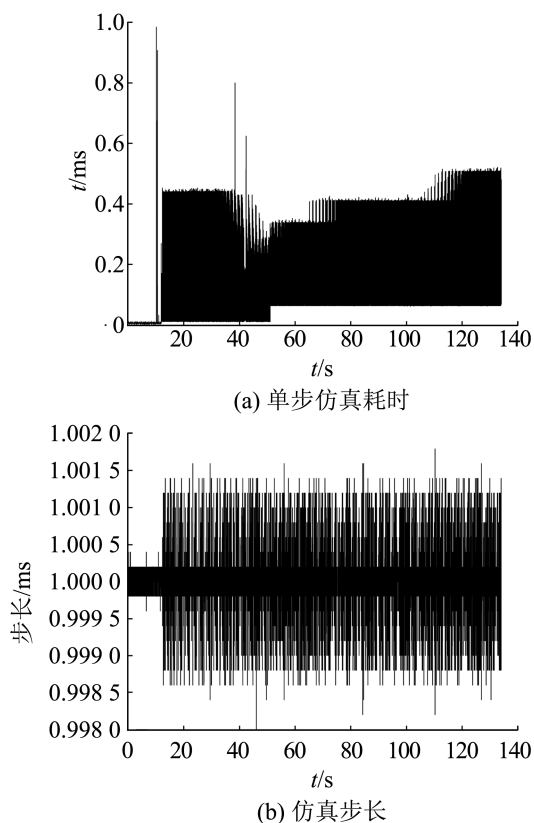


图 4 实时性能参数
Fig. 4 Parameters of real-time performance

表 1 两种仿测软件实时性能参数对比

Table 1 Comparison of real-time performance parameters of two kinds of simulation software

指标	传统仿测软件	新仿测软件
单步耗时平均值/ms	0.49	0.29
仿真步长平均误差/ μ s	0.43	0.28

通过图 4 及表 1 可知，对仿测软件进行超时检测算法优化后，单步仿真耗时平均值为 0.29 ms，仿真步长稳定在 1 ms，平均误差为 0.28 μ s，表明该仿测软件可满足 RTX 环境下仿真步长为 1 ms 的单测和实验实时性要求。

3.2 单体测试与半实物仿真试验

利用本文设计的仿测软件分别进行某型号舵机单测和控制系统半实物仿真实验，实验结果如图 5 所示。图 5(a)是某型号舵机的 4 个舵翼的舵指令与反馈的对比，图 5(b)是某型号半实物仿真三通道姿态角的数学仿真结果与半实物仿真结果的对比。

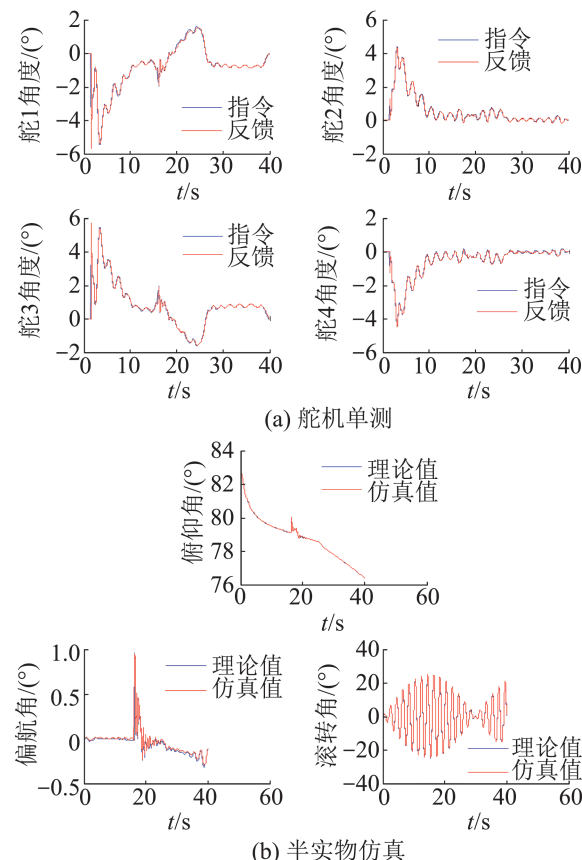


图 5 单体测试及半实物仿真结果
Fig. 5 Result of unit test & HILS

通过图5可知, 舵指令和舵反馈曲线表明舵机能很好地响应制导组合体发出的舵指令信号, 舵机单体性能良好。三通道姿态角的曲线对比结果表明半实物仿真流程顺利完成, 控制系统各部件功能正常, 控制精度满足总体要求。仿测实验结果表明, 该仿测软件为便携测试设备和实时仿真平台提供了通用、稳定、可靠的软件方案, 提高了软件通用性, 且实时性强、成本低, 相比传统仿测软件, 该仿测软件模块库的通用性和集成度有所提升, 也提高了仿测软件二次开发效率。

4 结论

为解决传统测试软件实时性有限、传统仿真软件通用性不足的问题, 高效完成部件单测和控制系统半实物仿真实验, 本文设计了一种实时仿测软件, 主要完成工作为以下三部分:

(1) 设计了一种基于 Windows/RTX 分层结构的仿真测试软件, 在保证测试软件实时性的同时增强了仿真软件的通用性, 降低了实现复杂度, 优化了测试用例配置, 增加了用户使用的灵活度。

(2) 采用无锁循环缓冲区+多线程技术, 解决了 RTX 环境下仿真步长小于 1 ms 的串口数据收发超时问题; 提出一种超时检测算法, 保障了仿真主节点与工作节点的时钟一致性。

(3) 基于该仿测软件完成了某型号舵机单测和控制系统半实物仿真实验, 实验结果表明该仿测软件可以满足实时仿测的实时性和可靠性要求, 提高了仿测软件二次开发效率和弹箭实验验证的可靠性和系统性。

本文设计的基于 Windows/RTX 的仿测软件有效兼顾了单元测试与控制系统半实物仿真验证, 为导弹/火箭的设计开发与测试交付提供了高效率平台, 仿测软件满足项目需求, 对降低武器系统实验成本有着一定作用。

参考文献:

[1] 邵云峰, 彭涛. 地空导弹综合测试技术发展及展望[J].

现代防御技术, 2012, 40(5): 1-7.

Shao Yunfeng, Peng Tao. Development and Prospect of Integrated Test Technology for Surface to Air Missile[J]. Modern Defence Technology, 2012, 40(5): 1-7.

[2] 张尊. 某型空空导弹飞行控制组件综合测试系统设计[J]. 河南科技, 2019(7): 28-29.

Zhang Zun. Design of an Integrated Test System for Flight Control Component of Certain Air-to-air Missile[J]. Henan Science and Technology, 2019(7): 28-29.

[3] 谢侃, 吉亚平. 某导弹自动驾驶仪的自动测试系统实现[J]. 自动化仪表, 2017, 38(6): 75-78, 85.

Xie Kan, Ji Yaping. Realization of the Automatic Test System for a Missile Autopilot[J]. Process Automation Instrumentation, 2017, 38(6): 75-78, 85.

[4] 马玉璘, 陈文浩, 刘伟, 等. 长征六号运载火箭动力测发控系统仿真测试平台的设计[J]. 计算机测量与控制, 2017, 25(5): 1-3.

Ma Yulin, Chen Wenhao, Liu Wei, et al. Design of Power Control and Testing Simulation System for CZ-6 Rocket[J]. Computer Measurement & Control, 2017, 25(5): 1-3.

[5] 常晓飞, 李信淦, 符文星. 一种便携式导弹飞控系统外场实时仿真测试系统设计[J]. 航空兵器, 2020, 27(3): 93-97.

Chang Xiaofei, Li Xingan, Fu Wenxing. Design of a Portable Outfield Real-time Simulation Test System for Missile Flight Control System[J]. Aero Weaponry, 2020, 27(3): 93-97.

[6] 王承军, 谢侃, 华伟. 地空导弹飞行控制系统的综合仿真测试装置搭建[J]. 自动化仪表, 2018, 39(10): 88-91, 95.

Wang Chengjun, Xie Kan, Hua Wei. Establishment of Integrated Simulation Test Equipment for Flight Control System of the Surface-to-air Missile[J]. Process Automation Instrumentation, 2018, 39(10): 88-91, 95.

[7] 张培忠, 高坤, 宁金贵. 基于 Simulink 的 RTX 实时环境导弹半实物仿真编程技术研究[J]. 兵器装备工程学报, 2018, 39(11): 1-6.

Zhang Peizhong, Gao Kun, Ning Jingui. Research on the Missile Hardware-in-loop Simulation Programming Technology Based on Real-time Extension of Simulink[J]. Journal of Ordnance Equipment Engineering, 2018, 39(11): 1-6.

[8] 崔佳, 霍鹏飞, 祁克玉. 基于实时性扩展的多线程半实物仿真方法[J]. 探测与控制学报, 2013, 35(3): 12-16.

Cui Jia, Huo Pengfei, Qi Keyu. Multi-threaded Hardware-in-loop Simulation Method Based on RTX[J]. Journal of Detection & Control, 2013, 35(3): 12-16.

[9] 鲁森魁. 基于 RTW 的嵌入式半实物仿真系统的研究与实现[D]. 大连: 大连海事大学, 2016.

Lu Senkui. The Research and Implementation of Embedded Semi-physical Simulation System Based on RTW[D]. Dalian: Dalian Maritime University, 2016.
[10] 田传艳, 胡军照, 刘继奎, 等. 一种低成本的半实物飞行

仿真系统[J]. 系统仿真学报, 2019, 31(6): 1123-1127.
Tian Chuanyan, Hu Junzhao, Liu Jikui, et al. Low-cost Hardware-in-the-loop Flight Simulation System[J]. Journal of System Simulation, 2019, 31(6): 1123-1127.