

基于面向对象的通用控制软件的设计与实现

刘伟, 孔袁莉, 雷斯聪, 郑柏, 李晶晶

(上海航天设备制造总厂有限公司, 上海 200245)

摘要: 控制软件是运载火箭中非常重要的一类软件; 传统模式下, 型号的控制软件是专用的, 型号之间的控制软件存在相似功能重复开发、开发和维护效率低的问题; 针对上诉问题, 采用面向对象分析和设计技术, 设计了一套基于面向对象的通用控制软件, 通过对控制软件的业务逻辑进行建模和分析, 抽象出满足特定规则的命令实体类, 使得命令实体间可以互相组合, 用户可以通过将命令实体进行组合而成为控制流程; 采用 XML 文件持久化控制流程, 提高数据文件的可读性和易更改性; 通过面向对象技术屏蔽细节, 特定需求通过扩展实现, 而无需更改现有的设计; 通过测试分析, 软件的结构设计合理, 扩展性和维护性良好。

关键词: 面向对象分析和设计; 持久化; XML

Design and Implementation of Universal Control Software

LIU Wei, KONG Yuanli, LEI Sicong, ZHENG Bai, LI Jingjing

(Shanghai Aerospace Equipment Manufacturing Co., Ltd., Shanghai 200245, China)

Abstract: Control software is a very important kind of software in Launch Vehicle project. In the traditional mode, the model control software is special, different control software has the similar problems of repeated development, low development and maintenance efficiency. Aiming at the above problems, a set of control software based on oriented object is designed by using object-oriented analysis and design technology. Through modeling and analyzing the business logic of the control software, the command entity classes that meet specific rules are abstracted, so that the command entities can be combined with each other, and the user can form the control process by combining the command entities. XML file persistence control process is adopted to improve the readability and changeability of data files. The details are shielded by the object-oriented technology, and the specific requirements are realized by extension without changing the existing design. Through test and analysis, the software architecture design is reasonable, and the expansibility and maintainability are more effective.

Keywords: object orient analysis & design; persistence; XML

0 引言

控制软件^[1-3]是运载火箭型号各类软件中非常重要的一类软件, 该类软件负责控制流程的执行和调度, 以自动的方式调度各子系统和各单机, 使子系统和各单机按照规定的逻辑在规定的时间内完成规定的任务。由于火箭测试的流程由控制软件来进行发起和判断, 因此控制软件是运载火箭型号中重要程度较高的一类软件。在传统的模式下, 各型号都会有特定

的控制软件, 软件由特定的人员进行研发和维护, 各型号之间的控制软件设计和实现以实现本型号的需求为主, 型号间的需求并不做过多的考虑。该种模式下, 由于不同设计人员采用的设计方案和使用技术的原因, 各型号之间的控制软件即使有相同的功能, 相互之间复用的难度也较大。这种模式下, 型号间相同的功能存在重复开发的情况, 并且在软件的维护上也存在效率不高的情况。

收稿日期: 2021-11-03; 修回日期: 2021-12-15。

作者简介: 刘伟(1986-), 男, 陕西汉中, 硕士研究生, 工程师, 主要从事运载火箭控制测试系统方向的研究。

引用格式: 刘伟, 孔袁莉, 雷斯聪, 等. 基于面向对象的通用控制软件的设计与实现[J]. 计算机测量与控制, 2022, 30(2): 145-150.

本文针对传统模式下,设计和维护效率较低、功能重复开发等缺点,为了提高软件的设计和效率,设计了基于面向对象^[4]的通用^[5-8]控制软件,对控制软件的业务逻辑和业务过程进行了分析,将控制逻辑抽象为可组合的类。用户可以根据需要自由组合各控制逻辑类,并将结果以 XML 的方式保存在本地磁盘。软件会读取并解析 XML 文件,将 XML 文件的内容解析为对应的逻辑控制类,以执行用户配置的控制逻辑。

1 软件结构与功能设计

1.1 软件结构

通用控制软件主要功能是对流程进行控制,需要根据已经和正在执行的控制命令,判断任务的执行是否正确,是否继续执行下一个控制命令。

通过对控制软件的需求进行整理和分析,对现有型号的控制软件的设计和实现方案进行归纳总结,将控制软件依据功能划分为相对独立的 4 个模块:数据源模块、数据解析模块、命令模块、工具模块,功能模块图如图 1 所示^[12]。

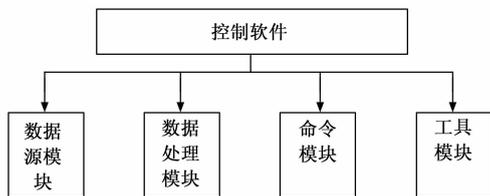


图 1 软件结构图

数据源模块的功能是实现数据传输。通过对具体需求的实现,命令传输可使用 Socket 套接字、串口和 Http 等传输方式^[13]。

数据处理模块的功能是实现数据转换。主要功能有两个:第一是将命令和命令参数转换为数据源可传输的数据形式;第二是将数据源接收到的数据转换为命令或命令的参数。常见的形式是对网络传输(TCP、UDP)字节序列和实体类进行相互转换^[14]。

命令模块是控制软件的核心模块。用户根据实际的功能需求,可以使用现有的命令组合成需要的控制功能,也可以自定义新的命令类,实现当前不具备的功能。命令的关系是松散的,命令之间可以相互组合为新的命令,组合出的命令可以像普通的命令一样进行进一步的组合^[15]。

功能模块是辅助模块。对软件代码中的一些常用

操作进行封装,减少常用功能的重复开发。

1.2 软件功能

通用控制软件由控制命令、命令存储及解析、命令数据传输和命令数据正确性判断 4 部分组成。为了提高控制类软件的通用性和开发、维护的效率,对传统模式下的业务进行进一步分析,将具有相同功能的业务抽象封装为类,将可变的功能抽象为接口,为后续的功能扩展提供接口。通过将软件功能进行模块化,对软件核心功能控制命令进行可配置化处理,提高了软件开发、维护的规范性和效率。

为了验证通用控制软件方案的正确性,需要用户控制命令的执行并将执行的状态向用户反馈。软件在 Visual Studio 2010 平台上采用 C# 语言进行设计和实现,以 Windows 窗体界面风格实现人机交互。

2 软件整体设计

2.1 软件结构

通用控制软件核心原则就是面向对象方法中的“对扩展开发,对修改封闭”原则,是指将控制类软件的相似的功能进行整合,用户的可变需求通过扩展实现,而不需要更改原有逻辑,通过软件的可扩展性保证软件的灵活性和稳定性^[16]。

在通用控制软件的设计过程中,不仅需要考虑到软件的扩展性和可维护性,而且还需要考虑到对现有型号的专用控制软件的兼容性,以确保设计的软件能够更好地适应型号的需求^[17]。通过对软件的需求进行分析,对现有型号的专用控制软件的设计方案和使用场景进行归纳总结,通过使用面向对象分析与设计技术,将当前涉及到的控制操作转换为软件代码中的类,目前包括 1 个父类和 10 个实现类,父类参考建造者设计模式,通过模板类的形式为子类提供统一的业务逻辑的执行顺序,实现类根据需求实现特定的操作,实现类之间可以以积木的形式组合成新的命令,完成复杂的控制逻辑。软件类图如图 2 所示。

2.2 软件工作流程

通用流程控制软件的执行流程如图 3 所示。

为了提高软件维护的灵活性和便利性,控制流程被保存在本地磁盘上的 XML 文件,用户通过修改该 XML 文件即可达到更改控制逻辑的目的。

软件工作的具体流程如下:软件启动后,软件会

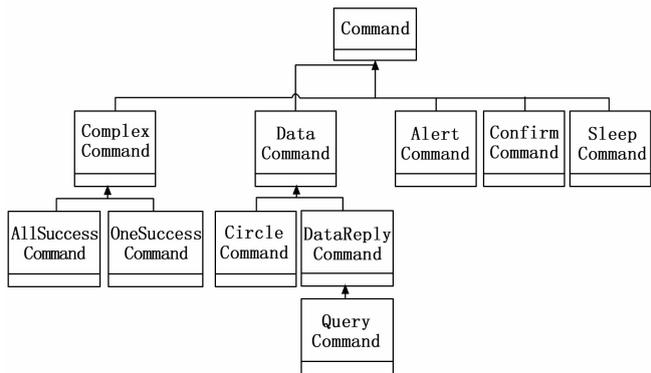


图 2 软件类图

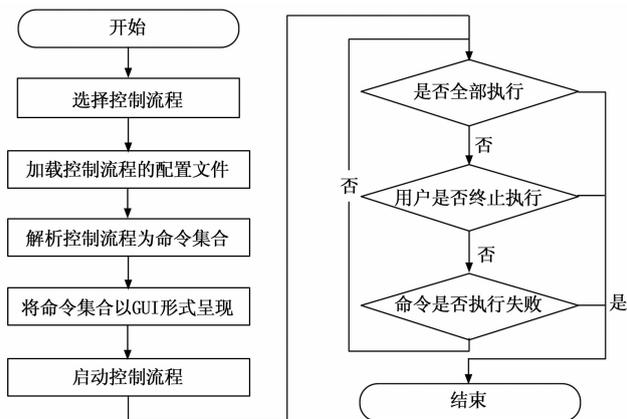


图 3 软件执行流程图

根据用户配置的保存控制流程的 XML 文件的路径, 自动加载该 XML 文件并解析为控制命令。在成功解析到控制命令后, 软件会将控制命令以人机交互界面的形式呈现, 便于用户使用^[18]。

3 软件设计关键技术

3.1 基于面向对象的业务抽象

在通用控制软件的设计时, 通过对现有型号的控制软件的业务功能进行分析, 发现现有型号的控制软件在业务功能上和流程控制逻辑上有相似的部分, 这一部分属于冗余的重复开发的部分。

面向对象分析与设计 (OOAD, object orient analysis & design) 的优点就是能够构建与现实世界相对应的问题模型, 并保持他们的结构、关系和行为为模式, 通过对业务逻辑的合理的分析和设计, 有效的避免重复的部分^[19]。

通过面向对象分析和设计技术对软件的业务逻辑进行抽象和分析, 将业务逻辑中需求的实体提取成

类, 功能提取成方法。各个类的含义如下:

Command 类: 命令类的基类, 定义命令的基本属性。

ComplexCommand 类: 命令的一种实现, 是一组命令的集合, 将多个命令组合成一个命令, 完成命令的执行和执行成功、失败的判断。

AllSuccessCommand 类: 复合命令的一种实现, 是指所包含的所有命令均执行成功, 该命令执行成功, 否则该命令执行失败。

OneSuccessCommand 类: 复合命令的一种实现, 是指所包含的任意一个命令执行成功, 该命令执行成功, 否则该命令执行失败。

DataCommand 类: 命令的一种实现, 数据命令类, 是指该命令需要进行数据交互。

CircleCommand 类: 数据命令的一种实现, 循环数据命令, 是指该命令需要多次进行主动的数据推送, 而不需要数据回馈。

DataReplyCommand 类: 数据命令的一种实现, 数据回馈命令类, 是指该命令需要主动推送数据并需要回馈数据。

DataQueryCommand 类: 数据回馈命令类的一种实现, 数据查询类, 是指该命令需要主动推送数据并对回馈数据进行判断。

AlertCommand 类: 命令的一种实现, 警告命令, 该命令是对用户操作的一种提升, 用户可在控制流程中, 通过弹出警告对话框以实现控制流程的用户确认。

ConfirmCommand 类: 命令的一种实现, 确认命令, 该命令是对用户操作的一种提升, 用户可在控制流程中, 通过弹出确认对话框以实现控制流程的用户选择。

SleepCommand 类: 命令的一种实现, 延时命令, 该命令是用于自定义延时, 用户可根据需要定义延时命令的时机和延时的时间。

通过对业务逻辑进行建模和分析^[9], 将命令抽象成类。同时考虑到用户需求和软件设计的扩展性, 将一些非功能性需求的功能也提升为命令, 如界面中的弹出窗口等, 以提升软件的维护性, 提高软件的维护效率, 降低软件的维护成本^[20]。

在分析出功能性需求和非功能性需求的命令后, 再通过面向对象中的继承和多态技术, 达到屏蔽细节

而实现不同功能的目的。举例而言, AlertCommand 和 SleepCommand 是两个功能完全不同的命令, 但是这两个类均继承于 Command 类, 则这两个类就可以以相同的方式使用。AlertCommand 实现弹出一个警告对话框, 只有在用户选择确认后流程才继续执行, 否则流程一直等待; SleepCommand 则是一个延时命令, 延时会自动执行, 直到延时完成或用户终止流程^[21]。

在上诉的抽象设计下, 不用的命令之间可以相互组合, 多个命令也可以组合成一个命令。组合出的命令又可以和其他命令进一步组合成更复杂的命令。这种组合方式类似“搭积木”的方式, 软件业务中有不同的类, 但是提供相同的“外形”(接口/父类), 那么不同的类就可以在一定的规则下进行自由组合。用户通过这种“搭积木”似的命令组合, 已达到用户期望的控制逻辑的目的。

3.2 XML 序列化

XML (extensible markup language), 可扩展标记语言, 是一种基于文本、跨平台的结构型标记语言, 可用于保存和传输数据, 用户可以非常方便地使用文本编辑工具对 XML 文件进行编辑, 完成测试流程的添加、修改和删除操作。

用户将多个命令按照一定规则组合成测试流程, 测试流程需要持久化^[10-11]后才能永久保存, 否则就只能临时存在于内存中, 软件关闭后数据丢失, 无法再次使用。为了便于用户对测试流程进行维护, 采用 XML 文件作为测试流程持久化的方式, 通过序列化和反序列化的方式高效的实现持久化文本与软件代码的相互转换。

在通用控制软件中, 为了便于管理和维护, 设计了继承“AllSuccessCommand”类的子类“TestProject”作为一个项目, 其他的命令实现类可以以自身或者多个实现类组合成的新命令作为项目命令集合的一个子项, 通过类似“搭积木”的方式将各个实现类组合成一个实现特定要求的测试项目。

举例而言, 新建一个名为“测试项目”的测试流程, 设计项目包括一个名为“测试一”的任意成功命令; “测试一”由一个名为“测试一 1”的任意成功命令和一个名为“测试一 2”的全部成功命令组成; “测试一 1”由一个名为“测试一 11”的选择确认命令和一个名为“测试一 12”的查询命令组成; “测试

一 2”由一个名为“测试一 21”的确认命令和一个名为“测试一 22”的选择确认命令组成。

测试流程结构如图 4 所示。

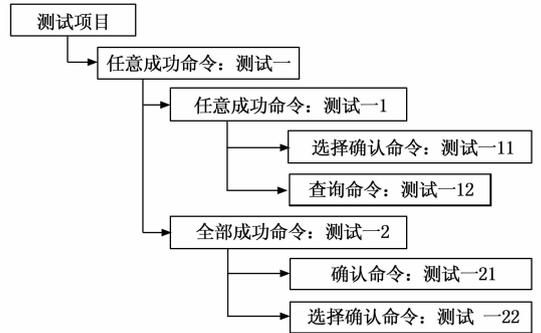


图 4 测试项目结构

测试项目的执行流程如图 5 所示。

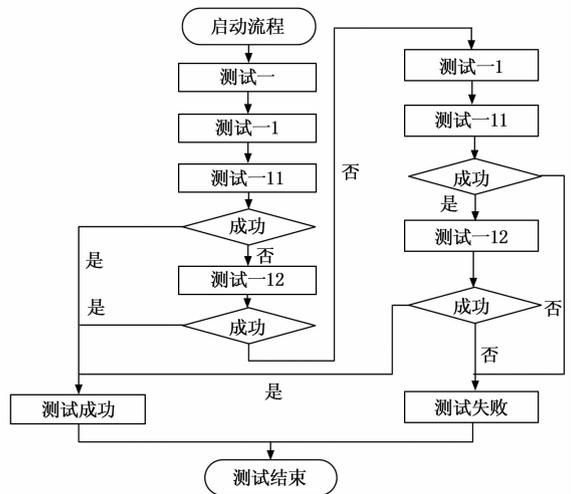


图 5 测试项目执行流程

测试项目持久化为 XML 文件的情况如下所示：

```
<? xml version="1.0"? >
< TestProject xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
<Name>测试流程</Name>
<Comment>临时测试项目</Comment>
<Commands xsi:type="OneSuccessCommand" Caption="测试一" WaitTime="0" Index="1">
<Commands xsi:type="OneSuccessCommand" Caption="测试一 1" WaitTime="0" Index="1">
<Commands xsi:type="ConfirmCommand" Caption="测试一 11" WaitTime="0" />
<Commands xsi:type="DataQueryCommand" Cap-
```

```

tion="测试-12" WaitTime="1000" TimeOut="5000"
  Attempts="1" AttemptInterval="500">
  <Content ZT="37099" GNH="1" BM="1" BZZ="
"0" PCZ="0" />
</Commands>
</Commands>
</Commands>
<Commands xsi:type="AllSuccessCommand" Cap-
tion="测试-2" WaitTime="0" Index="2">
  <Commands xsi:type="AlertCommand" Caption="
"测试-21" WaitTime="0" />
  <Commands xsi:type="ConfirmCommand" Caption
="测试-22" WaitTime="0" />
</Commands>
</Commands>
</TestProject>

```

举例中的测试流程的设计过程, 测试流程的新增或修改完全是在对配置文件的新增或修改的方式上完成的, 不需要进行任何代码的修改, 实现了软件代码设计和维护的分离, 维护人员只需要按照规定的方式进行 XML 文本文件的新增和修改即可实现测试流程的新增和修改。而且在软件设计和实现时按照模块化和插拔式的原则, 使得基础的各种控制命令和组合出的控制命令可以进行无差别的组合, 通过类似“搭积木”的方式, 将各种基础命令和组合命令组合成具有更复杂功能的命令。

4 测试结果与分析

4.1 案例实现

为了验证本文的设计和实现思路的可行性, 软件二次开发和维护的效率。二次开发的过程以实现现有的 XX-6 型号的等效控制软件功能的方式进行。

XX-6 型号等效控制软件的通讯方式采用 TCP 网络的方式, 数据交互对象为一个上位机软件, 根据型号对控制软件的设计要求, 二次开发的过程主要由 3 个步骤组成:

1) 根据数据源模块的接口要求, 设计一个 TCP 网络通讯数据源模块, 完成控制软件与交互上位机软件的 TCP 网络通讯。

2) 根据数据处理模块接口的要求, 依据 XX-6 型号的数据格式, 设计满足型号实际需求的数据处理模块, 完成命令、命令参数与二进制数据的相互转换;

3) 根据命令模块要求, 依据 XX-6 型号控制逻辑要求, 软件提供的基础命令类能够满足功能要求, 只需按照要求新建测试流程的 XML 文件。

在通用控制软件基础功能的基础上, XX-6 型号控制软件的开发过程只需要对差异性的模块进行二次开发, 相同的功能模块无需重复开发, 控制流程的修改可以通过修改文件的方式实现, 开发和维护的效率显著提高。

4.2 结果与不足

在开发过程中, 控制软件相似功能的部分, 如简单命令成功、失败逻辑控制、复合命令成功、失败判断及控制、控制流程持久化等无需重复开发; 特殊需求, 如各型号的数据解析、数据合格性判断等可通过扩展的方式实现, 而不影响原有的业务逻辑。

在维护过程中, 由于采用了统一的设计思路, 测试流程也存储在 XML 文件中, 测试流程的维护的效率也得到了提高, 降低了后期维护的成本。测试结果达到了预期的目的。

但是在开发过程中也发现了一些不足之处, 控制流程的修改是通过对 XML 文件的修改来实现, 但是 XML 文件的编辑是直接手动修改文件, 目前还没有提供可视化的编辑方式, 用户需要熟悉 XML 文件中的每个标签的含义, 编辑时的错误也不容易发现, 后续提供可视化的操作接口, 用户可以在软件中编辑测试流程并自动保存为 XML 文件, 这样可以减少人为的文本编辑错误, 用户也无需了解 XML 文件的标签所代表的含义, 降低测试流程维护的难度, 进一步提高软件维护的效率。

5 结束语

针对传统模式下, 各型号的控制软件相互间通用性低、相似功能重复开发的问题。设计了一套基于面向对象的通用控制软件。对软件业务流程中的功能性需求和非功能性需求进行抽象和分析, 将业务需求中的实体抽象成命令类, 并将部分非功能需求也提升成命令, 使得用户可以通过类似“搭积木”的方式组合各种命令。

通过采用统一的设计思想和通用化设计, 提高软件开发和维护的效率, 节省了软件开发和维护工作的人力成本, 在应对当前的高强度任务方面具有重要意义。

参考文献:

- [1] 韩逸非, 冯娜娜. 引射式跨声速风洞流场控制软件设计 [J]. 计算机测量与控制, 2021, 29 (1): 66-70.
- [2] 曹建. 计算机网络自动检测控制系统软件开发设计 [J]. 数字技术与应用, 2020, 38 (2): 126-127.
- [3] 苏亚娟, 荣音. 一种支持多型号控制板卡的接口软件设计 [J]. 数字技术与应用, 2019, 37 (11): 132-133.
- [4] 陈洸文, 邹昶, 聂国健, 等. 基于面向对象思想的软件系统分析与设计 [J]. 电子产品可靠性与环境试验, 2020, 38 (6): 24-28.
- [5] 周绍磊, 周正. 一种通用自动测试系统软件平台的设计与实现 [J]. 计算机测量与控制, 2003, 11 (7): 525-527.
- [6] 杨文, 黄文涛. 通用自动测试系统的软件设计与实现 [J]. 工业控制计算机, 2012, 25 (1): 79-80, 83.
- [7] 高月芳, 綦羽, 王栋. 通用软件体系架构的设计与应用 [J]. 计算机工程与设计, 2009, 30 (2): 348-350.
- [8] 尹志杰, 童维勤, 支晓莉. 基于 .NET 的通用软件开发平台的研究与实现 [J]. 计算机应用与软件, 2007 (7): 36-38.
- [9] 陈长喜, 郭立红, 唐雪松, 等. 基于 UML 的 C3I 系统软件建模 [J]. 计算机测量与控制, 2006, 14 (9): 1260-1262.
- [10] 秦奕青. 对象持久化常用方法研究 [J]. 北京机械工业学院学报, 2003 (1): 20-24.
- [11] 孙栋海, 宋顺林. 对象/关系映射在 .NET 框架下的应用研究 [J]. 计算机应用研究, 2006 (8): 58-60.
- [12] 王学旗, 马东来. 一种科普展品通用控制系统 [J]. 科学技术与工程, 2020, 20 (23): 9427-9432.
- [13] 于大海, 袁杰, 白亮. 主机控制软件在测发控系统中的应用研究 [J]. 计算机测量与控制, 2019, 27 (4): 102-105.
- [14] 刘素婵. 自动测试软件的通用化设计和实现 [J]. 计算机与网络, 2010 (1): 45-47.
- [15] 黄灿, 卜林森, 王珉, 等. 可快速配置的多轴装配系统控制软件设计 [J]. 机电工程, 2019, 36 (1): 84-89.
- [16] 张悦. 通用软件接口测试平台的设计与实现 [J]. 数字技术与应用, 2019, 37 (12): 145-146.
- [17] 李珊珊, 吴传贵. 基于 FC-AE-ASM 总线的通用分析软件的开发 [J]. 测控技术, 2019, 38 (9): 67-70.
- [18] 申鹏亮, 翟正军, 周健. 基于 ICD 的通用化航电测试软件研究 [J]. 测控技术, 2013, 32 (11): 124-127.
- [19] 周阳明. 基于 Python 语言的自动测试系统通用软件平台实现 [J]. 电子设计工程, 2019, 27 (5): 81-85.
- [20] 夏磊, 刘毅, 周靖宇. 自动测试系统通用测试平台技术研究 [J]. 电子测试, 2017 (6x): 76-77.
- [21] 黄晓晴, 梁晓芬, 郑永龙. 通用 ATS 软件开发平台研究与设计 [J]. 自动化仪表, 2013, 34 (3): 55-58.
- [14] 钱哨, 陈丹. 基于数据分发服务的发布订阅系统 [J]. 电子技术与软件工程, 2017 (6): 153-153.
- [15] 周宝, 杜承烈. 一种内嵌软件传感器的双软总线体系结构研究 [J]. 计算机测量与控制, 2008, 16 (11): 1766-1769.
- [16] 王伟, 杨本朝, 李光松, 等. 异构冗余系统的安全性分析 [J]. 计算机科学, 2018, 45 (9): 183-186, 194.
- [17] COSGROVE J S, ESPOSITO T, CROUSE P. The use of GMSEC architecture for multi-mission flight dynamics orbit and attitude support [C] //AIAA/AAS Astrodynamics Specialist Conference and Exhibit, 2013: 18-21.
- [18] SMITH D, GRUBB T, ESPER J. Linking and combining distributed operations facilities using NASA's "GMSEC" system architecture [C] //Space Ops 2008 Conference, ESA and EUMETSAT in association with AIAA), 2008: 1-12.
- [19] 周晓春, 姜小祥, 周希辰. 基于串行 Rapid IO 的舰载相控阵雷达波束控制设计 [J]. 科学技术与工程, 2014, 14 (13): 62-66.
- [20] 彭树生, 王洋. 车载快速搜索/跟踪卫星的控制系统设计 [J]. 计算机测量与控制, 2005, 13 (9): 932-934.
- [21] 蔡广礼. 卫星移动通信及其天线的发展应用 [J]. 工程技术: 文摘版, 2016 (7): 291-291.
- [22] 侯飞, 柏利, 乔淑君. 数字化相控阵天线测试方法及测试系统设计 [J]. 计算机测量与控制, 2017, 25 (1): 47-49, 53.

~~~~~  
(上接第 136 页)