

小卫星可复用软件技术研究

郝佳宁, 刘畅, 韩延东, 马磊

(航天东方红卫星有限公司, 北京 100094)

摘要: 小卫星应用领域的不断扩展、功能和结构的日趋复杂以及数量的急剧攀升, 使得星载软件的规模和复杂度快速增加, 软件可复用性设计变得愈加重要; 首先分析了面向体系应用的星载软件功能和技术需求, 然后对星载软件体系架构进行了分层模块化设计, 最后针对星务中心计算机软件, 详细描述了星载软件框架的静态结构, 动态结构以及可复用构件的运行模式; 在小卫星向体系化应用发展的趋势下, 分层模块化软件体系架构和软件框架技术将提高星载软件的可复用性, 为实现小卫星快速集成, 快速测试和快速生产打下重要基础。

关键词: 小卫星; 软件体系架构; 软件框架技术; 软件构件

Research on Reusable Onboard Software Technology of Small Satellite

Hao Jianing, Liu Chang, Han Yandong, Ma Lei

(DFH Satellite Co. Ltd., Beijing 100094, China)

Abstract: As the application fields of small satellite are expanding, the functions become more complicated and the number grows rapidly, the scale and complexity of onboard software are increasing continuously. The technology trend makes the reusable design of software more important. This paper firstly analyzes the technology needs of satellite for system-oriented application, and then presents the layered and modular software architecture as well as the related key technologies. At last, the paper describes in detail the static and dynamic structure of the framework as well as the run modes of software components. The layered and modular software architecture and the software framework technology are critical to implement the rapid assembly, integration and test of small satellite.

Keywords: small satellite; software architecture; software framework; software component

0 引言

经过近 30 年的发展, 小卫星在各领域的应用已全方位展开, 包括遥感、通信、导航、空间探测、空间科学研究、技术演示验证等。随着效费比的不断提升, 小卫星逐渐成为全球航天发展热点, 发射数量急剧增长, 在民用、军用和商业航天体系中占据越来越重要的地位。

小卫星应用领域的不断扩展、功能和结构的日趋复杂, 以及数量的急剧攀升, 必然要求星载软件具有与之相适应的软件体系结构和可复用设计模式。针对这种情况, 世界各个航天技术强国均在积极探索, 希望找到低成本、高效率、短周期的卫星软件开发方式。2005 年, 美国空军研究实验室提出了 SPA 体系结构 (Space Plug-and-Play Avionics) 和卫星数据模型, 通过即插即用特性来支持卫星软件快速集成^[1-2]。2010 年, NASA AMES 航天中心提出了 SMART (the Small Multi-purpose Advanced Reconfigurable Technology) 架构以实现低成本可重构小卫星的设计和开发^[3]。欧空局提出了基于时空划分的模块化综合电子软件架构^[4]。洛克希德马丁公司开发了基于 JAVA 的分布式体系结构, 支持面向服务的航天器模型, 通过规范化的

服务来支持上层软件的可复用^[5,6]。德国 Constance 大学和瑞士 ETH Zurich 大学提出了 OBS (On-Board Software) 软件框架和 AOCS (Attitude and Orbit Control System) 软件框架等技术来提高星载软件设计开发的可复用性和可扩展性^[6-7]。在卫星标准化上, 空间数据系统咨询委员会提出了星上互联协议和标准接口服务 (SOIS: spacecraft onboard interface services)^[8]。

本文首先分析了面向体系应用的星载软件功能和技术需求, 然后对星载软件体系架构进行了分层模块化设计, 最后针对星务主机软件, 详细描述了基于软件框架和构件技术的星载软件静态结构, 动态结构以及可复用构件的运行模式。

1 面向体系应用的星载软件功能和技术需求

1.1 功能需求和功能关系界面

面向体系应用, 星载软件应该具有如下功能: 1) 支持虚拟卫星的任务设计迭代过程; 通过卫星数据和服务模型来支撑任务策划阶段的卫星任务仿真、开发和迭代调试过程; 2) 支持星载设备快速开发调试; 通过通用软件框架和卫星数据和服务模型支持星载软件的开发、调试和虚拟联试; 3) 支持整星 AIT 阶段的快速测试; 提供各分系统自测试功能实现平台, 实现各分系统的快速自测试, 配合整星系统测试; 4) 支持卫星发射场阶段的快速任务装订; 可以根据不同优先级的发射预案, 灵活快速的对任务参数进行

收稿日期: 2018-11-05; 修回日期: 2018-12-18。

作者简介: 郝佳宁(1987-), 男, 硕士研究生, 工程师, 主要从事航天器产品保证管理方向的研究。

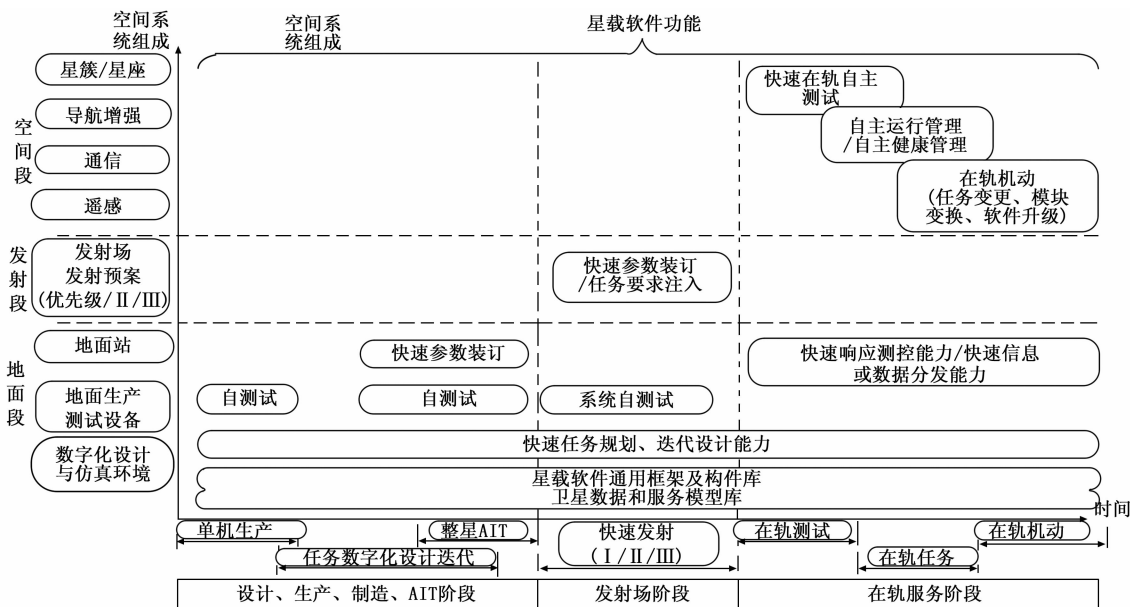


图 1 星载软件在空间系统中的功能关系界面

装订, 如任务轨道、任务位置、数据传输说明、临时地面站等信息; 5) 实现卫星的各项在轨功能。如图 1 所示, 星载软件的功能涉及空间体系中的各个组成部分和各个操作阶段。

1.2 星载软件的技术需求

面向体系应用的星载软件应该具有三个主要技术特点: 软件的可复用性; 软件运行的硬件平台无关性; 软件开发的快速性; 这三个技术特点决定着星载软件的技术需求, 主要包括: 软件框架技术、软件中间件 (即软总线技术) 技术、卫星数据和服务模型技术。

软件框架技术是实现软件可重构, 可复用的关键技术。软件框架通过建立可复用的领域模型, 然后依据领域模型产生出该领域应用软件的体系结构, 以此为基础识别、提炼和开发该领域的构件, 形成构件库, 同时将控制流抽象为软件功能接口, 最后通过例化实现具体的星载软件。软件中间件技术是实现硬件平台无关性的关键途径。中间件软件驻留在多个星载设备节点上, 屏蔽了底层硬件具体信息, 管理连接到这些节点上的设备和服务。卫星数据和服务模型是面向体系应用星载软件快速开发的重要技术。该模型对硬件设备的功能进行抽象描述, 为软件集成提供数据和服务支持。该技术从数据的角度来描述卫星, 将传统意义上卫星各个分立设备的功能用“数据模型”和“服务”来分类描述, 以便于卫星软件的设计、维护和扩展。

2 星载软件体系架构设计

2.1 天上综合电子系统介绍

国内小卫星一般采用分布式设计, 如图 2 所示, 包括星务主机、天上网络、嵌入式管理执行单元 (management executive unit, MEU) 和天上其它设备。天上网包括天上控制网和信息网两种。天上控制网将天上各功能模块有机

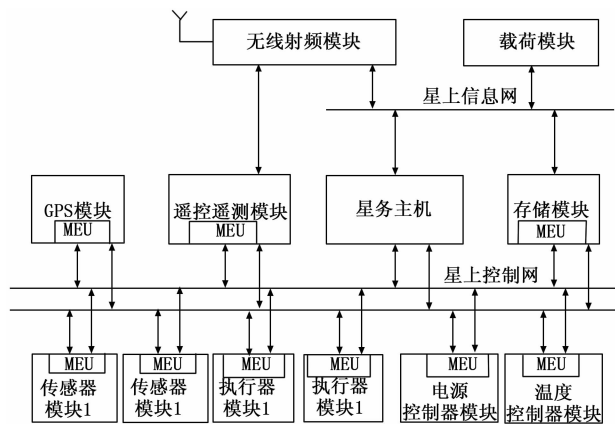


图 2 天上综合电子系统介绍

地连接起来, 实现天上信息交换和共享, 实时地完成天上运行的管理、测量、控制和任务调度, 实现卫星运行操作控制。天上信息网作为整星高速信息总线, 实现天上大容量的信息交换、处理和传送。嵌入式管理执行单元实现各设备的网络通信的接口^[9]。

2.2 星载软件体系架构设计

为了实现面向体系应用的星载软件, 需要从整星的角度对天上软件进行设计。整星软件的体系结构如图 3 所示, 包括管理层、功能层、服务层、协议层。管理层包括卫星配置管理, 卫星平台管理和卫星载荷任务管理以及系统故障诊断与恢复管理; 功能层提供管理层各个模块所需的各种服务, 例如飞行动力学计算模块、能源管理模块、通信管理模块等; 服务层就像驻留在每个天上上网节点上的软件交换机 (或软件总线) 一样实现纵向和横向的数据交换。协议层主要包含各种硬件设备的驱动。

图 3 中的软件体系结构是从整星的角度对软件进行分层设计, 从单机角度来看, 每个单机不一定具有所有的层

任务类型	光学遥感卫星				低轨通信卫星系统				微波遥感卫星			
软件框架技术	管理层	卫星平台管理		载荷任务管理		配置管理		故障管理				
	功能层	编队飞行控制	姿轨控制	轨道控制	能源管理	通信管理	数据管理	温度控制	空间链路管理	路由管理	载荷控制	
中间件技术	服务层	下位机通信	路由表服务	星间链路控制	温度测量	温度闭环控制				
		中间件软件										
		嵌入式、操作系统										
卫星数据模型技术	协议层	XML协议		CCSDS协议				TCP/IP协议				
		交换网络协议										
		MEU		智能传感器	智能执行器	处理器	内部连接	交换机或集线器				
		传感器	执行器									

图 3 整星分层模块化体系结构

次。如图 4 所示，例如星务主机具有任务的所有层次，某些星上设备只具有中间件层和协议层。在这种软件体系结构中，中间件层同软件交换机一样完成纵向和横向的数据交换。

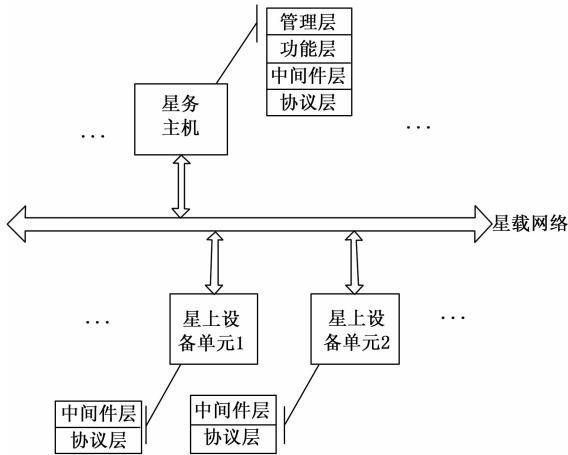


图 4 分布在各单机上的软件

3 星载核心控制软件框架设计

3.1 软件静态框架设计

星载核心控制软件的静态框架包括四层：管理层、功能层、服务层和协议层，如图 5 所示。管理层包括同步管理构件、遥测管理构件、遥控管理构件、故障诊断恢复管理构件、控制器管理构件、在轨操作构件、数据流管理构件，这些构件主要实现各层功能构件的协调和管理。功能

层主要包括各种算法构件、数据池和事件池。管理层构件是实现软件框架技术的核心组成部分。

- 1) 同步管理构件：整个软件的运行严格地遵守时间片约定，该构件用于对各层构件的同步运行管理；
- 2) 遥测管理构件：用于调度相关的功能层构件向数据池发送最新的遥测数据，或者从数据池中收集所需要的遥测数据；
- 3) 遥控管理构件：用于从事件池中读取并解析遥控指令，根据解析结果向事件池发送消息；
- 4) 故障诊断恢复管理构件：该功能构件包括了两部分，一部分位于管理层，用于对事件池中的故障消息进行检测并作出相应的处理。另一部分嵌入在服务层构件和协议层构件上用于报告故障消息，并接收管理层的故障诊断恢复指令。
- 5) 控制器管理构件：用于对各个算法构件运行的管理；
- 6) 在轨操作管理构件：用于根据遥控指令和控制器管理构件运行的结果完成对各种驱动器的操作控制；
- 7) 数据流管理构件：用于对数据池和事件池的管理和更新操作。

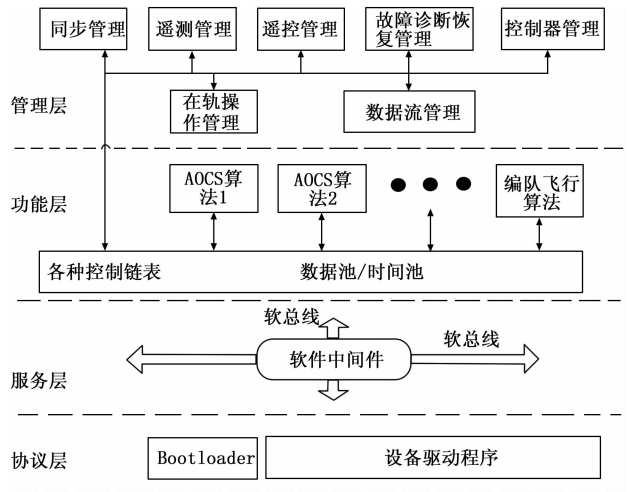


图 5 星载核心控制软件的静态结构

3.2 软件动态框架设计

星载核心控制软件的动态框架如图 6 所示。在单位运行周期内，软件运行分为若干个时隙，每个时隙中运行管理层的部分管理构件。在运行管理层构件时，该构件会完成其余三层相关功能的调用。此外，在每个时隙中都存在一个刷新存储系统的背景任务。例如，时隙 1 = {同步管理；遥控管理；故障诊断恢复管理；数据流管理}；时隙 3 = {同步管理；控制器管理；故障诊断恢复管理；数据流管理；在轨操作管理}。

管理层中的每个构件都对应着动态变化的对象链表。这些对象链表的节点分别对应着各种算法对象，遥测数据对象，异常消息对象等。管理层构件运行时，将轮询链表中的对象节点。对象节点中包括该对象的数据和对该类数



图 6 星载核心控制软件的动态结构

据需要进行的操作。如图 7 所示, 以同步管理构件对应的同步对象链表和遥测管理构件对应的遥测数据对象链表为例, 当同步管理构件运行时会轮询管理层各个构件。当轮询到遥测管理构件时, 遥测管理构件调用其对应的遥测数据对象链表, 如果有新的数据产生, 遥测管理构件不对遥测数据进行操作, 而是由遥测数据对象中的操作(类中的成员函数)负责将新数据送至数据池中, 完成遥测数据采集的工作。这种运行机制实现了领域模型和具体特例的解耦, 从而提高了软件的可复用性。

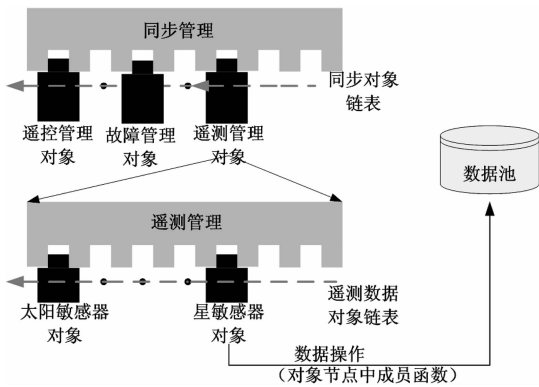


图 7 星务主机软件架构中链表示例

4 试验结果与分析

以小卫星为例, 其核心控制软件即星务中心计算机软件采用多任务加多中断的工作模式, 来满足多状态多功能需求的实时性操作要求。控制软件由系统软件和应用软件组成, 主要负责卫星运行期间的遥控数据管理、遥测数据发送管理、星上时间系统的管理、温度测量和温控的管理、整星各分系统和设备的管理、整星信息资源的调度和分配管理以及整星安全管理等, 是卫星能够正常运行的核心系统。系统软件主要包括实时多任务操作系统和中断服务程序。应用软件主要由数据处理程序及运行管理程序组成, 根据所需完成的功能, 主要包括初始化任务、调度管理任务、遥控遥测数据处理

任务、时间程控任务、内存刷新任务等模块, 各模块构成静态框架层中的管理层部分, 管理层的实施运行完成整星功能的构建。

通过将软件静态框架进行分层设置, 应用软件开发人员即可不必关注协议层的具体操作方式, 主要通过设置各任务间的序列方式、对不同任务内部的具体功能进行具体化修改达到完成整星预设功能的目的如图 8 所示。

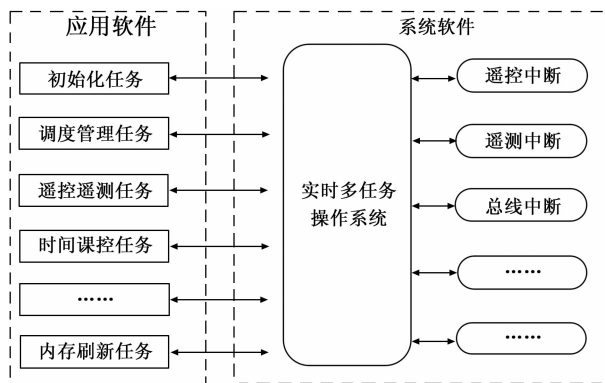


图 8 软件静态结构组成

星务中心计算机软件采用实时操作系统, 为多任务运行模式。首先在初始化任务中执行初始化功能、创建各种其它的功能任务, 再通过中断机制每秒激活调度管理任务, 由该任务按照一定顺序和要求依次调用遥控遥测任务、时间程控任务、内存刷新等任务。软件运行动态控制流程如图 9 所示。

每个任务在基于 DFH-OS 系统的基础上, 在软件初始时设置好优先级, 软件开始运行后, 即按照默认顺序执行, 完成整星的数据管理功能。为了保证在轨软件的正确运行及应对不同突发情况导致的软件变更, 整星软件中运行有故障诊断恢复模块和上注任务模块。当对遥测轮询采集数据构成的数据池进行特定数据分析, 如果选取的特征数据在一定时间达到软件内容的预设条件, 整星软件将根据预先设计好的算法逻辑采取动作, 保证软件恢复到正确状态; 同时由于整星一些参数在前期的不确定性, 并需保证软件

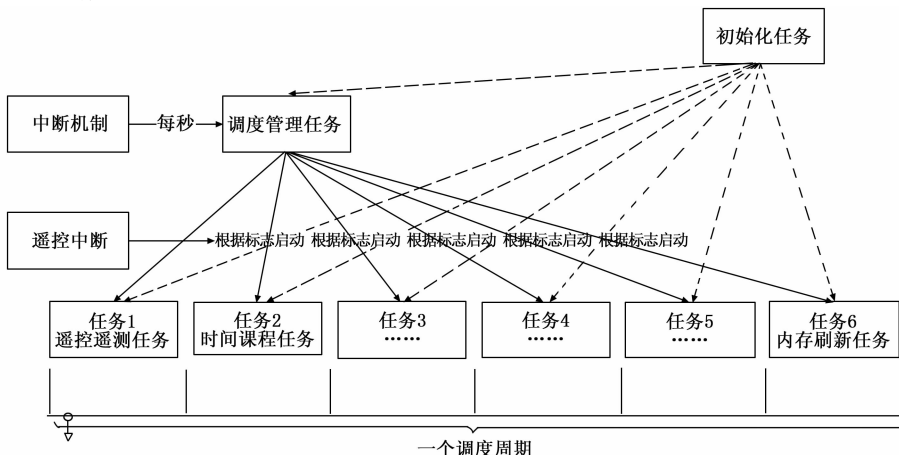


图 9 软件动态运行结构图

的可复用性和测试覆盖性,因此在整星测试阶段,首先预设好相关参数,对可变化的参数设置变更链接,然后进行整星测试,待最终一些参数确定后,通过上注变更相应链接即可达到完善最终软件的目的,增强软件的可复用性。

当无法通过更新计划内的构件对应的对象链表来达到完善修改软件的目的时,即当卫星在轨运行期间出现一些突发意外情况需要变更某些未计划改变的软件构件时,此时即可通过上注附加任务来完成变更,既保证了软件的完整性和复用性,又能适应多种不同情况保证正确性。

基于该静态和动态结构设计,针对某具体小卫星星务中心计算机软件,软件设计师在开发软件阶段,选择好复用软件构件部分后,可只重点关注特定型号的特定需求及各中断模块、各任务模块之间的关联及变化,无须关注修改模块内部的具体实施方式。以某些卫星数据为例,经统计,星务中心计算机软件新增加的、修改的代码行数约占代码总行数的 10%~20%,可复用的代码行数约占代码总行数的 80%~90%,极大地提高了软件的可复用性。

5 结论

在小卫星向体系化应用发展的趋势下,星载软件规模和复杂性快速提高,软件可复用性设计变得日益重要。本文在分析了面向体系应用的星载软件功能需求和技术需求的基础上,对小卫星典型软件体系架构进行了设计,提出采用软件框架技术实现软件可复用性设计。该框架技术通过软件构件、对象链表和数据池实现了领域模型和具体特例的分离,实现了星载软件的可复用性,为星载软件的快速开发、测试和集成打下良好的基础。

(上接第 119 页)

综上所述分析,欲提高姿态校正后的测量精度需采用小体积三分量磁传感器,且需对其进行正交、标度、零位等固有误差的校正以保证求解的准确性。

4 结束语

舰艇磁性检测是获取舰艇磁场的重要手段,实际中由于磁传感器姿态倾斜从而导致无法获得准确的磁测量值。将标准姿态且无误差的磁传感器置于倾斜磁传感器处,依据标准测量值采用 APO 算法对倾斜传感器测量值求解可得出其姿态角矩阵,进而计算获得准确的舰艇磁场值。本文对该方法进行了仿真与船模实验,验证了方法的准确性,同时指出实际中应采用小体积且无固有误差的三分量磁传感器以保证求解的精度,若能将磁传感器的固有误差与姿态误差一同校正,可使测量结果进一步精确,这也是该方法今后需要改进的方向。

参考文献:

[1] 王华征. 新型消磁决策专家系统体系结构及现代化消磁站设计

参考文献:

- [1] Lyke J C, Rosengren J C. System data Model (SDM) Source Code [R]. USA: Air Force Research Lab, 23 Aug 2012.
- [2] Preble J. Building SPA PnP Satellites [A]. Proc. of the AIAA 7th Responsive Space Conference [C]. 2009: RS7 - 2009 - 5002.
- [3] Alhorn, Dean C, et al. SMART: the future of spaceflight avionics [R]. USA: Marshall Space Flight Center, NASA, November 2010.
- [4] Armbruster P. ESA developments for on-board data systems [A]. Proc. of 2014 NASA/ESA Conference on Adaptive Hardware and Systems (AHS) [C]. 2014.
- [5] Orogo C, et al. Java-based Plug-N-Play (Flight) control systems for responsive space [A]. Proc. of the 4th Responsive Space Conference 2006 [C]. AIAA-RS4 2006 - 6002.
- [6] Pasetti A. Software frameworks and embedded control systems [M]. Berlin Germany: Springer, 2002.
- [7] Ziemke C, Kuwahara T, Kossevc I. An integrated development framework for rapid development of platform-independent and reusable satellite on-board software [J]. Acta Astronautica, 2011, 69: 583 - 594.
- [8] CCSDS 850.0 - G - 1. Spacecraft onboard interface services [S]. Washington D. C., USA: CCSDS Secretariat, 2007: 2.1 - 2.12.
- [9] 李孝同, 施思寒, 李冠群. 微小卫星综合电子系统设计 [J]. 航天器工程. 2008, 17 (1): 30 - 35.
- [1] [D]. 上海: 上海交通大学, 2009.
- [2] 阎齐方, 汪永浩, 张迪. 舰艇动态磁性检测方法 [J]. 船电技术, 2016; 36 (10): 17 - 19.
- [3] 隗燕琳, 陈敬超, 李贵乙, 等. 基于 Levenberg - Marquardt 算法的动态磁性检测站磁定位方法研究 [J]. 计算机测量与控制, 2017; 25 (9): 213 - 215.
- [4] 李华, 李庆民. 一种基于遗传算法的磁探头姿态角解算方法 [J]. 兵工学报, 2007; 28 (12): 1537 - 1539.
- [5] 王华征, 单志超. 舰船磁场测量真值的一种估算方法 [J]. 武汉理工大学学报, 2010; 34 (5): 891 - 894.
- [6] 周耀忠, 张国友. 舰船磁场分析计算 [M]. 北京: 国防工业出版社, 2004.
- [7] 周亢, 闫建国, 屈耀红. 全角度姿态角解算方法研究与仿真 [J]. 系统仿真学报, 2009; 21 (6): 1697 - 1700.
- [8] Shelokar P S, Siarry P, Jayaraman V K, et al. Particle swarm and ant colony algorithms hybridized for improved continuous optimization [J]. Appl Math Compu, 2007, 188: 129 - 42.
- [9] 喻学才, 张田文. 粒子群优化的多群蚂蚁算法 [J]. 哈尔滨工业大学学报, 2010; 42 (5): 766 - 769.