

通导系统通用自动测试平台设计

康祥熙

(中国西南电子技术研究所, 成都 610036)

摘要: 国内机载通导设备多采用手动测试的方式, 其形态原始效率低下, 因此通用自动测试平台的研究很有必要; 平台架构具备 6 个层次, 层级功能明确, 层间松耦合; 硬件平台借鉴典型的测试系统架构, 衍生出适合通导系统测试的物理结构; 软件平台采用集成 Teststand 引擎并围绕引擎二次开发的思路进行设计; 界面动态加载、报表与测试数据绑定技术为软件框架异构和测试报表自动生成奠定了基础, 对分遍历测试方法则提升了测试效率; 自动测试平台成功研制, 并实际应用于多个项目; 依托该平台形成了机载通导设备从科研、生产到运维的产品全生命周期的自动测试能力, 为探索新型综合保障模式, 提升作战飞机的持续可用性, 发挥良好作战效能奠定了技术基础。

关键词: 通导系统; 自动测试平台; 通用性; 保障模式

Design of General Automatic Test Platform for Communication Navigation System

Kang Xiangxi

(South West China Institute of Electronic Technology, Chengdu 610036, China)

Abstract: Manual testing is often used in domestic communication navigation equipment, and that testing mode is traditional and inefficient. Therefore, it is necessary to study the general automatic test platform (ATP). The structure of ATP has six levels, with clear hierarchical functions and loosely coupled layers. Based on the typical test system architecture, the hardware platform derives a physical structure suitable for testing communication navigation system. The software platform adopts such design idea of integrating TestStand engine and secondary development. Interface dynamic loading, report and test data binding technology laid the foundation for software frame structured and test report automatic generation. According to the content described in this paper, ATP was successfully developed and applied to projects. Based on ATP, the automatic testing capability of airborne communication navigation equipment from scientific research, production to operation and maintenance has been formed, which built a technical foundation for exploring a new integrated support mode, improving the sustained availability of aircraft and playing a good battle effectiveness.

Keywords: communication navigation system; automatic test platform; generality; support mode

0 引言

机载通导设备多, 信号交联关系复杂。在设备研制、型号转段和产品交付等阶段都会对设备进行大量的测试, 力求设备能够满足科研生产技术要求, 达到相应状态。现阶段对机载设备的测试仍较多沿用传统的方式, 例如通过仪器搭建测试系统, 手动操作人工记录的方式, 也有通过开发专用型测试系统, 针对特定任务开展测试。前者测试方式属于较原始的形态, 后者的测试方式虽然能够在一定程度上提高测试效率, 但系统研制成本高, 且适应性不强, 测试资源无法在多个测试任务间共享, 不通用。

研究适用于机载通导系统的多层次、松耦合、通用化自动测试平台, 避免传统测试系统中常见的一个型号一套测试软件的弊端, 从而解决软件架构不统一、测试程序在测试任务间不能共享的问题。最终实现多型号、多任务、同平台的通用型自动测试能力。

1 架构研究

通用自动测试平台架构在参照国际国内标准的同时, 结合航空产品测试特点, 将平台架构^[1]划分为 6 个层次, 如图 1 所示, 分别是测试应用层、测试程序层、资源调度层、仪器驱动层、ICD 协议适配层和物理设备层。平台每层涉及到的流程与模型均由 ATML 描述^[2], 主要有测试描述^[3]、仪器描述^[4]、UUT 描述^[5]和适配器描述^[6]等。除此之外平台架构提供满足 STD 标准^[7]和 ATML 标准的中间数据层, 规范数据格式, 分布实时存储。自动测试平台中的所有被测件、真实交联设备、仿真模型、信号仿真工具、数据显示记录工具等之间没有直接的接口, 在模块与模块之间不产生数据交换。任何模块 (DLL/LIB、COM 组件、EXE、VI 等) 只能与中间数据层产生数据交换, 从而解决了模块耦合的问题。中间数据层的引入使得自动测试系统在开发阶段无须提前约定通信接口, 更多关注模型功能与算法本身的开发。其次中间数据层提升了数据监控、数据管理与数据显示接入的便捷性。

测试应用层: 测试应用层作为整个软件架构的最顶层, 主要为用户提供人机交互接口管理界面, 实现自动测试配

收稿日期: 2019-05-30; 修回日期: 2019-07-05。

作者简介: 康祥熙 (1984-), 男, 硕士研究生, 工程师, 主要从事机载设备自动测试方向的研究。

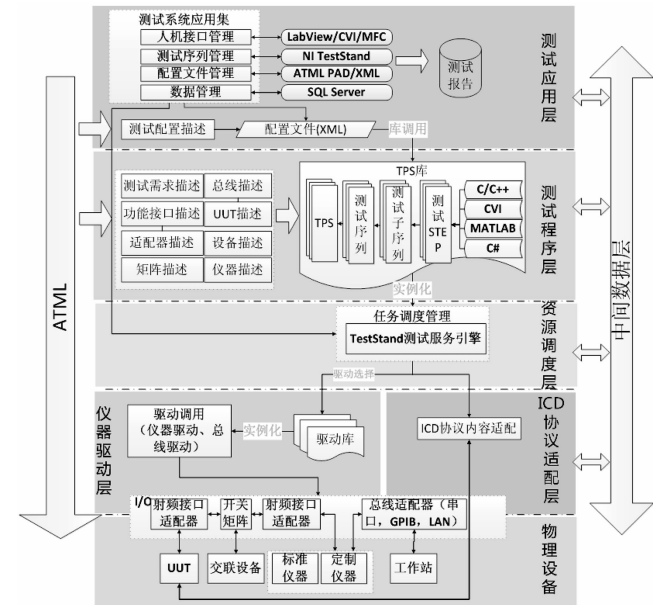


图 1 通用自动测试系统平台架构

置文件生成及管理，测试数据显示及管理，测试报告生成及管理，测试序列管理功能。

测试程序层：测试程序层为自动测试系统的核心业务层。该层提供了 TPS 库。TPS 库的调用依赖于测试应用层的测试项目配置，由配置决定哪些 TPS 被调用。TPS 库由测试序列组成，测试序列又由各级子序列构成，子序列由最基础的测试步组成。该层通过 TestStand 集成诸如 C/C++、C#、CVI 和 MATLAB 等外部应用，所有的测试序列均由 ATML 描述其接口和逻辑。

资源调度层：任务调度管理借助于 TestStand 引擎实现，实现 TPS 调用运行，是整个测试系统运行的核心^[8]。

仪器驱动层：仪器驱动层包含总线驱动、通用仪器（VISA 封装）和非标定制仪器驱动。仪器驱动层是调度控制层与物理设备层的中间层，通过仪器驱动层实现了仪器的实例化配置和控制运行。

ICD 协议适配层：ICD 协议适配层包含 ICD 协议内容适配和总线接口适配。测试应用基于 ICD 和总线通硬件设备通信，不同项目 ICD 存在差异，为了让应用开发无须过多关注 ICD 协议与总线接口，采用 ICD 协议内容和总线接口适配的方法屏蔽低层级内容。ICD 协议适配层的引入有助于实现 ICD 到变量，变量到 ICD 的转换，降低了 TPS 的开发难度。

物理设备层：物理设备层为自动测试系统架构得最底层。物理设备层包含了执行测试任务的具体测试设备、被测对象、交联设备、模拟器和工作站。

2 硬件平台

2.1 典型系统物理架构

典型的开放式的自动测试系统的物理结构^[9]主要由控制子系统、仪器子系统、激励子系统和接口适配器组成。

其硬件结构框图如图 2 所示。控制子系统通过控制总线连接仪器子系统和激励子系统。控制子系统控制激励子系统产生激励信号，信号通过接口适配器输出至待测设备，待测设备响应并通过接口适配器将响应信号回传至仪器子系统，完成测试测量。

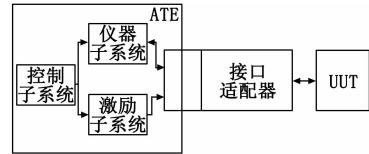


图 2 自动测试系统硬件结构框图

2.2 物理架构

经典的自动测试系统物理架构已包含测试系统所需的基本要素，能够满足一般产品测试需求。而航空通导设备因产品的复杂性往往需要具备诸如监控、调试等更多功能。设计适用于通导系统的自动测试平台物理架构，需结合自身测试需求。如图 3 所示为某型机载通导设备自动测试系统物理架构，其在保留控制子系统、激励测量子系统（专用激励器和通用仪器）、适配子系统的基础上增加了监控子系统和电源子系统。在此架构基础上如需实现基于复杂场景的动态测试，应增加场景模拟子系统、动态信号解算子系统。

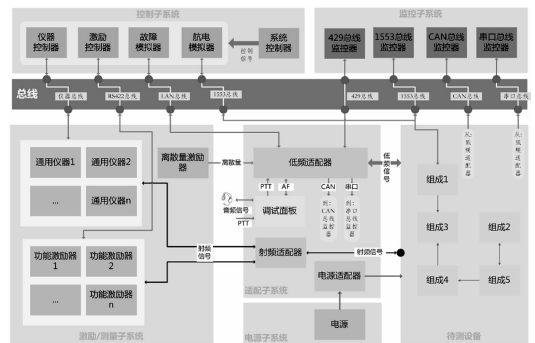


图 3 某型机载通导设备自动测试系统物理连接图

控制子系统通过 RSS422、GPIB 总线连接并控制激励/测量子系统。激励子系统通过适配子系统（射频适配器和总线适配器）实现测试通路选择、信号调理和转发至待测设备，待测设备接收激励信号，完成对应功能信号处理，产生响应和数据分析结果，通过适配子系统，再次将响应回传至测量子系统，通用仪器接收回传信号，处理并将结果上传至控制子系统存储和显示。监控子系统则是通过监控点实现对诸如 429、1553、CAN 等常用航空总线信号实时采集和 ICD 分析，实现测试过程、测试异常的监控。电源子系统为待测产品提供直流 28 V 或交流 115 V 供电，并可实现加电控制。

3 软件平台

3.1 设计思路

TestStand 是美国 NI 公司一款自动测试开发与管理软件，其在支持图形化测试序列编辑、测试步骤顺序拖曳调整、多种代码模块识别集成、串行并行测试等方面优点突

出。TestStand 主要用于完成测试管理、测试执行、任务汇报等复杂的业务逻辑。测试业务与待集成外部应用可以分开开发，在一定程度稍降低了开发人员耦合交织的情况发生，减少了开发人员的重复性工作，将开发重点转移到测试要求和测试指标，最终提高了开发效率，缩短了自动化测试和验证系统的研制周期和部署时间。

鉴于 TestStand 上述优点，且提供标准的 COM 组件接口和外部应用（VI、DLL、EXE 等）调用接口，通导系统通用自动测试平台软件设计采用集成 TestStand 引擎，围绕引擎二次开发的思路进行。

- 1) 采用 C# 语言开发图形用户界面，实现测试任务管理和调用；
- 2) 采用集成 TestStand 作为运行引擎的方式，实现测试任务运行；
- 3) 采用 Office 作为测试报告模板编辑工具，实现测试报告的自动生成。

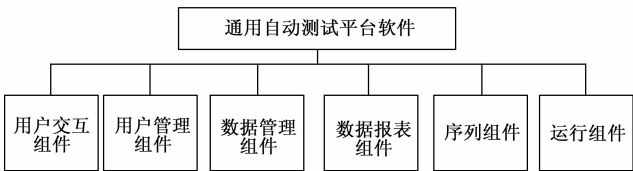


图 4 软件组成

3.2 软件组件

通用自动测试平台软件采用异构的软件框架，按功能分为多个组件，可实现软件自由裁剪和扩展，具体组件组成如图 4 所示，主要由用户交互组件、用户管理组件、数据管理组件、数据报表组件、序列组件和运行组件组成。用户交互组件是整个系统的总控界面，完成人机交互工作。运行组件作为核心调度组件，实现过程调度和数据推送功能。数据管理组件作为所有状态和信息的存储地，用于实现过程数据的持久化。



图 5 软件主界面

图 5 所示为自动测试平台软件主界面，软件主界面采用左树右表的布局。左侧树型列表提供测试项目勾选和管理，中间测试步骤区域则提供测试序列简单编辑和运行过程查看，下方波形显示区域提供测试数据图形化显示，右侧区域为序列状态查看，顶部工具栏则提供测试运行、暂停、停止、串并行模式选择和系统设置等功能。

3.3 交互流程

软件组件之间交互流程如图 6 所示。用户登录界面验证使用人员权限，加载测试人员、开发人员、管理员等角色配置文件，主界面根据配置文件构建界面内容与风格。管理员通过用户管理组件实现用户的新增和删除，以及权限分配。在主界面用户操作选择测试序列，点击开始则启动引擎并通知运行组件将测试序列加载至引擎执行，执行过程中调用底层驱动控制测试硬件实现总线信号数据收发、功能激励器激励信号产生、通用仪器测量和结果采集等自动化测试。测试结论数据上传至数据报表组件。数据报表组件打开测试序列对应模板，完成数据写入和测试报告生成。在测试过程若需要查看数据和显示的时候，则需要在序列编辑时，通过 DLL 组件进行设置，设置完成便可在测试运行过程中，实现测试数据实时数值显示和二维波形显示。

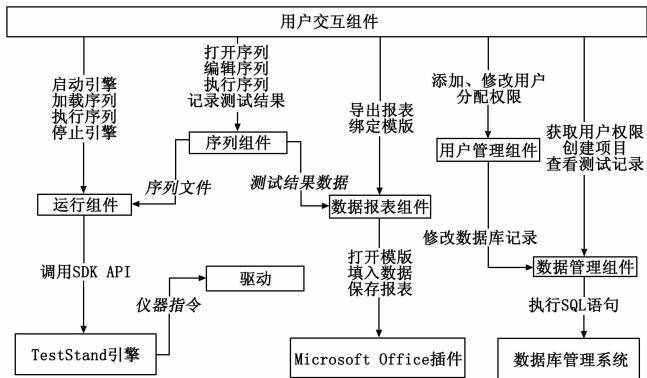


图 6 软件组件交互流程

同时，每次测试过程中的测试数据均会存入数据库中，需要导出历史报表时可以先查询出感兴趣时间的测试结果，然后将测试结果依据数据书签映射关系导入至报表模板，生成测试报告。

4 关键技术

4.1 界面动态加载技术

自动测试用户界面并非固定不变，可根据登录人员权限动态加载软件界面。软件界面采用 Dock 界面框架，可通过拖拽操作进行窗口布局，不同测试人员可以有专属测试操作界面，且界面窗口异构，扩展性强。

1) 数据驱动图形。

对于主界面的波形图绘制部分采用数据驱动的方式进行管理。绘图显示窗口区域加载绘图插件，绘图插件根据实际需求开发，以 DLL 的形式存放于本地，绘图插件加载完毕，则将需要绘图的变量与测试变量绑定，测试数据由序列组件通过 TestStand 引擎获取，在获取到数据后通过接口适配调用对应绘图插件解析数据包并调用 PlotData 方法绘图，其结构关系如图 7 所示。

2) 自定义图形界面加载。

用户自定义的绘图插件采用动态加载的方式进行管理。要求实现的插件代码为 C# 编写，其 DLL 中的主类类名必须

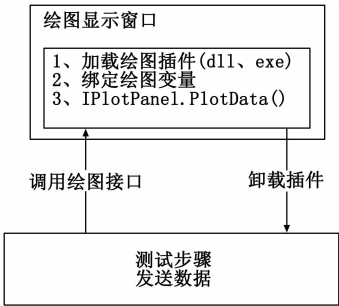


图 7 动态波形显示结构原理

固定，而且必须继承 Form 类。如果要使用 Dock 框架则必须继承 Dock 框架的 DockContent 类。DLL 加载原理如下：

- (1) 利用反射进行动态加载和调用
`Assembly assembly = Assembly.LoadFrom (DllPath);`
- (2) 加载 dll 后，需要使用 dll 中某类
`Type type = assembly.GetType ( "TypeName" );`
- (3) 实例化类型
`Object obj = Activator.CreateInstance (type, params []);`
- (4) 调用类型中的某个方法，需要首先得到此方法
`MethodInfo mi = type.GetMethod ( "MehtodName" );`
- (5) 然后对方法进行调用，多态性利用参数进行控制
`mi.Invoke (obj, params []);`

4.2 报表与测试数据绑定技术

测试报表是自动测试系统最终输出结果。通过测试报表直接反映测试数据和测试结论。为满足各式各样的测试报告样式需求，报表模板需能够由用户任意定制。报表与测试数据绑定技术研究能够解决此问题。通过代码验证和书签方式对 Word 报表进行指定位置的内容填入和修改，通过报表组件建立 TestStand 和 Office Word 的连接，实现测试序列的变量和报表中待存储数据的映射（即：书签—变量），并将映射关系存入数据库。在导出报表时依据对应关系填写报表内容。图 8 描述了绑定报表模板的流程图。

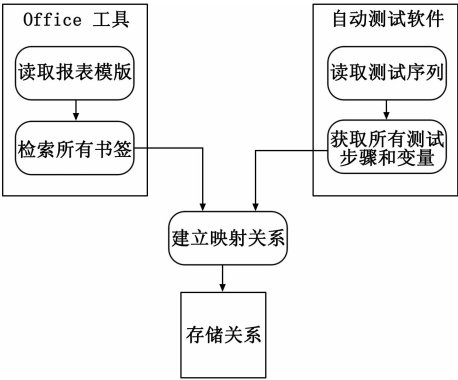


图 8 绑定报表模板的流程图

4.3 对分快速遍历测试法

自动测试一般采用固定步长遍历的方法进行诸如灵敏度、动态范围等指标的测试。其主要思路是将激励信号从初

值开始按固定步长递进增大或减小，直至遍历出预期的测量值。采用固定步长遍历测试的最大遍历测试次数一般为：

$$\text{遍历测试次数} \max = (b - a) / \text{step} + 1$$

以某功能接收灵敏度测试为例，将功能激励器输出信号电平幅度的初值设置为 -110 dBm，以 1 dBm 为步进不断增加激励电平幅度，每增加一个步进就测量一次输出，看是否满足指标要求，不满足则继续遍历，直到激励信号电平幅度增大到 -90 dBm，此时测量获得了预期的输出。这种固定步长的遍历方法虽然不影响测试精度，但遍历次数较多，仅接收灵敏度测试需要进行 21 次遍历，花费较长时间。为解决传统遍历测试法遍历次数多、测试效率低的问题，针对线性系统可采用对分遍历区间的方法缩短遍历次数，提升测试效率。

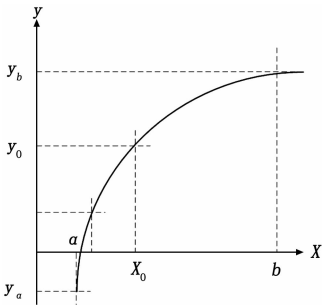


图 9 对分遍历法示意图

首先设置激励信号为 a ，测量此时的输出值，记为 y_a ，要保证待测设备输出的 y_a 为无效电平，设置激励信号为 b ，测量此时的输出值，记为 y_b ，要保证待测设备输出的 y_b 为有效电平。满足上述条件后，此时将遍历区间对分为 $[a, x_0]$ ， $[x_0, b]$ 两个子区间；然后判断 $y(x_0)$ 是否有效，如果有效，对分 $[a, x_0]$ ，无效对分 $[x_0, b]$ ，直至 $|x_n - x_{n-1}| \leq \text{tolerance}$ 。

对分遍历法的流程示意如图 10。

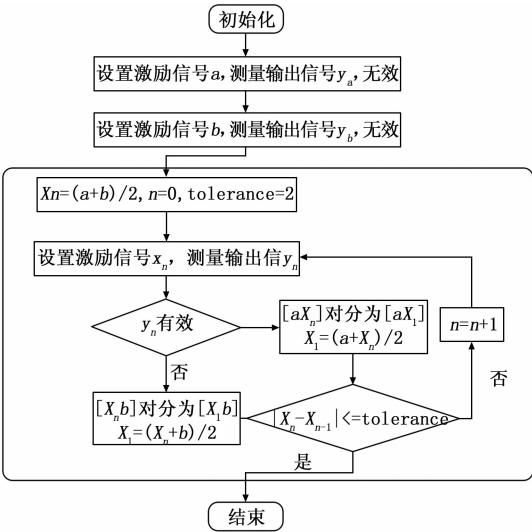


图 10 对分遍历法流程图

同样以该功能接收灵敏度测试为例，激励电平幅度的初值为 -110 dBm，而该测试项指标要求优于 -88 dBm。因

此采用对分遍历法进行测试，在同样测试精度要求下，获得预期输出时的遍历次数不超过 6 次。

5 实验结果与分析

在联试场地操作自动测试平台对通导系统分机 1、分机 2 的通信功能和导航功能进行了主要性能指标测试，并对测试时长进行统计，具体数据见表 1。

表 1 测试时长统计

分机		功能	测试项	手动测试时长	自动测试时长
分机 1	通信功能 1		测试准备	24	1
			模式 1 发射功率测试	10.92	3.2
			模式 1 接收灵敏度测试	17.3	7
			模式 2 发射功率测试	10.52	2.7
			模式 2 接收灵敏度测试	51.5	27
			模式 3 接收误包率测试	22.22	5.3
			模式 3 发射功率测试	16.8	1.1
	通信功能 2		测试准备	26	9
			模式 1 通信测试	40.24	2.6
			模式 2 通信测试	15.7	3.5
			模式 3 通信测试	12.32	3.7
	通信功能 3		测试准备	30	1
			功率测试	6.12	1.6
			话音对通测试	8.6	2.7
			话音灵敏度测试	17.4	5.1
分机 2	导航功能 1		测试准备	18	1
			灵敏度测试	13.5	3.4
	导航功能 2		测试准备	22	1
			灵敏度测试	25.36	6.7
			时间测试	3.4	2.1
			精度测试	7.4	3.6
	导航功能 3		测试准备	21	1
			灵敏度测试	16	3.5
	导航功能 4		测试准备	23	1
			精度测试	13.32	2.4
			灵敏度测试	7	6.1
	导航功能 5		测试准备	22.4	1
			功率测试	13.2	2.1
			灵敏度测试	34	5.4
			精度测试	10.2	1.5
	导航功能 6		测试准备	14.4	1
			范围测试	58	8.2
			功率测试	17.82	2.5
			模式测试	10.4	0.9
			XX1 精度测试	30	2.1
			XX2 精度测试	5.2	1.5
			XX3 精度测试	9.4	2.6
			灵敏度测试	58	9.2
	导航功能 7		测试准备	19	1
			XX 测试	6.4	3.2
			灵敏度测试	11	6.2
合计(分钟)				799.04	156.7
合计(小时)				13.3	2.6

由表 1 可知，采用传统手动测试方法对 通导系统测试的时间为 13.3 小时，采用自动测试方法的测试时长为 2.6 小时，实现了测试效率 5.1 倍提升。

除此之外，由于采用自动测试平台测试，在测试准备阶段只需 1 人便可以完成加电检查和加电操作等测试准备操作，在测试阶段由 1 人运行测试平台、选择测试序列和运行步骤，测试自动运行过程中，测试人员可以不在现场，而完全交由系统自动完成待测设备/激励器/通用仪器/设置、测试、测量和测试结论步骤。相比传统手动测试 2~4 人的人力需求，采用自动测试平台测试人力需求缩减至 1 人。

6 总结

采用本文所述架构设计的通导系统通用自动测试平台，是一套具备图形化的测试序列编辑环境，能够实现一键串行或并行测试的以科研测试、生产检测和综合维修保障为应用需求背景的通用、高效、便捷、可扩展的包含成熟软硬件平台的自动测试开发运行环境，现已实际应用于多个项目。

通导系统自动测试平台的使用不仅简化了测试流程，规避了人带来的测试质量不稳定，测试一致性不高，测试操作繁琐等缺点，还提升了测试效率，解放了人力资源，同时为探索内外场飞机综合保障 [10] 的新型模式提供了一条可行的发展之路。

参考文献：

[1] 张若时. 面向信号的 ATS 软件平台研究 [D]. 成都：电子科技大学，2012.

[2] IEEE SCC20, Automatic Test Markup Language (ATML) for Exchanging ATE and Test Infomation via XML [S], 1671 - 2010.

[3] IEEE SCC20, IEEE Standard for Automatic Test Markup Language (ATML) Test Description [S], 1671.1 - 2017.

[4] IEEE SCC20, IEEE Standard for Automatic Test Markup Language (ATML) Instrument Description [S], 1671.2 - 2012.

[5] IEEE SCC20, IEEE Standard for Automatic Test Markup Language (ATML) UUT Description [S], 1671.3 - 2017.

[6] IEEE SCC20, IEEE Standard for Automatic Test Markup Language (ATML) Test Adapter Description [S], 1671.5 - 2015.

[7] IEEE SCC20, IEEE Standard for Signal and Test Definition [S], 1641 - 2010.

[8] 胡典钢. TestStand 工业自动化测试管理 [M]. 1 版. 北京：电子工业出版社，2016.

[9] 吕英军. 军用飞机低频设备通用自动测试系统研究与设计 [D]. 吉林：吉林大学通信工程学院，2008

[10] 左 力. 航空装备维修保障模式研究 [J]. 电子设计工程，2017, 25 (3): 1 - 4.