

DOI:10.19651/j.cnki.emt.2108504

基于 LabWindows/CVI 的流场控制系统设计

王政¹ 周宣赤^{1,2} 高维金^{1,2} 崔春^{1,2}

(1.中国航天空气动力技术研究院 北京 100071; 2.北京航天益森风洞工程技术有限公司 北京 100071)

摘要: 针对当前某小型高速风洞信息化程度不高、流场须要通过手动调节方式实现稳定的情况,根据风洞流场控制流程,设计基于 LabWindows/CVI 的上位机监控系统,并在某小型高速风洞实现部署;系统通信采用基于 DataSocket 的 OPC 通信协议,上位机通过接收 NI OPC 服务器的数据,高效且可靠得实现与下位机 PLC 的信息交互;上位机软件运用多线程、异步定时器等技术,实现数据的实时显示、历史数据记录及数据存储等功能,并通过可视化界面进行人机交互,方便操作人员使用。经大量试验分析,新的流场控制系统调节速度快、精度高、稳定可靠,满足某型高速风洞流场控制系统的设计要求,对于降低能源能耗、提高风洞信息化和自动化水平,具有一定的参考价值。

关键词: LabWindows/CVI;OPC;流场控制;PLC

中图分类号: TP271.3 **文献标识码:** B **国家标准学科分类代码:** 510.8060

Design of flow field control system based on LabWindows/CVI

Wang Zheng¹ Zhou Xuanchi^{1,2} Gao Weijin^{1,2} Cui Chun^{1,2}

(1. China Academy of Aerospace Aerodynamics, Beijing 100071, China;

2. Beijing Aerospace Yisen Wind Tunnel Engineering Technology Co., Ltd., Beijing 100071, China)

Abstract: According to the situation that a small high-speed wind tunnel at a low level of informatization and the flow field of the wind tunnel needs to be manually adjusted to achieve stability. Based on the flow field control process, this paper designs a LabWindows/CVI-based upper computer control system and applies it in a small high-speed wind tunnel. The system communication uses the OPC communication protocol based on DataSocket, The upper computer can efficiently and reliably realize the information interaction with PLC by receiving the data of Ni OPC server. The upper computer software uses multi-threading, asynchronous timer and other technologies to realize the functions of real-time data display, historical data record and storage, and friendly human-machine interface which is convenient for operators to use. Through a lot of test and analysis, the newly designed flow field control system has fast response, stable operation and high accuracy, which meets the design requirements of high-speed wind tunnel control system. It has certain reference value for reducing energy consumption, improving information technology and automation level of wind tunnel.

Keywords: LabWindows/CVI;OPC;flow field control;PLC

0 引言

LabWindows/CVI 是美国国家仪器公司推出的虚拟仪器上位机设计软件,主要用于测控系统上位机软件的开发,通过丰富的底层函数库与直观的图像化功能模块,结合 OPC、串口通信等技术,在工业控制等领域有着广泛的应用^[1-3]。目前,采用 LabWindows/CVI 作为上位机开发软件是一种极为高效、可靠的方法^[4],文献[2]使用 LabWindows/CVI 开发上位机监控软件,采用基于 DataSocket 技术和 OPC 协议的实时网络通信方法,验证了

CVI 实现 OPC 通信技术的可行性;文献[3]同样采用 CVI 作为开发平台,用串行通信方式编写设备通信程序,实现上位机与欧姆龙 PLC 的通信,为系统通信方式多样性提供了借鉴;文献[5]研究了虚拟仪器技术与下位机在流场控制系统中的应用,为控制架构的建立具有一定的指导意义,但缺乏对系统整体架构设计的讨论,对 LabWindows/CVI 在系统中的运用缺少代码支持和借鉴。

针对上述现状,文中提出一种基于 LabWindows/CVI 的流场控制系统设计方法,采用基于 DataSocket 技术的 OPC 通信协议,通过 OPC 服务器搭建上位机与下位机

收稿日期:2021-12-03

PLC 之间的通信桥梁,简化了两者之间信息交互的开发难度,并根据工艺流程增加合适的功能模块,构建完整的流场控制系统,并在某小型高速风洞上实现运用。

1 系统架构及技术原理

在风洞试验过程中,稳定流场的建立须要在引射器管路压力稳定的情况下,调节主气路总温、总压、静压达到工艺要求的范围并保持稳定,从而为风洞试验建立稳定可靠、品质优良的流场。在此过程中,系统除了控制多个气动调节阀、电磁调节阀、开关阀等可控执行器的输出量外,现场大量的传感器信号需要实时采集和存储,供操作人员监控及后续的数据处理^[6]。

本文系统通过 LabWindows/CVI 实现上位机指令下达、数据实时显示和存储功能^[7],将当前流场的品质参数实时反馈给操作人员,结合下位机 PLC 数据采集和丰富的逻辑控制功能,能够胜任流场控制领域多种复杂的控制任务^[8];根据上位机发送的控制指令和设定的流场目标参数,PLC 接收指令从而完成相应电磁阀门、气动阀门的远程调节和自动控制。本设计既可实现手动开环控制执行器阀门输出开度,又能根据需求通过设置相应的流场目标温度、压力值,实现流场控制系统的闭环自动控制。基于 LabWindows/CVI 与 PLC 的流场控制系统结构框图如图 1 所示。

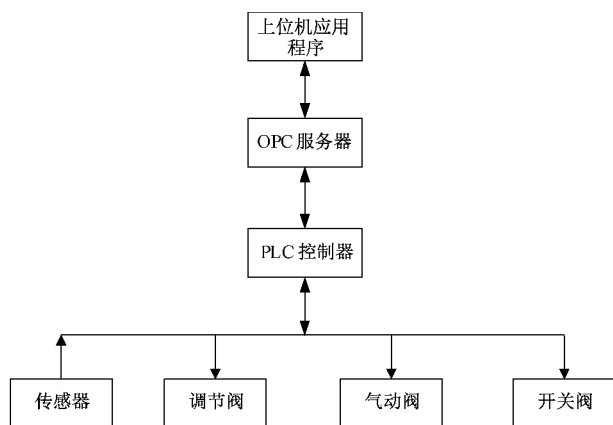


图 1 流场控制系统网络拓扑图

1.1 OPC 技术简介

OPC 是一种基于微软 OLE/COM 技术的通用技术规范,近年来在工业控制领域的发展较为迅速^[9]。作为一种标准的通信协议,OPC 通信采用客户/服务器模式,这样使得上位机与下位机之间的通信更加可靠,互操作性更强^[5]。OPC 通信结构原理如图 2 所示。以本系统为例,采用 NI OPC Server 作为服务器软件,利用内部集成的 OPC 标准接口模型,选取匹配的下位机 PLC,并新建 OPC 组集合对象(OPC group),并在此集合对象下定义多个项对象(OPC item),例如传感器信号、开关阀信号等,并与 PLC 端定义的物理地址一一映射,实现 OPC 服务器与现场 PLC 的信息交互。

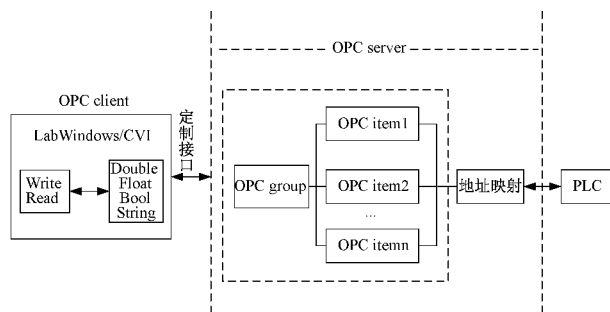


图 2 OPC 通信结构原理

1.2 PLC 简介

可编程逻辑控制器 PLC 在数据采集或测控领域用途极为广泛,具备优良的逻辑控制功能、智能控制模块和较强的抗干扰能力。

本系统选用西门子 1200 系列 PLC 作为下位机主控单元,其结构紧凑、具备丰富的指令集,通过扩展模块即可实现模拟量采集、数字通讯等功能,广泛应用于中小型测控生产领域^[10],性能完全满足中小型风洞的流场控制。

2 软件架构设计

基于 LabWindows/CVI 设计的流场控制系统软件包含多个线程,分为一个主线程和多个次线程。前者用于实现并运行人机交互的用户界面,响应操作人员的输入输出事件;后者用于后台数据处理,包括数据采集、数据显示、实时曲线、数据存储等,实现其他的对时序有特殊要求的功能。

上位机程序初始化阶段,加载 DataSocket 服务器、多线程/异步定时器初始化函数、Excel 服务器函数获取相关函数句柄;结合多线程、异步定时器等技术,其中次线程 1 用于采集从下位机 PLC 上传的数据,次线程 2 用于历史数据读取,次线程 3 用于实时曲线显示,次线程 4 用于数据存储^[2]。上位机软件设计原理如图 3 所示。

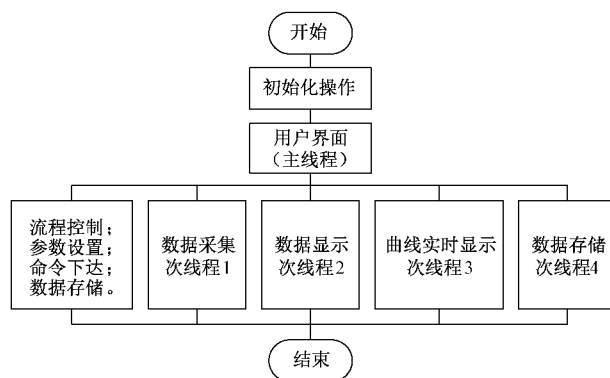


图 3 上位机软件设计原理

3 系统功能模块设计

当程序有多个任务需要同时进行时,通常将这些繁琐

的线程封装在一个进程内,这样可以有效减少软件的开销。针对多线程设计,LabWindows/CVI 支持多线程和异步定时器^[1,11],可有效提高系统后台辅助的能力,使得人机交互的功能更为完善、流畅。

3.1 多线程运行机制

对于数据采集和显示,本设计采用多线程机制实现。多线程主要用于实现循环执行的任务,中央处理器轮换着分配线程时间片,通过时间片轮转,实现线程的多任务处理^[12]。

流场控制上位机采用多线程实现接收 OPC 服务器实时数据的功能,可以通过调用线程创建函数 CmtScheduleThreadPoolFunction()实现,定义的线程回调函数 Thread_Function 完成相应的逻辑功能。设计程序示例如下:

```
/* 创建数据采集子线程,程序位于系统使能启动按钮回调函数中 */
int CVICALLBACK Start_callback (int panel, int control,
int event, void * callbackData, int eventData1, int
eventData2)
{
    ...
/* DEFAULT_THREAD_POOL_HANDLE 表示缺省的
函数句柄,ThreadFunction_callback 表示定义的线程回调
函数,ThreadID 标识唯一的线程 ID 号 */
    CmtScheduleThreadPoolFunction ( DEFAULT _
THREAD_POOL_HANDLE, ThreadFunction_callback,
0, &ThreadID);
    i_Threadflag=1;//标志位
    ...
}
/* 线程回调函数 */
int CVICALLBACK ThreadFunction_callback ( void *
functionData)
{
    unsigned int size;
    while(i_Threadflag)
    {
/* 传感器数据采集,一次采集包含 24 个数据的浮点型数
组 */
        DS_GetDataValue ( ds _ AI _ OPC _ sensorHandle,
CAVT_DOUBLE | CAVT_ARRAY, &.value, 24 *
sizeof(double), &.size, NULL);
    }
    return 0;
}
/* 释放子线程 */
i_Threadflag=0;
```

```
CmtWaitForThreadPoolFunctionCompletion (DEFAULT_
THREAD_POOL_HANDLE, ThreadID, OPT_TP_
PROCESS_EVENTS_WHILE_WAITING);
```

```
CmtReleaseThreadPoolFunctionID ( DEFAULT _
THREAD_POOL_HANDLE, ThreadID);
```

3.2 异步定时器运行机制

上位机数据显示和存储分别创建两个线程,使用异步定时器来实现。异步定时器多用于在固定的时间间隔,循环执行设定的工作流程^[13,14]。异步定时器创建的次线程按照设定的频率将采集到的数据同步到界面数据表中,用于历史数据显示;另一个线程按照设定的时间间隔,对采集的数据进行本地保存。

异步定时器运行次线程的代码,需要加载资源库 toolslib. fp。首先通过异步定时器创建函数 NewAsyncTimer,设定存储的循环时间及相应任务的回调函数;SetAsyncTimerAttribute 用于定时器使能控制,最后,异步定时器使用完毕后通过 DiscardAsyncTimer 函数关闭,释放占用的内部资源。以数据存储为例,部分程序示例如下:

```
/* 定义函数句柄 Asyncsave_handle */
int Asyncsave_handle;
/* 新建异步定时器并使能用于数据存储,存储频率 1/d_
frequent */
Asyncsave_handle = NewAsyncTimer (d_frequent,
-1, 1, asynsave_callback, 0);
/* 存储功能实现相应子函数示例 */
int CVICALLBACK asyn_callback (int reserved, int
timerId, int event, void * callbackData, int eventData1,
int eventData2)
{
    switch (event)
    {
        case EVENT_TIMER_TICK:
            SetTableCellRangeVals ( valuepanel,
VALUEPANEL_VALUETABLE, MakeRect (2, 5, 24,
1), &.value[24], VAL_COLUMN_MAJOR);
            break;
    }
    return 0;
}
/* 失能异步定时器 */
SetAsyncTimerAttribute ( Asyncsave _ handle,
ASYNC_ATTR_ENABLED, 0);
/* 关闭异步定时器 */
DiscardAsyncTimer (Asyncsave_handle);
```

3.3 通信接口设计

西门子 1200 系列支持 OPC 协议,只需在 TIA 编程软

件中将控制器设置为“允许来自远程对象的 PUT/GET 通信访问”,相关背景数据块设置为非优化的块选项,即可通过 OPC Server 访问 PLC 硬件的相关存储空间。上位机软件及通讯功能采用 LabWindows/CVI 2013 及 NI OPC server 实现。CVI 支持 DataSocket 函数库,通过 DataSocket 技术实现与 OPC Server 通信,满足进程间的数据交换的需求。

在流场控制系统中,通信数据类型主要包括用于开关信号的布尔型数据,用于实现阀门开关控制及限位开关信号反馈的采集;还包括浮点型数据,用于参数设置、现场传感器采集及阀门开度反馈信息采集;字符型数据,用于显示界面的数据显示^[15]。对于数据量较多的浮点型数据,可在 PLC 相应的背景数据块内将浮点型数据设置为数组类型,上位机通过 OPC 按照读取数组的形式获取现场数据,提高上位机读取数据的效率。上位机程序在主程序中初始化 DataSocket 服务器加载函数^[16],且需要包含 datasocket.h 头函数。部分程序示例如下。

```
/* 初始化阶段启动本地 DataSocket 服务器程序,在系统主线程启动前启动 */
```

```
DS_ControlLocalServer (DSConst_ServerLaunch);
RunUserInterface ();
```

```
/* 主线程结束前,结束本地 DataSocket 服务器程序 */
```

```
DS_ControlLocalServer (DSConst_ServerClose);
```

```
/* 定义 DataSocket 函数句柄 ds_AI_OPC_sensorHandle, 用于现场传感器采集 */
```

```
DSHandle ds_AI_OPC_sensorHandle;
```

```
/* 定义 DataSocket 函数句柄 ds_PC_startbuttonHandle, 用于阀门开关量控制 */
```

```
DSHandle ds_PC_startbuttonHandle;
```

```
/* 高压阀门开关回调函数 */
```

```
int CVICALLBACK HighPressSWITCH_callback (int panel, int control, int event, void * callbackData, int eventData1, int eventData2)
```

```
{
    BOOL i_S=1;
    BOOL i_R=0;
    /* 阀门开关量控制 */
    DS_SetDataValue(ds_PC_startbuttonHandle, CAVT_BOOL, &i_S, 0, 0);
}
```

```
/* 多线程回调函数 ThreadFunction_callback */
```

```
int CVICALLBACK ThreadFunction_callback (void * functionData)
```

```
{
```

```
...
```

```
/* 一次读取 24 个浮点型数据 */
```

```
while(i_Threadflag)
```

```
{
    DS_GetDataValue (ds_AI_OPC_sensorHandle,
CAVT_DOUBLE | CAVT_ARRAY, &value, 24 *
sizeof(double), &size, NULL);
    ...
}
return 0;
}
```

3.4 数据保护机制

在流场控制系统中,由于多个线程共享进程的某些全局变量或虚拟地址空间,极易造成数据冲突;为了避免这种情况的发生,对于系统定义的全局数据变量,需要对资源进行保护,以免这些数据在被多个线程同时读取时,发生无法预料的问题^[17]。系统上位机采用线程安全队列是避免发生数据死锁。本设计部分程序示例如下:

```
/* 创建数据队列,数据量大小 24 */
```

```
CmtNewTSQ(24, sizeof(double), OPT_TSQ_DYNAMIC_SIZE, &queue_Handle);
```

```
/* 将数据写入安全队列 */
```

```
CmtWriteTSQData(queue_Handle1, value[0], QUEUE1_NUM0, SQ_INFINITE_TIMEOUT, NULL);
```

```
/* 将数据读出安全队列 */
```

```
CmtReadTSQData (queueHandle1, value [0], QUEUE1_NUM0, SQ_INFINITE_TIMEOUT, NULL);
```

3.5 人机交互设计

软件按照实现功能划分,将相应控制按钮设置在不同的子窗口内。系统主界面通过组态画面,按照实际的布局将整个风洞试验区域的实体结构虚拟出来,包括各可控执行器的位置、传感器空间布置等。界面操作模块包括手动控制、自动控制选择开关,系统使能开关、阀门开关按钮,可控执行器手动调节滑块等。操作人员按照工艺要求,设置流场目标参数即可下达命令实现相关功能。上位机软件主窗口如图 4 所示。

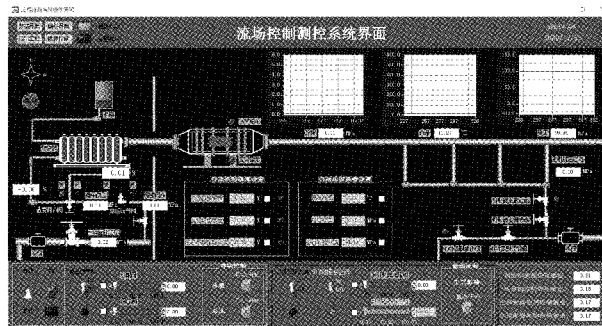


图 4 上位机软件主界面

上位机除主界面外,还包括多个子窗口,用于实现传感器系数标定、曲线实时显示和查看历史数据等功能,通过界面功能划分方便操作人员进行监控和指令下达。流场控制

系统上位机设计了多个子窗口,包括数据界面显示、参数界面设置、曲线界面等。例如新建数据显示界面子窗口 ValueShowPanel,通过函数 LoadPanel (0, "ValueShowPanel. uir", VALUEPANEL) 加载子窗口至内存中,函数 DisplayPanel (valuepanel) 用于显示子窗口界面;子窗口关闭调用 HidePanel (valuepanel) 实现,此时子窗口并未从内存中释放;当上位机进程需要结束时,通过函数 DiscardPanel (valuepanel) 完成内存释放。数据列表显示界面如图 5 所示。



图 5 数据列表显示界面

4 测试结果分析

测试阶段,手动模式下通过上位机界面控制滑动调节模块即可对可控执行器进行开环调节,控制原理与风洞改造前一致,系统能够实现正常的手动调节工作,增加了数据存储功能,提高了信息化程度。

自动模式下,主要对设定的目标值主引射器压力、总温值、总压值稳定状态的结果进行分析。试验中设置主引射器压力 1 MPa,总温 180 ℃,总压 2 MPa。按照操作流程,主引射器首先开启,静压达到试验要求的稳定值后,总温、总压启动自动调节功能,完成试验后,首先关闭总温、总压,使压力和温度降至一定范围后,关闭主引射器,系统测试结束。

4.1 稳态分析

试验过程中主引射器压力曲线如图 6 所示,虚线表示达到稳态值±5%的区间范围。主引射器控制模块自起始点(22.6, 0.104)开始调节,达到坐标点(30.2, 0.95)后压力曲线收敛,并快速达到稳定状态并保持,调节时间 T_s 约为 7.6 s;坐标点(153.6, 1)测试结束,关闭主引射器调节阀。

试验过程中总温曲线如图 7 所示。总温控制模块自起始点(25.2, 34.45)开始调节,达到坐标点(71.94, 189)总温达到稳态值±5%的范围并收敛,直至温度输出达到稳定状态,调节时间 T_s 约为 46.7 s。

总压曲线如图 8 所示。总压控制模块自起始点(26, 0.1033)开始调节,达到坐标点(70.64, 2.1)后总压达到稳定状态±5%的范围并收敛,调节时间 T_s 约为 46 s。

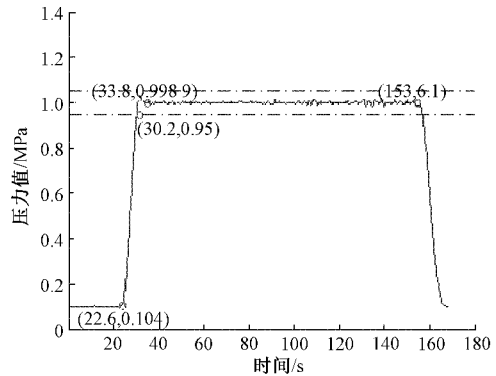


图 6 主引射器自动调节压力曲线

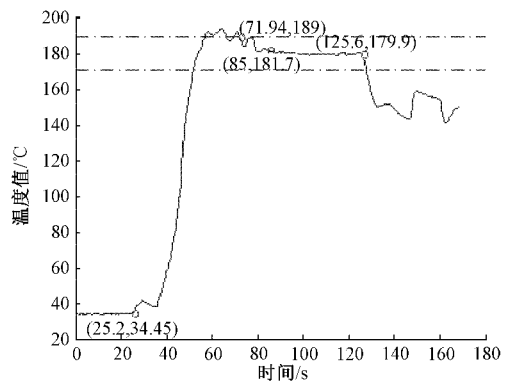


图 7 总温自动调节温度曲线

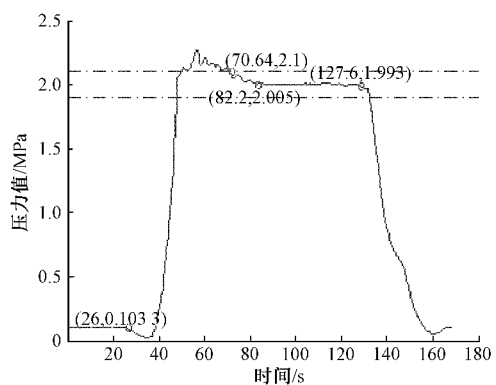


图 8 总压自动调节压力曲线

4.2 精度分析

流场测试精度分析通过贝塞尔法计算标准偏差 σ 的方法来表征控制系统精度^[18],如式(1)所示; σ 越小表明测量分散度越小,系统控制精度越高。

$$\sigma = \sqrt{\frac{\sum_{i=1}^n (X_i - \bar{X})^2}{n-1}}, n \geq 10 \quad (1)$$

系统数据采样频率 5 Hz,主引射器压力值选取图 6 坐标点(33.8, 0.9989)至(153.6, 1)数据区域,采集点 n 值为 600;总温温度值选取上图 7 坐标点(85, 181.7)至(125.6, 179.9)数据区域,采集点 n 值为 204;总压压力值

选取上图 8 坐标点(82.2, 2.005)至(127.6, 1.993)数据区域,采集点 n 值为 228。平均值 \bar{X} 与标准差实验结果如表 1 所示。

表 1 实验结果

标签	主引射器压力值	总温值	总压值
n	600	204	228
\bar{X}	0.999 8	180.092 4	2.003 1
σ	0.004 3	0.492 9	0.003 5

上述稳态及精度分析表明,新的流场控制系统自动调节功能响应速度快,调节时间远优于手动模式;控制精度高,流场稳定阶段引射器压力值、总温值、总压值波动量较小,产生的流场稳定,满足设计需求。

5 结 论

本文系统已在某小型高速风洞试验区域成功部署,大量测试表明,基于 LabWindows/CVI 的流场控制系统运行稳定可靠,OPC 通信有效解决数据采集过程中的大吞吐量数据传输问题,响应性能良好;自动控制模式下,下位机 PLC 能够迅速达到设定的稳定状态,解决了改造前手动调节的不便性,大幅增加了流场稳定的有效时间、提高了流场品质。新的流场控制系统对于减少试验过程中的气源损耗,提高小型风洞控制领域的自动化水平,具有较高的参考价值和实用价值。

下一阶段,我们将对风洞流场控制系统做进一步优化,并增加数据采集分析功能,为后续的故障诊断、健康监测等提供支撑。

参考文献

- [1] 王建新,隋美丽. LabWindows/CVI 虚拟仪器测试技术及工程应用[M]. 北京:化学工业出版社,2011.
- [2] 陈策富,谢利理,李永立. 航空机轮加载测控系统软件的设计与实现[J]. 计算机测量与控制,2012(4): 1129-1131.
- [3] 李亚,邵引平. 基于 LabWindows/CVI 的远程接口单元测试系统软件设计[J]. 计算机测量与控制,2020,28(7): 148-152,157.
- [4] 薄志峰. 基于 LabWindows/CVI 的电动舵机自动化测试系统设计[J]. 国外电子测量技术,2015,34(5): 66-69.
- [5] 姚惠元. 基于 labVIEW 与 OPC 的冰风洞主控系统设计[C]. proceedings of the 中国空气动力学会测控技术专委会四次学术交流,2013.
- [6] 杜文鹏. 基于 LabWindows/CVI 的风洞数据处理系统的设计与实现[D]. 成都:电子科技大学,2012.
- [7] 杨广利,杨日杰,韩建辉,等. 基于 LabWindows/CVI 的主动声响信号发射机设计[J]. 国外电子测量技术,2017,36(6): 81-84.
- [8] 殷凤龙,梁小冬,廖洪波,等. 基于 LabWindows/CVI 的单关节机械臂随动控制研究[J]. 机械,2019,46(5): 37-41.
- [9] 汤亚玲,郭健,张学锋,等. 基于 OPC 技术的轮毂装配线监控系统[J]. 计算机系统应用,2021,30(3): 103-109.
- [10] 刘俊杰,吴青峰,朱志斌. 基于 PLC 的速调管测试台架控制系统设计[J]. 工业控制计算机,2018,31(10): 142-143.
- [11] 徐春梅,牛斌. LabWindows/CVI 在开发测试软件中的应用[J]. 电子测试,2020(2): 66-68.
- [12] 江司珂,郭雨,陈俊,等. 基于多线程技术的关口电表检测系统的研究[J]. 电测与仪表,2018,55(22): 129-134.
- [13] 王嘉,范蟠果,毛恒. 一种改进型多线程机制在数据采集系统中的应用[J]. 计算技术与自动化,2019,38(4): 42-46.
- [14] 李晔,姚全珠,王战敏. 基于 LabWindows/CVI 的多线程实时监控系统的研究与开发[J]. 电脑与信息技术,2008,16(6): 63-65.
- [15] 李一滨,李蓓. 基于 LabWindows/CVI 风洞数据采集系统的应用研究[J]. 测控技术,2011,30(11): 19-21.
- [16] 谭三,杨兴林,赖青贵. 基于 LabWindows/CVI 中 DataSocket 实现分布式系统[C]. proceedings of the 测控,计量与仪器学术年会,2004.
- [17] 单体强,陈雷,张万发. LabWindows/CVI 多线程技术的应用研究[J]. 电子设计工程,2012,20(15): 52-54,58.
- [18] 陈佳,袁朝辉,郭强,等. 射流管伺服阀前置级的动态流场分析[J]. 仪器仪表学报,2017,38(7): 1731-1737.

作者简介

王政,工学硕士,工程师,主要研究方向为测控检测技术、控制算法分析与应用技术研究等。

E-mail:wangzh2009@126.com