

# IRIG 标准颤振信号的在线监测方法研究

祁晓鹏, 朱洪翔, 吉陈力

(中国飞行试验研究院, 西安 710089)

**摘要:** 颤振试飞是飞行试验中的一项重要课题; 颤振参数的实时在线显示为颤振参数在线校准的最重要环节, 基于此提出了一种基于 IRIG 标准的颤振信号的在线实时监测方法; 首先对飞行数据的 IRIG 记录标准进行了研究, 并介绍了其中颤振参数的记录格式; 然后对 BCB 界面化设计和 Matlab 数据分析的混合编程进行了研究, 利用混合编程和多线程的组合方式, 完成了网络数据提取、数据包获取、通道数据获取与显示和 FFT 变换与显示; 通过对标准信号的在线实时监测, 验证了软件的正确性和可行性, 满足颤振参数校准的需求, 最终实现了对颤振信号的在线监测。

**关键词:** IRIG; BCB; Matlab; 混合编程; 在线监测

## On-line Monitoring for IRIG Standard Chatter Singal

Qi Xiaopeng, Zhu Xongxiang, Ji Chenli

(Chinese Flight Test Establishment, Xi'an 710089, China)

**Abstract:** The flutter test is an important topic in flight test. The real-time online display of the flutter parameters is the most important part of the on-line calibration of the flutter parameters. Based on this, an on-line real-time monitoring method based on IRIG standard is proposed. Firstly, the standard of IRIG recording of flight data is studied, and the recording format of the flutter parameters is introduced. Then the BCB interface design and Matlab data analysis of mixed programming are studied, using the combination of programming and multi thread combination, completed the network data extraction, data packet capture, channel data acquisition display and display FFT. The accuracy and feasibility of the software are verified by on-line real-time monitoring of the standard signal, and the requirement of the flutter parameters calibration is met.

**Keywords:** IRIG; BCB; Matlab; mixed programming; on-line monitoring

## 0 引言

颤振飞行试验是目前飞机试飞的重要科目, 通过试飞验证, 最终确定飞机的颤振特性和颤振余量<sup>[1]</sup>。目前的颤振数据分析多是事后处理, 给前期的参数校准带来了诸多的不便。本文利用 BCB 的界面化和 Matlab 的数据分析进行混合编程, 对颤振信号进行在线提取和分析, 利用快速傅立叶变换(FFT)<sup>[2]</sup>对颤振信号进行分析, 解算出相应的模态参数。经过对颤振信号进行在线的显示和监测, 验证了本文的方法和所编写的软件的可行性。

## 1 IRIG 记录标准

基于 IRIG<sup>[3]</sup>标准的记录器以太网端口是一个多种用途的 IEEE802.3u (即“以太网”和“100 基时”)端口, 具有很多不同的功能, 包括: 系统监视和配置、控制、状态、升级和进入某些确定的运行环境、数据传送等, 这些端口都符合 IRIG106 网络标准。记录器能将一个控制网络端口中的 TCP 端口当作一个记录数据源, 而不需备用的 mR-X01 多路传输适配器。端口 5510 至端口 5519 都可以用作于数据端口。这些数据端口同样可以用作于提取记录中的数据。

IRIG106 数字记录标准的数据是由基本包元素构成的, 其中包元素数据包括: 包头、包体和包尾, 还有一个可选的第二包头。包头: 对包内的数据类型、长度、状态信息等进行详细的说明。其中包的同步标示为固定值 0xEB25。包体: 不同的数据类型在包体中有不同的数据结构, 例如模拟量的包体包括了通道信息(通道号和通道总数)和通道对应的有效数据。包尾:

用于包的数据校验。模拟量基本包元素的数据格式如表 1 所示。

表 1 数据包信息

|                       |             |    |
|-----------------------|-------------|----|
| Packet Sync Patten    | 2 字节        | 包头 |
| Channel ID            | 2 字节        |    |
| Packet Length         | 4 字节        |    |
| Data Length           | 4 字节        |    |
| Data Type Version     | 1 字节        |    |
| Sequence Number       | 1 字节        |    |
| Packet Flags          | 1 字节        |    |
| Data Type             | 1 字节        |    |
| Relative Time Counter | 6 字节        |    |
| Header Check Sum      | 2 字节        |    |
| Channel Specific Data | 通道总数 * 2 字节 | 包体 |
| Data 1.....Data n     | 模拟量为 2 字节   |    |
| .....                 |             |    |
| Data 1.....Data n     |             |    |
| Data Checksum         | 4 字节        | 包尾 |

其中模拟量包体部分分为两块: 通道信息和通道对应的有效数据。比如通道数为 3, 则通道信息为 4001 0300 4003 0300 4006 0300, 对应分别是第一通道、总通道数、第三通道、总通道数、第六通道、总通道数。包头信息包含了判断的所有信息, 通过包头可以获取整个包的长度, 也可以获取包的数据长度, 然后通过 Channel Specific Data 来确定对应的通道号与通道数据, 将不同通道的数据分别存储在不同文件中, 以便于后续对数据进行回放。

## 2 软件编程

### 2.1 混合编程

Matlab 具有强大的数据处理功能, 但 Matlab 程序的执行

收稿日期: 2015-10-27; 修回日期: 2015-11-23。

作者简介: 祁晓鹏(1987-), 男, 河北衡水人, 硕士, 工程师, 主要从事飞行试验、测试系统方向的研究。

效率非常低, 而且其平台上开发的程序不能脱离本身的运行环境; C++builder 具有强大的用户界面开发功能, 程序执行效率高, 但程序编写相对繁琐, 结合两者的优势可以快速开发出功能强大的应用系统。Matcom 是 Mathtool 公司开发的第三方控件, 它可以将 *m* 脚本文件和 *m* 函数转化为具有相同功能的 .h 和 .cpp 文件。相比 Matlab 自带的编译器, 使用 Matcom 转化代码要简单许多。而且使用 Matcom 工具可以方便生成动态链接库文件 (dll)、.h 头文件和可执行文件 .cpp, 这样程序可以脱离 Matlab 环境, 提高了代码的复用率和执行速度<sup>[4]</sup>。本文将利用 Matlab 中的快速傅立叶变换编写 .m 脚本文件, 通过 Matcom 将 .m 脚本文件转换为 .h 和 .cpp 文件, 最后完成傅立叶变换计算。

### 2.2 程序内容

软件内程序包括网络数据获取、数据包获取、通道数据获取和 FFT 变换。

网络数据获取程序如下:

```
allcount=ClientSocket1->Socket->ReceiveLenght(); //网络包数据长度
tes=ClientSocket1->Socket->ReceiveBuf((Char*)recdatabuf,allcount); //读取网络数据,数据存入 recdatabuf 数组中
```

网络数据获取完毕后, 需要提取出其中完整的数据包, 才能正确的提取出通道数据。

数据包提取程序如下:

```
if(k1>=12)
{
packetdata[k1]=recdatabuf[i];
k1++;
if(packetdata[3]!=0&&packetdata[5]!=0&&packetdata[7]/256!=33)
{
yupan=true;//BOOL 控制信息
}
if(k1==15)
{
packetlength=packetdata[2]/2;
//数据包长度
}
}
if(k1<12)
{
packetdata[k1]=recdatabuf[i];
k1++;
}
}
当提取出完整的数据包后需要对要显示的数据进行提取。通道数据提取程序为:
```

通道数据提取程序为:

```
if(packetdata[1]==channel1p&&.J1->Checked)
{
for(intk2=12+channelnum[0]*2;k2<packetlength-2;k2++)
{
if((k2-12-channelnum[0]*2)%channelnum[0]==channel1show&&.J1->Checked)
{
array1c[nNum]=packetdata[k2];
nNum++;
}
}
}
```

当通道数据达到一定数目的时候然后对数据进行 FFT 变换。

```
Mm psdcall=mypsd (array1, 1024, 1024);
```

//对 array1 中数据进行 FFT 变换

以上程序从数据提取到最后的数据显示和 FFT 变换进行了详细的介绍, 完成了整个软件的工作流程。

### 3 采集软件

基于 C++builder 与 Matlab 混合编程实现了对基于 IRIG106 标准的记录器进行在线监测的方法。软件的整体分为操作区和显示区: 其中操作区的功能为设置通道信息和显示通道选择功能, 分别为控制区、板卡通道信息区、数据回放、系统时间和采样率, 其中控制区包括颤振信号的实时显示开始、停止、数据回放和关闭程序的按钮; 板卡通道区包括通道 ID 设置、板卡通道、板卡通路、实时数据通路控件, 其中 ID 为板卡对应的 ID, 例如: 0x9014, 板卡通道需要同相应板卡的设置相同, 以便于正确选择相应的通道数据, 板卡通路和实时数据通路用于选择实时显示相应通道数据; 数据回放区包括数据回放起始时间、结束时间控件; 系统时间显示设备的系统时间; 采样率为板卡的采样频率, 用于 FFT 变换。显示区分别显示选择通道实时数据和通道数据的 FFT 变换, 可以实现多路数据的在线显示。

本文应用了 C++builder 的网络编程<sup>[5]</sup>和多线程功能完成对本软件的设计。本文中所实现的软件的工作流程如图 1 所示。

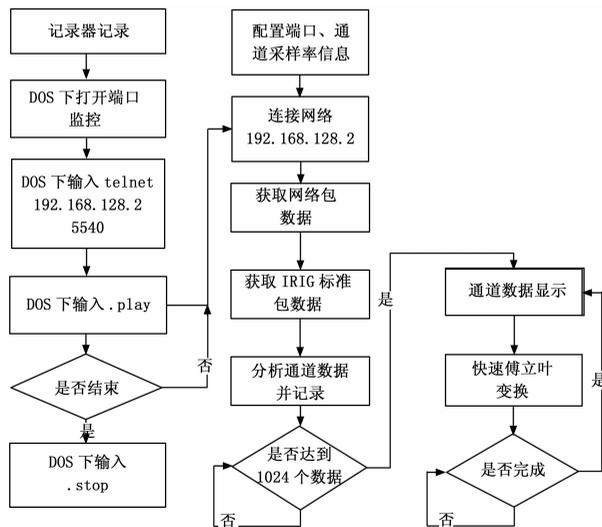


图 1 工作流程图

首先完成 Telnet 命令输入, 使记录器处于监控状态, 然后打开模拟量监控软件, 对相应的通道进行在线监测, 同时对监测的通道进行快速傅立叶变换。

### 4 软件验证

为了验证软件的可行性, 本文进行了大量的实验, 利用信号发生器分别产生正弦波、方波、三角波等信号, 然后用记录器对产生的信号进行记录, 同时用模拟量监控软件对相应通道进行监测, 图 2 为监测软件的不同信号波形图和对应的 FFT 图。

### 5 结束语

本文利用 C++builder 和 Matlab 混合编程实现了对基于

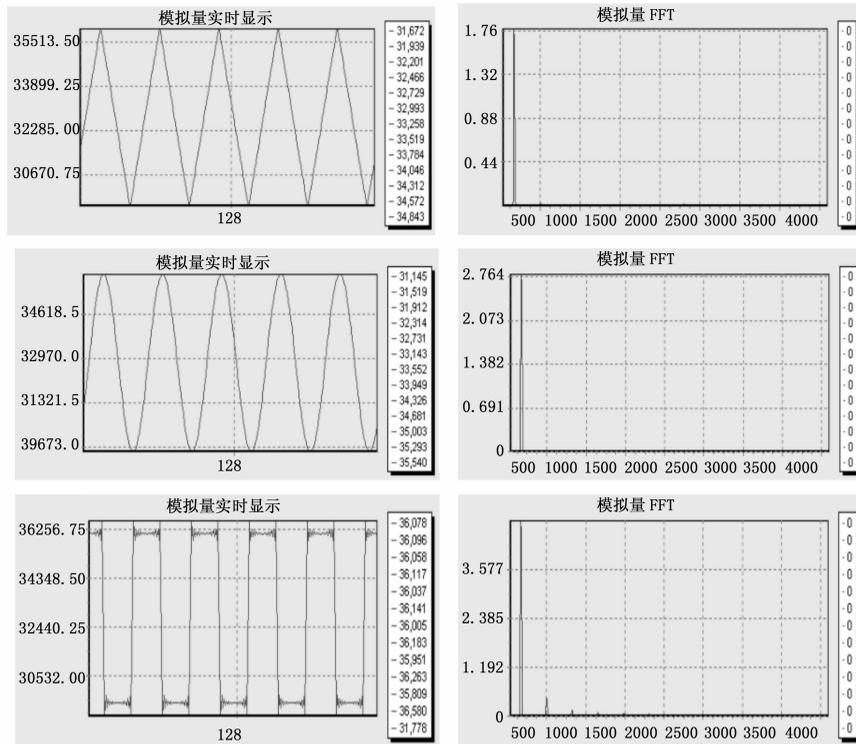


图 2 信号波形图和 FFT

IRIG106 标准记录器中模拟量的在线监测, 验证了对标准记录器在线监测的可行性, 为后续的其他类型信号的在线监测提供了依据。对标准记录器的信号进行在线监测, 通过实时的观看可以提供很多便利条件, 提高日常的工作效率。

参考文献:

[1] 管 德. 飞机气动弹性力学手册 [M]. 北京: 航空工业出版社, 1994.

[2] 王 济, 胡 晓. Matlab 在振动信号处理中的应用 [M]. 北京: 中国水利水电出版社, 2006.

[3] RCC Document 106-07. Telemetry Standard [S]. September 2007 (9).

[4] Mathews J H, Fink K D. 数值方法 [M]. 北京: 电子工业出版社, 2002.

[5] 袁 辉. C++Builder 网络编程核心技术 [M]. 北京: 机械工业出版社, 2003.

(上接第 12 页)

表 1 软件测试问题

| 测试输入(过程)  | 测试问题(结果)   |
|---|--|
| 主控箱模型发送数据, 输入 ID = 0x3c1, dlc = 8, data0 = 0x66, data1 = 0x19, data2 = 0x00, data3 = 0x00。查看底盘车状态界面。    | 期望结果: 底盘车横倾角值 15.02, 对应表盘指针向左打满, 报警指示灯变红。实测结果: 表盘指针超出左边界, 报警指示灯未变红。  |
| 主控箱模型发送数据, ID = 0x4c0, dlc = 8, data0 = 0x00, data1 = 0x00, data2 = 0x00, data3 = 0x00。查看显示终端控制输出界面。    | 期望结果: 控制输出界面吊机回转“左转”指示灯变绿。实测结果: 吊机回转“左转”指示灯不亮, “右转”指示灯变绿, 与通讯协议不符。   |
| 显示终端进入手柄标定界面, 按键操作输入零位 = 128, 死区 = 50, 最小值 = 50, 最大值 = 200, 按 F8 进行数据标定。期望结果: 用户面板查看到的标定信息与通讯协议相关定义相一致。 | 实测结果: 用户面板查看标定信息为 ID = 0x621, dlc = 8, data0 = 0x20, data1 = 0x04, data2 = 0x10, data3 = 0x01, data4 = 0x80, data5 = 0x32, data6 = 0x32, data7 = 0xC8, 与通讯协议相关定义不一致。 |

能够提供极端或异常的测试环境, 增强了软件测试的充分覆盖性, 缩短了软件测试周期并提高了测试效率。

参考文献:

[1] 巫 莉, 黄江峰. 电气控制与 PLC 应用 [M]. 北京: 中国电力出版社, 2008.

[2] 周凤余, 鲁守银. CAN 总线系统智能节点设计与实现 [J]. 微计算机信息, 1999, 15 (6): 9-12.

[3] 古 乐, 史九林. 软件测试技术概论 [M]. 北京: 清华大学出版社, 2004.

[4] Hawkins J, Howard R B, Haung V. Nguyen automated real-time testing for embedded control system [J]. IEEE, 2002, 6 (03): 647-652.

[5] 唐 科. 软件仿真环境下的嵌入式软件测试的研究 [D]. 成都: 电子科技大学, 2005.

[6] 钟德明, 刘 斌, 阮 镛. 嵌入式软件仿真测试环境软件体系结构研究 [J]. 北京航空航天大学学报, 2005, 31 (10): 1130-1134.

[7] 宋 文, 于林宇, 刘 军. 通用嵌入式软件测试环境在武器装备软件测试中的应用 [J]. 测控技术, 2008, 24 (10): 65-68.

[8] 权 巍, 王俊杰, 邢忠宝. ARINC429 总线通讯软件测试环境的构建 [J]. 计算机测量与控制, 2008, 16 (12): 1783-1785.