

# 基于 WPF 技术的试飞数据实时监控系统设计研究

陈咏梅\*

(中国商飞民用飞机试飞中心, 上海 201323)

## 摘要:

立足于当前大型民用飞机试飞模式,集合数据处理及实时监控特点,在 Visual Studio 环境下,利用 WPF 软件编程技术,构建了基于数据层、业务逻辑层、表示层的试飞数据实时监控系统。系统利用多种显式控件,充分模拟了试飞数据监控状态,通过 WPF 数据驱动模型及合理的线程资源调配,实现了对不同类型数据的实时监控、分析、存储和回放,为后续民机试飞测试技术的创新提供了参考依据。

关键词: WPF;线程;实时监控;实时分析;实时回放

中图分类号: V217+.2

文献标识码: A

OSID:



## 0 引言

试飞数据实时监控系统作为民机试飞过程中的重要决策性支持系统,主要功能是利用可视化控件实时显示试飞数据,反应飞机当前的试飞状态、试验效果等信息,为指挥人员、保障人员、飞机设计人员等提供决策依据。因此,构建实时性能卓越、功能丰富的试飞数据实时监控系统对于增强试飞安全、改进试飞方法策略、提高试飞效率等具有重要意义。

在当前试飞模式下,国内的主流试飞机构往往采用 Labview、DataView、VC++ 等环境开发实时监控程序,在试飞过程中显示试飞数据,而不能进行复杂分析和处理<sup>[5]</sup>,对于试验点有效性的确认以及飞机安全性状态的判断需要技术人员进行二次解析、分析,这一重复性步骤降低了试飞效率,不利于缩短试飞周期。

本文立足于民机试飞模式,从试飞安全监控、实时分析等两个角度,基于数据、逻辑、显示三层架构,利用 WPF (Windows Presentation Foundation) 技术设计了试飞数据监控系统。该系统采用多线程同步技

术在传统监控形式上集成了数据实时存储、实时分析、实时回放、信息标记与共享等功能,系统特点包括:

- 1) 应用 WPF 技术能够逼真的模拟 PFD、MFD 等机载画面;
- 2) 三层架构易于系统维护;
- 3) 业务逻辑、运算、分析与显示相互独立,实时性能得到较大提高。

## 1 系统结构

试飞数据实时监控系统采用数据层、业务逻辑层与显示层的三层结构,其原理如图 1 所示。

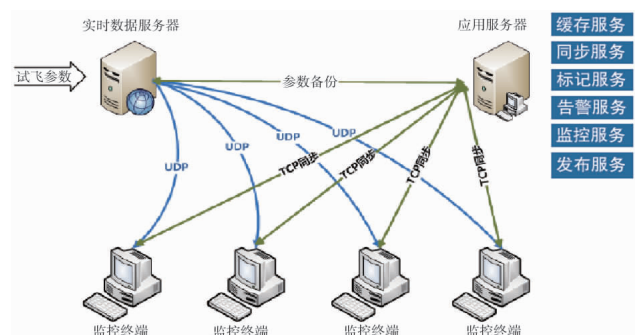


图 1 系统结构原理图

[基金项目] 17511105000,大型民机试飞数据高速实时处理及异常状态检测分析平台,2017.07-2019.06。

\* 通信作者: E-mail: chen Yongmei@comac.cc

引用格式: 陈咏梅. 基于 WPF 技术的试飞数据实时监控系统设计研究[J]. 民用飞机设计与研究, 2019(3):55-59. CHEN Y M. Design of Real-time Flight Data Monitoring System Based on the WPF[J]. Civil Aircraft Design and Research, 2019 (3):55-59(in Chinese).

在数据层,试飞参数通过网络接口传入实时数据服务器,由实时数据服务器经数据解析、封装,然后以 UDP 广播的方式发送至监控终端和应用服务器。应用服务器作为应用层主要设备,常驻运行数据备份缓存、同步、标记与共享、告警、实时分析等服务线程,同时与监控终端之间建立面向连接的 TCP 协议,以响应监控终端请求完成数据实时回放、分析等功能。主要服务功能如下:

1) 备份缓存服务:负责接收数据层发送的数据,并将数据以二进制流的格式写进数据文件以作备份,以响应监控终端的请求数据回放的请求。

2) 同步服务:负责实现监控终端数据的实时回放功能,同步服务为每一台接入系统的监控终端建立 10 min ~ 15 min 的数据缓冲池,当接收到监控终端的回放请求时,同步服务根据监控终端的请求时刻从缓冲池中选择相应时长的试飞数据、封装后传输至监控终端,若请求时刻不在缓冲池中,则同步服务会从备份二进制文件中读取相应的数据发送给监控终端进行回放。

3) 实时分析:负责对试飞数据进行实时分析,并将分析结果发送给监控终端进行实时显示;

4) 告警服务:负责调用告警事件对试飞数据进行告警判断,同时结果发送给实时监控终端。

表示层为运行于实时监控终端的监控画面程序,采用 WPF 技术,利用 C# 编程语言编辑开发,数据的显示方式包括动态曲线、双态灯、数值显示、虚拟仪表、刷新列表。

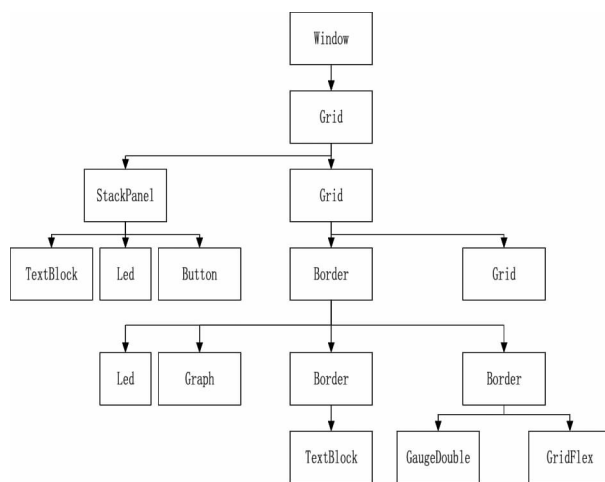
## 2 系统实现

### 2.1 实时监控程序开发

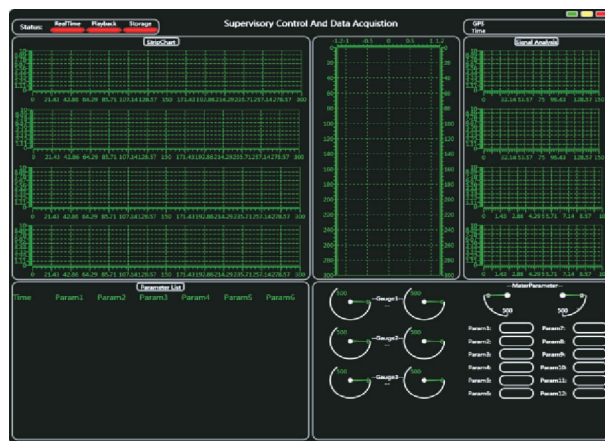
实时监控程序通过各类显示控件实时刷新数据,包括前台 UI 和参数驱动两个部分。监控程序的前台 UI 基于 WPF 中的 XAML 语言<sup>[4]</sup>,采用其矢量图绘制引擎实现,本系统主要显示数据的趋势、具体数值、参数状态等,由绘制的曲线绘制框、条图仪、虚拟仪表、数值显示框以及结合 Grid、StackPanel、DockPanel 等布局控件组成,其结构如图 2 所示:

利用多线程和 DataBinding 可实现参数与现实控件的绑定与驱动。处理原则如下:

1) 数据显示刷新率不大于 60 Hz 采用 Data Binding:利用数据驱动显示控件的依赖属性实现,其模型如图 3 所示。



(a) UI 逻辑树图



(b) 监控画面程序 UI

图 2 前台 UI 和参数驱动

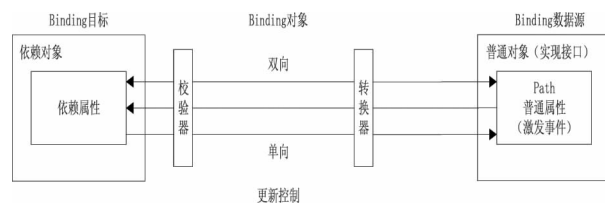


图 3 WPF 数据驱动模型

程序语言可描述为: `DisplayControl. SetBinding ( DependencyProperty dp, Bindingbase binding )`, 其中:

- (1) DisplayControl——显示控件,作为 Binding 目标;
- (2) DependencyProperty——以 DependObject 为宿主的依赖属性,借助其 SetValue 及 GetValue 方法进行读取和写入;
- (3) Bindingbase——用以声明 Binding 的数据源。

2)数据显示刷新率大于60Hz或曲线动态绘制采用多线程,利用委托统一主辅线程显示数据,流程如图4所示。

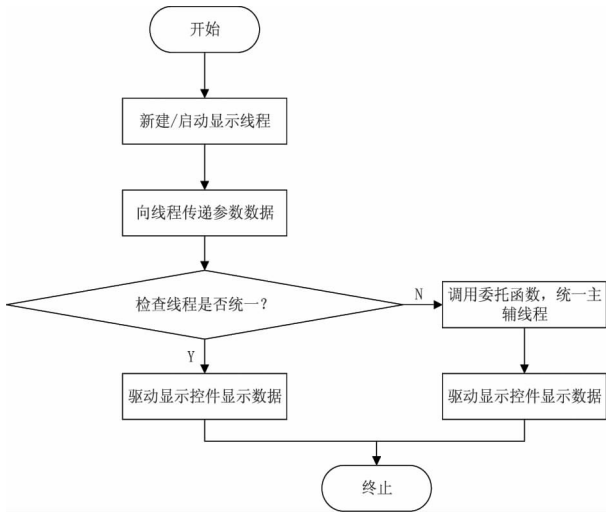


图4 多线程显示

### 2.2 备份缓存及同步服务

如前所述,数据备份缓存及同步服务主要是利用数据文件及内存实时存储数据,以作为实时回放的数据源,实现流程如图5所示。

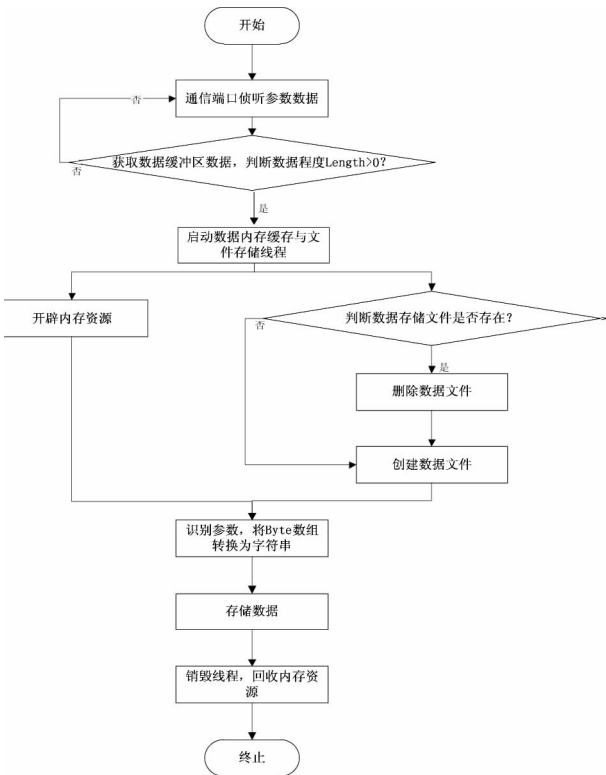


图5 试飞数据实时备份存储流程

首先从数据缓冲区中提取数据,并判断数据数组长度;当获取的数据长度大于0(即缓冲区中有数据)时,提取数据字节数组;最后将提取的字节写入数据存储文件。

### 2.3 实时回放

实时回放是指在试飞数据实时监控的过程中监控人员可对历史数据进行实时回看,以检查数据有效性和飞行异常,回放过程与实时监控链路相对独立,回放结束后可切换至实时监控状态。根据实际需求,本系统支持5 min~10 min以及全时间历程的数据回放。系统工作时,应用服务器侦听实时监控终端的回放请求,根据回放时间刻度从内存数据文件中提取历史数据完成回放,其工作流程如图6所示。

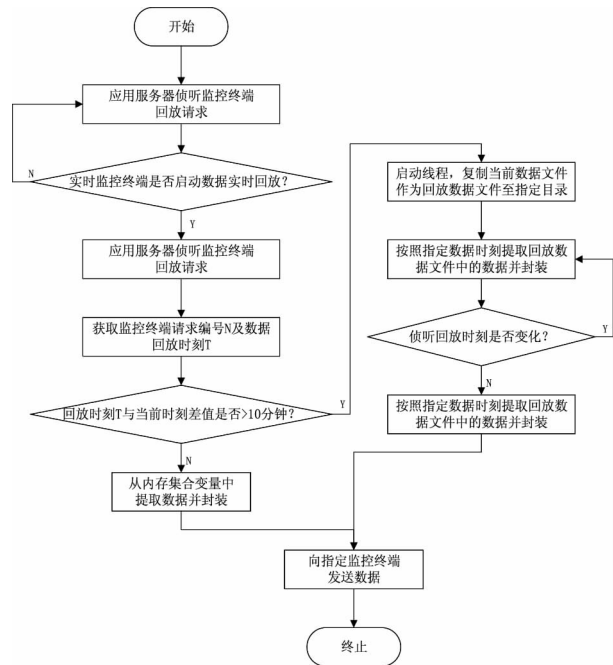


图6 试飞数据实时回放服务流程

### 2.4 实时分析

本系统基于实际试飞需求,集成了快速傅里叶变化(FFT)和数字滤波等典型数据分析方法,同时对实时分析状态下的系统性能进行了分析。

#### 2.4.1 快速傅里叶变换(FFT)

对于复数序列  $x_0, x_1, \dots, x_{n-1}$ , 其离散傅里叶变换可以表示为:

$$y_j = \sum_{k=0}^{n-1} e^{-\frac{2\pi ijk}{n}} x_{kj} = 0, 1, 2, \dots, n-1 \quad (1)$$

FFT 利用  $e^{-\frac{2\pi ijk}{n}}$  的周期性及对称性,可表示为:

$$x(k) = \sum_{r=0}^{n/2-1} x_1(r) W_{n/2}^{kr} + W_n^k \sum_{r=0}^{n/2-1} x_2(r) W_{n/2}^{kr} \quad (2)$$

式中:

$$W_n^k = e^{-\frac{2\pi k i}{n}} \quad (3)$$

### 2.4.2 数字滤波

本系统采用 Butterworth 数字滤波器,主要涉及参数包括滤波器阶数和截止频率,可通过式(4)计算得出<sup>[2]</sup>。

$$\omega_c = \frac{\omega_s}{2N \sqrt{(10^{\frac{R_s}{10}} - 1)}} \quad (4)$$

式中:

- $\omega_c$ ——截止频率;
- N——滤波器阶数;
- $\omega_s$ ——阻带边界频率;
- $R_s$ ——阻带衰减(dB)。

实时分析过程利用多线程技术实现,应用服务器根据实时监控终端请求完成数据计算过程,同时将计算结果发送给订阅的实时监控终端,实现流程如图 7 所示。

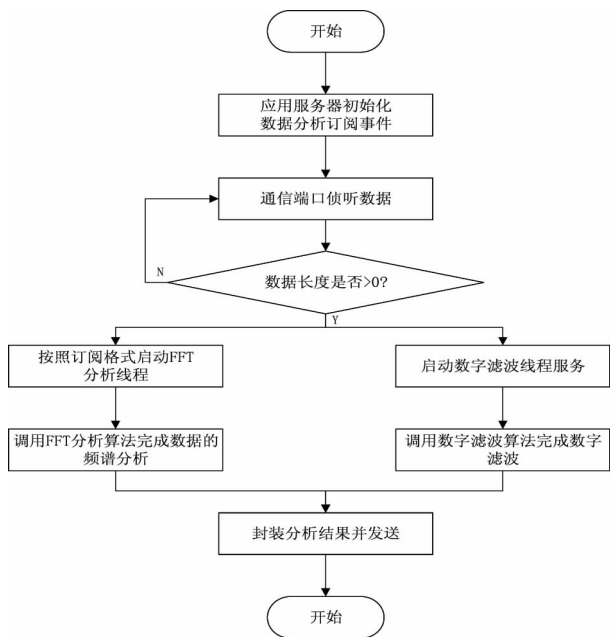


图 7 数据实时分析流程

## 3 试验

试验数据源为仿真数据源,采用 UDP 广播通信方式,每隔 100ms 发出 200 个参数,应用层格式协议如图 8 所示。

绝对时间戳	连接字符	标识字符	分隔字符	参数1	连接字符	标识字符	分隔字符	参数2	.....	结束符
-------	------	------	------	-----	------	------	------	-----	-------	-----

图 8 基于 UDP 的应用层协议格式

其中:

连接字符为“&”;标识字符为参数名称;分隔字符为“,”;结束字符为“#”。

因此,应用层数据可表示为“12: 23: 59: 242&Time, 7. 6284&Param1, ..., 4. 6790&Param200 #”。实时监控程序与应用服务器在收到数据后,首先将 Byte 数组按照约定编码格式(即 Unicode)转换成字符串<sup>[3]</sup>,然后以分隔符(“,”)将字符串拆分为参数数组,最后对该数组进行轮循,针对每一项按照连接字符(“&”)拆分并根据参数标识符定位、绑定参数。实时监控程序运行状态如图 9 所示。

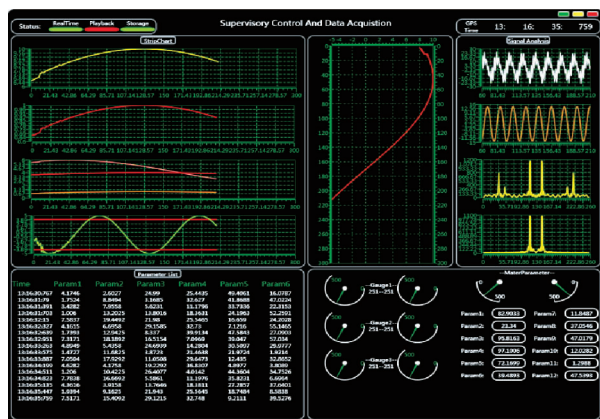


图 9 实时监控程序运行状态

试验中用于实时分析的参数由三个正弦信号复合而成,频率分别为 0.5、1.5,幅值为 10、5、15,如式(5)所示:

$$f(t) = 10 * \sin(2\pi t * 0.5 + \frac{\pi}{3}) + 5 * \sin(2\pi t * 1 + \frac{\pi}{3}) + 15 * \sin(2\pi t * 1.5 + \frac{\pi}{3}) \quad (5)$$

分析结果如图 10 所示,表 1 为原始信号,表 2 为滤波后幅值为 15 的正弦信号,表 3、表 4 分别是原始信号及滤波后信号的 FFT 分析结果。根据式(6)、(7)计算信号频率成分及对应幅值。

$$f = n \cdot f_s / N \quad (6)$$

$$A = A_f \cdot 2/n \quad (7)$$

式中:

- f——信号频率;
- n——数据量;
- $f_s$ ——FFT 计算后的频率;
- N——FFT 幅频谱数据点数,这里取值 512;
- A——信号幅值;
- $A_f$ ——FFT 计算后的幅值。

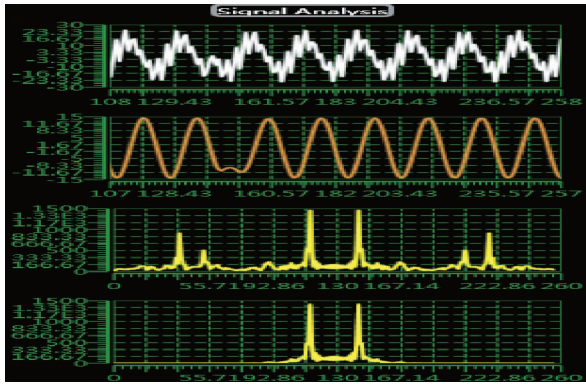


图 10 数据实时分析结果

根据 FFT 分析出的双边幅频谱可见,FFT 分析过程准确地识别了信号中的频率成分。系统延时可利用同步状态下的绝对时间差计算  $\Delta t = |t_1 - t_2| = 77\text{ms}$ ,其中  $t_1$  是接收数据的绝对时间, $t_2$  是完成处理后的绝对时间。

## 4 结论

本文数据层、业务逻辑层、表示层三层架构,采用 WPF 技术设计了试飞数据实时监控系统。通过

试验,系统模拟了试飞状态,以 UDP/IP 为传输协议,在保证系统实时性能的前提下,合理的调配任务线程实现了在实时监控过程中对试飞数据的实时存储、备份、分析和回放等功能,为试飞数据监控模式技术创新提供了新的参考依据。

### 参考文献:

- [1] 陈亚勇,等. MATLAB 信号处理详解[M]. 北京:人民邮电出版社,2001.
- [2] 万永革. 数字信号处理的 MATLAB 实现[M]. 北京:科学出版社,2012.
- [3] 郑阿奇. Visual C#网络编程[M]. 北京:电子工业出版社,2011.
- [4] 刘铁猛. 深入浅出 WPF[M]. 北京:中国水利水电出版社,2010.
- [5] 柳昌庆,王启广. 测试技术与实验方法[M]. 徐州:中国矿业大学出版社,2001.

### 作者简介

陈咏梅 女,学士,高级工程师。主要研究方向:机载测试。  
E-mail: chen Yongmei@comac. cc

## Design of Real-time Flight Data Monitoring System Based on the WPF

CHEN Yongmei \*

(Flight Test Center of the COMAC, Shanghai 201323, China)

**Abstract:** Grounded on current civil aircraft flight test mode, and combined with the features of the data processing and monitoring, this article uses the WPF software programming techniques in visual studio to build a flight data real-time monitoring system based on the structure of data layer, business logic, presentation layer. The routine supervision, analysis, storage, and playback of different type of data completed with WPF data driven model and proper thread allocate source based on a variety of explicit controls, which offers reliable reference for the upcoming revolution of civil flight test technique.

**Keywords:** WPF(Windows Presentation Foundation); thread; real-time monitoring; real-time analysis; real-time playback

\* Corresponding author. E-mail: chen Yongmei@ comac. cc