

高精度原子钟频率稳定度测试系统的软件设计

谢彦^{1,2} 蔡勇¹

(1. 中国科学院上海天文台, 上海 200030; 2. 中国科学院研究生院, 北京 100049)

提 要

结合高精度原子钟频率稳定度测试系统的工作原理,介绍了测试系统的软件设计。基于软件的需求分析,运用对象模型方法,重点分析了系统的框架结构、自动测试流程和数据处理及其算法。实验结果证明,该软件可以实现测试过程的稳定和自动化。

主题词: 天文观测设备与技术 — 原子钟 — 频率稳定度 — Allan 方差 — 测试系统
— Visual C++ — OMT 方法

分类号: TP311.5, TH714.1⁺4

1 引 言

原子钟具有优秀的稳定度和准确度性能,是时间计量领域重要的计量工具^[1]。作为高精度的时间频率标准,原子钟被广泛应用于基础天文学、通信系统、空间任务、导航与制导、雷达系统等重要领域。

频率稳定度即频率源信号的平均频率随机变化量,是决定原子钟性能的主要指标。它反映了原子跃迁频率对外界各种干扰因素的敏感程度和这些因素本身的稳定性。目前,国际通行的频率稳定度特性表征方法有两种:时域法和频域法^[3,4]。时域法以阿伦方差表征;频域法以射频谱、相位噪声谱密度、频率噪声谱密度以及相对频率起伏谱密度表征。这些特性指标主要用于不同实验室同类设备之间的比对或同一应用中不同设备之间的比对,用于衡量实际应用中频率源的稳定度。

由于单一的测试方法和通用的测试系统无法满足原子钟对频率稳定度指标高精度的测试要求,因此必须采用完备的测试方法和性能优越的测试系统以实现对原子钟频率稳定度各类指标参数的测量。通过模拟电路技术、数字集成电路技术和微控制器技术的运用,能提高测试系统的精度和稳定性;计算机技术和算法设计的引入,使测试系统有可能实现系统化、智能化、自动化。

2 频率稳定度测试系统原理

高精度频率稳定度测试系统是用于测量原子钟输出信号频率稳定度的专用测试系统,它

通过采集原子钟输出信号在一定时间范围内的平均频率来计算并显示其时域频率稳定度参数。测试系统由两部分组成:硬件部分和软件部分。硬件部分实时采集原子钟输出信号在取样时间内的周期计数值,并将其通过串口传递至 PC 机。软件部分即应用程序,运行于 PC 机,接收来自硬件部分的数据值,经一系列算法处理后以多种方式显示输出;同时系统能判断测试流程并发出相应控制信息,控制系统协调工作,从而实现测量的自动化。

2.1 测试系统硬件结构

测试系统硬件部分有两路输入信号:待测频标和参考频标。倍频后待测频标和经倍频并调偏的参考频标混频,得到频率为取样带宽的差拍信号。差拍信号经整形、检波电路转换为 TTL 标准的脉冲信号,作为机内计数器的开门和关门信号。机内计数器输出的周期计数值与其基准信号周期乘积的倒数即为在取样时间范围内倍频后的待测频标的平均频率。硬件系统由单片机协调系统时序,向 DDS 写入控制字以设置调偏频率;同时响应来自串口的控制命令,控制机内计数器的开门、关门时间,并将周期计数值通过串口按位传递至软件部分。测试系统硬件结构如图 1 所示。

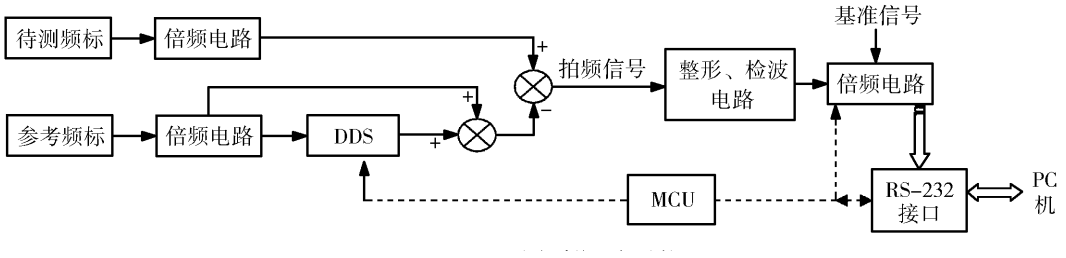


图 1 测试系统硬件结构

Fig. 1 Hardware structure of the testing system

2.2 系统自校原理

系统自校是一种常用的鉴定测试系统上限灵敏度的方法。其原理如图 2 所示。它利用自相关原理,将输入频标既作为待测频标信号,又作为参考频标信号,同时加到测试系统硬件部分的待测和参考输入端。由于输入的两路频标信号是自相关的,因此信号自身的影响未扩大,在输入频标信号满足一定条件时^[2],系统输出的结果与输入频标的性能无关,即反映了系统自身的上限灵敏度。

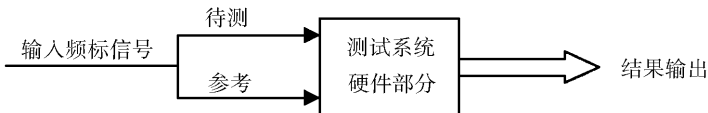


图 2 系统自校原理图

Fig. 2 Schematic of system self - calibration

2.3 调偏频率产生原理

应用于系统中直接数字合成技术(DDS, Direct Digital Synthesis)可将参考频标调偏,以产生拍频信号。通过向 DDS 芯片写入控制字,获得调偏频率。调偏频率产生公式为:

$$f_{\text{OUT}} = \frac{\Delta\text{phase} \times \text{CLKIN}}{2^{32}}$$

上式中, $\Delta phase$ 为 32 位控制字; $CLKIN$ 为 DDS 输入信号频率, 单位为 MHz; f_{OUT} 为 DDS 输出信号频率, 单位为 MHz。

2.4 Allan 方差原理

原子钟输出信号频率的不稳定是由随机噪声引起的。在实际测量中, Allan 方差的标准方差是描述随机噪声时域统计特性的通用标准^[3,4]。

Allan 方差的定义以相对频率起伏的采样方差为基础, 以取样时间内信号的平均频率代替瞬时频率, 记为:

$$\sigma_2^2(\tau) = \frac{1}{2(m-1)} \sum_{i=1}^{m-1} \frac{[\overline{f_{i+1}} - \overline{f_i}]^2}{f_0^2} \quad (1)$$

Allan 方差的标准方差为:

$$\sigma_2(\tau) = [\sigma_2^2(\tau)]^{1/2} \quad (2)$$

其中: $\overline{f_i}$ 为取样时间 τ 内拍频信号的平均频率, m 为取样次数 ($m > 1$), f_0 为被测信号的中心频率。由图 1 可见, 在测试系统的硬件中, 拍频信号的平均频率由机内计数器获得, 被测信号由输入信号倍频后获得, 因此有:

$$\begin{aligned} \overline{f_i} &= \frac{C_i \cdot T_{base}}{T_b \cdot \tau} = \frac{C_i \cdot f_b}{f_{base} \cdot \tau} \quad (T_{base} = \frac{1}{f_{base}}, T_b = \frac{1}{f_b}) \\ \tau &= N \cdot T_b \\ f_0 &= M \cdot f_{in} \end{aligned} \quad (3)$$

其中: C_i 为取样时间 τ 内机内计数器周期计数值的和, T_{base} 为基准信号的周期, T_b 为拍频信号周期, N 为取样时间与拍频信号周期的倍乘系数, M 为输入信号经倍增单元的倍频数, f_{in} 为输入信号频率。

将(3)式代入(1)、(2)式, 有:

$$\sigma_2(\tau) = \frac{N}{M \cdot f_{in} \cdot f_{base} \cdot \tau^2} \sqrt{\frac{\sum_{i=1}^{m-1} (C_{i+1} - C_i)^2}{2(m-1)}} \quad (4)$$

(4)式即为测试系统的 Allan 方差的计算公式。

3 软件需求分析

3.1 需实现的功能

应用程序既是测试系统硬件部分与用户的接口, 又承担数据分析、显示及测试系统流程控制的任务, 是测试系统实现智能化和自动化的主要方式。应用程序实现的主要功能有: 系统自校、DDS 频偏调节、自动化测量、数据处理、存储与显示输出。

3.2 应满足的设计要求

根据测试系统的工作特点, 系统软件部分的设计应满足以下六个设计要求: 强大的数据分析处理能力; 高度的自动化; 高可靠性; 高效率; 良好的健壮性; 界面的友好。除此之外, 代码的可移植性、可扩展性、可读性等也是需要充分考虑的内容。

3.3 对开发工具与开发方法的要求

测试系统软件部分的实现功能和设计要求决定了开发工具应具有对计算机硬件资源强大的管理能力、高速可靠的数据处理能力和可视化图形界面。综合以上考虑, Visual C++ 是适用

于开发测试系统的一个有效的软件开发工具。

由于 Visual C++ 是面向对象的高级程序语言,因此开发方法选择软件工程中基于面向对象的对象模型技术(OMT)方法,以提高程序的可重用性、可维护性和稳定性。对于高精度频率稳定度测试系统这样一个大的复杂系统的应用程序开发而言,运用 OMT 方法,可使用对象模型、动态模型和功能模型从不同方面描述系统,以实现对该系统的全面分析。

4 软件架构设计

4.1 软件框架

Visual C++ 程序的开发是建立在程序框架的基础之上的,必须根据程序的特点和要求建立适合的程序框架。测试系统应用程序需要分别以不同形式同时显示多个视(View),因此更适合采用基于 MFC(Microsoft Foundation Class)的多文档界面(MDI)结构;各个视类型不同,但视的数据来源于同一个文档(Document),建立“文档-多视”结构,可以为一个文档提供多个不同形式的视,使数据的显示和处理分开,保证了代码的封装性。对应地,需建立多个文档模板(Document Template),管理各自的视类型和相同的文档和外框窗口(Frame)。

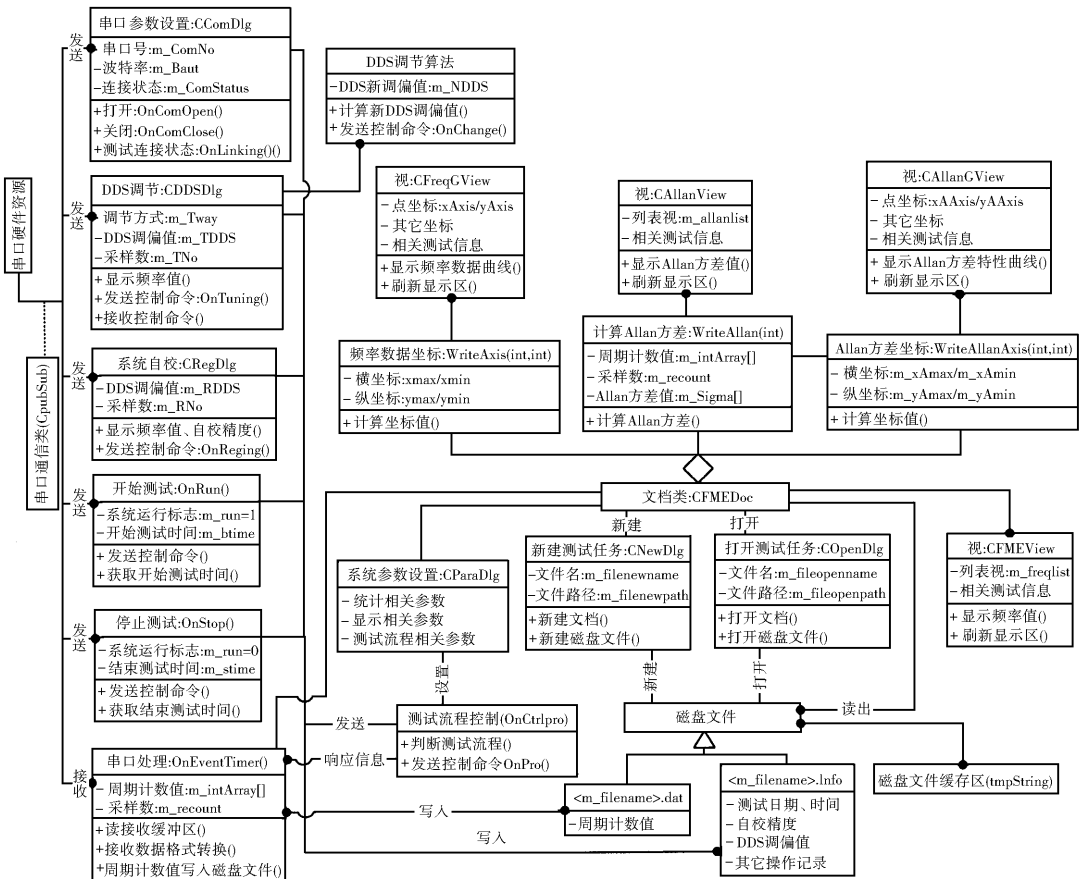


图 3 应用程序对象图

Fig. 3 Object-diagram of application system

基于 MFC 创建的 MDI 工程 FME. dsw 用于管理应用程序系统资源。其中,应用程序类 (CFMEApp) 管理控制流程,发送测试开始、停止命令并根据测试流程调用相应模块;主框架类 (CMainFrame) 装载各消息响应函数,响应系统定时器和用户发出的消息;文档类 (CFMEDoc) 进行数据的处理;各视类实现数据的多种形式显示。

4.2 应用程序对象模型

应用程序的对象模型用于描述系统中的对象和它们之间联系的静态结构。应用程序各模块按功能划分,以类和函数的形式封装。各模块之间通过 MFC 构建的消息映射、命令传递和部分全局变量通信。通常对象模型用对象图表示,如前文中图 3 所示。

5 系统自动测试流程分析

测试系统软件部分对自动测试流程的控制是实现测试过程自动化的主要途径。自动测试流程的控制关键在于对当前运行状态的判断及根据预置的测试流程适时发送正确的控制指令,调用相应的模块。在预置测试任务的前提下,不需要用户的参与,系统自动实现“系统自校→DDS 调节→10ms 稳定度测量→DDS 调节→100ms 稳定度测量→DDS 调节→1s 及以上稳定度测量”的测试流程。测试系统控制流程如图 4 所示。

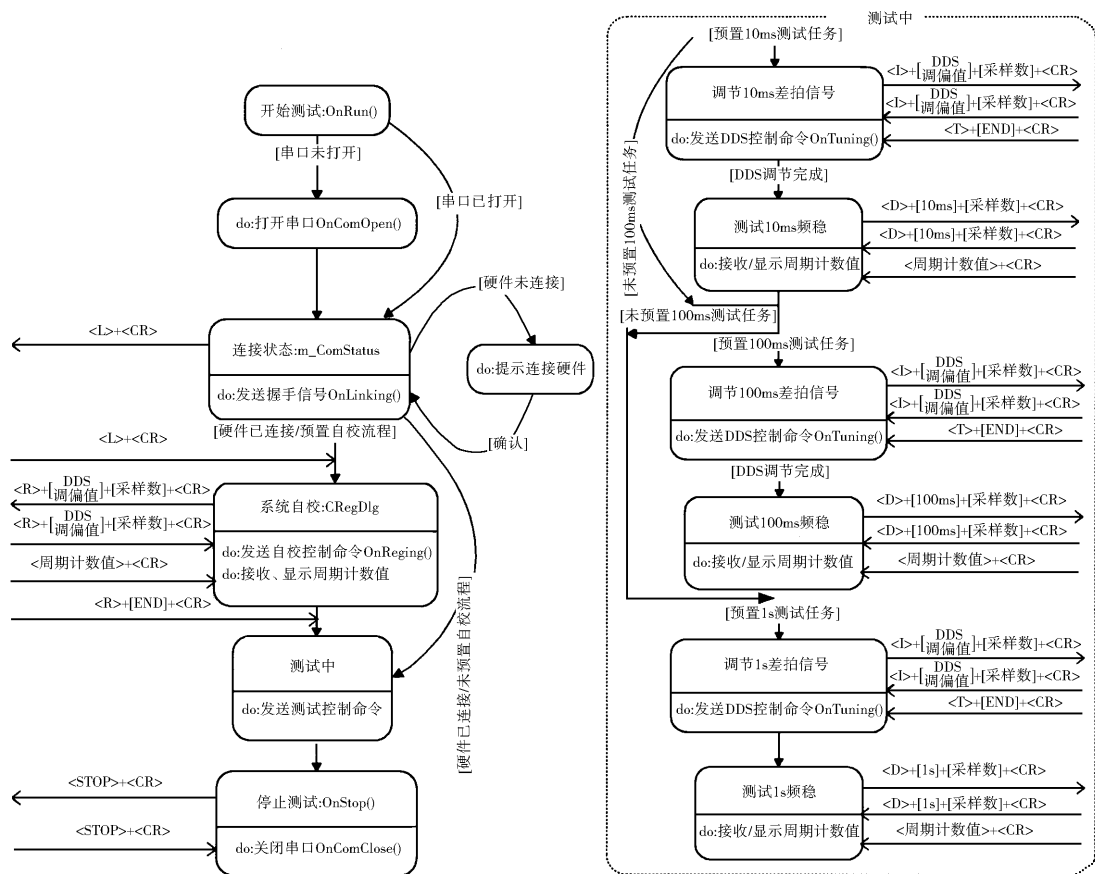


图 4 系统控制流程状态图

Fig. 4 State-chart of system control flow

为保证流程无误,在运行状态改变的時刻需增加握手信号,使测试系统硬件部分与软件部分交互联结,确认当前运行状态和新的控制命令。握手信号格式为: <标识位> + [操作数 1] + [操作数 2] + <CR>; 周期计数值发送格式为: <周期计数值> + <CR>。

6 数据结构与数据处理

6.1 数据结构

程序的长时间运行需考虑内存的占用是否一直递增。若随着程序的运行,数据变量不断增加,必然增大系统的负担,占用更多的内存最终导致程序瘫痪。因此,应严格控制程序中变量和函数占用的资源,及时释放不必要的变量空间。

以周期计数值存储数组 $m_intArray$ 为例:采用类型安全数组类 $CArray$ 类存储;数组长度增加到最大时,将 $m_intArray$ 看作是一个循环结构的数组,新的数据覆盖循环结构中最先进入的数据,使数组长度固定。即:设数组最大长度为 len ,则数组元素为 $m_intArray[0], m_intArray[1], \dots, m_intArray[len - 1]$,串口接收的第 n 个数据 ($n = 1, 2, 3, \dots$) 对应地存放于 $m_intArray[(n - 1) \% len]$ 。循环数组存储原理如图 5 所示。

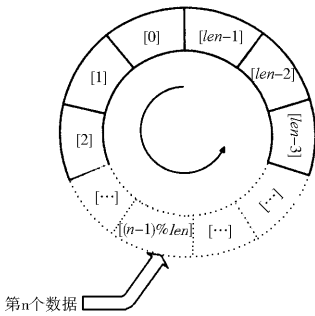
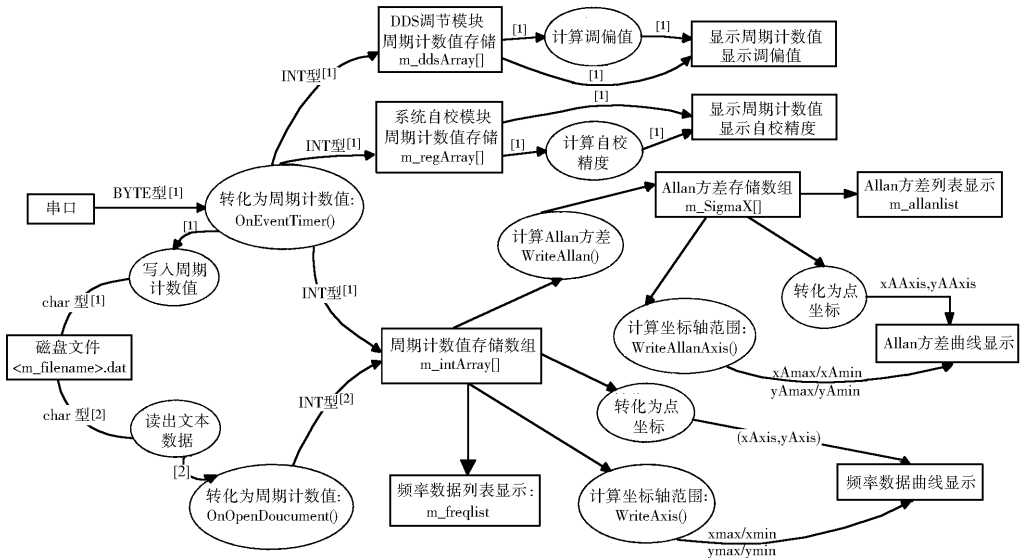


图 5 循环数组 $m_intArray$ 存储原理示意图
Fig.5 Drawing of $m_intArray$ storage principle

采用这种循环结构的存储更新方式,不仅解决了应用程序长时间运行带来的内存占用问题,而且也有利于简化算法代码的结构和数据的显示输出处理。但是由于数据值和数组中的元素是多对一的关系,因此数组的最大长度不应小于各视类同时显示的数据值范围,否则会造成显示错误。

6.2 数据处理



注: [1]为“新建测试任务”数据路径; [2]为“打开测试任务”数据路径

图 6 周期计数值数据流图

Fig.6 Dataflow graph of periodic count value

应用程序内数据值的转移和计算是数据处理的主要任务。应用程序的数据流图(图 6)描述周期计数值数据流从应用程序输入,经过计算和内部数据存储,至输出的全过程。

6.3 Allan 方差的计算

设当前拍频信号周期为 T_b , 机内计数器在第 n 个周期的周期计数值为 D_n 。取样次数为 m , 取样时间 τ 为拍频信号周期的倍数, 如式(3)的第 2 式所示。 $\sigma_2(\tau)$ 的计算从当前拍频信号的第 1 个周期计数值开始, 则式(4)中:

$$C_i = \sum_{n=(i-1)\cdot\tau+1}^{i\cdot\tau} D_n \quad (5)$$

将式(5)代入式(4)中, 有:

$$\sigma_2(\tau) = K \cdot \sqrt{\sum_{i=1}^{m-1} \left(\sum_{n=i\cdot\tau+1}^{(i+1)\cdot\tau} D_n - \sum_{n=(i-1)\cdot\tau+1}^{i\cdot\tau} D_n \right)^2} \quad (6)$$

$$K = \frac{N}{M \cdot f_{in} \cdot f_{base} \cdot \tau^2 \sqrt{2(m-1)}}$$

应用程序要求对 Allan 方差进行实时的计算, 以便实时显示当前的统计结果。同时, 高速的数据采集也要求算法有更高的效率。因此, 在 WriteAllan 函数的实现算法中, 避免了重复运算, 且将运算分散到每一次的新数据采集中, 从而减少了一次性计算的运算时间, 保证了程序运行的连贯性。具体实现算法如下:

函数外变量声明:

```
int m, len; //m:取样次数; len:m_intArray 数组最大长度
double a, b, K, tmpSigma, m_Sigma[ ]; //a, b:tau 内周期计数值总和; K:如式(6)所示;
void WriteAllan( int n, int tau, const CArray< double, double& > &m_Sigma)
{ //n:第 n 个周期计数值 n=1,2,3,……; tau:取样时间; m_Sigma:Allan 方差存储数组
  if((n-1)%(2*tau) == 0) a = m_intArray[(n-1)%len]; // (1)
  if((n-1)%(2*tau) >= 1 && (n-1)%(2*tau) <= (tau-1)) a + = m_intArray[(n-1)%len]; // (2)
  if((n-1)%(2*tau) == tau) b = m_intArray[(n-1)%len]; // (3)
  if((n-1)%(2*tau) >= (tau+1) && (n-1)%(2*tau) <= (2*tau-1)) b + = m_intArray[(n-1)%len]; // (4)
  if(n%tau == 0 && n%(tau*m) > tau) tmpSigma + = pow((a-b), 2) * K; // (5)
  if(n%(tau*m) == 0) { // (6)
    if((n-1) < len) { m_Sigma. Add(sqrt(tmpSigma)); tmpSigma = 0; }
    else { m_Sigma[n/(tau*m) - 1] = sqrt(tmpSigma); tmpSigma = 0; }
  }
}
```

如图 6 所示, 在当前拍频信号下, 从第 1 个周期计数值开始, 每增加一个新的值便调用一次 WriteAllan 函数, 计算对应的 $\sigma_2(\tau)$ 。代码(1)、(2)和代码(3)、(4)交替计算 τ 时间内机内计数器计数值的总和, 如式(5)所示, 从而获得相邻取样周期的 C_{i+1} 和 C_i 。代码(5)计算相邻取样周期计数值差的平方和, 即式(4)中 $\sum_{i=1}^{m-1} (C_{i+1} - C_i)^2$ 部分。代码(6)计算当前组的 Allan 方差值。该算法的时间复杂度为:

$$T(n) = n + \text{Int}\left(\frac{n}{\tau}\right) + \text{Int}\left(\frac{n}{m \cdot \tau}\right) - 1$$

7 实现界面

应用程序系统自校模块 CRegDlg 及自校流程实现界面如图 7 所示;DDS 设置模块 CDDS-Dlg 及 DDS 调节流程和算法实现界面如图 8 所示。

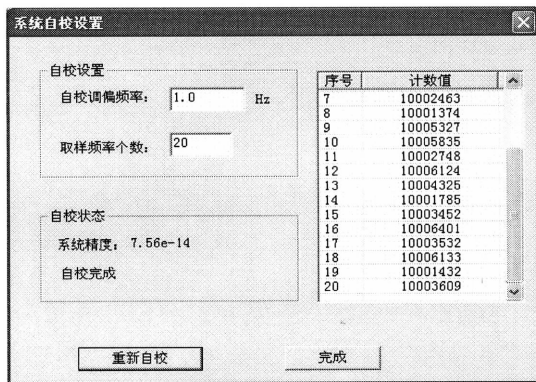


图 7 系统自校对话框

Fig. 7 Dialog box of system self-calibration

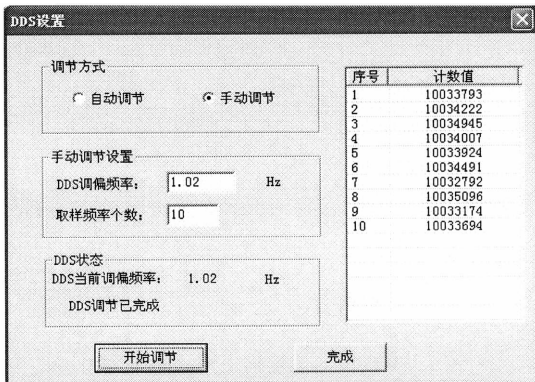


图 8 DDS 设置对话框

Fig. 8 Dialog box of DDS settings

应用程序主界面如图 9 所示。在主框架下,“Frequency Data-List”子窗口实现 CFMEView 类;“Frequency Data-Graph”子窗口实现 CFreqGView 类;“Frequency Stability-List”子窗口实现 CAllanView 类;“Frequency Stability-Graph”子窗口实现 CAllanGView 类。

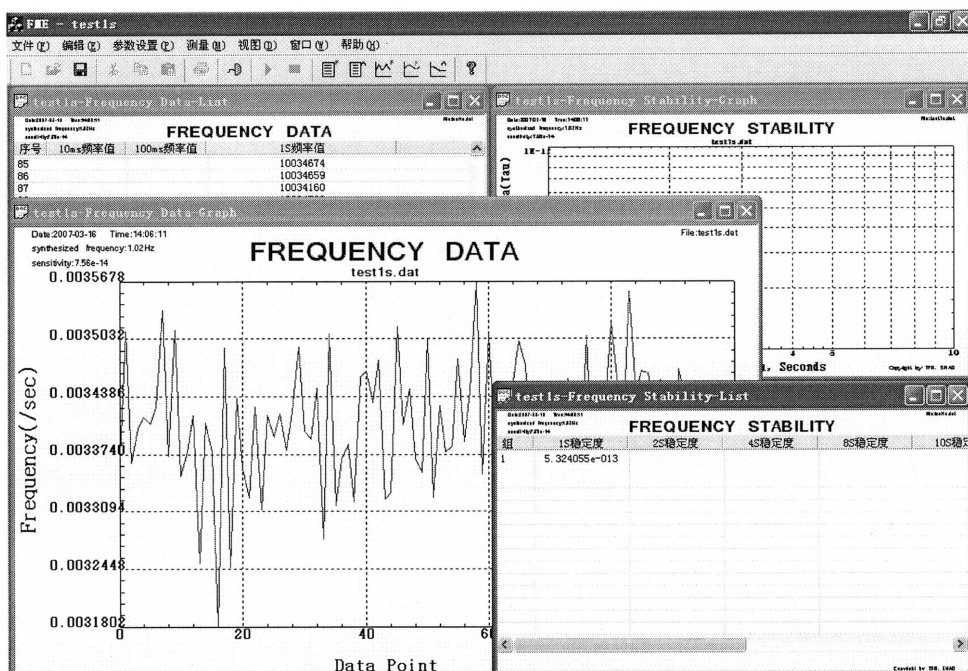


图 9 应用程序主界面

Fig. 9 Main interface of application system

8 结 束 语

本文采用 OMT 的开发方法,首先分析高精度频率稳定度测试系统软件部分的设计需求。在此基础上分别利用对象模型、动态模型和功能模型详细论述了测试系统软件的框架结构、自动控制流程的实现和数据采集、存储、分析、转移,并提出具体的实现算法,以此从不同角度全面描述这一复杂系统。实验结果证明,测试系统软件部分可以实现测试流程智能化、自动化的设计要求,同时有助于提高系统的稳定性、可靠性和可扩展性。

参 考 文 献

- [1] 王庆华,翟造成. 空间电子技术,1999, (4): 1~9
- [2] 杨大豪. 频率稳定度特性和测量技术. 上海: 微型电脑编辑部, 1982
- [3] 李金海. 误差理论与测量不确定度评定, 第 1 版. 北京: 中国计量出版社, 2003
- [4] Barnes J A, Chi A R, Cutler L S. *et al.* IEEE Trans. Instrum. Meas. , May 1971
- [5] 蔡勇, 林传富, 沈季良. 中国科学院上海天文台年刊, 2000, (21): 148
- [6] 侯俊杰. 深入浅出 MFC, 第 2 版, 武汉: 华中科技大学出版社, 2001
- [7] 宛延闯, 定海. 面向对象分析与设计, 北京: 清华大学出版社, 2001

SOFTWARE DESIGN OF HIGH PRECISION FREQUENCY STABILITY TESTING SYSTEM

XIE Yan^{1,2} CAI Yong¹

(1. Shanghai Astronomical Observatory, Chinese Academy of Sciences, Shanghai 200030;

2. Graduate School of the Chinese Academy of Sciences, Beijing 100049)

Abstract

Combined with the principle of testing system with high precision for frequency stability of atomic clocks, the design of software is introduced. Based on the demands of the software, the frame and the flow of automatic test and data processing including its algorithm are analyzed in detail with OMT method. The software is proved by experimental results to be stable and automatic in testing process.

Key words astronomical facilities and technique — atomic clock — frequency stability — Allan variance — testing system — Visual C++ — OMT method