

采用MATLAB与C++混合编程生成扬声器功率实验信号^①

周静雷, 董 姣, 常 诚, 李 倩

(西安工程大学 电子信息学院, 西安 710048)

摘 要: 针对在 C++ Builder 中生成白噪等扬声器功率实验所用测试信号的难度较大, 以及 MATLAB 与 C++ Builder 混编过程中的兼容性问题, 文章提出了一种 MATLAB 与 C++混合编程的方法. 首先, 运用 Matlab 生成动态链接库 DLL; 其次, VS 调用该 DLL 文件并编写接口函数以生成可供 C++ Builder 调用的 DLL 文件; 最后, 在 C++Builder 中设计应用程序以生成扬声器功率实验测试信号. 为验证所生成信号的准确性, 文章对信号的均值、方差、自相关性、频谱、功率谱密度、概率密度、三分之一倍频程的 RTA 等参量进行计算, 通过将计算结果与标准信号进行对比分析, 发现混合编程生成的信号可作为扬声器功率实验信号使用.

关键词: Matlab; VS2010; C++Builder; 混合编程; 功率实验信号

Mixed Programming Between Matlab and C++ to Generate Power Test Signal

ZHOU Jing-Lei, DONG Jiao, CHANG Cheng, LI Qian

(School of Electronics and Information, Xi'an Polytechnic University, Xi'an 710048, China)

Abstract: According to the problem that it is difficult to generate white noise and some other speaker power test signals in C++ Builder and the problems of compatibility in mixed programming between Matlab and C++ Builder, a mixed method combined Matlab with C++ is put forward. Firstly, a dynamic link library DLL is generated in Matlab. Then, VS calls the DLL file and writes interface function to create another DLL file which can be used in C++ Builder. Finally, design the application program to generate the speaker power test signal in C++ Builder. To verify the accuracy of the generated signal, some parameters of signal are calculated, such as the mean, variance, self-correlation, spectrum and power spectral density, the probability density and RTA of one third octave. Comparing the calculated results with the standard signal's, it shows that the signal generated by mixing programming can be used for speaker power test.

Key words: Matlab; VS2010; C++ Builder; mixed programming; power test signal

随着电声行业不断发展, 扬声器已成为人们生活娱乐中必不可少的一部分, 与此同时消费者对扬声器音质的要求也不断提升, 人们开始追求立体环绕声效, 甚至是 Hi-Fi. 完美的扬声器音质必须经过无数次的测量修改参数方可得到, 其测量过程首先需要通过上位机对被测扬声器输入测试信号, 而信号的准确性直接影响了测量结果, 因此对该测试信号的研究十分必要. 扬声器功率试验所用测试信号一般包括白噪声、粉红噪声、模拟节目信号等. 目前对生成测试信号方法研究主要以模拟方法为主, 数字方法的研究成果较少. 文献[1]使用模拟电路产生高斯白噪声, 过程复杂且生

成信号的性能不够完善; 文献[2]使用信号发生器生成各参数已经设定好的高斯白噪声, 其灵活性不高且只能生成白噪声, 其信号种类受限. 论文中生成实验信号的算法程序最终将用于基于 C++语言研发的虚拟仪器—扬声器功率试验测试系统的上位机软件中, 该测试信号算法程序要求被其上位机调用, 并且具有与开发语言 C++ Builder 兼容的接口. 直接在 C++ Builder 中编写复杂度较高的信号处理算法程序, 完成的难度较大, 而且精度和正确性都难以保证. 而这些算法在 Matlab 中开发, 不仅难度较低且具有高的精度, 因此只需将 Matlab 开发的该算法程序移植到 C++Builder

^① 收稿时间:2015-08-13;收到修改稿时间:2015-10-08

中,就可满足复杂多样的实际需求.

针对以上存在的问题及对信号复杂多样的需求,文章提出用软件之间混编^[3]的方法生成信号,并且对生成信号各参数进行具体分析以验证信号准确性.鉴于 Matlab 中生成的 DLL 在 C++ Builder 中被调用时存在兼容问题,因此提出了利用 Matlab、Microsoft Visual Studio 2010 的 C++环境(下文简称为 VS)、C++Builder 混编的方法生成功率实验测试信号,相比模拟方法可灵活控制信号,相比之前的数字方法研究过程更简单准确且信号种类更多样,同时解决了 Matlab 与 C++ Builder 混编过程中的兼容性,为功率实验信号生成提供了一种设计方法.

1 信号生成

扬声器功率实验信号生成流程如图 1 所示,主要包括各软件内部程序编写,及软件间通过.dll^[4]文件的连接.

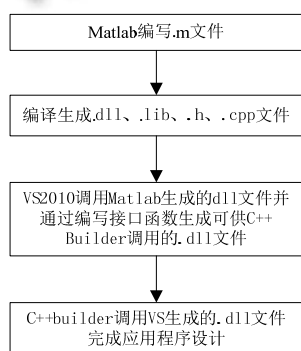


图 1 系统总体框图

1.1 Matlab 生成 DLL 文件

Matlab 主要任务是建立与 VS 的连接,并生成.dll 和其它文件,其流程如图 2 所示.

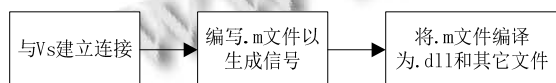


图 2 Matlab 生成 DLL 文件流程

首先,在 Matlab 的 Command Window 窗口中运行 mex -setup、mbuild^[5]-setup 命令,在出现的可匹配编译器中选择 VS2010,以建立 Matlab 与 VS 的连接.然后,编写.m 文件,下面以白噪声的生成为例(下文都以白噪声为例)说明程序编码.白噪声是功率实验所用测试信号中最基本的信号,其生成原理依据 Box-muller 算

法,该算法具体为:

1) 生成服从均匀分布的随机数序列 U1、U2,可通过 randn 函数生成;

2) 对随机序列进行正态化处理,其公式如下,

$$Z = \sqrt{-2 * 1n\pi(U1)} * \cos(2 * \pi * U2) \quad (1)$$

如在 C++ Builder 中直接生成白噪,则需要对上述算法进行详细编写得到最终白噪序列 Z,其过程较复杂.而利用混合编程的方法则更简单、有效,只需在 matlab 的 m 文件中编写 function 函数去调用 Matlab 自带的白噪生成函数,功能函数如下所示.

```
function [whitenoise]=leinwgn(m,n,p)
```

```
whitenoise=wgn(m,n,p);
```

wgn 是 Matlab 中自带生成高斯白噪声的函数,其代表意思为 White Gaussian Noise.其中 m 和 n 分别代表生成白噪声的行数及列数, p 为白噪声强度.

最后,通过编译命令对生成的.m 文件进行编译,将得到.dll 文件.在 Matlab 窗口中运行 mcc-W cpplib:WgnDll -T link:lib leinwgn.m 命令,其语法功能是将 m 文件转换为对应的 C/C++文件,运行完成后将在 m 文件所在目录下生成.dll, .cpp, .lib, .h 等文件.

1.2 VS 编写接口函数

VS^[6]主要完成接口程序的编写并生成可供 C++ builder 调用的 DLL 文件,其基本流程如图 3 所示.

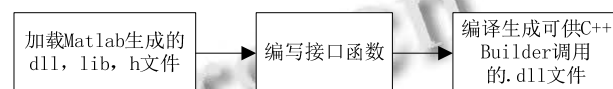


图 3 VS 生成 DLL 文件流程

首先,在 VS 软件中新建 MFC DLL 项目并对其进行设置,并将在 Matlab 中生成的.dll, .lib, .h 文件加载到 VS 中.下面具体说明加载步骤和方法:

1) 在 VS 中,新建项目->MFC DLL 工程->项目名称->WhiteNoiseDLL->完成;

2) 复制 Matlab 生成的.dll, .lib, .h 文件到所建项目路径下;

3) 对项目设置头文件及库文件路径,其方法为,项目->WhiteNoiseDLL 属性->配置属性->VC++ 目录->在包含目录及库目录中分别添加头文件.h 及库文件.lib 的路径;

4) 添加具体外部依赖项,方法为,WhiteNoiseDLL 属性->链接器->输入->附加依赖项,

此处添加由 Matlab 生成的.lib 文件名。

5) 添加具体头文件, 方法为, 解决方案 WhiteNoiseDLL->头文件->添加->现有项->选择由 Matlab 生成的.h 文件。

然后, 编写项目的.cpp 文件, 以完成接口函数的编写。具体编写以下 4 部分:

- 1) 在头文件部分加入 Matlab 生成的.h 文件名;
- 2) 打开并初始化项目及动态链接库文件;
- 3) 编写接口函数, 其函数名为 extern "C" double __declspec(dllexport) _cdecl WNS(double *wdArray, int m, int nLength, double p)。其中 m, nLength, p 即 Matlab 中 m 文件的输入参数 m, n 及 p。指针 wdArray 代表输出的 whitenoise 数组;
- 4) 终止化动态链接库文件。

最后, 对编写的.cpp 文件进行编译, 将在“Debug”文件夹下生成所建项目的.dll、.lib 等文件, 该文件可供 C++ Builder 调用。

1.3 C++ Builder 完成应用程序开发

C++ Builder 主要完成上位机的编写。通过调用 VS 生成的.dll 文件, 完成应用程序的开发最终生成扬声器功率实验测试信号, 其基本流程如图 4 所示。

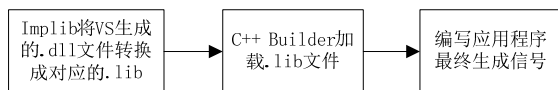


图 4 C++ Builder 生成测试信号流程

首先, 转换由 VS 生成的.dll 文件, 由于其生成的.dll 不能被 C++ Builder 识别, 需要通过 C++ Builder 自带的 implib 工具将.dll 文件转换为对应的可被软件识别的.lib 文件; 然后, 新建项目并加载转换后的.lib 文件; 最后, 编写.cpp 程序, 通过应用程序给各参数进行赋值, 并调用 VS 中编写的接口函数 WNS(white, int m, int nLength, double p), 输出的 white 数组即为最终的白噪声信号。

经过混合编程, 生成白噪声信号如图 5 中第一幅“白噪声”所示。

通过对混合编程方法生成的白噪声进行进一步分析, 发现生成的白噪声信号通过每倍频程增加 3dB 的滤波器后将得到粉噪声信号; 白噪声信号通过模拟节目信号电路后将得到模拟节目信号, 而粉噪声及模拟节目信号都是扬声器功率实验中比不可少的测试信号。在应用中,

由混合编程生成测试信号的应用范围也将会很广, 不仅可以应用于扬声器功率实验, 也可以在其它测试仪器中作为测试信号使用。

2 信号参量分析

针对生成的信号, 文章计算分析了均值, 方差, 自相关性, 概率密度, 频谱, 功率谱密度, 三分之一倍频程 RTA 等参量。对于上述参量, 均值和方差的算法较简单, 可以直接在 C++ Builder 中编写完成, 无需进行混编; 计算其它参量则需要利用与上文相同的混合编程方法进行分析。以下具体说明对参量的分析。

1) 自相关函数^[7], 代表信号与其延迟信号间的相似程度。自相关函数公式如下式:

$$R_x(\tau) = \int_{-\infty}^{\infty} x(t)x(t-\tau)dt \quad (2)$$

当延迟时间 τ 为 0 时, 函数值最大, 其值等于信号的能量。对于白噪声, τ 为 0 时函数值等于强度 P。Matlab 中对应的函数为 [y, lag]=xcorr(whitenoise, 'unbiased'), lags 为信号的延时或超前, 信号的自相关性是相对于自身存在一定超前和滞后而言的。

2) 概率密度函数^[8], 代表白噪声在某一确定点出现的可能性。其公式如下:

$$P(a < x \leq b) = \int_a^b f(t)dx \quad (3)$$

高斯白噪声的概率密度原理上服从正态分布, Matlab 中对应的函数为 [f,y]=ksdensity(whitenoise)。

3) 频谱^[9], 频谱分析又称为 FFT(傅里叶)变换, 代表信号在不同频率的能量分布。其公式如下:

$$X(k) = \sum_{n=0}^{N-1} x(n)W_N^{nk}, W_N = e^{-j\frac{2\pi}{N}} \quad (4)$$

在 Matlab 只需调用 Y=abs(fft(whitenoise, nfft))函数即可实现复杂的 fft 算法, 其中 nfft 是信号傅里叶变换采样点数, 其值越大结果越精确。

4) 功率谱密度^[10], 代表每单位频率波的功率。其公式如下:

$$X(e^{j\omega}) = \sum_{n=-\infty}^{+\infty} x(n)e^{-j\omega n} = \sum_{n=0}^{N-1} x(n)e^{-j\omega n}, \hat{S}_x(e^{j\omega}) = \frac{1}{N} |X(e^{j\omega})|^2 \quad (5)$$

白噪声的功率谱与信号的自相关函数为傅里叶变换对, Matlab 中对应的函数为 [P, f]=periodogram(whitenoise, window, nfft, Fs), window 为所加窗, Fs 为采样频率。

5) 三分之一倍频程 RTA, RTA 是“Real Time

Analysis”,实时分析。文章中计算信号在三分之一倍频段的能量分布,即对三十一个频率点的有效值进行计算,其中需要用到傅里叶变换。

上述五个参量分析,所用的算法相对生成白噪声的算法较复杂,若直接在 C++ Builder 中完成程序设计,所需的工作量相对混合编程会大很多,且准确度难以保证,进一步证实了混合编程在此次扬声器功率实验信号生成中的重大意义。

将经过混合编程生成信号最终用于实验的测试信号,运行系统测得结果如图 5 所示。

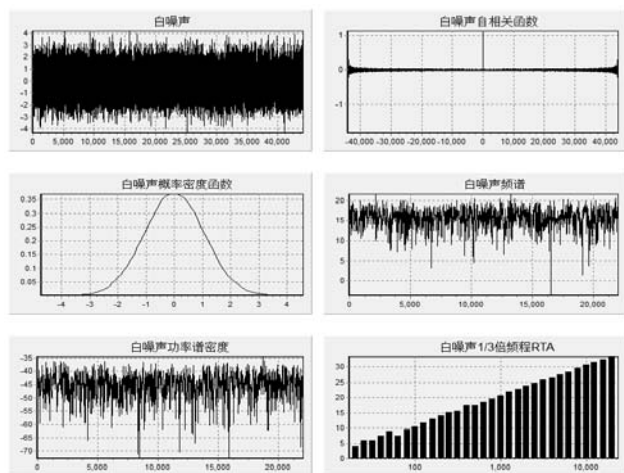


图 5 混合编程生成白噪声的参量分析

3 信号验证

为验证信号的准确性,本文将混编生成的信号与国家标准^[11]规定的噪声信号进行对比并对其参量进行分析对比。首先将国家标准白噪声信号保存为.wav 文件;其次将该 wav 文件通过 wavread 函数读取到 Matlab 中,并编写与上文参量分析对应的 m 文件运行后,分析结果如图 6 所示。

对比图 5 与图 6 参量分析结果,可通过以下几点确定混合编程生成的白噪声与标准信号特性相同,其可以作为扬声器功率实验测试信号。

- 1) 信号的自相关函数在 $x=0$ 处为冲击函数,幅值等于强度 p ;
- 2) 信号的概率密度函数服从正态分布;
- 3) 信号频谱在整个频段内服从等带宽频带所含噪声能量相等;
- 4) 信号的功率谱密度在整个频段内均匀分布;
- 5) 对数坐标中,信号的能量符合每倍频程增加 3dB。

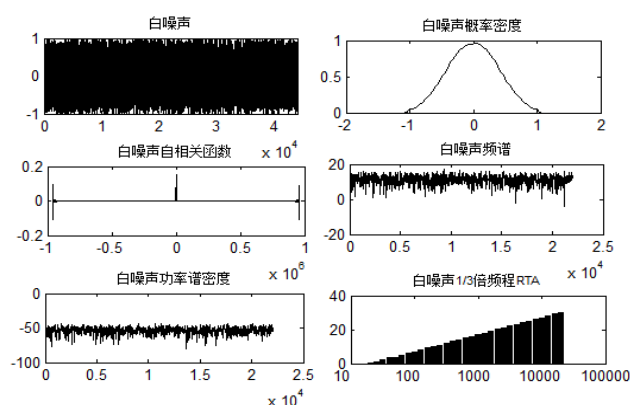


图 6 国家标准中白噪声参量分析

4 结论

通过对 MATLAB 与 C++ 的混合编程方法生成的白噪声信号进行参量分析与验证,可以看到:使用混合编程方法生成扬声器功率实验测试信号过程中,Matlab 可以既准确又容易生成信号,这样弥补了 C++ Builder 在信号开发方面的不足;Matlab 在对白噪声各参量分析时,也充分发挥了其优势;混合编程可以弥补 Matlab 与 C++ Builder 两者的不足,同时也可以将各自优势很好结合。混合编程为扬声器功率实验测试信号的生成提供了一种新生成方法,具有一定的实际意义。

参考文献

- 1 盘书宝.一种基于 FPGA 的高斯噪声源设计.桂林航天工业学院学报,2014,4:330-334.
- 2 付俊,李光灿.一种高斯白噪声信号发生器的设计与实现.计算机测量与控制,2012,5:1436-1438.
- 3 李璞,赵地. Matlab 与 VC++ 混合编程技术在信号采集中的应用.计算机与网络,2013,12:56-58.
- 4 左明成,武云. VC++ 和 MATLAB 混合编程的新应用.计算机技术与发展,2014,12:15-20.
- 5 罗伯特·J·希林,桑德拉·L·哈里斯,殷勤业,王文杰,邓科等译. 数字信号处理导论——MATLAB 实现. 西安:西安交通大学出版社,2014.
- 6 朱磊. MATLAB 与 C 的对比研究. 宜春学院学报, 2012,4:49-50.
- 7 岳磊. 伪随机序列的构造及其自相关函数分析[学位论文]. 西安:西安电子科技大学,2014.
- 8 屈毅,穆丽宁,赖展翅. 非高斯随机分布系统自适应控制算法的研究. 现代电子技术,2014,4:53-55.
- 9 侯盼卫,杨录,王建军. 基于 FFT 系数的正弦信号频率估计算法. 光电技术应用,2013,6:58-62.
- 10 张晓勇,罗来源. 参数未知的舰船辐射噪声检测方法. 声学学报,2015,4:511-518.
- 11 GB/T12060.2-2011. 声系统设备第 2 部分:一般术语解释和计算方法. 北京:中国质检出版社,2011.