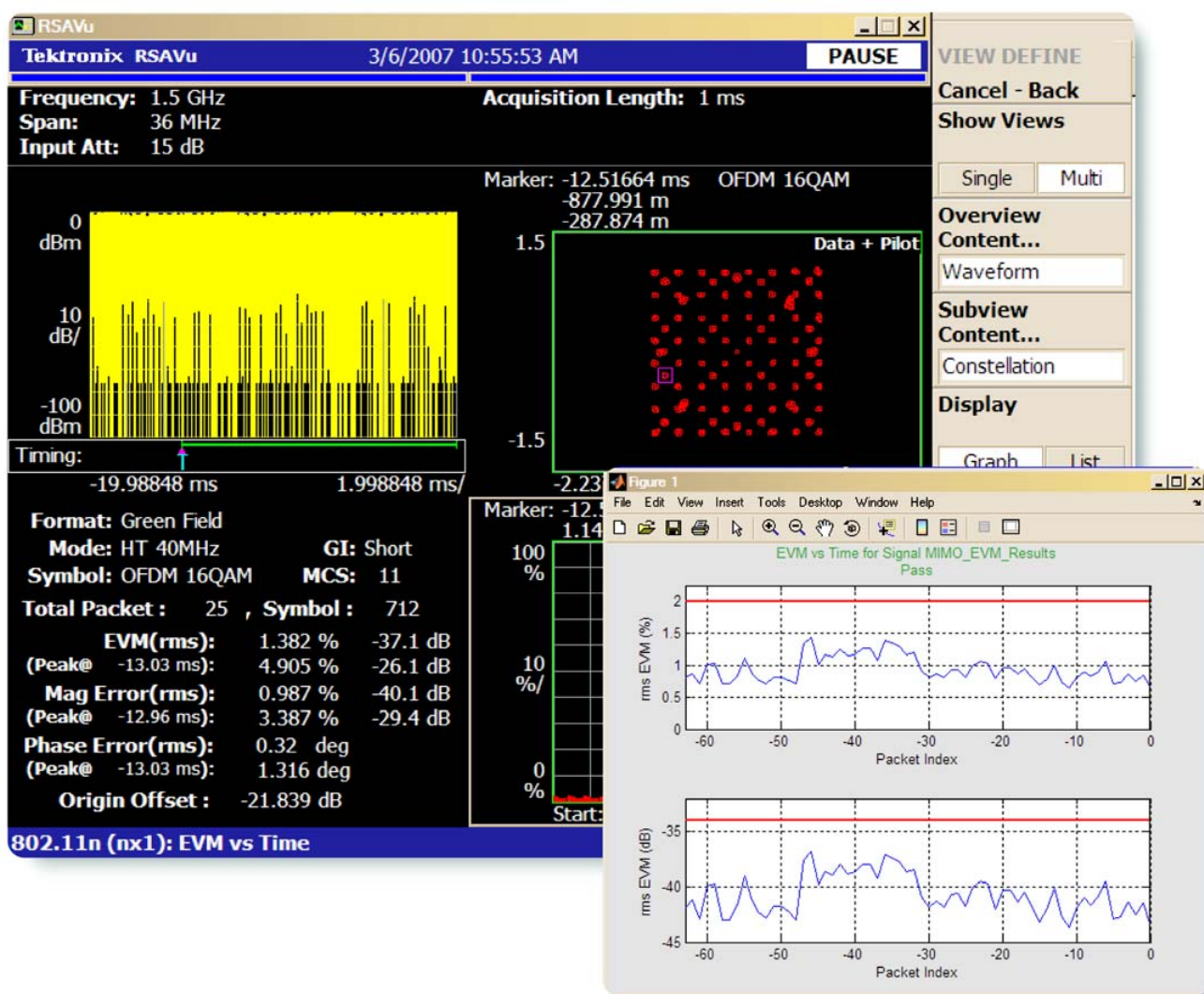


RSAVu 远程控制演示



概述

本文使用了两个实例，一个是 C++ 编程实例，另一个是 MATLAB¹ 编程实例，介绍了怎样通过 SCPI 命令与 RSAVu 远程通信。这些实例分析了 802.11n MIMO 2x2 信号，以编程方式从 RSAVu 得到测量结果并生成了一份

文本报告。一条 2x2 MIMO 通道有两台发射机和两台接收机，因此用户需要两台实时频谱分析仪捕获 MIMO 信号。本演示中的信号是使用两台泰克 RSA6114A 实时频谱分析仪捕获的。

¹ MATLAB 是 The MathWorks 公司注册商标。1994–2007

RSAVu 远程控制演示

► 应用说明

这些编程实例执行下述任务，分析 802.11n MIMO 信号：

1. 使用 TekVISA 连接 RSAVu；
2. 在 RSAVu 中设置 802.11n MIMO 2x2 测量模式；
3. 把存储的波形加载到 RSAVu 中；
4. 在 RSAVu 上运行分析；
5. 使用编程接口，从 64 个分组的子载波星座(SCC)测量中提取测量结果；
6. 关闭到 RSAVu 的连接；
7. 处理结果，生成由 EVM 均方根值组成的简单文本报告。

MATLAB 编程实例还创建了一条曲线，针对限制线比较测得的 EVM 值，从而确定被测设备是否满足 EVM 规范。

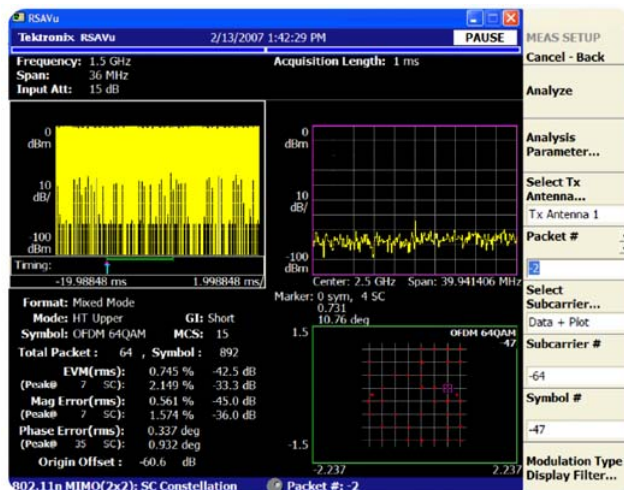
图 1 是在 RSAVu 中进行的 802.11n MIMO 2x2 测量的屏幕截图，右下角是这个分组的子载波星座图，左下角是数值读数。对这个分组，信号有一个 64-QAM 星座。捕获的两个波形没有附加的衰落，因此估算的信道响应是平坦的。在 64 个分组上使用各种信号格式和调制类型，64 个分组依次重复。

演示要求

为进行演示，必须满足下述要求：

1. 在 PC 上安装 RSAVu，打开远程接口；
2. 在 PC 上安装 TekVISA v3.0 (必须在安装 RSAVu 后安装)；
3. 有运行 802.11n MIMO 2x2 选件的 license；
4. 为进行 MATLAB 演示，必须有合法的 MATLAB license。

另外我们还为 C++ 和 MATLAB 实例提供了源代码，这一 MIMO 实例中使用的原则也可以用于 RSAVu 的其它测量模式中。



► 图 1. 802.11n MIMO 2x2 解调实例。

RSAVu 设置

在 RSAVu 上启动编程接口步骤：

1. 在安装 RSAVu 后，安装 TekVISA v3.0。如果已经安装旧版本的 TekVISA，那么在安装 RSAVu 后需要先卸载旧版本 TekVISA，才能启动远程控制功能。
2. 在安装 TekVISA 后，运行 RSAVu，点击 RSAVu 标题栏，出现前面板显示屏，选择“System”，然后在“Soft Key”菜单上选择“Remote Setup ... > Remote Interface > On”，打开远程控制功能。
3. 在 Windows 托盘中黄色 TekVISA 图标上按鼠标右键，选择“Instrument Manager...”。在运行 TekVISA Instrument Manager 时，在可用的仪器列表中应看到下述信息：GPIB8::1::INSTR，如果编写自己的程序，那么将使用这个地址与 PC 上的 RSAVu 通信。这些实例已经编程，默认使用这个地址通信。

一旦在 RSAVu 中打开远程功能，可以准备运行实例。

C++ 编程实例

安装

解压文件 RSAVu_MIMO_Demo.zip，建立一个目录 RSAVu_MIMO_Demo，必须把这个目录复制到 C 盘根目录下，才能运行 C++ 实例 (实例中的某些目录名称采用硬编码方式)。通过双击可执行文件 C:\RSAVu_MIMO_Demo\C\C_RSAVu_Demo.exe，可以启动目录。源代码位于同一目录下的 C_RSAVu_Demo.cpp 中。

运行实例

双击可执行文件后，会出现一个控制台窗口，如图 2 所示。在第一次启动控制台时，会显示信息“Starting the 802.11n MIMO measurements in RSAVu。”实例中执行的基本步骤如下：

1. VISA 设置
2. RSA 设置
3. 从 RSAVu 提取测量数据
4. 关闭会话
5. 处理结果

VISA 设置

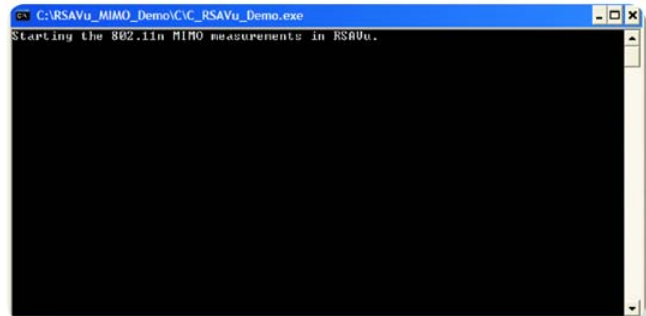
从源代码中可以看出，实例中的第一步是初始化 VISA 系统，打开到 RSAVu 的连接。通过引用头文件 visa.h，这一实例使用了其内置的函数和类型。visa.h 头文件在 TekVISA 安装过程中也随之安装。如需了解 TekVISA 函数有关的更多信息，请参阅与 TekVISA 一起安装的 TekVISA 编程人员手册。

下面显示了连接 RSAVu 的步骤(为清楚起见，这里省略了整个源代码中存在的参数设置和错误校验)。

```
status = viOpenDefaultRM (&rm );
status = viOpen (rm, "GPIB8::1::INSTR", VI_NULL,
VI_NULL, &vi );
```

为存取 RSAVu 上的数据，本实例使用格式化的 I/O 服务 viPrintf 和 viQueryf，因此我们先使用 viSetAttribute 函数，设置连接，启动这些服务。

```
status = viSetAttribute (vi, VI_ATTR_WR_BUF_OPER_
MODE, VI_FLUSH_ON_ACCESS );
status = viSetAttribute (vi, VI_ATTR_RD_BUF_OPER_
MODE, VI_FLUSH_ON_ACCESS );
```



► 图 2. C++ 演示启动屏幕。

然后，把超时周期设置为 10 秒(这一属性的设置单位为毫秒，因此我们设置 10000，表示超时周期为 10 秒)。

```
status = viSetAttribute(vi, VI_ATTR_TMO_VALUE, 10000);
```

现在打开与 RSAVu 的连接，准备开始向 RSAVu 发送 SCPI 命令。

RSA 设置

第一步是把 RSAVu 设置于 802.11n MIMO 2x2 测量模式下，设置默认值，然后选择子载波星座(SCC)测量。SCC 测量中包括我们为演示采集的 EVM 均方根值，以及调制质量有关的其它测量。

```
status = viPrintf( vi, ":INST:SEL \"DEMM2WLAN\" ");
status = viPrintf( vi, ":CONF:M2WLAN\" ");
status = viPrintf( vi, ":SENS:M2WLAN:MEAS SCC\" ");
```

下一步是加载两个 MIMO 波形，即两台 RSA6114A 采集的信号。这些波形文件名为 MIMO_RX1.tiq 和 MIMO_RX2.tiq 存储在 TIQ_Files 目录下。注意：第一个 MIMO 信号必须在第二个信号之前加载。

```
status = viPrintf( vi, ":MMEM:LOAD:RX1
\"C:\\\\RSAVu_MIMO_Demo\\\\TIQ_Files\\\\MIMO_RX1.
TIQ\" ");
status = viPrintf( vi, ":MMEM:LOAD:RX2
\"C:\\\\RSAVu_MIMO_Demo\\\\TIQ_Files\\\\MIMO_RX2.
TIQ\" ");
```

然后我们把分析长度设为 6.5ms，对这个信号而言，6.5ms 代表 64 个分组信号长度。

```
status = viPrintf (vi, ":SENS:M2WLAN:ANAL:
LENG 6.5E-3\" ");
```

RSAVu 远程控制演示

► 应用说明

然后，演示实例控制RSAVu去分析两个MIMO波形，为64个分组生成SCC测量数据。由于这一操作所需时间要比大多数操作长，因此我们把超时周期提高到4分钟，然后在操作完成后把它重设成10秒。然后，我们扫描要完成的操作，等待设成1的OPC。

```
status = viSetAttribute( vi, VI_ATTR_TMO_
VALUE, 240000);
status = viPrintf( vi, ":SENS:M2WLAN:IMMn");
while( intOPC != 1) status = viQueryf( vi, "*OPC?n",
"%d",&intOPC );
status = viSetAttribute(vi, VI_ATTR_TMO_VALUE, 10000);
```

从 RSAVu 提取测量结果

在分析结束后，可以从64个分组中提取测量数据。在通过SCPI命令调用时，SCC测量返回多个值，因此代码把返回的文本字符串解析成各种SCC测量值。本例在生成文本报告时只使用EVM均方根值，但要先从SCC测量返回的字符串中解析所有结果。

在实例运行到这一部分时，控制台窗口将显示“Retrieving 802.11n MIMO results from RSAVu。”。由于RSAVu将从-63到0索引分组，因此我们创建一个称为iPacket的索引，在循环计数器i从0递增到63时，这个索引会从-63递增到0。

```
int numResults = 64;           // Number of packets to
                                read from RSAVu
int iPacket = -numResults;     // Match RSAVu's packet
                                index (negative indexing)
for( i=0; i < numResults; ++i)
{
    // Advance to the next packet in RSAVu
    iPacket++;
    status = viPrintf( vi, ":SENS:M2WLAN:PACK:
NUMB %i\n", iPacket );
    // Retrieve the measurement data
    status = viQueryf( vi, ":FETCH:M2WLAN? SCCn",
"%s", buffer );
```

```
    pch = strtok( buffer, ",");
    peakEVM_per[i] = atof( pch );
    peakEVM_dB[i] = atof( pch );
    rmsEVM_per[i] = atof( pch );
    rmsEVM_dB[i] = atof( pch );
    SC_number_EVM[i] = atoi( pch );
    peak_MError_per[i] = atof( pch );
    peak_MError_dB[i] = atof( pch );
    rms_MError_per[i] = atof( pch );
    rms_MError_dB[i] = atof( pch );
    SC_number_MError[i] = atoi( pch );
    peak_PError_deg[i] = atof( pch );
    rms_PError_deg[i] = atof( pch );
    SC_number_PError[i] = atoi( pch );
}
```

关闭会话

现在我们已经采集了所有测量结果，接下来可以关闭会话，并处理采集的结果。

```
viClose( vi );
viClose( rm );
```

处理结果

本演示其余部分用一个小的实例说明了可以怎样处理从RSAVu中编程采集的数据。在本例中，将生成一个文本文件，汇总部分基本测试信息(RSA的型号和序列号)、用百分比和dB表示的64个分组的EVM均方根值

最大值和平均值、用百分比和 dB 表示的每个分组的 EVM 均方根值。注意，用 dB 表示的平均值是线性 EVM 值的 dB 值，而不是 dB 值的平均值，因为对数不是线性过程，所以应先取平均，然后再取对数。

```
GenerateReport (resultsFilename, serial_RSA1, serial_
RSA2, numResults, rmsEVM_per, rmsEVM_dB );
```

生成的报告名称为 MIMO_EVM_Results.txt，显示如下：

```
Filename: MIMO_EVM_Results.txt
RSA #1: RSA6114A_B000001
RSA #2: RSA6114A_B000002
Max EVM (%): 1.438987
Mean EVM (%): 0.943646
Max EVM (dB): -36.838865
Mean EVM (dB): -40.503814
```

```
=====
| rms | rms
Packet | EVM (%) | EVM (dB)
-----+-----+-----
-63 | 0.8068 | -41.86
-62 | 0.8695 | -41.21
-61 | 0.7174 | -42.88
-60 | 1.0089 | -39.92
(Packets -59 through -6 snipped for brevity)
-5 | 0.7185 | -42.87
-4 | 0.7296 | -42.74
-3 | 0.8570 | -41.34
-2 | 0.7453 | -42.55
-1 | 0.8489 | -41.42
0 | 0.6822 | -43.32
```

MATLAB 编程实例

MATLAB 编程实例与 C++ 编程实例非常类似，但内容略有增加。首先，除保存 EVM 值的平均值和最大值外，

它还在文本报告中保存最小 EVM 值。第二，它把所有 SCC 测量结果存储到一个 MATLAB 文件中。最后，它对照一个限制线建立 EVM 均方根值随时间变化曲线，标出超出限制线的任何点，在曲线上标出 PASS/FAIL 指示符。此外，MATLAB 实例中使用的 SCPI 命令使用完整版本的命令，而 C++ 实例使用的则是缩写版本。如果你读取代码，那么完整版本阅读和理解起来要容易得多，而缩写版本要求发送的字节数要少得多。用户可以选择自己偏爱的格式，这两种版本实现的功能相同。

安装

MATLAB 实例对其在硬盘上的安装位置没有限制。为运行实例，启动 M A T L A B，然后把目录变成 RSAVu_MIMO_Demo\MATLAB，运行 MATLAB 脚本 MATLAB_RSAVu_Demo.m。

用户参数设置

由于没有编译 MATLAB 实例，因此可以根据需要改变部分设置。用户最可能改变的设置位于文件顶部“User Setup Parameters”段中。fileRX1 和 fileRX2 中的文件名应指向以后将加载的 MIMO 信号的文件位置。在默认情况下，这些位置指向压缩文件中包括的 TIQ_Files 目录。

生成文本报告和保存的变量使用的文件名由 resultsFilename 变量设置，这个变量应该没有扩展名。在默认情况下，它设置为 MIMO_EVM_Results。程序在以后将为其创建的文本报告创建相应的扩展名.txt，为保存的变量创建相应的扩展名.mat。

然后，在 rsa1 和 rsa2 文件中存储捕获 MIMO 波形用的两台 RSA 的型号和序列号。程序只使用这两个文件，把信息写入文本报告顶部。

最后，可以改变 EVM 均方根值极限(默认值为 2.0)，这是用百分比表示的 EVM 值，程序将使用这个值确定 EVM 结果是通过还是未通过限制线测试。

RSAVu 远程控制演示

► 应用说明

VISA 设置

与 C++ 实例一样，第一步是设置 VISA 连接。与以前一样，我们将设置和打开到 GPIB8::1::INSTR 的连接，然后把超时周期设为 10 秒。注意在 C++ 实例中，这一属性用毫秒设置，而 MATLAB 命令中则使用秒设置。

```
instrVISA = visa( 'tek', ['GPIB8::1::INSTR'] );  
fopen( instrVISA );  
set( instrVISA, 'Timeout', 10.0 );
```

RSA 设置

然后，RSAVu 查询当前模式，如果还没有选择，那么切换到 802.11n MIMO 2x2 模式。如果不能选择（一般由于找不到有效 license），程序将退出。可以使用 fprintf 和查询命令，发送及从 RSAVu 提取数据。与 C++ 实例一样，为清楚起见，这里只复现了主命令，而没有包括初始化和错误校验等命令。

```
strMode = query( instrVISA, ':INSTRument:SElect?' );  
if length(strMode) < 11 ||  
strcmp(strMode(1:11), "DEMM2WLAN") == 0,  
    fprintf( instrVISA, ':INSTRument:SElect  
"DEMM2WLAN" );  
    strMode = query( instrVISA, ':INSTRument:SElect?' );  
    if length(strMode) < 11 || strcmp(strMode(1:11),  
"DEMM2WLAN") == 0,  
        % Can't set the proper mode, so time to quit.  
        fclose(instrVISA);  
        return;  
    end  
end
```

然后，为这一模式设置默认参数，然后选择子载波星座 (SCC) 测量。

```
fprintf( instrVISA, ':CONFigure:M2WLAN' );  
fprintf( instrVISA, ':SENSe:M2WLAN:MEASurement  
SCCConste' );
```

现在已经设置 RSAVu 进行 802.11n MIMO 2x2 分析，可以加载两个 MIMO 波形。

```
fprintf( instrVISA, [':MMEMory:LOAD:RX1 "  
fullpathRX1 "' ] );  
fprintf( instrVISA, [':MMEMory:LOAD:RX2 "  
fullpathRX2 "' ] );
```

与 C++ 实例一样，必须先加载 RX1 文件。然后，我们把分析长度设置成 6.5ms，这是我们希望分析的 64 个分组的长度。

```
fprintf( instrVISA, ':SENSe:M2WLAN:ANALysis:  
LENGth 6.5E-3' );
```

然后，把超时周期设置成 4 分钟，开始分析，等待操作完成，然后把超时周期设回到 10 秒。

```
set(instrVISA,'Timeout',240.0); % Timeout = 4 minutes  
fprintf( instrVISA, ':SENSe:M2WLAN:IMMEDIATE' );  
Wait4OPC( instrVISA ); % Wait for  
operation complete  
set(instrVISA,'Timeout',10.0); % Timeout = 10 seconds
```

在检测到操作完成时，准备开始提取测量结果。

从 RSAVu 中提取测量结果

与 C++ 实例中一样，下一步是对上一节中分析的 64 个分组，提取每个分组的测量结果。我们将把所有 SCC 测量数据存储到称为 SCC 的结构中，结构中的每个变量存储相关测量的 64 个结果，如用百分比表示的 rms EVM 和用 dB 表示的 rms EVM。由于 SCC 测量用文本字符串方式返回大量结果，因此演示实例解析字符串，找到 SCC 测量的各个结果。与前面一样，由于 RSAVu 从 -63 到 0 索引分析的 64 个分组，因此这一实例创建一个称为 iPacket 的变量，在循环计数器 i 从 1 递增到 64 时从 -63 递增到 0。在 MATLAB 版本中，我们从 1 启动循环计数器，而不是 C++ 版本中的从 0 开始，因为 MATLAB 的矢量索引从 1 开始，而 C++ 则从 0 开始。

```

numResults = 64;           % Number of packets to read
                           % from RSAVu

iPacket = -numResults;    % Match RSAVu's packet
                           % index (negative indexing)

for i=1:numResults,
    % Advance to the next packet in RSAVu.
    iPacket = iPacket + 1;

    fprintf( instrVISA, [':SENSe:M2WLAN:PACKet:
    NUMBer ' num2str(iPacket)];

    % Retrieve the measurement data for this packet
    str_SCC = query( instrVISA, [':FETCH:
    M2WLAN? SCConste'] );
    [temp,rem] = strtok(str_SCC,',');

    SCC.peakEVM_per(i)     = str2num(temp);
                           [temp,rem] = strtok(rem,',');

    SCC.peakEVM_dB(i)     = str2num(temp);
                           [temp,rem] = strtok(rem,',');

    SCC.rmsEVM_per(i)     = str2num(temp);
                           [temp,rem] = strtok(rem,',');

    SCC.rmsEVM_dB(i)     = str2num(temp);
                           [temp,rem] = strtok(rem,',');

    SCC.SC_number_EVM(i) = str2num(temp);
                           [temp,rem] = strtok(rem,',');

    SCC.peak_MError_per(i) = str2num(temp);
                           [temp,rem] = strtok(rem,',');

    SCC.peak_MError_dB(i) = str2num(temp);
                           [temp,rem] = strtok(rem,',');

    SCC.rms_MError_per(i) = str2num(temp);
                           [temp,rem] = strtok(rem,',');

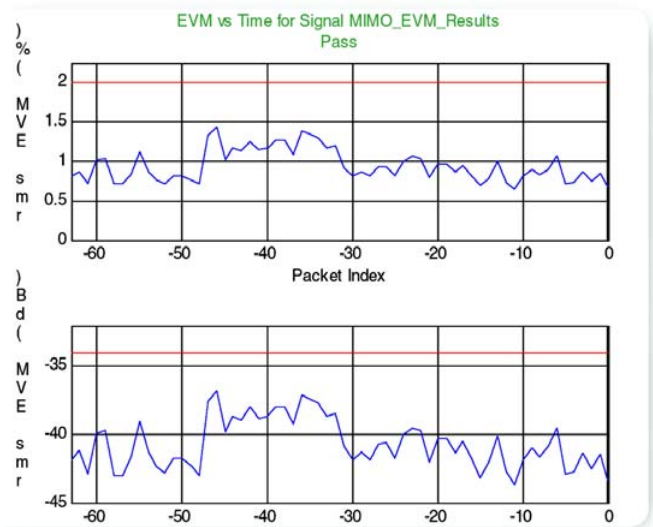
    SCC.rms_MError_dB(i) = str2num(temp);
                           [temp,rem] = strtok(rem,',');

    SCC.SC_number_MError(i) = str2num(temp);
                           [temp,rem] = strtok(rem,',');

    SCC.peak_PError_deg(i) = str2num(temp);
                           [temp,rem] = strtok(rem,',');

    SCC.rms_PError_deg(i) = str2num(temp);
                           [temp,rem] = strtok(rem,',');

    SCC.SC_number_PError(i) = str2num(temp);
end
    
```



► 图 3. EVM 均方根值与 2.0% 限制线相比测试通过图。

关闭会话

在从 RSAVu 采集所需的全部数据之后，可以关闭连接。
fclose(instrVISA);

处理结果

第一步是把采集的所有 SCC 测量数据保存到一个 MATLAB 数据文件中，以便在以后需要时检索。

```

save(resultsFilename,'SCC'); % Save all SCC results
                              into a MAT file
    
```

文件名基于我们在文件顶部“User Setup Parameters”中设置的文件名，自动增加扩展名.mat。然后，生成一份文本报告，其在很大程度上与 C++ 版本类似，唯一的差别是除保存 EVM 值的最大值和平均值外，我们还保存最小 EVM 值(用百分比和 dB)。

最后，创建一个图形，比较 EVM 均方根值与限制线，标明 PASS 或 FAIL 情况。未通过的每个点将在轨迹上标一个红色菱形。红色实线轨迹表示限制线，蓝色轨迹表示从 RSAVu 采集的 EVM 均方根值。如果所有点都低于限制线，标题将显示 PASS (绿色); 如果有一个或多个点超过极限，标题将显示 FAIL (红色)。图 3 是通过限制线的曲线实例。

泰克科技(中国)有限公司
上海市浦东新区川桥路1227号
邮编: 201206
电话: (86 21) 5031 2000
传真: (86 21) 5899 3156

泰克北京办事处
北京市海淀区花园路4号
通恒大厦1楼101室
邮编: 100088
电话: (86 10) 6235 1210/1230
传真: (86 10) 6235 1236

泰克上海办事处
上海市静安区延安中路841号
东方海外大厦18楼1802-06室
邮编: 200040
电话: (86 21) 6289 6908
传真: (86 21) 6289 7267

泰克广州办事处
广州市环市东路403号
广州国际电子大厦2807A室
邮编: 510095
电话: (86 20) 8732 2008
传真: (86 20) 8732 2108

泰克深圳办事处
深圳市罗湖区深南东路5002号
信兴广场地王商业大厦G1-02室
邮编: 518008
电话: (86 755) 8246 0909
传真: (86 755) 8246 1539

泰克成都办事处
成都市人民南路一段86号
城市之心23层D-F座
邮编: 610016
电话: (86 28) 8620 3028
传真: (86 28) 8620 3038

泰克西安办事处
西安市东大街
西安凯悦(阿房宫)饭店345室
邮编: 710001
电话: (86 29) 8723 1794
传真: (86 29) 8721 8549

泰克武汉办事处
武汉市武昌区民主路788号
白玫瑰大酒店924室
邮编: 430071
电话: (86 27) 8781 2760/2831
传真: (86 27) 8730 5230

泰克香港办事处
香港铜锣湾希慎道33号
利园3501室
电话: (852) 2585 6688
传真: (852) 2598 6260

如需了解更多信息, 请访问泰克公司网站: www.tektronix.com.cn



© 2007 年 Tektronix, Inc. 版权所有。 版权所有。 Tektronix 产品, 不论已获得专利和正在申请专利者, 均受美国和外国专利法的保护。 本文提供的信息取代所有以前出版的资料。 本公司保留变更技术规格和售价的权利。 TEKTRONIX 和 TEK 是 Tektronix, Inc. 的注册商标。 本文提及的所有其它商号分别为其各自所有公司的服务标志、商标或注册商标。 2/07 FLG/WOW 37C-20552-0

Tektronix
Enabling Innovation