



(12) 发明专利

(10) 授权公告号 CN 1440603 B

(45) 授权公告日 2011.07.13

(21) 申请号 01812417.8

代理人 杨凯 陈霁

(22) 申请日 2001.07.06

(51) Int. Cl.

(30) 优先权数据

00202378.6 2000.07.07 EP

01200038.6 2001.01.08 EP

H04L 1/24(2006.01)

H04L 27/26(2006.01)

(85) PCT申请进入国家阶段日

2003.01.06

(56) 对比文件

GB 2031197 A, 1980.04.16, 全文.

FR 2783374 A, 2000.03.17, 全文.

(86) PCT申请的申请数据

PCT/EP2001/007833 2001.07.06

审查员 张畅

(87) PCT申请的公布数据

W002/005473 EN 2002.01.17

(73) 专利权人 荷兰应用自然科学研究组织

地址 荷兰德尔夫特

(72) 发明人 R·F·M·范登布林克

B·M·范登霍伊维尔

(74) 专利代理机构 中国专利代理(香港)有限公

司 72001

权利要求书 2 页 说明书 22 页 附图 6 页

(54) 发明名称

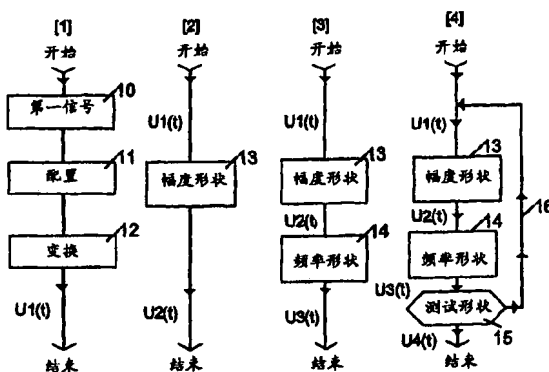
生成用于通信系统之中或之上具有预定质量标准

标准的信号的方法

(57) 摘要

这里公开了一种具有预定质量标准并用于通信系统的信号,一种生成这种信号的方法和系统,一种利用这种信号测试通信系统操作的方法以及配置为运用这种方法的(电信)通信系统。用于生成具有预定质量标准的方法包括以下步骤:表示第一信号(10),所述第一信号具有多个频率分量,每个频率分量具有频谱幅度性质和相位性质,以及通过配置(11)信号的频谱幅度性质来处理所表示信号,以及通过根据预定质量标准配置

(11) 信号的频谱幅度性质来处理所表示信号。



1. 一种配置用于通信系统之中或之上具有至少一种预定质量标准的信号的方法,所述方法包括以下步骤:

- 表示第一信号,所述第一信号包括多个频率分量,所述每个频率分量具有频谱幅度特性和相位特性,以及

- 通过根据所述或每种预定质量标准配置所述频谱幅度特性并配置随机相位特性来处理所述表示的第一信号,以及

- 将所述已处理的所表示的第一信号从频域变换到时域。

2. 如权利要求 1 所述的方法,其特征在于所述第一信号由指定所述各频率分量的频谱幅度和相位的第一组数字表示。

3. 如权利要求 1 所述的方法,其特征在于所述第一信号由具有实部和虚部的第二组复数表示,所述实部和虚部的组合指定所述各频率分量的频谱幅度和相位。

4. 如权利要求 1 所述的方法,其特征在于未经处理的第一信号由第三组数字表示,其中的每个数字在时域中指定所述第一信号的幅度。

5. 如权利要求 4 所述的方法,其特征在于还包括以下步骤:将所述第三组数字从时域变换到频域,从而由第四组数字表示所述第一信号,所述第四组数字指定所述各频率分量的频谱幅度和相位。

6. 如权利要求 4 所述的方法,其特征在于还包括以下步骤:将所述第三组数字从时域变换到频域,从而由具有实部和虚部的第五组复数表示所述第一信号,所述实部和虚部的组合指定所述各频率分量的频谱幅度和相位。

7. 如以上权利要求中的任何一项所述的方法,其特征在于还包括以下步骤:对所述的变换所述已处理的所表示第一信号所得的结果进行后处理,从而实现另一预定质量标准的进一步匹配。

8. 如权利要求 1 所述的方法,其特征在于所述的变换所述已处理的所表示第一信号所得的结果在时域中由第六组数字表示。

9. 如权利要求 1-6、8 中的任何一项所述的方法,其特征在于所述或各种预定质量标准包括至少一个已调载波,所述或各个已调载波包括由载频、载波幅度、调制深度和调制宽度组成的组中的任一些。

10. 如权利要求 1-6、8 中的任何一项所述的方法,其特征在于还包括以下步骤:根据预定时域幅度分布配置所述已处理的所表示第一信号。

11. 如权利要求 10 所述的方法,其特征在于还包括根据预定频谱幅度包络配置所述已处理的所表示第一信号。

12. 如权利要求 11 所述的方法,其特征在于由迭代过程进行所述时域幅度分布和频谱幅度包络中的至少一个。

13. 如权利要求 12 所述的方法,其特征在于所述迭代过程包括将所述已处理的所表示第一信号的所述时域幅度分布和频谱幅度包络中的任一些与预定时域幅度分布和预定频谱幅度包络进行比较。

14. 如权利要求 1-6、8、11-13 中的任何一项所述的方法,其特征在于具有所述预定质量标准的信号是噪声信号。

15. 一种测试通信系统的操作的方法,所述方法包括以下步骤:

-
- 根据以上权利要求中的任何一项生成具有至少一种预定质量标准的信号, 以及
 - 通过所述通信系统传送所述信号。

生成用于通信系统之中或之上具有预定质量标准的信号的方法

技术领域

[0001] 本发明一般涉及通信系统,更具体地说,涉及用于通信系统的信号,用于生成这种信号的方法和系统,利用这种信号测试通信系统的操作的方法,测试系统以及被配置为运用这种方法的(电信)通信系统。

[0002] 发明背景

[0003] 为了测试诸如 xDSL 收发信机以及电缆或网络的通信系统和通信设备,特别需要这样的测试信号,它们以代表每条电缆上具有大量的系统或系统装置的实际配置情况的方式加载到通信系统和通信装置。

[0004] 通过在实际(有噪声)测试条件下测量系统或系统装置的传输性能,人们可以改进系统或者装置的设计,和/或证明它们的性能符合诸如 ETSI、ITU 或 ANST 或其它(电信)通信团体发布的标准。

[0005] 一种执行这种性能测试的方法是生成被称为减损(impairment)的信号。更具体地说,可以将减损细分为:

[0006] (i) 串音噪声,它具有由例如来自相邻 xDSL 系统的谱包络和频谱幅度分布表征的噪声分布;

[0007] (ii) 干扰噪声,它由离散频率分量组成,也被称为射频干扰音(rfi-tone),具有由起源于例如无线电或业余广播的多个离散频率分量以及频谱幅度、调制深度和调制宽度参数表征的噪声分布,以及

[0008] (iii) 脉冲噪声,由例如开关操作和开关元件引起的信号脉冲表征。

[0009] 对于干扰噪声来说,频率可以随时间变化(摆动)。

[0010] 一种用于生成减损的装置被称为减损发生器,它被配置为生成所述串音噪声和干扰噪声中的至少一种,特别用于通信系统之中或之上。

[0011] 在实践中,为了测试通信系统和通信装置是否符合标准,已定义了各种噪声分布,其中这些噪声分布根据如诸如通信电缆中线对的长度和数量以及传输数据速率系统参数而变化。

[0012] 另外,诸如电缆、铜电缆或光纤或其它类型的电缆等各种不同类型或长度的传输媒体要求不同的噪声信号。

[0013] 用于生成噪声分布的方法和装置在本领域中是已知的。具体地说,众所周知,滤波技术和滤波器被用于从输入信号生成噪声,提供具有特定谱包络和频谱幅度分布的输出信号。

[0014] 但是,通过利用滤波技术和滤波器,在输入信号和输出信号间建立了因果关系。本领域的技术人员将意识到,这种类型的信号不太适合现实模拟实际操作的通信系统和通信装置。

[0015] WO 00/16181 公开了一种用于生成近似于预定幅度分布图的随机时域信号的方法和装置。在第一步骤中,通过对诸如白噪声信号的噪声信号进行滤波来创建信号,进而产生

具有预定谱包络的信号。在下一步骤中,将非线性函数应用于这种滤波后的噪声信号,以产生近似于预定幅度分布图的所需时域信号。再下一步,将脉冲响应滤波应用于时域信号,以校正其谱包络并得到具有所需谱包络的输出信号。非线性函数和脉冲响应滤波函数是根据提供的谱包络而选择的特殊函数。

[0016] WO 00/16181 限于这样一种情况:它仅提供一种只具有预定谱包络的时域信号。WO 00/16181 未对要提供的时域信号所具有的其它质量标准作出说明,特别未对相位特性作出说明。

[0017] 发明概述

[0018] 本发明的目的是提供一种改进信号,它被用于通信系统和通信装置,特别是用于根据预定(标准化)噪声分布测试这些系统和装置。

[0019] 在本发明的第一个方面中,公开了一种配置适合用在通信系统之中或之上具有预定质量标准的信号的方法,所述方法包括以下步骤:

[0020] - 表示第一信号,所述第一信号包括多个频率分量,每个分量具有频谱幅度特性以及相位特性,以及

[0021] - 通过根据所述或各种预定质量标准配置频谱幅度特性以及配置随机相位特性来处理所表示的第一信号。

[0022] 修改谱包络的传统方法是利用数字滤波器组。对本发明来说,所表示的第一信号和要提供的信号之间不必建立因果关系,所以这种传统方法决非理想。这种根据本发明对问题的理解大大简化了频率整形的完成。

[0023] 例如,从具有随机相位特性的第一信号开始,所表示的第一信号的频率整形操作足以提供满足预定质量标准的信号。可以以几种方式执行根据本发明的频率整形。

[0024] 在本发明的实施例中,第一信号由指定各频率分量的频谱幅度和相位的第一组数字表示。在保持信号的随机相位特性的同时,各频率分量的频谱幅度的调整足以实现频域中所表示信号的频率整形。

[0025] 在本发明的另一个实施例中,第一信号由具有实部和虚部的第二组复数表示,其中实部和虚部的组合指定各频率分量的频谱幅度和相位。通过充分地调整这些复数来实现频率整形,但是,在对所表示的第一信号进行调整之后要保持随机相位特性。

[0026] 在本发明的又一个实施例中,第一信号由第三组数字表示,各数字在时域中指定第一信号的幅度。通过利用例如快速傅里叶变换(FFT)算法将第三组数字从时域变换到频域,第一信号由指定各频率分量的频谱幅度和相位的第四组数字表示。可以由频率整形操作对第四组数字作进一步处理,如以上针对第一组数字所公开的那样。

[0027] 但是根据本发明,也可以将第三组数字从时域变换到频域,从而由第五组具有实部和虚部的复数表示第一信号。如上所述,为了频率整形,必须充分地调整第五组复数。

[0028] 在所表示的第一信号具有非随机相位特性的情况下,通过适当地配置第二、第四和第五组数字,可以得到随机相位特性。

[0029] 利用实数或复数的比例因子,可以由乘法运算调用频域中的缩放。将频率分量的频谱幅度的期望值除以其实际频谱幅度得到频率分量的频谱幅度的乘法比例因子。

[0030] 依照根据本发明的方法的另一个实施例,为了达到与所述或各种预定质量标准的进一步匹配,提供了已处理的所表示第一信号的后处理。

[0031] 但是,为了用于根据本发明的通信系统之中或之上,必须利用例如反快速傅里叶变换 (IFFT) 算法,将在频域中配置的所表示第一信号变换到时域。

[0032] 另外,以上公开的处理步骤还可以包括诸如信号的卷积或去卷积或者乘法或加法运算。在时域中,尤其可以由时域中第六组数字表示满足所述或各种预定质量标准的已处理的所表示第一信号。

[0033] 但是,通过以上方法,所提供的满足频率中诸如预定频谱幅度包络以及随机相位特性的质量标准的信号可能还不满足时域中诸如预定时域幅度分布的质量标准。

[0034] 在根据本发明的方法的再一个实施例中,所述或各种预定质量标准包括包含预定时域幅度分布以及预定频谱幅度包络的组中的任何一些。

[0035] 因此,在根据本发明的方法的另一个实施例中,根据预定时域幅度分布配置已处理的所表示第一信号。

[0036] 在根据本发明的方法的再一个实施例中,根据预定频谱幅度包络配置已处理的所表示第一信号。

[0037] 根据本发明,为了提供准确满足频域和时域中的预定质量标准的信号,时域幅度分布和频谱幅度包络中的至少一种是由迭代过程完成的。可以按照所需次数重复幅度和频率整形,直到两个形状都在合理精确度之内满足要求。

[0038] 在本发明的实施例中,迭代过程包括在任一迭代步骤后,将已处理的所表示第一信号的时域幅度分布和频谱幅度包络中的任一些与预定时域幅度分布与预定频谱幅度包络进行比较。

[0039] 已经发现,不需要在频率整形后执行全时域特征检查,以弄清时域特征是否足够接近要求。已证明,在实践中,波峰因数要求的简单检查足以能够决定是停止还是继续迭代。信号的波峰因数被定义为信号音平均或 rms (均方根) 值与信号音最大或峰值幅度的关系。

[0040] 以上公开的根据本发明的方法特别适合生成申音噪声等。

[0041] 在根据本发明的方法的第二方面中,如果要生成具有干扰噪声特征的信号,所述或各种预定质量标准包括至少一个已调载波,所述已调载波包括由载频、载波幅度、调制深度以及调制宽度组成的组中的任一些。

[0042] 通过根据一种质量标准或上述质量标准对所表示第一信号进行整形,可以容易并且非常有效地提供表示特定类型的干扰噪声、具有特定时域幅度分布以及预定频谱幅度包络的信号。

[0043] 根据本发明的方法,可以通过组合多个如上文所公开的内容处理的信号来提供满足所述或各种预定质量标准的信号。

[0044] 为了在例如通信网络或通信装置的测试中利用这种信号,必须利用例如 FFT 算法等将已处理的所表示信号从频域变换到时域。

[0045] 本发明还用于组合根据以上公开的第一和第二方面生成的信号。但也可以包括其它信号分量。

[0046] 具体地说,根据本发明的方法,具有所述或各种预定质量标准的信号是噪声信号。

[0047] 在本发明的第三方面,公开了一种测试通信系统的操作的方法,其中所述方法包

括以下步骤：

[0048] - 根据以上公开的本发明的方法,生成具有预定质量标准的信号,以及

[0049] - 通过测试中的通信系统传送这种信号。

[0050] 可以利用一组为代码格式并按预定顺序在装置上执行的指令来生成和存储所述信号。该组指令可以是软件代码,它在计算机上编译并存储在计算机或计算机网络或软盘或 CD-ROM 上,或通过因特网存储。软件和 / 或产生的信号还可以被存储在任意波形发生器 (AWG) 卡上,并且 AWG 可被用于生成这种信号或者从存储器再现存储信号。因此有可能存在一种可用信号库,其中这些信号存储在数据载体上,可被用于执行或使用根据本发明的方法。

[0051] 通信系统可以是诸如 xDSL 调制解调器、或这种调制解调器内的或用于这种调制解调器的芯片、或网络中的电缆、或用于 (电信) 通信的网络的装置。

[0052] 在本发明的第四方面中,公开了配置用于通信系统之上或之中的信号的另一种方法。信号最好是噪声信号。信号可包括串音噪声,它是在频域和时域中具有预定特性的随机信号。信号还可以包括射频干扰音,它具有离散频谱。这种信号中也可以包括其它信号分量。

[0053] 所述方法包括以下步骤：

[0054] - 在时域中表示具有时域幅度分布的第一信号,所述信号在频域中具有频谱密度,从而得到所表示信号；

[0055] - 根据非线性变换处理所表示信号,所述非线性变换得到至少一种预定质量标准；

[0056] - 至少通过预定频域幅度分布的反函数处理所表示信号的时域幅度分布。

[0057] 这种方法还可以包括以下步骤:将所表示信号的时域幅度分布与预定频域幅度分布进行比较,并在其后配置非线性变换从而得到已处理的所表示信号,该信号具有近似于预定频域幅度分布的时域幅度分布。

[0058] 在本发明的第五方面中,公开了一种方法,这种方法还包括以下步骤:将所表示信号的时域幅度分布与预定频域幅度分布进行比较,并在其后配置非线性变换从而得到已处理的所表示信号,该信号具有近似于预定频域幅度分布的时域幅度分布。

[0059] 根据本发明的第五方面,所述方法还可以包括以下步骤:在时域中表示具有幅度分布的第一信号,并且该信号在频域中具有频谱密度,从而得到所表示信号,以及在频域中对所表示信号进行滤波包括以下步骤:在频域中估计至少部分信号表示,并在其后在频域中处理所表示信号。

[0060] 可以组合本发明方法的第四和第五方面。本发明方法的第四和第五方面允许在不同的迭代步骤产生信号,其中这种信号具有预定幅度分布和 / 或具有预定频谱密度或具有根据预定质量标准的幅度分布和 / 或具有根据预定质量标准的频谱密度。预定质量标准可以是信号的波峰因数,它是信号音的最大值或峰值与信号音的平均值或 rms 值的关系。上文所列举的处理步骤可以包括以下步骤:快速傅里叶变换 (FFT) 或反快速傅里叶变换 (IFFT)。处理步骤还可以包括诸如信号的卷积或反卷积或者乘法或加法的运算。

[0061] 在本方法的第四方面中,对所表示信号的幅度分布进行处理包括预定幅度分布的函数,该函数可包括预定幅度分布的反函数。

[0062] 本发明的第四和第五方面所列举的方法还可以包括以下步骤：在频域中变换第一信号；将频域中的第一信号乘以谱包络从而得到相乘信号；以及之后在时域中表示相乘信号。

[0063] 在这些方法中，可将第一信号在频域中的表示生成为一组随机数、最好是复数，其中复数的模表示幅度，复数的幅角表示相位，以及可以根据高斯分布选择基本上每个复数的实部和 / 或虚部。各个复数可以基本上等于预定频谱密度的幅度。

[0064] 在本发明的第六个方面中，公开了一种信号，它包括至少一个随机噪声信号，这种随机信号具有根据预定质量标准的时域幅度分布以及根据预定质量标准的频域频谱密度，随机信号由随机数阵列组成。信号还可以包括离散频谱。可以利用一组为代码格式并按预定顺序执行的指令来生成噪声信号。该组指令可以是在计算机上编译并存储在计算机或计算机网络或软盘或 CD-ROM 上的、或通过因特网存储的软件代码。还可以将这种软件存储在任意波形发生器 (AWG) 卡上，并且 AWG 可被用于生成这种信号或从存储器再现存储信号。因此，有可能存在一种可用信号库，其中这些信号可被用于执行或使用本发明方法的第四和第五方面。

[0065] 在本发明的第七方面中，公开了一种生成包括至少一个随机噪声信号的信号，这种随机信号在时域中具有根据预定质量标准的幅度分布并在频域中具有根据预定质量标准的频谱密度。随机信号由随机数阵列构成，所述方法包括以下步骤：利用一组为代码格式并按预定顺序执行的指令来生成随机数组。所述方法还可以包括以下步骤：生成离散频谱，所述离散频谱利用角度测定函数并用噪声特性对基本上每个离散频率进行调制。可以利用一组为代码格式并按预定顺序执行的指令来组合随机噪声信号和离散频谱。

[0066] 在本发明的第八个方面中，公开了一组为代码格式并可按预定顺序执行的指令，这组指令被配置为生成随机噪声信号和离散频谱，这种随机信号在时域中具有根据预定质量标准的幅度分布并在频域中具有根据预定质量标准的频谱密度。这组指令可以是在计算机上编译并存储在计算机或计算机网络或软盘或 CD-ROM 上或通过因特网存储的软件代码。软件还可以被存储在任意波形发生器 (AWG) 卡上，并且这种 AWG 可被用于生成信号或从存储器再现存储信号。因此有可能存在一种可被利用的可用信号库。软件可以是 C 语言代码或可在 MATLAB 环境下编译。

[0067] 在本发明的第九方面中，公开了一种用于测试通信系统的操作的系统，它包括一组为代码格式的指令，该组指令可按预定顺序在装置上执行和编译，这组指令被配置为生成包括随机噪声信号和离散频谱中的至少一种的噪声信号，随机信号在时域中具有根据预定质量标准的幅度分布并在频域中具有根据预定质量标准的频谱密度。根据本发明的这个方面的测试系统可以包括用于生成噪声信号的减损发生器。

[0068] 将减损发生器连接到被测试的通信系统的连接部件（变压器、有源器件、衰减器等）可以具有不需要的频率从属响应。例如可以通过在减损发生器中生成特定测试信号来测量这种不需要的频率从属响应。可以通过乘以信号的期望频谱密度除以连接部件的不需要频率从属响应来补偿这种不需要的频率从属响应。

[0069] 在本发明的第十个方面中，公开了一种测试诸如 xDSL 调制解调器的通信系统的操作的方法，这种方法包括以下步骤：在经调制解调器收发的信号上叠加一个信号，其中所述一个信号包括随机噪声信号和离散频谱中的至少一种，所述随机信号在时域中具有根据

预定质量标准的幅度分布并在频域中具有根据预定质量标准的频谱密度,此外,所述噪声信号由随机数阵列构成。

[0070] 在本发明的第十一方面中,公开了一种测试通信系统的操作质量的方法。这种方法包括以下步骤:在经调制解调器收发的信号上叠加一个信号,其中所述一个信号包括随机噪声信号和离散频谱中的至少一种,所述随机信号在时域中具有根据预定质量标准的幅度分布并在频域中具有根据预定质量标准的频谱密度,此外,所述噪声信号由随机数阵列构成,并根据预定质量标准估计所收发的信号。

[0071] 另外,在本发明的第十二方面中,公开了一种改进通信系统的设计和/或生产的方法,这种方法包括以下步骤:在经调制解调器收发的信号上叠加一个信号,其中所述一个信号包括随机噪声信号和离散频谱中的至少一种,所述随机信号在时域中具有根据预定质量标准的幅度分布并在频域中具有根据预定质量标准的频谱密度,此外,所述噪声信号由随机数阵列构成;根据预定质量标准估计所收发的信号;以及迭代地配置调制解调器的设计,以便得到更接近用于估计所收发信号的质量标准。

[0072] 在本发明的第十三方面中,公开了包括这样一种信号的电信网络,这种信号包括随机噪声信号和离散频谱中的至少一种,这种随机信号在时域中具有根据预定质量标准的幅度分布并在频域中具有根据预定质量标准的频谱密度,另外,所述噪声信号由随机数阵列构成。

[0073] 可以组合上述各方面的特性和本发明的实施例。

[0074] 上述信号、方法和指令组将提供在诸如电话电缆或无线媒体等媒体上的更高质量信号传输。更好的信号传输允许为通信系统的用户在更大范围提供更多服务。

附图说明

[0075] 图1示意了利用根据本发明的方法操作的减损发生器在通信系统中进行性能测试的设置方框图。

[0076] 图2示意了根据本发明方法的流程图类型的实施例。

[0077] 图3在图示中示意了根据本发明用于生成干扰噪声的方法的实施例。

[0078] 图4示意了根据本发明方法的另一些流程图类型的实施例。

[0079] 图5示意了放大信号中高幅度峰值或信号音的幅度失真(非线性变换)函数 $Q(x)$ 。

[0080] 图6示意了本发明的例示实施例的流程图。

[0081] 图7-11示意了根据本发明的实施例得到的结果。

[0082] 实施例详述

[0083] 为了说明本发明,以下对本发明的信号和方法以及系统的各方面和实施例进行描述。本领域的技术人员将理解,可以设想并实现本发明的其它备选和等效实施例,并不背离本发明的真实精神。本发明的范围仅由所附权利要求书进行限制。

[0084] 在本发明的实施例中,公开了一种用于测试诸如xDSL收发信机的通信系统的操作的系统。对运行存取网络中的高穿透性系统情况的测试设备的设置进行描述。

[0085] 公开了一种配置用于通信系统之中或之上的信号的方法。

[0086] 传输性能测试的目的是以这样一种方式加载xDSL收发信机,这种方式表示运行

存取网络中的高穿透性系统情况。这种高穿透性方法允许：

[0087] (i) 组件和系统设计者量化性能,并用此改进他们的设计,从而证明符合标准;以及

[0088] (ii) 操作员定义应用于大多数运行情况的配置规则。

[0089] 图 1 示意了可能的测试设置 1 的功能描述。它包括：

[0090] ●测试回路 2,它是真电缆或电缆模拟器；

[0091] ●添加部件 3,它向测试回路 2 注入减损噪声；

[0092] ●高抗阻,以及非常均衡的差动电压探针 4,它连接到诸如频

[0093] 谱分析仪或 rms 电压表(未示出)等电平检测器 5,以及

[0094] ●测试中的 xDSL 收发信机(调制解调器)6、7。

[0095] 当测试中的 xDSL 系统需要外部分流器时(用于 POTS 或 ISDN 信号),可以将这些分流器包括在测试中的调制解调器 6、7 中。

[0096] 流经测试设备设置 1 的信号是从端口 Tx 到端口 Rx,这意味着测量上游和下游性能需要交换收发信机位置和测试“电缆”端。当端口 Tx 以及端口 Rx 的末端是测试中的 xDSL 收发信机(调制解调器)6、7 时,端口 Rx 的接收信号电平是节点 A2 和 B2 之间测量的电平。在这种测量期间断开减损发生器 8。在相同情况下,端口 Tx 的发送信号电平是节点 A1 和 B1 之间测量的电平。

[0097] 减损发生器 8 应注入到测试设置 1 中的噪声与频率有关。减损发生器 8 注入到测试设置 1 中的噪声应该是实际(频谱受影响)存取网络的实际表示,并且它是：

[0098] (a) 取决于测试回路 2 的长度,以及

[0099] (b) 在下游性能测试和上游性能测试中不同。

[0100] 在节点 A2 和 B2 之间测量的这种减损噪声通常是随机噪声、冲击噪声和谐波噪声(射频干扰音)的混合。一组特性被标识为“噪声分布”。

[0101] 用相当均衡的差动电压探针 4 测量信号和噪声电平。

[0102] 在全自动测试设置 1 中,测试回路 2、3、电压探针 4 以及电平检测器 5、测试中的调制解调器 6、7 以及减损发生器 8 可以连接到中央处理单元(CPU)9,如图中的虚线所示。本领域的技术人员将理解,与 CPU 9 的连接可涉及 CPU 9 进行的远程测试的数据链路。

[0103] 以下是与测试设备的使用有关的说明：

[0104] ●在满信号带上测量这种设置中的 rms 电压 U_{rms} [V],意味着 P [dBm] 的功率电平等于 $P = 10 \times \log_{10}(U_{\text{rms}}^2/R_v \times 1000)$ [dBm]；

[0105] ●在小频带 Δf (以赫兹(Hz)为单位)内测量这种设置中的 rms 电压 U_{rms} [V],意味着该滤波带内的功率频谱密度电平 P [dBm/Hz] 等于 $P = 10 \times \log_{10}(U_{\text{rms}}^2/R_v \times 1000/\Delta f)$ [dBm/Hz]；

[0106] ●带宽 Δf 标识滤波器的噪声带宽,并不是 -3dB 带宽。

[0107] 图 2 示意性地显示了用于根据本发明配置信号 $U_i(t)$ $i = 1, 2, 3, \dots$ 以用于通信系统之上或之中的方法的实施例。信号可包括串音噪声,它是在频域中和在时域中具有预定特性的随机信号。

[0108] 如图 2 的流程 [1] 所示,这种方法包括以下步骤:表示第一信号,所述第一信号包括多个频率分量,每个频率分量具有频谱幅度和相位特性,框 10 “第一信号”,并通过根据

至少一种预定质量标准配置频谱幅度特性来处理所表示信号,以及配置随机相位特性,框 11 “配置”,因此得到已处理的所表示信号。

[0109] 第一信号可由第一组数字表示,这组数字指定各频率分量的频谱幅度和相位。另外,第一信号可以由具有实部和虚部的第二组复数表示,实部和虚部的组合指定各频率分量的频谱幅度和相位。也就是说,复数的模表征频率分量的频谱幅度而复数的幅角表征频率分量的相位。

[0110] 根据本发明,对所表示第一信号 10 进行处理,以配置随机相位特性。但是,从具有随机相位特性的所表示第一信号 10 开始,对于根据预定质量标准对信号的频率进行整形,它足以对频率分量的频谱幅度进行整形。

[0111] 这种方法还可以包括以下步骤:在时域中表示第一信号,其中第一信号由第三组数字表示,其中每个数字指定第一信号在时域中的幅度。通过例如利用 FFT 算法将第三组数字从时域变换到频域,得到指定各频率分量的频谱幅度和相位的第四组数字。同样地,通过根据至少一种预定质量标准配置频谱幅度特性以及配置其随机相位特性来处理第四组数字。

[0112] 通过例如利用反 FFT 算法 (IFFT) 将已处理的所表示信号从频域变换到时域,框 12 “变换”,最终生成具有至少一种预定质量标准的信号 $U_1(t)$ 。

[0113] 尽管信号 $U_1(t)$ 满足频域中的至少一种预定质量标准,比如谱包络和 / 或预加重特性,还可能需要提供在时域中具有预定质量标准的信号。

[0114] 如图 2 中流程 [2] 和 [3] 所公开,即框 13 “幅度形状”和框 14 “频率形状”,时域中的质量标准可包括预定幅度分布和 / 或预定频谱幅度包络。

[0115] 这种方法还可以包括以下步骤:在不同的迭代步骤中产生信号,见图 2[4],框 15 “测试形状”和反馈回路 16。因此,信号可以具有预定时域幅度分布和 / 或预定频谱幅度包络和 / 或根据预定质量标准的频谱密度。

[0116] 至少一种预定质量标准可以是信号的波峰因数,也就是信号音的最大值或峰值与信号音的平均值或 rms 值的关系。

[0117] 可以利用一组为代码格式并可按预定顺序在装置上执行并编译的指令来生成并存储信号。这组指令可以是在计算机上编译并存储在计算机或计算机网络或软盘或 CD-ROM 上、或通过因特网存储的软件代码。这种软件还可以存储在任意波形发生器 (AWG) 卡上,并且 AWG 可被用于生成信号或从存储器再现存储信号。因此,有可能存在可用信号库,其中这些信号可被用于执行或使用本发明的方法的第一和第二方面。通信系统可以是诸如 xDSL 调制解调器 6、7 的装置,或这种调制解调器 6、7 内或用于这种调制解调器 6、7 的芯片或电信网络。

[0118] 进行图 2 的框 13 “幅度形状”中的处理,从而得到对时域特性的影响或控制。选择幅度失真(变换)函数 $Q(x)$ 来放大信号的高幅度峰值或信号音。可以从信号的实际幅度分布函数和预定幅度分布函数重构非线性变换函数 $Q(x)$ 。

[0119] 对于 0 到 T 之间时段 t 中的噪声信号 $f(t)$,信号的幅度分布 $F(a)$ 被定义为噪声 f 的绝对值大于 a 的时间部分。如果 $G(a)$ 是预定幅度分布(如增强高斯函数,见下文),并且 $G^{-1}(a)$ 是其反函数,则从噪声信号 $f(t)$ 产生中间或最终信号 $g(t)$ 的变换函数 $Q(x)$ 可被定义为:

[0120] $Q(x) = \text{sign}(x) \cdot G^{-1}(F(|x|))$ (1)

[0121] $g(t) = Q\{f(t)\}$ (2)

[0122] $\text{sign}(x) = x/|x|$ 其中 $x < 0$; $\text{sign}(x) = 0$ 其中 $x = 0$;

[0123] 因此, $g(t)$ 将具有预定幅度分布 $G(a)$ 。在许多情况下, $Q(x)$ 可以是解析函数但也可以用数字构造。增强高斯函数的例示如下:

[0124] 高斯类型噪声的幅度分布是:

[0125] $G(x) = 1 - \text{erf}\left(\frac{x}{\sqrt{2}\sigma}\right)$ (3)

[0126] 其中 $\text{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x dt \exp(-t^2)$ (4)

[0127] 并且其中 σ 是信号 RMS 值。

[0128] “增强”高斯分布被定义为:

[0129] $G(x) = \begin{cases} 1 - \left(a \frac{x}{A} + \text{erf}\left(\frac{x}{\sqrt{2}\sigma}\right) \right) / \left(a + \text{erf}\left(\frac{A}{\sqrt{2}\sigma}\right) \right) & 0 \leq x \leq A \\ 0 & x > A \end{cases}$, (5)

[0130] 如果 V_{RMS} 是噪声采样的期望 RMS 值, 并且 C_f 是期望波峰因数, 选择:

[0131] $A = C_f \times V_{\text{RMS}}$, 并且 (6)

[0132] $\sigma = \sqrt{((1+a)V_{\text{RMS}}^2 - A^2) \cdot a/3}$ (7)

[0133] 已证明有用的 a 的典型值是 0.001 和 0.01 之间的大小, 并且它表示增强高斯分布与实际高斯分布的偏差。

[0134] 在图 2 的框 14 频率形状中, 作为后处理步骤, 改进了信号的频域特性, 从而得到与所述或各种质量标准的进一步匹配。例如, 通过用 (中间) 信号 $U_2(t)$ 的测量频谱密度与预定频谱密度进行比较 (除), 可得到校正频率曲线。在下面对 FFT 函数的卷积的描述中的最佳模式实施例给出其例示。

[0135] 图 2 的部分 [4] 中, 示意了上文详细描述的步骤的迭代过程如何可以导致对用于通信系统之中或之上的完成或最终信号的进一步改进。执行迭代过程、即框 15 测试形状的频率形状测试以及反馈回路 16, 直到得到预定质量标准。

[0136] 图 3 在示意性图解中说明利用根据本发明的方法提供干扰噪声信号。

[0137] 如在前序中所公开的, 干扰噪声可以由离散载频 f_{ci} , $i = 1, 2, 3, \dots$ 上的多个频率分量表征。载频 f_{ci} 上的频率分量各自具有载波幅度 A_{ci} , $i = 1, 2, 3, \dots$, 以及 (如果可适用) 具有调制宽度、即相关载频 f_{ci} 的左手边和右手边的多个离散频率, 以及具有调制深度, 这种调制深度是与相应载频 f_{ci} 有关的边频幅度。

[0138] 图 3 在具有水平或频率轴 f 和垂直或幅度轴 A 的图形表示中, 仅作为例示显示了由两个载频 f_{c1} 和 f_{c2} 组成的信号, 这两个载频分别具有载波幅度 A_{c1} 和 A_{c2} 。

[0139] 载频 f_{c1} 的两侧旁配置了三个边频分量, 各边频具有幅度 A_1 。对于载频 f_{c2} 两侧的频率分量, 配置了两个边频分量, 各边频分量具有幅度 A_2 。

[0140] 根据本发明, 为了提供具有至少一种预定质量标准的信号, 必须对频率分量的幅度进行整形, 如在图 3 中用虚线 1 和 2 所示。

[0141] 从具有随机相位特性的所表示第一信号开始,根据本发明,通过对幅度进行整形,在要提供的具有预定质量标准的信号中保持随机相位特性。

[0142] 图 4 示意性地显示用于配置在通信系统之中或之上使用的信号的方法的另一实施例,具体地说,这种方法在已经提供随机相位特性时使用。该信号包括串音信号,它是在频域中和在时域中具有预定特性的随机信号。该信号还可以包括具有离散频谱的射频干扰音。另外,这种信号中可以包括其它信号分量。

[0143] 这种方法还可以包括以下步骤:在时域中表示第一信号,所述第一信号具有幅度分布,并且该信号在频域中具有频谱密度,进而得到所表示信号;以及根据非线性变换处理所表示信号,所述非线性变换实现预定质量标准。这在图 4 中被示为幅度整形,流程 [2-4]。

[0144] 这种方法还可以包括以下步骤:在时域中表示第一信号,所述第一信号具有幅度分布,并且该信号在频域中具有频谱密度,进而得到所表示信号;根据预定频谱密度质量标准处理所表示信号,直到得到具有频谱密度的信号。这在图 4 中被示为频率整形,流程 [2-4]。频率整形步骤还可以包括以下步骤:在频域中对所表示信号进行滤波,其中包括在频域中估计至少部分的信号表示,以及其后处理频域中的信号表示的步骤。

[0145] 这种方法还可以包括以下步骤:在不同的迭代步骤中产生信号,见图 4 的流程 [4]。因此,信号可以具有预定时域幅度分布和/或预定频谱密度或根据预定质量标准的时域幅度分布和/或频谱密度。预定质量标准可以是信号的波峰因数,也就是信号音的最大值或峰值与信号音的平均值比较的关系。可以利用一组为代码格式并可按预定顺序在装置上执行和编译的指令来生成并存储这种信号。

[0146] 同样地,这组信号可以是在计算机上编译并存储在计算机或计算机网络或软盘或 CD-ROM 上、或通过因特网存储的软件代码。这种软件还可以被存储在任意波形发生器(AWG)卡上,并且 AWG 可被用于生成信号或从存储器再现存储信号。因此,有可能存在可用信号库,它可以用于执行或使用本发明的方法。通信系统可以是诸如 xDSL 调制解调器、或这种调制解调器内或用于这种调制解调器的芯片、或电信网络的装置。在图 4 中详细说明了以下实施例。

[0147] 利用软件生成随机数,框 16“创建噪声”。在硬件中可以生成白噪声。对随机数进行过滤直到得到预定频谱密度。所生成的随机数各自表示一个频率分量。通过调整复数的幅度来执行得到预定频谱密度的必要处理,之后进行 IFFT 处理,从而产生期望噪声信号。执行这种方法的另一种方式是生成表示各频率分量相位的随机数,并在其后复数的幅度被配置为近似于或等于预定频谱密度。

[0148] 为得到对时域特性的影响或控制,进行框 13“幅度形状”的处理。图 5 示意了选择幅度失真(变换)函数 $Q(x)$ 来放大信号中的高幅度峰值或信号音。如以上针对公式 (1-7) 所述的,可以从信号的实际幅度分布函数和预定幅度分布函数重构非线性变换函数 $Q(x)$ 。

[0149] 在图 4 的框 14“频率形状”中,改进了信号的频域特性。通过用(中间)信号的预定频谱密度与测量频谱密度进行比较(除)得到校正频率曲线。在后续对 FFT 函数的卷积中描述的最佳模式实施例中给出其例示。

[0150] 在图 4 的流程 [4] 中,就象图 2 中流程 [2] 一样,再次示意了上文详细描述的步骤的迭代过程如何可以导致用于通信系统之上或之中的最终进一步改进。执行迭代过程,直到得到预定质量标准。

[0151] 如上所述,通过根据本发明的方法,利用诸如减损发生器 8 的装置,可以生成表示串音噪声和干扰噪声的信号,见图 1,其中的减损发生器可被配置为提供由串音噪声信号和干扰噪声信号构成的信号,同时如果需要,可以在要提供的输出信号中加入其它信号分量。

[0152] 在本发明的实施例中,可以有利地将要提供的信号作为时域中的第六组数字提供,例如数字阵列。

[0153] 图 6 在流程类型的图中示意了运行在个人计算机 20 上的本发明的例示实施例。被称为 SPOCS 的框 21 生成减损噪声,它包括框 22 “白噪声”、FFT 提供的框 23 “频谱整形”、IFFT 从框 23 的输出产生的框 24 “期望噪声信号”,框 24 的结果信号被存储在 AWG 卡 25 中。在串音情况中,即框 26,建立噪声 PSD(功率频谱密度),框 27,由框 23 对噪声 PSD 进一步处理。

[0154] 下文公开了本发明的一组指令的最佳模式实施例。下文给出的代码在 MATLAB 环境下编译。%符号后面给出了关于代码功能的注释。对于本领域的技术人员来说,所提供的代码是不解自明的。

[0155] 图 7-9 示意了通过最佳模式实施例得到的结果。图 8 示意了生成噪声样本以及噪声分布的 PSD 的频谱曲线。图 9 示意了生成噪声样本在时域中的曲线。图 10 示意了生成噪声样本的分布函数曲线。图 11 示意了生成噪声样本的累积分布函数曲线。图 7 示意了图形用户界面以及 AWG 控制的设置。

```
[0156] % -----
-----
[0157]     函数 DemoImpair2 ;
[0158] % -----
-----
[0159] % DemoImpair2
[0160] % Matlab 编程语音编写的代码,演示了减损发生器的基本算法。
[0161] %
[0162] %所演示的算法全面控制预定质量标准,例如 :
[0163] %     - 在用连续频谱产生噪声时的频域和时域特性(幅度;概率分布)
[0164] %     - 在用离散频谱生成噪声时的载波幅度、载频、调制深度和调制
[0165] %宽度
[0166] %独立计算两种类型的噪声,并且在时域中作为数字阵列表示。
[0167] %可以通过增加这些阵列元素的方式同时得到两种类型的噪声。
[0168] %
[0169] %
[0170] %
[0171] %
[0172] % (c) 2000-2001 KPN Research ;
[0173] %
[0174] %演示函数
[0175] % DemoXtalkNoise           - 表示产生连续噪声的过程
```

```

[0176] % DemoIngressNoise      - 表示产生离散噪声的过程
[0177] %
[0178] %主函数 :噪声被表示为随机数阵列
[0179] % DefineShape              - 初始化所有用户可定义参数
[0180] % CreateNoiseCont           - 产生连续噪声
[0181] % CreateNoiseDiscr_Fast     - 产生离散噪声,快速算法
[0182] % CreateNoiseDiscr_Slow     - 产生离散噪声,慢速算法
[0183] % FrequencyShape            - 修改连续噪声的频谱密度
[0184] % AmplitudeShape            - 修改连续噪声的幅度分布
[0185] %
[0186] %支持函数 :
[0187] % CalcSpec                    - 计算噪声的频谱密度
[0188] % CalcNBSV                    - 计算噪声的窄带信号电压
[0189] %
[0190] % CalcCrest                    - 计算噪声的波峰因数
[0191] % CalcDistrib                  - 计算噪声的概率分布
[0192] % CalcCumDistrib              - 计算噪声的累积分布
[0193] % CalcSmooth                  - 平滑频谱,如在实际频谱分析仪中那样
[0194] %
[0195] % CalcEnhancedGaussDistribution - 近高斯分布的示例
[0196] % CalcDemodulation            - 计算载波上调制的噪声
[0197] % -----
-----
[0198] Shape = DefineShape ;
[0199] DemoXtalkNoise(Shape) ;
[0200] DemoIngressNoise(Shape) ;
[0201] % -----
-----
[0202] function[U,t] = DemoXtalkNoise(shape) ;
[0203] % -----
-----
[0204] %演示用连续频谱生成噪声
[0205] %如对于串音测试
[0206] R      = Shape.R ;
[0207] CF_min = Shape.Xtalk.CF_min ;
[0208] [U,t] = CreateNoiseCont(Shape);plot(t,U) ;          title(' Xtalk method
1' ) ;shg ;pause
[0209] [X,f] = CalcSpcc(U,t) ;          plot(f,X) ;          title(' Xtalk method
1) ;shg ;pause

```

```

[0210] [X, f] = CalcSpec(U, t);          plot(f, dBm(X, R));title(' Xtalk method
1' );shg;pause
[0211] [P, u] = CalcCumDistrib(U);      plot(u,P);          title(' Xtalk method
1' );shg;pause
[0212] %
[0213] U = AmplitudeShape(U, Shape);    plot(t,U);          title(' Xtalk method
2' );xhg;pause
[0214] [X, f] = CalcSpec(U, t);          plot(f, X)          title(' Xtalk method
2' );shg;pause
[0215] [X, f] = CalcSpec(U, t);          plot(f, dBm(X, R));title(' Xtalk method
2' );shg;pause
[0216] [P, u] = CalcCumDistrib(U);      plot(u,P);          title(' Xtalk method
2' );shg;pause
[0217] %
[0218] U = FrequencyShape(U, Shape);    plot(t, U);          title(' Xtalk method
3' );shg;pause
[0219] [X, f] = CalcSpec(U, t);          plot(f, X);          title(' Xtalk method
3' );shg;pause
[0220] [X, f] = CalcSpec(U, t);          plot(f, dBm(X, R)); title(' Xtalk method
3' );shg;pause
[0221] [P, u] = CalcCumDistrib(U);      plot(u,P);          title(' Xtalk method
3' );shg;pause
[0222] %
[0223] for i = 2:10
[0224]     i
[0225]     U = AmplitudeShape(U, Shape);
[0226]     [X, f] = CalcSpec(U, t);
[0227]     %
[0228]     U = FrequencyShape(U, Shape);
[0229]     [X, f] = CalcSpec(U, t);
[0230]     %
[0231]     if CalcCrest(U) > CF_min, break;end;
[0232] end;
[0233] [P, u] = CalcCumDistrib(U);
[0234] plot(t, U);          title(' Xtalk method 4' );shg; %暂停
[0235] plot(f, dBm(X, R)); title(' Xtalk method 4' );shg; %暂停
[0236] plot(u, P);          title(' Xtalk method4' );shg; %暂停
[0237] % -----

```

```

[0238] function[U, t] = DemoIngressNoise(Shape) ;
[0239] % -----
-----

[0240] %演示用离散频谱生成噪声,如对于干扰测试
[0241] R = Shape.R ;
[0242] [U, t] = CreateNoiseDiscr_Fast(Shape) ;
[0243] % [U, t] = CreateNoiseDiscr_Slow(shape) ; %给出同一结果
[0244] [X, f] = CalcNBSV(U, t) ;plot(f, dBm(X, R)) ;title(' Ingress method' ) ;
shg ;pause
[0245] for ToneNr = [1:2]
[0246] [Uac, Uac_rms] = CalcDemodulation(U, t, Shape, ToneNr) ;
[0247] plot(t, Uac) ;title(' demodulated ingress noise of one carrier' ) ;shg ;
pause
[0248] [P, u] = CalcDistrib(Uac/Uac_rms) ;
[0249] plot(u, P) ;title(' distribution of demod noise' ) ;shg ;pause
[0250] end ;
[0251] % -----
-----

[0252] function[Shape] = DefineShape ;
[0253] % -----
-----

[0254] %创建应该生成的噪声的噪声分布,按照
[0255] % - 频谱密度(本例示中实际上是矩形)
[0256] % - 概率分布(本例示中是近高斯分布)
[0257] % - 信号音和调制
[0258] %以伏特每平方根赫兹为单位的频谱
[0259] % -----
-----

[0260] Fmax = 4E6 ;F1 = 300E3 ;Fh = 700E3 ;N = 2^18 ;R = 135 ;
[0261] cf = 5.5 ; %期望波峰因数
[0262] cf_min = 5.1 ; %期望波峰因数
[0263] m = N/2 ;
[0264] %
[0265] Shape.N = N ; %时间取样数
[0266] Shape.m = m ; %频率取样数
[0267] Shape.dF = Fmax. / (m-1) ; %频率间隔
[0268] Shape.dT = 1 / (N*Shape.dF) ; %时间间隔
[0269] Shape.R = R ; %期望噪声源的阻抗
[0270] %定义串音噪声目标(频谱密度&幅度分布)

```

```

[0271] Shape.Xtalk.freq = [0:m-1]' *Shape.dF;
[0272] Shape.Xtalk.spec = (Shape.Xtalk.freq >= Fl).*(Shape.Xtalk.freq <=
Fh)*(1/300);
[0273] Shape.Xtalk.DistU = 0:cf/1000:cf;
[0274] Shape.Xtalk.Distp = CalcEnhancedGaussDistribution(Shape.Xtalk.DistU,
cf); % P
[0275] Shape.Xtalk.CF_min == cf_min;
[0276] %定义干扰噪声目标 (RFI 音)
[0277] P_dBm = [-70;-50;-60;-60;-40;-60;-60;-40;-70;-40]; % dBm@135 ohm
[0278] P = (10).^(P_dBm/10)*1E-3;
[0279] Shape.Ingress.ToneU = sqrt(P*135); % U = sqrt(P*R);有效值
[0280] Shape.Ingress.ToneF = [99;207;333;387;531;603;711;801;909;981]*1E3;
[0281] Shape.Ingress.ModDepth = 0.32*ones(10,1); % = mod index > 0.8, at CF >
2.5)
[0282] Shape.Ingress.ModWidth = 2*4.5E3*ones(10,1); % = -10kHz..+10kHz)
[0283] % -----

[0284] function[U, t] = CreateNoiseCont(Shape);
[0285] % -----

[0286] %创建噪声电压 U(t),具有预定频域特性(频谱),但具有无控制的时域特性
[0287] % (分布)
[0288] N = Shape.N; %要产生的取样数
[0289] % U = rand(N,1); %均匀分布的白噪声
[0290] U = randn(N,1); %高斯分布的白噪声
[0291] U = FrequencyShape(U, Shape); %整形噪声
[0292] t = [0:N-1]' *Shape.dT; %相关时间轴
[0293] % -----

[0294] function[U, t] = CreateNoiseDiscr_Fast(Shape);
[0295] % -----

[0296] %创建电压 U(t),具有 AM 已调载波 (RFI 音);每个具有单独的预定频率、
[0297] %幅度、调制宽度以及调制深度。
[0298] %每个载波音下边带的随机相位被镜像,以将任意 QAM 调制转换到
[0299] % (不是镜像为)更严格的 AM 调制(全镜像)
[0300] %标记:本算法中 X 指的是(近谐波)干扰噪声信号的傅里叶级数的分量,
[0301] %但是它在(伪随机)串音噪声信号的情况下指的是频谱密度
[0302] %

```

```

[0303] %
[0304] %计算时间随着取样数量大约线性增长
[0305] %全部计算时间中的大约 80%由反傅里叶变换造成
[0306] % -----
-----
[0307] N = Shape.N; %取样数量
[0308] m = Shape.m; %该数量的一半
[0309] Nc = round(Shape.Ingress.ToneF/Shape.dF)+1; %载频的指数(只有正)
[0310] Nm = round(Shape.Ingress.ModWidth/Shape.dF/2); %调制次数
[0311] %分量
[0312] Xc = 0.5*Shape.Ingress.ToneU; %载波幅度的幅度
[0313] Xm = Shape.Ingress.ModDepth.*Xc./sqrt(2*Nm); %调制频带的幅度
[0314] X = zeros(N,1); %初始化
[0315] Xc = Xc.*exp(j*1000*rand(size(Xc))); %随机载波相位
[0316] Xcc = (Xc.*Xc)./abs(Xc.*Xc);
[0317] for k = 1:length(Nc) %对于所有已调载波,进行:
[0318] Nmp = Nc(k)+[1:Nm(k)]'; %定位上边带
[0319] %频率
[0320] Nmn = Nc(k)-[1:Nm(k)]'; %定位上边带
[0321] %频率
[0322] Xmp = Xm(k).*exp(j*1000*rand(size(Nmp))); %创建上边带
[0323] Xmn = conj(Xmp)*Xcc(k); %镜像下边带
[0324] X(Nmp) = Xmp; %插入上边带
[0325] X(Nmn) = Xmn; %插入下边带
[0326] end;
[0327] X(Nc) = Xc; %插入所有载波
[0328] X(N:-1:m+2) = conj(X(2:ceil(m))); %附加频谱
[0329] % (负频率)
[0330] % U = real(ifft(X))*N; %变换到时域
[0331] U = real(fft(X)); %变换到时域
[0332] % (快了 10%)
[0333] t = Shape.dT*[0:N-1]'; %相关时间轴
[0334] % -----
-----
[0335] function[U,t] = CreateNoiseDiscr_Slow(Shape);
[0336] % -----
-----
[0337] %创建电压 U(t),具有预定频率上的 RFI 音、幅度和调制宽度以及调制

```

```

[0338] %深度。
[0339] %本算法是直接的,非常低效的,并且只用于演示
[0340] %可以证明 CreateNoiseDiscr_Fast 返回相同的结果
[0341] % -----
-----
[0342] N = Shape.N; %要生成的取样数量
[0343] m = Shape.m;
[0344] f = [0:N-1]' *Shape.dF;
[0345] t = [0:N -1]' *Shape.dT;
[0346] Fc = Shape.Ingress.ToneF; %载波频率列表
[0347] Fc = Shape.dF*round(Fc/Shape.dF); %强制整数周期
[0348] %
[0349] U = 0;
[0350] for k == 1:length(Shape.Ingress.ToneF);
[0351] % --产生噪声调整,具有 U_avg = 0 并且 U_rms = 1
[0352] Nm = round(Shape.Ingress.ModWidth(k)/Shape.dF/2);
[0353] Xn0 = ([1:N] <= (Nm+1))'; %形状调制噪声幅度
[0354] Xn = Xn0.*exp/(j*1000*rand(N,1)); %形状调制噪声相位
[0355] Xn(1) = 0; %消去 DC 分量
[0356] Xn(N:-1:m+2) = conj(Xn(2:ceil(m))); %附加频谱(负频率)
[0357] Noise = real(ifft(Xn)); %变换到时域
[0358] Noise = Noise/sqrt(sum(Noise.*Noise)/N); %强制 rms = 1
[0359] % -- 执行调制
[0360] Carrier = Shape.Ingress.ToneU(k) *cos(2*pi*Fc(k)*t+1000*rand);
[0361] Modulate = Shape.Ingress.ModDepth(k)*Noise;
[0362] U = U+Carrier.*(1+Modulate);
[0363] end;
[0364] % -----
-----
[0365] function[U] = FrequencyShape(U, Shape)
[0366] % -----
-----
[0367] %重新整形取样 U 的频谱,如目标形状指定的
[0368] %输入:
[0369] % U: 取样的连续值
[0370] % Fs: 取样频率
[0371] % 频谱: 期望 PSD(以伏特每平方根赫兹为单位)
[0372] % -----
-----

```

```

[0373] N = length(U) ;
[0374] m = length(Shape.Xtalk.spec) ; % m = N/2
[0375] t = [0:N-1]' *Shape.dT ;
[0376] %执行频率调整
[0377] Scaling = Shape.Xtalk.spec./CalcSpec(U, t) ;
[0378] X = fft(U) ; %变换到频域
[0379] X(1) = 0 ; %消去 DC 分量
[0380] X(2:m+1) = X(2:m+1).*Scaling ; %调整频谱 (正频率)
[0381] X(N:-1:m+2) = conj(X(2:ceil(m))) ; %附加频谱 (负频率)
[0382] U = real(ifft(X)) ; %变换到时域
[0383] % -----

-----

[0384] function[U] = AmplitudeShape(U, Shape)
[0385] % -----

-----

[0386] %本函数利用幅度相关 (非线性) 失真函数 Q(x) 对函数 U 的幅度分布进行
[0387] %整形。
[0388] %结果是  $U(t) = Q\{U(t)\}$ 
[0389] %
[0390] %令 FF 为取样的实际累积分布函数, 以及
[0391] %令 GG 为期望累积分布函数,
[0392] %则给出的失真函数为:
[0393] %  $Q(x) = GG^{-1}\{FF(x)\}$ 
[0394] % -----

-----

[0395] U0 = sqrt(sum(U.*U)/length(U)) ; %缩放因子
(用于标准化)
[0396] %计算失真函数 Q
[0397] [DistP1, DistU1] = CalcCumDistrib(U/U0) ; %实际分布
[0398] Q = interp1(Shape.Xtalk.DistP, Shape.Xtalk.DistU, DistP1) ; %失真函数
[0399] U = U0*interp1(DistU1, Q, abs(U/U0)).*sign(U) ; %执行失真
[0400] % plot(DistU1, Q) ;shg ; %暂停
[0401] % -----

-----

[0402] function[X, f] = CalcSpec(U, t) ;
[0403] % -----

-----

[0404] %当要在指定分辨率带宽上 '测量' 时, 计算信号的频谱密度
[0405] %

```

```

[0406]   RBW = 1E3 ;                               % RBW :U 的频谱的期望分辨率
[0407]   N = length(U) ;m = N/2 ;
[0408]   dT = t(2)-t(1) ;                             %时间间隔
[0409]   dF = 1/dT/N ;                               %频率间隔
[0410]   f = [0:m-1]' *dF ;                           %所有正频率
[0411]   X = fft(U)*dT ;                               %到频域
[0412]   X = abs(X(2:m+1)) ;                           %无 DC 并且无负频率
[0413]   X = sqrt(CalcSmooth(X.*X, f, RBW)) ; %在整个带宽 RBW 上对它进行平均
[0414]   % -----

[0415]   function[X, f] = CalcNBSV(U, t) ;
[0416]   % -----

[0417]   %当要在指定分辨率带宽上‘测量’时,计算信号的频谱密度
[0418]   %
[0419]   RBW = 1E3 ;                                     % RBW :U 的频谱的期望分辨率
[0420]   N = length(U) ;m = N/2 ;
[0421]   dT = t(2)-t(1) ;                             %时间间隔
[0422]   dF = 1/dT/N ;                               %频率间隔
[0423]   f = [0:m-1]' *dF ;                           %所有正频率
[0424]   % X = fft(L)*dT*sqrt(dF) ;                     %到频域
[0425]   % X = fft(L)/N ;                               %到频域
[0426]   X = fft(U)/N*2 ;                               %到频域
[0427]   X = abs(X(2:m+1)) ;                           %无 DC 并且无负频率
[0428]   % X = sqrt(CalcSmooth(X.*X, f, RBW)) ;       %在整个带宽 RBW 上对它进行平均
[0429]   % -----

[0430]   function[CF] = CalcCrest(U)
[0431]   % -----

[0432]   %计算信号 U(t) 的波峰因数,它是峰值除以 rms 值
[0433]   %
[0434]   Urms = sqrt(sum(U.^2)/length(U)) ;
[0435]   Upeak = max(abs(U)) ;
[0436]   CF = Upeak/Urms ;
[0437]   % -----

[0438]   function[F] = CalcEnhancedGaussDistribution(x, Cf) ;

```

```

[0439] % -----
-----

[0440] %生成累积分布函数 F(x),它被标识为‘增强高斯分布’
[0441] %
[0442] % Cf =波峰因数
[0443] Alpha = 1e-3 ;
[0444] Sigma = sqrt((1+Alpha)-Cf^2*Alpha/3) ;
[0445] x = x.*(x > 0).*(x < Cf)+Cf*(x >= Cf) ;
[0446] denominator = Alpha+erf(Cf/(sqrt(2)*Sigma)) ;
[0447] F = 1-(Alpha*x/Cf+erf(x/(sqrt(2)*Sigma)))/denominator ;
[0448] % -----
-----

[0449] function[DistP,DistU,P] = CalcDistrib(U)
[0450] % -----
-----

[0451] %计算信号 U 的幅度分布
[0452] N = length(U) ;
[0453] Nbins = 100 ;
[0454] [cumbin,xx] = hist(U,Nbins) ;
[0455] dX = xx(3)-xx(2) ;
[0456] DistP = cumbin(:)/N/dX ; %强制 sum(DistP)*dX
[0457] DistU = xx(:) ;
[0458] Urms = sqrt(sum(U.*U)/N) ;
[0459] P = exp(-0.5*(DistU/Urms).^2) ;P = P/sum(P)/dX ;
[0460] DistU = [DistU,DistU] ;
[0461] DistP = [DistP,P] ;
[0462] % -----
-----

[0463] function[DistP,DistU] = CalcCumDistrib(U)
[0464] % -----
-----

[0465] %计算信号 U 的(向后)累积幅度分布
[0466] len = length(U) ;
[0467] Ueff = sqrt(sum(U.*U)/length(U)) ;
[0468] U = abs(U/Ueff) ;
[0469] % --- 估计分布函数
[0470] Nbins = min([50,floor(len/10)]) ;
[0471] [cumbin,xx] = hist(U,Nbins) ;
[0472] BinWidth = xx(2)-xx(1) ;

```

```

[0473] DistU = xx-BinWidth/2; %偏移
[0474] for n = [Nbins-1:-1:1];cumbin(n) = cumbin(n)+cumbin(n+1);end
[0475] DistP = cumbin/len;
[0476] % --- 当它们利用这个结果时,改进其它程序的数值稳定性
[0477] DistU = [0,DistU(2:end)]; %从 x = 0 开始
[0478] DistP = [DistP,1/len];
[0479] DistU = [DistU,xx(Nbins)+0.999*BinWidth/2]; %增加最终(信号)点
[0480] DistP = [DistP,1e-100];
[0481] DistU = [DistU,xx(Nbins)+(1.001)*BinWidth/2]; %因数 1.001 得到稳定性
[0482] % -----

[0483] function[PSD,freq] = CalcSmooth(PSD,freq,RBW)
[0484] % -----

[0485] %模拟实际频谱分析仪,具有有限分辨率的带宽以及高斯整形带通滤波器
[0486] % PSD = "功率频谱密度",它是"频谱密度"的去平方(de square),
[0487] %单位为伏特每平方赫兹
[0488] %
[0489] % -----

[0490] \ = length(PSD);
[0491] df = freq(2)-freq(1);
[0492] br = 3*floor(RBW/df);
[0493] factor = 2*br+1;
[0494] if(factor > 1)
[0495] .. ff = df*(-br:br); %平滑间隔
[0496] .. mask = exp(-ff.*ff/(2*RBW^2));
[0497] .. mask = mask/sum(mask); %分辨率带通滤波器的高斯掩蔽
[0498] .. xhelp = [PSD;zeros(2*br,1)];
[0499] lp = filter(mask,1,xhelp); %智能卷积
[0500] PSD = yhelp(br+1:end-br);
[0501] end;
[0502] % -----

[0503] function[Uac,Uac_rms] = CalcDemodulation(U,t,Shape,ToneNr);
[0504] % -----

[0505] %解调已在离散噪声的载波上调制的噪声,只是为了演示,以证明离散噪声
[0506] %满足预定参数

```

```

[0507] %
[0508] %解调器利用同步检测,不是锁相的
[0509] %结果是全解调频带上的未知衰减
[0510] %通过测量 DC 电平进行后向校正,并且放大解调信号直到此 DC 电平被
[0511] %归一化为 1 伏特
[0512] %
[0513] %证明:(psi 未知)
[0514] %
[0515] %令  $Urf = \cos(w*t+psi)*(1+Uac)$ ; %=用“1+Uac”调制载波
[0516] %  $Uc = \cos(w*t)$ ; %=载波
[0517] %  $Ud = Urf*Uc$ ; %=同步检测信号
[0518] %
[0519] %则  $Ud = 1/2*(\cos(psi+2*w*t)+\cos(psi))*(1+Uac)$ ;
[0520] %  $Ulf = \cos(psi)/2*(1+Uac)$ ; %低通滤波之后
[0521] %  $Udc = \cos(psi)/2$ ; %=通过平均 Ulf
[0522] %  $Uac = (Ulf/Udc)-1$ ;
[0523] % -----
-----
[0524] N = Shape.N;
[0525] Fc = Shape.Ingress.ToneF(ToneNr); %选择载频
[0526] Fc = Shape.dF*round(Fc/Shape.dF); %强制整数周期
[0527] ModWidth = Shape.Ingress.ModWidth(ToneNr);
[0528] ModDepth = Shape.Ingress.ModDepth(ToneNr);
[0529] %
[0530] Ud = U.*cos(2*pi*/Fc*t); %已调载波的同步检测
[0531] Nm = round(1.1*ModWidth/Shape.dF/2); %计算滤波频率
[0532] mask = zeros(N,1);mask([1:Nm,N-Nm:N]) = 1; %创建滤波器
[0533] Ulf = real(iffilt(fft(Ud).*mask)); %执行低通滤波
[0534] Udc = sum(Ulf)/N; %找到未归一化 DC 电平
[0535] Uac = Ulf/Udc-1; %归一化整个电平,并且
去掉 DC
[0536] %
[0537] Uac_rms = sqrt(sum(Uac.*Uac)/N); %必须等于 ModDepth,因为 Udc = 1
[0538] Scale = Uac_rms/ModDepth; %必须为“1”

```

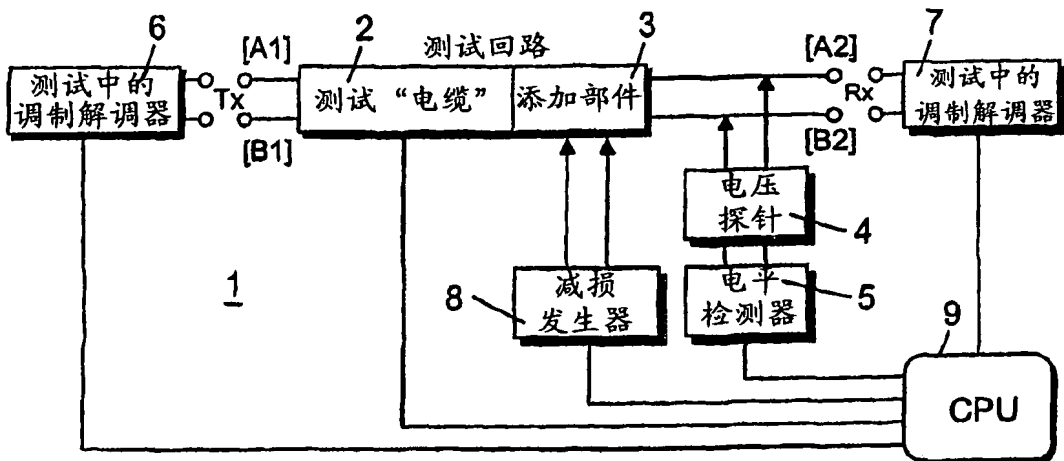


图 1

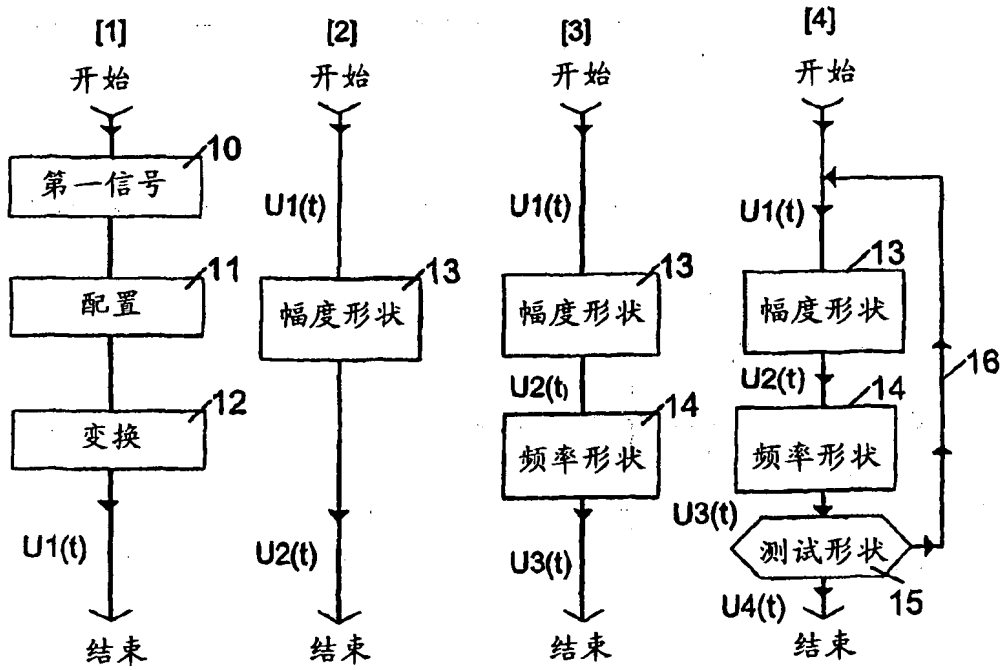


图 2

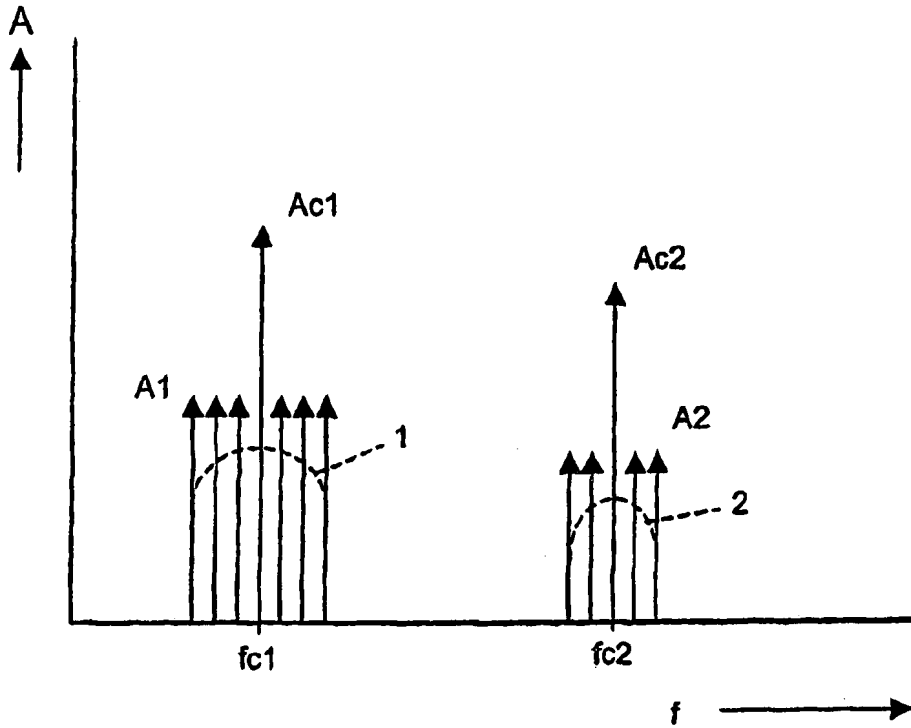


图 3

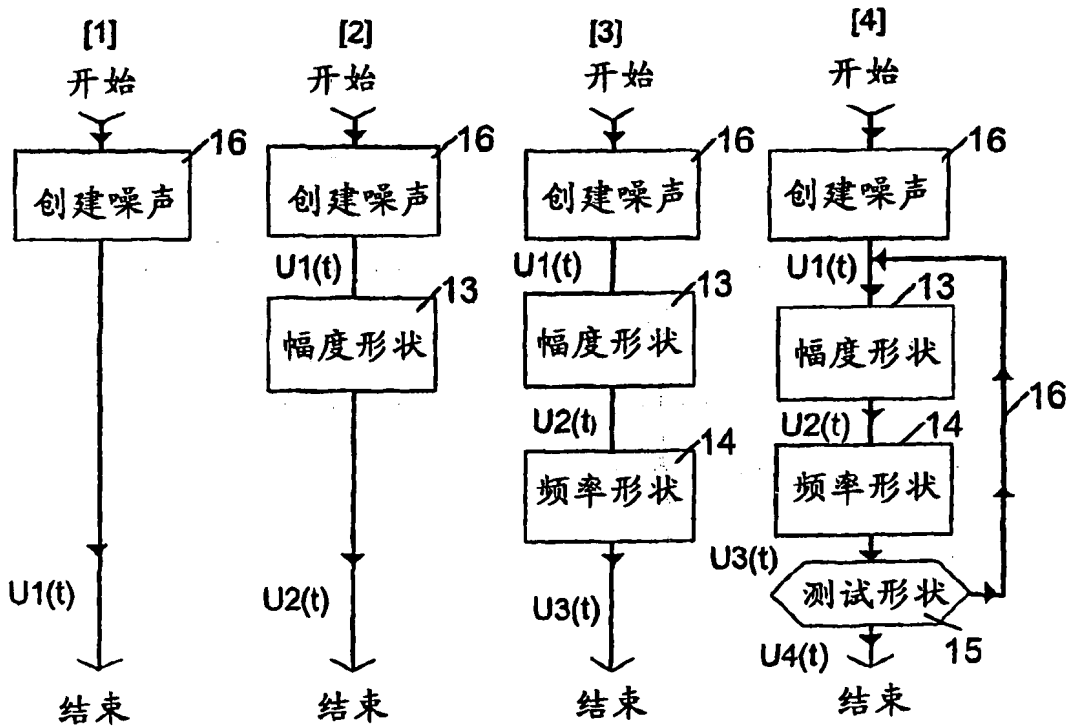


图 4

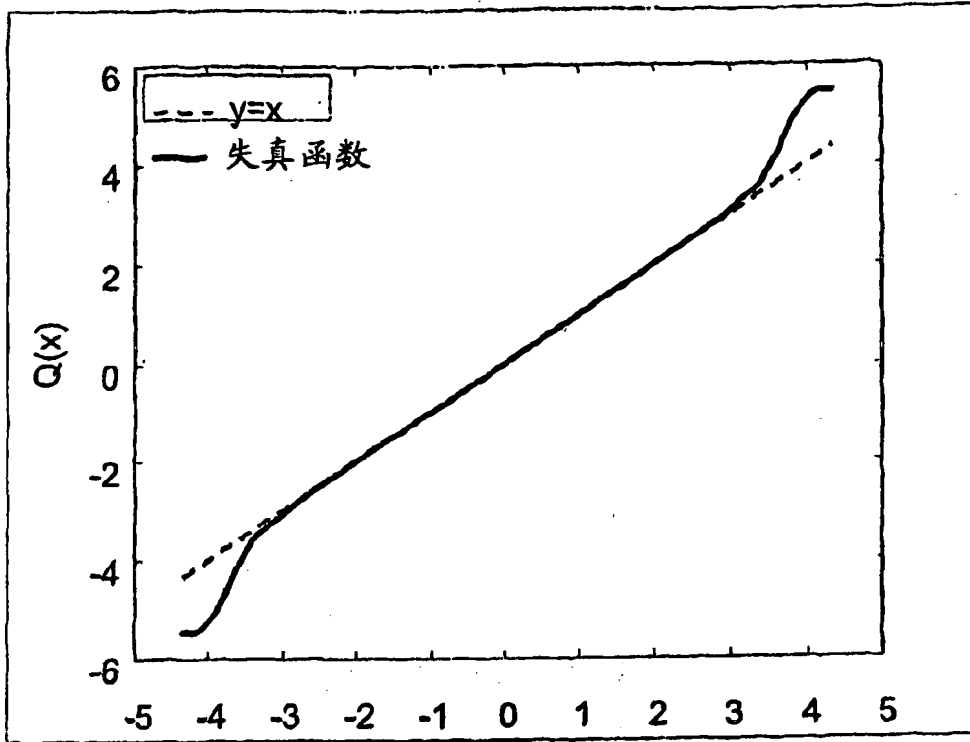


图5

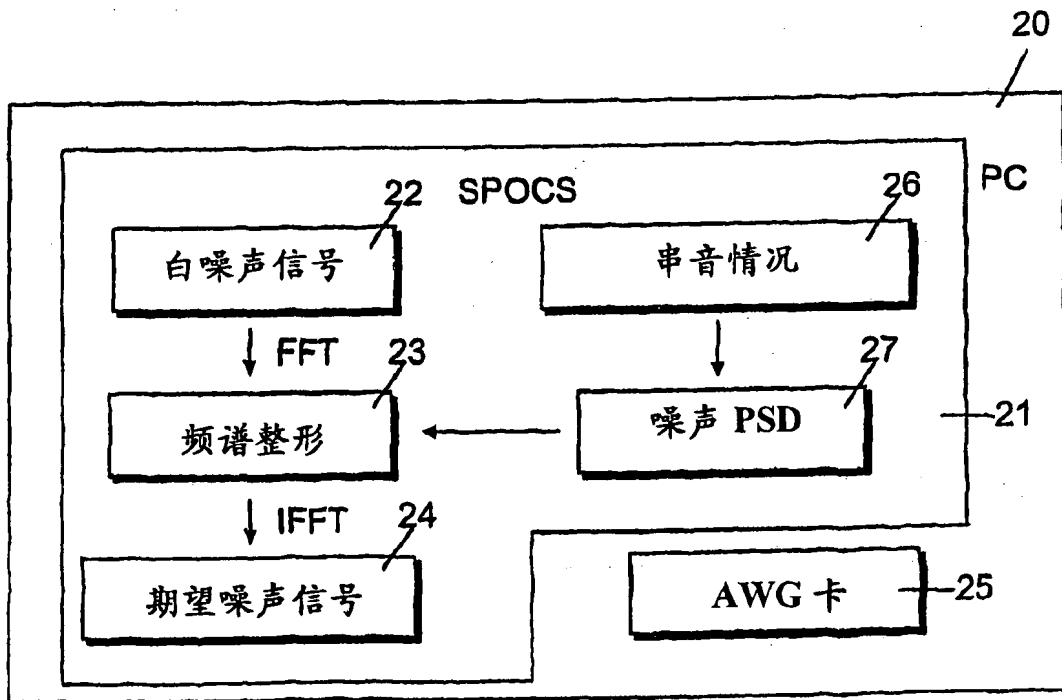


图6

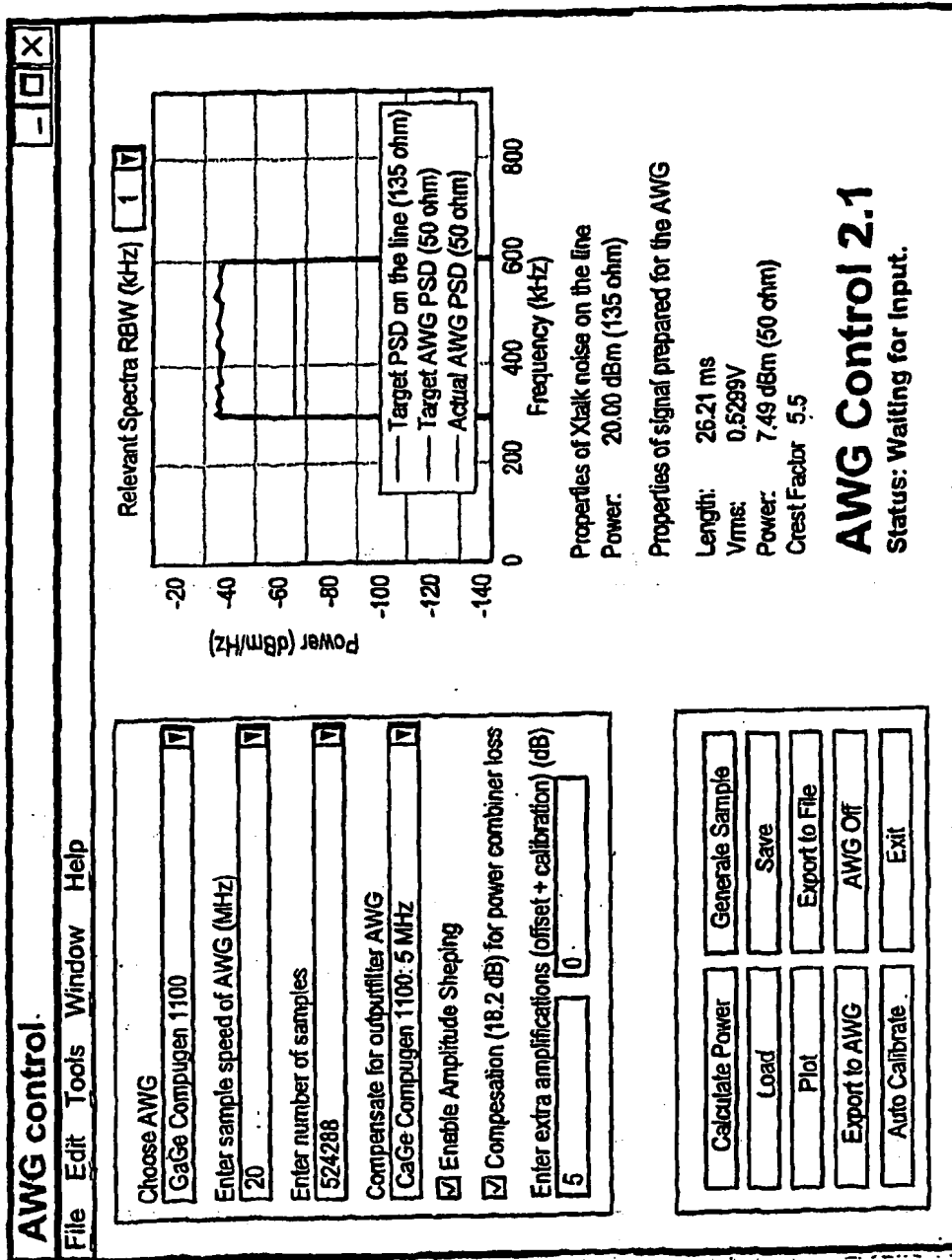


图 7

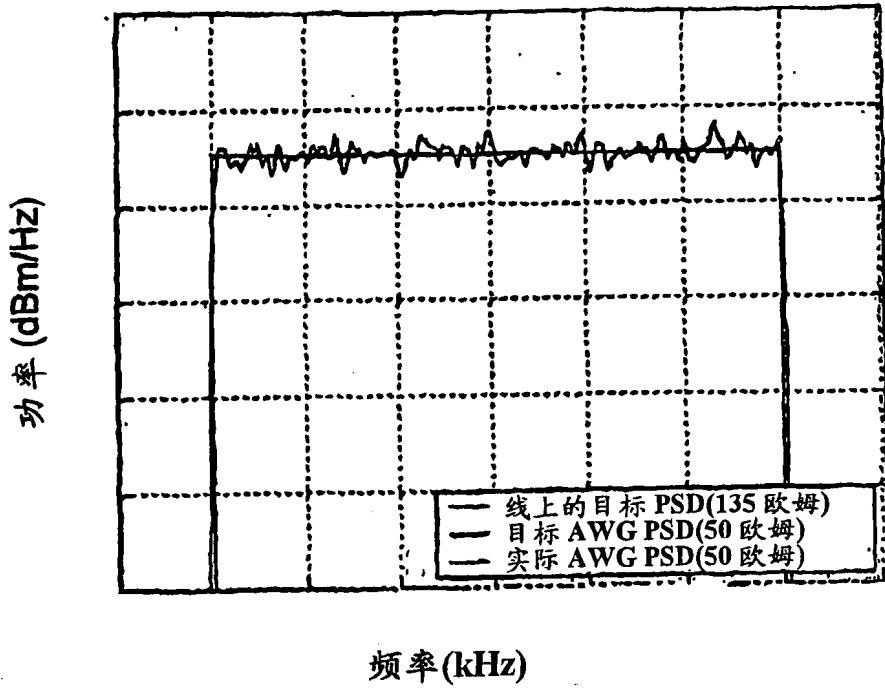


图 8

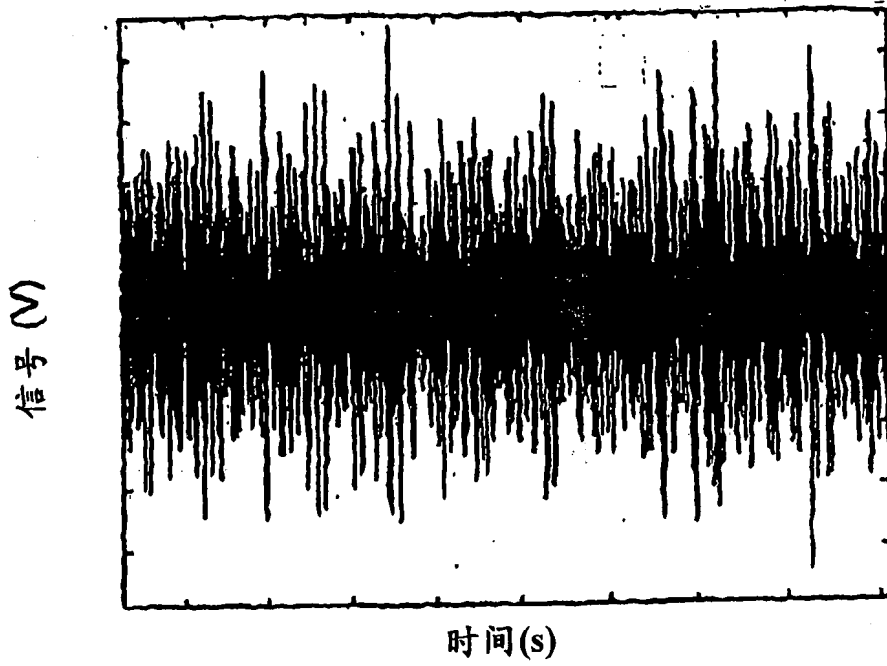


图 9

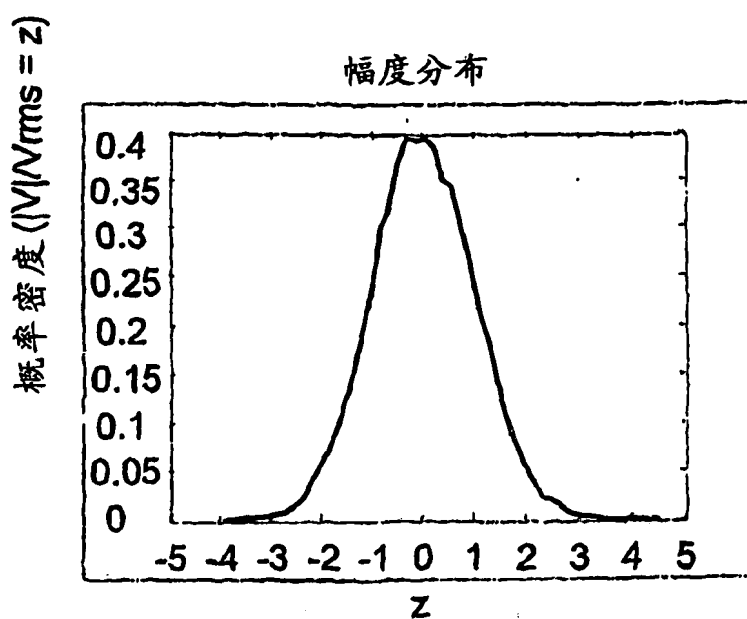


图 10

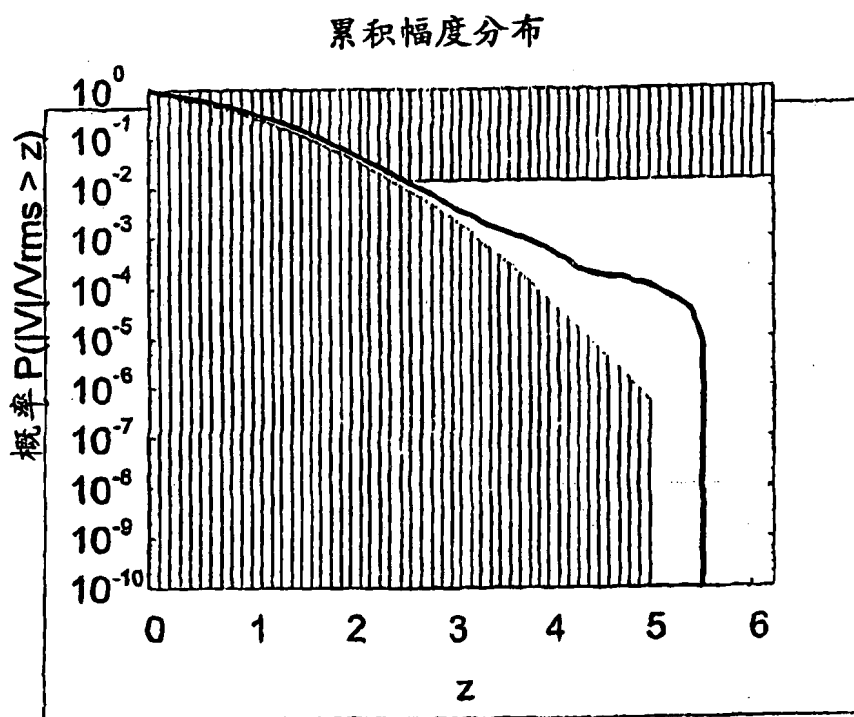


图 11