第36卷 第15期 计 算 机 工 程 2010年8月
Vol.36 No.15 Computer Engineering August 2010
·工程应用技术与实现· 文章编号：1000—3428(2010)15—0233—03 文献标识码：A 中图分类号：TP301.6

高阶QAM信号盲均衡算法及其FPGA实现

吴 迪，霍亚娟，葛临东，王 彬
(解放军信息工程大学信息工程学院，郑州 450002)

摘 要：针对高阶 QAM 信号的盲均衡问题，提出一种将广义多模算法(GMMA)与判决引导(DD)算法相结合的混合型盲均衡算法GMMA-DD。该算法融合 GMMA 算法与 DD 算法的优势，采用均方误差控制 2 类误差信号在总误差信号中所占的比例，从而提高高阶 QAM 信号的盲均衡性能。为便于现场可编程门阵列(FPGA)实现，对 GMMA-DD 算法进行改进，以降低算法复杂度、节省硬件资源。在 FPGA 上采用串行结构，以流水线的方式设计并实现该算法。实验结果表明，该盲均衡器能够较好地完成对高阶 QAM 信号的均衡。

关键词：高阶 QAM 信号；混合型盲均衡算法；现场可编程门阵列实现

Blind Equalization Algorithm of High-order QAM Signal and Its FPGA-based Implementation

WU Di, HUO Ya-juan, GE Lin-dong, WANG Bin
(Institute of Information Engineering, PLA Information Engineering University, Zhengzhou 450002)

【Abstract】 Aiming at blind equalization problem of QAM signal, this paper proposes a hybrid blind equalization algorithm named GMMA-DD by combining Generalized Multi-Modulus Algorithm(GMMA) and Decision-Directed(DD) algorithm. It combines the benefits of GMMA and DD algorithm, and utilizes the mean square error to control the proportion of two kind of error signals in total error signals to improve the performance of blind equalization. An improved algorithm with lower computational complexity and less FPGA resources is presented and implemented to simplify FPGA-based implementation by using serial and pipelining structures. Experimental results show that the algorithm has good performance for high-order QAM signals.

【Key words】 high-order QAM signal; hybrid blind equalization algorithm; FPGA-based implementation

1 概述

在高速数字通信系统中，为了获得更高的信道频带效率，常常采用高阶 QAM 调制方式。由于信号在传输过程中受到通信信道带宽限制和多径传播等影响，会引起码间干扰(ISI)，因此在接收端需要引入均衡技术。盲均衡技术因无需发射端发送训练序列而得到了广泛的应用。现场可编程门阵列(FPGA)的飞速发展为盲均衡技术的硬件实现提供了可能。

针对高阶 QAM 信号，文献[1]提出了广义多模算法(Generalized Multi-Modulus Algorithm, GMMA)，但是其区域的划分及相应模值的选择较为困难，容易造成稳态误差较大。文献[2]首次提出了混合型盲均衡算法，该类算法在用于更新均衡器系数的总误差信号中同时含有基于传输码元不同特性的误差信号。文献[3]将 CMA 算法与判决引导(Decision-Directed, DD)算法相结合，提出了一种自恢复混合盲均衡算法 HY-NCMA，降低了稳态误差，但对于高阶 QAM 信号，稳态误差仍较大。文献[4]提出一种基于半径大小的混合型盲均衡算法 RMDA，根据判决误差的半径选取步长和比例控制参数。但是该算法本质上仍是采用单一模值，不适用于高阶 QAM 信号，而且比例控制参数的选择也需要通过实验确定，降低了算法的灵活性。本文提出一种新的将 GMMA 算法与 DD 算法相结合的混合型盲均衡算法 GMMA-DD。同时，为了在 FPGA 中实现该算法，对 GMMA-DD 算法进行了改进，降低了算法的复杂度。实验结果表明，处理 256QAM 信号时，改进的 GMMA-DD 算法能够较好地恢复传输数据。

2 GMMA-DD算法及其改进算法

2.1 GMMA-DD混合型盲均衡算法

混合型盲均衡算法通过控制不同收敛过程中不同特性的误差信号在总误差信号中所占的比例来突出某一误差信号的作用。

GMMA 是一种适用于高阶 QAM 信号的盲均衡算法，它分别将星座图的实部和虚部分成几个不同的子区域，对分布于不同子区域的信号采用不同的模值。GMMA 的突出优点是使代价函数与星座图更加匹配，对高阶 QAM 信号能够达到更小的稳态误差。DD 算法是一种利用传输码元瞬时特性的盲均衡算法，只有当信号“眼图”张开时才能保证可靠的收敛，否则会造成误判，引起错误传播，实际应用中一般做跟踪处理。

由于误差信号的统计平均 *MSE* 具有较好的抗噪声性能，因此本文采用 *MSE* 控制 GMMA 误差信号和 DD 误差信号的比例。GMMA-DD 算法总的误差公式可以表示为

$$\begin{cases} e_{r,\text{GMMA-DD}}(n)=\lambda e_{r,\text{GMMA}}(n)+(1-\lambda)\cdot e_{r,\text{DD}}(n) \\ e_{i,\text{GMMA-DD}}(n)=\lambda e_{i,\text{GMMA}}(n)+(1-\lambda)\cdot e_{i,\text{DD}}(n) \end{cases} \tag{1}$$

基金项目：河南省基础与前沿研究基金资助项目(082300413205)；解放军信息工程大学信息工程学院基金资助项目(YP08JH31036)

作者简介：吴 迪(1984－)，男，硕士研究生，主研方向：信号与信息处理；霍亚娟，硕士研究生；葛临东，教授、博士生导师；王 彬，讲师

收稿日期：2010-01-22 **E-mail**：fjwudi2006@163.com

其中，

$$\begin{cases} e_{r,\mathrm{GMMA}}(n)=y_r(n)\left(\left|y_r(n)\right|^2-R_{r,m}\right) \\ e_{i,\mathrm{GMMA}}(n)=y_i(n)\left(\left|y_i(n)\right|^2-R_{i,m}\right) \end{cases} \tag{2}$$

$$\begin{cases} e_{r,\mathrm{DD}}(n)=y_r(n)-\hat{a}(n) \\ e_{i,\mathrm{DD}}(n)=y_i(n)-\hat{b}(n) \end{cases} \tag{3}$$

$y_r(n)$、$y_i(n)$为均衡器输出信号的实部和虚部；$\hat{a}(n)$、$\hat{b}(n)$为判决信号的实部和虚部；$R_{r,m}$、$R_{i,m}$表示信源符号实部和虚部对应不同区域的模值；λ为误差控制系数，选取原则为

$$\lambda=\begin{cases} 1 & \text{if } MSE\geqslant d \\ \dfrac{MSE}{d} & \text{if } MSE<d \end{cases} \tag{4}$$

其中，d为星座图中判决区域中的点与对应的判决点的最大距离的平方；λ的初始值为1，在初始收敛阶段，GMMA误差所占的比例较大，随着MSE的减小，λ逐渐趋于0。由于是由DD算法主导，因此可以得到较小的稳态误差。

该算法相应的抽头系数更新公式为

$$\boldsymbol{W}(n+1)=\boldsymbol{W}(n)-\mu e_{\mathrm{GMMA\text{-}DD}}\boldsymbol{X}^*(n) \tag{5}$$

其中，$\boldsymbol{W}(n)$为抽头系数向量；$\boldsymbol{X}(n)$为均衡器输入向量；μ为步长因子。

2.2 GMMA-DD算法的改进

用FPGA实现盲均衡算法既要考虑算法的性能，又要综合考虑算法的复杂度。GMMA-DD算法对于硬件实现来说复杂度较高，为了尽可能节约硬件资源，需要对GMMA-DD算法做一些改进。

在计算GMMA误差时，每得到一个$e_{r,\mathrm{GMMA}}(n)$或$e_{i,\mathrm{GMMA}}(n)$就需要3个乘法器和1个减法器。简化方式如下：

$$\begin{cases} e_{r,\mathrm{GMMA}}(n)=sign\left(y_r(n)\right)\left(y_r^2(n)-R_{r,m}^2\right) \\ e_{i,\mathrm{GMMA}}(n)=sign\left(y_i(n)\right)\left(y_i^2(n)-R_{i,m}^2\right) \end{cases} \tag{6}$$

这样，在计算$e_{r,\mathrm{GMMA}}(n)$或$e_{i,\mathrm{GMMA}}(n)$时就可以节约1个乘法器资源，同时没有改变误差的本质，对算法的收敛性能没有大的影响。

GMMA-DD算法的抽头系数调整方向取决于式(5)中$e_{\mathrm{GMMA\text{-}DD}}\boldsymbol{X}^*(n)$的符号。若只取$e_{\mathrm{GMMA\text{-}DD}}\boldsymbol{X}^*(n)$的符号，则既可以保证调整方向的正确性，又能够降低算法的复杂度。本文对式(5)做如下处理：

$$\boldsymbol{W}(n+1)=\boldsymbol{W}(n)-\mu e_{\mathrm{GMMA\text{-}DD}}\cdot sign\left(\boldsymbol{X}^*(n)\right) \tag{7}$$

此时，展开为I、Q 2路后若采用串行处理，每计算一个$\boldsymbol{W}$将减少4个乘法器资源；采用并行处理时，对于m阶的抽头系数，将减少$4m$个乘法器资源。当步长μ选为2的幂次时，在硬件实现时可以完全由移位操作代替乘法运算。

由于误差控制系数λ是由MSE决定的，因此在均衡过程中需要实时计算MSE的值。本文采用文献[5]中带有遗忘因子η的MSE估计公式实时估计迭代过程中MSE的值：

$$MSE_{n+1}=\eta MSE_n+(1-\eta)\left|y_n-\hat{y}_n\right|^2 \tag{8}$$

其中，$\hat{y}_n$为判决信号；η体现了对数据的记忆性，反映了算法的跟踪能力，本文取$\eta=0.99$。采用这种方法复杂度低，易于控制。

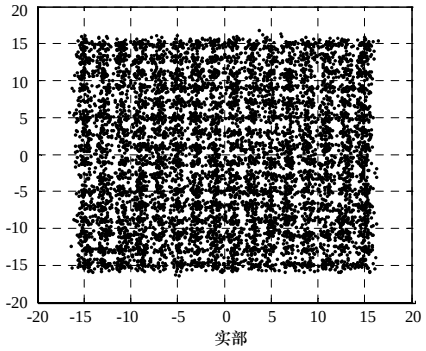
3 算法仿真实验

本文通过计算机仿真验证GMMA-DD及其改进算法均衡的有效性和收敛性能，并与GMMA、HY-NCMA、RMDA做比较。均衡器采用17阶横向抽头结构，中心抽头系数初始化为1，其他抽头系数初始化为0。仿真信道的冲激响应为[-0.005-i0.004, 0.009+i0.03, -0.024-i0.104, -0.218+i0.273, 0.049-i0.074, -0.016+i0.02]，调制方式为256QAM，信噪比为30 dB。对于256QAM信号，其子区域的划分及相应R_m值的选取与文献[1]相同。HY-NCMA的迭代步长取0.000 000 05，RMDA算法的参数选择如表1所示。

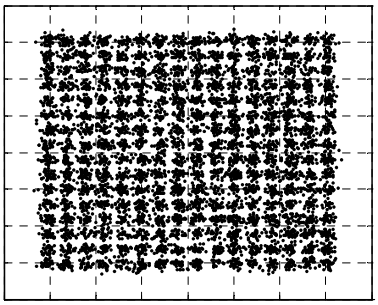
表1 RMDA算法参数

区域	范围	步长	比例参数
1	$e\geqslant 1$	2×10^{-7}	1.0
2	$2/3\leqslant e<1$	5×10^{-8}	1.0
3	$1/3\leqslant e<2/3$	5×10^{-8}	0.7
4	$1/6\leqslant e<1/3$	1×10^{-8}	0.3
5	$e<1/6$	5×10^{-9}	0.0

图1(a)为256QAM信号采用GMMA-DD算法得到的均衡器输出星座图，迭代步长为0.000 000 25。图1(b)为采用改进的GMMA-DD算法得到的均衡器输出星座图，迭代步长为0.000 02。可见，2种算法均能比较正确地恢复出传输信号。



(a)GMMA-DD算法均衡器输出



(b)GMMA-DD改进算法均衡器输出

图1 GMMA-DD算法及其改进算法的均衡器输出

图2给出了GMMA、GMMA-DD、HY-NCMA、RMDA及GMMA-DD改进算法的码间干扰性能曲线。

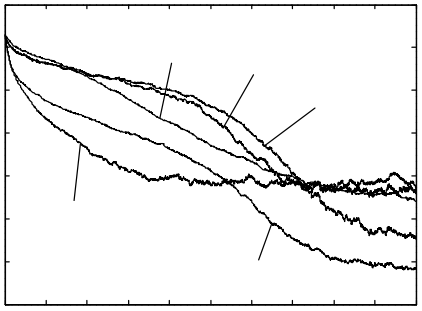


图2 5种算法性能比较

从图中可以看出，RMDA 算法收敛速度快，这是它根据不同区域采用不同的步长所带来的优势，但由于它采用一个模值均衡，因此稳态误差并没有得到改善。HY-NCMA 算法对高阶 QAM 信号的稳态误差与 GMMA 算法相当，而 GMMA-DD 算法的稳态误差大大降低，收敛速度与 GMMA 算法相当。改进后的 GMMA-DD 算法不仅进一步降低了稳态误差，收敛速度也比 GMMA-DD 算法快。

4 盲均衡器的FPGA实现

根据改进的 GMMA-DD 算法，本文在 FPGA 上实现了一个 9 阶的盲均衡器，其系统结构如图 3 所示。由改进的 GMMA-DD 算法可以确定盲均衡器主要由 FIR 滤波模块、GMMA 误差计算模块、判决误差提取模块、总误差控制模块以及系数调整模块组成。

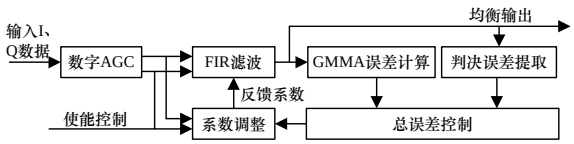


图 3　GMMA-DD 算法的盲均衡器系统结构

对于 QAM 信号，需要设置一个幅度调节系统来保证均衡器输入信号幅度的平稳性。数字 AGC 是一种在输入信号幅值变化很大的情况下，通过调节可控增益放大器的增益，使输出信号幅值基本恒定或仅在较小范围内变化的一种电路。数字 AGC 可以使均衡器的模值相对固定，便于算法的收敛。

4.1 功能模块的实现

系统各模块的实现方式如下：

(1)数字 AGC 模块。数字 AGC 主要由 3 部分组成，分别为功率差值提取、环路滤波和积分器。功率插值提取模块得到当前信号的功率并与额定功率进行比较，得到功率误差，用以调整增益系数；环路滤波的作用是滤除高频分量，得到等效调整因子；积分器是个增益累加器，根据环路滤波器的输出调整增益值，从而自适应地改变当前输出信号的幅度，达到自动调整输出信号功率的目的。参考功率需要按照所给信号的大小及 QAM 星座图模板来确定，在本系统中，2 路信号 I、Q 都为 16 bit，十进制数表示最大值为 32 767。设定 256QAM 星座点最小距离的量化值为 256，由此可得最大幅度的量化值为 15 000，据此可以算出参考功率的量化值为 2 785 280。

(2)FIR 滤波模块。在 FPGA 中，4 个 16 bit 乘法器和 2 个 32 bit 加法器需要 8 个 9 bit 的 DSP block 和 65 个 ALUT 单元。如果不采用复用技术，对于一个 9 阶的滤波器就需要使用 72 个 DSP 单元，占用大量的硬件资源。本文使用一个系统时钟为 80 MHz 的采集卡，码元速率最高为 5 MHz。综合考虑，可以采用串行方式设计一个 9 阶的 FIR 滤波器。收到信号时，需要保存当前以及之前的 8 个码元信号值，存入缓存 $dataI_temp(i)$，$dataQ_temp(i)$, $i=0,1,\cdots,8$，并开始计数。同时，数据使能通知系数调整模块以时钟速率输出调整后的系数送给 FIR 滤波模块。在每个系统时钟下，送入的系数与缓存中的数据作一次复数乘法，并把结果加入输出缓存器中。这样经过 9 个时钟后就可以得到滤波后的结果，给出输出使能。系数调整模块同样有一个计数器用于计数输出系数的个数，经过 9 个时钟后停止输出系数直到下一个数据输入使能的到来。

(3)GMMA 误差计算模块。先根据 FIR 滤波器的输出 $y_r(n)$ 和 $y_i(n)$ 确定其在星座中所属的区域，不同的区域对应不同的 $R_{r,m}^2$、$R_{i,m}^2$。对于确定的星座模板，$R_{r,m}^2$、$R_{i,m}^2$ 是一个确定的 32 bit 数据。同时将输入数据送入平方器，16 bit 数据经平方后也成为了 32 bit 数据。这 2 个操作可以同时在一个时钟下运行。输出的误差结果根据输入数据的符号确定，以实部为例，可以写成如下形式：

$$e_{r,\mathrm{GMMA}}(n)=\begin{cases} y_r^2(n)-R_{r,m}^2 & y_r(n)\geqslant 0 \\ R_{r,m}^2-y_r^2(n) & y_r(n)<0 \end{cases} \tag{9}$$

(4)判决误差提取模块。判决误差提取模块根据最小距离将 FIR 滤波后的数据判决为星座中的点，根据式(3)求出判决误差。判决器实际上是一个多路选择器，根据确定的星座模板，$y_r(n)$、$y_i(n)$ 与不同的阈值比较，输出相应的星座点。该模块与 GMMA 误差计算模块同时进行，共需 2 个时钟周期。

(5)总误差控制模块。总误差控制模块根据式(1)计算，输入值为上述 2 个模块得到的 2 类误差，分 2 个步骤完成：1)估计当前的 *MSE* 值。遗忘因子 η 为浮点型数据，为了保证精度，采用 16 bit 量化时，精度采用 14 bit，即将数据的小数点右移 14 bit，运算结束后再左移 14 bit。输入的判决误差取绝对值后再与遗忘因子相乘。2)计算 λ，并根据得到的 λ 计算总误差。这部分是一个多路选择器，通过比较 *MSE* 与 *d* 的大小来选择合适的 λ。在改进算法中，$MSE(k)/d$ 可以采用移位器实现。得到的误差值送给系数调整模块。

(6)系数调整模块。系数调整模块根据总误差计算模块得到的误差值及输入信号经过延迟缓存的值来确定新的系数。改进算法对信号值取符号，消除了误差值与信号值的乘法操作，系数更新过程只有实数的加减操作，节约了硬件资源。步长因子 μ 是一个很小的数，在本设计中取 2^{-30}，从而可采用移位器来实现，节约了乘法器资源。数据使能到来时，每个时钟输出一个已经计算好的系数值给 FIR 滤波器模块，这样就通过数据使能实现了系数更新模块与 FIR 滤波器模块的同步。

4.2 仿真时序分析

本设计采用硬件描述语言 VHDL 在 Quartus II5.0 编程实现，时序仿真结果见图 4，每完成一个自适应过程需要 16 个时钟周期，符合设计要求。图中的 coef_en 为系数使能，当数据使能 data_en 到来时，系数以时钟速度送出，经过 9 个时钟后停止，实现了系数与 FIR 滤波的同步，与设计要求一致。result_en 为均衡输出使能，误差计算根据当前的均衡结果得到，其中，edd_en 与 emma_en 为 2 类误差的使能，它们共同复用一个时钟周期。

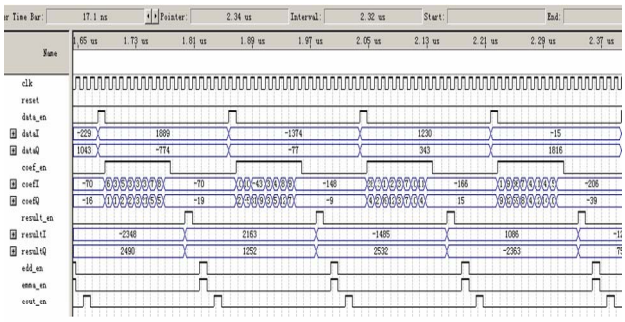


图 4　盲均衡器仿真时序

4.3 盲均衡器的测试与实现结果

本文设计的盲均衡器加载于通用数字信号处理卡上的

（下转第 239 页）