

一种最小二乘/奇异值分解算法

曹新容, 黄联芬, 赵毅峰

(厦门大学信息科学与技术学院, 厦门 361005)

摘要: 针对预失真技术中存在记忆非线性放大器预失真的问题, 分析数字预失真器的结构和常用预失真器的识别算法, 对经典最小二乘/奇异值分解(LS/SVD)算法进行改进, 以较少资源获得较高性能。仿真结果表明, 改进的LS/SVD算法能实现记忆非线性放大器的快速、高效线性化, 提高记忆非线性放大器的性能。

关键词: 数字预失真; Hammerstein 模型; 最小二乘; 奇异值分解

Least Square/Singular Value Decomposition Algorithm

CAO Xin-rong, HUANG Lian-fen, ZHAO Yi-feng

(School of Information Science and Technology, Xiamen University, Xiamen 361005)

【Abstract】 Aiming at the problem of predistortion for memory nonlinear amplifier in predistortion technology, this paper analyzes the structure of digital predistortion and recognition algorithm of common predistortion, improves classical Least Square/Singular Value Decomposition (LS/SVD) algorithm. Improved LS/SVD algorithm can obtain better performance by less resource. Simulation results show the proposed algorithm can realize fast and effective linearization of memory nonlinear amplifier, and improve its performance.

【Key words】 digital predistortion; Hammerstein model; Least Square(LS); Singular Value Decomposition(SVD)

1 概述

目前无线通信迅速发展, 已成为社会和经济发展的主动力。在当今无线通信系统的设计中, 功率放大器越来越重要, 功率放大器是无线传输系统的主要部件, 其内在的非线性是无线通信系统中的主要非线性源。当传输信号通过非线性放大器后信号失真, 影响信号的准确接收, 对相邻用户产生干扰。在 WCDMA、多载波基站等宽带应用中, 功率放大器(Power Amplifier, PA)表现出记忆效应。此时, PA 输出依赖当前输入和历史输入。

数字预失真技术是一种简单易行的克服 PA 非线性的方法, 通过在放大器前构造非线性失真的逆特性, 达到线性化目的。当信号为宽带信号时, 放大器表现出记忆非线性的特性, 当用无记忆的预失真器对记忆放大器进行预失真时, 线性化会迅速恶化。为了适应今后无线通信高速率宽带的发展要求, 记忆预失真器的研究将越来越广泛。

2 数字预失真技术

2.1 数字预失真模型

通常采用的模型包括 Wiener 模型、Hammerstein 模型和非线性抽头延时多项式模型^[1]。Wiener 模型是记忆非线性的一种常用模型, 形式简单、有明确的物理意义, 模拟宽带应用中 PA 的非线性记忆特性。Hammerstein 模型具有 Wiener 模型的逆特性, 被用作 Wiener 放大器的预失真器。非线性抽头延时多项式模型作为 Volterra 级数的简化形式, 包含非线性和记忆性, 是一种形式简单、系数较少的记忆非线性模型, 被用于记忆预失真器的设计。

数字预失真技术要构造放大器的非线性特性的逆。如何准确获取放大器的逆是数字预失真技术的重点。

假设放大器性能满足:

$$y = G(x), \text{ 构造预失真器性能 } y = F(x), \text{ 使得 } F = G^{-1}.$$

这样两者级联, 使 $G(F(x)) = x$, 实现放大器的预失真。

2.2 放大器求逆模型

为了能让预失真系统在硬件上实现运行, 要考虑算法复杂度、硬件可实现性。在整个系统中, 本文采用非线性抽头延时多项式模型描述放大器的逆, 将其作为预失真器的模型。将放大器输出作为求逆表达式的输入, 放大器的输入作为表达式的输出, 直接求解放大器的逆。将系数代入预失真器, 获得满意的预失真性能, 放大器求逆模型如图 1 所示。

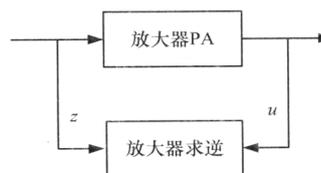


图 1 放大器求逆模型

$$z(n) = \sum_{k=0}^{K-1} \sum_{q=0}^{Q-1} a_{kq} u(n-q)^k$$

当算法收敛时, 采用最小二乘(Least Square, LS)法直接解得系数:

$$\hat{a} = [a_{0,0}, \dots, a_{K-1,0}, a_{0,1}, \dots, a_{K-1,Q-1}]$$

其中, K 为多项式阶数; Q 为记忆深度; 系数 a 的因子个数为 $K \times Q$ 。当 K 和 Q 较大时, 此算法的运算量较大, 硬件资源的占用量也很大。特别在前向通路中, 所有数据经过系数 a 的运算, 增大系统运算量, 降低系统效率。

2.3 Hammerstein 模型

为了降低硬件资源开销、系统复杂度和更新系数数量,

作者简介: 曹新容(1983-), 男, 硕士研究生, 主研方向: 无线通信; 黄联芬, 高级工程师、博士; 赵毅峰, 助教、博士研究生

收稿日期: 2009-02-09 **E-mail:** hlf@xmu.edu.cn

预失真器模型采用 Hammerstein 模型, 如图 2 所示。

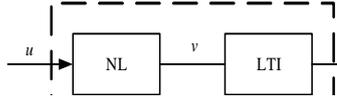


图 2 Hammerstein 模型

Hammerstein 模型相当于一个无记忆非线性(NonLinearity, NL)模块和一个线性时不变模块(Linear Time-invariant, LTI)的级联, 分别求得 NL 和 LTI 的系数。

令 b 和 c 分别作为 NL 和 LTI 的系数:

$$v(n) = \sum_{k=0}^{K-1} b_k [u(n)]^k, \quad z(n) = \sum_{q=0}^{Q-1} c_q v(n-q)$$

文献[2]采用最小二乘/奇异值分解(Less Square/ Singular Value Decomposition, LS/SVD)法求解这 2 组系数。

定义 $a_{kq} = b_k c_q$

采用 LS 算法, 得多项式模型系数 a :

$$\hat{a} = [a_{0,0}, \dots, a_{K-1,0}, a_{0,1}, \dots, a_{K-1,Q-1}]$$

改为矩阵:

$$a = \begin{bmatrix} a_{0,0} & a_{0,1} & \dots & a_{0,Q-1} \\ a_{1,0} & a_{1,1} & \dots & a_{1,Q-1} \\ \vdots & \vdots & & \vdots \\ a_{K-1,0} & a_{K-1,1} & \dots & a_{K-1,Q-1} \end{bmatrix} = bc^T$$

矩阵 a 的秩为 1, 采用 SVD 分别求系数 b 和 c :

$$A = [a_{0,0}, a_{1,0}, \dots, a_{K-1,0}] = bc_0$$

$$B = [a_{0,0}, a_{0,1}, \dots, a_{0,Q-1}]^T = b_0 c^T$$

假设 $c_0 = 1$, 由此可得:

$$b = A;$$

$$c = \frac{B}{b_0}, \quad b_0 = a_{0,0}, \quad \text{满足假设 } c_0 = 1.$$

这样系数因子由原有的 $K \times Q$ 个减少为 $K+Q$ 个, 在系统前向通路的预失真器中, 降低运算量, 提升系统效率。

2.4 LS/SVD 算法的实现与改进

在 LS 算法中, 硬件实现的一个难点是矩阵求逆。一般在硬件上是采用 QR 分解, 即

$$Ua = z, \text{ 令 } U = QR, R \text{ 为上三角矩阵, 而 } Q \text{ 满足 } Q^T Q = 1.$$

由此可得:

$$QRa = z, \quad Ra = Q^{-1}z = Q^T z = z'$$

在确定 z' 和 R 后, 可得系数 a 。

整个过程称为 QRD_RLS, 在硬件利用 CORDIC (Coordinate Rotation Digital Computer) 方便实现。但由于本文 LS 算法中用到的系数因子个数较多, 有 $K \times Q$ 个, 因此需要的 CORDIC 单元也很多, 将占用大量的硬件资源。此时, 要减少需要求解系数的个数, 从而减少资源占用。

观察 LS/SVD 算法, 通过 LS 算法, 得 $K \times Q$ 阶的系数矩阵 a :

$$a = \begin{bmatrix} a_{0,0} & a_{0,1} & \dots & a_{0,Q-1} \\ a_{1,0} & a_{1,1} & \dots & a_{1,Q-1} \\ \vdots & \vdots & & \vdots \\ a_{K-1,0} & a_{K-1,1} & \dots & a_{K-1,Q-1} \end{bmatrix}$$

在进行系数分解时, 对系数矩阵 a 的第 1 行和第 1 列数据进行运算处理。在建立矩阵求系数时, 若只对该行该列数据建立矩阵求解, 则会节省很大资源, 求放大器逆时所需的

系数个数将由 $K \times Q$ 个减至 $K+Q$ 个。

这样在算法收敛时可得:

$$z = Ua$$

以此得系数:

$$a = [a_{0,0}, \dots, a_{K-1,0}, a_{0,1}, a_{0,2}, \dots, a_{0,Q-1}]$$

此时, 系数 a 有 $K+Q-1$ 个因子, 改为矩阵形式时, 只有矩阵的第 1 行和第 1 列有数据, 其他空余部分以 0 表示, 得:

$$a = \begin{bmatrix} a_{0,0} & a_{0,1} & \dots & a_{0,Q-1} \\ a_{1,0} & a_{1,1} & \dots & a_{1,Q-1} \\ \vdots & \vdots & & 0 \\ a_{K-1,0} & a_{K-1,1} & \dots & 0 \end{bmatrix}$$

针对矩阵的第 1 列和第 1 行, 用 LS/SVD 方法分解, 得系数:

$$b = [a_{0,0}, a_{1,0}, \dots, a_{K-1,0}]$$

$$c = \frac{1}{a_{0,0}} [a_{0,0}, a_{0,1}, \dots, a_{0,Q-1}]$$

3 仿真实验结果

在相同条件下, 针对同样的 PA 模型, 输入 3 载波 WCDMA 宽带信号, 验证比较改进的 LS/SVD 算法和经典 LS/SVD 算法的性能比较, 如表 1 所示。采用误差矢量幅度 (Error Vector Magnitude, EVM) 表征带内失真情况, 邻信道功率比 (Adjacent Channel Interference Ratio, ACPR) 表征相邻信道干扰情况。验证 2 种方法对 PA 的非线性克服效果, 比较两者的性能差异。

$$EVM = \sqrt{\frac{E[|e^2|]}{E[|x^2|]}} \times 100\%$$

$$ACPR = 10 \lg \frac{\int_{f_3}^{f_4} s(f) df}{\int_{f_1}^{f_2} s(f) df}$$

其中, $s(f)$ 为功率谱密度; $[f_1, f_2]$ 为传输信道频带; $[f_3, f_4]$ 为相邻信道频带。

表 1 2 种算法的 ACPR 性能比较

	理想输出信号	LS/SVD 输出信号	经典 LS/SVD 算法输出信号	无预失真输出信号
EVM/(%)	0.00	1.15	1.05	21.15
左邻信道 ACPR/dB	-54.25	-51.48	-51.21	-39.66
右邻信道 ACPR/dB	-54.22	-50.70	-51.68	-43.26

2 种算法的 ACPR 性能比较如图 3 所示。

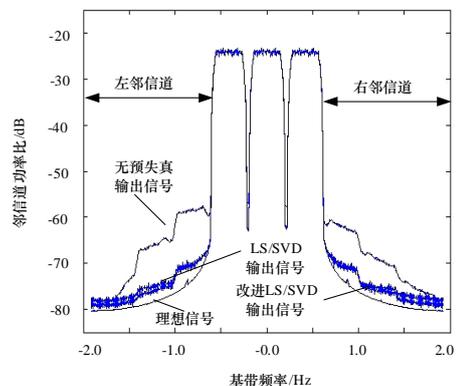


图 3 2 种算法的 ACPR 性能比较

(下转第 282 页)

对消息中的数据进行处理,连接内部数据库进行数据操作。计算订单消息的部门代码如下所示:

```
Set OutputRoot.Properties.MessageSet='OdMsgSet'; //设置消息集
Set OutputRoot.Properties.MessageTyp='DeliveryForm'; //设置输出消息类型
Set OutputRoot.properties.MessageFormat='XML1'; //设置输出消息格式
Set OutputRoot.MRM.OrderID=InputRoot.MRM.OrderID; //将输入消息字段赋值给输出消息对应字段
Set OutputRoot.MRM.OrderDate=InputRoot.MRM.OrderDate; Set OutputRoot.MRM.Price=InputRoot.MRM.Quantity*10; //根据输入消息字段计算出消息字段值
```

(3)根据流程,连接节点,完成整个消息流的设计。

5 系统实时性能测试和分析

在评价系统实时性能时,信息交互层消息传送性能和消息流响应时间为 2 个重要因素^[6]。当有大量消息需要信息交互层进行传输时,消息被分成若干个记录。信息交互层传输时间响应公式为

$$t_{MQ} = t_r \times (D/v) + t_c \times (D/v - 1) \quad (1)$$

其中, t_{MQ} 为信息交互层总传输时间; t_r 为每个记录传输时间; D 为消息长度; v 为每个记录长度; t_c 为启动每个传输的响应时间。消息流响应时间是消息在消息流中传输时间,包括消息在每个节点的处理时间和传递时间,有时还包括访问数据库的时间。通常消息流中节点数越多,处理环节越多,消息处理的响应时间越长。因此,在设计消息流时,高频率的消息路径要尽量短。

以订单部门为例,节点个数与企业服务总线响应时间的关系如图 5 所示,消息大小与企业服务总线响应时间的关系如图 6 所示。

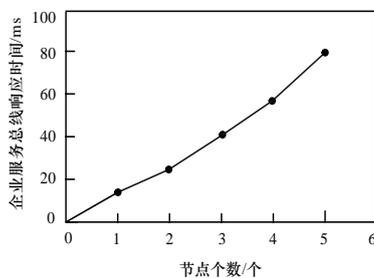


图 5 节点个数与企业服务总线响应时间的关系

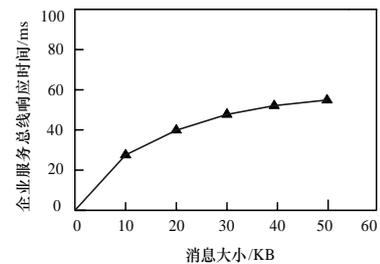


图 6 消息大小与企业服务总线响应时间的关系

在设计消息流时,高频率的消息路径要尽量短,这样消息在消息流中响应时间也越短。从整体上,整合后的系统消息响应时间较短,体现了基于 ESB 的整合系统的实时性。

6 结束语

为了解决现今企业内部系统的信息孤岛、系统整合困难和业务更新困难的问题,提出基于 ESB 的运输管理系统的设计。随着企业的快速发展,基于 ESB 的系统将在更多的企业、更广的领域得到应用。

参考文献

- [1] 陈 靖. 用企业服务总线对 SOA 的改进与应用[D]. 重庆: 重庆大学, 2005.
- [2] 吕新艳. 基于 SOA 的通用物流服务平台的设计与实现[D]. 杭州: 浙江大学, 2006.
- [3] Sixto O J. Getting on Board the Enterprise Service Bus[J]. Communications of the ACM, 2007, 40(4): 15-17.
- [4] Fu Ning, Zhou Xingshe, Wang Kaibo, et al. Distributed Enterprise Service Bus Based on JBI[C]//Proc. of the 3rd International Conference on Grid and Pervasive Computing. Kunming, China: [s. n.], 2008: 292-297.
- [5] 陈宇翔. 精通 WebSphere Message Broker[M]. 北京: 中国水利水电出版社, 2007.
- [6] Abrahim R. A New Generation of Middleware Solutions for a Near-real-time Data Warehousing Architecture[C]//Proc. of IEEE International Conference on Electro/Information Technology. Chicago, IL, USA: [s. n.], 2007: 192-197.

编辑 陆燕菲

(上接第 279 页)

通过比较可知,数字预失真技术对功率放大器的带内失真和临信道干扰有很好抑制克服性能。改进 LS/SVD 算法与经典 LS/SVD 算法在性能上没有较大差别,特别针对记忆性不是特别强的放大器,具有高性能,低运算量的优点。

4 结束语

改进的 LS/SVD 算法忽略非线性抽头延时多项式模型中过去时刻输入值的高次方量,在克服放大器记忆性上存在一些不足,特别是对有较深记忆性(即记忆阶数 Q 较大)的放大器的线性化效果。

今后研究的重点包括如何降低运算量,更好地克服放大器非线性和记忆性,达到更好的线性化。

参考文献

- [1] Kim J, Konstantinou K. Digital Predistortion of Wideband Signals Based on Power Amplifier Model with Memory[J]. Electronics Letters, 2001, 37(23): 1417-1418.
- [2] Bai Erwei. An Optimal Two Stage Identification Algorithm for Hammerstein-wiener Nonlinear Systems[C]//Proceedings of American Control Conference. Philadelphia, USA: [s. n.], 1998: 2756-2760.

编辑 陆燕菲