

---

参赛密码 \_\_\_\_\_  
(由组委会填写)

第十届华为杯全国研究生数学建模竞赛

学 校                                  桂林电子科技大学

---

参赛队号                                  10595005

---

队员姓名	1.	张晓凤
	2.	朱喜燕
	3.	蒋淑洁

---

参赛密码 \_\_\_\_\_  
(由组委会填写)



## 第十届华为杯全国研究生数学建模竞赛

题 目 功率放大器非线性特性及预失真建模

### 摘 要：

功率放大器(PA)用于将已调信号放大到需要的功率，然后通过天线将信号发射出去，它是现代无线通信系统中的关键部件。但 PA 本身具有非线性特性及记忆效应，严重地影响了通信系统的正常传输。本文通过对 PA 进行研究，主要解决了如下问题：

问题一(A)，对无记忆功放的非线性特性描述，首先，本文分别建立 Saleh 模型和 5 阶复系数多项式模型；然后，通过最小二乘法提取模型参数；最后通过计算 NMSE 来定量评价两个模型的精确度。结果表明 5 阶复系数多项式模型的准确度更高，NMSE 可达到-53.16dB，而 Saleh 模型只能达到-33.96dB。

问题一(B)，针对无记忆功放的非线性特性，首先，本文分别建立 5 阶、6 阶无记忆多项式预失真模型对其进行线性化；然后，通过自适应的 RLS 算法提取模型参数；最后，计算得到模型的 NMSE、EVM 及线性化后最大幅度放大倍数，见表 2。结果表明无记忆多项式预失真模型能很好地对无功放系统进行线性化。

问题二(A)：对有记忆功放的非线性特性描述，首先，本文建立不同记忆深度与阶数的记忆多项式模型对其进行描述；然后，利用最小二乘法提取模型参数；最后，计算 NMSE 值见表 3。

问题二(B)：针对有记忆功放的非线性特性，首先，本文建立 3 阶 5 阶记忆多项式预失真模型对其进行线性化；然后，通过自适应的 RLS 算法提取模型参数；最后，计算得到模型的线性化后功放最大幅度放大倍数为 9.45，NMSE = -43.37dB，EVM = 0.68%。结果表明，本文建立的有记忆多项式预失真模型能很好的对有功放系统进行线性化。

问题三：对于信号的功率谱密度，本文利用 matlab 自带的 pmem 函数进行直接求解，结果见图 4-11。由图可得，未加预仿真模型时，功放输出的 ACPR 为-21dB，加入预失真模型后，功放输出的 ACPR 改善为-40dB。

关键字：Saleh 模型，复系数多项式模型，无记忆多项式模型，记忆多项式模型，RLS

## 一、 问题重述

信号的功率放大是电子通信系统的关键功能之一，其实现模块称为功率放大器(PA, Power Amplifier)，简称功放。功放的输出信号相对于输入信号可能产生非线性变形，这将带来无益的干扰信号，影响信息传递和接收，此现象称为非线性失真。传统电路设计上，可通过降低输出功率的方式减轻非线性失真效应。

功放非线性属于有源电子器件的固有特性，研究其机理并采取措施改善，具有重要意义。目前已提出了各种技术来克服改善功放的非线性失真，其中预失真技术是被研究和应用较多的一项新技术，其最新的研究成果已经被用于实际的产品(如无线通信系统等)，但在新算法、实现复杂度、计算速度、效果精度等方面仍有相当的研究价值。

请研究的几个问题：

### 1. 无记忆功放

A. 建立无记忆功放的非线性特性的数学模型，然后用 NMSE 评价所建模型的准确度。

B. 根据建立无记忆功放的非线性特性的数学模型，以线性化原则以及“输出幅度限制”和“功率最大化”约束，建立预失真模型。写出目标误差函数，计算线性化后最大可能的幅度放大倍数，运用评价指标参数 NMSE/EVM 评价预失真补偿的结果。

### 2. 有记忆功放

A. 建立有记忆功放的非线性特性的数学模型，然后用 NMSE 评价所建模型的准确度。

B. 根据建立有记忆功放的非线性特性的数学模型，以线性化原则以及“输出幅度限制”和“功率最大化”约束，以框图的方式建立预失真处理的模型实现示意图(提示：可定义基本实现单元模块和确定其之间关系，组成整体图)，然后计算预失真模型相关参数。运用评价指标参数 NMSE/EVM 评价预失真补偿的计算结果。

### 3. 拓展研究

相邻信道功率比(Adjacent Channel Power Ratio, ACPR)是表示信道的带外失真的参数，衡量由于非线性效应所产生的新频率分量对邻道信号的影响程度。其定义为

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

其中  $s(f)$  为信号的功率谱密度函数， $[f_1, f_2]$  为传输信道， $[f_2, f_3]$  为相邻信道。功率谱密度的计算可通过对信号的自相关函数进行 Fourier 变换计算，也可以通过直接法等计算(假定本题涉及的信号为时间平稳信号)。

如果题 2 所附的数据采样频率  $F_s = 30.72 \times 10^6$  MHz，传输信道按照 20MHz 来算，邻信道也是 20MHz。根据给出的数据，请计算功放预失真补偿前后的功率谱密度，并用图形的方式表示三类信号的功率谱密度(输入信号、无预失真补偿的功率放大器输出信号、采用预失真补偿的功率放大器输出信号)，最后用 ACPR 对结果进行分析评价。

---

## 二、 问题假设

1. 不考虑时间、温度、环境等外界因素的影响。
2. 不考虑功放温度漂移，老化等特性的变化。
3. 不考虑外部信号或电路自身对该功率放大器的干扰。
4. 不考虑当输入信号、负载和元件自身发生变化时，却造成系统的不稳定性的影响。
5. 不考虑功放的特性随时间变化，假设在一定时间内功放的特性都是恒定的。
6. 假设功率放大器的非线性特性是可逆的。

## 三、 符号说明

1.  $x(t)$ : 输入信号
2.  $y(t)$ : 输出信号
3.  $\theta(t)$ : 信号相角
4.  $r(t)$ : 输入信号模值
5.  $a_k$ :  $k$  阶射频复系数
6.  $c_k$ :  $k$  阶基带复系数
7.  $F(\cdot)$ : 幅度特性的复系数多项式
8.  $\Psi(\cdot)$ : 相位特性的复系数多项式
9.  $G(\cdot)$ : 幅度特性函数
10.  $\Phi(\cdot)$ : 相位特性函数
11.  $b$ : 记忆多项式参数
12.  $Y$ : 记忆多项预失真幅值系数
13.  $P$ : 记忆多项预失真相角系数
14.  $K$ : 记忆多项式阶数
15.  $M$ : 记忆多项式深度

其他符号见正文。

## 四、 模型建立与求解

### 4.1 问题一：无记忆功放非线性特性与预失真建模

#### 4.1.1 问题 1 (A)：无记忆功放非线性特性建模

##### 1. 数据分析

功放的非线性既可以从频率也可以从时域来观察，而常用的 AM-AM 和 AM-PM 失真则是对非线性器件比较直观的描述。AM-AM 变换是指输出信号幅度与输入信号幅度存在的非线性关系，AM-PM 变换是指输出信号相位与输入信号幅度的非线性关系。本文利用 MATLAB 绘制问题一中数据的 AM-AM 和 AM-PM 曲线如下图所示：

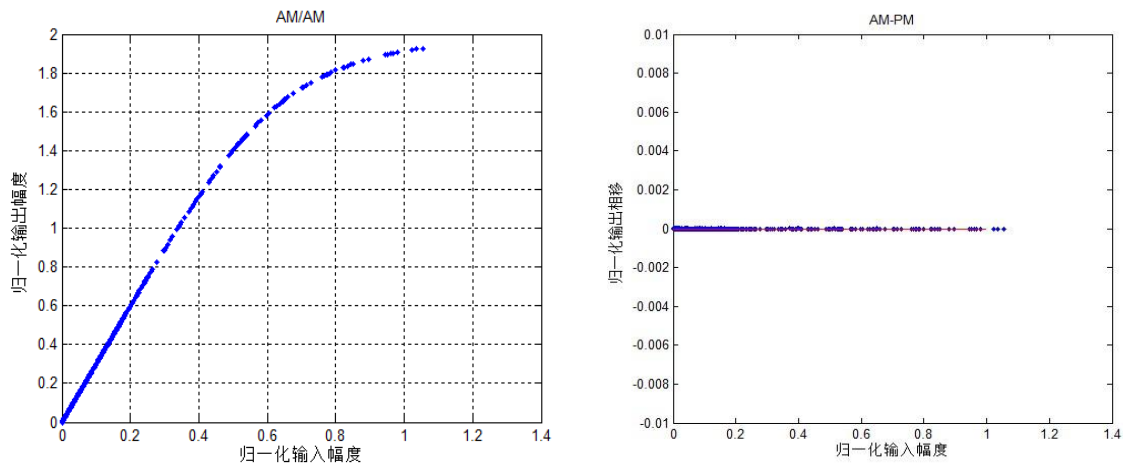


图 4-1 AM—AM 特性和 AM—PM 特性曲线

由图 4-1 可知两条曲线都是光滑的，说明此功放为准无记忆功放，这样的 AM-AM 与 AM-PM 特性曲线称为放大器的静态特性。

##### 2. 无记忆功放模型——Saleh 模型和复系数幂级数模型

无记忆功放是指功放的当前输出信号与历史的输入成分无关而只与当前的输入信号有关，无记忆功放模型被证明为最简单而且比较准确的行为模型，一般适用于窄带信号和温度不变的功放系统中使用。在无记忆模型中，通常采用复系数幂级数模型、Saleh 模型及 Rapp 模型等。

本文分别建立 Saleh 模型及复系数幂级数模型对无记忆功放的非线性特性进行描述，并通过绘制 AM-AM, AM-PM 图对模型进行定性评价，利用 NMSE 指标对模型进行定量评价。

##### (1).Saleh 模型

Saleh 模型是根据对行波管功率放大器(traveling wave tube amplifier, TWTA)的输入输出数据进行统计分析后得到的，TWTA 的 AM-AM 和 AM-PM 失真相对来说都比较明显，并且该模型的参数较少，参数的提取也比较方便，是目前一种常用的无记忆功放模型<sup>[3]</sup>。

假设功放的输入信号为：

$$x(t) = r(t) \cdot \exp[j\theta(t)] \quad (1)$$

则功放的输出信号为：

$$y(t) = A[r(t)] \exp\{j\theta(t) + \Phi[r(t)]\} \quad (2)$$

式中， $A[r(t)]$  和  $\Phi[r(t)]$  分别表示功放的 AM-AM 和 AM-PM 特性函数。为了简单起见，令  $A[r(t)]$  用  $r$  表示输入信号幅度，则模型的特性函数为<sup>[7]</sup>：

$$A(r) = \frac{\alpha_A r}{1 + \beta_A r^2} \quad (3)$$

$$\Phi(r) = \frac{\alpha_\Phi r^2}{1 + \beta_\Phi r^2} \quad (4)$$

式(3)和(4)中， $r$  为输入信号的包络幅度， $\alpha_A$ ， $\beta_A$ ， $\alpha_\Phi$ ， $\beta_\Phi$  为待定参数，本文通过 Yang<sup>[9]</sup>最新提出的智能优化算法——布谷鸟搜索算法对参数进行寻优，得到待定参数  $\alpha_1$ ， $\beta_1$ ， $\alpha_Q$ ， $\beta_Q$  分别为  $\alpha_1=3.1344$ ， $\beta_1=0.5920$ ， $\alpha_Q=0.0100$ ， $\beta_Q=10.0000$ 。

## (2).复系数幂级数模型

由于功放的输入输出都为射频实信号，而功放建模与预失真都在基带进行，基带信号为复信号，需要完成射频实信号到基带复信号的转换。实系数幂级数是不能表征功放的 AM-PM 特性的，因此需要复系数幂级数来对功放进行建模<sup>[4]</sup>。射频中，功放的输入输出特性用  $K$  阶幂级数可以表示为：

$$y_{RF}(t) = \sum_{k=1}^K a_k x_{RF}^k(t) \quad (5)$$

其中， $a_k$  为  $k$  阶射频复系数， $x_{RF}(t)$  为功放的输入射频信号。设  $x_{BB}(t)$  为输入射频信号  $x_{RF}(t)$  的复包络， $y_{BB}(t)$  为功放输出射频信号  $y_{RF}(t)$  的复包络，由带限信号的基带等效有：

$$y_{BB}(t) = \sum_{k=1}^K c_k x_{BB}(t) |x_{BB}(t)|^{k-1} \quad (6)$$

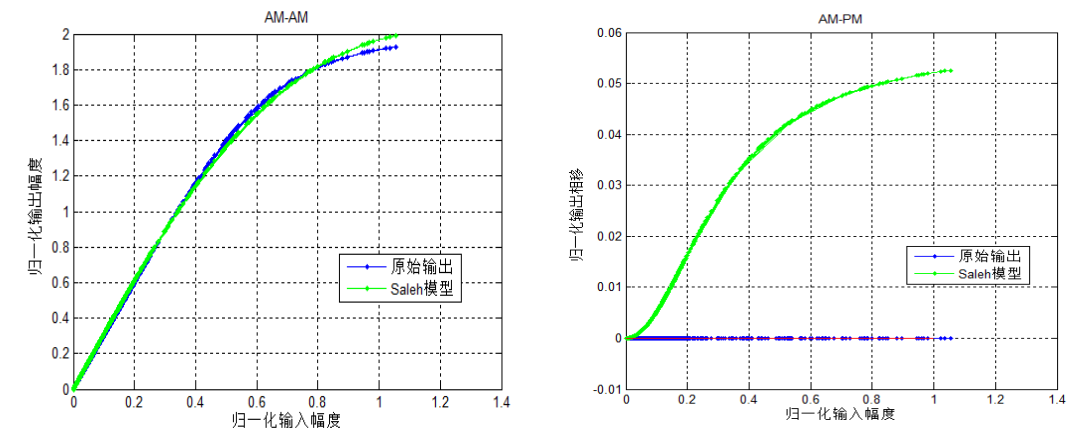
其中， $c_k$  为  $k$  阶基带复系数， $||$  表示取模。幂级数模型是一个无记忆模型，其有记忆模型可对应为记忆多项式，记忆多项式的推导过程与幂级数相似<sup>[4]</sup>。

## 3. 仿真结果与模型评价

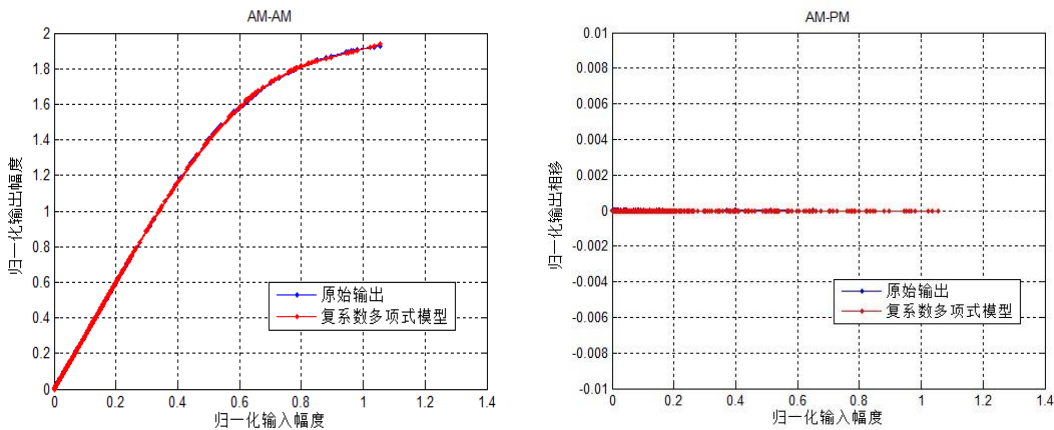
本文建立 5 阶复系数幂级数模型，并利用 matlab 求解可以得到模型参数如下 [ 2.9680- 0.0000i, 0.3099 - 0.0000i, -0.1537 - 0.0000i, -3.4245 - 0.0000i, 2.2082 -

0.0000i ]。

首先，本文利用绘制 Saleh 模型、复系数幂级数模型与原始输出信号的 AM-AM 及 AM-PM 图来对模型的准确性进行定性评价，结果如下图所示：



(a) Saleh 模型 AM-AM 与 AM-PM 曲线图



(b) 复系数多项式模型 AM-AM 与 AM-PM 曲线图

图 4-2 Saleh 模型和复系数多项式功放的 AM-AM 与 AM-PM

由上图的 AM-PM 曲线可以明显看出 Saleh 模型的相角偏移对于复系数多项式模型更大，因此可得出复系数幂级数模型比 Saleh 模型能更准确的描述无记忆功放的非线性特性。

然后，通过计算两个模型的 NMSE 来对模型的准确度进行定量评价。Saleh 模型和复系数幂级数模型的 NMSE 值如下表：

表 1 Saleh 模型和复系数幂级数模型的 NMSE

近似程度评价系数	Saleh 模型	5 阶复系数幂级数模型
NMSE	-33.2900	-53.4415

由上表可知复系数幂级数模型的 NMSE 值相对 Saleh 模型更小，则说明 5 阶复系数幂级数模型能更好的描述无记忆功放的非线性特性。

综上可得：本文建立的 5 阶复系数幂级数模型相对 Saleh 模型能更准确描述无记忆功放的非线性特性。

#### 4.1.2 问题 1(B)：无记忆功放非线性预失真模型

##### 1. 无记忆多项式预失真模型原理

针对无记忆放大器的特性使用上述复系数多项式模型模拟，由于预失真模型应该具备放大器非线性特性的逆特性<sup>[5]</sup>，通过两个模型的叠加使得模型的输入输出特性呈线性。通过复系数多项式预失真模型得到预失真原理图 4-3 如下所示。

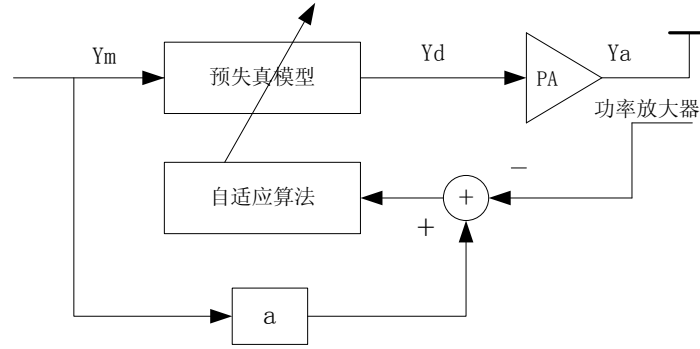


图 4-3 复系数多项式预失真原理图

设输入的正交信号为  $v_m = x_l + jx_Q$ ，通过  $R/P$  转换成极坐标形式  $v_m = r_m e^{j\theta_m}$ ，则相应的预失真器输出信号为：

$$v_d = r_d e^{j\theta_d} = F(r_m) e^{j(\theta_d + \Psi(r_m))} \quad (7)$$

函数  $F(\cdot)$  和  $\Psi(\cdot)$  分别是预失真模型的幅度和相位特性的复系数多项式，其呈现特性为功率放大器特性的逆，其中幅度和相位特性的复系数多项式展开形式为<sup>[5]</sup>

$$F(r_m) = f_1 r_m + f_2 r_m^1 + \cdots + f_k r_m^k = V^T R_f \quad (8)$$

$$\Psi(r_m) = \Psi_0 + \Psi_1 r_m + \Psi_2 r_m^2 + \cdots + \Psi_k r_m^k = P^T R_\Psi$$

在公式 (8) 中， $R_f = [r_m, r_m^2, \cdots, r_m^k]^T$ ， $R_\Psi = [r_m, r_m^2, \cdots, r_m^k]^T$ ， $V = [f_1, f_2, \cdots, f_k]^T$ ， $P = [\Psi_0, \Psi_1, \Psi_2, \cdots, \Psi_k]^T$ 。为了使功率放大器输出呈现线性化，则函数  $F(r_m)$  和  $\Psi(r_m)$  应满足如下公式。

$$G(F(r_m)) = \alpha r_m \quad (9)$$

$$\Psi(r_m) + \Phi(F(r_m)) = 0 \quad (10)$$

在公式(9)和(10)中， $\alpha$  为期望放大器的幅度增益为常数， $G(\cdot)$  和  $\Phi(\cdot)$  分别是功率放大器的幅度和相位特性函数。



## 2. 模型参数提取方法

### (1). LMS 自适应算法

首先给出 LMS 算法的原理推导<sup>[5]</sup>：为了得到最佳的预失真器系数向量  $\mathbf{V}$  和  $\mathbf{P}$ ，根据上述预失真模型原理，得到目标函数使下式(11)和(12)的均方差达到最小。

$$J_1(\mathbf{V}) = E\left(\left(\alpha r_m - G(\mathbf{V}^T \mathbf{R}_f)\right)^2\right) \quad (11)$$

$$J_2(\mathbf{P}) = E\left(\left(\Phi(\mathbf{V}^T \mathbf{R}_f) + \mathbf{P}^T \mathbf{R}_\Psi\right)^2\right) \quad (12)$$

#### 1)系数向量 $\mathbf{V}$ 的迭代

按照 LMS 迭代算法和目标函数写出系数向量  $\mathbf{V}$  的迭代公式为：

$$\mathbf{V}_{k+1} = \mathbf{V}_k - \mu \nabla_{\mathbf{V}} J_1(\mathbf{V}_k) = \mathbf{V}_k + \mu \mathbf{R}_f G(\mathbf{V}^T \mathbf{R}_f) (\alpha r_m - G(\mathbf{V}^T \mathbf{R}_f)) \quad (13)$$

在(13)中， $\mu$  是一个正常数的步长因子，应选择足够小，以保证算法的稳定性和收敛性。 $G$  表示对幅值求导数，由于  $G(F(r_m)) = \alpha r_m$  在工作区域是一条渐增的接近于月牙形的曲线，所以  $G(\mathbf{V}^T \mathbf{R}_f)$  可以近似为 1，则表达式可以写成如下式(14)所示<sup>[5]</sup>。

$$\mathbf{V}_{k+1} = \mathbf{V}_k + \mu \mathbf{R}_f (\alpha r_m - G(\mathbf{V}^T \mathbf{R}_f)) \quad (14)$$

#### 2)系数向量 $\mathbf{P}$ 的迭代

系数向量  $\mathbf{P}$  是依赖于向量  $\mathbf{V}$  的收敛的，最佳  $\mathbf{P}_{opt}$  是使相位预失真与放大器产生的相位偏移正好抵消。利用 LMS 算法，同理有：

$$\mathbf{P}_{k+1} = \mathbf{P}_k + \nu \mathbf{R}_\Psi \left( -\Phi(\mathbf{V}_{opt}^T \mathbf{R}_f) - \mathbf{P}^T \mathbf{R}_\Psi \right) \quad (15)$$

在公式(15)中， $\nu$  也是一个正常数的步长因子，只要选择足够小  $\nu$ ，就可以保证 LMS 算法的稳定性和收敛性。

### (2). 基于 RLS 自适应算法

虽然 LMS 算法能很好地改善功放的线性化，但 LMS 算法的收敛速度很慢。为了快速收敛，可以考虑采用 RLS 算法(递归最小二乘算法)，其中  $\mathbf{p}(\cdot)$ ， $\mathbf{f}(\cdot)$  分别与  $\Psi(\cdot)$ ， $\mathbf{F}(\cdot)$  等价。为了得到最佳的预失真器系数向量  $\mathbf{V}$  和  $\mathbf{P}$ ，要使下面两式的均方误差函数最小。

$$\varepsilon_f(n) = \sum_{i=1}^n \lambda_f^{n-i} |e_f(i)|^2 \quad (16)$$

$$\varepsilon_{\psi}(n) = \sum_{i=1}^n \lambda_{\psi}^{n-i} |e_{\psi}(i)|^2 \quad (17)$$

其中均方误差函数  $\varepsilon_f(n)$ ,  $\varepsilon_{\psi}(n)$  分别是幅度和相位预失真器的代价函数<sup>[8]</sup>, 常数  $\lambda_f$ ,  $\lambda_{\psi}$  称为遗忘因子, 区间为(0,1]。我们定义  $e_f(i)$ ,  $e_{\psi}(i)$  如下:

$$e_f(i) = r_{pd}(i) - V^T(n) R_{f,o}(i) \quad (18)$$

$$e_{\psi}(i) = \theta_d(i) + P^T(n) R_{\psi,o}(i) \quad (19)$$

在公式(18)中,  $r_{pd}$  是预失真器输出信号幅度,  $R_{f,o}(i) = [r_{og}, r_{og}^2, r_{og}^3, \dots, r_{og}^L]^T$ ,  $r_{og} = \frac{r_o}{\alpha}$ ,  $r_o$  是放大器输出信号的幅度。在公式(19)中,  $\theta_d = \theta_o - \psi_{pd}$ ,  $\theta_o$  是放大器输出信号的相位,  $\psi_{pd}$  是输入信号相位和预失真相位之和,  $R_{\psi,o} = [1, r_{og}, r_{og}^2, r_{og}^3, \dots, r_{og}^k]^T$ 。将公式(18), (19)分别代入(16), (17)得到下式:

$$\varepsilon_f(n) = \sum_{i=1}^n \lambda_f^{n-i} (r_{pd}^2(i) - 2r_{pd}(i)V^T(n)R_{f,o}(i) + V^T(n)R_{f,o}(i)R_{f,o}^T(i)V(n)) \quad (20)$$

$$\varepsilon_{\psi}(n) = \sum_{i=1}^n \lambda_{\psi}^{n-i} (\theta_d^2(i) + 2\theta_d(i)P^T(n)R_{\psi,o}(i) + P^T(n)R_{\psi,o}(i)R_{\psi,o}^T(i)P(n)) \quad (21)$$

对公式(19)关于  $V$  求导, 公式(20)关于  $P$  求导得:

$$\nabla_V \varepsilon_f(n) = \sum_{i=1}^n \lambda_f^{n-i} (2R_{f,o}(i)R_{f,o}^T(i)V(n) - 2R_{f,o}(i)r_{pd}(i)) \quad (22)$$

$$\nabla_P \varepsilon_{\psi}(n) = \sum_{i=1}^n \lambda_{\psi}^{n-i} (2R_{\psi,o}(i)R_{\psi,o}^T(i)P(n) + 2R_{\psi,o}(i)\theta_d(i)) \quad (23)$$

要达到最佳系数  $V$ , 就要使  $\nabla_V \varepsilon_f(n) = 0$ , 从公式(22)可以得到下式:

$$[\sum_{i=1}^n \lambda_f^{n-i} R_{f,o}(i)R_{f,o}^T(i)]V(n) = \sum_{i=1}^n \lambda_f^{n-i} R_{f,o}(i)r_{pd}(i) \quad (24)$$

令  $Q_f(n) = \sum_{i=1}^n \lambda_f^{n-i} R_{f,o}(i)R_{f,o}^T(i)$ ,  $Z(n) = \sum_{i=1}^n \lambda_f^{n-i} R_{f,o}(i)r_{pd}(i)$ , 则公式(3-34)可写为:

$$Q_f(n)V(n) = Z(n) \quad (25)$$

对公式(24)利用 RLS 算法, 可以得到下面的更新公式:

$$K_f(n) = \frac{\lambda_f^{-1} Q_f(n-1) R_{f,o}(n)}{1 + \lambda_f^{-1} R_{f,o}^T(n) Q_f(n-1) R_{f,o}(n)} \quad (26)$$

$$\varepsilon_f(n) = r_{pd}(n) - V^T(n-1)R_{f,o}(n) \quad (27)$$

$$V(n) = V(n-1) + k_f(n)\varepsilon_f(n) \quad (28)$$

$$Q_f(n) = \lambda_f^{-1}Q_f(n-1) - \lambda_f^{-1}k_f(n)R_{f,o}^T(n)Q_f(n-1) \quad (29)$$

其中 $Q_f(0) = \delta_f^{-1}I$ ， $\delta_f$ 是一个正常数， $V(0)$ 是任一非零向量。同理可以推导出 $P$ 的更新公式：

$$K_\psi(n) = \frac{\lambda_\psi^{-1}Q_\psi(n-1)R_{\psi,o}(n)}{1 + \lambda_\psi^{-1}R_{\psi,o}^T(n)Q_\psi(n-1)R_{\psi,o}(n)} \quad (30)$$

$$\varepsilon_\psi(n) = \theta_d(n) + P^T(n-1)R_{\psi,o}(n) \quad (31)$$

$$P(n) = P(n-1) + k_\psi(n)\varepsilon_\psi(n) \quad (32)$$

$$Q_\psi(n) = \lambda_\psi^{-1}Q_\psi(n-1) - \lambda_\psi^{-1}k_\psi(n)R_{\psi,o}^T(n)Q_\psi(n-1) \quad (33)$$

其中 $Q_\psi(0) = \delta_\psi^{-1}I$ ， $\delta_\psi$ 是一个正的常数， $P(0)$ 是任一非零向量。

### (3). 约束条件分析

AM-AM 曲线如图 4-4 中的曲线 II，当输入的幅值达到某一定值时输出与输入的幅度呈非线性关系，即输入增加时，输出不再呈线性增加而是出现功率饱和。

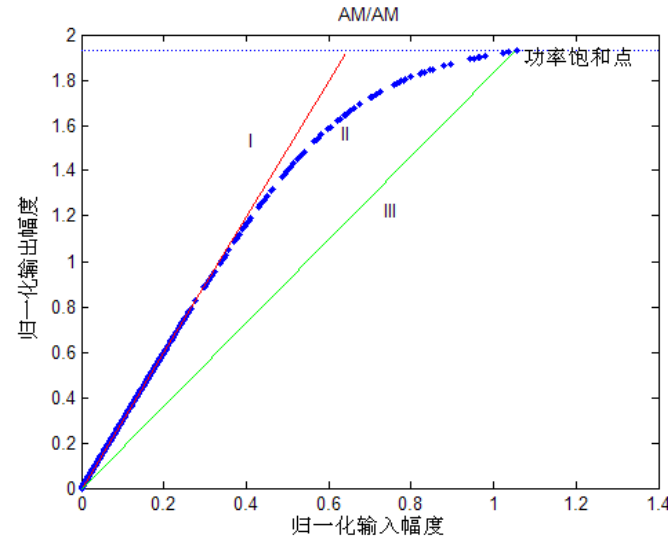


图 4-4 AM-AM 曲线

预失真输入输出幅度的确定取决于预失真器增益(或系统增益)的选择，有两种增益选择方式：第一种目标是实现增益最大(4-4 中的曲线 II)，第二种是功率最大(图 4-4 中的曲线 III)，而增益较低。这里所说的增益是指预失真器的增益或者指插入预失真器之后系统的总增益，而功放的增益不会变化。根据题目要求，本文选择第二种方式。在使功率最大化的情况下尽量提高预失真模型的输出幅度。

### 3. 仿真结果及评价

本文分别建立 5 阶无记忆多项式预失真模型和 6 阶无记忆多项式预失真模型对功放的非线性进行校正。利用 matlab 求解得到预失真模型的系数分别为：

$$V_5 = 0.6542 + 9.7829e-15i$$

$$-0.4830 - 1.3927e-13i$$

$$2.0479 + 6.4426e-13i$$

$$-3.3435 - 1.1073e-12i$$

$$1.9839 + 6.2749e-13i$$

$$V_6 = 0.5761 + 3.3258e-14i$$

$$0.73805 - 6.1814e-13i$$

$$-4.3059 + 3.7396e-12i$$

$$10.8596 - 9.5350e-12i$$

$$-12.2345 + 1.1006e-11i$$

$$5.2311 - 4.6459e-12i$$

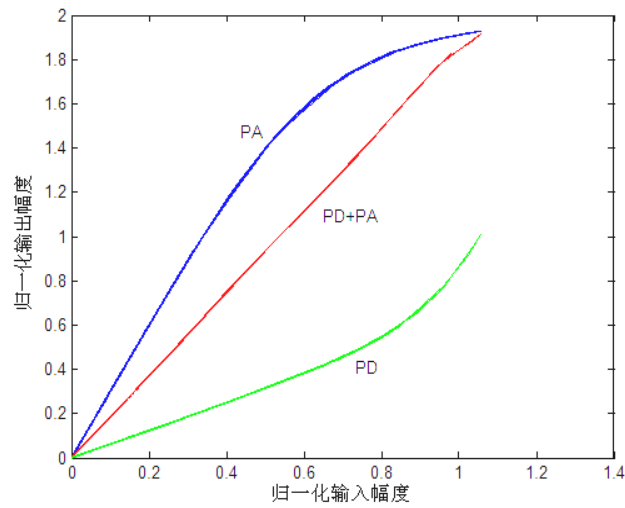


图 4-5 有、无预失真时 AM/PM 特性曲线 (K=5)

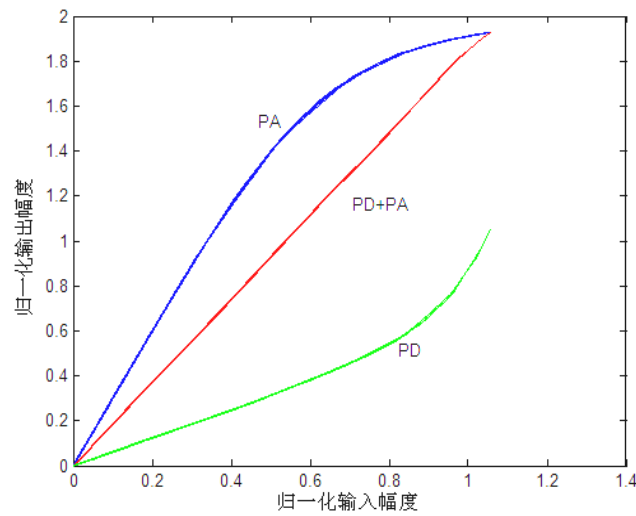


图 4-6 有、无预失真时 AM/PM 特性曲线 (K=6)

图 4-5 和 4-6 中 PA 曲线为无预失真下功放输入、输出信号的 AM/AM 特性曲线；

PD 为基带信号通过预失真器扰动后的预失真信号的 AM/AM 特性曲线即功放的逆特性；PA+ PD 曲线代表了系统在记忆多项式预失真器的作用下得到的预失真后的 AM/AM 特性曲线。由图可知预失真器对功率放大器的静态非线性失真和记忆效应进行了补偿，使 AM/AM 特性曲线比较接近于清晰的直线，非线性失真得到明显的改善。

表 2 K 取不同值时的各指标参数取值

K	EVM(%)	NMSE(dB)	G1	MaxOut
5	0. 58	-44.7050	1.858	1.9123
6	0.34	-49.201632	1.852	1.9271

表 2 中给出了多项式取不同阶数时预失真模型的 EVM 和 NMSE 参数值。其中，EVM 用于衡量整体模型对信号的幅度失真程度，NMSE 反映了模型与物理实际模块的接近程度，EVM 和 NMSE 的值越小越好。其中 G1 为线性化后最大可能的幅度放大倍数，MaxOut 为线性化后输出的最大幅值。

## 4.2 问题二：有记忆功放的非线性特性及预失真建模

### 4.2.1 问题 2(A) 有记忆功放的非线性特性建模

#### 1. 记忆效应

我们实际应用的功放电路由于偏置电路中电抗性元件(电容、电感)的存在，以及周围环境、温度、器件老化等因素，使得功放在当前时刻的输出不仅与此时的瞬态输入有关，还与历史时刻的输入有关，这就是功率放大器的记忆特性。

一般情况，当输入窄带信号时，功放的记忆效应可以忽略不计，但随着宽带通信系统的不断发展，功放的记忆特性表现越来越明显，严重影响了系统性能，所以在实际设计时必须考虑。根据产生的原因不同，可以将其分为热记忆效应和电记忆效应。放大器的记忆效应表现为<sup>[3]</sup>：

- (1)当前输出不仅与当前输入有关，还与过去输入有关；
- (2)功放的 AM-AM 与 AM-PM 特性曲线不仅与输入信号的幅值有关，还与输入信号包络的频率有关；
- (3)双音测试时，IM3 不对称；
- (4)连续谱信号测试时，ACPR 不对称。

#### 2. 数据分析

利用 MATLAB 对问题二中的数据绘制 AM-AM 与 AM-PM 曲线，如图 4-7 所示。由图可以看出 AM-AM 与 AM-PM 不再是光滑的曲线，而是而是一簇曲线，这样的 AM-AM 与 AM-PM 特性曲线称为放大器的动态特性，说明此功放记忆效应的存在<sup>[4]</sup>。

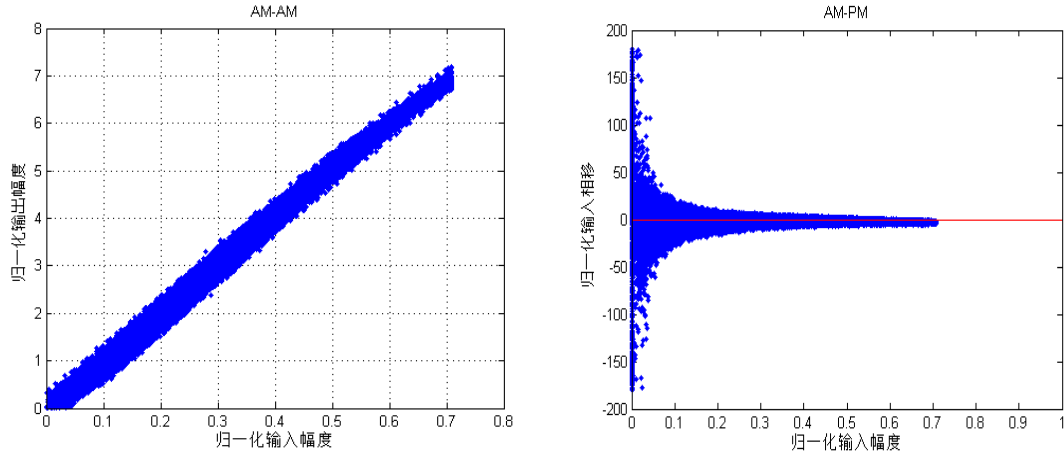


图 4-7 记忆非线性功放的 AM-AM 与 AM-PM

### 3. 有记忆的功放模型——记忆多项式模型

有记忆模型无论是在模型的结构方面还是在参数的辨识方法上面，都比无记忆模型复杂的多。目前比较常用的有记忆功放模型有：Volterra 模型、Hammerstein 模型、Wiener-Hammerstein 模型、Wiener 模型、记忆多项式模型等<sup>[3]</sup>。

其中记忆多项式模型实质上是 Volterra 模型的简化版本，它只考虑了 Volterra 核的对角项，与普通 Volterra 级数相比其参数有明显的减少，可以看成是 Volterra 级数复杂度和记忆性非线性程度的一种折中，便于实现、应用广泛，可以模拟功放的不平衡和动态的 AM-AM 和 AM-PM 特性。因此本文利用采用记忆多项式模型对有记忆的功放特性进行建模，其表达式如下所示：

$$y(n) = \sum_{k=1}^K \sum_{m=0}^M \alpha_{km} x(n-m) |x(n-m)|^{k-1} \quad (34)$$

式中， $K$ 、 $M$  分别表示多项式阶数和记忆深度。通过  $M$  来控制方程的记忆深度，假设  $\Phi_k(x) = |x(n-m)|^k x(n-m)$  将上述表达式转化为<sup>[3]</sup>：

$$y_{RF}(t) = \sum_{k=1}^K a_k \Phi_k(x(n-m)) \quad (35)$$

给出 PA 的  $x(t)$  输入和 PA 的  $y(t)$  输入，求取参数  $a_k$ 。

假设  $X = [x_1(x), x_2(x), \dots, x_n(x)]$  和  $Y = [y_1(x), y_2(x), \dots, y_n(x)]$ ，为简化模型将  $\Phi_k(x) = [\Phi_k(x(1)), \Phi_k(x(2)), \dots, \Phi_k(x(n-m))]$  和  $\Phi = [\Phi_1(x), \Phi_2(x), \dots, \Phi_{n-m}(x)]$ 。然后对  $n-m$  的矩阵扩展  $\Phi = [\Phi_1(x), \Phi_2(x), \dots, \Phi_{n-l}(x); \Phi_{n-l+1}(x), \dots, \Phi_n(x)]$  对应  $Y = [y_1(x), y_2(x), \dots, y_n(x)]$  得出<sup>[6]</sup>：

$$Y = \Phi b \quad (36)$$

$$b = (\Phi^H \Phi)^{-1} \Phi^H Y \quad (37)$$

#### 4. 实验仿真与模型评价

本文设定  $K=3,5,7$  和  $M=2,3,4$ , 利用最小二乘法对 9 种组合情况下的记忆多项式参数  $b$  进行提取, 结果见附录 A。

根据上述求解的多项式系数, 可以得到不同情下的记忆多项式模型, 并可绘制出不同模型下的 AM-AM、AM-PM 特性曲线图。此处本文以阶数为 5、记忆深度为 4 时记忆多项式模型为例, 绘制其 AM-AM、AM-PM 特性曲线, 结果如下:

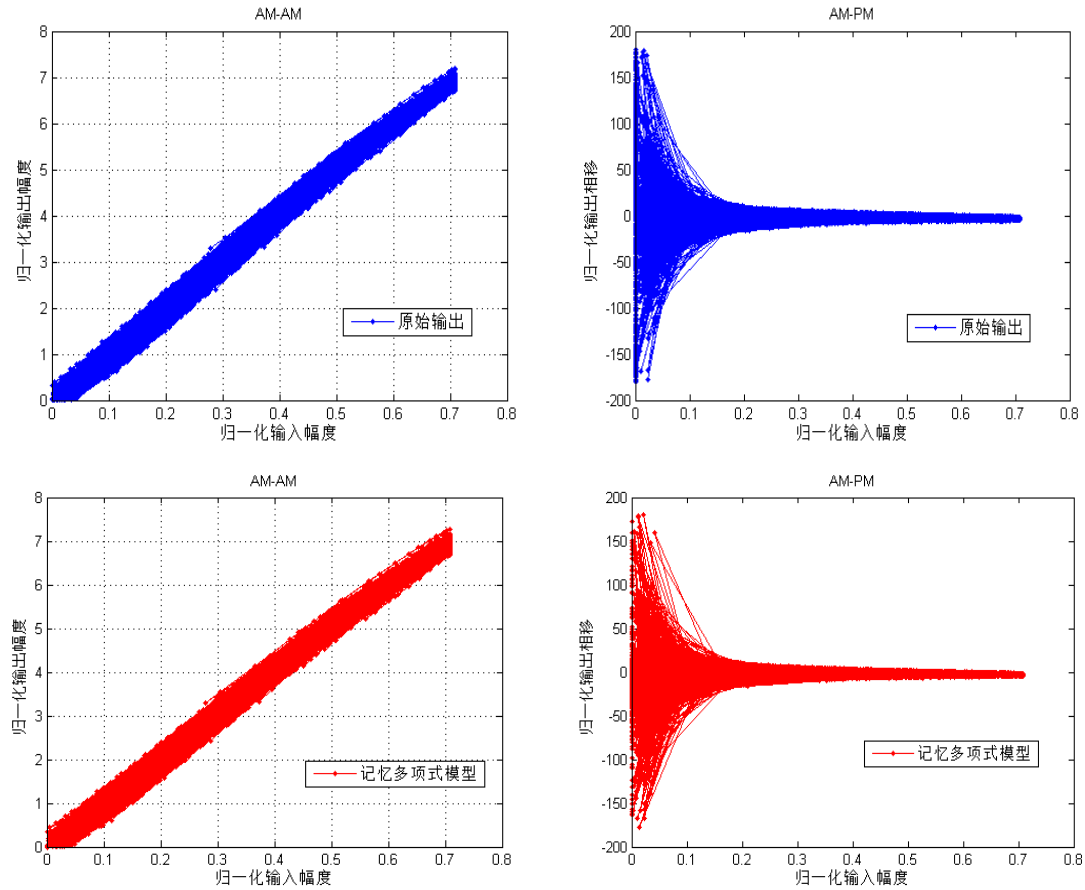


图 4-8  $K=5$ ,  $M=4$  时记忆多项式模型的 AM-AM 与 AM-PM 曲线

由上图可直观看出记忆多项式模型输出值与原始输出具有很高的相似度, 即表明本文求得的记忆多项式模型能较好的对有记忆功率放大器的非线性特性进行描述。

通过 NMSE 指标定量评价当多项式阶数为  $K=3$ ,  $K=5$ ,  $K=7$  和记忆深度为  $M=2$ ,  $M=3$ ,  $M=4$  时的记忆多项式模型对有记忆功率放大器的非线性特性的描述程度。所得结果如下表 3 所示:

表 3 记忆多项式模型的 NMSE 参数

阶数 深度	记忆多项式模型		
	M=2	M=3	M=4
K=3	-39.7042	-42.6899	-43,1104
K=5	-40.4848	-44.5408	-45,2329
K=7	-40.5030	-44.6042	-45,3014

由上表可知,随着多项式阶数和记忆深度的增加模型精确度越高,反映出的 NMSE 越小,但随着记忆多项式模型的阶数增加,模型求解越复杂。实际中可根据不同需要适当的多项式阶数和记忆深度建立记忆多项式模型。

#### 4.2.2 问题 2(B): 有记忆功放预失真处理模型

##### 1. 预失真处理模型——记忆多项式模型

前面曾经介绍过模拟记忆型功放的记忆多项式模型,由于它能够比较准确地描述记忆型功放的非线性特性,而且其计算复杂度与 Volterra 模型相比又降低了很多,因此记忆多项式模型是描述记忆型功放的一种有效和常用的模型。现将其模型公式重写如下:

$$y(n) = \sum_{k=1}^K \sum_{m=0}^M \alpha_{km} x(n-m) |x(n-m)|^{k-1} \quad (38)$$

为了达到线性化效果,通常理想的预失真函数与功放的非线性函数要成互逆关系,因此预失真函数的形式与描述功放非线性函数的形式类似,可用另一个同类型的非线性函数来表示。而对于用记忆多项式模型描述的功放,相应地,数字预失真器也可采用另一个记忆多项式来描述<sup>[8]</sup>。

$$z(n) = \sum_{k=1}^K \sum_{m=0}^M h_{kq} x(n-m) |x(n-m)|^{k-1} \quad (39)$$

##### 2. 预失真记忆多项式模型实现示意图

本文根据记忆多项式建立预失真处理的模型实现示意图如下:

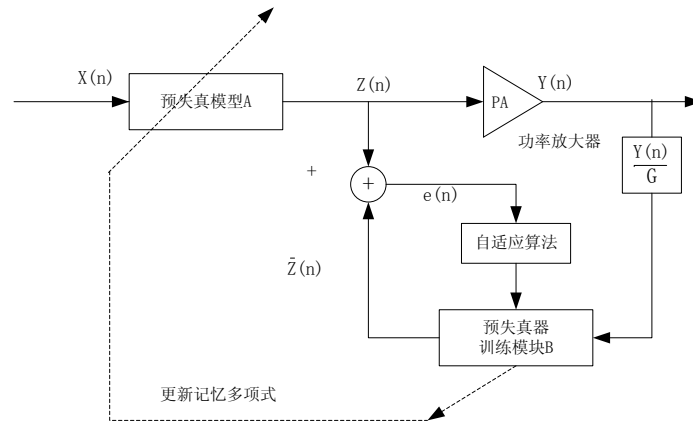


图 4-9 预失真处理的模型实现示意图

此模型也称为间接学习结构数字预失真器。在图 4-9 的反馈路径中,“预失真器训练模块 B”用于预失真器参数的识别,它的输入是  $y(n)/G$ ,  $G$  是功放的线性增益,输出是  $\hat{z}(n)$ 。实际的“预失真器 A”是反馈路径的精确复制,“预失真器 A”的输入是  $x(n)$ ,输出是  $z(n)$ 。理想地,我们希望  $y(n) = Gx(n)$ ,这使得  $z(n) = \hat{z}(n)$ 。



$\hat{y}(n)$ 从而使误差项  $e(n)=0$ 。假设给定了  $y(n)$ 和  $z(n)$ ，接下来的主要任务和工作是在满足“输出幅度限制”和“功率最大化”约束的条件下找到“预失真器训练模块 B”的参数，这样就能够得到“预失真器 A”。这里需要假设功放的非线性特性是可逆的，以使  $e(n)=0$  时相应地有  $y(n)=Gx(n)$ 。

### 3. 参数提取

参数提取方法通常有 LMS 及 RLS，RLS 比 LMS 算法收敛速度快得多，能够达到全局最优，因此本文采用 RLS 算法对参数进行提取。

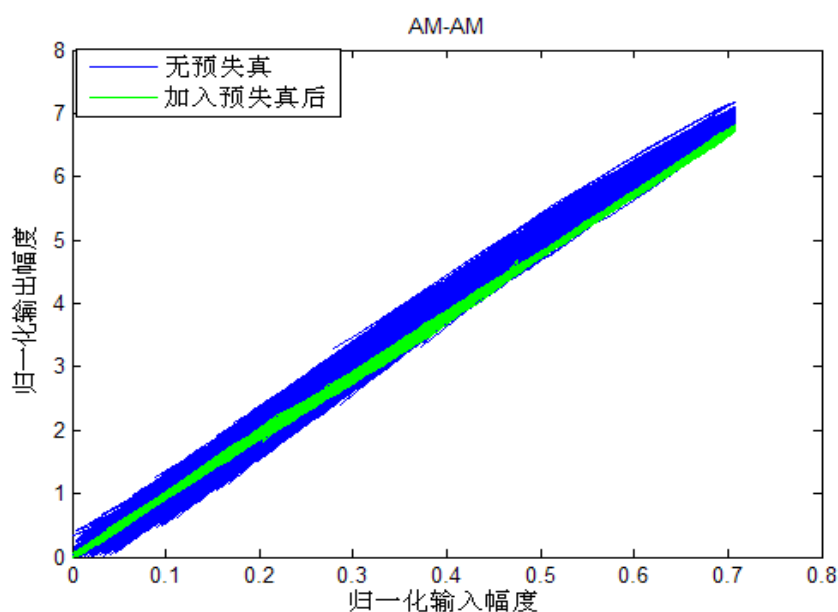
### 4. 仿真结果及评价

本文采用深度为 3，阶数为 5 的记忆多项式模型作为预失真模型，并通过 RLS 算法自适应学习，可以求出模型的参数  $b$  如下：

$b =$

$1.5657 + 0.0176i, -0.7233 - 1.1917i, 0.8452 + 3.4528i, 0.0586 - 4.2613i, -0.1075 + 2.1027i$   
 $-1.0981 + 0.4194i, 0.5077 + 0.7655i, -1.0498 - 2.4610i, 1.0974 + 3.3133i, -0.7349 - 1.7631i$   
 $0.6980 - 0.2849i, -0.9444 - 0.8252i, 2.6722 + 2.8334i, -3.6270 - 4.1226i, 2.0031 + 2.2365i$

加入预失真器前后的 AM-AM 与 AM-PM 响应如图 4-10 所示：



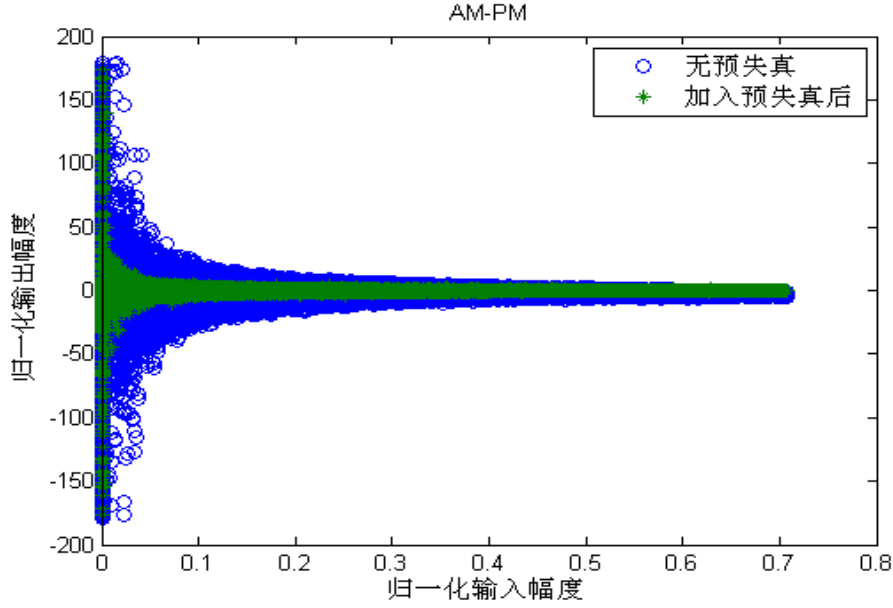


图 4-10 记忆多项式模型输出的 AM-AM 和 AM-PM 曲线

由图 4-10 可知，预失真器的记忆，已经对功放的记忆做了相当程度的补偿。线性功放的 AM-AM 与 AM-PM 相对原始功放已经变得很细了，说明线性功放的记忆效应已经明显减弱。

此时，在满足题目要求的两个“输出幅度限制”和“功率最大化”的约束情况下，线性化后功放最大可能的幅度放大倍数为 9.45 倍， $NMSE = -43.37\text{dB}$ ， $EVM = 0.68\%$ 。

### 4.3 问题三：功率谱密度的计算

从时域数据上分析，功放的记忆效应表现为当前输出不仅仅与当前时刻输入有关，还与前几个时刻输入和输出有关；从频域分析，功放的记忆效应体现为功放的响应随着频率而变化，这体现为互调分量的不对称性，即上边带互调分量幅度和相位均不相同，而且随着频率变化，即功放的非线性体现为带外频谱泄露或频谱扩展或称为频谱再生。带内失真和带外频谱扩散会造成带内误码率上升，带外对邻信道产生干扰，严重影响通信系统的性能。

相邻信道功率比（Adjacent Channel Power Ratio, ACPR）是表示信道的带外失真的参数，衡量由于非线性效应所产生的新频率分量对邻道信号的影响程度。其定义为

$$ACPR = 10 \log_{10} \frac{\int_{f_2}^{f_3} s(f) df}{\int_{f_1}^{f_2} s(f) df} \quad (40)$$

ACPR 越小表明一定带宽范围内临道间干扰越小。ACPR 在一定程度上也反映了信号的失真程度，ACPR 越大说明功放的非线性越强，那么信号失真越严重。

其中  $s(f)$  为信号的功率谱密度函数， $[f_1, f_2]$  为传输信道， $[f_2, f_3]$  为相邻信道。

#### 4.3.1 功率谱密度计算原理

设一个能量信号 $s(t)$ 的能量为  $E$ ，则此信号的能量由下式决定：

$$E = \int_{-\infty}^{\infty} s^2(t) dt \quad (41)$$

若此信号的傅里叶变换为 $S(f)$ ，则有 Parseval 定理得知：

$$E = \int_{-\infty}^{\infty} s^2(t) dt = \int_{-\infty}^{\infty} |S(f)|^2 df \quad (42)$$

式（42）表示 $|S(f)|^2$ 在频率轴 $f$ 上的积分等于信号能量，所以称 $|S(f)|^2$ 为能量谱密度。由于功率信号具有无穷大的能量，式（41）不存在，所以不能计算功率信号的能量谱密度，但是，可以求出它的功率谱密度，为此，我们先将信号 $s(t)$ 截短为长度等于  $T$  的一个截短信号 $s_T(t)$ ， $-T/2 < t < T/2$ 。这样 $s_T(t)$ 就成为一个能量信号了，对于这个能量信号，我们就可以用傅里叶变换求出其能量谱密度 $|S_T(f)|^2$ ，并由 Parseval 定理可得

$$E = \int_{-T/2}^{T/2} s_T^2(t) dt = \int_{-\infty}^{\infty} |S_T(f)|^2 df \quad (43)$$

于是，我们可将

$$\lim_{T \rightarrow \infty} \frac{1}{T} |S_T(f)|^2 \quad (44)$$

定义为信号的功率谱密度 $P(f)$ ，即

$$P(f) = \lim_{T \rightarrow \infty} \frac{1}{T} |S_T(f)|^2 \quad (45)$$

本文所用数据可以看为有限长离散序列，对数据进行分析可得其功率 $P = 0.0981$ 。

#### 4.3.2 仿真结果及评价

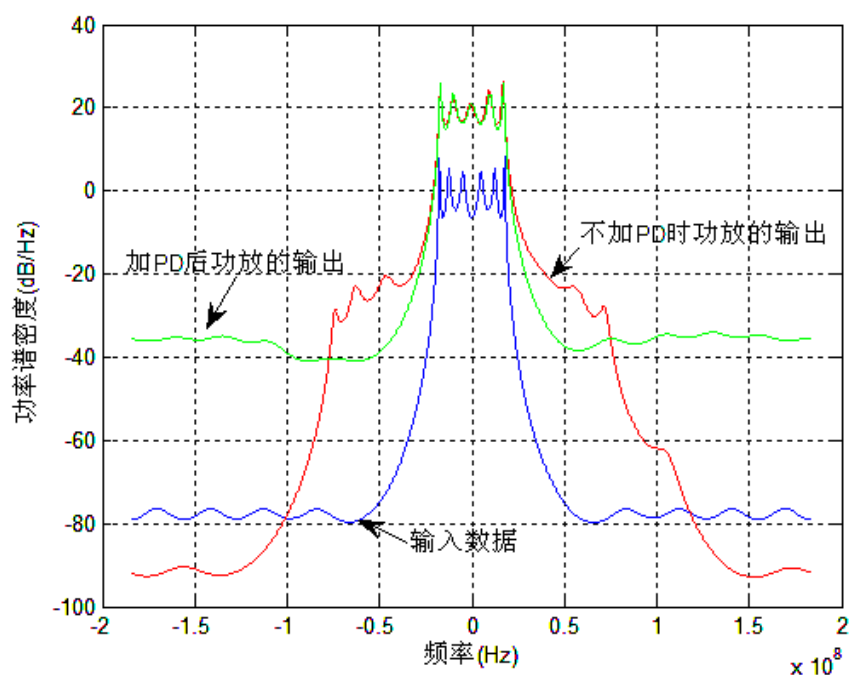


图 4-11 功率谱密度

由图 可得：未加预失真模型时，功放输出的 ACPR 为-21dB，加入预失真模型后，功放输出的 ACPR 改善为-40dB。

## 5. 模型的优缺点

### 优点：

1. 采用的 Saleh 模型简单，参数少，能够精确描述功率放大器的非线性和记忆效应。
2. 幂级数模型可以方便地用最小二乘算法辨识其中的参数。
3. RLS 算法能很好地改善了功率放大器的非线性，失真提高了系统的线性化性能。
4. 记忆多项式模型能很好的描述功率放大器的记忆效应，忽略了所有不同时刻输入信号的交叉项，相对于 Volterra 级数运算复杂度大大降低。

### 缺点：

1. Saleh 模型的实际效果没有复系数多项式建模效果优越。
2. 幂级数模型如果采用很高的阶数，模型的输出结果容易发生“振荡”。因此，仅仅依靠增加阶数未必能获得很高的模型精度。
3. RLS 算法的电路实现复杂。

### 参考文献：

- [1]. 张小梅，胡方明，任爱锋。基于数字预失真技术的功放线性化研究[J]。计算机仿真，2012，29(7)：393-397。
- [2]. 金哲。宽带通信中有记忆射频功率放大器的建模与预失真方法 [D]。杭州：浙江大学，2007。
- [3]. 黄勇。射频功放数字预失真技术的研究[D]。杭州电子科技大学，2012。
- [4]. 金绍春。射频功放数字预失真技术研究[实现][D]。电子科技大学，2012。

- [5]. 姚应磊. 功率放大器的基带预失真方法研究[D]. 西安: 西安电子科技大学, 2010.
- [6]. Raviv Raich, et al. Orthogonal Polynomials for Power Amplifier Modeling and Predistorter Design. IEEE Trans. Vehicular technology, 2004, 53(5):1468-1479
- [7]. 张小梅, 胡方明, 任爱锋. 基于数字预失真技术的功放线性化研究[J]. 计算机仿真, 2012, 29(7): 393-397.
- [8]. 赵毅峰, 姚彦, 曹新容. 数字预失真技术设计及实现[J]. 厦门大学学报: 自然科学版, 2010, 49(004): 501-504.
- [9]. Kim J, Konstantinou K. Digital predistortion of wideband signals based on power amplifier model with memory[J]. Electronics Letters, 2001, 37(23): 1417-1418.
- [10]. Yang X S, Deb S. Cuckoo search via Lévy flights[C]//Nature & Biologically Inspired Computing, 2009. NaBIC 2009. World Congress on. IEEE, 2009: 210-214.
- [11]. Kevin G. Gard, Hector M. Gutierrez, Michael B. Steer, "Characterization of Spectral Regrowth in Microwave Amplifier Based on the Nonlinear Transformation of a Complex Gaussian Process", IEEE Transaction on Microwave Theory and Techniques, Vol. 47, No. 7, July 1999: 1059-1069
- [12]. Understanding CDMA Measurements for Base Station and their Components, Agilent Application Note 1311 (5968-0953E)
- [13]. Josef Wolf, Bob Buxton "Measure Adjacent-Channel Power With A Spectrum Analyzer", Microwave & RF, January 1997: 55-64
- [14]. 应鲁曲, "W-CDMA RF 功率放大器线性性能的研究", 解放军理工大学学报自然科学版, Vol. 2 No. 5 Oct. 2001: 15-19

## 附录 A: 记忆多项式模型的系数表

表 1 M=4, K=7 记忆多项式模型的系数表

10.2712-1.2741i	-0.5325+6.1003i	57.0957-7.9121i	-247.4491-28.4620i	440.6682+ 87.5888i
-385.7173-91.8145i	138.8227+36.6073i	-3.9101-2.4886i	2.6803+ 5.1985i	-17.4302-42.9545i
41.7045+148.2761i	27.7404-231.0096i	-115.5091+169.3771i	62.4003- 51.0784i	-1.5097+ 5.1309i
-0.1968+ 0.0824i	20.8520+ 17.8534i	-82.2382- 80.2802i	55.6299+ 116.3429i	40.5797- 63.4543i
-35.2014+ 11.8788i	9.1244- 5.1250i	0.5872- 0.2135i	-14.7235-9.5849i	60.1789+41.6168i
-54.0482- 47.6503i	-3.8133+ 6.5283i	11.9227+ 8.2179i	-5.7145+ 2.5409i	-0.3069- 1.4038i
-1.5527+ 16.2531i	19.3139- 61.4904i	-76.7475+ 104.5927i	107.9137- 85.4954i	-50.3531+28.4973i

表 2 M=3, K=7 记忆多项式模型的系数表

0.1483 - 0.0330i	-0.0074 + 0.0618i	0.6045 - 0.0992i	-2.6210 - 0.2432i
4.8384 + 0.9315i	-4.3925 - 1.0945i	1.6141 + 0.4643i	-0.2293 + 0.0597i
0.0303 + 0.0495i	-0.2571 - 0.3713i	0.7432 + 1.3533i	-0.7675 - 2.4399i
0.1892 + 2.1584i	0.0511 - 0.7697i	0.2853 - 0.0818i	-0.0066 + 0.0048i
0.3247 + 0.0840i	-1.2798 - 0.5623i	2.0542 + 1.2352i	-1.5409 - 1.1891i
0.4832 + 0.4472i	-0.1218 + 0.0429i	0.0088 - 0.0105i	-0.2398 + 0.0417i
1.0551 - 0.0177i	-2.0475 - 0.0958i	1.9440 + 0.1353i	-0.7548 - 0.0593i

表 3 M=2, K=7 记忆多项式模型的系数表

$0.0271 + 0.0085i$	$0.0004 + 0.0602i$	$0.4690 - 0.1000i$
$-2.1069 - 0.1632i$	$3.9660 + 0.6872i$	$-3.6575 - 0.8082i$
$1.3481 + 0.3360i$	$0.1290 - 0.0626i$	$0.0185 + 0.0457i$
$-0.0057 - 0.3253i$	$-0.1281 + 1.0476i$	$0.4757 - 1.7631i$
$-0.6292 + 1.5247i$	$0.2862 - 0.5366i$	$-0.0741 + 0.0415i$
$0.0091 + 0.0051i$	$-0.0166 + 0.0476i$	$-0.0034 - 0.2390i$
$0.0154 + 0.4595i$	$0.0156 - 0.4210i$	$-0.0190 + 0.1523i$

表 4 M=4, K=5 记忆多项式模型的系数表

$9.9189 - 1.2861i$	$7.0911 + 6.7254i$	$-2.4976 - 15.8923i$	$-27.1568 + 10.2870i$	$24.9293 + 0.4217i$
$-3.9500 - 2.3153i$	$3.4519 + 1.5806i$	$-23.3173 - 13.3113i$	$70.5808 + 37.6043i$	$-54.0334 - 30.1197i$
$-1.5441 + 5.0724i$	$0.6299 + 1.2436i$	$14.9258 + 6.1613i$	$-68.4772 - 32.3858i$	$57.2033 + 30.1734i$
$9.1316 - 5.1091i$	$0.0796 - 0.5348i$	$-10.2452 - 5.4511i$	$45.7001 + 24.4204i$	$-38.0660 - 22.1815i$
$-5.6677 + 2.4693i$	$-1.2751 + 0.2052i$	$7.0537 + 2.9629i$	$-19.1837 - 10.8445i$	$14.2539 + 32.1418i$

表 5 M=3, K=5 记忆多项式模型的系数表

$14.4611 - 3.3133i$	$7.4119 + 6.8854i$	$-4.3146 - 18.6977i$	$-19.1156 + 18.5343i$
$18.3385 - 6.0562i$	$-22.9116 + 6.1373i$	$2.4023 + 1.0762i$	$-18.3923 - 5.1083i$
$46.1544 + 13.2887i$	$-33.9901 - 11.0081i$	$28.3955 - 8.2316i$	$2.3699 + 1.8782i$
$6.7854 - 5.4537i$	$-30.9010 + 2.5875i$	$26.7243 + 2.5303i$	$-12.0764 + 4.2646i$
$-1.7673 - 0.7521i$	$-0.6030 + 3.5126i$	$10.4842 - 3.8910i$	$-10.3563 + 0.5956i$

表 6 M=2, K=5 记忆多项式模型的系数表

$2.3817 + 0.8299i$	$7.2057 + 6.7518i$	$-8.5753 - 17.6188i$
$-5.9477 + 17.9044i$	$7.8913 - 6.7926i$	$12.8493 - 6.1265i$
$2.7472 + 1.4883i$	$-7.1258 - 8.1464i$	$12.7595 + 14.8382i$
$-8.0378 - 8.7688i$	$-7.4098 + 4.1277i$	$0.8973 + 1.1323i$
$-1.8229 - 0.9476i$	$0.4693 - 1.0764i$	$0.9224 + 1.2354i$

表 7 M=4, K=3 记忆多项式模型的系数表

$10.2088 - 0.7646i$	$6.2511 + 1.7795i$	$-9.3003 - 3.0300i$	$-3.8963 - 2.3465i$	$-0.7737 + 0.1227i$
$6.8914 + 1.6087i$	$-1.3487 + 5.4638i$	$2.8338 + 0.6434i$	$-10.1899 - 2.9423i$	$9.0422 - 5.4275i$
$-1.7568 - 0.6033i$	$6.8912 + 2.1373i$	$-5.7410 + 2.5585i$	$0.3587 + 0.5871i$	$-1.8280 - 1.0181i$

表 8 M=3, K=3 记忆多项式模型的系数表

$14.7397 - 2.8618i$	$6.1250 + 1.5444i$	$-7.8829 - 2.4515i$	$-22.8264 + 6.3560i$
$-0.2982 + 0.9711i$	$1.7181 - 0.4944i$	$28.5748 - 8.2039i$	$2.0007 - 0.8024i$
$-2.3906 + 0.3913i$	$-12.2138 + 4.2032i$	$-0.9580 + 0.7458i$	$1.0983 - 0.6030i$

表 9 M=3, K=3 记忆多项式模型的系数表

$2.6491 + 1.3193i$	$5.5537 + 1.9981i$	$-7.2502 - 2.7754i$
$12.8923 - 6.0183i$	$1.3665 - 0.3246i$	$-0.1585 + 0.3985i$
$-7.2751 + 4.2061i$	$0.0382 + 0.7153i$	$-0.1612 - 0.7132i$

## 附录 B: MATLAB 仿真程序

### 问题一: A. Saleh 模型建模仿真程序

```
function [NMSE]= MySaleh1(v)
%% *****数据导入*****
load pa_in_out_memoryless;
n = size(pa_in_memoryless,1);
plot(abs(pa_in_memoryless),abs(pa_out_memoryless));
%% *****输入数据绘图*****
figure(1);
plot(abs(pa_in_memoryless),abs(pa_out_memoryless),'b. ');
title('AM-AM')
xlabel('归一化输入幅度')
ylabel('归一化输出幅度')
grid on;
figure(2);
plot(abs(pa_in_memoryless),angle(pa_out_memoryless)/pi*180-angle(pa_in_memoryless)/pi*180,'b.',[0 1],[0 0], 'r')
title('AM-PM')
xlabel('归一化输入幅度')
ylabel('归一化输出相移')
%% *****模型的建立*****
y = zeros(size(pa_in_memoryless));
v=[3.1344 0.5920 0.0100 10.0000]; %模型参数
for j = 1:n
    y(j) = saleh(pa_in_memoryless(j),v);
end
error=pa_out_memoryless-y; %真实值与拟合曲线差值
nmse = (error)*(error)/(pa_out_memoryless'*pa_out_memoryless);
NMSE = 10*log10(nmse);
%% *****数据绘图*****
figure(3);
plot(abs(pa_in_memoryless),abs(pa_out_memoryless),'b.-',abs(pa_in_memoryless),abs(y),'g.-')
title('AM-AM')
xlabel('归一化输入幅度')
ylabel('归一化输出幅度')
legend('原始输出','Saleh 模型');
grid on;
figure(4);
plot(abs(pa_in_memoryless),angle(pa_out_memoryless)/pi*180-angle(pa_in_memoryless)/pi*180,'b.-',abs(pa_in_memoryless),angle(y)/pi*180-angle(pa_in_memoryless)/pi*180,'g.-',[0 1],[0 0], 'r')
title('AM-PM')
xlabel('归一化输入幅度')
ylabel('归一化输出相移')
legend('原始输出','Saleh 模型');%,'多项式模型'
grid on;
%% *****数据绘图*****
Fs=30.27*12*1000000;
[pxx,f]=pmem(y,50,10000, Fs);
```

---

```
[px,f]=pmem(pa_out_memoryless,50, 10000,Fs);
figure(5)
q1 = fftshift(pxx);
q2 = fftshift(px);
plot(f-150/2,10*log10(q1),'*-',f-150/2,10*log10(q2),'o-');
title('模型输出与功放输出功率密度谱');
xlabel('频率(MHz)')
ylabel('归一化的功率(dB)')
legend('模型输出','功放输出');
box off;
figure(6)
plot(f-150/2,10*log10(q1)-10*log10(q2));
xlabel('频率(MHz)')
ylabel('归一化的功率(dB)')
title('模型输出与功放输出功率谱密度差')
box off;
end
```

```
function [y]=saleh(x,v)
a1 = v(1);b1 = v(2);
a2 = v(3);b2 = v(4);
v=sqrt(-1);
    ain = abs(x);%输入信号的幅度
    thetain = angle(x);%输入信号的相位
    aout = a1*ain/(1+b1*ain^2);%输出的幅度
    thetapm = a2*ain^2/(1+b2*ain^2);%改变的相位
    thetaout = thetain+thetapm;%输出的相位
    y = aout*exp(v*thetaout);%输出的信号
end
% End of function file.
```

#### 问题一：B. 无记忆预失真建模程序

%采用 Memory Polynomial 作为预失真模型

clear all

format long

load pa\_in\_out\_memoryless;

K = 5;

G=1.858;

%% 有记忆

n = length(pa\_in\_memoryless);

PA\_in1= pa\_in\_memoryless;

PA\_out = pa\_out\_memoryless;

PA\_in=PA\_in1;

H = matrix\_generate1(PA\_in,K);

%输入矩阵，用于估计误差

Estimate=pinv(H)\*PA\_out;

%计算滤波系数

\*\*\*\*\*使用求出参数来求解滤波输出\*\*\*\*\*

% H1=matrix\_generate1([PA\_in(end-M+1:end);PA\_in],K);%输入矩阵

y=H\*Estimate;

%求解滤波输出

\*\*\*\*\*误差求解\*\*\*\*\*

er=(y-pa\_out\_memoryless)\*(y-pa\_out\_memoryless);

%方差

qi=pa\_out\_memoryless'\*pa\_out\_memoryless;

%功放输出信号模值平方

NMSE=10\*log10(sum(er)/sum(qi));

%滤波器均方误差

%下面迭代 20 次

Zn=PA\_in;

%b1=[1 0 0 0 0 0].'; %列向量





---

```

title('AM-PM')
xlabel('归一化输入幅度')
ylabel('归一化输入相移')
% *****多项式模型参数设定*****
n=length(PA_in);
M=4;
K=5;
% *****最小二乘法求解*****
H = matrix_generate1(PA_in,M,K); %输入矩阵，用于估计误差
Estimate=pinv(H)*(PA_out(M+1:n)) %计算滤波系数
% *****使用求出参数来求解滤波输出*****
H1=matrix_generate1([PA_in(end-M+1:end);PA_in],M,K);%输入矩阵
y1=H1*Estimate; %求解滤波输出
% *****误差求解*****
er=(y1-PA_out)*(y1-PA_out); %方差
% e1=abs(y-PA_out); %滤波器输出误差
qi=PA_out'*PA_out; %功放输出信号模值平方
NMSE=10*log10(sum(er)/sum(qi)); %滤波器均方误差
% *****数据绘图*****
% figure(3);
% plot(abs(pa_in_memory),abs(pa_out_memory),'b.-',abs(pa_in_memory),abs(y1),'r.-')
% title('AM-AM')
% xlabel('归一化输入幅度')
% ylabel('归一化输出幅度')
% legend('原始输出','复系数多项式模型');
% grid on;
% figure(4);
% plot(abs(pa_in_memory),angle(pa_out_memory)/pi*180-angle(pa_in_memory)/pi*180,'b.-',abs(pa_in_
memory),angle(y1)/pi*180-angle(pa_in_memory)/pi*180,'r.-',[0 1],[0 0],'r')
% title('AM-PM')
% xlabel('归一化输入幅度')
% ylabel('归一化输出相移')
% legend('原始输出','复系数多项式模型');
% grid on;
% *****AM/AM 和 AM/PM 曲线对比图*****%
u =1:n;
figure(3)
plot(abs(PA_in),abs(PA_out),'b.-');
legend('原始输出')
xlabel('归一化输入幅度')
ylabel('归一化输出幅度')
grid on;
title('AM-AM')
figure(5)
plot(abs(PA_in),abs(y1),'r.-');
legend('记忆多项式模型');
xlabel('归一化输入幅度')
ylabel('归一化输出幅度')
grid on;
title('AM-AM')
figure(4)
plot(abs(PA_in),angle(PA_out./PA_in)/pi*180,'b.-');
legend('原始输出');
xlabel('归一化输入幅度')

```



---

```

% qi=pa_out_memory'*pa_out_memory;           % 功放输出信号模值平方
% NMSE1=10*log10(er/qi);                       % 滤波器均方误差
% *****
% 通过间接学习提取预失真参数
% 下面迭代 10 次
Zn=PA_in;
for i=1:20
    Yn1=1/g*MPM_PA_mode2(Zn,M,K,Estimate);      %% Yn1=MPM_PA_model(Zn,Cpa);%% Zn
为预失真输出, 功放输入 Zn 为预失真输出, 功放输入
    H = matrix_generate1(Yn1,M,K);              % 输入矩阵, 用于估计误差
    b1 = pinv(H)*(Zn(M+1:n));                    % 计算滤波系数
    Zn = MPM_PA_mode2(PA_in,M,K,b1);            %% 预失真输出
end
Zn = MPM_PA_mode2(PA_in,M,K,b1);%Zn=MPM_PA_model(Xn,b1);

index = find(abs(Zn)>MaxA);                      % 判断是否超过最大幅值, 如果超过了赋值为
最大幅值
% figure(1)
% plot(abs(Zn)/MaxA);

Yn2 = MPM_PA_mode2(Zn,M,K,Estimate);% Yn2=MPM_PA_model(Zn,Cpa);% 预失真输出
% figure(1);
% plot(abs(PA_in),abs(PA_out),'.',abs(PA_in),abs(Yn2),'.g',abs(PA_in),abs(Zn),'.r')
figure(1)
plot(abs(PA_in),abs(PA_out),",abs(PA_in),abs(Yn2),'g')
    xlabel('归一化输入幅度');
ylabel('归一化输出幅度');
title('AM-AM');
legend('无预失真','加入预失真后')
figure(2)
plot(abs(PA_in),angle(PA_out./PA_in)/pi*180,'o',abs(PA_in),angle(Yn2./PA_in)/pi*180,'*');
xlabel('归一化输入幅度')
ylabel('归一化输出相移');
title('AM-PM')
legend('无预失真','加入预失真后')
% *****误差求解*****
er=(Yn2-g*pa_in_memory)*(Yn2-g*pa_in_memory);    % 方差
qi=(g*pa_in_memory)*(g*pa_in_memory);            % 功放输出信号模值平方
NMSE2=10*log10(er/qi);                           % 滤波器均方误差
EVM = sqrt(er/n/(qi/n));
% *****

问题 3. 功率谱求解程序
% *****判断是能量信号还是功率谱*****
clc,clear
load('pa_in_out_memory.mat')
pa_in=abs(pa_in_memory);
pa_out=abs(pa_out_memory);
E=sum(pa_in.^2);
P=E/length(pa_in);
% *****求功率谱*****
clc,clear;
load('pa_in_out_memory.mat')
load('YN1.mat')
pa_in=pa_in_memory;

```

---

```

pa_out=pa_out_memory;
Nfft=4096;
Fs=30.72*12*1000000;
[Pxx1,f1]=pmem(pa_in,14,Nfft,Fs);
[Py1,f2]=pmem(pa_out,14,Nfft,Fs);
[Pynew,f3]=pmem(Yn2,14,Nfft,Fs);
figure(1)
Pxx1=fftshift(Pxx1);
Py1=fftshift(Py1);
Pynew=fftshift(Pynew);
plot(f1-Fs/2,10*log10(Pxx1));
hold on
plot(f2-Fs/2,10*log10(Py1),'r')
hold on
plot(f3-Fs/2,10*log10(Pynew),'g')
xlabel('频率(Hz)');
ylabel('功率谱密度(dB/Hz)');
grid on;
h = annotation('textarrow',[0.45 0.40],[0.2 0.23]);
set(h, 'string','输入数据', 'fontsize', 10);
h = annotation('textarrow',[0.63 0.6],[0.61 0.57]);
set(h, 'string','不加 PD 时功放的输出', 'fontsize', 10);
h = annotation('textarrow',[0.27 0.24],[0.57 0.5]);
set(h, 'string','加 PD 后功放的输出', 'fontsize', 10);

```