

RF 通訊的數位預失真：從等式到實現方案

本文介紹數位預失真 (DPD) 的數學基礎知識，以及其如何在收發器的微處理器和硬體中實現。闡述為何現代通訊系統需要 DPD，並探討如何透過數學模型捕捉現實世界的訊號失真。

■作者：Claire Masterson / ADI 系統工程師

簡介

DPD 是數位預失真的首字母縮寫，許多射頻 (RF) 工程師、訊號處理愛好者和嵌入式軟體發展人員都熟悉這一術語。DPD 在行動通訊系統中隨處可見，使功率放大器 (PA) 能夠有效為天線提供最大功率。隨著 5G 使基地台中的天線數量增加，頻譜變得更加擁擠，DPD 開始成為一項關鍵技術，支援開發經濟高效且符合規格要求的行動通訊系統。

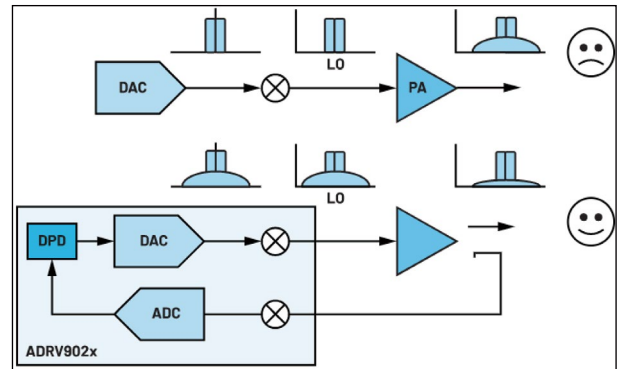
對於 DPD，無論從純粹的數學角度出發，還是在微處理器上實現更受限制，我們許多人都有自己獨特的見解。您可能是負責評估 RF 基地台產品中 DPD 性能的工程師，或者是一名演算法開發人員，很想知道數學建模技術在實際系統中的實現方式。本文目的在拓寬您的知識面，協助您從各個角度全面瞭解這個主題。

什麼是 DPD？為什麼要使用 DPD？

當基地台射頻裝置輸出 RF 訊號時 (參見圖 1)，需要先將其放大，然後再透過天線發射。我們使用 RF PA 來執行此操作 (放大)。在理想情況下，PA 接收輸入訊號，然後輸出與其輸入成正比的更高功率訊號。在執行此操作期間，PA 會盡可能保持高能效，將提供給放大器的大部分直流電源都轉化為訊號輸出功率。

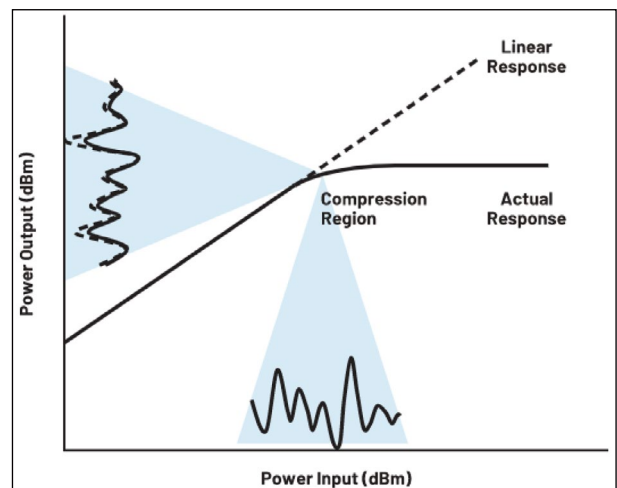
但這不是一個理想的世界。PA 由電晶體構成，電晶體是主動元件，本身具有非線性。如圖 2 所示，

圖 1：採用和未採用 DPD 技術的簡化射頻結構框圖。



如果我們在其「線性」區域使用 PA (這裡的線性是相對而言；所以加了引號)，則輸出功率與輸入功率相對成比例。此方法的缺點是 PA 的使用效率通常很低，提供的大部分功率都會作為熱量流失。我們

圖 2：PA 輸入功率與輸出功率之間的關係圖 (顯示了樣本輸入 / 輸出訊號的投影)。



通常希望在 PA 開始壓縮時使用。這表示如果輸入訊號增加了設定量 (例如 3 dB)，PA 輸出不會增加同樣的量 (可能只增加 1 dB)。很顯然，此時放大器使訊號嚴重失真。

這種失真發生在頻域中的已知位置，具體取決於輸入訊號。圖 3 顯示了這些位置，以及基頻與這些失真產物之間的關係。在 RF 系統中，我們只需要對基波訊號附近的失真進行補償，這些訊號是奇階交調產物。系統濾波處理頻外產物 (諧波和偶階交調產物)。圖 4 顯示 RF PA 的壓縮點附近的輸出。交調產物 (特別是三階) 清晰可見，就像是圍繞著目標訊號的「裙擺」。

DPD 目的在透過觀察 PA 輸出來表徵這種失真，要瞭解所需輸出訊號，隨之更改輸入訊號，使得 PA

圖 3: 雙音輸入交調和諧波失真的位置。

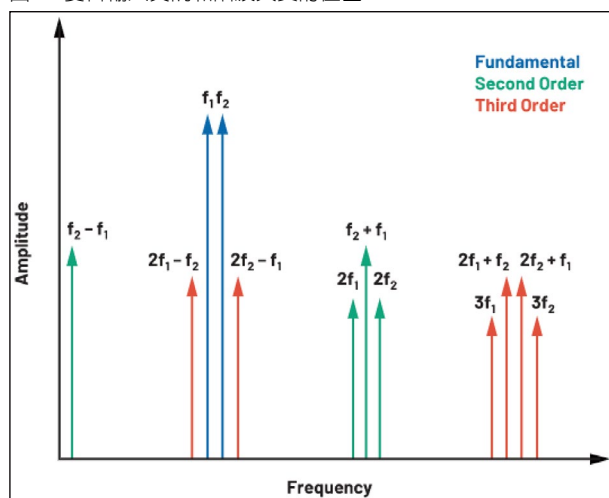
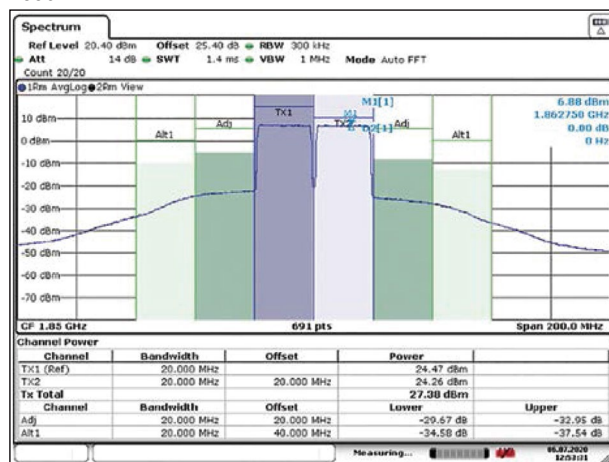


圖 4: 2 × 20 Mhz 載波通過 SKY66391-12 RF PA。中心頻率 = 1850 MHz。



輸出接近理想值。只有在相當具體的情況下才能有效地實現這一目標。我們需要配置放大器和輸入訊號，使放大器有一定程度的壓縮但未完全飽和。

PA 失真建模背後的數學計算

看到希臘字母和其他數學符號，會不會讓人想起過去可怕的大學考試？不止是您如此！如果大家得到的首批參考資料中有一份是涉及很多數學知識的學術論文，那麼很可能會打退堂鼓，不想去看它。這篇論文「射頻功率放大器數位預失真的廣義記憶多項式模型」¹ 具有開創性，它介紹了針對 DPD 廣泛使用的廣義記憶多項式 (GMP) 方法。如果您只涉足訊號處理方面的工作，上述主題內容對您而言可能有點深奧。所以，我們先嘗試拆解一下 GMP 方法，以更直覺瞭解數學在其中的作用。

Volterra 級數是 DPD 的重要數學基礎，它用於建立具有記憶的非線性系統模型。記憶僅僅表示系統的當下輸出取決於當下和過去的輸入。Volterra 級數很常用 (所以功能強大)，在電氣工程以外的許多領域都有使用。對於 PA DPD，Volterra 級數可以精簡使用，使其在即時數位系統中更易實現，也更穩定。GMP 就是這樣一種精簡方法。

圖 5 顯示如何使用 GMP 對 PA 的輸入 x 和輸出 y 之間的關係進行建模。可以看到，該等式的三個單獨的求和塊彼此都非常相似。我們先來看看下方用紅色圈出來的第一個。 $|x(\dots)|^k$ 項是指輸入訊號的包絡，其中 k 是多項式階。將記憶集成到系統中。如果 $L_a = \{0, 1, 2\}$ ，那麼該模型允許輸出 $y_{GMP}(n)$ 由當下的輸入 $x(n)$ 和過去輸入 $x(n-1)$ 和 $x(n-2)$ 決定。圖 6 分析多項式階 k 對樣本向量的影響。向量 x 是單個 20 MHz 載波，在複基頻上表示出來。去除記憶部分，以簡化 GMP 建模等式。 $|x| \cdot |x|^k$ 圖顯示的失真與圖 4 中的實際失真非常相似。

每個多項式階 (k) 和記憶延遲 (l) 都有相關的複值權重 (a_{kl})。在選擇模型的複雜程度之後 (其中包括 k 和 l 的值)，需要根據已知輸入訊號的 PA 輸出實際觀測值來求解這些權重。圖 7 將簡化的等式轉

我們來看看圖 6 中等式的第二行和第三行，為

升 DPD 的建模精度。在我們嘗試對放大器的行為建模時，這些項提供了額外的自由度。注意， M_b 、 M_c 、 K_b 和 K_c 不包含 0；否則，會重複第一行的項。

現在有了模型架構，我們從數學角度來解決該問題的最一個方面是如何求解權重係數。在實際場景中，人們傾向於求解上述模型的倒數。事實上，這些模型係數能夠彼此互換，可以使用相同的權重對接收到的 PA 輸出向量進行後處理，以消除非線性，並透過發送的發射訊號損失失真，使 PA 輸出盡可能呈現線性。圖 8 所示的框圖中，顯示了如何對權重係數進行估算和預失真。

在逆模型中，將圖 7 給出的矩陣等式互換，給出 $\mathbf{X} = \mathbf{Y}\mathbf{w}$ 。其中，矩陣 $\mathbf{Y}$ 的構成方式

$$\begin{aligned}
 y_{GMP}(n) = & \sum_{l \in L_a} \sum_{k \in K_a} a_{kl} x(n-l) |x(n-l)|^k \\
 & + \sum_{k \in K_b} \sum_{l \in L_b} \sum_{m \in M_b} b_{klm} x(n-l) |x(n-l-m)|^k \\
 & + \sum_{k \in K_c} \sum_{l \in L_c} \sum_{m \in M_c} c_{klm} x(n-l) |x(n-l+m)|^k
 \end{aligned}$$

$$y_{GMP}(n) = \sum_{l \in L_a} \sum_{k \in K_a} a_{kl} x(n-l) |x(n-l)|^k$$

$$+ \sum_{k \in K_b} \sum_{l \in L_b} \sum_{m \in M_b} b_{klm} x(n-l) |x(n-l-m)|^k$$

$$+ \sum_{k \in K_c} \sum_{l \in L_c} \sum_{m \in M_c} c_{klm} x(n-l) |x(n-l-m)|^k$$

Simplify

$\triangleright K_a = \{0, 1, 2, 3, 4\}$
 $\triangleright L_a = 0$

$$y_{GMP}(n) = \sum_{k \in K_a} a_{kl} \boxed{c(n) |x(n)|^k}$$

Plot Highlighted Element for Each Value of k
 $y_{GMP}(n)$ is the Sum of These Elements, Each Multiplied by a Complex Weight, a_{kl}

Convert From Equations to Matrix Operations on Vectors/Buffers of Data

$$y_{GMP}(n) = \sum_{k \in K_d} a_k x(n) |x(n)|^k$$

$\triangleright K_d = \{0, 1, 2, 3, 4\}$
 $\triangleright L_d = 0$

$$\begin{bmatrix} y_{GMP} \end{bmatrix} = \begin{bmatrix} x & x|x|^1 & x|x|^2 & x|x|^3 & x|x|^4 \end{bmatrix} \begin{bmatrix} a_{00} \\ a_{10} \\ a_{20} \\ a_{30} \\ a_{40} \end{bmatrix} \Rightarrow \hat{y} = XW$$

與其他示例中 X 的構成方式相同，如圖 9 所示。在本例中，包含了一個記憶項，且減少了包含的多項式的階數。為了解 w ，我們需要得出 Y 的倒數。 Y 不是方形的（是一個瘦長矩陣），所以需要使用「偽逆」矩陣進行求解（參見等式 1）。這是從最小二乘意義上求解 w ，也就是說，最小化了 X 和 Yw 之間的差的平方，正合我們的心意！

$$w = (Y^H Y)^{-1} Y^H x \quad (1)$$

有鑑於是在具有不同訊號的真實環境中使用，我們可以對其進一步優化。在這裡，係數是基於之前的值進行更新，因此受到限制。 μ 是 0 和 1 之間的常數值，用於控制每次反覆運算時權重的變化量。如果 $\mu = 1$ ， $w_0 = 0$ ，那麼此等式立即恢復到基本最小二乘解。如果將 μ 設為小於 1 的值，則需要多次反覆運算才能使係數收斂。

$$w_{i+1} = w_i + \mu(Y^H Y)^{-1} Y^H e, \quad e = x - \hat{x} \quad (2)$$

注意，這裡描述的建模和估算技術並非是執行 DPD 的唯一方式。也可以使用其他技術，例如基於動態偏差減少的建模來替代或作為附加方法使用。上述用於求解係數的估算技術具

有多種實現方式。有鑑於本文篇幅短小，並非書籍，所以此處不再贅述。

如何在微處理器中實現這一項技術？

我們前面介紹了相關數學知識。下一個問題：如何在實際通訊系統中進行應用？它在數位基頻中實現，一般在微處理器或 FPGA 中實現。ADI 的 RadioVerse 收發器產品（例如 ADRV902x 系列）內建微處理器核心，其結構有助於輕鬆實現 DPD。

在嵌入式軟體中實現 DPD

涉及兩個方面。一是 DPD 執行器，對即時發送的資料執行即時預失真，二是 DPD 自我調整引擎，基於觀察到的 PA 輸出來更新 DPD 係數。

對於如何在微處理器或類似元件中即時執行 DPD 和許多其他訊號處理概念，關鍵在於使用查閱資料表 (LUT)。LUT 允許用更簡單的矩陣索引操作來代替成本高昂的運行時計算。我們來看看 DPD 執行器如何對發送的資料樣本應用預失真。代表符號如圖 8 所示，其中 $u(n)$ 表示要傳輸的新資料樣本， $x(n)$ 表示預失真版本。圖 10 顯示在給定場景下，獲取一個預失真樣本所需的計算。這是一個相對受限的示例，最高多項式階為三階，只有一次記憶選取和一個交叉項。即使在這種情況下，要獲取這樣一個資料樣本，也需要進行大量乘法、指數運算和加法運算。

在這種情況下，使用 LUT 可以減輕即時計算負擔。可以將圖 10 所示的等式改寫成圖 11 所示的樣式，其中輸入 LUT 的資料會變得更加明顯。每個 LUT 都包含等式中突顯項的結果值，它們對應

圖 9: 以矩陣形式表示的逆演算法等式。有些記憶包含在其中。

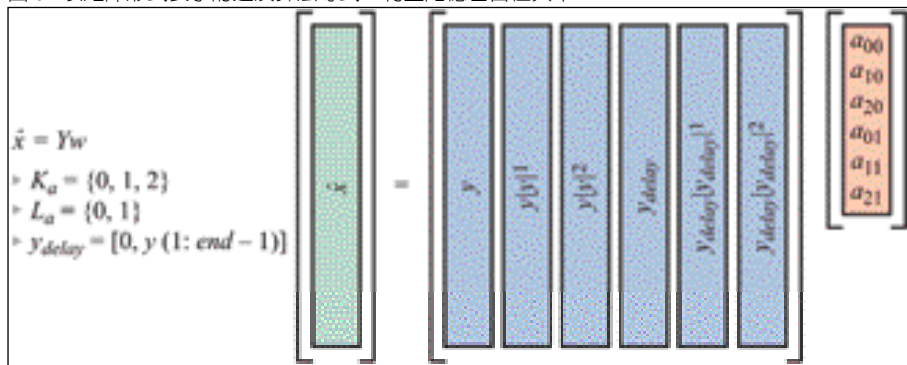


圖 10: 具有一次記憶選擇和一個三階交叉項元素的三階預失真計算案例。

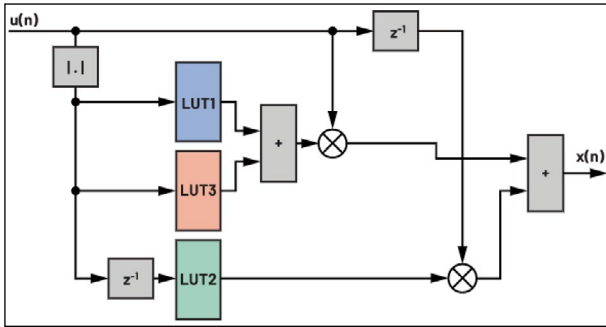
$$\begin{aligned} & K_a = \{0, 1, 2\} \\ & L_a = \{0, 1\} \\ & K_b = \{2\} \\ & L_b = \{0\} \\ & M_a = \{1\} \end{aligned} \quad x(n) = a_{00}u(n) + a_{10}u(n)|u(n)|^1 + a_{20}u(n)|u(n)|^2 + a_{01}u(n-1) + a_{11}u(n-1)|u(n-1)|^1 + a_{21}u(n-1)|u(n-1)|^2 + b_{20}|u(n-1)|^2$$

圖 11: 對等式項重新分組，以顯示 LUT 的結構。

$$x(n) = u(n) \left[a_{00} + a_{10}|u(n)|^1 + a_{20}|u(n)|^2 \right] + u(n-1) \left[a_{01} + a_{11}|u(n-1)|^1 + a_{21}|u(n-1)|^2 \right] + u(n) \left[b_{20}|u(n-1)|^2 \right]$$

LUT1
LUT2
LUT3

圖 12: 使用 LUT 可能實現 DPD 的框圖。



$|u(n)|$ 的多個可能值。解析度取決於在可用硬體中實現的 LUT 大小。當下輸入樣本的幅度大小基於 LUT 的解析度進行量化，可以作為索引，用於存取給定輸入的正確 LUT 元素。

圖 12 顯示如何將 LUT 整合到我們示例案例的完全預失真執行器實現方案。注意，這只是其中一種可能的實現方法。在仍然保持相同輸出的情況下，可以做出更改，例如：可以將延遲元素 z^{-1} 移動到 LUT2 右側。

自我調整引擎負責求解用於計算執行器中的 LUT 值的係數。這涉及到求解等式 1 和 2 中描述的 w 向量。偽逆矩陣運算 $(Y^H Y)^{-1} Y^H$ 會耗費大量計算資源。等式 1 可以改寫為

$$Y^H Y w = Y^H x \quad (3)$$

如果 $C_{YY} = Y^H Y$ ， $C_{Yx} = Y^H x$ ，等式 3 會變成

$$C_{YY} w = C_{Yx} \quad (4)$$

C_{YY} 是矩形矩陣，可以透過柯列斯基分解方法分解為上三角矩陣 L 和共軛轉置矩陣 ($C_{YY} = L^H L$) 的乘積。這樣我們可以透過引入一個虛擬變數 z 來求解 w ，求解方法如下：

$$L^H z = C_{Yx} \quad (5)$$

然後，重新代入這個虛擬變數，求解

$$L w = C_{Yx} \quad (6)$$

因為 L 和 L^H 分別是上、下三角矩陣，所以花費很少的計算資源，就可以求解等式 5 和等式 6，得出 w 。自我調整引擎每次運行，得出 w 的新值時，都需要更新執行器 LUT 來體現這一點。根據觀察到

的 PA 輸出，或者操作員掌握的待傳輸訊號的變化情況，自我調整引擎可以按照設定的定期間隔或不規則的間隔執行操作。

在嵌入式系統中實現 DPD 需要進行大量檢查和平衡，以確保系統的穩定性。最重要的是，發送資料緩衝器和捕捉緩衝器資料的時間要一致，以確保它們之間建立的數學關係是正確的，且在長時間之後仍然保持正確。如果這種一致性喪失，那麼自我調整引擎返回的係數將不能對系統執行正確的預失真，可能導致系統不穩定。還應檢查預失真執行器輸出，確保訊號不會使 DAC 飽和。

結論

本文從基礎數學的角度研究 DPD 及其在硬體中的實現方法，希望藉此揭示關於 DPD 的一些奧秘。本文對該主題的探討只是冰山一角，可能有助於推動讀者進一步研究通訊系統中訊號處理技術的應用情況。Pratt 和 Kearney 的研究是關於在有線通訊系統超寬頻寬用例中使用 DPD 的一個不錯的參考資料。² ADI 的 RadioVerse 收發器產品可以整合 DPD 這類演算法，為客戶提供高度整合的 RF 硬體和可配置的軟體工具。

參考電路

¹ Dennis R. Morgan、Zhengxiang Ma、Jaehyeong Kim、Michael G. Zierdt、John Pastalan。「射頻功率放大器數位預失真的廣義記憶多項式模型」。IEEE 訊號處理論文集，第 54 卷第 10 期，2006 年 10 月。

² Patrick Pratt、Frank Kearney。「超寬頻數位預失真 (DPD)：在電纜分配系統中實現帶來的優勢 (功率和性能) 和挑戰」。《類比對話》，第 51 卷第 3 期，2017 年 7 月。CTA