

白皮书

# 使用 MATLAB 进行射频功率放大器建模并借助 DPD 改善发射器线性度

本文探讨使用 MATLAB® 和 Simulink® 对射频功率放大器 (PA) 和数字预失真算法 (DPD) 进行建模和仿真的方法。这些工具让您可以快速开发思路、验证设计和构建强大的射频系统。

本文探讨使用 MATLAB® 和 Simulink® 对射频功率放大器 (PA) 和数字预失真算法 (DPD) 进行建模和仿真的方法。这些工具让您可以快速开发思路、验证设计和构建强大的射频系统。

## 简介

射频功率放大器位于大多数射频系统的前端，包括无线通信和雷达系统，对于确保无线系统保持合适的发射范围至关重要。然而，这些放大器存在一些非理想特性，因此在真实环境中部署时，会导致所驱动系统的性能下降。为了减缓这种性能下降，您可以应用 DPD 等线性化算法作为发射器或接收器链的一部分。DPD 算法旨在预先使要发射的波形失真，以抵消功率放大器的非线性效应。由于这些波形通常为宽频，因此通常必须在 FPGA/ASIC 上实现它们以达到实时性能。

应着眼于整个系统来设计 DPD 算法，因为 DPD 算法的性能很大程度取决于它所部署到的系统。仿真可提供整个系统的早期行为模型，并使在早期开发 DPD 算法成为可能，而无需依赖硬件。使用 Simulink 进行系统设计和仿真还为日后在开发过程中验证算法和其他组件提供了一个框架，并可生成在实验室中使用真实射频硬件很难实现的可重复结果。此外，您可以使用 Simulink 创建算法的位准确性和周期准确性模型，如同它们在固件上运行一样。您还可以通过这些模型生成产品级质量的 HDL 代码。

在本文中，我们将讨论射频功率放大器设计相关的挑战，重点介绍射频缺陷，以及对 PA 建模和借助 DPD 增加线性度的工作流程 (图 1)。

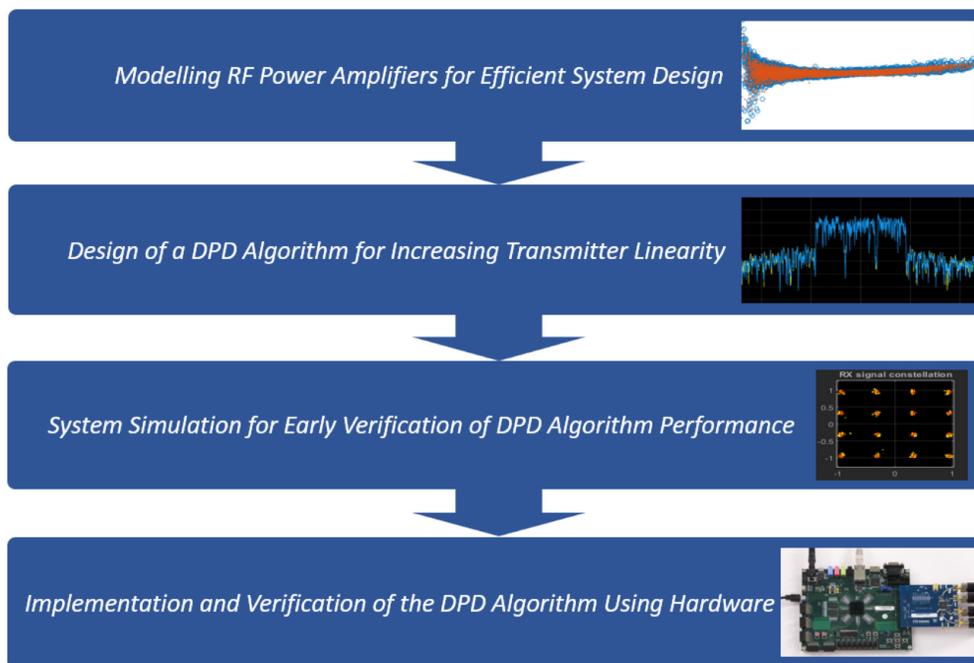


图 1: DPD 开发工作流程。

## 射频功率放大器设计面临的挑战

射频发射器的设计越来越具有挑战性。诸如 5G 等新标准的出现和对多重标准发射器的需求，需要我们做出一系列截然不同的设计决策。这包括针对以下各项进行权衡取舍：

- **大小：**发射器可能必须设计成小型便携设备，或占用的空间较小。
- **重量：**设计的便携性可能是确保其可用性的关键。
- **电源：**设备可通过太阳能或电池供电。
- **带宽：**发射器必须处理不断增加的信号带宽。
- **线性度：**放大器增益将通过输入功率以非线性方式增加。
- **频率：**系统可能需要在 10 Hz 到 100 GHz 的频率范围内运行。
- **整合的技术：**砷化镓等材料在用于加工晶体管时提升了高频性能。
- **MIMO 架构：**例如，为了在功能和灵活性之间实现良好的平衡，考虑使用混合波束成形架构。
- **可配置性：**系统必须设计为能够通过嵌入式算法重新配置，以便能够在不同条件下运行。

您选择的设计策略将对以上多个权衡取舍有影响。例如，增加发射器的带宽通常会更耗电，需要使用更大更重的组件和更专业的 IC 技术。由于这些组件的工作是确保信号功率足够大，因此在设计中，它们的需求往往占主导地位。

PA 包含许多非理想特性（包括非线性度和噪音），这些特性也必须经过评估。线性度用于描述输入功率和输出功率的行为。理想情况下，行为应呈完美线性，但实际并不是这样，如图 2 所示。

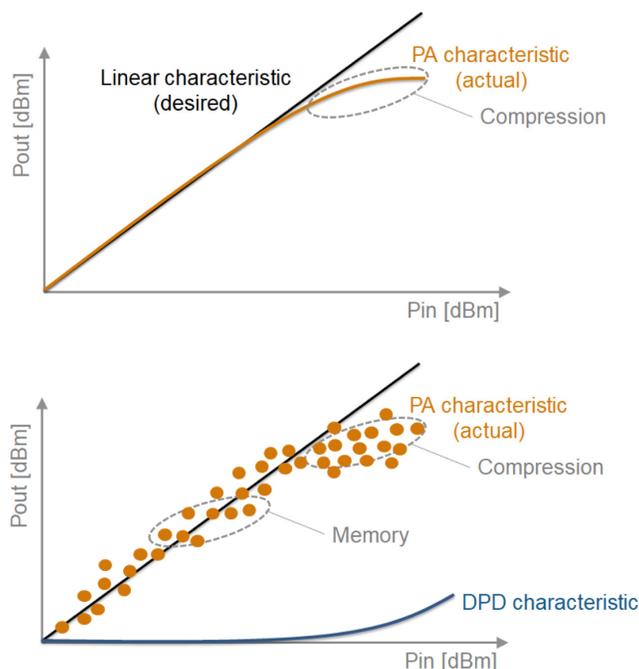


图 2: 上图: 显示压缩的非线性功率放大器的输入/输出功率特征 (橙色) 和理想的线性特征 (黑色)。下图: 显示记忆效应的非线性功率放大器的输入/输出功率特征 (橙色)、静态 DPD 特征 (蓝色) 到实现理想的线性特征 (黑色)。

输入功率增加到超出特定点时, 输出功率开始通过一个被称为**压缩**的效应以非线性方式增加。一般情况下, 随着系统的目标载频增加, 输入功率较小时会出现此情况, 从而导致高频放大器效率降低。更重要的是, 还会发生波形失真。此处更复杂的一点是, 功率放大器包含记忆效应, 因此输出功率将依赖于当前时间点和之前时间点的输入功率。这显示在图 2 的两个图形的下方区域。可使用包括 DPD 在内的各种技术线性化 PA 中的行为, 以使其更具确定性。

快速原型开发是实现最佳放大器和线性化设计的关键, 这种设计才能满足上述诸多要求。不过, 这么多条复杂系统需求会导致设计空间激增, 从而使物理原型设计太过耗时, 且无法扩展。仿真可帮助生成和测试许多原型, 缩短开发时间并实现最佳射频发射器设计。

## 对射频功率放大器进行建模以实现高效的系统设计

功率放大器行为受非线性行为和记忆效应的影响, 因此需要依赖信号带宽、频率、峰值平均功率比、调制和加载条件等输入信号特征。采集这些效应的大多数 PA 模型都是从 Volterra 系列衍生而来, 通常使用复杂的识别程序。目前已存在专有 (“黑箱”) 功率放大器模型, 但是这些模型往往存在问题, 因为它们缺乏透明度, 在复杂系统仿真中使用时很难自定义和理解。

MATLAB 提供了一个简单、快捷且可定制的 (“白箱”) 拟合程序, 用于创建随后可被集成到更大系统仿真的 PA 模型。此模型基于 [1]:

$$y_{MP}(n) = \sum_{k=0}^{K-1} \sum_{m=0}^{M-1} a_{km} x(n-m) |x(n-m)|^k .$$

其中:

a 是 PA 模型系数

k 是非线性度, K 是对 PA 进行建模所依据的度数

m 是记忆深度值, M 是 PA 模型的记忆深度

请注意, 当  $M = 1$  时, 上述模型描述的是纯静态特征, 例如 AM-AM/AM-PM。

该拟合程序使用来自 PA 的输入和输出数据, 基于选择的 K 值和 M 值提取 PA 模型系数。在完成拟合并创建 PA 模型后, 会评估拟合的质量。然后, 对多项式阶数 K 的值和记忆深度 M 的值进行迭代, 直到模型可以按预期执行。迭代可以手动处理, 也可以在 MATLAB 中自动处理。

例如, 在输入 LTE 调制测试信号时, 请思考以下在 PA 的输入和输出中采集的复杂 IQ 数据。图 3 显示了以时间函数表示的输入和输出信号幅度。

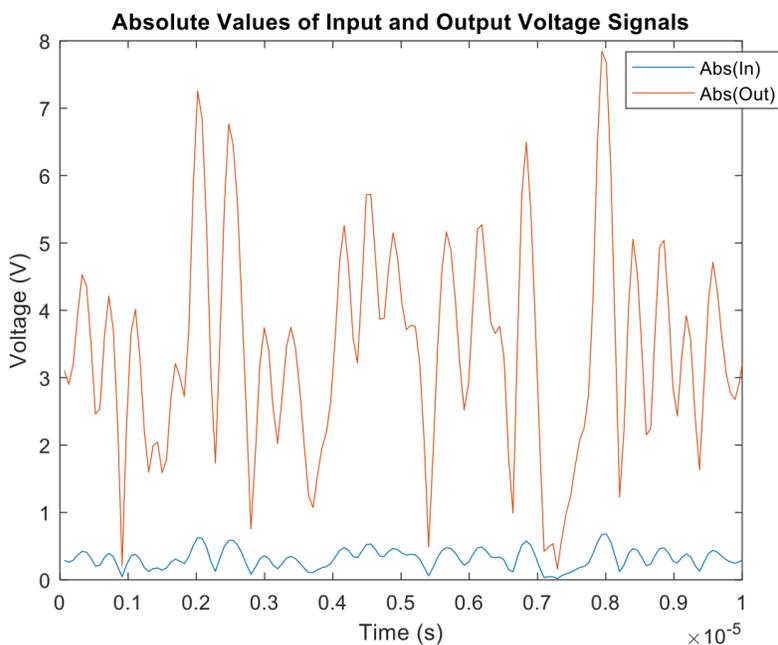


图 3: 输入/输出 PA 特征会绘制为时间的函数, 以显示输入波形的绝对值 (蓝色) 和输出波形的绝对值 (红色)。

图 4 显示了通过该数据绘制的功率传输特征。此处，X 轴显示输入功率，每个点代表瞬时的功率增益。笔直的水平线说明 PA 呈完美线性分布，沿此线条散开的点表明 PA 中存在记忆效应。因此，在此情况下，可以推断出 PA 略微呈非线性，且具有记忆效应。

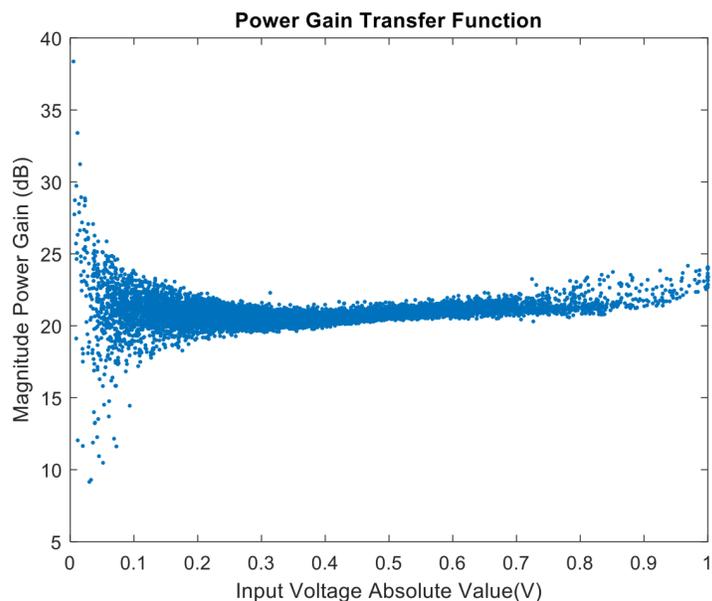


图 4: PA 的输入/输出功率特征。X 轴是输入电压的绝对值, Y 轴是 PA 功率增益的幅度。

现在,使用基于 MATLAB 的拟合程序提取系数,以馈送 PA 模型并衡量拟合质量。借助 MATLAB 反斜杠运算符,白箱 MATLAB 函数 `poweramp_helper.m` 可用于计算最小平方拟合求解。图 5 显示了一个代码片段。

```

43 -         switch modType
44 -             case 'memPoly'
45 -                 numCols = degLen;
46 -                 xTerms = zeros(xLen-memLenM1,memLen*degLen);
47 -
48 -                 for timeIdx = memLen:xLen
49 -                     % Common input variable entry terms for all rows in
50 -                     % coefficient matrix
51 -                     xTime = x(timeIdx-(0:memLenM1));
52 -                     xTerms(timeIdx-memLenM1,1:memLen) = xTime;
53 -                     for colIdx = 2:degLen
54 -                         colIdxM1 = colIdx-1;
55 -                         startPos = colIdxM1*memLen+1;
56 -                         endPos = startPos+memLen-1;
57 -                         xTerms(timeIdx-memLenM1,startPos:endPos) = ...
58 -                             xTime.*abs(xTime).^colIdxM1;
59 -                     end

```

图 5: 使用 MATLAB 实现 PA 模型系数的识别程序示例。

将一半的数据集用于创建模型, 然后使用完整数据集评估拟合质量, 方法是将输入数据馈送到 PA 模型, 然后将模型的输出数据与从真实 PA 记录的数据进行比较。首先, 从相对低的阶数  $K=3$  和记忆深度  $M=3$  开始, 可生成图 6 所示的拟合。

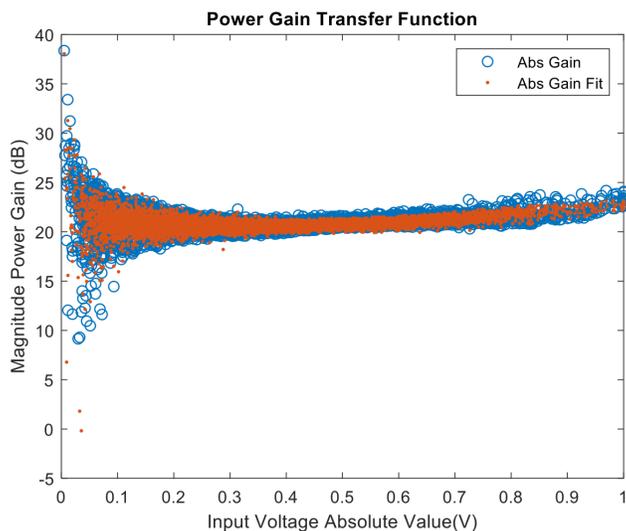


图 6: PA 的输入/输出功率特征, 显示了测量的征 (蓝色)、记忆深度  $M=3$  且多项式阶数  $K=3$  (红色) 时的拟合结果。

与真实数据 (蓝色) 相比, 模型点 (红色) 显示出较少的记忆效应以及稍多一些的线性度。用于衡量拟合质量的信号标准差将返回 5.148%。这可同时采集非线性行为和记忆效应。

将多项式长度增加到 4 ( $K=4$ ), 同时保证记忆深度增加到 3 ( $M=3$ ) 可改进 PA 模型, 从而使信号标准差降低到 4.2384%, 如图 7 所示。

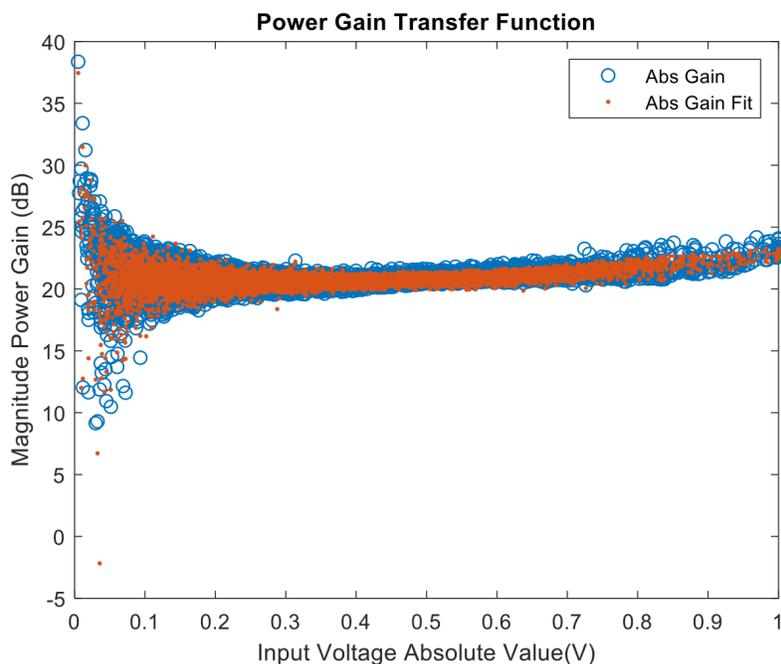


图 7: PA 的输入/输出功率特征, 显示了测得的特征 (蓝色) 和记忆深度  $M=3$  且多项式阶数  $K=4$  (红色) 时的拟合结果。

然而, 较高的  $K$  值和  $M$  值并不总意味着拟合已改进。相反, 过度拟合的模型可能会导致数值不稳定。对于此数据集尤其如此, 因为该数据集来自具有良好线性的 PA。例如, 将多项式长度和记忆深度均增加到 10 会导致信号标准差增加到 4.6972%, 与更复杂模型的拟合度会较差。

考虑拟合质量时, 分析频域和时域的质量指标很重要。图 8 显示了当记忆深度  $M=3$  且多项式长度  $K=4$  时, 通过拟合的 PA 模型所预测的信号及所测得信号的频谱。

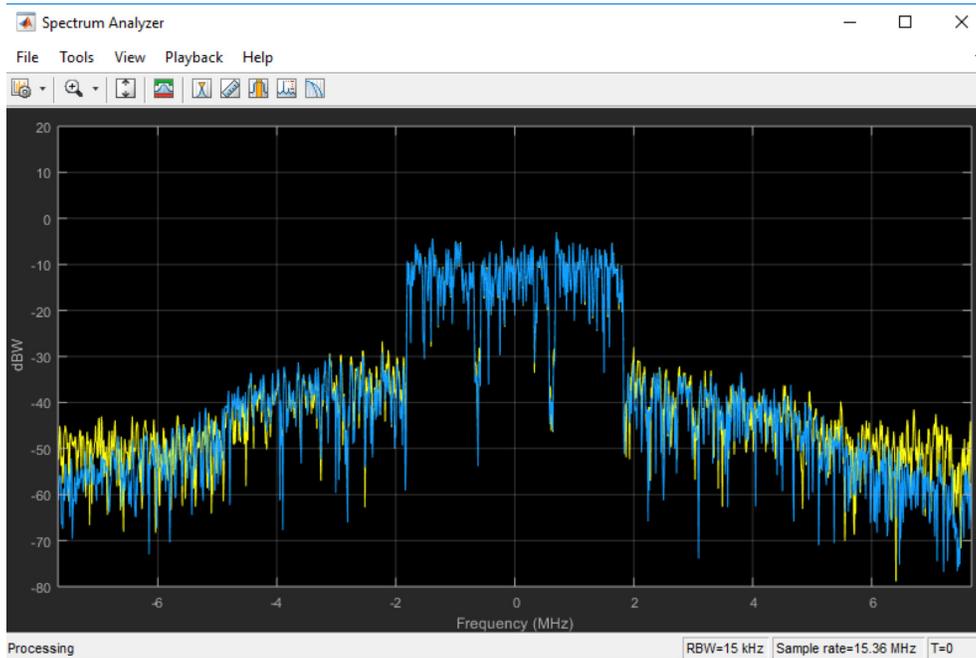


图 8: PA 输出特征的功率频谱, 显示了测得的特征 (黄色) 和记忆深度  $M=3$  且多项式阶数  $K=4$  时的拟合结果 (蓝色)。

此处, 黄色线条表示采集的 PA 输出数据的频谱, 蓝色线条是使用采集的 PA 输入数据输入时从 PA 模型输出收集的数据频谱。通过绘制这两个频谱之间的差异可得出图 9 显示的信号。

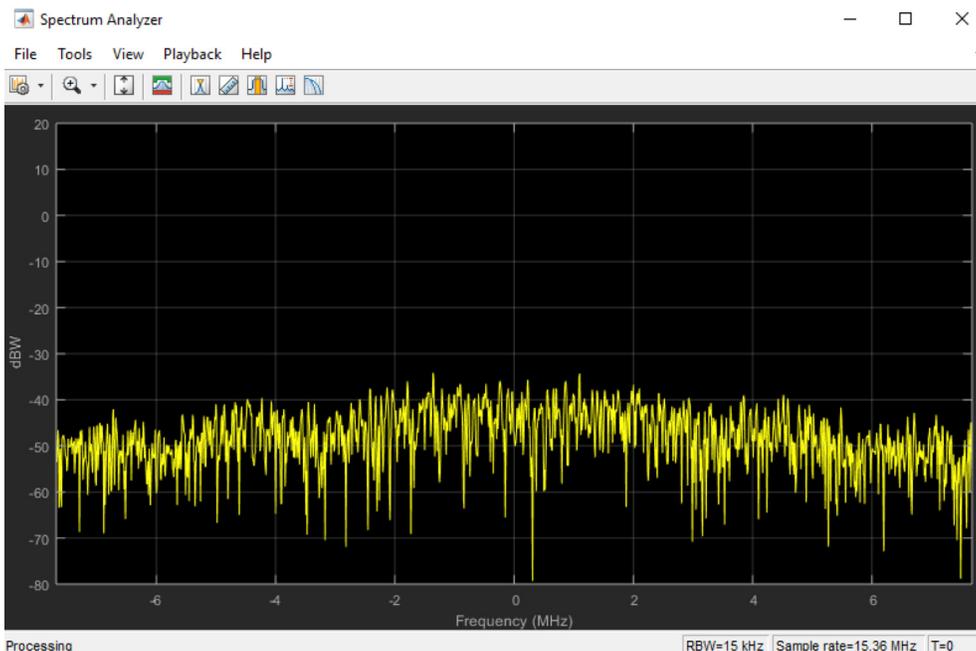


图 9: 错误信号的功率频谱图, 表示记忆深度  $M=3$  且多项式阶数  $K=4$  时, 使用一个记忆多项式模型测得的 PA 输出与预测的 PA 输出之间的差异。

如图所示, 对于此模型, 频域中的拟合通常较好。不过, 在主通带和频谱边缘, 拟合性能欠佳, 这是可以改善的。为进一步改善拟合质量, 可使用包含超前或滞后记忆交叉项的 PA 模型:

$$\begin{aligned}
 y_{\text{GMP}}(n) = & \sum_{k=0}^{K_a-1} \sum_{l=0}^{L_a-1} a_{kl} x(n-l) |x(n-l)|^k \\
 & + \sum_{k=1}^{K_b} \sum_{l=0}^{L_b-1} \sum_{m=1}^{M_b} b_{klm} x(n-l) |x(n-l-m)|^k \\
 & + \sum_{k=1}^{K_c} \sum_{l=0}^{L_c-1} \sum_{m=1}^{M_c} c_{klm} x(n-l) |x(n-l+m)|^k
 \end{aligned}$$

[1] 中也详细说明了此模型和相关拟合程序, 并在 `poweramp_helper.m` 脚本中作为白箱函数提供。该模型有助于更好地描述记忆效应, 同时仍可提供因果性响应。使用  $M=5$  的记忆深度以及  $L=7$  的多项式长度生成一个模型, 可将信号标准差大幅降低到 2.3833% (图 10)。同样, 黄色线条表示采集的 PA 输出数据的频谱, 蓝色线条表示输入采集的 PA 输入数据后建模的 PA 的频谱。

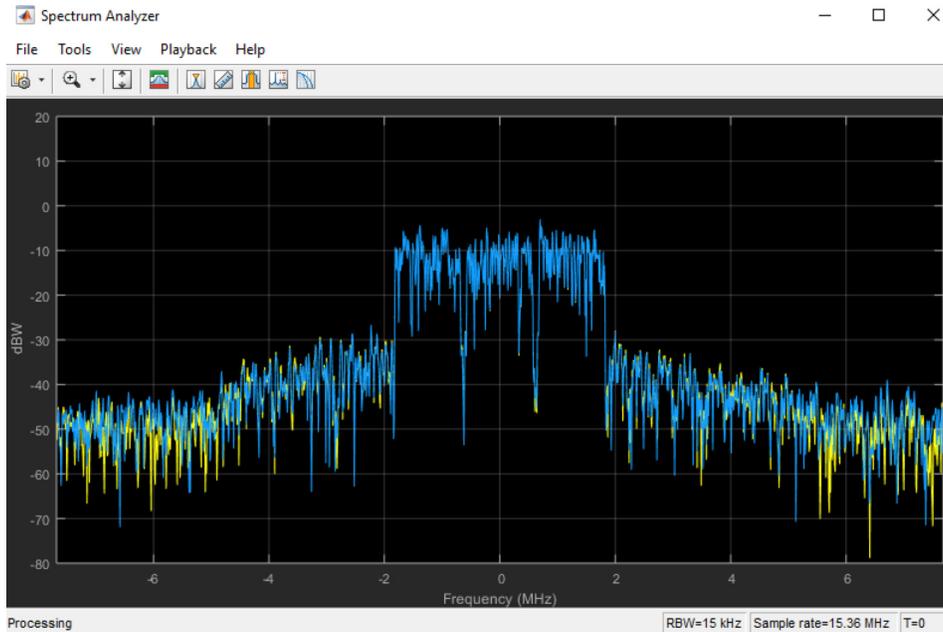


图 10: PA 的输出特征功率频谱, 显示了测得的特征 (黄色) 和使用含交叉项的记忆多项式模型 (记忆深度等于 5 且多项式阶数等于 7) 的拟合结果 (蓝色)。

在频域中，所预测信号的最大误差刚好是 -50 dB，如图 11 中所示。

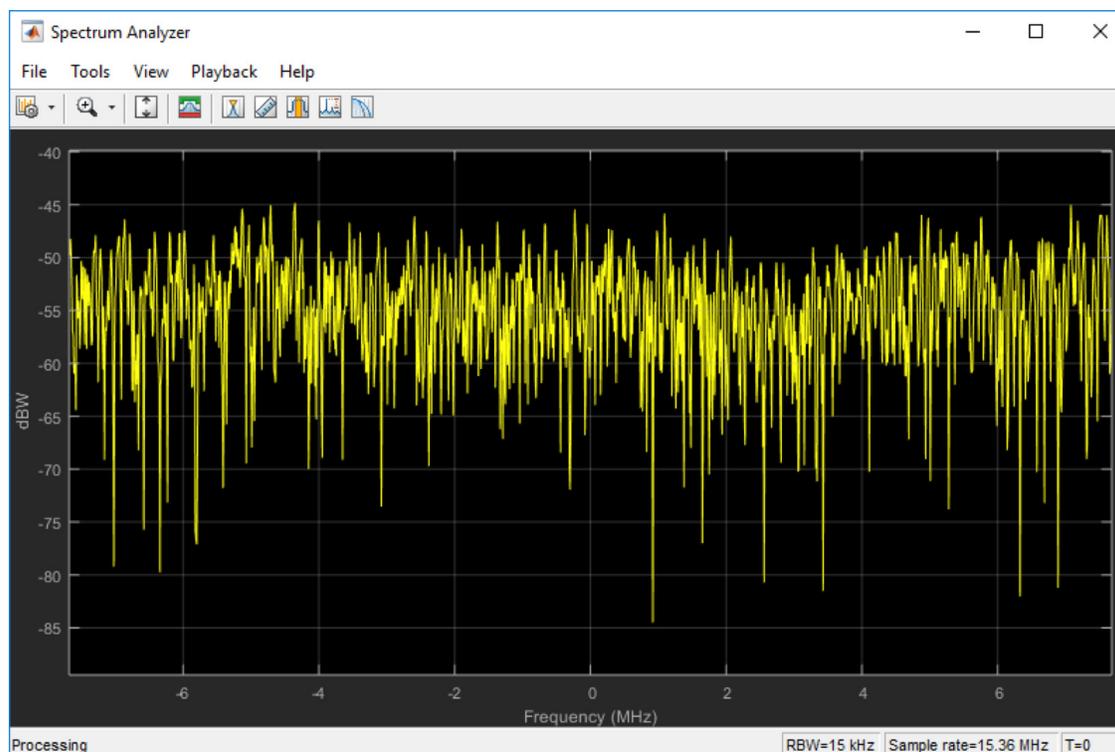


图 11: 错误信号的功率频谱图，表示记忆深度等于 5、多项式阶数等于 7 时，使用一个包含交叉项的记忆多项式模型测得的 PA 输出和预测的 PA 输出之间的差异。

因此，记忆交叉项可改善 PA 模型的质量，代价是会增加复杂性。由于此处使用的 MATLAB 代码已进行白箱处理，因此可以查阅和修改它。还可以使用相同的框架对完全不同的 PA 建模和拟合程序进行建模，例如那些基于 Doherty 模型的程序 [2,3]。

在这些示例中，使用的数据是通过 PA 传递波形收集的。为了创建良好的 PA 模型，此处使用的波形必须具有代表性。如果需要，可以在 MATLAB 中综合地创建这些波形。具体而言，对于 LTE、WLAN 和 5G 系统，LTE System Toolbox™ 和 WLAN System Toolbox™ 中的信号生成器可生成符合标准的波形。Instrument Control Toolbox™ 也可用于在基于实验室的测试和测量硬件间来回传递综合生成的或预先记录的波形，以针对自动化测试目的练习 PA。

## 设计用于改善发射器线性度的 DPD 算法

可将数字预失真 (DPD) 算法放在 PA 之前，以将发射的信号进行预失真处理。DPD 特征被设计成 PA 特征的反向，从而可生成一个总体呈线性的行为，因为这两个效应会相互抵消。因此，了解和描述 PA 的行为是 DPD 设计的关键部分。

不过, DPD 算法的实际设计和实现比逆转非线性特征复杂得多(图 12)。由于 PA 受记忆影响, DPD 算法必须能够自适应, 并且必须在闭合的反馈回路配置中进行运算, 才能补偿 PA 效应。此外, DPD 算法是在基带中实现的数字化算法, 而 PA 是在高频运行的射频组件。需要上变频和下变频阶段来使两个系统一起运行。与进行此操作相关的非理想特性将影响 DPD 行为。由于时序缺陷会使系统变得不稳定, 因此它们尤其重要。还必须考虑 PA 上天线的负载效应, 以及用于通过自适应方式计算 DPD 系数的 DPD 策略。

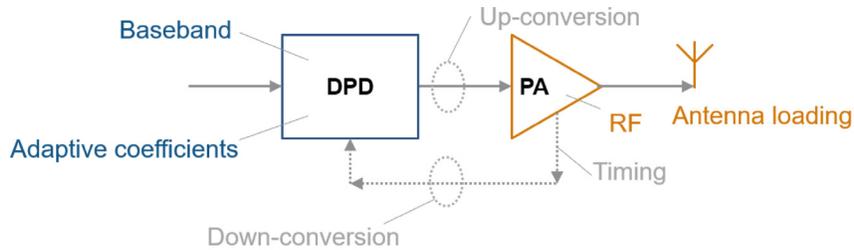


图 12: 包括功率放大器和数字化预失真算法的闭环系统图示。

确定适合的 PA 模型后, 可以开发 DPD 算法。对多个信号域 (射频、数字和模拟) 进行建模时, Simulink 可以很好地适应建模环境。不过, 有些模块包含了 MATLAB 代码, 用于对设计进行进一步说明。对基于 [4] 的 DPD 方法进行建模。此方法描述了由递归预测性错误方法 (RPEM) 实现的间接学习架构。顶层结构包含一个 Simulink 模块, 用于根据系数当前值和来自 PA 及另一个 Simulink 模块的当前输出, 通过自适应方式计算系数, 以将系数应用于输入信号后再反馈送到 PA (图 13)。

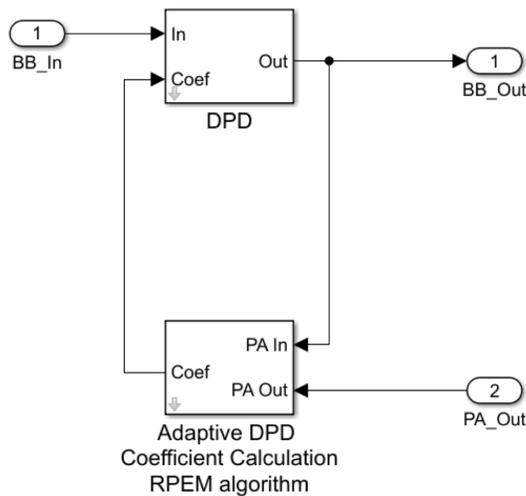


图 13: 由递归预测性错误方法 (RPEM) 实现的间接学习架构。底部模块根据系数当前值和 PA 的当前输出计算 DPD 系数; 顶部模块先应用这些系数并对输入信号进行预失真处理, 然后再反馈送到 PA。

系数计算模块包含 RPEM 内核，如图 14 所示。

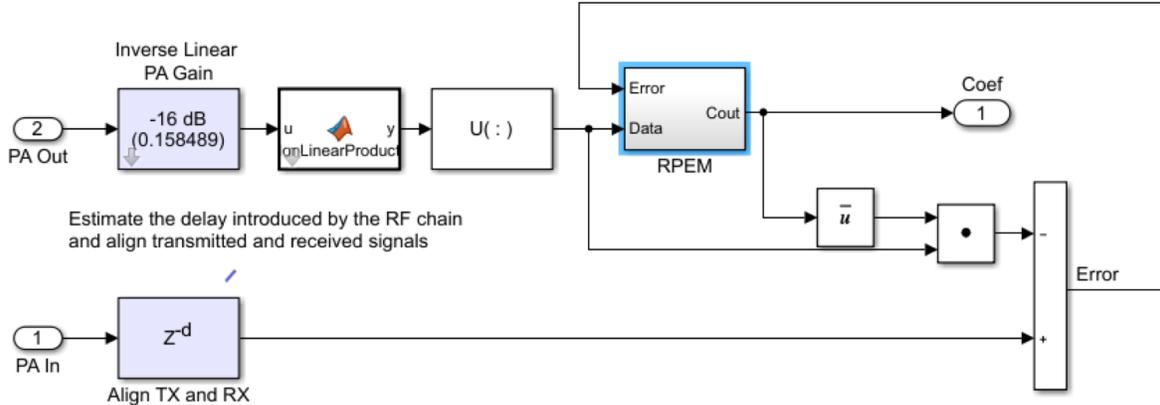


图 14: 用于以自适应方式计算 DPD 系数的 Simulink 模块图。

图 14 中的 RPEM 模块根据 [2] 计算系数，如图 15 所示。

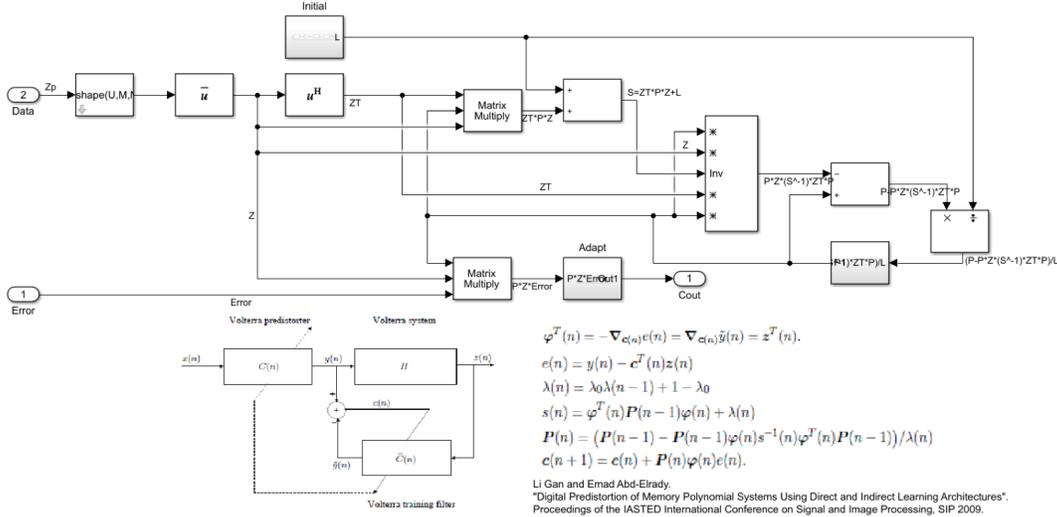


图 15: 用于按照 RPEM 算法以自适应方式计算 DPD 系数的 Simulink 模块图。通过一个注释来说明实现的计算。

RPEM 内核根据记忆长度 M 和多项式深度 K 计算系数，其计算策略与用于计算 PA 多项式系数的策略相似。M 和 K 的值越高，计算级别就越高。这通常会导致要满足更大的资源和延迟需求才能在真实硬件中实现它。

图 16 描述了应用计算出的系数对输入信号进行预失真处理的 DPD Simulink 模块。

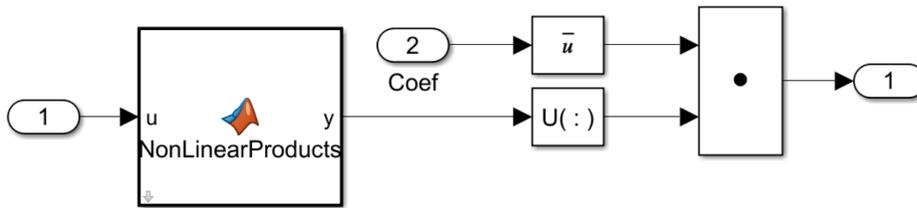


图 16: 用于对 PA 输入信号进行预失真处理的 DPD 算法的 Simulink 模块图。MATLAB 函数用于执行输入信号和 DPD 系数间的矩阵乘法。

此设计中使用了基本的 Simulink 模块。Simulink 可用于创建替代性策略，以通过相似框架计算系数并将系数应用于输入信号。

### 进行系统仿真以对 DPD 算法性能早期验证

现在，应在系统仿真模型中一同集成和测试 PA 和 DPD 算法模型。为此，使用 RF Blockset™ [5] 执行一个电路包络仿真。这样，便可实现快速射频仿真，同时也考虑到非线性效应、阻抗失配和时序缺陷 [6]。

图 17 显示了一个使用两个正弦波音调馈送的发射器的简单行为模型。

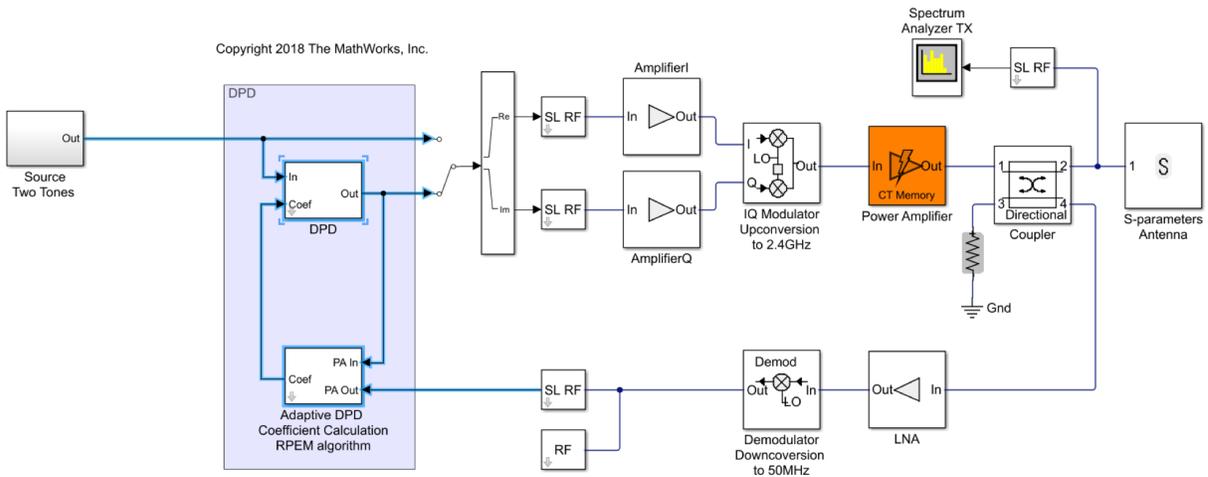


图 17: 射频发射器的简单行为模型、观察者接收器路径和使用双音调信号测试的自适应 DPD 算法的 Simulink 模块图。

发射器模型包含不同的放大阶段、正交调制器和使用以前计算的 PA 系数的 PA 模型（橙色）。PA 通过使用 S 参数描述的天线加载。耦合器用于“观察”PA 的输出，并将其降频转换为中频。发射器模型中的所有射频组件模块均来自 RF Blockset 库，并使用电路包络仿真。此外，组件模型包含射频缺陷，例如图 18 所示的输入放大器中的非线性度。

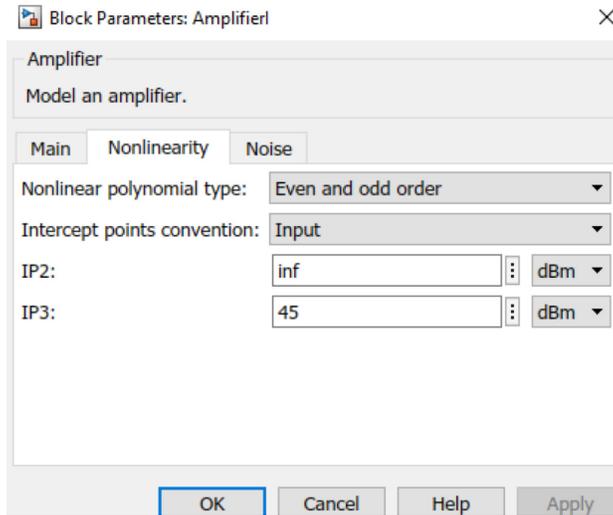


图 18: 非线性输入放大器的掩码参数。

S 参数天线模型可从以 VNA 为特征的真实天线捕捉，或者使用制造商提供的数据。还可以在 MATLAB 中使用 Antenna Toolbox™ 来构建具有代表性的天线模型，并提取将集成到此模型中的 S 参数 [7]。Antenna Toolbox 提供了完整的 EM 求解器，并使用矩量法来分析来自预置库的天线以及定制天线结构的性能。

周围测试平台将一个双音正弦波注入发射器，并使用频谱分析仪将来自 PA 的输出可视化并计算第三阶输出 IP 系数 (OIP3)。在启用 DPD 算法前，OIP3 值大约为 -36 dBm，其中第三阶音调为 -32 dBc，如图 19 所示。

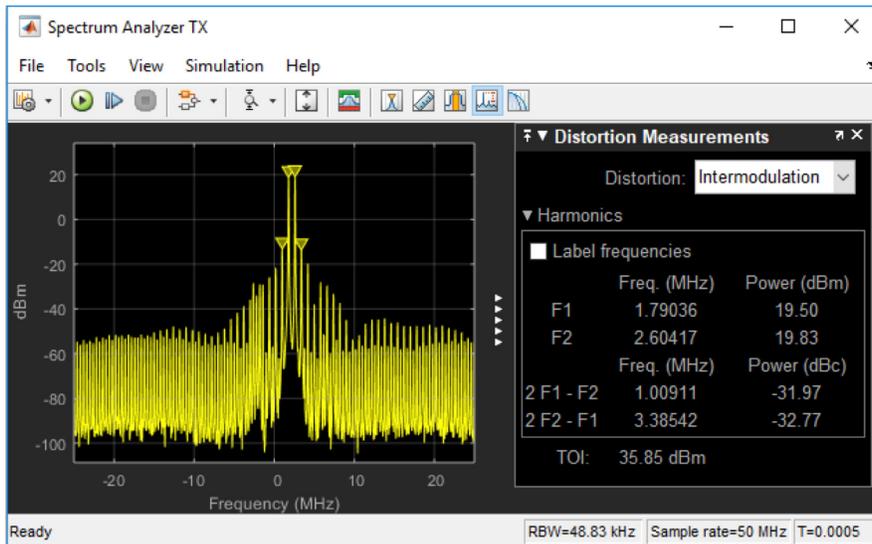


图 19: 使用双音激励测试的射频发射器输出信号的功率频谱。禁用 DPD 后，三阶交调点大约为 36 dBm。

不过, 在启用 DPD 算法后, OIP3 值下降到 -40 dBm, 其中第三阶音调位于 -41 dBc, 如图 20 所示。

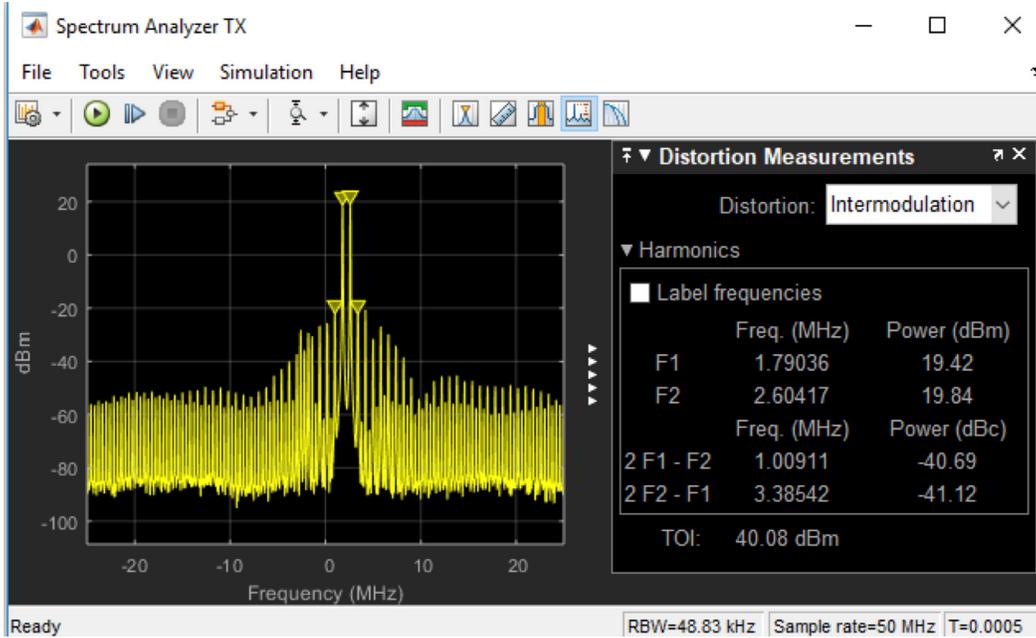


图 20: 使用双音激励测试的射频发射器输出信号的功率频谱。启用 DPD 后, 三阶交调点大约为 40 dBm。

可通过与 PA 建模练习相似的方式修改 DPD 算法的阶数和记忆长度, 并观察对系统级别 DPD 策略性能的影响。此仿真模型为不同 DPD 配置的快速迭代测试提供了便利, 帮助更轻松、快速地优化算法。它还为 DPD 方法的早期验证提供了环境, 甚至可能在建立 PA 的任何物理原型之前。

此模型还可使用调制的波形进行测试。在图 21 中, 使用 QAM 信号, 并通过衡量应用 DPD 算法之前和之后 EVM 的差异来评估 DPD 算法的效果。

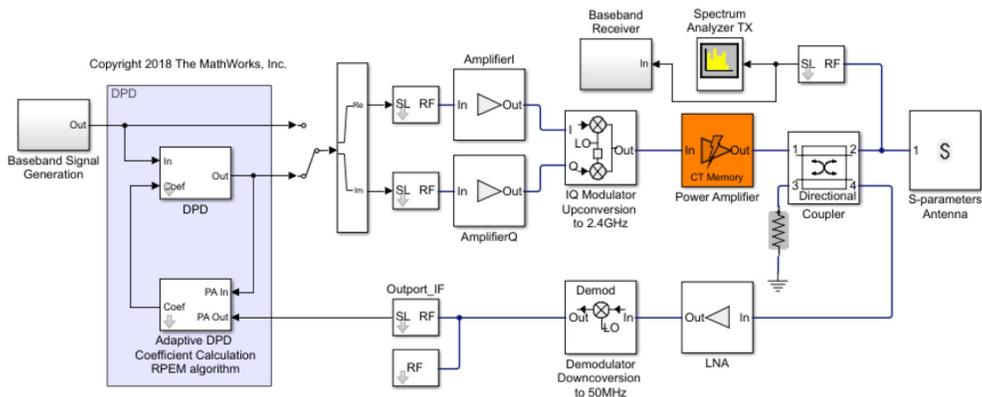


图 21: 射频发射器的简单行为模型、观察者接收器路径和使用 QAM 信号测试的自适应 DPD 算法的 Simulink 模块图。

启用 DPD 算法之前, EVM 的 RMS 值为 17.5% (图 22)。

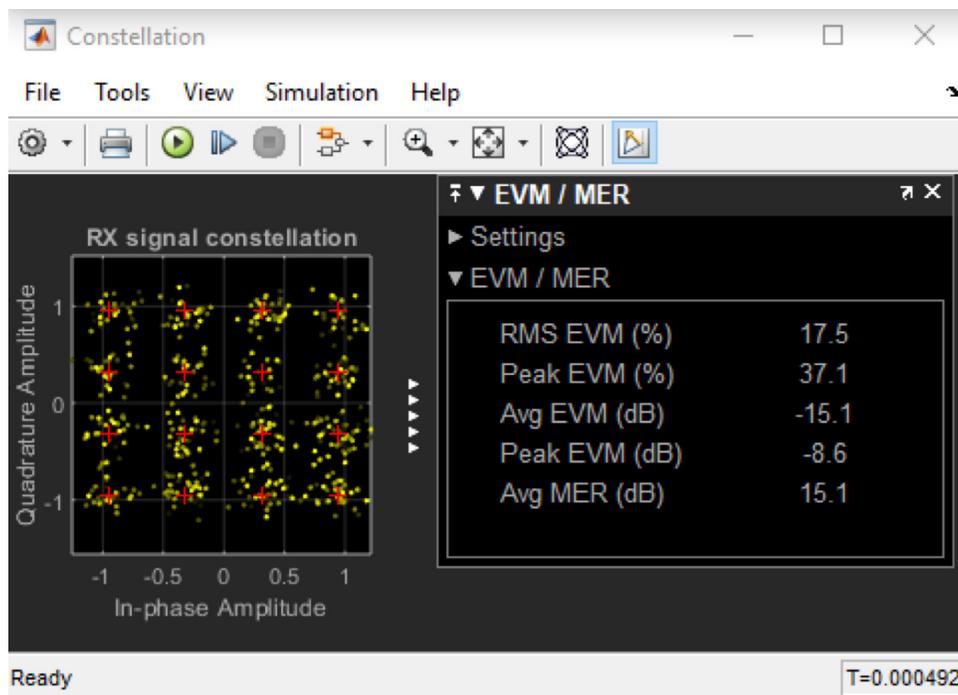


图 22: 使用 QAM 信号测试的射频发射器输出信号的星座图。禁用 DPD 后, EVM 为 17.5%。

启用 DPD 算法后, 该值下降到 4.0% (图 23)。

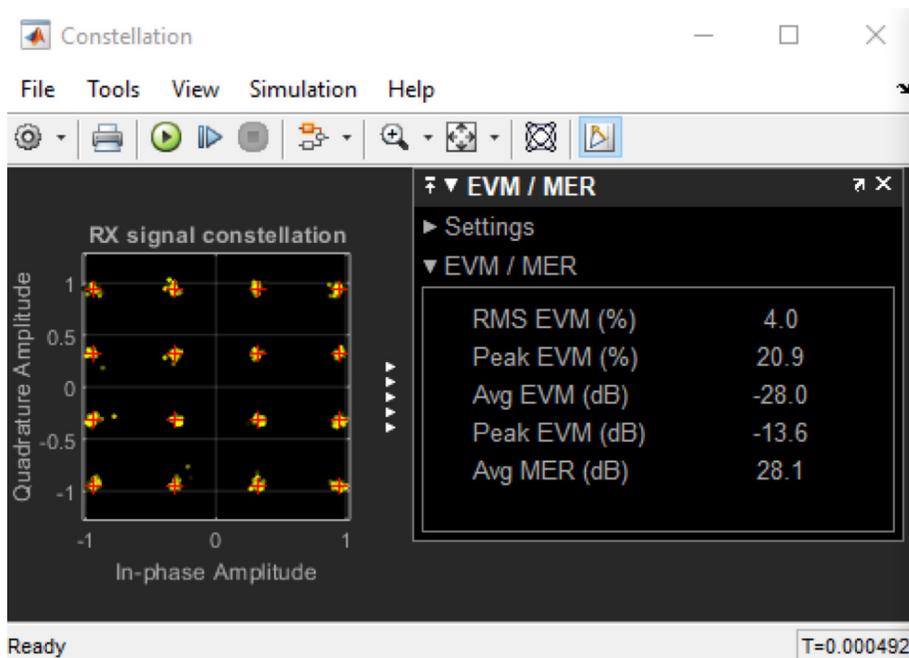


图 23: 使用 QAM 信号测试的射频发射器输出信号的星座图。启用 DPD 后, EVM 为 4.0%。

应用 DPD 算法后,星座图的各个点之间更紧凑。

在开发 DPD 算法并在仿真中进行测试后,下一步是使用真实 PA 测试它。具体做法是,将 DPD 算法保留在 Simulink 中,并让该算法与测试和测量硬件进行接口通信,以通过该 DPD 来驱动基带信号,输出到真实 PA,然后返回 Simulink 进行分析和 DPD 调整。这可通过 Instrument Control Toolbox [8] 来实现。在使用 Simulink 的华为,此方法已成为 DPD 设计工作流程的一部分。在 [9] 中,作者说明了使用 MATLAB 和 Simulink 进行 DPD 设计的优势,这些优势让他们能够:

- 执行设计的闭环仿真,其中包含模拟/射频和数字组件,例如用于射频功率放大器的数字预失真 (DPD)
- 在研发过程一开始,就可以使用一个无缝射频仪器接口快速开发灵活的高性能硬件开发平台
- 快速构建软件和硬件之间的自动验证平台
- 使用一个单一平台进行硬件开发,包括参考模型、定点转换和 C 和 RTL 代码自动生成
- 重复使用模型以进行浮点、定点和 RTL 代码的位真验证

## 使用硬件实现和验证 DPD 算法

在仿真中使用不同 DPD 策略进行实验后,下一步是在 Simulink 外部建立算法的原型。随着发射器带宽增加,DPD 算法必须以更快的速率运算,以实现实时性能并满足延迟需求。因此,FPGA 是常见的用于 DPD 算法测试和验证的原型设计架构。算法通常作为定点被部署到 FPGA,因此第一步是执行快速原型开发,将 DPD 算法从浮点转换为定点,并分析定点精度对算法性能的影响。

Fixed-Point Designer™ [10] 提供了一个引导式工作流程,协助将模型从浮点转换为定点。在部署到 FPGA 之前,可先在仿真中创建位真仿真模型并测试性能。这在具有反馈配置的模型(例如此处讨论的 DPD 模型)中尤其重要,因为使用有限精度算术运算会影响稳定性和锁定时间。值得注意的是,当算法位于 FPGA 上时,很难发现和克服上述问题。仿真有助于更轻松地在设计周期的早期诊断出这些问题。

要转换在上一部分的系统仿真中建模的 DPD 算法,可在 DPD 算法之前放置一个数据类型转换模块(图 24)。

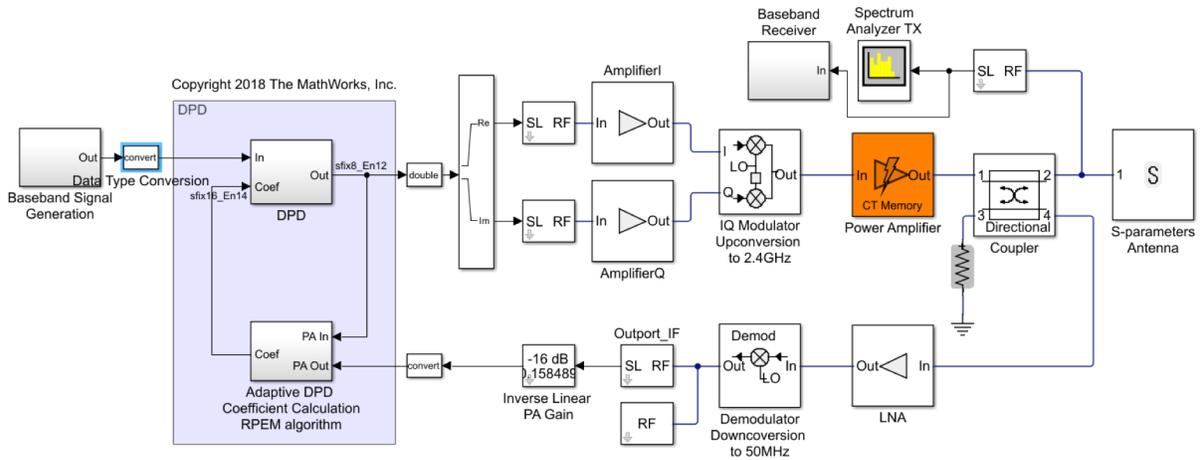


图 24: 射频发射器的简单行为模型、观察者接收器路径和使用有限精度算术运算的自适应 DPD 算法的 Simulink 模块图。

配置数据类型转换模块参数，将来自浮点的输入信号转换为 12 位有符号的信号，其分数长度为 16 位 (图 25)。

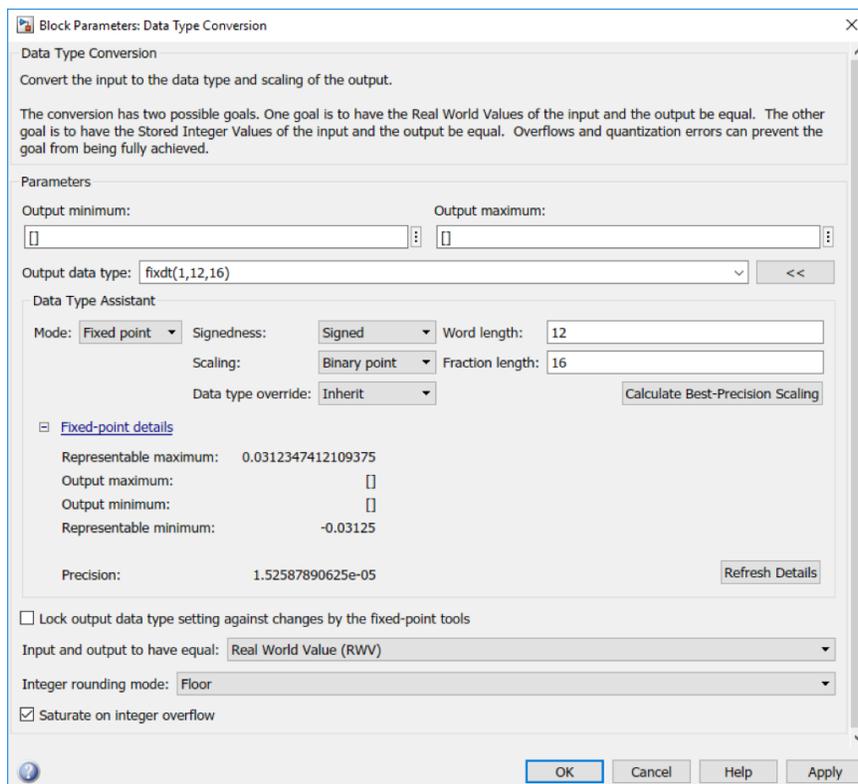


图 25: 数据类型转换模块的参数，用于将双浮点精度的输入信号转换为定点有限精度算术运算。

如图 25 所示, 这意味着可表示范围大约在  $\pm 0.03$  之间, 信号的可表示范围最接近  $1.5e-5$ 。使用向下取整运算符来实现此操作需要的取整。对于 FPGA 实现而言, 向下取整是一个非常好的运算符, 因为它只需要简单的逻辑即可实现。此外, 超出最大值和最小值限制的信号将饱和而不是循环。此行为实现起来较复杂, 但在有多个反馈回路时比较安全。在整个 DPD 设计中, 将采用这两个用于取整和饱和的策略。

递归预测性错误方法使用发射器输入和 PA 输出来计算系数。PA 输入和预失真的 PA 输出之间的误差在反馈回路中闭合, 以计算未来的 DPD 系数, 如图 26 所示。

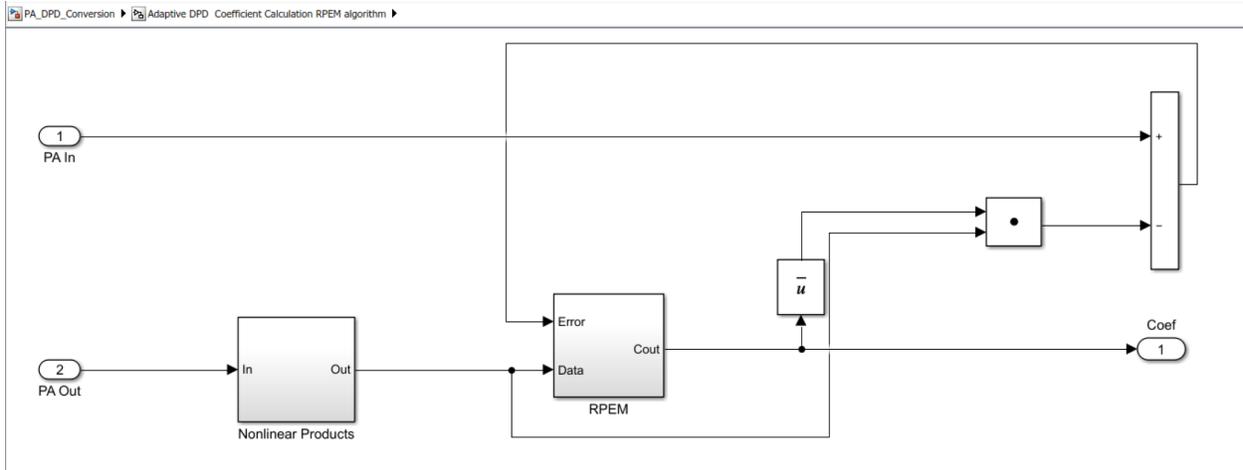


图 26: 通过有限精度算术运算计算 DPD 系数的 RPEM 算法的定点实现。

DPD 系数的计算须使用多矩阵乘法、倒置和反馈回路, 因此将此算法转换为定点并不容易。具体而言, 我们并不清楚, 在不需要使用过多资源的情况下, 每一个内部信号为了让算法正确高效地工作所需的字长和分数长度。Simulink 中的定点工具可用于帮助转换上述系数预估子系统。

理想情况下, 将所有信号线设置为改写为双精度浮点数。然后, 定点工具运行完整模型, 并收集和分析将在每条信号线转换的部分系统的数据, 如图 27 中的报告所示。

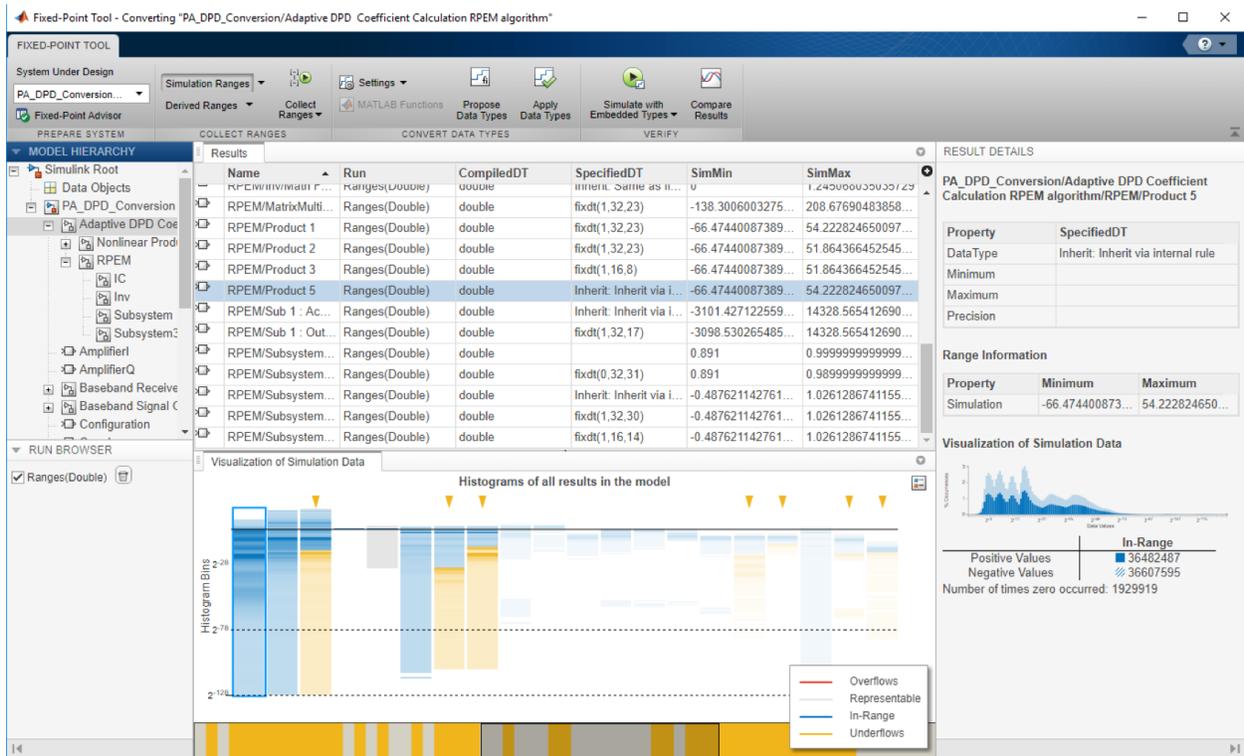


图 27: DPD 子系统中所有信号的仿真动态范围的定点工具报告。

正如模块“产品 5”的输出所示，仿真期间生成的输出范围大约在 -66.5 至 54 之间。另外，还显示了代表仿真运行期间数据值的直方图（图 28），让您了解信号线需要的动态范围。

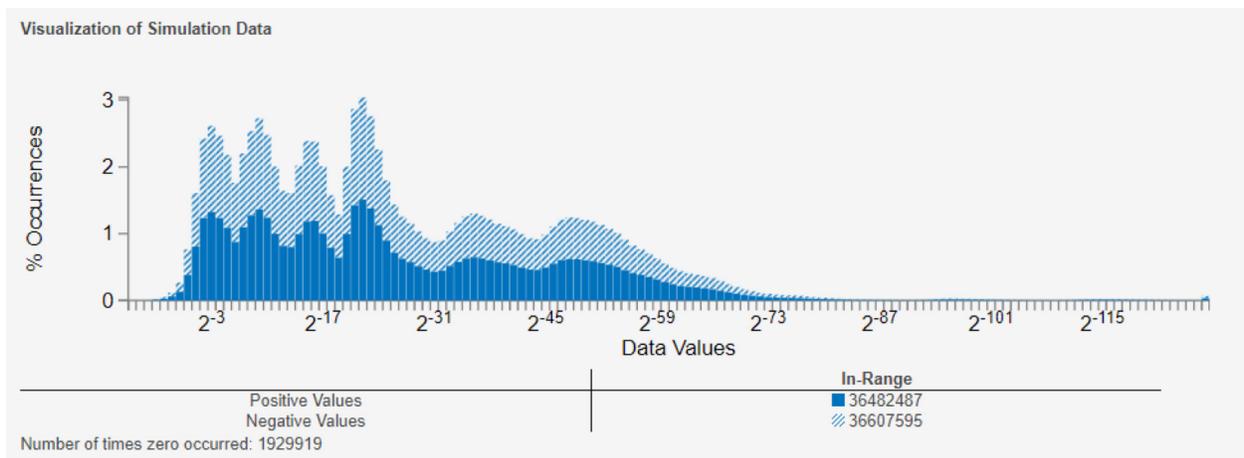


图 28: 用于计算 DPD 系数的某个内部信号的仿真动态范围的直方图。

此处，信号动态范围很大，输出有正有负。此分析有助于明智地选择适当的数据类型。

在给定固定字长或分数长度下，此工具可根据待设计系统中的每条信号线推荐适合的分数长度或字长。例如，您可以将字长配置为 16 位，并请求将计算的最佳分数长度，如图 29 所示。

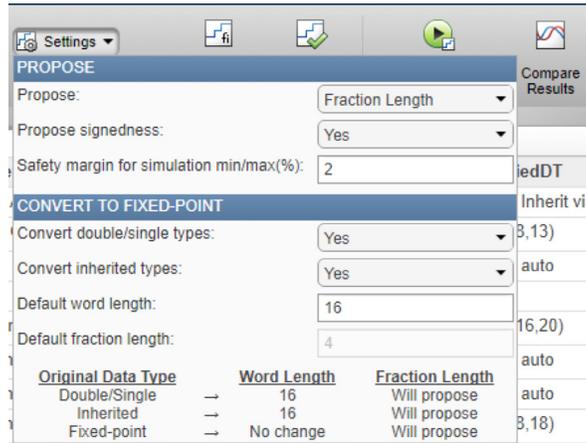


图 29: 在定点工具中自动计算分数长度。

根据此配置请求建议，将生成图 30 所示的建议数据类型。

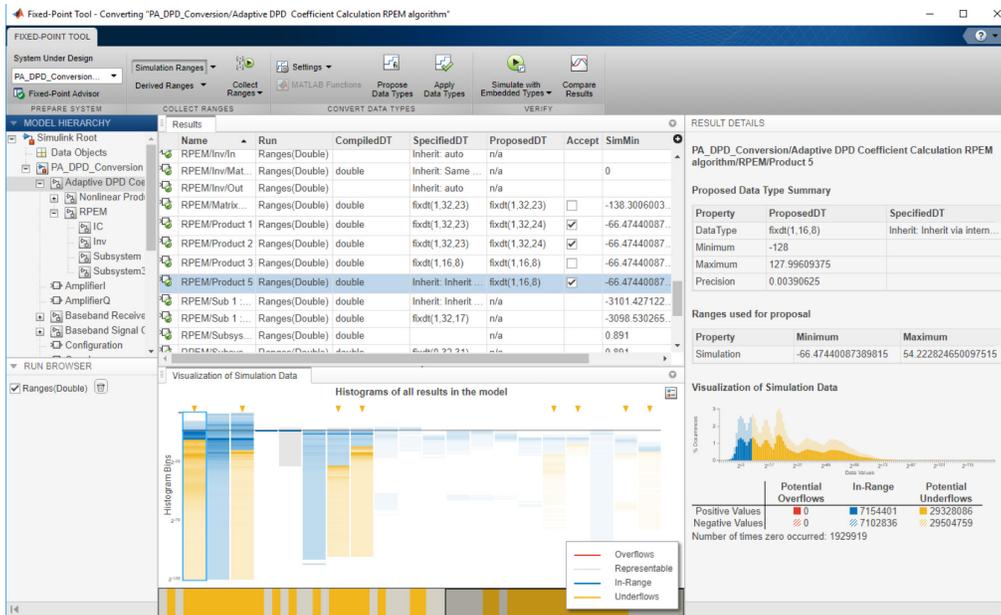


图 30: 自动计算指定字长的信号分数长度后的定点工具报告。

定点工具建议“产品 5”模块的输出应使用 8 分数位标记。这将生成可表示范围  $\pm 128$ ，精度大约在 0.004。可接受建议的所有数据类型，然后使用这些新类型重新运行仿真。在此情况下，EVM 现在从使用双精度实现的 4% 增加到 4.9% (图 31)。

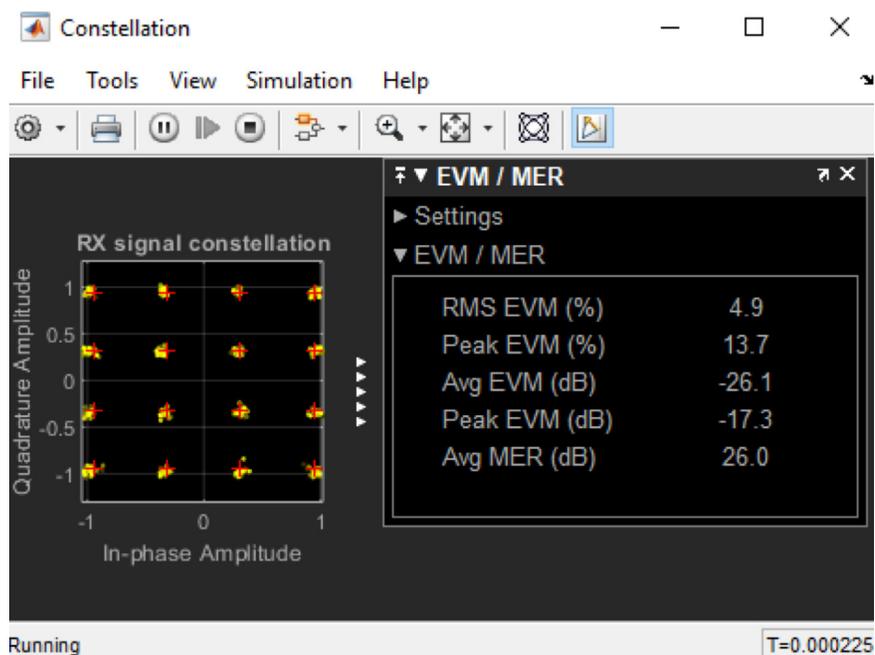


图 31: 通过 DPD 算法的有限精度算术运算, 使用 QAM 信号测试的射频发射器输出信号的星座图。

在针对“产品 5”模块抓取的仿真范围的直方图中, 发生了许多下溢, 如图 32 所示。

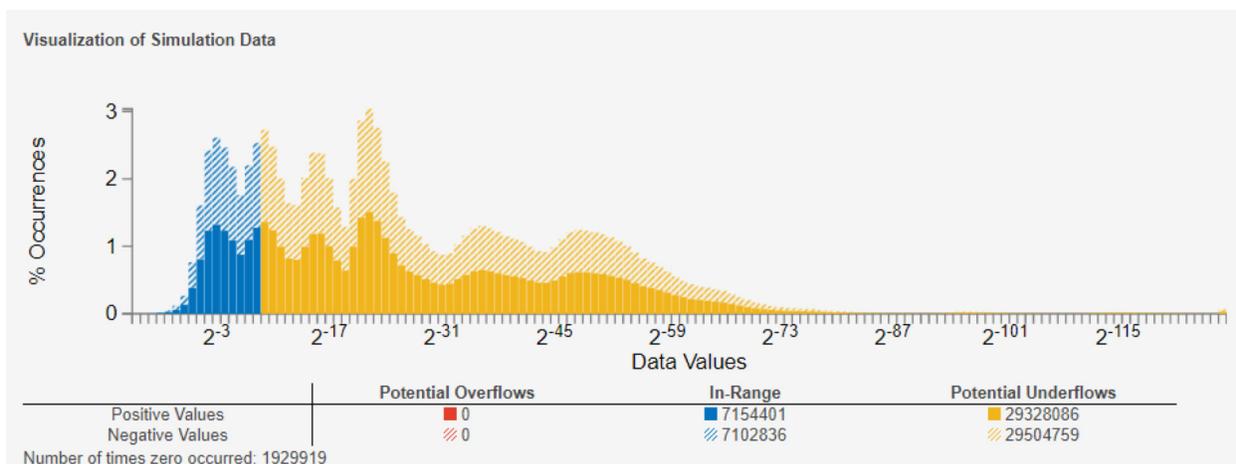


图 32: 用于计算 DPD 系数的某个内部信号的仿真动态范围直方图, 图中显示了 16 位字长覆盖的动态范围 (蓝色), 以及受下溢 (因缺乏足够的精度导致) 影响的数据 (黄色)。

可在定点工具和重新指定的系统中轻松修改定点配置。现在，字长已增加到 32 位，这样可更好地覆盖仿真数据。定点工具建议的“产品 5”模块输出的分数长度现在为 24，精度现在下降到接近 6e-8。图 33 显示了记录的数据点直方图更新后的覆盖范围。

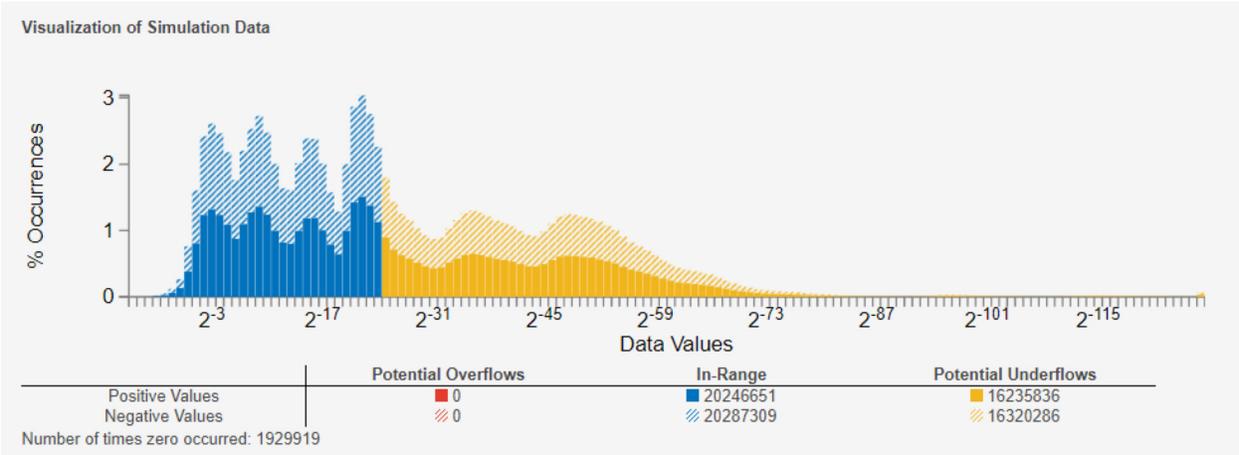


图 33: 用于计算 DPD 系数的某个内部信号的仿真动态范围直方图，图中显示了 32 位字长覆盖的动态范围（蓝色），以及受下溢（因缺乏足够的精度导致）影响的数据（黄色）。

此外，EVM 也下降到 4.3%。在设计信号线数据类型以完成整体设计时，可将单个信号线进行隔离和分析。在此过程中，可通过有选择性地关注关键模块和预估对系统性能的影响来转换整个模型。通过选择一个较大的字长，即使仅为一个内部模块这样选择，EVM 也会下降，且算法将更快地收敛。然后，必须针对提升的性能是否值得付出额外成本做出决策。

现在，用于计算 DPD 系数的 Simulink 子系统已转换为定点等效项。用于将 DPD 系数应用到输入信号的 Simulink 子系统也必须转换，因为它将部署到 FPGA。为此，作为替代方案，将使用新的和完全自动化的方法来完成上述辅助转换工作流程。此处，工具将使用一个优化方法为每条信号线建议最佳字长和分数长度，以符合用户定义的可接受转换误差。

此工作流程基于 MATLAB，但可驱动 Simulink 模型。首先，必须定义待设计系统中使用的可接受字长范围，以及在系统输出时可接受的最大相对误差和绝对误差。这些约束将馈送优化算法，该算法可建议适当的数据类型。对于 DPD 子系统，可接受的字长被指定为 8 到 32 位之间：

#### %% 创建选项对象

```
options = fxpOptimizationOptions;
options.AllowableWordLengths = 8:32;
```

将可接受的子系统输出的相对容差和绝对容差分别指定为  $1e-3$  和  $1e-4$ :

%% 设置在比较时将使用的行为容差

%% 原始基线模型的定点设计

```
addTolerance(options, sud, 1, 'RelTol', 1e-3);
```

```
addTolerance(options, sud, 1, 'AbsTol', 1e-4);
```

然后开始优化:

%% 将最大时间设置为 30 分钟

```
options.MaxTime = 60 * 30;
```

```
options.Verbosity = 'High';
```

%% 运行优化

```
result = fxpopt(model, sud, options);
```

%% 探索最佳定点设计

```
bestSolution = explore(result);
```

连续迭代允许为每条信号线建议分数长度和字长，直至达到所需的绝对容差和相对容差。对于 DPD 子系统，这需要大约 10 分钟的时间。转换后，仿真数据检查器会显示应用定点数据类型之前和之后的 DPD 模块的仿真输出，以及这两个信号之间的差异（图 34）。



图 34: 错误信号图表示使用浮点精度和定点精度计算的 DPD 模块的输出信号仿真结果之间的差异。

图 34 显示最终定点结果和浮点结果之间的差异在可接受的容差范围之内。按照此差异，可以使用定点工具对转换进行精确调校。

现在，DPD 系数计算子系统以及将这些系数应用于输入信号的 DPD 子系统均已转换为定点，通过仿真证实了这些子系统在系统级仿真中仍可以正常工作。借助 HDL Coder™，现在可以生成可读取、可合成的 VHDL® 或 Verilog® 代码，然后将它们部署到 FPGA。下面的例子将针对 DPD 系统进行说明。第一步是确保转换的 Simulink 子系统仅包含支持 HDL 代码生成的模块。这可通过运行 HDL 兼容性检查器实现，如图 35 所示。



图 35: DPD 模块的 HDL 兼容性检查器报告。

可通过右键单击子系统并选择“Generate HDL Code for Subsystem”（生成子系统的 HDL 代码）生成 HDL 代码。生成的代码显示在 HTML 代码生成报告中（图 36）。

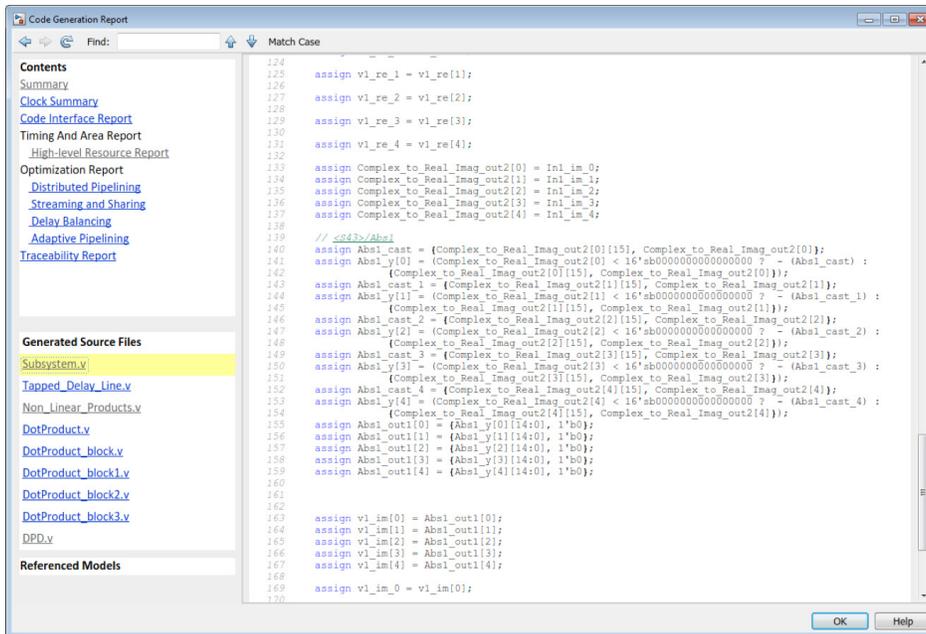


图 36: 包含为 DPD 模块自动生成的 HDL 代码的报告。

可以在模型中选择其他特定于 HDL 的选项，以优化资源利用率、速度和实现的设计领域。可修改配置选项，并对设计和生成的代码进行迭代。图 37 显示了资源利用率，在对设计进行迭代时可能需要关注该利用率，以确保它与使用的 FPGA 拟合。

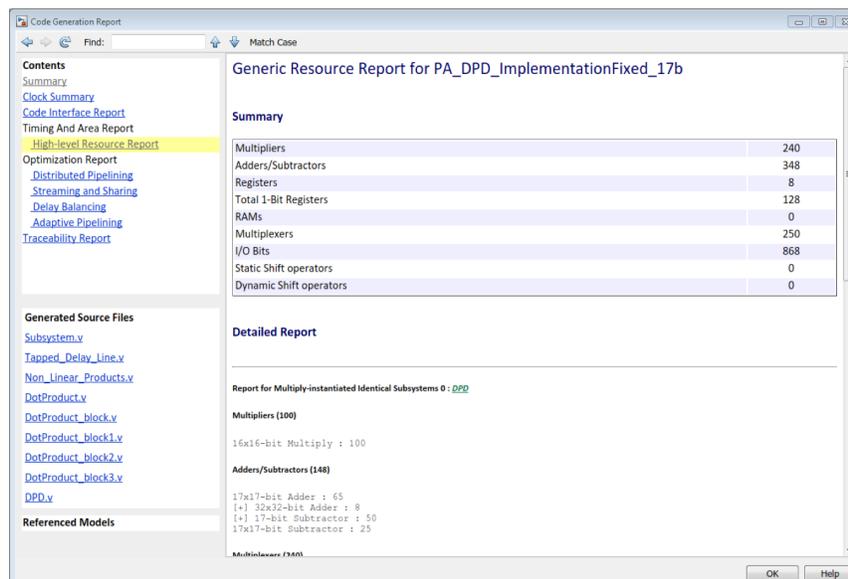


图 37: 为 DPD 模块自动生成 HDL 代码所需资源的报告。

生成的 HDL 代码可用于在 FPGA 上进行快速原型开发,也可改善生产质量,并可用于最终设计的 FPGA 或 ASIC 实现。可使用 HDL Verifier™ 通过生成协同仿真测试平台来验证生成的 HDL 代码。此产品与 Mentor Graphics® ModelSim® 或 Cadence® Incisive® 等第三方 HDL 仿真器进行交互,并重复使用 Simulink 测试平台来验证 HDL 仿真器中运行的 HDL 代码与 Simulink 模型具有相同行为。当转换的系统是闭环系统的一部分时,这一点特别有用。因为在仿真过程中,此交互允许以自适应方式更改将传递到 HDL 仿真器的输入,具体取决于以前时间步长中 HDL 仿真器的输出。如果预先记录了 HDL 测试平台激励,则这不可能实现。图 38 显示了自动生成的协同仿真测试平台,该平台通过 Simulink 运行。

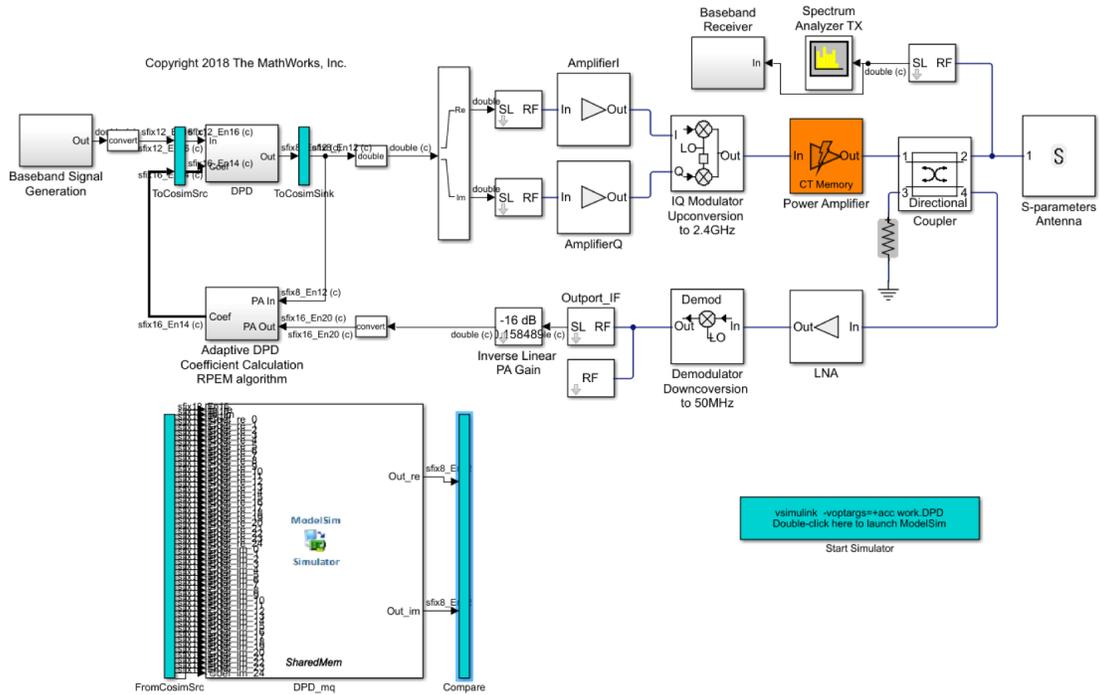


图 38: 用于将 DPD 算法的 HDL 实现与 Mentor Graphics ModelSim 进行协同仿真的 Simulink 测试平台。

点击蓝色框可启动 HDL 仿真器, 在此例中, 仿真器是 ModelSim。然后, 它自动载入 HDL 代码并配置仿真器以进行协同仿真。在上述操作完成后, 点击 Simulink 中的播放按钮开始将数据传递到 ModelSim 环境中, 并针对每个时间步长将输出发送回 Simulink。

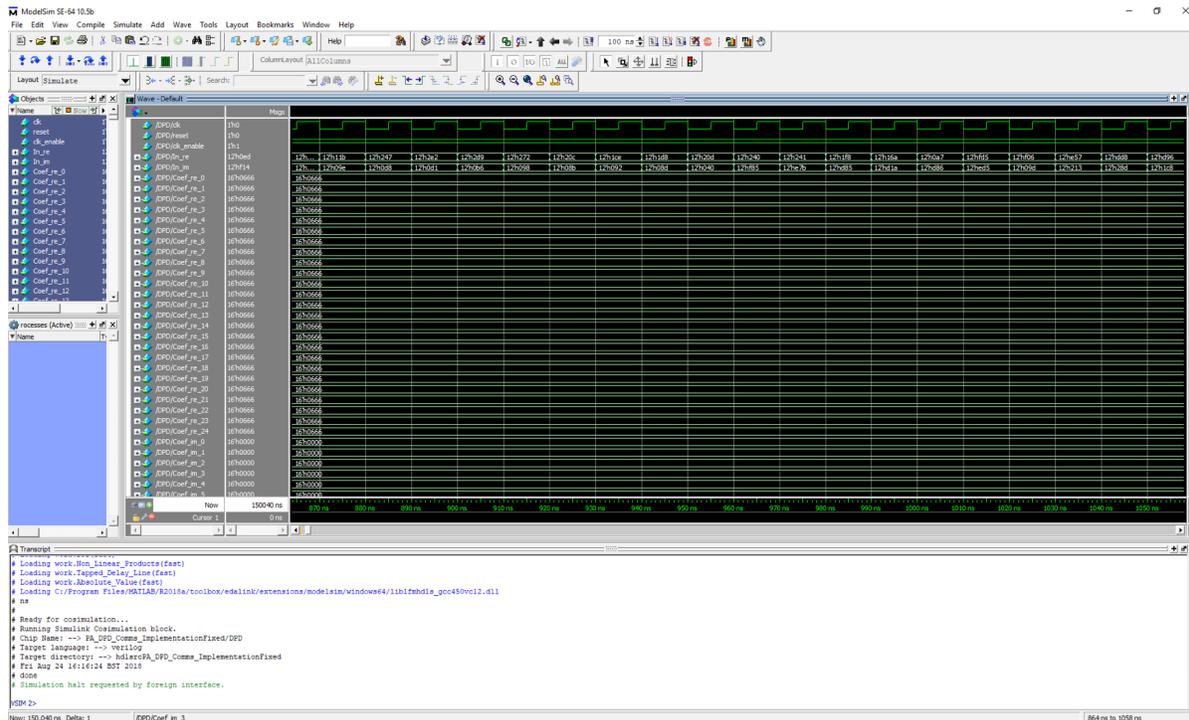


图 39: Mentor Graphics ModelSim 中显示的协同仿真测试平台的仿真结果。

在完成仿真后,可在 HDL 仿真器 (图 39) 或 Simulink 中分析结果。还可以将由 HDL Verifier 插入的示波器作为测试平台生成流程的一部分,在 Simulink 中比较 Simulink 模型和生成的 HDL 代码之间的差异。下面图 40 中的示波器输出显示了来自 DPD 子系统的真实信号输出和虚拟信号输出。在每种情况下,顶部图形显示 Simulink 模型输出,中间图形显示 ModelSim 中运行的 HDL 代码的输出,底部图形显示两者之间的差异。很明显,在这两种情况下,Simulink 模型和 HDL 代码具有同等效果。

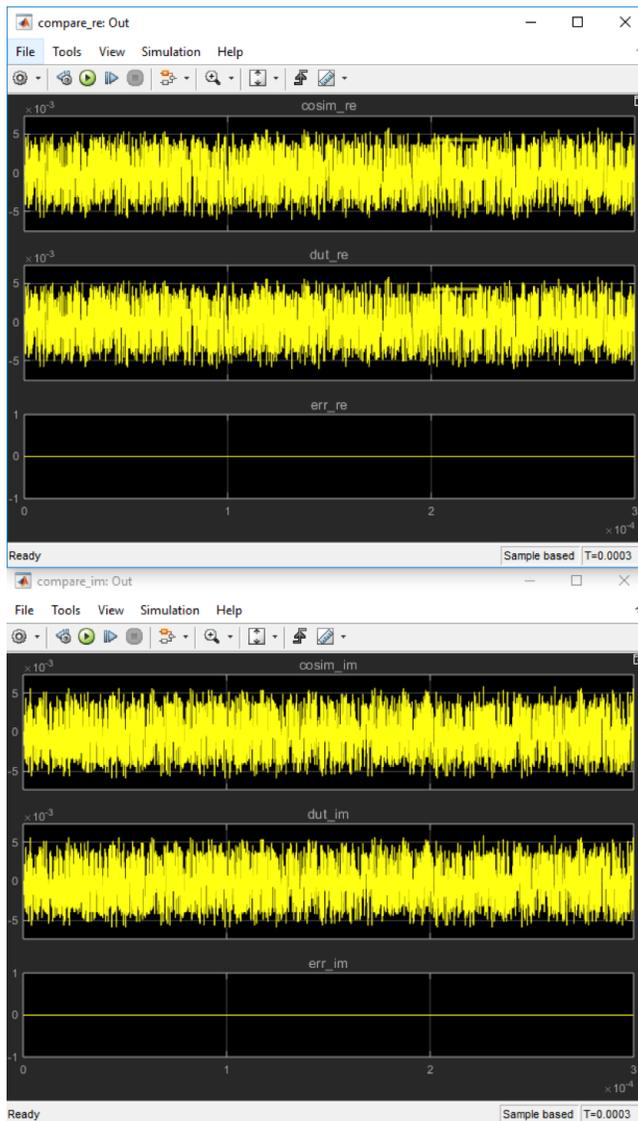


图 40: 参考 Simulink 模型的输出和 DPD 模块的 HDL 实现的比较图。顶部示波器绘制的是输出数据的实部，底部示波器绘制的是虚部。在每个示波器中，顶部踪迹是协同仿真输出，中间踪迹是 Simulink 输出，底部踪迹是代表这两者之间差异的错误信号。

通过遵循一个相似的工作流程，HDL 验证还可用于涉及 FPGA 的 HIL 仿真。这证实，使用 HDL 合成时，此算法行为是相同的。

总之，Simulink 模型可用于直接创建和验证实现。借助基于模型的设计，可使用相同环境来设计、测试、开发原型和实现您的算法。有关 HDL 工作流程的更多详情和示例及视频，可在 HDL Coder 和 HDL Verifier 产品页面 [11] 上找到。

## 结束语

在本文中，通过让您在早期阶段了解射频系统性能以及从仿真到实现的工作流程，我们讨论了 DPD 算法的自上而下设计流程，并介绍了如何使用仿真和基于模型的设计加快此流程并降低风险。该流程的第一步是使用真实的组件特征进行射频功率放大器建模。接着探讨了射频组件缺陷，以及通过拟合收集的射频功率放大器数据创建模型的策略。

接着将 PA 模型与 DPD 算法的模型一起放在闭环中，并使用 OIP3 和 EVM 指标执行系统级仿真和分析，以评估线性化技术的质量。

然后，将 DPD 算法从浮点实现转换为定点实现。接着，评估对量化、时序和其他系统缺陷的影响。最后，通过 DPD 算法生成了 HDL 代码，并介绍了用于优化和验证生成的 HDL 代码的技术。

可以使用 MATLAB 和 Simulink 来开发 DPD 算法，在整个开发过程中，可使用相同的模型来保持一致性、降低风险、加快设计和实现，并完善实验室测试和原型开发。您可以在 *RF Blockset* 产品页面上找到有关此内容的更多信息。

## 参考文献

- [1] Morgan, et.al., “A Generalized Memory Polynomial Model for Digital Predistortion of RF Power Amplifiers” (射频功率放大器数字预失真的广义记忆多项式模型) 《IEEE Transactions on Signal Processing》第 54 卷第 10 期, 2006 年 10 月。
- [2] A. Zhu, “Decomposed Vector Rotation-Based Behavioral Modelling for Digital Predistortion of RF Power Amplifiers” (用于射频功率放大器数字预失真的基于分解矢量旋转的行为建模), 《IEEE Transactions on Microwave Theory and Techniques》第 63 卷第 2 期, 2015 年 2 月。
- [3] O. Hammi, “Augmented Twin-Nonlinear Two-Box Behavioral Models for Multicarrier LTE Power Amplifiers” (针对多载波 LTE 功率放大器增强的孪生非线性双箱行为模型), The Scientific World Journal, 第 2014 卷, Article ID 762534。
- [4] L. Gan, “Adaptive Digital Predistortion of Nonlinear Systems” (非线性系统的自适应数字预失真), 奥地利格拉兹科技大学电气与信息工程系博士论文, 2009 年 4 月。
- [5] RF Blockset。 [mathworks.com/products/simrf.html](https://mathworks.com/products/simrf.html)。
- [6] 电路包络基础知识: Simulate High Frequency Components with RF Blockset (通过 RF Blockset 对高频组件进行仿真)。 [mathworks.com/help/simrf/gs/minimize-computations-for-rf-simulations.html](https://mathworks.com/help/simrf/gs/minimize-computations-for-rf-simulations.html)。
- [7] Antenna Toolbox。 [mathworks.com/products/antenna.html](https://mathworks.com/products/antenna.html)。
- [8] Instrument Control Toolbox。 [mathworks.com/products/instrument.html](https://mathworks.com/products/instrument.html)。
- [9] E. Zhu, “Developing a Radio Frequency System for Wireless at Huawei” (华为公司开发用于无线通信的射频系统) 2018 年 9 月。 [mathworks.com/content/dam/mathworks/case-study/huawei-customer-case-study-portrait.pdf](https://mathworks.com/content/dam/mathworks/case-study/huawei-customer-case-study-portrait.pdf)。
- [10] 定点设计优化。  
[mathworks.com/products/fixed-point-designer/features.html#data-type-exploration-and-optimization](https://mathworks.com/products/fixed-point-designer/features.html#data-type-exploration-and-optimization)。
- [11] HDL 代码生成和验证。 [mathworks.com/solutions/hdl-code-generation-verification.html](https://mathworks.com/solutions/hdl-code-generation-verification.html)。