

Pré-distorcedor digital descrito em linguagem VHDL e baseado em polinômio com memória.

Leonardo de Andrade Santos, Sibilla Batista da Luz Franca,
Eduardo Gonçalves de Lima

¹ Universidade Federal do Paraná, Curitiba, Brasil
leonard.andrade@ufpr.br

Resumo - A evolução da comunicação sem fio impulsionou aplicações como IoT, onde a eficiência energética é crucial para dispositivos móveis e estações de rádio. No entanto, essa eficiência compromete a linearidade dos amplificadores de potência (PA), afetando modulações de fase e amplitude. Sendo assim, uma alternativa para contornar esse obstáculo é a implementação de um (DPD) em cascata com um PA. Portanto, este trabalho propõe a implementação de um código em VHDL para processar as características de transferências inversa de um amplificador de potência em tempo real, minimizando não linearidades e consumo de energia. A modelagem foi feita inicialmente em software e implementada em FPGA (Field-Programmable Gate Array) para validação e análise de desempenho. Durante o desenvolvimento, testes e análises foram realizados com um polinômio de memória de segundo grau e um sinal de amostra memorizado, mostrando uma redução efetiva nas distorções do PA. Foi utilizada a métrica de Erro Quadrático Médio Normalizado (NMSE - Normalized Mean Squared Error) para quantificar a qualidade da modelagem implementada. O NMSE de -23,57 dB mostrou eficiência na redução de distorções. Assim a etapa seguinte consistiu na modelagem do PA em vírgula fixa, a fim de atingir o NMSE mais próximo do atingido em vírgula flutuante com a menor resolução em bits, visando alcançar o melhor desempenho em hardware. A modelagem em vírgula fixa com 8 bits aprimorou o desempenho. Por fim, o modelo foi implementado em hardware na FPGA Virtex5 XC5VLX50T, utilizando um total de 150 registradores, 692 LUTs e 4 unidades DSP48E, operando a uma frequência de 61,5 MHz.

I. INTRODUÇÃO

A evolução dos sistemas de comunicação móveis, impulsionada pela crescente demanda por comunicações mais rápidas e eficientes, tem levado à implementação de uma variedade de serviços, incluindo aplicações multimídia, desenvolvimento web e aplicações IoT [1]. No entanto, essa evolução também trouxe desafios significativos, como a necessidade de melhorar a eficiência energética, tanto para dispositivos móveis, visando aumentar a autonomia da bateria, quanto para estações de rádio base, visando reduzir o consumo de energia devido às perdas de calor. Para atender a essas demandas, estratégias de modulação que alteram tanto a fase quanto a amplitude de ondas portadoras em radiofrequência se tornaram essenciais [2]. Além disso, a modulação na amplitude requer

linearidade na transmissão para evitar erros e interferências na comunicação entre usuários vizinhos [3]. Essa complexa tarefa recai sobre o projetista do PARF (Amplificador de Potência de Rádio Frequência), que enfrenta o desafio de desenvolver um hardware eficiente em termos energéticos e linear ao mesmo tempo, uma vez que esses dois objetivos podem entrar em conflito [4]. Uma solução para contornar esse desafio é a implementação de um pré-distorcedor de Sinais Digital em Banda Base, que visa compensar a distorção causada pelo PARF [3]. O DPD (Pre-distorcedor Digital) é conectado em cascata ao PARF e requer um modelo de alta precisão e baixa complexidade computacional para representar as características de transferência direta e inversa do PARF. Existem duas abordagens para modelar o PARF: modelos físicos, que são detalhados e computacionalmente complexos, e modelos empíricos, que se baseiam em medições de entrada e saída do PARF, com menor complexidade computacional, mas com uma possível diminuição da precisão. Devido às exigências rigorosas de frequência de operação, a paralelização das operações torna-se essencial, e as FPGAs (Matriz de Portas Programáveis em Campo) emergem como uma alternativa viável para a implementação de circuitos pré-distorcedores [5]. As FPGAs são dispositivos lógicos programáveis que permitem a reconfiguração física de componentes de eletrônica digital, acelerando processos e suportando operações paralelas e sequenciais. Neste contexto, este projeto foi planejado com o objetivo geral de construir e validar um código em linguagem de descrição de hardware capaz de processar, em tempo real, as características de transferência inversa de um amplificador, reproduzindo suas não linearidades. Além disso, o código deve ocupar a quantidade mínima de recursos lógicos do circuito digital que irá sintetizá-lo, visando à redução do consumo de energia. Para atingir esse objetivo, o trabalho foi desenvolvido com base nos seguintes objetivos específicos: modelar com precisão o amplificador de potência (PA) em software; modelar o pré-distorcedor digital (DPD) em software a partir da modelagem do PA e implementar o DPD em hardware utilizando uma linguagem de descrição de hardware (HDL).

II. MODELAGEM MATEMÁTICA

A. Séries de Volterra

Segundo [6] série de Volterra pode ser vista como uma extensão multidimensional da série de Taylor para sistemas dinâmicos. A modelagem começa com a representação do sis-

tema através de uma série infinita de integrais convolucionais, onde cada termo da série corresponde a uma ordem de não linearidade e memória.

A saída $y(t)$ de um sistema pode ser expressa pela equação 1:

$$y(t) = h_0 + \sum_{n=1}^{\infty} \int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} h_n(\tau_1, \tau_2, \dots, \tau_n) \prod_{i=1}^n x(t - \tau_i) d\tau_i \quad (1)$$

onde h_n são os núcleos de Volterra, que caracterizam a resposta do sistema para a n -ésima ordem de não linearidade e $x(t)$ é a entrada do sistema.

Os núcleos de Volterra h_n são funções de várias variáveis que capturam a dinâmica do sistema em diferentes ordens. Para a maioria das aplicações práticas, a série é truncada para incluir apenas um número finito de termos, já que a identificação de todos os núcleos de uma série infinita é impraticável.

B. Polinômio de memória

Um modelo simples, utilizado na modelagem comportamental simplificada das séries de Volterra considerando apenas componentes unidimensionais¹ é o MP, que é um modelo compacto, de baixo custo computacional e linear em seus parâmetros. O MP gera baixo erro quando aplicado a PAs que apresentam pouco efeito de memória. O DPD e pós distorção apresentam característica inversa a do PA [7], portanto o mesmo modelo pode ser utilizado. A equação 2 apresenta o MP conforme é ilustrado por [7]:

$$y(n) = \sum_{p=1}^P \sum_{m=0}^M h_{p,m} x(n-m) |x(n-m)|^{p-1} \quad (2)$$

Como a proposta do trabalho é a implementação em hardware desse modelo, torna-se necessário paralelizar operações aritméticas de forma a alcançar altas taxas de operação. Nesse contexto, as FPGAs apresentam-se como uma alternativa viável para a implementação de circuitos pré-distorcedores.

III. IMPLEMENTAÇÃO EM SOFTWARE

Nesta etapa, foi realizada a implementação do modelo DPD em software, utilizando a linguagem de programação Python. Esta linguagem é amigável e amplamente difundida na comunidade acadêmica. Para essa modelagem, foram coletados sinais de entrada e saída de um amplificador de potência classe AB, que utiliza um HEMT fabricado com tecnologia GaN. O amplificador foi excitado por um sinal portador de frequência de 900 MHz, modulado por um sinal de envelope WCDMA 3GPP com aproximadamente 3,84 MHz de largura de banda. Os dados de entrada e saída do amplificador de potência foram medidos usando um VSA Rohde & Schwarz FSQ com uma taxa de amostragem de 61,44 MHz, conforme disponível em [8].

Em seguida, realizou-se o cálculo da estimativa do sinal utilizando números com vírgula fixa. Para verificar a precisão dessa estimativa em relação ao sinal original, calculou-se

¹Cada termo do somatório é composto por amostras no mesmo instante, por exemplo: $x(n)|x(n)|$, $x(n-1)|x(n-1)|$; termos bidimensionais são compostos por amostras em instantes de tempos distintos, como por exemplo: $x(n)|x(n-1)|$

o NMSE. Para essa validação, os dados foram inicialmente divididos em conjuntos de extração e validação, que são dados utilizados para fazer o cálculo do modelo e para fazer a validação do modelo calculado, respectivamente. A matriz de regressão foi calculada com os dados de extração. Esse cálculo é essencial para a extração dos coeficientes do polinômio de memória. Após a extração dos coeficientes, calculou-se o modelo do PA, que foi então validado com os dados reservados para isso. O NMSE obtido para um polinômio de 2º grau com uma amostra memorizada foi de -23,57 dB.

Em seguida, o algoritmo foi ajustado para operar com números em vírgula fixa afim de atingir a menor resolução possível, buscando o menor NMSE simulado. Por se tratar de um cálculo em vírgula fixa, fez-se necessário uma readequação do resultado obtido entre cada multiplicação de forma a manter a resolução inicial.

IV. IMPLEMENTAÇÃO EM FPGA

Essa etapa consiste na implementação do DPD em FPGA. Para isso, é necessário realizar paralelizações nas operações aritméticas. A Figura 1 ilustra como esse processo está dividido entre cada ciclo de clock. A cada ciclo, duas operações são realizadas em paralelo: o sinal atual é elevado ao quadrado e registrado, enquanto ocorre o somatório do produto entre os sinais do mesmo instante de tempo e seus respectivos coeficientes. Esse processo ocorre P vezes para os P graus do polinômio de memória. Portanto, a saída do DPD é incompleta para os primeiros P períodos de clock, pois, nesses primeiros ciclos, realiza-se o cálculo com base em entradas de sinais anteriores que ainda não ocorreram, resultando em uma saída incompleta.

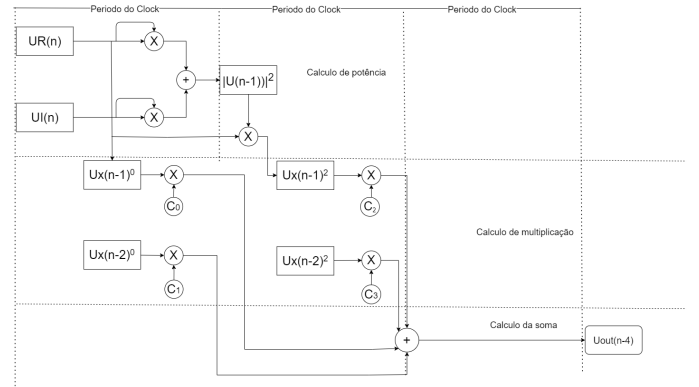


Figura 1: Processo de cálculo da saída

V. RESULTADOS

A. Modelagem do PA

Para fazer a modelagem em software foi utilizada a linguagem de programação Python. Para isso, separou-se os dados citados na seção III, em dados de extração e dados de validação, os quais são utilizados para extração dos coeficientes do modelo do MP e validação do modelo encontrado, respectivamente. Para fazer a validação do modelo utilizou-se a métrica do NMSE, que consiste em calcular o erro quadrático médio do valor medido pelo VSA (Analisador de Sinal Vetorial)

para o valor calculado pelo modelo. Portanto, quanto menor o NMSE mais fiel é o modelo do PA. Nesta etapa obteve-se um NMSE de -23.57 dB, para cálculos em vírgula flutuante, onde o resultado está presente no gráfico da figura 2.

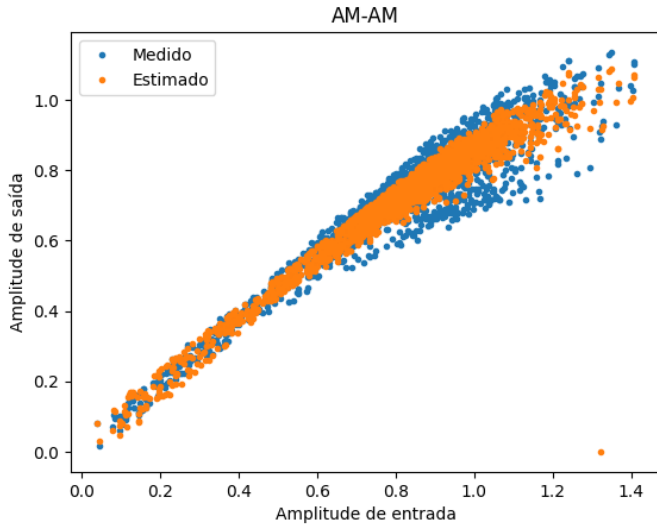


Figura 2: Modelo do PA em vírgula flutuante

B. Definição do número de bits

Após concluída a modelagem matemática, realizou-se a modelagem do PA para então ser feito o levantamento da quantidade de bits necessários para a implementação do DPD em hardware minimizando os erros de quantização. Para isso foi necessário refazer a extração dos coeficientes, mas desta vez com os dados normalizados para valores de 0 a 2^{bits} . O resultado desse levantamento está presente no gráfico na figura 3.

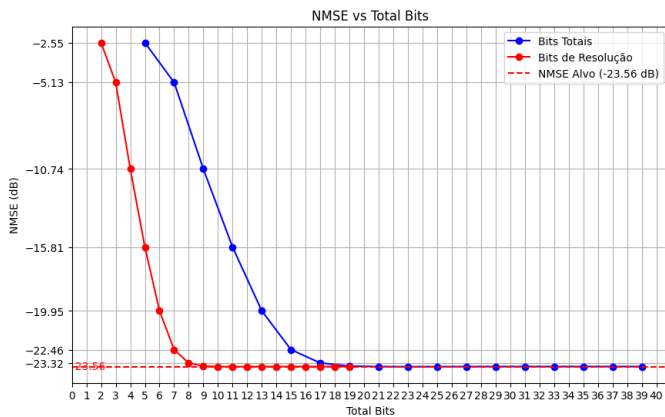


Figura 3: Gráfico Número de bits x NMSE

Neste gráfico observa-se duas curvas, a curva em azul apresenta a quantidade total de bits contando com os bits de overflow necessárias para as operações de multiplicação, enquanto a curva em vermelho representa a quantidade de bits de resolução do sinal. Analisando este gráfico observou-se que não existem ganhos significativos no erro a partir de 8 bits, portanto foi feita a modelagem do PA utilizando uma re-

solução de 8 bits. O resultado alcançado está ilustrado pela figura 4.

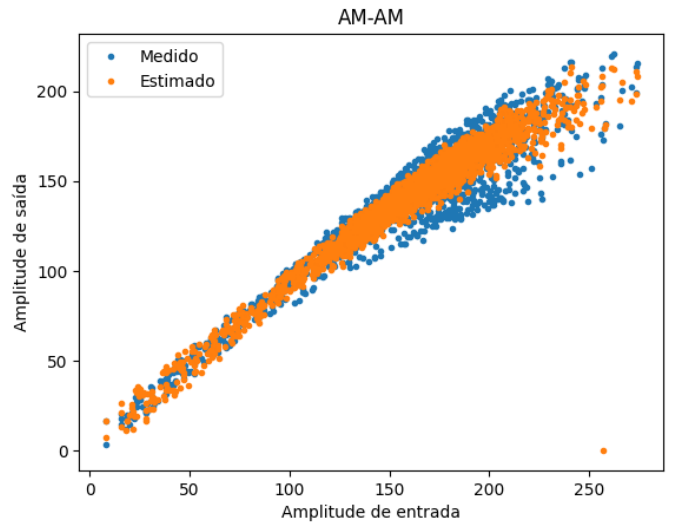


Figura 4: Modelo do PA em vírgula fixa

C. Modelagem do DPD

A partir dos resultados obtidos foi possível fazer a modelagem do DPD, para isso foi feito o mesmo processo de modelagem do PA, porém para alcançar a característica de transferência inversa do PA foi invertido a ordem dos dados de entrada e saída para extração dos coeficientes do DPD. O resultado desta modelagem está ilustrado pela figura 5 a seguir.

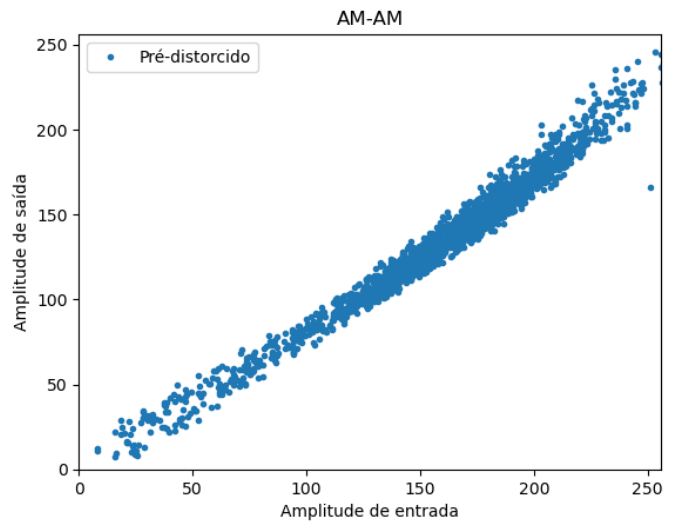


Figura 5: Modelo do DPD em vírgula fixa

D. Implementação do DPD em FPGA

E por fim foi implementado o código em VHDL para FPGA. Para que essa arquitetura de hardware apresentasse uma boa performance, todas as operações aritméticas (soma e multiplicação) são realizadas de forma síncrona. Então foi necessário dividir cada uma em processos distintos. A saída de um

processo alimenta um *buffer*, que serve como entrada para o próximo processo. A Figura 6 ilustra essa arquitetura de maneira simplificada.

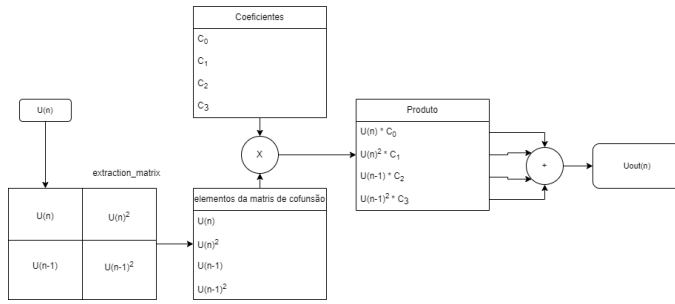


Figura 6: Processo de cálculo da saída

O resultado dessa implementação foi simulado em uma FPGA Virtex5 XC5VLX50T, utilizando um total de 150 registradores, 692 LUTs e 4 DSP48Es, operando a uma frequência de 61,44 MHz. A Figura 7 apresenta o resultado dessa implementação, representado pelos "x" em vermelho, em comparação com o resultado da simulação em Python, indicado pelos "•" em azul.

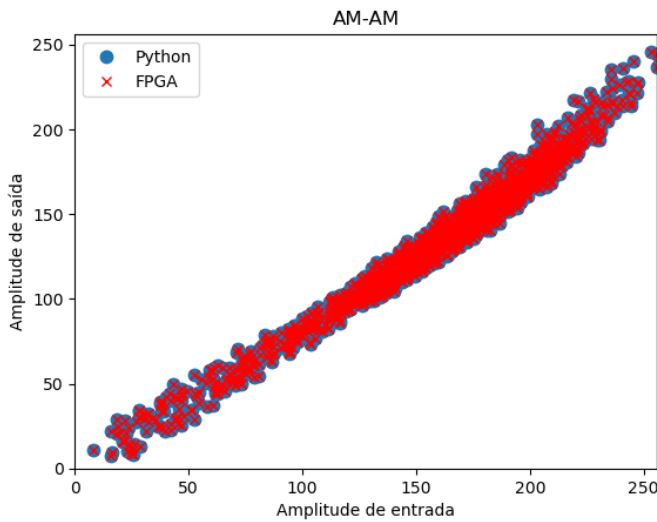


Figura 7: Processo de cálculo da saída

VI. CONCLUSÃO

A evolução dos sistemas de comunicação sem fio tem promovido a implementação de diversos serviços móveis, tornando essencial que esses sistemas operem com máxima eficiência. Nesse cenário, a implementação de um DPD em cascata com o PA surge como uma alternativa de baixo custo e interessante para melhorar o desempenho desses sistemas.

Assim, validou-se a implementação de um código em linguagem de descrição de hardware capaz de processar em tempo real as características de transferência de um amplificador, reproduzindo suas não linearidades e efeitos de memória. Além disso, o código foi projetado para ocupar o mínimo possível de recursos lógicos do circuito digital que o sintetizará, visando reduzir o consumo de energia.

O desenvolvimento deste trabalho seguiu a seguinte

seqüência: inicialmente, foi realizada a modelagem do PA em vírgula flutuante, utilizando o método do Polinômio de Memória (MP) com um polinômio de 2º grau e uma amostra de memória. A validação dessa modelagem foi feita com a métrica NMSE, obtendo-se um valor de -23,57 dB. Em seguida, foi realizada a otimização da quantidade de células lógicas no processo, limitando a resolução para 8 bits, após constatar que valores superiores não apresentavam melhoria significativa no NMSE. Essa resolução foi então utilizada para a amostragem dos sinais. Posteriormente, a modelagem do DPD foi feita em software, apresentando comportamento inverso ao do PA e atendendo aos requisitos do projeto. A etapa seguinte iniciou a implementação do modelo em hardware, e os próximos passos incluem a validação e otimização do circuito para alcançar a melhor performance possível.

REFERÊNCIAS

- [1] Elton John, "Modelagem comportamental de amplificadores de potência de radiofrequência usando termos unidimensionais e bidimensionais de séries de Volterra", 2016.
- [2] Peter Kenington, "High Linearity RF Amplifier Design", 2000.
- [3] Steve Cripps, "RF Power Amplifiers for Wireless Communications", 2006.
- [4] Joel Huanca Chavez, "Estudo comparativo entre as arquiteturas de identificação de pré-distorcedores digitais através das aprendizagens direta e indireta", 2018.
- [5] Volnei Pedroni, "Eletrônica Digital e VHDL", 2010.
- [6] Eduardo Gonçalves de Lima and Giovanni Ghione, "Behavioral modeling and digital base-band predistortion of RF power amplifiers", 2009.
- [7] Luis Schuartz and Eduardo Lima, "Polinômios com Memória de Complexidade Reduzida e sua Aplicação na Pré-distorção Digital de Amplificadores de Potência", 2017.
- [8] Elton J Bonfim and Eduardo G De Lima, "A Modified Two Dimensional Volterra-Based Series for the Low-Pass Equivalent Behavioral Modeling of RF Power Amplifiers", vol. 47, pp. 27-35, 2016.