

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

¹Universidade 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). Sendo assim, uma alternativa para contornar esse obstáculo é a implementação de um Pré distorcedor Digital (DPD) em cascata com um PA. Portanto, este trabalho propõe a implementação de um código em VHDL para processar as características de transferências de um DPD 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. A etapa seguinte modelou o PA em vírgula fixa, buscando o melhor NMSE com a menor resolução, onde 8 bits aprimoraram 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 62,5 MHz.

I. INTRODUÇÃO

A evolução dos sistemas de comunicação móveis, impulsionada pela demanda por serviços como multimídia, desenvolvimento web e IoT, trouxe desafios, como a necessidade de melhorar a eficiência energética de dispositivos móveis e estações de rádio base [1]. Para atender a essas demandas, estratégias de modulação que alteram fase e amplitude das ondas portadoras se tornaram essenciais [2]. A modulação em amplitude requer linearidade na transmissão para evitar erros e interferências [3], o que representa um desafio para o projetista do PARF, que deve desenvolver um hardware eficiente e linear ao mesmo tempo, objetivos que podem entrar em conflito [4]. Uma solução é o pré-distorcedor de sinais digitais em banda base, que compensa a distorção do PARF [3]. O DPD precisa de um modelo preciso e de baixa complexidade computacional, e pode ser modelado de forma física ou empírica, sendo a segunda opção mais simples, mas com menor precisão. Para atender aos rigorosos requisitos de

frequência de operação, a paralelização das operações torna-se essencial, e as FPGAs surgem como uma alternativa viável [5]. Neste contexto, o objetivo deste projeto é criar e validar um código em linguagem de descrição de hardware capaz de processar, em tempo real, as características de transferência inversa de um amplificador, minimizando o uso de recursos lógicos e o consumo de energia. O projeto inclui modelar o amplificador e o pré-distorcedor digital em software e implementar o DPD em hardware.

II. MODELAGEM MATEMÁTICA

A. Série de Volterra

A série de Volterra é uma generalização multidimensional da série de Taylor, aplicada a sistemas dinâmicos. A modelagem começa com a representação do sistema por uma série infinita de integrais convolucionais, onde cada termo representa uma ordem de não linearidade e memória. A saída $y(t)$ do sistema pode ser expressa pela equação A:

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

onde h_n são os núcleos de Volterra, caracterizando a resposta do sistema para a n -ésima ordem de não linearidade, e $x(t)$ é a entrada do sistema. Em muitas aplicações práticas, a série é truncada para incluir um número finito de termos devido à impraticabilidade de identificar todos os núcleos de uma série infinita [6].

B. Polinômio de memória

O Polinômio de Memória (MP) é um modelo simples, utilizado na modelagem simplificada das séries de Volterra, considerando componentes unidimensionais. Este modelo é compacto, de baixo custo computacional e linear em seus parâmetros, sendo eficaz para amplificadores de potência (PAs) com pouco efeito de memória. Como o DPD e o pós-distorcedor apresentam características inversas às do PA, o mesmo modelo pode ser utilizado. A equação B descreve o MP conforme ilustrado por [7]:

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

Para implementar esse modelo em hardware, é necessário paralelizar as operações aritméticas, visando altas taxas de

operação. Nesse sentido, as FPGAs são uma alternativa viável para a implementação de circuitos pré-distorcedores.

III. IMPLEMENTAÇÃO EM SOFTWARE

Nesta etapa, foi implementado o modelo DPD em software utilizando Python, com sinais de entrada e saída de um amplificador de potência classe AB baseado em HEMT GaN. O amplificador foi excitado por um sinal de 900 MHz modulado por WCDMA 3GPP, com largura de banda de aproximadamente 3,84 MHz. Os dados foram coletados usando um VSA Rohde & Schwarz FSQ a 61,44 MHz de taxa de amostragem [8].

A seguir, foi calculada a estimativa do sinal utilizando números de vírgula fixa e validada com o NMSE. Para isso, os dados foram divididos em conjuntos de extração e validação. A matriz de regressão foi calculada com os dados de extração para obter os coeficientes do polinômio de memória. O modelo foi validado com os dados de validação, resultando em um NMSE de -23,57 dB para um polinômio de 2º grau com uma amostra memorizada.

O algoritmo foi então ajustado para trabalhar com números em vírgula fixa, buscando a menor resolução possível e o menor NMSE simulado, o que exigiu a readequação dos resultados entre as multiplicações para 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. Para isso o processo A cada ciclo, duas operações são realizadas em paralelo: o sinal atual é elevado ao quadrado e registrado, enquanto ocorre a multiplicação da matriz de extração pelos seus respectivos coeficiente e em um terceiro processo é realizado o somatório desses sinais multiplicados. 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.

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.

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 1.

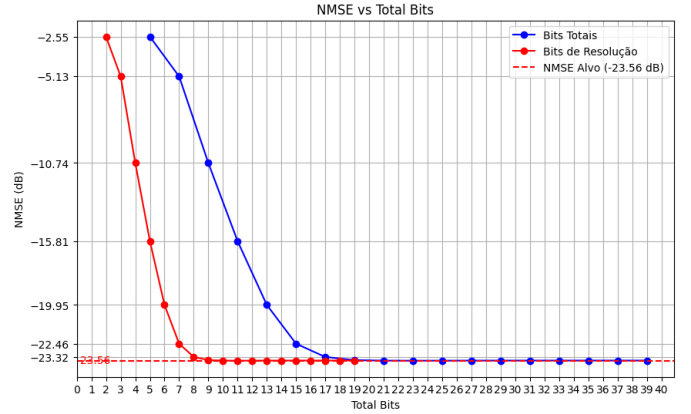


Fig. 1: Gráfico Número de bits x NMSE

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 resolução de 8 bits.

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.

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 2 ilustra essa arquitetura de maneira simplificada.

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 3 apresenta o resultado dessa implementação, representado pelos "x" em vermelho, em comparação com o resultado da simulação em Python, indicado pelos "o" em azul.

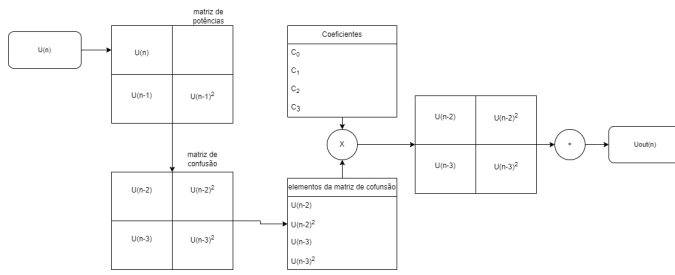


Fig. 2: Processo de cálculo da saída

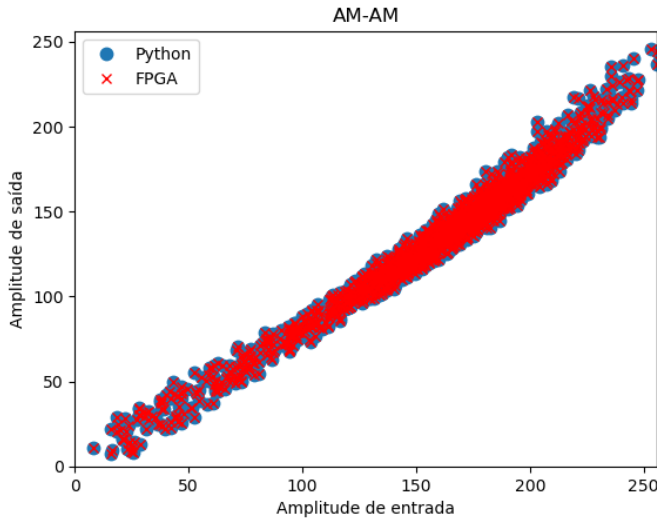


Fig. 3: Processo de cálculo da saída

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 sequência: 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 hard-

ware, 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.