

# Implementação eficiente de algoritmos de equalização cega em FPGA

R. F. Nogueira, F. R. M. Pavan, G. Stolfi, M. D. Miranda

Escola Politécnica, Universidade de São Paulo, São Paulo, SP, Brasil

E-mails: ruanfn@lcs.poli.usp.br, frmp@lcs.poli.usp.br, gstolfi@lcs.poli.usp.br, maria@lcs.poli.usp.br

**Introdução:** Em sistemas de comunicação, os algoritmos de equalização adaptativa autodidata têm um papel fundamental na transmissão eficiente de informações. Depois do célebre algoritmo do módulo constante (CMA) proposto por Godard [1], vários trabalhos surgiram na literatura especializada com objetivos de melhorar seu desempenho, entender seus limites de funcionamento e prever seu funcionamento em situações práticas adversas. Recentemente, em [2] foi proposto um esquema *dual-mode* de equalização com o CMA normalizado (DM-NCMA). Apesar de a estabilidade desse algoritmo ter sido demonstrada teoricamente e avaliada com simulações numéricas, o seu bom comportamento ainda não foi verificado experimentalmente na prática. Propomos investigar o desempenho do algoritmo DM-NCMA implementado em um *hardware* dedicado operando em tempo real.

**Métodos:** Inicialmente, o DM-NCMA é executado e avaliado no *software* MATLAB. Posteriormente, é feita uma implementação em FPGA e, por fim, são realizadas a análise e a comparação dos dados obtidos na implementação. Um diagrama de blocos do esquema proposto para implementação em FPGA e avaliação do algoritmo operando em tempo real é apresentado na Figura 1. O bloco FPGA + Processadores representa a plataforma *Xilinx ZYNQ-7000*, escolhida conforme as necessidades estabelecidas de memória e processamento. Esta plataforma contém uma interface entre uma seção de lógica programável, referente à FPGA XC7Z020-CLG484, e uma seção com dois processadores *Dual-core ARM Cortex-A9 MPCore* [3]. A plataforma possui uma CPU de 600 MHz e capacidade de operar com um único processador ou em modo de multiprocessamento. A velocidade de processamento da lógica programável da FPGA é de 300 MHz. Tendo em vista que a implementação de projetos em placas de lógicas programáveis necessita de uma linguagem de descrição de *hardware*, utiliza-se a ferramenta *HDL CODER* do MATLAB. Essa aplicação permite realizar uma descrição dos algoritmos em VHDL - *VHSIC Hardware Description Language*. Após a conversão das descrições dos algoritmos, utiliza-se a plataforma de desenvolvimento ISE - *Integrated Synthesis Environment*, um ambiente produzido pela *Xilinx* para síntese e análise de projetos em FPGA. Para realizar a aquisição dos dados, utiliza-se um osciloscópio que permite gravar os resultados obtidos. Os dados então retornam ao MATLAB para posterior estudo.

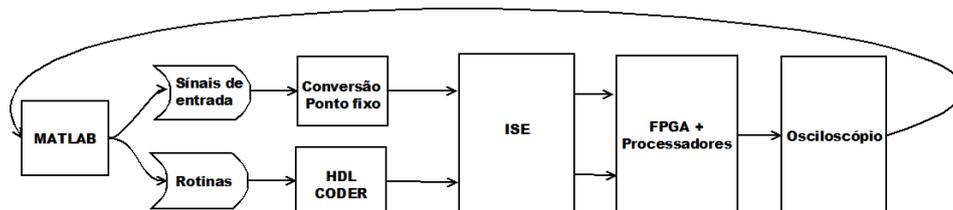


Figura 1: Diagrama de blocos simplificado do esquema para implementar e avaliar o DM-NCMA.

**Resultados e conclusões esperados:** O principal propósito do trabalho é avaliar a eficiência em termos da robustez numérica, do custo computacional e da qualidade das estimativas do algoritmo DM-NCMA em uma implementação prática. A implementação em FPGA exercerá a função de expor a funcionalidade desse algoritmo operando em tempo real. Entretanto, o esquema proposto pode ser estendido a outros algoritmos de equalização autodidata.

**Agradecimentos:** Ruan F. Nogueira é bolsista da Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP), processo 2017/07183-4. Flávio R. M. Pavan é bolsista CAPES.

## Referências Bibliográficas

- [1] D. N. Godard. "Self-recovering equalization and carrier tracking in two dimensional data communication system". *IEEE Trans. Commun.*, vol. 28, pp. 1867–1875, Nov. 1980.
- [2] F. R. M. Pavan, M. T. M. Silva e M. D. Miranda. "Avoiding divergence in the constant modulus algorithm for blind equalization of MIMO systems". *Proc. IEEE Sensor Array and Multichannel Signal Processing Workshop (SAM)*, Rio de Janeiro, 2016, pp. 1–5.
- [3] Zynq-7000: All Programmable SoC with Hardware and Software Programmability. Disponível em: <http://www.xilinx.com/products/silicon-devices/soc/zynq-7000.html>. Acesso em: 4 abr. 2017.