

Implementação e Comparação das Estruturas Paralela, Semi-Paralela e Seqüencial de um Equalizador Largamente Linear em FPGA

Cesar H. V. Vargas, Francisco J. A. de Aquino e Carlos A. F. da Rocha

Resumo— Neste artigo apresentam-se os resultados da implementação em uma placa FPGA de equalizadores adaptativos LMS largamente linear (LMS-LL). Foram utilizadas e comparadas três estruturas de implementação: a paralela, a semi-paralela e a seqüencial. São apresentados os resultados de uma estimação de recursos para um equalizador largamente linear com quatro coeficientes complexos. Independentemente da estrutura utilizada, o equalizador LL apresenta uma taxa de convergência melhor do que o equalizador linear adaptativo LMS convencional, quando o sinal recebido é impróprio. Foram realizadas diversas simulações, utilizando a ferramenta *Systems Generator*, fornecida pela Xilinx. A placa Virtex-4 foi utilizada para executar os algoritmos.

Palavras-Chave— Algoritmo LMS, filtragem adaptativa, equalização, processamento largamente linear, FPGA.

Abstract— In this paper we present the results of the implementation in FPGA board of an adaptive widely linear LMS (WL-LMS) equalizer. We use and compare three structures of implementation: the parallel, the semi-parallel and the sequential. We present the results of resource estimation to a widely linear equalizer with four complex coefficients. Independently of the used structure, the WL equalizer presents a convergence rate better than the conventional adaptive linear LMS equalizer, when the received signal is improper. Several simulations have been carried out through the *System Generator* tool supplied by Xilinx Inc. The Virtex-4 board was used to implement the algorithms.

Keywords— LMS algorithm, widely linear processing, adaptive filtering, equalization, FPGA.

I. INTRODUÇÃO

As técnicas de equalização adaptativa são usadas intensivamente em sistemas de comunicação, para compensar a interferência entre símbolos (ISI) provocada pelo canal. Em general, o equalizador adaptativo é um filtro FIR (*Finite Impulse Response*) que utiliza o algoritmo LMS (*Least Mean Square*) para a adaptação dos seus coeficientes.

Este trabalho foi parcialmente financiado pelo Centro Federal de Educação Tecnológica do Ceará (CEFET-Ce), Universidade Federal de Santa Catarina (UFSC) e Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) processo número 484391/2006-2.

Cesar H. V. Vargas é aluno de doutorado na UFSC no Grupo de Pesquisa em Comunicação (GPqCom), bolsista CAPES/CNPq – IEL Nacional – Brasil, e-mail: cesar.vidal@eel.ufsc.br. Francisco J. A. de Aquino é professor no CEFET-Ce e aluno de doutorado na UFSC (GPqCom), e-mail: fcoalves_aq@cefet-ce.br. Carlos A. F. da Rocha é professor no Departamento de Engenharia Elétrica da UFSC, laboratório GPqCom. E-mail: aurelio@eel.ufsc.br.

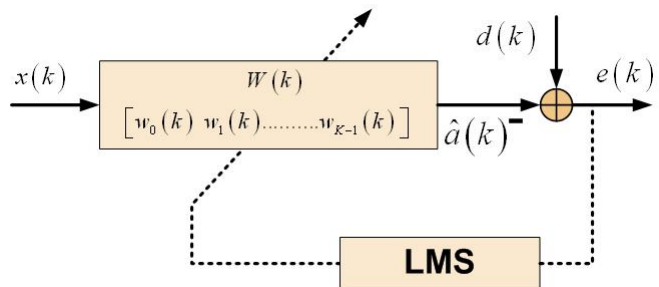


Fig. 1. Filtro transversal adaptativo clássico.

Uma característica importante do algoritmo LMS é a sua simplicidade [1]. A Fig. 1 mostra um filtro FIR linear adaptativo, que pode ser implementado, por exemplo, em uma FPGA (*Field Programmable Gate Arrays*).

A tecnologia de FPGA foi introduzida nos anos 1980 como um dispositivo para processamento de lógica digital. As FPGAs foram projetadas utilizando circuito multi-nível, que poderiam incluir circuitos complexos em uma única pastilha. As FPGAs são pré-fabricadas e, por tanto, são de uso fácil e de baixo custo [2].

O processamento largamente linear pode ser usado, por exemplo, em detecção multi-usário [3], equalização cega e treinada [4], [5], na formatação de feixe [6] e em sistemas MIMO (*multiple-input-multiple-output*) [7]. Apresenta um ganho de desempenho significativo na presença de seqüências complexas impróprias, quando comparado à filtragem linear convencional.

Tipicamente, um processo impróprio surge quando um sinal pertencente a uma constelação real (por exemplo, M-PAM) é transmitido por um canal banda base complexo. Muitos sistemas de comunicação podem ser modelados através deste cenário, por exemplo, na transmissão usando as modulações OQAM (*Offset Quadrature Amplitude Modulations*), sistema GSM (*Global System for Mobile Communication*) e sistemas transmitindo CPM (*Continuous-Phase Modulation*) binário com índice de modulação $h = 1/2$ [4].

Neste artigo, apresenta-se um equalizador largamente linear implementado em FPGA usando as estruturas paralela, semi-paralela e seqüencial. A adaptação dos coeficientes do filtro é realizada utilizando o algoritmo LMS em sua versão largamente linear. Estas estruturas são comparadas através de sua utilização em filtros equalizadores de canais de comunicação digital SISO (*single-input-single-output*), implementados em *System Generator*.

Este artigo é organizado como segue. Na Seção II são introduzidos os fundamentos do processamento largamente linear aplicado ao problema de equalização de canal. Na Seção III apresenta-se as principais características de uma FPGA da série VirtexTM-4 da Xilinx e o ambiente de simulação. Em seguida, apresenta-se o equalizador largamente linear e alguns resultados de simulação. Finalmente, a Seção V conclui o artigo com algumas observações e perspectivas futuras para este trabalho.

II. EQUALIZADOR LARGAMENTE LINEAR

A. Sinal Impróprio

Seja uma seqüência $x(k)$ complexa, discreta, aleatória, com média nula. Usualmente, as estatísticas de segunda ordem de $x(k)$ são descritas pela função de autocovariância (FAC), que é definida por:

$$\mu(k_1, k_2) = E\{x(k_1)x^*(k_2)\}, \quad (1)$$

onde $(.)^*$ é a operação de conjugação complexa. Entretanto, tem sido observado que a FAC não é inteiramente suficiente para descrever as estatísticas de segunda ordem em alguns casos [8]. Logo, é necessário introduzirmos uma outra função chamada de pseudo-autocorrelação (FPAC), definida por:

$$\rho(k_1, k_2) = E\{x(k_1)x(k_2)\}. \quad (2)$$

Assim, estas duas funções, $\mu(k_1, k_2)$ e $\rho(k_1, k_2)$, são necessárias para uma descrição completa das estatísticas de segunda ordem de $x(k)$. Quando $\rho(k_1, k_2)$ é igual a zero, a FPAC pode ser omitida e o processo é conhecido como *próprio*. Por outro lado, se o processo $x[k]$ apresenta uma FPAC não nula, então este processo é dito ser *impróprio*. Um processo impróprio é estacionário no sentido amplo (*wide-sense stationary – WSS*) se, e somente se, $E\{x(k)\} = m_x$ é uma constante e tanto a FAC quanto a FPAC são independentes do tempo discreto k [8].

O processamento largamente linear faz o uso de uma observação complexa imprópria $x(k)$, isto é, usa tanto $x(k)$ quanto o seu conjugado complexo $x^*(k)$, para realizar uma estimativa do sinal $\hat{d}(k)$ desejado, como indica a Fig. 2.

B. Modelo do Sistema

Neste artigo considera-se o sistema de comunicação digital como descrito pela Fig. 2, onde $a(k)$, $s(k)$, $n(k)$, $x(k)$ e $\hat{d}(k)$ representam o sinal transmitido (pertencente a uma constelação real, com média nula), a saída do canal (complexo), o ruído (considerado complexo, com igual variância nas partes real e imaginária, aditivo, gaussiano e branco), o sinal recebido e a saída do equalizador, respectivamente.

O canal de comunicação digital, que pode ser representado por um filtro FIR (*finite impulse response*) de coeficientes complexos e comprimento L , apresenta os efeitos combinados do filtro de transmissão, do canal contínuo no tempo, do filtro de recepção e da amostragem. Considera-se ainda que o transmissor gera uma seqüência de símbolos independentes identicamente distribuídos (i.i.d) e

de variância σ_a^2 . Já o ruído apresenta uma variância σ_n^2 e é considerado próprio.

O sinal recebido, após uma amostragem à taxa de símbolo e perfeita sincronização de portadora, é dado por:

$$x(k) = \sum_{m=0}^L h(m)a(k-m) + n(k), \quad (3)$$

onde $h(m)$ indica a resposta ao impulso do canal. Note que $x(k)$ é um processo impróprio e que o processamento largamente linear pode ser aplicado vantajosamente.

C. Equalização Largamente Linear

Nos esquemas clássicos de equalização, como mostra a Fig. 1, apenas o sinal de entrada $x(k)$ é utilizado no processo de filtragem pelo equalizador, para formar a estimativa do sinal transmitido. Já no processamento largamente linear, como o sinal de observação é impróprio, o conjugado $x^*(k)$ também é utilizado para fazer a estimativa do sinal transmitido/desejado $d(k)$, como mostra a Fig. 3 [4], [8].

Em equalização largamente linear treinada, que utiliza o critério MMSE (mínimo erro quadrático médio), os filtros f e g são obtidos com o objetivo de minimizar $E\{|e(k)|^2\}$, onde $e(k) = a(k-k_0) - \hat{d}(k)$ e k_0 indica um atraso, que deve ser escolhido cuidadosamente [9]. Note que, para o caso de um sinal pertencente a um alfabeto real A , por exemplo, $A = \{\pm 1, \pm 3, \dots, \pm K\}$ e transmitido por um canal complexo, o sinal recebido é impróprio.

Vale a pena resaltar que é suficiente adaptar apenas o filtro f , pois $g = f^*$. Conseqüentemente, o algoritmo LMS largamente linear tem a mesma complexidade computacional de um algoritmo linear convencional. De fato, desde que o equalizador largamente linear pode ter a mesma ordem do canal para compensar os seus efeitos [4], [8], a complexidade computacional pode ser grandemente reduzida.

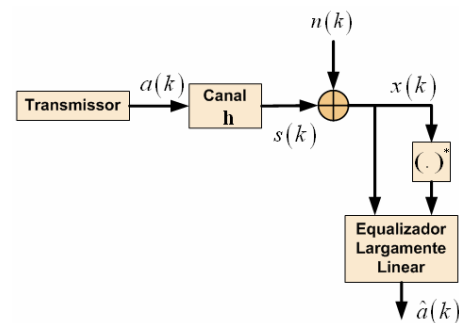


Fig. 2. Diagrama de blocos de um sistema de comunicação digital usando processamento largamente linear.

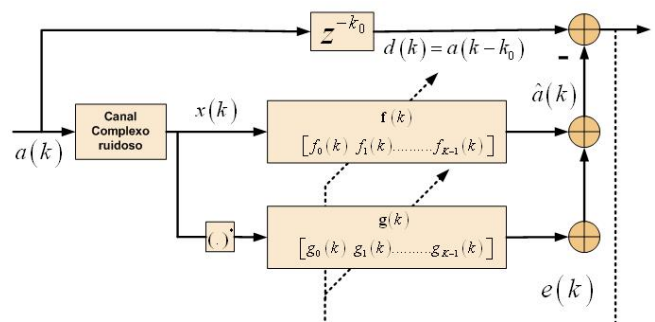


Fig. 3. Equalizador largamente linear treinado.

O modelo ilustrado na Fig. 3 pode ser visto como mostrado na Fig. 4. Note que este modelo é semelhante àquela representação polifásica de um equalizador fracionalmente espaçado (FSE) com uma taxa de amostragem $2/T$, onde T é o intervalo de símbolo. É bem conhecido da literatura [10] que na equalização ZF (*zero-forcing*) de canais polifásicos, o equalizador ZF FIR precisa ter apenas a mesma ordem dos subcanais, desde que estes subcanais tenham raízes disjuntas, por exemplo, nenhuma raiz real e nenhum par de raízes conjugadas.

Em todo o artigo, será considerado que o canal é verdadeiramente complexo, isto é, $\Re\{\mathbf{h}\} \neq 0$ e $\Im\{\mathbf{h}\} \neq 0$ (onde $\Re\{\cdot\}$ é a parte real e $\Im\{\cdot\}$ é parte imaginária), o que é usual em muitas aplicações [4].

O algoritmo LMS largamente linear (WL-LMS), para a adaptação iterativa dos coeficientes do equalizador, pode ser expresso pelo seguinte conjunto de equações [4]:

$$\begin{aligned}\hat{a}(k) &= \mathbf{f}^H(k) \mathbf{x}(k) + \mathbf{g}^H(k) \mathbf{x}^*(k) \\ e(k) &= a(k - k_0) - \hat{a}(k) \\ \mathbf{f}(k+1) &= \mathbf{f}(k) + \mu_{LL} e^*(k) \mathbf{x}(k) \\ \mathbf{g}(k+1) &= \mathbf{g}^*(k+1),\end{aligned}\quad (4)$$

onde μ_{LL} é o passo de adaptação do algoritmo. Na próxima seção apresenta-se o System Generator e a família de placas de FPGAs VIRTEX-4 e algumas considerações sobre a implementação de filtros FIR em FPGA.

III. FPGA VIRTEX-4 E SYSTEM GENERATOR

Nos últimos anos, a tecnologia de dispositivos FPGAs têm evoluído significativamente, alcançando elevados níveis de densidade, altos índices de desempenho e menores custos de fabricação (Fig. 5). Esta evolução tem tornado cada vez menor a distância entre FPGAs e CIs (Circuitos Integrados) para fins específicos. Além dos avanços em capacidade, desempenho e custos, os fabricantes de FPGAs têm introduzido, no decorrer dos anos, cada vez mais recursos de reconfigurabilidade.

Os recursos de reconfigurabilidade recentemente implantados pelos fabricantes de FPGAs têm possibilitado o projeto de sistemas dinamicamente reconfiguráveis. O termo “dinamicamente reconfigurável” indica a possibilidade de se alterar parcialmente a funcionalidade de um dispositivo sem prejudicar o funcionamento de sua lógica restante, que pode estar em operação [11].

A FPGA Virtex-4 produzida pela XILINX, compreende o seguinte conjunto de famílias de FPGAs:

- Virtex-4 LX: aplicações de alto desempenho.
- Virtex-4 SX: aplicações DSP (*Digital Signal Processing*) de alto desempenho.
- Virtex-4 FX: solução completa e de alto desempenho para aplicações de plataformas embarcadas.

Os componentes da Virtex-4 são uma evolução dos componentes já existentes em outras famílias (Virtex, Virtex-E, Virtex-2, Virtex-2 Pro e Virtex-2 Pro X) [12], [13].

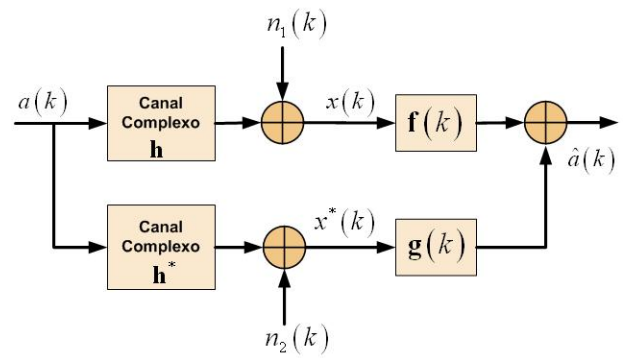


Fig. 4. Equalizador largamente linear: modelo polifásico equivalente.

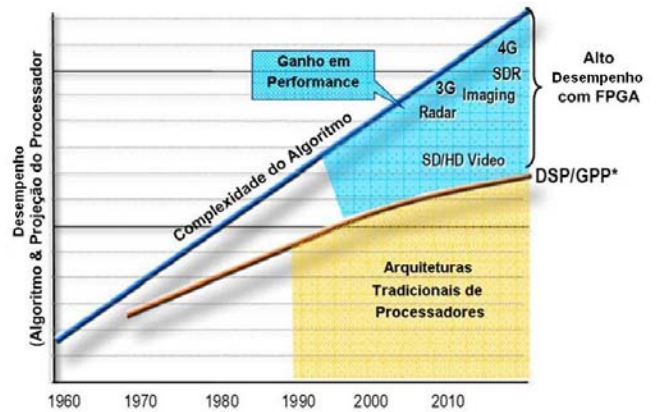


Fig. 5 Desempenho exigido em função do tempo [15]

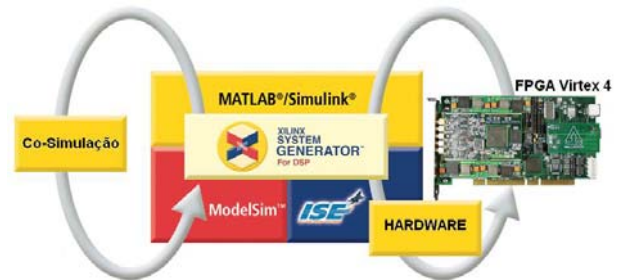


Fig. 6 Fluxo de projeto em Xilinx System Generator [14]

A. System Generator

O *System Generator* é uma ferramenta de projeto integrado, em nível de sistema, para FPGAs, que utiliza o Simulink™, como suporte de desenvolvimento e é apresentado em forma de uma biblioteca (*blockset*) [14].

Como ilustrado na Fig. 6, o *System Generator* através de co-simulação gera um arquivo de configuração (*.bit) necessário para a programação da FPGA [14]. O projeto utilizando *System Generator* facilita a implementação dos equalizadores em FPGA, pois todos os detalhes de implementação são realizados nesta ferramenta e sem a necessidade de se estar conectada com a placa.

B. Implementação de Filtros FIR em FPGA

Uma ampla variedade de arquiteturas de filtros é disponível em FPGA, devido à flexibilidade desta tecnologia. O tipo de estrutura é determinado tipicamente pela quantidade de processamento requerida no número de ciclos de relógio disponível. Os fatores mais importantes são: a taxa de amostragem e o número de coeficientes.

A Figura 7 mostra como o incremento na taxa de amostragem e o incremento no número de coeficientes do filtro influenciam na escolha de uma determinada estrutura. A Tabela I mostra as relações que permitem calcular a taxa de amostragem de entrada do sinal de entrada para o filtro FIR [15].

Utilizando uma das ferramentas do *System Generator*, foi realizada uma estimativa de área em FPGA para um equalizador largamente linear de 4 coeficientes complexos. A Tabela II mostra os resultados obtidos. Como se pode observar, a estrutura seqüencial e semi-paralela consomem um menor número de recursos (*slices*, *flip-flops*, LUTs, etc), conseqüentemente, ocupam uma menor área em FPGA que a estrutura paralela. Vale a pena ressaltar que os resultados mostrados na Tabela II dependem da precisão numérica escolhida dos elementos utilizados para o projeto dos equalizadores. A área ocupada pelas estruturas seqüencial e semi-paralela é uma função não linear do número de coeficientes, isto é, a quantidade de recursos demandados cresce lentamente com o aumento do número de coeficientes do filtro, como mostra a Fig. 8. Já na estrutura paralela, a demanda por recursos da FPGA é maior, pois para cada novo coeficiente do filtro é necessário acrescentar novos elementos (multiplicadores, somadores, etc).

A seguir, será realizada uma comparação de desempenho destas estruturas na equalização de um canal FIR.

C. Comparação das Estruturas

Para a avaliação das estruturas apresentadas é utilizado um canal de fase não-mínima com dois zeros próximos ao círculo de raio unitário descrito pela resposta ao impulso $\mathbf{h} = [0,6791; -0,1698; 0,1698 - 0,4176j; -0,0302 + 0,5531j]$, como mostra a Fig. 9.

Nesta simulação, a constelação utilizada foi uma 4-PAM, considerou-se uma relação sinal-ruído de 30 dB, o passo de adaptação $\mu_{LL} = 0,001$. As curvas de aprendizado, mostradas na Fig. 9, foram obtidas com a mediação de 1000 realizações independentes. Mostra-se também, nesta figura, o resultado da equalização deste canal de teste por um equalizador LL de “precisão infinita” simulado em Matlab®. Os resultados obtidos mostram que o equalizador com a estrutura paralela tem um desempenho muito próximo ao de precisão infinita.

Os equalizadores com estrutura seqüencial e semi-paralela apresentam um desajuste aceitável, porém maior que o desajuste da estrutura paralela. Além disso, observa-se que, para este canal, a taxa de convergência da estrutura paralela é maior, devido ao menor número de arredondamentos e aproximações realizados.

IV. OUTROS RESULTADOS DE SIMULAÇÃO

Nesta seção, compara-se o equalizador largamente linear com a sua versão estritamente linear usando o algoritmo LMS de passo fixo.

Nas simulações a seguir, os canais em tempo discreto são complexos e as seqüências transmitidas são mapeadas usando uma constelação 4-PAM. A relação sinal-ruído foi ajustada em 30 dB. Os resultados da simulação foram obtidos mediando 1000 realizações independentes. A Tabela III mostra os parâmetros usados nas simulações.

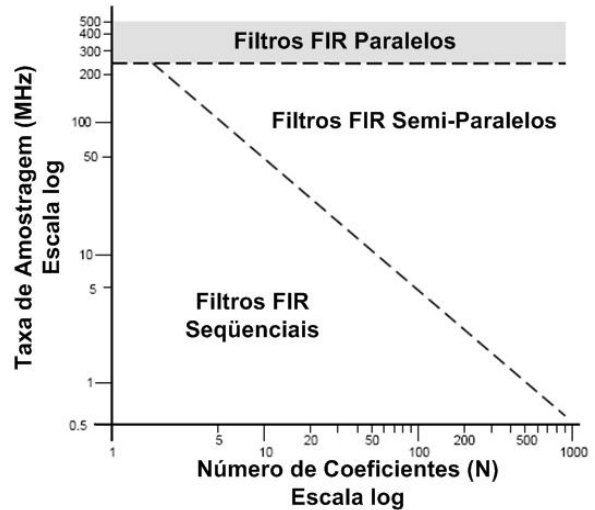


Fig. 7 Comparação das estruturas dos filtros [15].

TABELA I: TAXA DE AMOSTRAGEM PARA AS ESTRUTURAS DE FILTRO FIR [15]

| Estrutura | Taxa de amostragem |
|------------------------|--|
| <i>Paralela</i> | <i>Velc. do relógio</i> |
| <i>Seqüencial</i> | $\frac{\text{Velc. do relógio}}{\text{Num. de coef.}}$ |
| <i>Semi - Paralela</i> | $\frac{\text{Velc. do relógio}}{\text{Num. de coef.}} \times \text{Num. de Mult.}$ |

TABELA II: ESTIMAÇÃO DE RECURSOS

| FPGA Virtex 4 xc4vsx35 – 10ff668 | | | |
|----------------------------------|-----------------------|--------------------------|---------------------|
| | Filtro FIR Seqüencial | Filtro FIR Semi-Paralelo | Filtro FIR Paralelo |
| Slices | 797 | 623 | 4288 |
| Flip Flops | 882 | 594 | 6918 |
| BRAMs | 2 | 4 | 0 |
| LUTs | 1235 | 875 | 6687 |
| IOBs | 110 | 141 | 98 |
| Bem. Mults. | 16 | 10 | 16 |
| TBUFs | 0 | 0 | 0 |

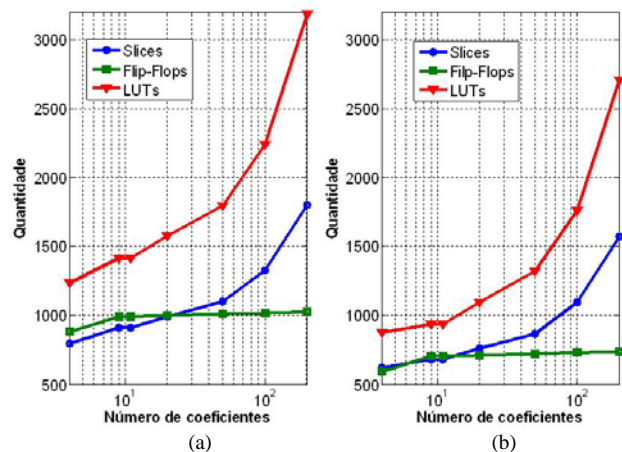


Fig. 8 Demanda de recursos com número de coeficientes para as estruturas (a) seqüencial e (b) semi-paralela.

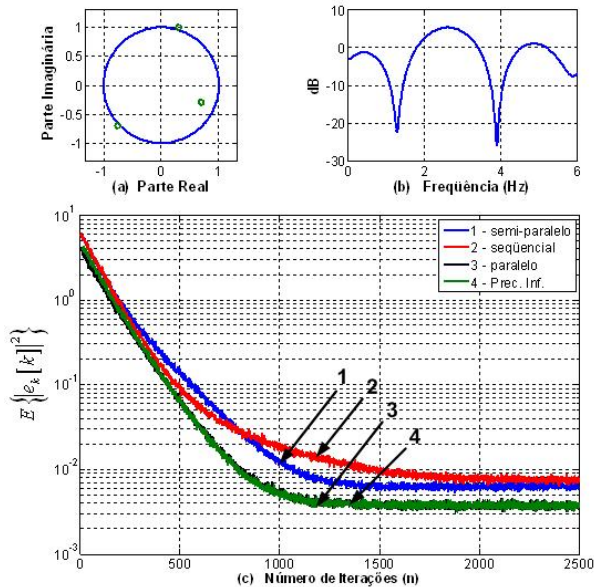


Fig. 9. Comparação das estruturas: (a) raízes do canal, (b) resposta em frequência do canal, (c) curvas de aprendizado.

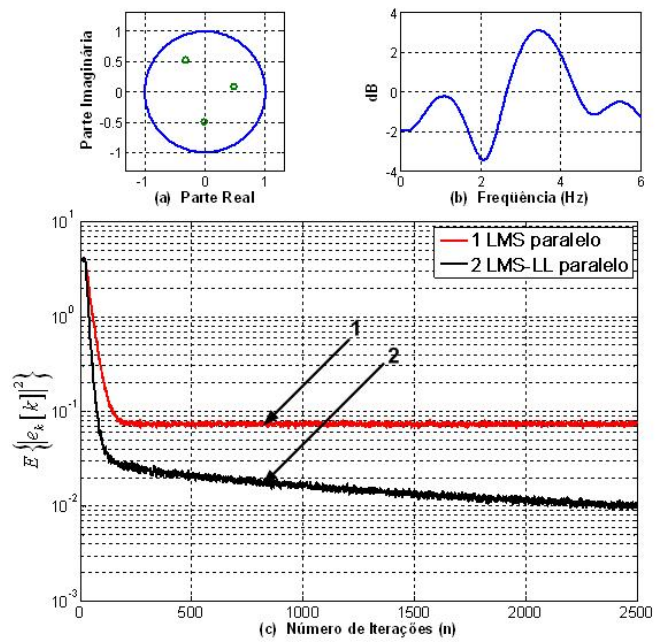


Fig. 10. Simulação I: (a) raízes do canal, (b) resposta em frequência do canal, (c) curvas de aprendizado dos algoritmos para o canal A.

TABELA III : COMPARAÇÃO DOS EQUALIZADORES

| Canal | LMS | LMS-LL |
|-------|--|--|
| A | Número de coef.: 4 Estrutura: paralela passo: 0,004 | Número de coef.: 4 Estrutura: paralela passo: 0,004 |
| B | Número de coef.: 56 Estrutura: seqüencial passo: 0,002 | Número de coef.: 11 Estrutura: seqüencial passo: 0,002 |

A. Simulação I.

O canal A apresenta uma resposta ao impulso $\mathbf{h} = [-0,1216-1,9799j; -0,1950+0,3102j; 0,2659-0,2216j; -0,1772+0,2393j]$. Este canal é de fase mínima e seus zeros estão longe do círculo de raio unitário. A Fig. 10 (a) e (b) mostra os zeros do canal e a sua resposta em frequência. A Fig. 10 (c) apresenta as curvas de convergências do erro quadrático médio (MSE – “mean-square-error”) para os algoritmos LMS e LMS largamente linear. A partir desta figura, pode-se observar que o algoritmo LMS largamente linear apresenta um menor MSE e uma melhor taxa de convergência.

B. Simulação II.

O canal B tem a seguinte resposta ao impulso: $\mathbf{h} = [-0,1375+0,2474j; 0,176-0,264j; 0,2749+0,33j; -0,22-0,385j; -0,44+0,4948j]$. Este canal é de fase não mínima e tem um zero próximo ao círculo de raio unitário, apresentando também uma forte distorção, como é mostrado pela Fig. 11. As curvas de aprendizado mostram que o algoritmo LMS largamente linear apresenta um menor tempo para a convergência e um menor MSE.

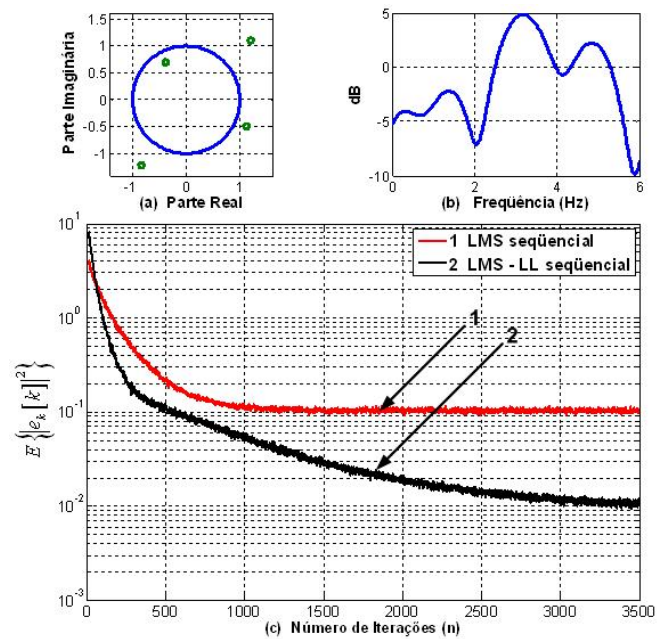


Fig. 11. Simulação II: (a) raízes do canal, (b) resposta em frequência do canal, (c) curvas de aprendizado dos algoritmos para o canal B.

V. CONCLUSÃO

Neste artigo, apresentou-se três estruturas para o projeto de um equalizador adaptativo largamente linear. Foi usado o algoritmo LMS de passo fixo. Estas estruturas foram comparadas entre si e validadas através de simulação para a equalização de canais de comunicação. A estrutura paralela apresentou o desempenho mais próximo ao de precisão infinita, porém, uma maior área em FPGA é ocupada quando comparada com as estruturas semi-paralela e seqüencial. Por outro lado, as duas últimas demandam a mesma quantidade de recursos da FPGA, aproximadamente.

Como continuidade deste trabalho planeja-se: usar o algoritmo LMS com passo variável, implementar um equalizador DFE largamente linear treinado e um filtro de erro de predição largamente linear como um equalizador autodidata.

REFERÊNCIAS

- [1] B. Widrow e S. D. Stearns. *Adaptive Signal Processing*. Prentice-Hall, 1985.
- [2] S. Hauck, "The Roles of FPGA's in Reprogrammable Systems," *Proceedings of the IEEE*, vol. 86, no. 4, pp. 615-638, April 1998.
- [3] G. Gelli, L. Paura e A. R. P. Ragozini, "Blind widely linear multiuser detection," *IEEE Commun. Lett.*, vol. 4, no. 6, pp. 187-189, Jun. 2000.
- [4] W. H. Gerstacker, R. Schober e A. Lampe, "Receivers with widely linear processing for frequency-selective channels," *IEEE Commun. Lett.*, vol. 7, no. 9, pp. 1512-1523, Sep. 2003.
- [5] R. Schober, W. H. Gerstacker, L. H.-J. Lampe, "Data-Aided and Blind Stochastic Gradient Algorithms for Widely Linear MMSE MAI Suppression for DS-CDMA", *IEEE Trans. on Signal Processing*, vol. 52, pp. 746-755, no. 3, Mar 2004.
- [6] T. McWhorter e T. Schreier, "Widely-Linear Beamforming", *The Thirty-Seventh Asilomar Conference on Signals, Systems and Computers*, vol. 1, pp. 753-759, Nov., 2003.
- [7] W. H. Gerstacker, F. Obernosterer, R. Schober, A. T. Lehmann, A. Lampe e P. Gunreben, "Equalization Concepts for Alamouti's Space-Time Block Code", *IEEE Trans. Communications*, vol. 52, pp. 1178-1190, 2004.
- [8] B. Picinbono, P. Bondon, "Second-order statistics of complex signals", *IEEE Transactions on Signal Processing*, Vol. 45, pp. 411-420, Feb. 1997.
- [9] P.A. Voois; I. Lee, e J. M. Cioffi, "The effect of decision delay in finite-length decision feedback equalization", *IEEE Transactions on Information Theory*, Vol. 42, pp. 618-621, Mar. 1996.
- [10] C.B. Papadias e D.T.M. Slock, "Fractionally spaced equalization of linear polyphase channels and related blind techniques based on multichannel linear prediction", *IEEE Trans. on Signal Processing*, Vol. 47, pp. 641-654, Mar. 1999.
- [11] Lysaght, P; Dunlop, J., "Dynamic Reconfiguration of Field Programmable Gate Arrays". *Proceedings of the 3rd International Workshop on Field Programmable Logic and Applications (FPL'93)*, Oxford, UK Abingdon EE&CS Books, p. 82-94, 1993.
- [12] J. Hwang, B. Milne, N. Shirazi and J. Stroemer, *System Level Tools for DSP in FPGAs*, Xilinx Inc. 2001.
- [13] XILINX Inc. Achieving Breakthrough Performance in Virtex-4 FPGAs. White Paper: Virtex-4 FPGAs. May, 2006.
- [14] XILINX Inc. System Generator for DSP version 8.1. ed., San Jose California : XILINX Inc. , 2005.
- [15] XILINX Inc. DSP: Designing for Optimal Results ed. 1.0: XILINX Inc., March 2005, pp. 63-74.