

## IMPLEMENTAÇÃO DE UM FILTRO DE ERRO DE PREDIÇÃO LARGAMENTE LINEAR EM FPGA

**Francisco J. A. de AQUINO (1)**

(1) Centro Federal de Educação Tecnológica do Ceará, Av. 13 de Maio, no. 2081, Coordenação de Telemática, (85)  
telefone, fax, e-mail: [fcoalves\\_aq@cefet-ce.br](mailto:fcoalves_aq@cefet-ce.br)

### RESUMO

Neste trabalho de pesquisa teórico e experimental, apresentamos uma implementação em FPGA (*Field Programmable Gate Arrays*) de um filtro de erro de predição largamente linear (FEPLL). O FEPLL é uma estrutura que pode ser usada na equalização cega (não supervisionada) de canais de comunicação sendo um dos frutos do nosso trabalho de doutorado. O objetivo básico deste trabalho é descrever o FEPLL, suas principais características e como ele foi implementado em FPGA, levando em conta os problemas precisão numérica finita e do comprimento da palavra digital que são típicos de uma implementação em *hardware*. Inicialmente, discutimos sobre o problema de equalização do canal de comunicação e mostramos algumas das diversas propostas existentes na literatura para a solução desse problema, em especial com o uso do processamento largamente linear. No artigo completo, definimos o que é processamento largamente linear e de alguns conceitos necessários à sua compreensão. Também justificamos a necessidade do uso das técnicas de equalização cega. Nesse contexto, a equalização usando o FEPLL se mostra uma alternativa de baixo custo computacional quando o sinal a ser equalizado é impróprio, além de ser facilmente implementável em *hardware*, usando, por exemplo, uma placa FPGA. Adicionalmente, no FEPLL implementado incluímos circuitos de correção de fase e de ganho proporcionando uma boa recuperação do sinal transmitido, mesmo quando o canal gera uma distorção severa. Usamos um algoritmo do tipo LMS para ajustar os coeficientes do FEPLL. Realizamos diversas simulações utilizando a ferramenta *System Generator*, fornecida pela fabricante Xilinx, para mostrar o desempenho do equalizador proposto implementado em FPGA.

**Palavras-chave:** FPGA, filtragem adaptativa, equalização cega, processamento largamente linear.

## 1. INTRODUÇÃO

Em geral, em um sistema de comunicação digital, o canal de comunicação é variante no tempo e introduz uma distorção significativa no sinal enviado do transmissor ao receptor. A principal distorção introduzida pelo canal é a interferência entre símbolos (IES). Uma forma efetiva de se combater a IES é com o uso de um equalizador adaptativo (PROAKIS, 1995). Neste contexto, o equalizador nada mais é que um filtro digital capaz de compensar, ao menos parcialmente, os efeitos danosos do canal de comunicação sobre a mensagem digital transmitida. Como os parâmetros do canal variam no tempo, especialmente no caso do canal radiomóvel, o equalizador também precisa ser constantemente ajustado.

A forma tradicional de se ajustar os parâmetros do equalizador adaptativo é por meio de uma sequência de treinamento que é periodicamente transmitida e do uso de um algoritmo adaptativo (QURESHI, 1985). Entretanto, nem sempre existe uma sequência de treinamento disponível. Nesse caso, a adaptação do equalizador é realizada de forma cega ou não supervisionada. A Figura 1 ilustra, de forma simplificada, um sistema de comunicação digital usando um equalizador adaptativo na recepção dos dados. Nessa figura estão excluídos vários detalhes do esquema de equalização que serão discutidos posteriormente neste artigo. O algoritmo adaptativo empregado por ser, por exemplo, o LMS (*Least Mean Square*).

Nos últimos anos, vários pesquisadores têm proposto o uso do processamento largamente linear como uma forma de aumentar o desempenho dos equalizadores (GELLI et al., 2000; GERSTAKER et al., 2003; AQUINO et al., 2007). O uso do processamento largamente linear é vantajoso quando o sinal recebido é impróprio (PICINBONO e CHEVALIER, 1995). Um exemplo de sinal impróprio ocorre quando um sinal real (pertencente a uma constelação PAM, por exemplo) é transmitido por um canal complexo. Outros dois exemplos importantes são o caso de uma transmissão OQAM (*offset quadrature amplitude modulation*) e GSM (*Global System for Mobile Communication*), como mostrado por Gerstaker et al. (2003).

Neste artigo, o objetivo principal é apresentar a implementação em FPGA (*Field Programmable Gate Arrays*) de um filtro de erro de predição largamente linear (FEPLL) e alguns resultados de simulação atestando o seu desempenho. O FEPLL pode ser utilizado como um equalizador adaptativo cego de baixa complexidade computacional. A teoria básica sobre o funcionamento do FEPLL também pode ser encontrada em (AQUINO et al., 2007). Uma grande vantagem da implementação de circuitos em FPGA é grande facilidade em reconfigurar o *hardwar* desenvolver e testar protótipos rapidamente.

Para alcançar o objetivo citado, o artigo encontra-se organizado como segue. Na Seção 2, são apresentados os conceitos fundamentais sobre o processamento largamente linear. A Seção 3 discute sobre o FEPLL e a sua implementação em FPGA. Na Seção 4, são mostrados os resultados de simulação. Finalmente, na Seção 5, são apresentados os comentários finais e as conclusões.

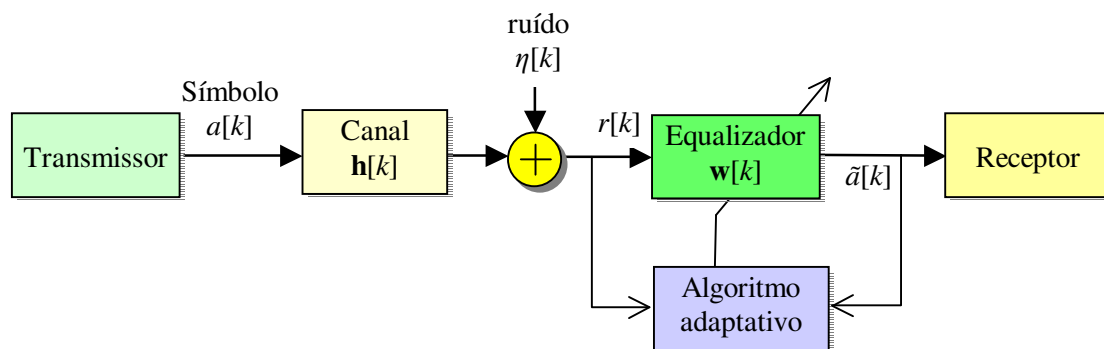


Figura 1. Modelo simplificado de um sistema de comunicação digital com equalização.

## 2. CONCEITOS MATEMÁTICOS FUNDAMENTAIS

Nesta seção, são definidos os conceitos básicos sobre processamento largamente linear. Para uma descrição detalhada dessa teoria, ver Picinbono (1994) e Picinbono e Chevalier (1995).

Seja uma sequência  $x[k]$  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:

$$\mu[k_1, k_2] = E\{r[k_1]r^*[k_2]\}, \quad [\text{Eq. 01}]$$

onde  $(.)^*$  é a operação de conjugação complexa.

Em geral, se  $r[k]$  corresponde à amostragem de um sinal aleatório passa faixa de média nula, a Equação 01 é igual a zero (ver, por exemplo, THERRIEN, 1992) e pode, portanto, ser omitida. Este é o caso, por exemplo, do sinal analítico de qualquer sinal estacionário e, de forma mais geral, de qualquer sinal circular. Como uma consequência óbvia, observamos que os sinais reais não podem ser circulares. Contudo, não existe qualquer razão específica para que a pseudo-covariância de sinais complexos seja sempre nula, sendo, por isso, necessária para definir completamente as estatísticas de segunda ordem (NESSER e MASSEY, 1993). É, portanto, necessário introduzir uma outra função chamada de pseudo-autocorrelação (FPAC), definida por:

$$\rho[k_1, k_2] = E\{r[k_1]r[k_2]\}. \quad [\text{Eq. 02}]$$

Então, 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 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,  $E\{r[k]\} = m_r$  é uma constante e tanto a FAC quanto a FPAC são independentes do tempo discreto  $k$ .

O processamento largamente linear faz o uso de uma observação complexa imprópria  $r[k]$ , isto é, usa tanto  $r[k]$  quanto o seu conjugado complexo  $r^*[k]$ , para realizar uma estimativa do valor aleatório  $\tilde{a}[k]$  desejado. Matematicamente, isso pode ser expresso por:

$$\tilde{a}[k] = \sum_{n=0}^L c_n^*[k]r[k-n] + \sum_{n=0}^L d_n^*[k]r^*[k-n] = \mathbf{c}^H[k]\mathbf{r}[k] + \mathbf{d}^H[k]\mathbf{r}^*[k], \quad [\text{Eq. 03}]$$

onde  $c_n[k]$  e  $d_n[k]$  são valores complexos e podem ser, por exemplo, os coeficientes de um filtro largamente linear,  $\mathbf{r}[k]$ ,  $\mathbf{c}[k]$ ,  $\mathbf{d}[k]$  são vetores coluna contendo os elementos  $r[k]$ ,  $c_n[k]$  e  $d_n[k]$ , respectivamente; o superescrito  $H$  significa a operação de conjugação complexa e transposição combinadas (*hermitiana*).

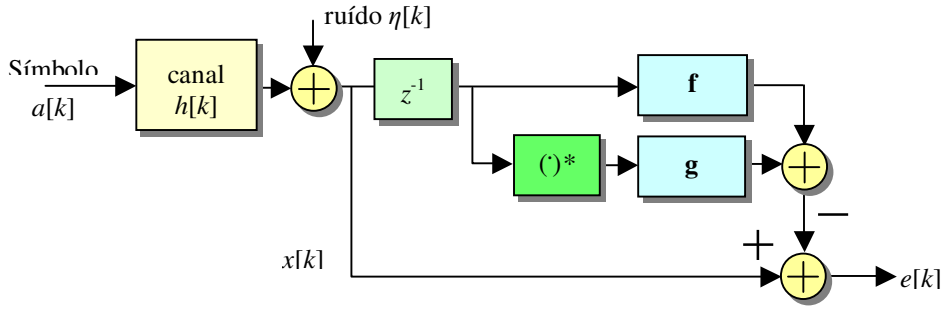
## 3. IMPLEMENTAÇÃO EM FPGA DO FEPLL

Nesta seção, apresentamos o FEPLL e o algoritmo LMS usado para ajustar os seus parâmetros. Em seguida, falamos brevemente sobre a tecnologia FPGA e a sobre a implementação do FEPLL.

### 3.1. O filtro de erro de predição largamente linear (FEPLL)

A Figura 2 mostra um filtro de erro de predição largamente linear. Este filtro, da mesma forma que o equalizador LL, apresenta a operação de conjugação complexa e os filtros  $\mathbf{f}$  e  $\mathbf{g}$  que atuam em paralelo. É interessante notar que a saída do FEPLL, da mesma forma que no caso do FEPL, é o próprio sinal de erro  $e[k]$ .

O FEPLL pode, ao contrário do filtro de erro de predição (FEP) estritamente linear, equalizar satisfatoriamente canais complexos de fase não mínima ou com nulos espectrais, desde que os zeros do canal não sejam zeros reais ou pares de zeros complexos conjugados próximos ao CRU, quando são usados sinais impróprios (AQUINO *et al.*, 2007).



**Figura 2. Filtro de erro de predição largamente linear (FEPLL).**

O sinal de entrada  $a[k]$  deve ser descorrelacionado para que o FEPLL opere apropriadamente. Esta não é uma restrição importante, pois, em geral, nos sistemas de comunicação os símbolos transmitidos se distribuem de forma aproximadamente uniforme em um dado alfabeto. Além disso, em vários sistemas de comunicação são usados embaralhadores (*scramblers*) e desembaralhadores (*unscramblers*) que eliminam seqüências longas de bits iguais. Adicionalmente, lembramos que o sinal  $a[k]$  deve ser impróprio, isto é, ele deve pertencer a uma constelação real ( $M$ -PAM), ou do tipo OQAM (BPSK, MSK, GMSK ou OQAM).

Podemos encontrar uma análise detalhada do FEPL, não adaptativo, em Haykin (1990). No contexto de equalização multicanal (FSE), o primeiro trabalho usando predição linear é apresentado por Slock (1994) e aprofundado por Abed-Meraim *et al.* (1997). Abed-Meraim *et al.* mostram que o FEPL em sistemas superamostrados é robusto ao problema de sobre determinação da ordem do canal, o que não ocorre com outros métodos baseados em estatísticas de segunda ordem, tais como os métodos baseados em subespaço.

### 3.2. FEPLL ADAPTATIVO – ALGORITMO LMS

Apresentaremos agora um algoritmo adaptativo para o filtro de erro de predição LL. Este algoritmo é, em essência, o algoritmo LMS aplicado ao FEPLL.

A Figura 3 apresenta o esquema para o filtro de erro de predição LL adaptativo, o dispositivo de decisão é do tipo *hard* (decisão símbolo a símbolo). As equações (4.46) e (4.47) definem o algoritmo LMS adaptado para o filtro de predição LL. Ao contrário do algoritmo adaptativo apresentado para o equalizador, precisamos deixar os filtros **f** e **g** evidenciados, pois estes não são mais complexos conjugados um do outro. Assim, podemos escrever o algoritmo LMS como

$$e[k] = x[k] - \mathbf{f}^H[k-1]\mathbf{x}[k-1] - \mathbf{g}^H[k-1]\mathbf{x}^*[k-1], \quad [\text{Eq. 04}]$$

$$\begin{bmatrix} \mathbf{f}[k] \\ \mathbf{g}[k] \end{bmatrix} = \begin{bmatrix} \mathbf{f}[k-1] \\ \mathbf{g}[k-1] \end{bmatrix} + \mu e^*[k] \begin{bmatrix} \mathbf{x}^*[k-1] \\ \mathbf{x}[k-1] \end{bmatrix}, \quad [\text{Eq. 05}]$$

onde  $\mu$  é o passo de adaptação. Podemos realizar as correções de fase e de ganho (CAG – controle automático de ganho), para compensar o efeito multiplicativo de  $h_0$  (primeiro coeficiente do canal), usando as seguintes expressões:

$$\phi[k] = \phi[k-1] + \mu_\phi \text{angle}(e[k] \exp(j\phi[k-1])), \quad [\text{Eq. 06}]$$

$$G[k] = G[k-1] + \mu_G \left( E\{|a[n]|^2\} - |x[n]|^2 \right), \quad [\text{Eq. 07}]$$

$$\gamma_k = \sqrt{|G[k]|}, \quad [\text{Eq. 08}]$$

onde  $\mu_\phi$  e  $\mu_G$  correspondem aos passos de adaptação da fase e do ganho, respectivamente. Para a inicialização dos algoritmos podemos escolher  $\mathbf{f}[0] = \mathbf{g}[0] = \mathbf{0}$ ,  $\phi[0] = 0$  e  $G[0] = 1$ .

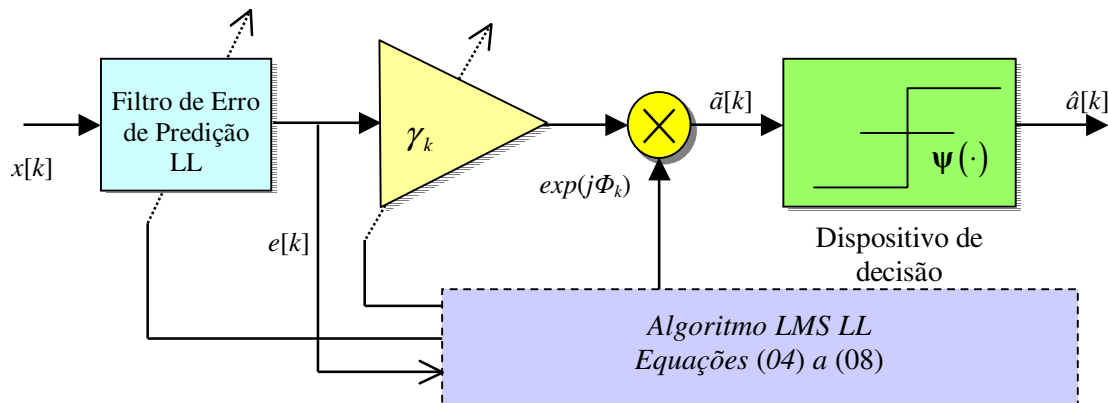


Figura 3. Esquema de equalização cega usando o filtro de predição LL.

### 3.3. TECNOLOGIA FPGA

A tecnologia de dispositivos FPGA foi introduzida nos anos 1980 como um dispositivo para processamento de lógica digital. As FPGA's foram projetadas utilizando circuito multi-nível, que poderiam incluir circuitos complexos em uma única pastilha. As FPGA's são pré-fabricadas e, portanto, são de uso fácil e de baixo custo. Além disso, dado que elas são facilmente reconfiguradas e permitem a correção de erros (bugs) de sistemas em desenvolvimento, as FPGA's são ideais para prototipagem (REED, 2002; HAUCK, 1998).

Convém observar que em uma implementação prática, vários fatores que, em geral, não são considerados nos projetos (puramente) teóricos começam a ter uma importância fundamental: a precisão numérica (número de bits), a latência das operações matemáticas, a área ocupada em silício, memória disponível, o consumo de energia, entre outros.

Vale a pena lembrar que o projetista também tem à sua disposição duas outras categorias de hardware além das FPGA's: ASIC's (*application specific integrated circuits*) e DSP's (*digital signal processor*). Enquanto um DSP representa o tipo de hardware mais generalizado que pode ser programado para desempenhar várias funções, um ASIC é um circuito especializado e só pode ser usado para desempenhar a função para qual foi projetado.

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

Existem vários fabricantes de FPGA's, entre eles ALTERA, Achronix Semiconductor e XILINX. Em nossas implementações usamos exclusivamente uma placa FPGA da família Virtex-4 produzida pela XILINX. Esta família compreende o seguinte conjunto de FPGA's:

- 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) (HWANG et al., 2001; XILINX Inc, 2006).

O System Generator é uma ferramenta de projeto integrado, em nível de sistema, para FPGA's, que utiliza o Simulink™ (Matlab®), como suporte de desenvolvimento e é apresentado em forma de uma biblioteca

(*blockset*) (XILINX Inc., 2006). Todo o projeto de um circuito em FPGA pode ser realizado no ambiente Simulink como mostra a Figura 4.

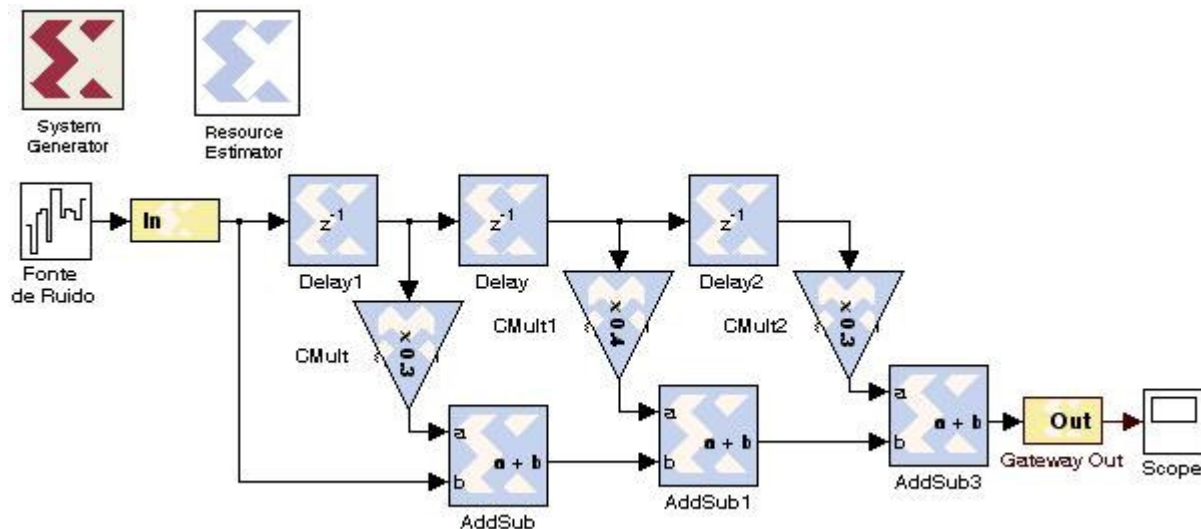


Figura 4. Exemplo de um projeto em FPGA usando *System Generator* em ambiente Simulink.

Como ilustrado na Figura 5, o System Generator, através de co-simulação, gera um arquivo de configuração (\*.bit) necessário para a programação da FPGA (XILINX Inc., 2005). 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 conectado à placa FPGA.

Convém lembrar que System Generator não tem por objetivo substituir o VHDL ou o Verilog, mas o seu uso leva a um ganho de produtividade ao reduzir o tempo gasto nos problemas de linguagem de programação e permitir que o foco fique sobre outras dificuldades (XILINX Inc., 2005).

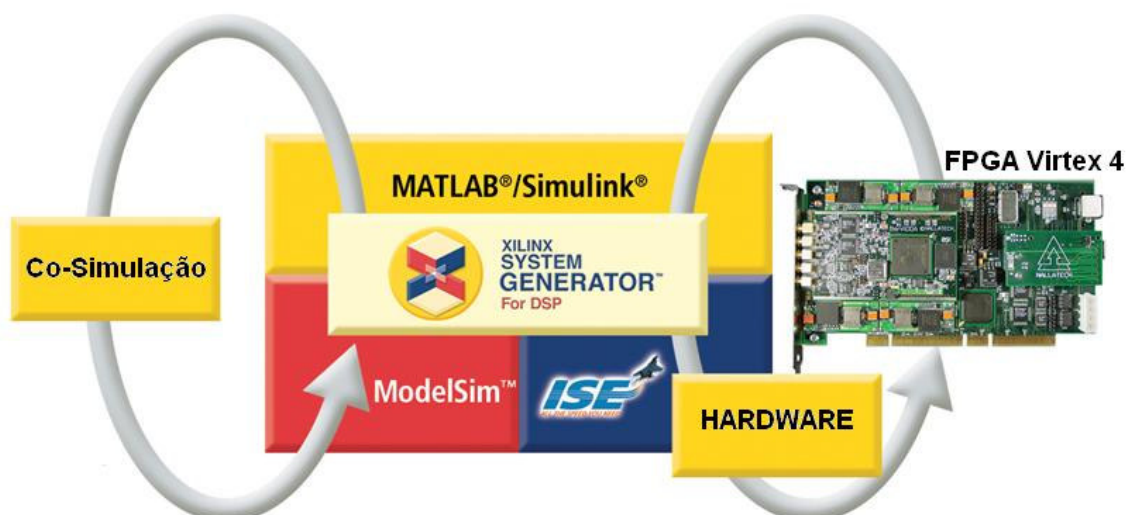


Figura 5. Fluxo de projeto em *Xilinx System Generator*.

### 3.4. IMPLEMENTAÇÃO EM FPGA DO FEPLL

Agora, apresentamos a implementação em FPGA de um FEPLL de cinco coeficientes. Esta implementação inclui a correção de fase e ganho, mas com o algoritmo modificado em relação ao descrito pelas equações (04) a (08). As modificações foram necessárias para manter uma baixa complexidade e porque todas as operações (somadas, multiplicações) são realizadas com números reais e não com valores complexos. Outro detalhe é que, devido à implementação realizada ser do tipo paralela e a latência decorrente das operações de multiplicação, o algoritmo realizado é um LMS atrasado (Delayed LMS). Isso, entretanto, não é crítico para esta aplicação, pois o passo de adaptação é pequeno (HAIMI-COHEN *et al.*, 1990; LONG *et al.*, 1989; TOBIAS *et al.*, 2000). Na Figura 6 vemos detalhes da implementação do ajuste de ganho e na Figura 7 temos uma visão geral do FEPLL implementado.

Lembramos que os sinais complexos são separados em suas partes em fase e quadratura antes de iniciar o processamento. Desta forma, uma multiplicação complexa é efetuada com o uso de quatro multiplicações reais e mais duas somas reais.

O ajuste de ganho, de acordo com a Figura 6, é dado por:

$$g[k] = g[k-1] + (2^{-6} + 2^{-8}) \cdot (2 - g[k-3] E\{e_r[k]\}), \quad [\text{Eq. 09}]$$

com

$$E\{e_r[k]\} \cong 2^{-8} |e_r[k]| + (1 - 2^{-8}) |e_r[k-1]|, \quad [\text{Eq. 10}]$$

observamos que o uso de constantes de potência de 2 permitem que as multiplicações sejam facilmente implementadas (baixo custo em *hardware*) e com latência nula.

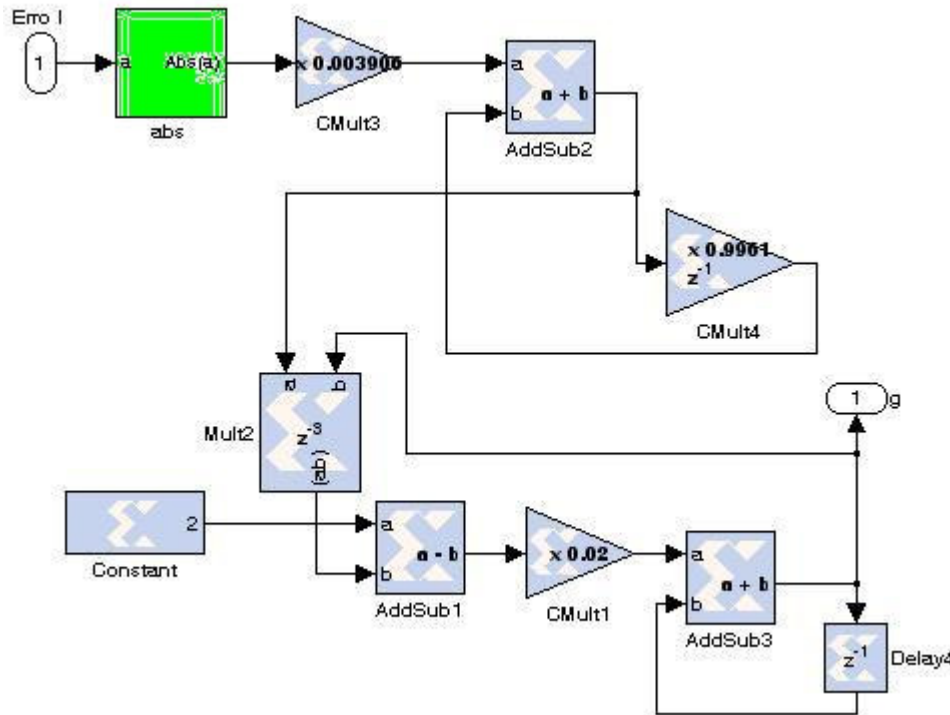


Figura 6. Implementação em FPGA: detalhe do ajuste do ganho.



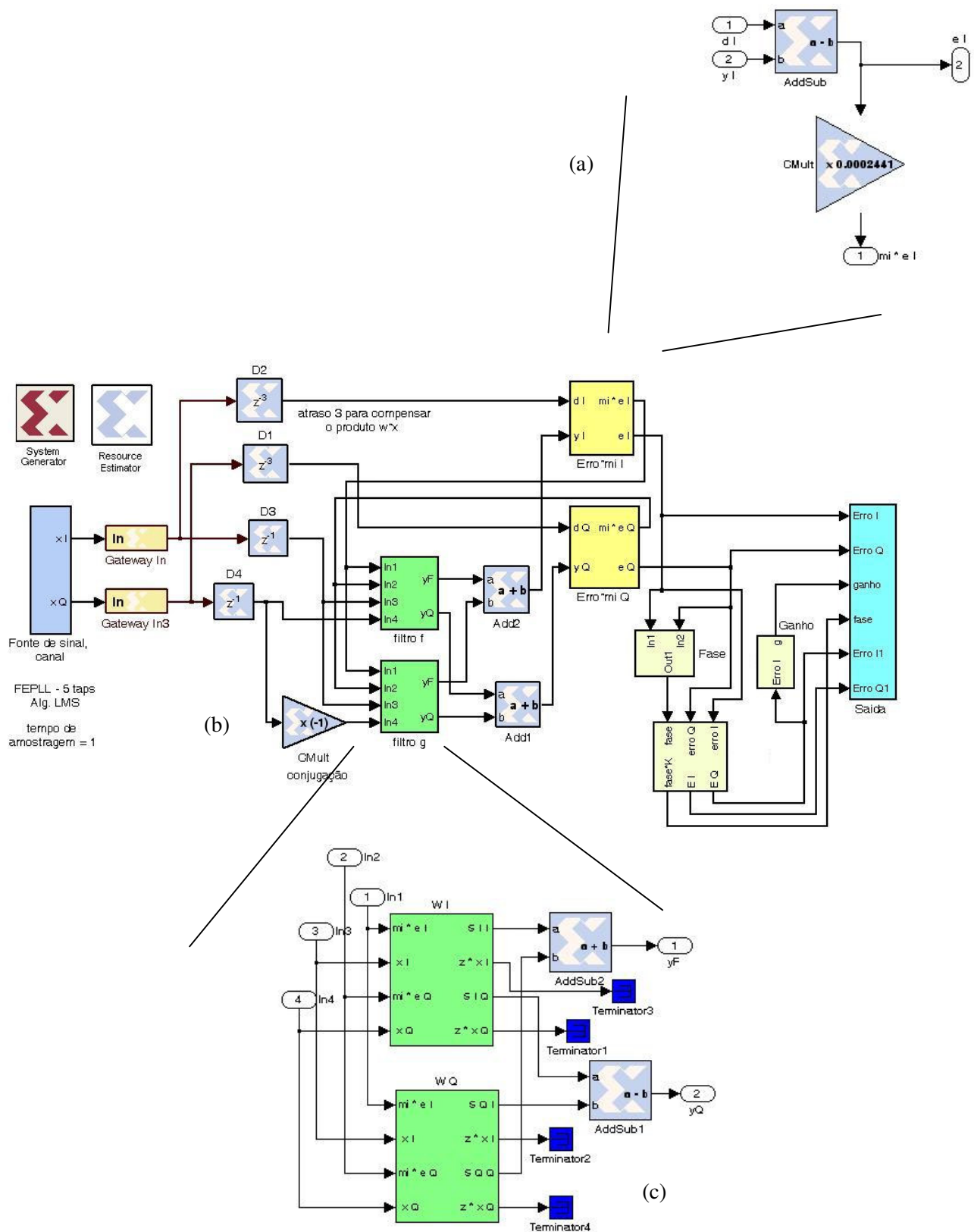


Figura 7. Implementação do FEPLL em FPGA: (a) detalhe do cálculo do erro, (b) visão geral do FEPLL, (c) detalhamento de um dos filtros transversais.



#### 4. RESULTADOS DE SIMULAÇÃO

Para demonstrar a eficácia do equalizador implementado, consideramos o seguinte problema de equalização: canal complexo de fase não mínima e com um zero próximo ao CRU definido pela resposta ao impulso  $\mathbf{h} = [0,5432 + 0,1207j; 0,0805 - 0,2012j; -0,2314 - 0,6840j; -0,2615 - 0,2314j]$ , um sinal transmitido pertencente a uma constelação 4-PAM e uma relação sinal ruído de 30 dB. Nesta simulação consideramos passo de adaptação  $\mu = 2^{-13} + 2^{-16}$ . O total de bits usados nos somadores e multiplicadores foi de 22 bits, sendo 16 bits para a parte fracionária.

Podemos observar na Figura 8(b) que na correção de ganho ocorre uma saturação do fator de correção do ganho devido à implementação escolhida (somador sem sinal e com apenas um bit após o ponto decimal).

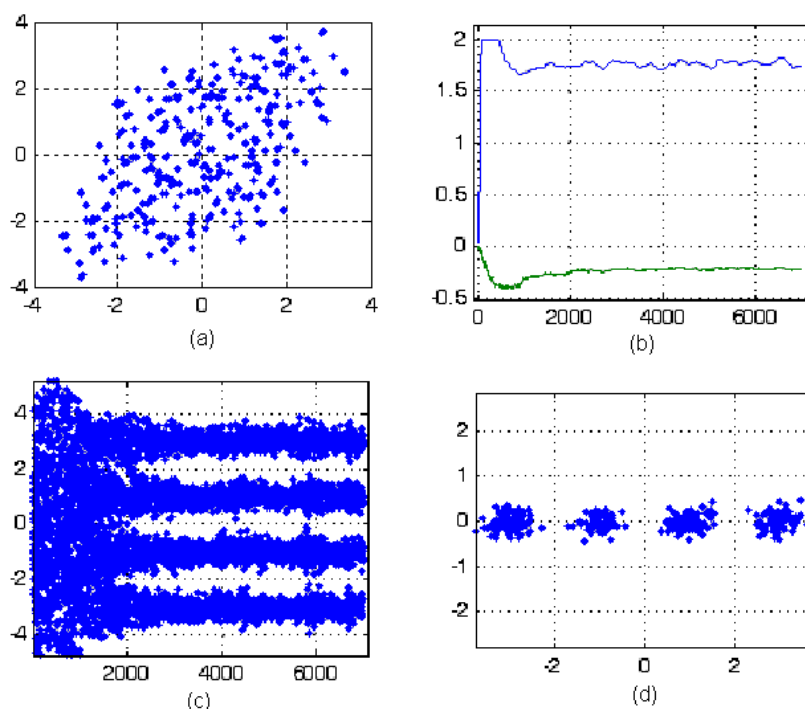


Figura 8. Equalização com o FEPLL implementado em FPGA. (a) Sinal após o canal, (b) ajuste de ganho e de fase, (c) evolução da constelação, (d) diagrama de olho após a convergência. Relação sinal-ruído de 30 dB.

#### 5. CONCLUSÕES

Neste artigo apresentamos a fundamentação do processamento largamente linear e a implementação em FPGA de um filtro de erro de predição largamente linear (FEPLL) com o objetivo de equalizar um canal de comunicação. Os resultados de simulação mostram a viabilidade do FEPLL como equalizador. Uma vantagem dessa abordagem é a baixa complexidade computacional e o melhor desempenho quando comparado à técnicas lineares convencionais de equalização cega.

#### REFERÊNCIAS

ABED-MERAIM, K., MOULINES, E., e LOUBATON, P., Prediction Error Method for Second-Order Blind Identification. In: *IEEE Trans. Signal Processing*, vol. 45, no. 3, pp. 694-705, Mar. 1997.

AQUINO, F. J. A., ROCHA, C. A. F., e RESENDE, L. S., Widely Linear Prediction for Blind Equalization. In: *IEEE International Conference on Communications - ICC - Glasgow, Scotland*, 1 CD-ROM, junho de 2007.

- GELLI, G., PAURA, L. e RAGOZINI, A. R. P., Blind widely linear multiuser detection. *In: IEEE Commun. Lett.*, vol. 4, no. 6, pp. 187–189, Jun. 2000.
- GERSTACKER, W. H., SCHÖBER, R., e LAMPE, A., Receivers with widely linear processing for frequency-selective channels. *In: IEEE Trans. on Communication*, vol. 51, no. 9, pp. 1512–1523, Sep. 2003.
- HAIMI-COHEN, R., HERZBERG, H., e BE'ERY, Y., Delayed Adaptive LMS Filtering: Current Results. *In: IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 3, pp.:1273 - 1276, abril 1990.
- HAUCK, S., The Roles of FPGA's in Reprogrammable Systems. *In: Proceedings of the IEEE*, vol. 86, no. 4, pp. 615-638, Abril 1998.
- HWANG, J., MILNE, B., SHIRAZI, N., e STROOMER, J., System Level Tools for DSP in FPGAs, Xilinx Inc. 2001.
- LONG, G., LING F., e PROAKIS, J.G., The LMS Algorithm with Delayed Coefficient Adaptation. *In: IEEE Trans. on Acoustics, Speech and Signal Processing*, vol. 37, no. 9, pp. 1397-1405, Set. 1989.
- NEESER, F. D. e MASSEY, J. L. Proper complex random processes with applications to information theory. *In: IEEE Trans. Inform. Theory*, vol. 39, pp. 1293–1302, July 1993.
- PICINBONO, B., On circularity. *In: IEEE Trans. Signal Processing*, vol. 42, pp. 3473-3482, 1994.
- PICINBONO, B. e CHEVALIER, P., Widely linear estimation with complex data. *In: IEEE Trans. Signal Processing*, vol. 43, pp. 2030–2033, Aug. 1995.
- PROAKIS, J. G., **Digital Communications**, McGraw-Hill, 3a. Edição, 1995.
- QURESHI, S.U.H., “Adaptive Equalization”. *In: Proc. IEEE*, vol. 53, pp. 1349-1387, Set., 1985.
- REED, J. H., **Software Radio: A Modern Approach to Radio Engineering**, Prentice Hall, Upper Saddle River, NJ, 2002.
- SLOCK, D.T.M., Blind fractionally-spaced equalization, perfect-reconstruction filter-banks and multichannel linear prediction, *In: Proc. ICASSP*, vol. 4, pp. 585–588, 1994
- TERRIEN, C.W., **Discrete Random Signals and Statistical Signal Processing**, Prentice-Hall International Editions, New Jersey, 1992.
- TOBIAS, O.J., BERMUDEZ, J.C.M., e BERSHAD, N.J., Stochastic analysis of the delayed LMS algorithm for a new model. *In: IEEE Proc. of the International Conference on Acoustics, Speech, and Signal Processing*, ICASSP'00, vol. 1, pp.:404-407, Jun. 2000.
- XILINX Inc. Achieving Breakthrough Performance in Virtex-4 FPGAs. White Paper: Virtex-4 FPGAs. May, 2006.
- XILINX Inc. **System Generator for DSP version 8.1.**, San Jose California: XILINX Inc., 2005.