

# Sobre o Desempenho de Filtros Adaptativos com Coeficientes em Somas de Potências de Dois em Realce de Sinais

Luiz Felipe da S. Coelho, Lisandro Lovisollo e Michel P. Tcheou

**Resumo**— Neste artigo, estudam-se os efeitos da representação numérica por somas de potências de dois (SPT) no desempenho de filtros adaptativos para realce de sinais. A representação SPT permite implementar filtros com baixa complexidade computacional e, portanto, baixo consumo de energia. Neste trabalho, aproximações SPT dos coeficientes de filtros adaptativos são estudadas considerando os filtros LMS (*Least Mean Squares*) e RLS (*Recursive Least Squares*). Consideram-se dois cenários para a aplicação da representação SPT: (i) obtém-se a representação SPT dos coeficientes de filtros já treinados, i.e., quando os filtros adaptativos já convergiram, com o intuito de investigar o deterioramento provocado pela aproximação SPT na resposta do filtro; e (ii) a aproximação SPT dos coeficientes dos filtros é realizada no laço de adaptação do algoritmo, visando investigar sua influência na evolução do processo de adaptação. O desempenho de filtros adaptativos em SPT se mostrou satisfatório. Ademais, o emprego de representações SPT no treinamento de filtros adaptativos LMS e RLS mostrou-se viável, desde que uma complexidade mínima (quantidade de termos SPT) seja imposta.

**Palavras-Chave**— Filtros Adaptativos, Realce de Sinal, Aproximação de Filtros, Somas de Potências de Dois.

**Abstract**— In this article, the effects of the sums of powers of two (SPT) implementation to approximate adaptive filters in a scenario of signal enhancement is studied. SPT-based filters have low computational complexity and therefore low energy costs to implement. Thus, the SPT approximation algorithm is implemented in two different adaptive filters algorithms, the LMS and the RLS. The SPT approximation is studied in two different ways: (i) SPT approximation of the already trained filter coefficients, i.e. when the adaptive filters have already converged, in order to investigate the influence of the SPT approach on the filter response; and (ii) the SPT approximation of the filter coefficients while the adaptation process is done, aiming to investigate its influence on the filter's evolution. The performance of adaptive filters with SPT coefficients has shown to be acceptable. Moreover, it has been found that the SPT format can be used in the training process of LMS and RLS adaptive filters, if a minimal complexity is imposed.

**Keywords**— Adaptive Filters, Signal Enhancement, Filter Approximation, Sums of Powers of Two.

## I. INTRODUÇÃO

Em diversas aplicações, sinais contaminados por ruído são um problema. Uma solução viável e amplamente utilizada para mitigar tal contaminação são as técnicas de realce de sinais, por exemplo, utilizando filtros adaptativos [1]. Com a captura de uma quantidade cada vez maior dos mais diversos tipos de sinais, nos mais diversos tipos de dispositivos, ou realizam-se processamentos cada vez mais eficientes em termos energéticos ou disponibilizam-se fontes de maior capacidade energética. Entretanto, observa-se em geral um misto

das duas estratégias (aumento da energia disponível e redução da complexidade computacional) [2]. Assim, estratégias para diminuir a complexidade do processamento e torná-lo mais eficiente são necessárias. Uma abordagem que permite reduzir a complexidade da implementação de filtros é o uso de representações numéricas alternativas como a de Somas de Potências de Dois (SPT - *Sums of Powers of Two*) que elimina a necessidade do uso de multiplicadores, reduzindo a complexidade computacional [3]–[10]. Consequentemente, neste artigo, investigam-se os possíveis efeitos da aproximação SPT dos coeficientes de filtros adaptativos.

Apresenta-se uma avaliação empírica do impacto que a representação SPT pode provocar no processo de realce de sinais usando filtros adaptativos. Para isso, comparam-se os desempenhos dos melhoramentos obtidos por filtros adaptados com os obtidos por suas versões aproximadas por SPTs. Mais especificamente, avaliam-se os impactos do uso de SPT em filtros adaptativos *Least-Mean-Square* (LMS) e *Recursive Least-Square* (RLS); cabe observar que esses algoritmos de adaptação de coeficientes de filtros são os fundamentos de diversos outros algoritmos de filtragem adaptativa [1].

A Seção II apresenta a aplicação – o realce de sinais usando filtros adaptativos, e sucintamente descreve os algoritmos LMS e RLS. Em seguida, a Seção III discute brevemente a aproximação de filtros de resposta finita ao impulso (FIR - *Finite Impulse Response*) via SPT. Na Seção IV, explicam-se os cenários de simulação; há dois diferentes cenários, que chamamos de *off-line* e de *in-loop*.<sup>1</sup> No cenário *off-line*, aproximam-se filtros já treinados, enquanto no cenário *in-loop* aplica-se a aproximação SPT dos coeficientes conjuntamente com a atualização dos mesmos (isto é, durante a iteração do algoritmo de adaptação). A Seção V apresenta os resultados obtidos por simulação. Por fim, a Seção VI apresenta as conclusões deste trabalho.

## II. REALCE DE SINAIS E FILTROS ADAPTATIVOS

O realce de sinais é um processo que visa à remoção do ruído contido no sinal capturado de forma a enfatizar o sinal de interesse. Seja  $\mathbf{x}$  o sinal de interesse que é corrompido por um ruído Gaussiano branco aditivo (AWGN - *Additive White Gaussian Noise*)  $\mathbf{n}_1$ ; assim, o sinal capturado é  $\mathbf{x} + \mathbf{n}_1$ . Para realçar sinais, deve-se empregar um segundo ruído  $\mathbf{n}_2$  que é processado de forma a estimar o ruído que corrompe o sinal de interesse. Para que isso seja possível,  $\mathbf{n}_2$  e  $\mathbf{n}_1$  devem ser correlacionados. Essa estimativa é obtida usando um filtro adaptativo  $\mathbf{w}$  cuja saída denominaremos  $\mathbf{y}$ . Ao subtraí-la do

Luiz Felipe da S. Coelho, Lisandro Lovisollo e Michel P. Tcheou Universidade do Estado do Rio de Janeiro, Rio de Janeiro-RJ, Brasil, e-mails: lfscoelho@ieec.org, lisandro@uerj.br e mtcheou@uerj.br.

<sup>1</sup>Além dos dois algoritmos e parâmetros como comprimento dos filtros e quantidade de termos SPT.

sinal capturado obtém-se o sinal de erro

$$e(n) = [x(n) + n_1(n)] - y(n), \quad (1)$$

conforme a Figura 1,  $n$  indica o tempo (em amostras). Se  $y(n)$  for igual a  $n_1(n)$  então  $e(n) = x(n)$ ; este sinal é dito de “erro” pois é utilizado no ajuste dos coeficientes do filtro adaptativo  $\mathbf{w}$  de forma estimar  $n_1(n)$  a partir de  $y(n) = \mathbf{n}_2^T(n)\mathbf{w}(n)$ .

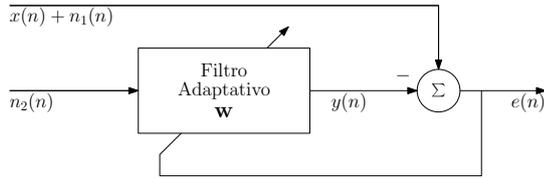


Fig. 1. Diagrama da filtragem adaptativa para o realce de sinais.

Para um realce efetivo, os ruídos  $\mathbf{n}_1$  e  $\mathbf{n}_2$  devem ser fortemente correlacionados e, o ruído  $\mathbf{n}_2$  e o sinal de referência  $\mathbf{x}$  muito fracamente correlacionados (preferencialmente não deve haver qualquer correlação entre eles). Para o erro  $e(n)$ , Equação (1), assumindo que  $\mathbf{x}$  é descorrelacionado de  $\mathbf{n}_1$  e  $\mathbf{n}_2$  obtemos o Erro Quadrático Médio (MSE - *Mean-Square-Error*)

$$\xi(n) = E[e^2(n)] = E[x^2(n)] + E\{[n_1(n) - y(n)]^2\}, \quad (2)$$

cujo mínimo é obtido quando  $y(n) = n_1(n)$  e é

$$\xi_{min}(n) = E[x^2(n)]. \quad (3)$$

As Equações (1)–(3) mostram que, sob certas condições, o esquema da Figura 1 pode resultar no valor ótimo, onde  $\mathbf{e} = \mathbf{x}$ . Na saída, a energia do sinal de interesse ( $\mathbf{x}$ ) é mantida, enquanto energia do ruído de interferência é atenuada, aumentando a Razão Sinal-Ruído (SNR - *Signal-to-Noise Ratio*)

$$\text{SNR} = 10 \log_{10} \left( \frac{P_x}{P_n} \right) \quad [\text{dB}], \quad (4)$$

onde  $P_x$  e  $P_n$  são respectivamente a potência do sinal  $\mathbf{x}$  e do ruído  $\mathbf{n}$ , presentes no sinal corrompido.

#### A. Least-Mean-Squares

A Equação (3) indica que à medida em que  $\mathbf{y}$  se aproxima de  $\mathbf{n}_1$ , mais próximo  $\mathbf{e}$  é de  $\mathbf{x}$ . Uma forma de adaptar o filtro  $\mathbf{w}$  para isso é através do algoritmo LMS. Neste caso, aproxima-se o gradiente da superfície MSE através da derivada em relação a  $\mathbf{w}$  [1], [11]. Sejam  $\mathbf{w}_{\text{LMS}}(k)$  um vetor coluna contendo os  $N$  coeficientes do filtro adaptativo de ordem  $N - 1$  na iteração  $k$ , e  $\mathbf{n}_2$  um vetor coluna contendo as  $N$  últimas amostras do ruído  $\mathbf{n}_2$ . Atualiza-se  $\mathbf{w}(k)$  usando

$$e_{\text{LMS}}(k) = [x(k) + n_1(k)] - \mathbf{n}_2^T(k)\mathbf{w}_{\text{LMS}}(k), \quad (5)$$

$$\mathbf{w}_{\text{LMS}}(k+1) = \mathbf{w}_{\text{LMS}}(k) + 2\mu e_{\text{LMS}}(k)\mathbf{n}_2^T(k), \quad (6)$$

onde  $\mu$  é o fator de atualização, e  $\mathbf{v}^T$  é a transposição de  $\mathbf{v}$ .

#### B. Recursive Least Squares

Alternativamente, pode-se empregar o algoritmo RLS para obter  $\mathbf{w}$ . A Equação (2) pode ser modificada para

$$\xi^d(k) = \sum_{i=0}^k \lambda^{k-i} e_{\text{RLS}}^2(i) = \sum_{i=0}^k \lambda^{k-i} [d(i) - \mathbf{n}_2^T(i)\mathbf{w}_{\text{RLS}}(k)]^2, \quad (7)$$

onde  $\mathbf{w}_{\text{RLS}}(k)$  é o vetor coluna de coeficientes do filtro adaptativo,  $\mathbf{n}_2$  é uma linha de atraso análoga à do LMS e  $e_{\text{RLS}}(i)$  é o sinal erro de saída *a posteriori* no instante  $i$  [1]. Na Equação (7), o parâmetro  $\lambda$  é o fator de esquecimento e  $0 \ll \lambda < 1$  – quanto mais próximo a 1, maior é o peso de entradas passadas na atualização dos coeficientes (a força da recursividade) [1].

O vetor  $\mathbf{w}_{\text{RLS}}(k)$  que minimiza a Equação (7) é obtido via

$$\mathbf{S}_D(k) = \frac{1}{\lambda} \left[ \mathbf{S}_D(k-1) - \frac{\mathbf{S}_D(k-1)\mathbf{n}_2(k)\mathbf{n}_2^T(k)\mathbf{S}_D(k-1)}{\lambda + \mathbf{n}_2^T(k)\mathbf{S}_D(k-1)\mathbf{n}_2(k)} \right], \quad (8)$$

$$\mathbf{p}_D(k) = \lambda \mathbf{p}_D(k-1) + [x(k) + n_1(k)]\mathbf{n}_2(k), \quad (9)$$

$$\mathbf{w}_{\text{RLS}}(k) = \mathbf{S}_D(k)\mathbf{p}_D(k). \quad (10)$$

Acima,  $\mathbf{S}_D(k)$  é o inverso da matriz de autocorrelação do ruído de entrada  $\mathbf{n}_2(k)$  e  $\mathbf{p}_D(k)$  é o vetor de correlação cruzada entre  $\mathbf{n}_2(k)$  e  $x(k) + n_1(k)$ . E, analogamente ao apresentado para o LMS (Equação (5)), o erro *a posteriori* do RLS será

$$e_{\text{RLS}}(k) = [x(k) + n_1(k)] - \mathbf{n}_2^T(k)\mathbf{w}_{\text{RLS}}(k). \quad (11)$$

### III. APROXIMAÇÃO DE FILTROS USANDO SPT

Foram vistos até aqui, algoritmos adaptativos para minimização do MSE entre o sinal de entrada e o desejado que resultam em um vetor de coeficientes melhorado  $\mathbf{w}(n)$ . Os valores dos coeficientes não tem nenhuma limitação de representação, entretanto, na prática haverá efeitos devido à quantização inerente às máquinas digitais [1]. Neste trabalho investiga-se o impacto da representação SPT [3]–[7] em filtros adaptativos [12].

Na representação em SPT empregam-se dígitos ternários (que podem valer  $-1, 0$  e  $1$ ), tal que um valor  $v$  é dado por

$$v = \sum_{j=1}^B b_j 2^{-j}, \quad \text{com } b_j \in \{-1, 0, 1\}, \quad (12)$$

onde  $B$  é a profundidade da representação. Em SPT, alguns números podem ser representados utilizando uma quantidade menor de dígitos relativamente à representação binária. Ao mesmo tempo, a representação SPT mantém a simplicidade aritmética. Podemos impor uma complexidade máxima (quantidade de termos  $b_j$  não nulos) à representação, às custas de um erro aceitável [7]. A compacidade de representação associada à simplicidade dos circuitos necessários para as operações de multiplicação (poucas somas) inerentes à implementação de filtros (convolução) permitem implementações eficientes de filtros digitais com complexidade computacional reduzida.

Considere um filtro FIR de coeficientes  $h(n)$ ,  $n = 1, 2, \dots, N$ , i.e., de ordem  $N - 1$ , e versões SPT  $\tilde{h}(n) =$

$\sum_{j=1}^B b_j(n)2^{-j}$ . A imposição de uma quantidade máxima de termos SPT  $M$  na implementação gera o problema otimização

$$\min \sum_{n=1}^N \left[ h(n) - \sum_{j=1}^B 2^{-j} b_j(n) \right]^2 \text{ s.t. } \sum_{n=1}^N \sum_{j=1}^B |b_j(n)| \leq M. \quad (13)$$

Este é um problema de otimização combinatorial, cujo objetivo é encontrar o filtro mais próximo a  $\mathbf{h}$  usando  $M$  ou menos termos SPT.

Uma forma de resolver o problema apresentado na Equação (13) é através do algoritmo de *Matching Pursuits with Generalized Bit-Plane* (MPGBP) [7]. O MPGBP é um algoritmo iterativo de aproximação em planos de *bits* generalizados [7]. O MPGBP é inspirado no *Matching Pursuits* (MP) [13] e decompõe iterativamente um vetor  $\mathbf{v} \in \mathbb{R}^N$  para fornecer uma aproximação com  $M$  termos  $\mathbf{v}^{(M)}$  de  $\mathbf{v}$

$$\mathbf{v} \approx \mathbf{v}^{(M)} = \sum_{i=1}^M \alpha^{k_i} \mathbf{c}_{l_i}. \quad (14)$$

onde  $\alpha^{k_i}$  é o peso do  $i$ -ésimo *codeword*  $\mathbf{c}_{l_i}$  e  $l_i$  é índice do *codeword* do dicionário  $\mathcal{D}$ , que contém  $\#\mathcal{D}$  *codewords*, i.e.  $\mathcal{D} = \{\mathbf{c}_l\}_{l=1, \dots, \#\mathcal{D}}$ .

A cada iteração do MPGBP, procura-se o *codeword* que mais se parece com o resíduo (ou o que falta representar). Para isso, calculam-se os produtos internos entre o sinal e os *codewords* e aquele correspondente ao maior produto interno é ponderado pelo produto interno e subtraído do resíduo, obtendo assim, o resíduo para a próxima iteração. Se  $\alpha = 1/2$  e os *codewords* possuem apenas coordenadas  $\pm 1$  e 0 então a expressão (14) é uma solução para o problema de otimização (13). Como qualquer  $\mathbf{c}_l$  possui coordenadas 0, 1 e  $-1$ , consequentemente  $k_i$  define o plano de bits e  $\mathbf{c}_{l_i}$  em quais coordenadas de  $\mathbf{v}$  o plano de bits  $\alpha^{k_i}$  contribui.

O MPGBP está esquematizado no Algoritmo 1 sucintamente. Ele impõe a restrição de que um *codeword* deve ter  $P = \lfloor \sqrt{N} \rfloor$  coordenadas não-nulas<sup>2</sup>, reduzindo a complexidade da busca (ver Algoritmo 1) e garantindo o menor erro de aproximação em função da quantidade de iterações. No algoritmo, a quantidade  $M$  de termos SPT na aproximação de um filtro  $\mathbf{w}$  pode ser avaliada e utilizada como critério de parada, impondo assim um limite à quantidade de somas a usar na implementação do filtro. Além disso, a quantidade de planos de *bits* distintos empregados na representação dos coeficientes  $B$  também pode ser empregada como critério de parada do algoritmo, caso desejado.

#### IV. SIMULAÇÕES

No sistema da Figura 1 considera-se que o sinal de interesse  $\mathbf{x}$  é um áudio (amostrado a 8192 Hz) ao qual é adicionado um ruído  $\mathbf{n}_1$  branco e Gaussiano. Por sua vez,  $\mathbf{n}_2$  é obtido a partir de um processo Auto-Regressivo (AR) de primeira ordem aplicado a  $\mathbf{n}_1$ , com os coeficientes  $[1 \ -0,95]$ . Tal AR tem resposta equivalente a de um filtro IIR (*Infinite Impulse Response*). Sua resposta  $H(z)$  é caracterizada por um polo. Enquanto que filtros FIR (caso dos filtros adaptativos LMS ou RLS) têm resposta  $W(z)$  caracterizada somente por zeros.

<sup>2</sup>Os operadores  $\lfloor \cdot \rfloor$  e  $\lceil \cdot \rceil$  retornam, respectivamente, o maior inteiro menor que e o menor inteiro maior que seus argumentos.

#### Algoritmo 1 MPGBP para Aproximação de filtros em SPT

**Entrada:** vetor/filtro  $\mathbf{w}$  e a quant. máxima de somas  $M$ .

**Saída:** vetor aprox.  $\mathbf{r}_M$  e profundidade do planos de *bits*  $B$ .

```

i = 1,  $\mathbf{r}_1 = \mathbf{w}$ 
enquanto i < M faça
  Iteração MPGBP:
   $\mathbf{c}_i \leftarrow \mathbf{0}$ 
  Ordenam-se as coordenadas de  $\mathbf{r}_i$  em ordem decrescente de seus
  valores absolutos
   $\mathcal{S}_P \leftarrow$  índices das P maiores coordenadas
  para todo  $n \in \mathcal{S}_P$  faça
    se  $r_i(n) > 0$  então  $c_i(n) = 1$  senão  $c_i(n) = -1$ 
  fim para
   $k_i \leftarrow -\lceil \log_2 \left( \frac{4}{3^P} \sum_{n \in \mathcal{S}_P} |r_i(n)| \right) \rceil$ 
   $\mathbf{r}_{i+1} \leftarrow \mathbf{r}_i - 2^{-k_i} \mathbf{c}_i$ 
   $B \leftarrow \max_i k_i$ 
   $i \leftarrow i + 1$ 
fim enquanto
    
```

Assim, espera-se que um filtro FIR de ordem um, seja capaz de reverter o processo e estimar  $\mathbf{n}_1$  corretamente. Ambos os ruídos tiveram a potência ajustada para uma SNR de 10 dB.

Uma distinção importante deve ser mencionada relativa às implementações SPT dos filtros, vislumbram-se dois diferentes cenários.

*a) SPT off-line:* a representação SPT é aplicada após o treinamento dos filtros. O Algoritmo 1 é usado nas aproximações dos filtros LMS e RLS. Como notação, utilizaram-se  $\tilde{\mathbf{w}}_{\text{LMS}}$  e  $\tilde{\mathbf{w}}_{\text{RLS}}$ , respectivamente, para os filtros LMS e RLS aproximados. Nesse cenário, empregaram-se filtros de diferentes comprimentos  $N$  e realizaram-se aproximações SPT com diferentes quantidades  $M$  somas permitidas.

*b) SPT in-loop:* a representação SPT é imposta durante a adaptação dos filtros. A atualização dos coeficientes do filtro, Equações (6) e (10), considera filtros cujos coeficientes já estão representados usando SPT. Como resultado, para o LMS temos as equações de atualização (análogas a (5) e (6))

$$\tilde{\epsilon}_{\text{LMS}}(k) = [x(k) + n_1(k)] - \mathbf{n}_2^T(k) \tilde{\mathbf{w}}_{\text{LMS}}(k), \quad (15)$$

$$\tilde{\mathbf{w}}_{\text{LMS}}(k+1) = \tilde{\mathbf{w}}_{\text{LMS}}(k) + 2\mu \tilde{\epsilon}_{\text{LMS}}(k) \mathbf{n}_2^T(k), \quad (16)$$

$$\tilde{\mathbf{w}}_{\text{LMS}}(k+1) = \text{SPT}\{\tilde{\mathbf{w}}_{\text{LMS}}(k+1)\}, \quad (17)$$

nas quais,  $\tilde{\mathbf{w}}(k)$  representa o vetor de coeficientes aproximados na iteração  $k$  e  $\tilde{\mathbf{w}}(k+1)$  é uma notação auxiliar para elucidar a obtenção  $\tilde{\mathbf{w}}(k+1)$  via a aproximação SPT.

No caso do algoritmo RLS, as Equações análogas às de atualização (8), (9), (10) e (11), ensejam mais aproximações SPT e por sua vez mais termos SPT serão necessários na implementação. Teremos

$$\check{\mathbf{S}}_D(k) = \frac{1}{\lambda} \left[ \check{\mathbf{S}}_D(k-1) - \frac{\check{\mathbf{S}}_D(k-1) \mathbf{n}_2(k) \mathbf{n}_2^T(k) \check{\mathbf{S}}_D(k-1)}{\lambda + \mathbf{n}_2^T(k) \check{\mathbf{S}}_D(k-1) \mathbf{n}_2(k)} \right], \quad (18)$$

$$\check{\mathbf{p}}_D(k) = \lambda \check{\mathbf{p}}_D(k-1) + [x(k) + n_1(k)] \mathbf{n}_2(k), \quad (19)$$

$$\check{\mathbf{p}}_D(k-1) = \text{SPT}\{\check{\mathbf{p}}_D(k-1)\} \text{ e } \check{\mathbf{S}}_D(k) = \text{SPT}\{\check{\mathbf{S}}_D(k)\} \quad (20)$$

$$\check{\mathbf{w}}_{\text{RLS}}(k) = \check{\mathbf{S}}_D(k) \check{\mathbf{p}}_D(k), \text{ e } \check{\mathbf{w}}_{\text{RLS}}(k) = \text{SPT}\{\check{\mathbf{w}}_{\text{RLS}}(k)\} \quad (21)$$

$$\tilde{\epsilon}_{\text{RLS}}(k) = [x(k) + n_1(k)] - \mathbf{n}_2^T(k) \check{\mathbf{w}}_{\text{RLS}}(k). \quad (22)$$

## V. RESULTADOS

Primeiramente, foram avaliados os desvios das respostas em frequências dos filtros LMS e RLS aproximados usando SPT *off-line* e *in-loop* (utilizando o MSE em dB) dos filtros originais. Os desvios são apresentados nas Figuras 2 e 3 para casos LMS e RLS, respectivamente. Avaliou-se também o SNR, Equação (4), do sinal realçado, para o treinamento LMS, Figura 4, e RLS, Figura 5, nos cenários *off-line* e *in-loop*.

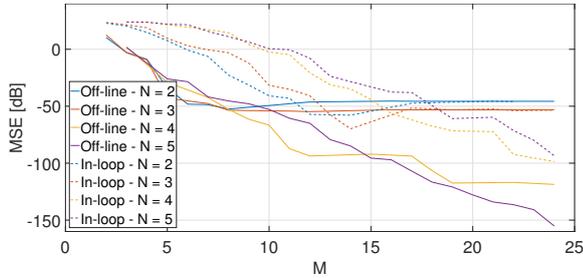


Fig. 2. MSE entre a resposta em frequência do filtro LMS original e as respostas em frequência dos filtros aproximados via *off-line* e *in-loop*, em função da quantidade  $M$  de termos SPT, para filtros com  $N = 2, 3, 4$  e  $5$ .

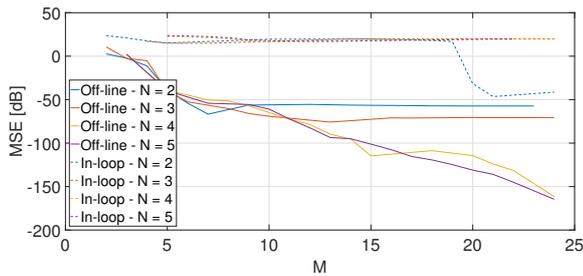


Fig. 3. MSE entre a resposta em frequência do filtro RLS original e as respostas em frequência dos filtros aproximados via *off-line* e *in-loop*, em função da quantidade  $M$  de termos SPT, para filtros com  $N = 2, 3, 4$  e  $5$ .

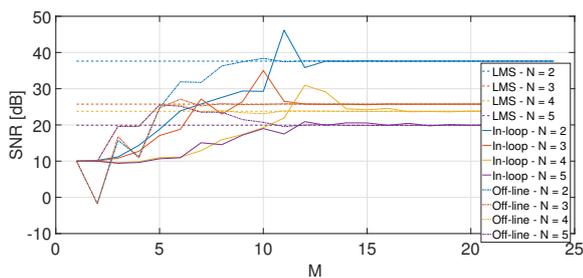


Fig. 4. SNR nos sinais já realçados para o processo utilizando LMS e suas aproximações em função da quantidade  $M$  de termos SPT, para filtros com  $N = 2, 3, 4$  e  $5$ .

Dos resultados, observa-se que o aumento da quantidade  $M$  de SPT implica na melhora da aproximação dos filtros LMS e RLS, tanto no cenário *off-line* quanto no *in-loop*. Observa-se porém que, pela natureza do RLS, o cenário *in-loop* requer mais termos SPT para atingir o mesmo nível de desempenho. Isso é bem evidenciado ao se observar a SNR em função de  $M$  para o filtro com  $N = 2$ , onde em  $M = 20$  observa-se um melhoramento abrupto do desempenho da aproximação, ver Figura 5. Nota-se ainda que filtros mais longos exigem mais

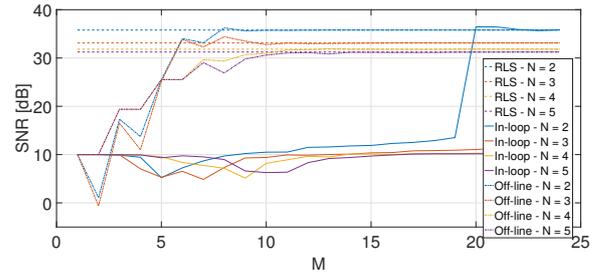


Fig. 5. SNR nos sinais já realçados para o processo utilizando RLS e suas aproximações em função da quantidade  $M$  de termos SPT, em função da quantidade  $M$  de termos SPT, para filtros com  $N = 2, 3, 4$  e  $5$ .

termos SPTs para atingir a mesma qualidade (tanto em termos de MSE da resposta como do SNR do sinal realçado).

Avaliou-se também a profundidade de planos de bits inerentes às implementações, para filtros LMS (Figura 6) e RLS (Figura 7), nos cenários *off-line* e *in-loop*. Vê-se que o aumento na quantidade  $M$  de SPT incorre em aumentos na profundidade do plano de bits.

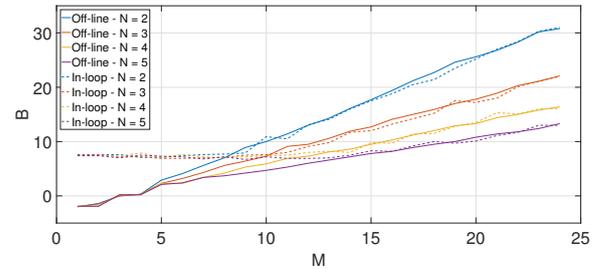


Fig. 6. Profundidade  $B$  do plano de bits para os filtros aproximados, em função da quantidade  $M$  de termos SPT, para filtros LMS com  $N = 2, 3, 4$  e  $5$ .

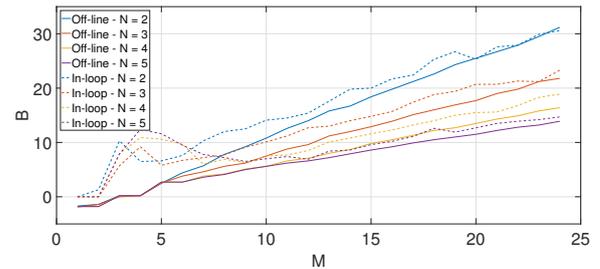


Fig. 7. Profundidade  $B$  do plano de bits para os filtros aproximados, em função da quantidade  $M$  de termos SPT, para filtros RLS com  $N = 2, 3, 4$  e  $5$ .

As avaliações anteriores permitem conhecer os efeitos finais do emprego da representação SPT para coeficientes de filtros adaptativos. Porém, o cenário *in-loop* requer análises específicas de forma a avaliar o impacto da representação SPT no processo de treinamento/adaptação dos filtros.

### A. LMS

A evolução dos coeficientes do filtro LMS quando emprega-se a representação SPT *in-loop* é apresentada na Figura 8, para  $N = 2$  e  $M = 11$  e  $16$ . A utilização de poucas SPT pode impactar severamente a aplicação, e até impedir a convergência

do filtro, a curva da Figura 9 para  $M = 6$  indica essa dificuldade, já para  $M = 11$  e  $16$  vemos que as aproximações *in-loop* convergem mais rápida e satisfatoriamente.

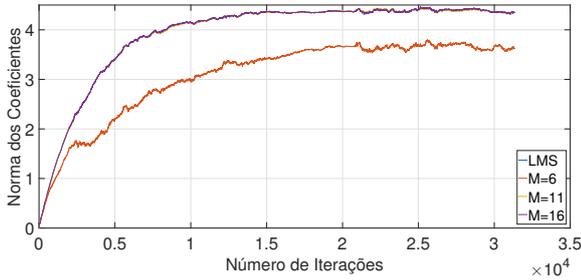


Fig. 8. Evolução da Norma dos Coeficientes do filtro LMS e de sua aproximação *in-loop*, para  $N = 2$  e  $M = 6, 11$  e  $16$

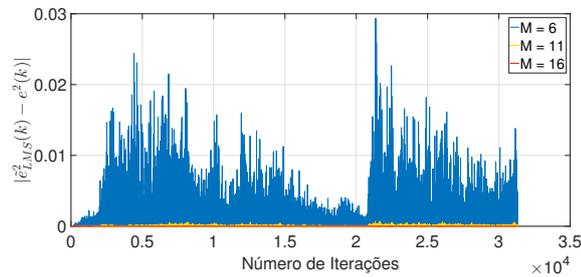


Fig. 9. Diferença entre a energia dos sinais pós-realce, de acordo com a evolução dos filtros LMS e sua aproximação *in-loop*, para  $N = 2$  e aproximações com  $M = 6, 11$  e  $16$

## B. RLS

A Figura 10 apresenta a evolução dos coeficientes do filtro RLS quando emprega-se a representação SPT *in-loop*. Verifica-se que o algoritmo RLS requer a utilização de mais SPT que o LMS, para um mesmo  $N$ , por exemplo, em  $M = 16$  o RLS não converge. Isso é corroborado pelas curvas na Figura 11, na qual se observa que a diferença da energia dos sinais pós-realce no RLS é claramente maior que no LMS.

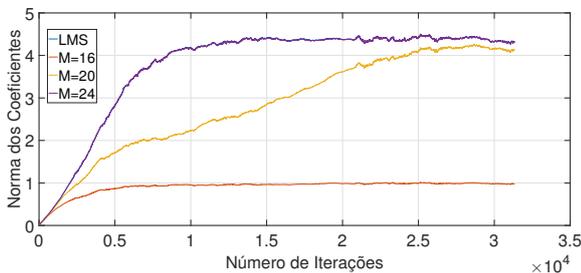


Fig. 10. Evolução da Norma dos Coeficientes do filtro RLS e de sua aproximação *in-loop*, para  $N = 2$  e  $M = 16, 20$  e  $24$ .

## VI. CONCLUSÕES

Nesse artigo, avaliaram-se, empiricamente, os efeitos da representação em SPT para filtros adaptativos. Os resultados mostram que filtros obtidos através dos algoritmos LMS ou RLS podem ter seus coeficientes aproximados via SPT, sem

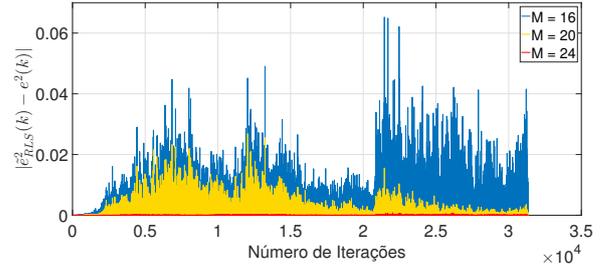


Fig. 11. Diferença entre a energia dos sinais pós-realce, de acordo com a evolução dos filtros RLS e sua aproximação *in-loop*, com filtros de  $N = 2$  e aproximações com  $M = 16, 20$  e  $24$ .

grande perda de desempenho. Mostram ainda que, no cenário *in-loop*, é possível treinar filtros impondo restrições à quantidade de SPT. Para filtros LMS e RLS, os impactos negativos da aproximação têm relação inversa com a quantidade  $M$  de somas permitidas – a complexidade da implementação. Além disso, percebeu-se que, no cenário *in-loop*, as aproximações para o RLS requerem maior complexidade que para o LMS. Aqui pretende-se contribuir para o conhecimento do impacto de representações SPT em filtros adaptativos, expondo alguns resultados simples que aspiram à melhor compreensão do assunto.

## REFERÊNCIAS

- [1] P. S. Diniz, *Adaptive filtering*. Springer, 1997.
- [2] A. Mammela and A. Anttonen, “Why will computing power need particular attention in future wireless devices?” *IEEE Circuits and Systems Magazine*, vol. 17, no. 1, pp. 12–26, 2017.
- [3] Y. C. Lim, S. Parker, and A. Constantinides, “Finite word length FIR filter design using integer programming over a discrete coefficient space,” *Acoustics, Speech and Signal Processing, IEEE Transactions on*, vol. 30, no. 4, pp. 661–664, aug. 1982.
- [4] Y. C. Lim and S. Parker, “FIR filter design over a discrete powers-of-two coefficient space,” *Acoustics, Speech and Signal Processing, IEEE Transactions on*, vol. 31, no. 3, pp. 583–591, jun. 1983.
- [5] J. Yli-Kaakinen and T. Saramaki, “A systematic algorithm for the design of multiplierless FIR filters,” in *Circuits and Systems, 2001. ISCAS 2001. The 2001 IEEE International Symposium on*, vol. 2, 6-9 2001, pp. 185–188 vol. 2.
- [6] P. Gentili, F. Piazza, and A. Uncini, “Efficient genetic algorithm design for power-of-two FIR filters,” in *Acoustics, Speech, and Signal Processing, 1995. ICASSP-95., 1995 International Conference on*, vol. 2, may 1995, pp. 1268–1271 vol.2.
- [7] E. A. da Silva, L. Lovisolo, A. J. Dutra, and P. S. Diniz, “FIR filter design based on successive approximation of vectors,” *IEEE Transactions on Signal Processing*, vol. 62, no. 15, pp. 3833–3848, 2014.
- [8] H. Samueli, “An improved search algorithm for the design of multiplierless FIR filters with powers-of-two coefficients,” *Circuits and Systems, IEEE Transactions on*, vol. 36, no. 7, pp. 1044–1047, jul. 1989.
- [9] J. Evans, “Efficient FIR filter architectures suitable for fpga implementation,” *Circuits and Systems II: Analog and Digital Signal Processing, IEEE Transactions on*, vol. 41, no. 7, pp. 490–493, jul. 1994.
- [10] M. Macleod and A. Dempster, “Multiplierless FIR filter design algorithms,” *Signal Processing Letters, IEEE*, vol. 12, no. 3, pp. 186–189, march 2005.
- [11] B. Widrow, J. M. McCool, M. G. Larimore, and C. R. Johnson, “Stationary and nonstationary learning characteristics of the LMS adaptive filter,” *Proceedings of the IEEE*, vol. 64, no. 8, pp. 1151–1162, 1976.
- [12] S. Choudhary, P. Mukherjee, M. Chakraborty, and S. S. Rath, “A SPT treatment to the realization of the sign-LMS based adaptive filters,” *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 59, no. 9, pp. 2025–2033, 2012.
- [13] S. G. Mallat and Z. Zhang, “Matching pursuits with time-frequency dictionaries,” *IEEE Transactions on signal processing*, vol. 41, no. 12, pp. 3397–3415, 1993.