

# IE009 - Processamento Adaptativo de Sinais

## Filtro de Volterra

O emprego de estruturas lineares para realizar tarefas de extração de informação, tanto no caso supervisionado, como no caso não-supervisionado, está diretamente associado ao compromisso entre tratabilidade matemática e desempenho que estes modelos oferecem. Entretanto, à medida que surgem cenários cada vez mais complexos, especialmente no contexto de processamento de sinais, passa a ser essencial recorrer a estruturas de caráter não-linear, as quais oferecem uma maior flexibilidade para criar o mapeamento das entradas do sistema para as suas saídas.

Neste tópico, vamos considerar, primeiramente, uma estrutura não-linear conhecida como filtro de Volterra, a qual baseia-se em expansões polinomiais do sinal de entrada para gerar a saída, como revela a expressão a seguir:

$$\begin{aligned} y(n) = & h_0 + \sum_{k_1} h_1(k_1)x(n - k_1) + \sum_{k_1} \sum_{k_2} h_2(k_1, k_2)x(n - k_1)x(n - k_2) \\ & + \sum_{k_1} \sum_{k_2} \sum_{k_3} h_3(k_1, k_2, k_3)x(n - k_1)x(n - k_2)x(n - k_3) + \dots, \end{aligned} \quad (1)$$

onde  $y(n)$  denota a saída deste sistema SISO não-linear,  $h(k_1, \dots, k_j)$  são os coeficientes do filtro, denominados *Volterra kernels*, e  $x(n)$  representa a entrada do sistema. Podemos dizer que a série de Volterra representa uma extensão da idéia da série de Taylor, uma vez que permite a modelagem de respostas dotadas de memória. Conseqüentemente, o filtro de Volterra é uma ferramenta que pode ser utilizada para modelar uma ampla classe de sistemas não-lineares, assim como a série de Taylor se mostra útil para expressar uma variedade de funções não-lineares.

Dada a equivalência entre os paradigmas de filtros lineares (filtragem temporal) e combinadores lineares (estimação linear), é possível escrever uma representação via série de Volterra para o caso em que os sinais presentes no vetor de entrada  $\mathbf{x}$  provêm de diferentes origens, como ocorre, por exemplo, em um arranjo de sensores:

$$\begin{aligned} y(n) = & h_0 + \sum_{k_1} h_1(k_1)x_{k_1}(n) + \sum_{k_1} \sum_{k_2} h_2(k_1, k_2)x_{k_1}(n)x_{k_2}(n) \\ & + \sum_{k_1} \sum_{k_2} \sum_{k_3} h_3(k_1, k_2, k_3)x_{k_1}(n)x_{k_2}(n)x_{k_3}(n) + \dots \end{aligned} \quad (2)$$

Considere que o número de entradas lineares seja igual a  $N$ . Em ambos os casos, podemos adotar uma notação vetorial, de modo a obter uma expressão para a saída do filtro bastante familiar à que

encontramos no cenário de filtragem ótima. Seja

$$\mathbf{x}_v(n) = \begin{bmatrix} x(n) \\ x(n-1) \\ \vdots \\ x(n-N+1) \\ x^2(n) \\ x(n)x(n-1) \\ \vdots \\ x(n)x(n-N+1) \\ x(n-1)x(n) \\ \vdots \\ x(n-N+1)x(n-N+2) \\ x(n-N+1)x(n-N+1) \\ x^3(n) \\ x(n)x(n)x(n-1) \\ \vdots \end{bmatrix} \quad (3)$$

o vetor com todos os termos polinomiais associados à entrada linear  $\mathbf{x}(n) = [x(n) \dots x(n-N+1)]^T$ , e

$$\mathbf{H} = \begin{bmatrix} h_1(0) \\ h_1(1) \\ \vdots \\ h_1(N-1) \\ h_2(0,0) \\ h_2(0,1) \\ \vdots \\ h_2(0,N-1) \\ h_2(1,0) \\ \vdots \\ h_2(N-1,N-1) \\ h_3(0,0,0) \\ \vdots \end{bmatrix} \quad (4)$$

o vetor contendo os *kernels* do filtro de Volterra<sup>1 2</sup>. Com isto, a saída do filtro pode ser escrita como<sup>3</sup>:

$$y_v(n) = \mathbf{H}^T \mathbf{x}_v(n). \quad (5)$$

Observe que a saída do filtro de Volterra é dada pela combinação linear dos sinais de entrada, assim como ocorre no caso de um filtro FIR. A diferença está na natureza do vetor de entrada  $\mathbf{x}_v(n)$ ,

<sup>1</sup>Embora a série de Volterra inclua um termo constante  $h_0$ , ele é irrelevante para esta discussão, e por isso foi omitido na montagem do vetor  $\mathbf{H}$ .

<sup>2</sup>No caso de uma filtragem espacial, basta substituir  $x(n-k)$  por  $x_k(n)$ ,  $k = 0, \dots, N-1$ , em todos os termos.

<sup>3</sup>A Equação (5) permite o cálculo simultâneo da saída do filtro de Volterra para diferentes instantes de tempo (amostras). Para isto, basta montar a matriz  $\mathbf{X}_v = [\mathbf{x}_v(0) \mathbf{x}_v(1) \dots \mathbf{x}_v(T_s-1)]$ , onde  $T_s$  indica o número total de amostras.

que agora inclui termos não-lineares referentes aos produtos cruzados entre as entradas lineares até uma determinada ordem  $M$ . Isto significa que o filtro de Volterra, embora seja não-linear, possui uma dependência linear com respeito aos seus parâmetros ajustáveis  $h(k_1, \dots, k_M)$ . Portanto, é possível obter uma solução fechada para os coeficientes deste filtro que minimiza a esperança do erro quadrático (i.e., no sentido MMSE). Ou seja, os conceitos de filtragem ótima e solução de Wiener podem ser estendidos para o filtro de Volterra, de modo que seus coeficientes ótimos são dados por:

$$\mathbf{H} = \mathbf{R}_{\mathbf{x}_v}^{-1} \mathbf{p}_{\mathbf{x}_v d}, \quad (6)$$

onde  $\mathbf{R}_{\mathbf{x}_v} = E\{\mathbf{x}_v(n)\mathbf{x}_v(n)^T\}$  e  $\mathbf{p}_{\mathbf{x}_v d} = E\{\mathbf{x}_v(n)d(n)\}$ , sendo  $d(n)$  o sinal que desejamos obter na saída do filtro.

Contudo, não podemos confundir esta solução ótima, e os termos nela envolvidos, com a solução de Wiener (clássica) determinada para um filtro FIR. Neste último caso, apenas as estatísticas de segunda ordem do sinal de entrada são utilizadas na matriz de autocorrelação, que possui uma estrutura do tipo Toeplitz. Já no caso do filtro de Volterra, a matriz  $\mathbf{R}_{\mathbf{x}_v}$  possui os seguintes elementos:

$$\mathbf{R}_{\mathbf{x}_v} = \begin{bmatrix} E\{x^2(n)\} & \cdots & E\{x(n)x(n-N+1)\} & E\{x^3(n)\} & \cdots & E\{x(n)x^2(n-N+1)\} & E\{x^4(n)\} & \cdots \\ E\{x(n-1)x(n)\} & \cdots & E\{x(n-1)x(n-N+1)\} & \vdots & & \vdots & \vdots & \vdots \\ \vdots & & \vdots & \vdots & & \vdots & \vdots & \vdots \\ \frac{E\{x(n-N+1)x(n)\}}{E\{x^3(n)\}} & \cdots & \frac{E\{x^2(n-N+1)\}}{E\{x^2(n)x(n-N+1)\}} & \frac{E\{x(n-N+1)x^2(n)\}}{E\{x^4(n)\}} & \cdots & \frac{E\{x^3(n-N+1)\}}{E\{x^2(n)x^2(n-N+1)\}} & \frac{E\{x(n-N+1)x^3(n)\}}{E\{x^5(n)\}} & \cdots \\ \vdots & & \vdots & \vdots & & \vdots & \vdots & \vdots \\ E\{x^2(n-N+1)x(n)\} & \cdots & E\{x^3(n-N+1)\} & E\{x^2(n-N+1)x^2(n)\} & \cdots & E\{x^4(n-N+1)\} & E\{x^2(n-N+1)x^3(n)\} & \cdots \\ \vdots & & \vdots & \vdots & & \vdots & \vdots & \vdots \end{bmatrix}$$

Observe que o primeiro quadrante destacado na matriz  $\mathbf{R}_{\mathbf{x}_v}$  corresponde precisamente à matriz de autocorrelação do caso linear, a qual contém as estatísticas de segunda ordem do sinal de entrada. Por outro lado, podemos perceber que surgem estatísticas de ordem superior, o que significa que o filtro de Volterra, ao considerar os termos polinomiais, consegue explorar com maior profundidade as propriedades estatísticas do sinal de entrada. Com efeito, utilizando um filtro de Volterra:

- de ordem  $M = 2$ , passamos a utilizar estatísticas de  $x(n)$  até a 4ª ordem;
- de ordem  $M = 3$ , passamos a utilizar estatísticas de  $x(n)$  até a 6ª ordem;
- de ordem  $M$  qualquer, surgem as estatísticas de ordem  $2M$ .

Esta evidência, aliada ao fato de que a relação não-linear estabelecida entre a entrada e a saída, por meio da série truncada de Volterra, é linear com respeito aos parâmetros ajustáveis  $h_i(\cdot)$ , encorajam o uso do filtro de Volterra como uma alternativa interessante aos filtros lineares.

No entanto, o perfil da série de Volterra é responsável por gerar uma constante ameaça de explosão de dimensionalidade diante do aumento gradual da ordem de aproximação e do número de entradas fornecidas ao filtro. Em outras palavras, conforme aumentamos a ordem dos termos utilizados, o número de parâmetros ajustáveis cresce significativamente; além disso, o acréscimo de entradas

também leva a um aumento significativo no número de *kernels*. Com efeito, é possível expressar o número de parâmetros (*kernels*) não-ambíguos (e.g., usamos apenas  $E\{x(n)x(n-1)\}$ , já que é igual a  $E\{x(n-1)x(n)\}$  quando  $x(n)$  é real) em função do número de entradas lineares ( $N$ ), como mostra a Equação (7).

$$N_{ker} = N + \frac{N(N+1)}{2} + \frac{N(N+1)(N+2)}{6} + \dots \quad (7)$$

Esta possibilidade de rápida expansão do número de parâmetros traz constantes preocupações referentes à aplicação prática desta estrutura. Apesar disto, se restringirmos a ordem do filtro para valores pequenos, é possível manter o número de coeficientes dentro de um limite razoável e, ao mesmo tempo, obter um avanço no sentido de explorar as estatísticas do sinal de entrada, para, então, produzir um mapeamento mais flexível e adequado quando comparado a um filtro linear.

Além disso, uma vez que mostramos que o filtro de Volterra admite uma solução ótima do tipo Wiener, podemos também realizar sua adaptação em tempo real, utilizando os algoritmos do tipo LMS e RLS. A diferença, novamente, está apenas na forma do vetor de entrada  $\mathbf{x}_v(n)$ , que agora inclui também os termos polinomiais.

## Exemplo: equalização

Considere um sinal  $s(n)$  do tipo 2-PAM, i.e.,

$$s(n) = \begin{cases} -1 & \text{com probabilidade } 0.5 \\ +1 & \text{com probabilidade } 0.5 \end{cases} \quad (8)$$

Pretendemos transmitir este sinal pelo canal cuja função de sistema é dada por:

$$H(z) = 0.5 + z^{-1} \quad (9)$$

O objetivo é, a partir do sinal efetivamente recebido  $x(n)$ , recuperar a informação sobre qual símbolo foi originalmente enviado pela fonte. Neste exemplo ilustrativo, optamos por não considerar a presença de ruído.

Embora pareça ser um canal relativamente simples, ele não pode ser equalizado com um filtro FIR quando o atraso de equalização é zero, uma vez que, neste caso, os estados do canal não podem ser linearmente separados, o que pode ser verificado na Figura 1.

Portanto, é preciso recorrer a uma estrutura não-linear para produzir um mapeamento suficiente flexível de modo a corretamente separar os estados do canal. Começamos, então, com um filtro de Volterra de 2ª ordem, e com duas entradas lineares:  $x(n)$  e  $x(n-1)$ . Logo, temos  $2 + 2 \times (3)/2 = 5$  coeficientes para ajustar. Calculando as estatísticas necessárias para montar a matriz  $\mathbf{R}_{\mathbf{x}_v}$  e o vetor

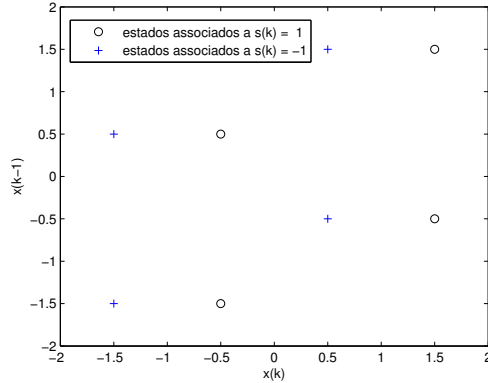


Figura 1: Estados do canal  $H(z) = 0.5 + z^{-1}$

de correlação cruzada  $\mathbf{p}_{\mathbf{x}\mathbf{v}\mathbf{d}}$ , obtemos a seguinte solução ótima:

$$\mathbf{H} = \begin{bmatrix} 0.4762 \\ -0.1905 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (10)$$

Observe que a solução ótima anula os coeficientes dos termos quadráticos, o que significa que o filtro obtido nada mais é do que um filtro FIR. Isto ocorreu por causa do efeito dos termos quadráticos sobre a simetria ímpar existente no sinal original  $s(n)$  e que se reflete também nos estados do canal, como mostrado na Figura 1: os termos polinomiais  $x^2(n)$ ,  $x(n)x(n-1)$  e  $x^2(n-1)$  fazem desaparecer esta simetria, e, portanto, comprometem a identificação do sinal original. Na verdade, o ideal é desconsiderar estes termos, pois não trazem informações relevantes sobre os estados do canal, o que é ratificado pelo fato de os coeficientes ótimos associados a estes termos na solução de Wiener serem nulos.

Diante destas evidências, a proposta seguinte é substituir os termos de segunda ordem pelos de 3ª ordem. Assim, com as duas entradas lineares  $x(n)$  e  $x(n-1)$ , temos  $2 + 2 \times 3 \times 4/6 = 6$  coeficientes para ajustar. De modo semelhante ao caso anterior, podemos montar a matriz  $\mathbf{R}_{\mathbf{x}\mathbf{v}}$  e o vetor de correlação cruzada  $\mathbf{p}_{\mathbf{x}\mathbf{v}\mathbf{d}}$  na tentativa de obter a solução de Wiener. Entretanto, a nova matriz  $\mathbf{R}_{\mathbf{x}\mathbf{v}}$  não é inversível, pois dois de seus autovalores são iguais a zero.

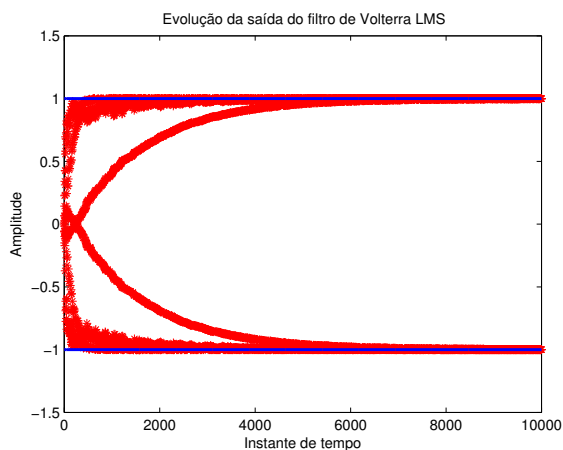
É importante destacar que isto não significa que um filtro de Volterra de 3ª ordem não é adequado para efetuar a equalização deste canal. Na verdade, uma vez temos mais parâmetros livres que o necessário, sobram graus de liberdade e o sistema fica indeterminado, i.e., existem infinitos vetores de coeficientes  $\mathbf{H}$  ótimos <sup>4</sup>.

Devido à simetria existente nos estados do canal, temos, na verdade, que separar corretamente apenas quatro estados (os outros quatro apresentam simetria ímpar em relação a este conjunto de

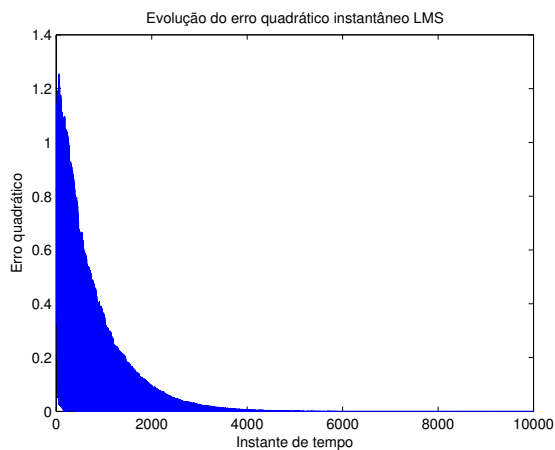
<sup>4</sup>Fato semelhante também pode ser observado no caso de filtros lineares.

estados, como mostra a Figura 1). Logo, uma estrutura dotada de quatro graus de liberdade já seria o suficiente. Por isso, usando todos os *kernels* de 3ª ordem do filtro de Volterra, não temos mais uma única solução ótima para determinar. Apesar disto, é perfeitamente possível utilizar e adaptar esta estrutura para equalizar o canal por meio dos algoritmos LMS e RLS.

Primeiramente, empregamos o algoritmo LMS: após alguns testes preliminares, fixamos os passos de adaptação dos coeficientes de 1ª e 3ª ordens como  $\mu_1 = 0.01$  e  $\mu_3 = 0.005$ , respectivamente. Foram consideradas, no total, 10000 amostras de entrada. As Figuras 2(a) e 2(b) apresentam a evolução da saída do filtro de Volterra e a progressão do erro quadrático instantâneo, respectivamente.



(a) Saída do filtro de Volterra



(b) Erro quadrático instantâneo

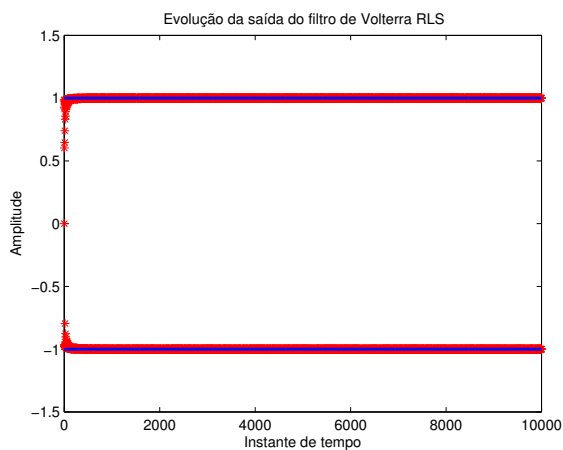
Figura 2: Resultados obtidos pelo filtro de Volterra treinado com o algoritmo LMS.

Podemos observar na Figura 2(a) que a saída do filtro de Volterra gradativamente converge para faixas estreitas em torno dos valores  $+1$  e  $-1$ , que são justamente os dois possíveis valores do sinal  $s(n)$  que desejamos recuperar, o que certamente é uma evidência de que a estrutura foi capaz de realizar a equalização do canal. Contudo, não apenas precisamos chegar próximos dos valores originais da

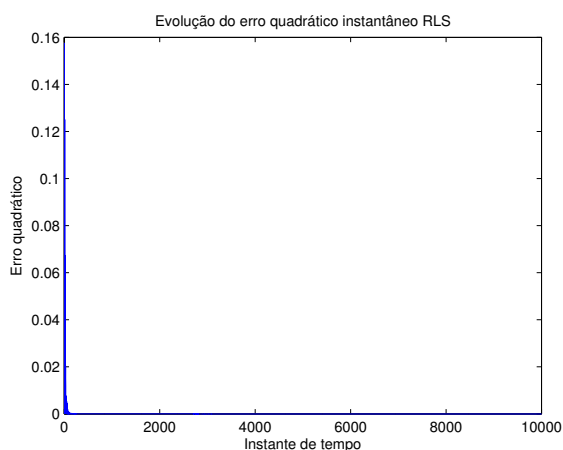
fonte, mas também é importante acertar a ordem em que eles aparecem para que a equalização seja bem-sucedida.

É possível perceber na Figura 2(b) que o erro quadrático instantâneo cai para valores muito pequenos à medida que o treinamento prossegue, o que significa que, após um período inicial de ajuste, a saída do filtro de Volterra se aproxima muito do valor verdadeiro da fonte. Portanto, podemos concluir que os resultados mostrados na Figura 2 indicam claramente que o filtro de Volterra de 3ª ordem consegue efetivamente separar os estados do canal, de modo a recuperar corretamente o sinal  $s(n)$ .

Agora, para o mesmo conjunto de amostras de entrada, e considerando um fator de esquecimento igual a  $\lambda = 0.995$ , empregamos o algoritmo RLS para ajustar os coeficientes do filtro de Volterra. As Figuras 3(a) e 3(b) apresentam a saída da filtro e o erro quadrático em cada instante, respectivamente.



(a) Saída do filtro de Volterra



(b) Erro quadrático instantâneo

Figura 3: Resultados obtidos pelo filtro de Volterra treinado com o algoritmo RLS.

Podemos verificar na Figura 3 que o filtro de Volterra rapidamente foi capaz de aprender a equalizar

o canal, pois sua saída convergiu para valores próximos dos desejados, e houve uma rápida redução no valor do erro quadrático instantâneo entre sua saída e o sinal desejado. Ambos os fatos comprovam que o filtro de Volterra, ajustado por meio do algoritmo RLS, também consegue separar corretamente os estados do canal.

Uma comparação interessante que podemos fazer entre as versões LMS e RLS do filtro de Volterra envolve a velocidade de convergência: conforme podemos verificar nas Figuras 2(b) e 3(b), o algoritmo RLS tende a convergir com maior rapidez que o LMS. Porém, em geral, o RLS se mostrou um pouco mais susceptível a variações na escolha do parâmetro  $\lambda$ , quando comparado ao LMS e às escolhas de  $\mu_1$  e  $\mu_2$ .

Outro aspecto a destacar é que os vetores de coeficientes do filtro de Volterra obtidos através dos algoritmos LMS e RLS são diferentes:

$$\mathbf{H}_{LMS} = \begin{bmatrix} -1.2317 \\ 1.0916 \\ 1.0432 \\ 0.1068 \\ -0.2131 \\ -0.5784 \end{bmatrix} \quad \mathbf{H}_{RLS} = \begin{bmatrix} -1.7494 \\ 0.7867 \\ 1.0695 \\ -0.1824 \\ 0.3649 \\ -0.5277 \end{bmatrix} \quad (11)$$

Isto se deve ao fato que ressaltamos anteriormente: uma vez que temos seis parâmetros livres, mas apenas quatro estados que efetivamente precisam ser separados, surgem infinitas soluções ótimas para o vetor de coeficientes do filtro de Volterra, sendo que todas estas conseguem equalizar o canal. Os vetores obtidos pelos algoritmos LMS e RLS correspondem a duas dentre as várias possibilidades que equalizam o canal.

Para comprovarmos toda a argumentação feita neste caso, apresentamos na Figura 4 a fronteira de decisão associada ao filtro de Volterra ajustado com o auxílio dos algoritmos LMS e RLS, juntamente com a fronteira de decisão do equalizador ótimo bayesiano, o qual assume conhecimento completo das características do ruído, do sinal original, e também dos estados do canal.

Conforme esperado, tanto o filtro de Volterra obtido pelo algoritmo LMS, quanto o filtro obtido pelo algoritmo RLS, produzem fronteiras de decisão que corretamente classificam (separam) os estados do canal, embora sejam bastante diferentes entre si, e também diferentes em relação à fronteira do equalizador bayesiano.

Por fim, vamos considerar um último ensaio: a fim de tornar possível o cálculo de uma solução ótima (fechada) para os coeficientes do filtro de Volterra, optamos por descartar dois termos (*kernels*) cúbicos. Agora, o vetor de entrada do filtro de Volterra corresponde a:

$$\mathbf{x}_v = \begin{bmatrix} x(n) \\ x(n-1) \\ x^3(n) \\ x^2(n)x(n-1) \end{bmatrix}.$$



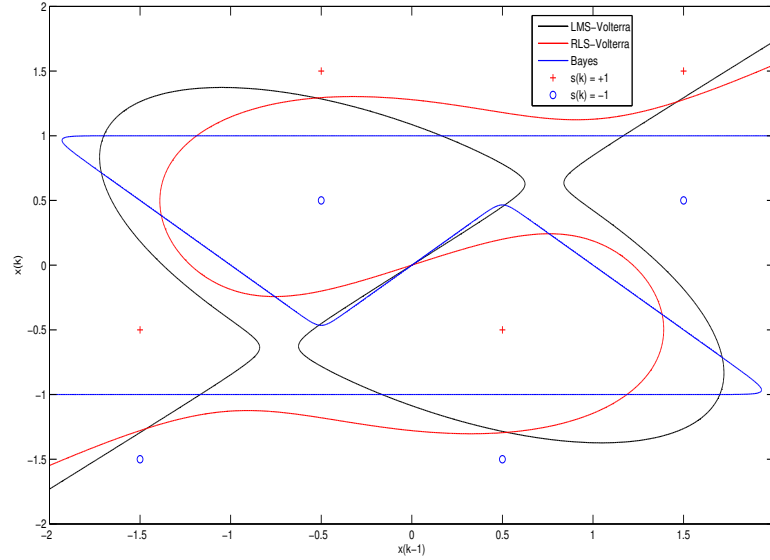
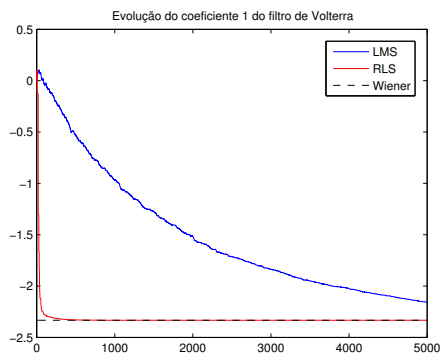


Figura 4: Fronteiras de decisão do filtro de Volterra (LMS e RLS) e do equalizador bayesiano.

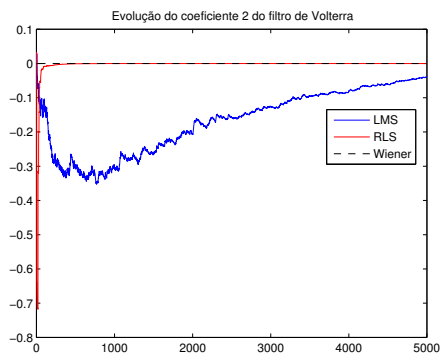
Desta maneira, construindo a matriz  $\mathbf{R}_{\mathbf{x}_v}$  e o vetor  $\mathbf{p}_{\mathbf{x}_v, \mathbf{d}}$ , determinamos a solução de Wiener:

$$\mathbf{H}_{\text{wiener}} = \begin{bmatrix} -2.3333 \\ -1.1102\text{e-}016 \\ 1.3333 \\ 0 \end{bmatrix}.$$

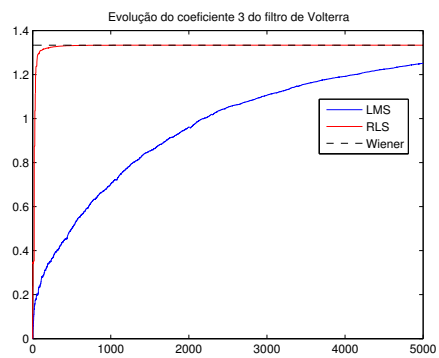
Utilizando os algoritmos LMS e RLS, realizamos o ajuste dos coeficientes do filtro de Volterra. As Figuras 5(a) a 5(d) apresentam a evolução dos quatro coeficientes rumo ao valor ótimo dado pela solução de Wiener. Conforme esperado, tanto para o LMS, quanto para o RLS, o vetor  $\mathbf{H}$  obtido ao final do processo de adaptação equivale à solução de Wiener. A diferença principal está na velocidade de convergência: o RLS conseguiu atingir a solução de Wiener com maior rapidez.



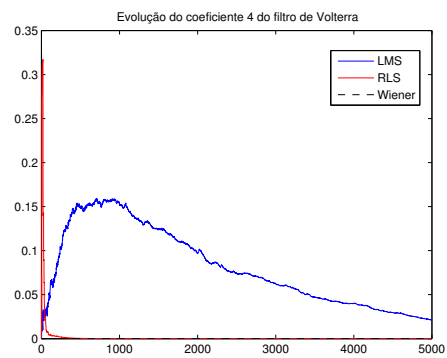
(a) Primeiro coeficiente



(b) Segundo coeficiente



(c) Terceiro coeficiente



(d) Quarto coeficiente

Figura 5: Evolução dos coeficientes do filtro de Volterra - LMS e RLS.