

Algoritmo do sinal do erro

- Tem como objetivo reduzir a complexidade computacional da equação de atualizações do vetor de coeficientes do filtro.
- Útil em aplicações ^{em tempo real} nas quais o tempo para atualizar os coeficientes é muito pequeno, como é o caso de cancelamento de eco e equalização de canais.
- Baseia-se na aplicação de uma operação de quantização no erro de modo que a equação de atualizações dos coeficientes passe a ser:

$$\underline{w}(n+1) = \underline{w}(n) + \mu Q[e(n)] \underline{x}(n)$$

na qual $Q[\cdot]$ é uma função de quantização que assume valores discretos, limitados e não-decrescentes.

- No algoritmo do sinal do erro, $Q[\cdot]$ é a função sinal:

$$Q[e(n)] = \text{sgn}[e(n)] = \begin{cases} 1, & e(n) > 0 \\ 0, & e(n) = 0 \\ -1, & e(n) < 0 \end{cases}$$

Assim:

$$\underline{w}(n+1) = \underline{w}(n) + \mu \text{sgn}[e(n)] \underline{x}(n)$$

- Se o valor do passo for uma potência de 2, as multiplicações envolvidas na atualização do vetor de coeficientes se transformam em operações de troca de sinal e ①

deslocamentos de bits.

- Note que a magnitude do passo não é mais modulada pelo valor do erro.

- Neste caso, com $\frac{\partial J[e(n)]}{\partial e(n)} = \text{sgn}[e(n)]$ podemos

concluir que o critério efetivamente explorado é $J[e(n)] = |e(n)|$, pois:

$$\frac{\partial |e(n)|}{\partial \underline{w}} = \frac{\partial |e(n)|}{\partial e(n)} \cdot \frac{\partial e(n)}{\partial \underline{w}} = \text{sgn}[e(n)] \cdot \underline{x}(n)$$

- O algoritmo do erro busca a solução ótima que minimiza a norma L_1 do erro (módulo) e não o erro quadrático médio.

Algoritmo do sinal do erro

Inicialização:

$$\underline{x}(0) = \underline{w}(0) = [0 \ 0 \ \dots \ 0]^T$$

Para $n \geq 0$, faça

$$e(n) = d(n) - \underline{x}^T(n) \underline{w}(n)$$

$$\underline{w}(n+1) = \underline{w}(n) + \mu \text{sgn}[e(n)] \underline{x}(n)$$

Fim.

Algoritmo LMS-Newton

- Incorpora estimativas das estatísticas de 2ª ordem dos sinais do ambiente para acelerar a convergência, especialmente quando o sinal de entrada é fortemente correlacionado.
- O custo da aceleração é um aumento na complexidade computacional do algoritmo.
- Método de Newton:

$$\underline{w}(n+1) = \underline{w}(n) - R_x^{-1} \nabla J(\underline{w})$$

Nesse caso, dados as estatísticas de segunda ordem de entrada e do vetor gradiente, chega-se à solução ótima em um único passo, independentemente do vetor de coeficientes inicial.

- O algoritmo LMS-Newton explora estimativas da inversa da matriz de autocorrelação e do vetor gradiente na definição do passo de adaptação, conforme a expressão

$$\underline{w}(n+1) = \underline{w}(n) - \mu \hat{R}_x^{-1}(n) \hat{\nabla}_{\underline{w}} J_w(n)$$

em que $\hat{R}_x^{-1}(n)$ é a estimativa de R_x^{-1} usando os dados disponíveis até o instante n e $\hat{\nabla}_{\underline{w}} J_w(n)$ denota o vetor gradiente estimado.

Estimativas estocásticas:

$$\begin{aligned} * \hat{\nabla}_{\underline{w}} J_w(n) &= \hat{R}_x \underline{w}(n) - \hat{p} x d \\ &= \underline{x}(n) \underline{x}^T(n) \underline{w}(n) - \underline{x}(n) d(n) \\ &= \underline{x}(n) [d(n) - d(n)] = -\underline{x}(n) e(n) \end{aligned}$$

$$* \hat{R}_x(n) = \frac{1}{n+1} \sum_{k=0}^n \underline{x}(k) \underline{x}^T(k)$$

$$= \frac{n}{n+1} \hat{R}_x(n-1) + \frac{1}{n+1} \underline{x}(n) \underline{x}^T(n)$$

Note que:

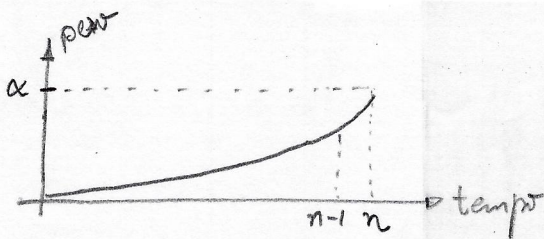
$$E\{\hat{R}_x(n)\} = \frac{1}{n+1} \sum_{k=0}^n E\{\underline{x}(k) \underline{x}^T(k)\} = R_x \rightarrow \text{Estimativa não polarizada da matriz de autocorrelações.}$$

O estimador proposto, ao fazer a média temporal no bloco de $(n+1)$ amostras, pressupõe que os dados estão associados a processos aleatórios estacionários. Em um cenário não-estacionário é interessante priorizar os dados atuais ou mais recentes em detrimento dos mais antigos. Em outras palavras, deve-se atribuir um peso maior às observações mais recentes na tentativa de diminuir a influência do passado na estimativa, buscando, assim, acompanhar a variação temporal dos estatísticos dos sinais.

$$\begin{aligned} \hat{R}_x(n) &= (1-\alpha) \hat{R}_x(n-1) + \alpha \underline{x}(n) \underline{x}^T(n) \\ &= \alpha \sum_{i=0}^{n-1} (1-\alpha)^{n-i} \underline{x}(i) \underline{x}^T(i) + \alpha \underline{x}(n) \underline{x}^T(n) = \alpha \sum_{i=0}^n (1-\alpha)^{n-i} \underline{x}(i) \underline{x}^T(i) \end{aligned}$$

em que α é uma constante ^{positiva} que deve ser escolhida no intervalo $0 < \alpha \leq 0,1$. para que se obtenha um bom compromisso entre a informação passada e presente do sinal de entrada.

Essa abordagem cria uma espécie de janela exponencial que pondera as amostras. O vetor de entrada mais atual, $\underline{x}(n)$, tem peso α , $\underline{x}(n-1)$ tem peso $\alpha(1-\alpha)$, $\underline{x}(n-2)$ tem peso $\alpha(1-\alpha)^2$ e assim por diante.



Note que

$$E\{\hat{R}_x(n)\} = \alpha \sum_{i=0}^n (1-\alpha)^{n-i} E\{\underline{x}(i)\underline{x}^T(i)\} = R_x \text{ para } n \rightarrow \infty$$

estimador não-polarizado

A princípio, seria necessário atualizar e calcular a inversa de $\hat{R}_x(n)$ a cada nova observação. Entretanto, é possível obter uma recursão direta para $\hat{R}_x^{-1}(n)$ utilizando-se o lema de inversão de matrizes:

$$[A + BCD]^{-1} = A^{-1} - A^{-1}B[DA^{-1}B + C^{-1}]^{-1}DA^{-1}$$

Na equação de atualização de $\hat{R}_x(n)$ temos:

$$\hat{R}_x(n) = \underbrace{(1-\alpha)\hat{R}_x(n-1)}_A + \underbrace{\alpha}_{C} \underbrace{\underline{x}(n)}_B \underbrace{\underline{x}^T(n)}_D$$

Usando o lema de inversão de matrizes, obtemos:

$$\hat{R}_x^{-1}(n) = \frac{1}{1-\alpha} \left\{ \hat{R}_x^{-1}(n-1) - \frac{\hat{R}_x^{-1}(n-1) \underline{x}(n) \underline{x}^T(n) \hat{R}_x^{-1}(n-1)}{\frac{1-\alpha}{\alpha} + \underline{x}^T(n) \hat{R}_x^{-1}(n-1) \underline{x}(n)} \right\} \quad (1)$$

Assim, o algoritmo LMS-Newton pode ser resumido nas seguintes etapas:

Inicialização: $\hat{R}_x^{-1}(-1) = \delta I$, com $\delta \geq 0$ e pequeno.
 $\underline{w}(0) = \underline{x}(-1) = [0 \ 0 \ \dots \ 0]^T$

Atualização: para $n \geq 0$:

1) $e(n) = d(n) - \underline{x}^T(n) \underline{w}(n) = d(n) - \underline{w}^T(n) \underline{x}(n)$

2) Atualização de $\hat{R}_x^{-1}(n)$ usando a equação (1)

3) $\underline{w}(n+1) = \underline{w}(n) + \mu \hat{R}_x^{-1}(n) \underline{x}(n) e(n)$.

• No algoritmo LMS convencional, a atualização do vetor de coeficientes do filtro acontece na direção do gradiente estocástico, que constitui uma aproximação do gradiente ideal. Por outro lado, no LMS-Newton, a atualização dos coeficientes se dá em uma direção que aproxima a direção do algoritmo de Newton. Deste modo, espera-se que a convergência do algoritmo LMS-Newton seja mais rápida que a do LMS tradicional.

• Pode ser mostrado que a convergência do LMS-Newton não depende do espalhamento dos autovalores de matriz de autocorrelação do sinal de entrada (razão $\lambda_{\max}/\lambda_{\min}$), diferentemente do que ocorre no LMS convencional.

• Por outro lado, o LMS-Newton é computacionalmente mais custoso e menos robusto devido ao cálculo da inversa da matriz de autocorrelação do sinal de entrada. Também é mais sensível a erros de quantização.

• É bastante semelhante ao algoritmo RLS (Recursive Least Squares)

Algoritmo LMS Normalizado

Objetivo: acelerar a convergência do LMS através do emprego de um passo de adaptação variável, sem recorrer a estimativas da matriz de autoconexão do sinal de entrada do filtro, como ocorre no LMS-Newton.

$$\underline{w}(n+1) = \underline{w}(n) + \mu_n e(n) \underline{x}(n) = \underline{w}(n) + \Delta \underline{w}(n)$$

Obter o passo μ_n que resulte no maior decréscimo possível do erro quadrático instantâneo.

$$e^2(n) = [d(n) - \underline{w}^T(n) \underline{x}(n)]^2 = d^2(n) - 2d(n) \underline{w}^T(n) \underline{x}(n) + \underline{w}^T(n) \underline{x}(n) \underline{x}^T(n) \underline{w}(n)$$

Se for aplicada uma perturbação no vetor de coeficientes, de modo que o novo vetor de coeficientes seja $\underline{\tilde{w}}(n) = \underline{w}(n) + \Delta \underline{\tilde{w}}(n)$, o erro quadrático resultante é:

$$\tilde{e}^2(n) = e^2(n) + 2 \Delta \underline{\tilde{w}}^T(n) \underline{x}(n) \underline{x}^T(n) \underline{w}(n) + \Delta \underline{\tilde{w}}^T(n) \underline{x}(n) \underline{x}^T(n) \Delta \underline{\tilde{w}}(n) - 2d(n) \Delta \underline{\tilde{w}}^T(n) \underline{x}(n)$$

A variação do erro quadrático devido à perturbação é definida como

$$\begin{aligned} \Delta e^2(n) &\triangleq \tilde{e}^2(n) - e^2(n) \\ &= -2 \Delta \underline{\tilde{w}}^T(n) \underline{x}(n) e(n) + \Delta \underline{\tilde{w}}^T(n) \underline{x}(n) \underline{x}^T(n) \Delta \underline{\tilde{w}}(n) \end{aligned}$$

Para acelerar a convergência, é necessário que $\Delta e^2(n)$ seja negativo e mínimo através da escolha apropriada de μ_n . Em outras palavras, é preciso escolher o passo μ_n que leva ao maior decréscimo possível de $e^2(n)$. Substituindo $\Delta \underline{\tilde{w}}(n) = \mu_n e(n) \underline{x}(n)$ na expressão acima, obtemos:

$$\Delta e^2(n) = -2\mu_n e^2(n) \underline{x}^T(n) \underline{x}(n) + \mu_n^2 e^2(n) (\underline{x}^T(n) \underline{x}(n))^2$$

Derivando em relação a μ_n e igualando a zero para determinar o mínimo de $\Delta e^2(n)$, temos:

$$\frac{\partial \Delta e^2(n)}{\partial \mu_n} = -2 e^2(n) \underline{x}^T(n) \underline{x}(n) + 2\mu_n e^2(n) (\underline{x}^T(n) \underline{x}(n))^2 = 0$$

$$\mu_n = \frac{1}{\underline{x}^T(n) \underline{x}(n)}$$

Como:

$$\frac{\partial^2 \Delta e^2(n)}{\partial \mu_n^2} = 2 e^2(n) (\underline{x}^T(n) \underline{x}(n))^2 \geq 0 \Rightarrow \text{ponto de mínimo}$$

e

$$-2 \frac{1}{\underline{x}^T(n) \underline{x}(n)} e^2(n) \underline{x}^T(n) \underline{x}(n) + \left(\frac{1}{\underline{x}^T(n) \underline{x}(n)} \right)^2 e^2(n) (\underline{x}^T(n) \underline{x}(n))^2 = -e^2(n) \leq 0 \rightarrow \Delta e^2(n) \text{ negativo}$$

Concluímos que $\mu_n = \frac{1}{\underline{x}^T(n) \underline{x}(n)}$ assegura $\Delta e^2(n)$ negativo e mínimo no instante n .

O algoritmo LMS normalizado utiliza o passo μ_n na direção definida pelo vetor gradiente. Usualmente, um fator μ constante é mantido na equação de atualizações de \underline{w} , uma vez que as derivações baseiam-se no erro quadrático instantâneo, e não no MSE (que é o que de fato se deseja minimizar). Além disso, para evitar passos exageradamente grandes quando $\underline{x}(n)^T \underline{x}(n)$ se aproxima de zero, acrescenta-se uma constante δ positiva no denominador de μ_n . Dessa forma, o vetor de coeficientes do filtro é adoptado conforme a expressão:

$$\underline{w}(n+1) = \underline{w}(n) + \mu \cdot \frac{1}{\delta + \underline{x}^T(n)\underline{x}(n)} \cdot e(n)\underline{x}(n)$$

estimativa instantânea da potência do sinal de entrada ($r_x(n)$)

Algoritmo LMS Normalizado

Inicialização:

$$\underline{x}(0) = \underline{w}(0) = [0 \ 0 \ \dots \ 0]^T$$

$$0 < \mu \leq 1$$

δ constante pequena e positiva.

Atualização:

Para $n \geq 0$, faça

$$e(n) = d(n) - \underline{x}^T(n)\underline{w}(n)$$

$$\underline{w}(n+1) = \underline{w}(n) + \frac{\mu}{\delta + \underline{x}^T(n)\underline{x}(n)} e(n)\underline{x}(n)$$

Fim

Algoritmo LMS no domínio transformado

Idéia: Aplicar uma transformação no sinal de entrada do filtro de modo a melhorar o condicionamento da matriz de autocorrelação, ou seja, diminuir o espalhamento de seus autovalores e, por conseguinte, acelerar a convergência do algoritmo de adaptação.

Sinal transformado: $\underline{s}(n) = T\underline{x}(n)$, em que $T T^T = I$ represente uma transformação ortonormal.

Tal transformação, produz o seguinte efeito na superfície MSE:

$$J_{MSE}(n) = J_{MIN} + \Delta \underline{W}^T R_x \Delta \underline{W}(n), \quad \Delta \underline{W}(n) = \underline{W}(n) - \underline{W}_0$$

No domínio transformado.

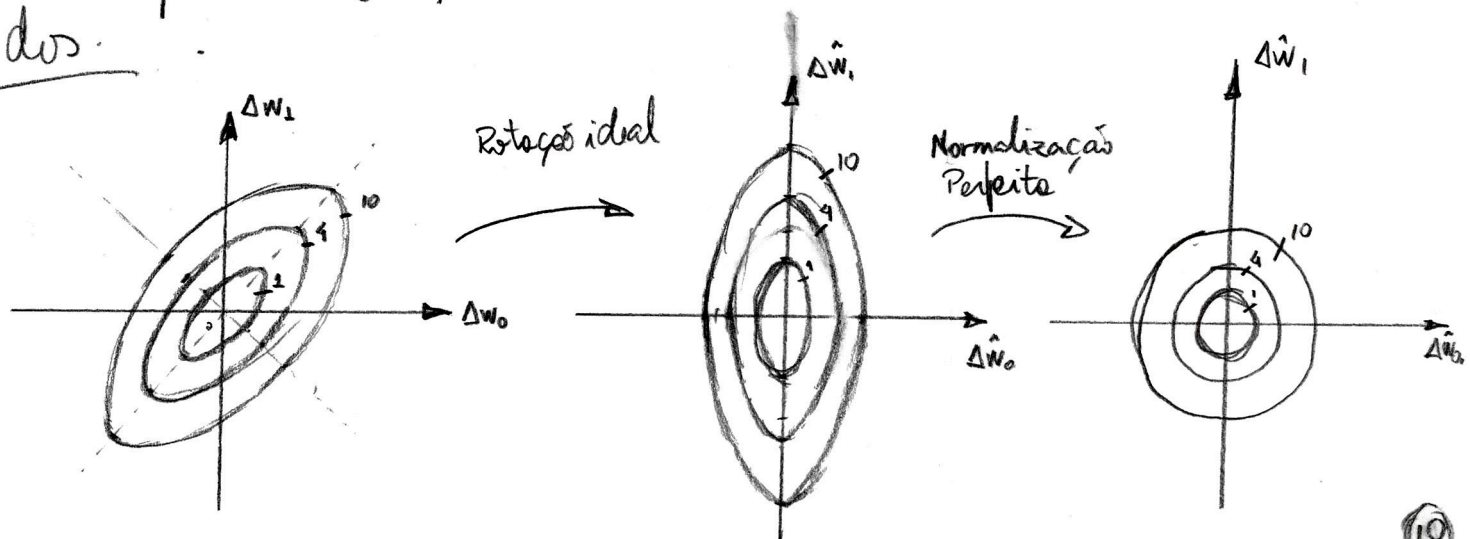
$$J_{MSE}^{(\hat{n})} = \hat{J}_{MIN} + \Delta \hat{\underline{W}}^T R_s \Delta \hat{\underline{W}}(n), \quad \Delta \hat{\underline{W}}(n) = \hat{\underline{W}}(n) - \hat{\underline{W}}_0$$

$$= \hat{J}_{MIN} + \Delta \hat{\underline{W}}^T(n) E \{ T \underline{x}(n) \underline{x}(n)^T T^T \} \Delta \hat{\underline{W}}(n)$$

$$= \hat{J}_{MIN} + \Delta \hat{\underline{W}}^T(n) T R_x T^T \Delta \hat{\underline{W}}(n)$$

↓
coeficientes do filtro
no domínio transformado.

A transformação T realiza uma rotação da superfície de erro. Caso, adicionalmente, a transformação faça uma normalização da potência do sinal, o espalhamento dos autovalores é alterado. No caso ideal, a superfície MSE se torna um hiperparabolóide esférico quando todos autovalores de R_s são iguais e não nulos, o qual corresponde ao domínio mais favorável para um algoritmo baseado no gradiente. A rotação perfeita, diagonaliza R_s e alinha os eixos dos elipses de erro quadrático constante com os eixos dos coeficientes, o que significa que os sinais $\underline{s}(n)$ são descorrelacionados.



A rotação ideal implica na diagonalização de R_x , pois, nesse caso $R_s = T R_x T^T$ é diagonal. Assim, a matriz T deve corresponder à ^{transposta} matriz de autovetores ortonormais de R_x , a matriz $Q^T = [q_1 \ q_2 \ \dots \ q_N]^T$.

$$R_s = E\{\underline{\Delta}(n) \underline{\Delta}^T(n)\} = E\{Q^T \underline{X}(n) \underline{X}^T(n) Q\} = Q^T E\{\underline{X}(n) \underline{X}^T(n)\} Q = Q^T R_x Q$$

$$= \Lambda = \begin{bmatrix} \lambda_1 & 0 & \dots & 0 \\ 0 & \lambda_2 & & \\ \vdots & & \ddots & \\ 0 & \dots & 0 & \lambda_N \end{bmatrix} \Rightarrow \text{matriz diagonal contendo os autovalores de } R_x$$

Essa matriz de transformação corresponde à Karhunen-Löve Transform (KLT). Note que a KLT não altera a excentricidade dos elipses de erro quadrático constante, pois os autovalores de Λ são os mesmos de R_x .

Para obter a excentricidade ideal, é necessário que todos os autovalores de R_s sejam iguais, ou, equivalentemente, que todos os elementos da diagonal de R_s sejam iguais. Para tanto, é necessário normalizar cada elemento de $\underline{s}(n)$ de modo que

$$E\{A \underline{s}(n) \underline{s}^T(n) A^T\} = I \Rightarrow A \Lambda A^T = I \Rightarrow a_{ii}^2 \lambda_i = 1 \Rightarrow a_{ii} = \sqrt{\frac{1}{\lambda_i}}$$

$$\text{Mas } \lambda_i = E\{s_i^2(n)\} = \sigma_i^2 \Rightarrow A = \Sigma^{-1} = \begin{bmatrix} 1/\sigma_1 & 0 & \dots & 0 \\ 0 & 1/\sigma_2 & & \\ \vdots & & \ddots & \\ 0 & \dots & 0 & 1/\sigma_N \end{bmatrix}$$

Portanto, cada elemento de $\underline{s}(n)$ deve ser normalizado pela sua potência para que a excentricidade seja ideal.

Fazendo $T = A \cdot Q^T$:

$$\underline{s}(n) = A Q^T \underline{x}(n) \rightarrow \text{signal no domínio transformado}$$

A matriz de autocorrelação de $s(n)$ é:

$$R_s = E\{\underline{s}(n) \underline{s}^T(n)\} = E\{A Q^T \underline{x}(n) \underline{x}^T(n) Q A^T\} = A Q^T R_x Q A^T$$

O vetor de correlação cruzada entre $s(n)$ e o sinal desejado é dado por:

$$p_{sd} = E\{\underline{s}(n) \underline{d}(n)\} = E\{A Q^T \underline{x}(n) \underline{d}(n)\} = A Q^T p_{xd}$$

A solução ótima no domínio transformado é

$$\begin{aligned} \underline{\hat{w}}_0 &= R_s^{-1} p_{sd} = (A Q^T R_x Q A^T)^{-1} A Q^T p_{xd} = (Q^T R_x Q A^T)^{-1} Q^T p_{xd} = \\ &= (R_x Q A^T)^{-1} p_{xd} = (Q A^T)^{-1} R_x p_{xd} = (Q A^T)^{-1} \underline{w}_0 \end{aligned}$$

Portanto, para obtermos a solução de Wiener a partir do filtro ótimo ^{no domínio transformado}, devemos usar a seguinte transformação:

$$\underline{w}_0 = Q A^T \underline{\hat{w}}_0$$

Como A é diagonal, $A = A^T$,

$$\underline{w}_0 = Q A \underline{\hat{w}}_0$$

ou

$$\underline{w}_0 = Q \Sigma^{-1} \underline{\hat{w}}_0$$

Observações:

- A KLT não pode ser computada de maneira eficiente em tempo real. Alternativamente, deve-se usar uma transformação unitária que aplique uma rotação na superfície de MSE próxima da rotação resultante da KLT para o sinal em questão. Para sinais de voz, por exemplo, a transformação discreta do cosseno (DCT, do inglês Discrete Cosine Transform) é uma boa aproximação de KLT.
- A matriz de normalização da potência do sinal, Σ^{-1} , transformado depende do conhecimento dos autovalores de R_x , ou de uma estimativa da potência de cada componente de $\underline{s}(n)$. O que dificulta seu cálculo em aplicações de tempo real.

Nesse sentido, ao invés de aplicar o algoritmo LMS convencional, uma alternativa é utilizar o LMS normalizado no domínio transformado:

$$\hat{w}_i(n+1) = \hat{w}_i(n) + \frac{\mu}{\delta + G_i^2(n)} e(n) s_i(n)$$

em que $G_i^2(n) = \alpha s_i^2(n) + (1-\alpha) G_i^2(n-1)$, com $0 < \alpha \leq 0,1$.

Na forma matricial, a equação de atualização dos pesos do filtro é dada por:

$$\underline{\hat{w}}(n+1) = \underline{\hat{w}}(n) + \mu e(n) \Sigma^{-2}(n) \underline{s}(n) e(n)$$

em que

$$\Sigma^{-2}(n) = \begin{bmatrix} [G_1^2(n) + \delta]^{-1} & 0 & \dots & 0 \\ 0 & [G_2^2(n) + \delta]^{-1} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & \dots & \dots & [G_N^2 + \delta]^{-1} \end{bmatrix}$$

Se o passo μ for escolhido adequadamente, os coeficientes do filtro adaptativo devem convergir para o valor ótimo:

$$\underline{\hat{w}}_0 = R_s^{-1} p_{sd}$$

$$= (T R_x T^T)^{-1} T p_{xd}$$

$$= T R_x^{-1} p_{xd} = T \underline{w}_0 \Rightarrow \boxed{\underline{w}_0 = T^{-1} \hat{\underline{w}}_0}$$

Portanto a matriz de transformações deve ser inversível para que seja possível obter o filtro ótimo a partir do filtro ótimo no domínio transformado. (14)