3
Equalização adaptativa no domínio da frequência

Em situações reais de comunicações digitais, os receptores precisam a todo momento equalizar as distorções causadas pelo canal (desconhecido). Neste capítulo, três algoritmos adaptativos (LMS, NLMS, RLS) são apresentados operando no domínio da transformada.

3.1
Equalização adaptativa

Um equalizador adaptativo implementa de forma recursiva uma solução seguindo um critério específico. Neste trabalho, como é visto no Capítulo 2, considera-se a solução MMSE (no domínio da freqüência) encontrada em (2-77) e em (2-82) por ter melhores resultados frente ao equalizador ZF. Num primeiro estágio de treinamento, blocos-piloto conhecidos pelo lado do receptor são enviados pelo transmissor. Nesta fase inicial, o filtro adaptativo ajusta seus parâmetros (*taps*). Uma vez alcançada a convergência, o sistema entra em seu modo de operação, onde o receptor desconhece a informação enviada e o equalizador tem que trabalhar, e se ajustar, sob estas circunstâncias.

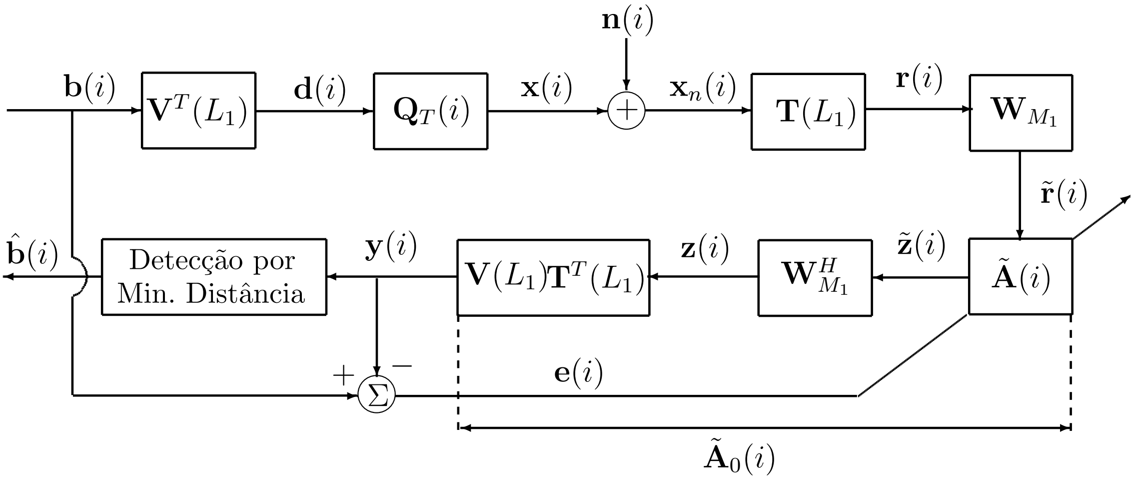

Figura 3.1: Estrutura da equalização adaptativa no domínio da freqüência.

A Figura 3.1 ilustra o diagrama de blocos da equalização adaptativa utilizada. No caso do MMSE, a função custo, tanto para sistemas CP quanto para ZP, fica da forma

$$J = \mathbb{E}\left[\|\mathbf{b}(i) - \tilde{\mathbf{A}}_0(i)\tilde{\mathbf{r}}(i)\|^2\right], \tag{3-1}$$

em que $\tilde{\mathbf{A}}_0(i)$ contém a matriz de equalização $\tilde{\mathbf{A}}(i)$, a IDFT M_1 pontos $\mathbf{W}_{M_1}^{\mathcal{H}}$ e o processamento $\mathbf{V}(L_1)\mathbf{T}^T(L_1)$ que remove as últimas L componentes de um bloco, caso seja o sistema ZP em questão.

O Capítulo 2 e Apêndice A mostram que a filtragem realizada no domínio da transformada de Fourier pode ser implementada por uma matriz de equalização diagonal com M_1 coeficientes. Ou seja, podemos rescrever (3-1) de forma equivalente

$$J = \mathbb{E}\left[\| \underbrace{\mathbf{b}(i) - \mathbf{V}(L_1)\mathbf{T}^T(L_1)\mathbf{W}_{M_1}^{\mathcal{H}}\tilde{\mathbf{R}}(i)\tilde{\mathbf{a}}(i)}_{\mathbf{e}(i)} \|^2\right], \tag{3-2}$$

onde $\tilde{\mathbf{R}}(i)$ representa a diagonalização das componentes do vetor $\tilde{\mathbf{r}}(i)$. O equalizador fica então representado por um vetor $\tilde{\mathbf{a}}(i)$ de dimensão $M_1 \times 1$

$$\tilde{\mathbf{a}}(i) = \left[\tilde{a}^{(i)}[0],\ \tilde{a}^{(i)}[1],\ \ldots,\ \tilde{a}^{(i)}[M_1 - 1]\right]^T. \tag{3-3}$$

No argumento da função (3-2) tem-se o vetor de erro associado ao i-ésimo bloco transmitido

$$\mathbf{e}(i) = \mathbf{b}(i) - \mathbf{V}(L_1)\mathbf{T}^T(L_1)\mathbf{W}_{M_1}^{\mathcal{H}}\tilde{\mathbf{R}}(i)\tilde{\mathbf{a}}(i). \tag{3-4}$$

No caso do CP, $L_1 = L$ e a equação (3-2) se torna

$$J = \mathbb{E}\left[\|\mathbf{b}(i) - \mathbf{W}_N^{\mathcal{H}}\tilde{\mathbf{R}}(i)\tilde{\mathbf{a}}(i)\|^2\right], \tag{3-5}$$

e no esquema de transmissão ZP, $L1 = 0$ e a equação (3-2) se reduz à

$$J = \mathbb{E}\left[\|\mathbf{b}(i) - \mathbf{W}_{MN}^{\mathcal{H}}\tilde{\mathbf{R}}(i)\tilde{\mathbf{a}}(i)\|^2\right]. \tag{3-6}$$

O erro associado ao i-ésimo bloco é representado por

$$\mathbf{e}_{CP}(i) = \mathbf{b}(i) - \mathbf{W}_N^{\mathcal{H}}\tilde{\mathbf{R}}(i)\tilde{\mathbf{a}}(i) \tag{3-7}$$

no caso da prefixação CP, e

$$\mathbf{e}_{ZP}(i) = \mathbf{b}(i) - \mathbf{W}_{MN}^{\mathcal{H}}\tilde{\mathbf{R}}(i)\tilde{\mathbf{a}}(i) \tag{3-8}$$

no caso da sufixação ZP. Este vetor de erro serve para atualizar os M_1 *taps* do filtro adaptativo.

Como é desejado minimizar o erro médio quadrático, toma-se o gradiente da função (3-2) e iguala-se o resultado à zero, encontrando assim, o ponto de mínimo dessa função quadrática. Observando que os coeficientes do equalizador são números complexos da forma $\tilde{\mathbf{a}} = \tilde{\mathbf{a}}_R + j\tilde{\mathbf{a}}_I$, o gradiente da função custo (real) é um vetor coluna de M_1 componentes, e é encontrado fazendo

$$\nabla_{\tilde{\mathbf{a}}} J = \begin{bmatrix} \frac{\partial J}{\partial \tilde{a}_R^{(i)}[0]} - j\frac{\partial J}{\partial \tilde{a}_I^{(i)}[0]} \\ \vdots \\ \frac{\partial J}{\partial \tilde{a}_R^{(i)}[\ell]} - j\frac{\partial J}{\partial \tilde{a}_I^{(i)}[\ell]} \\ \vdots \\ \frac{\partial J}{\partial \tilde{a}_R^{(i)}[M_1-1]} - j\frac{\partial J}{\partial \tilde{a}_I^{(i)}[M_1-1]} \end{bmatrix}. \tag{3-9}$$

Aplicando (3-9) em (3-2) e excluindo o índice i por simplificação, obtém-se

$$\begin{aligned} \nabla_{\tilde{\mathbf{a}}} J &= \nabla_{\tilde{\mathbf{a}}} \mathbb{E}\left[\mathbf{b}^{\mathcal{H}}\mathbf{b} - \mathbf{b}^{\mathcal{H}}\mathbf{V}(L_1)\mathbf{T}^T(L_1)\mathbf{W}_{M_1}^{\mathcal{H}}\tilde{\mathbf{R}}\tilde{\mathbf{a}} - \tilde{\mathbf{a}}^{\mathcal{H}}\tilde{\mathbf{R}}^{\mathcal{H}}\mathbf{W}_{M_1}\mathbf{T}(L_1)\mathbf{V}^T(L_1)\mathbf{b}\right. \\ &+ \left.\tilde{\mathbf{a}}^{\mathcal{H}}\tilde{\mathbf{R}}^{\mathcal{H}}\mathbf{W}_{M_1}\mathbf{T}(L_1)\mathbf{V}^T(L_1)\mathbf{V}(L_1)\mathbf{T}^T(L_1)\mathbf{W}_{M_1}^{\mathcal{H}}\tilde{\mathbf{R}}\tilde{\mathbf{a}}\right] \\ &= 2\mathbb{E}\left[-\tilde{\mathbf{R}}^T\mathbf{W}_{M_1}^*\mathbf{T}(L_1)\mathbf{V}^T(L_1)\mathbf{b}^* + \tilde{\mathbf{R}}^T\mathbf{W}_{M_1}^*\mathbf{T}(L_1)\mathbf{V}^T(L_1)\mathbf{V}(L_1)\mathbf{T}^T(L_1)\mathbf{W}_{M_1}^T\tilde{\mathbf{R}}^*\tilde{\mathbf{a}}^*\right]. \end{aligned} \tag{3-10}$$

Igualando o gradiente a zero e conjugando termos, chega-se à

$$\begin{aligned} \nabla_{\tilde{\mathbf{a}}} J &= 2\mathbb{E}\left[-\tilde{\mathbf{R}}^{\mathcal{H}}\mathbf{W}_{M_1}\mathbf{T}(L_1)\mathbf{V}^T(L_1)\mathbf{b} + \tilde{\mathbf{R}}^{\mathcal{H}}\mathbf{W}_{M_1}\mathbf{T}(L_1)\mathbf{V}^T(L_1)\mathbf{V}(L_1)\mathbf{T}^T(L_1)\mathbf{W}_{M_1}^{\mathcal{H}}\tilde{\mathbf{R}}\tilde{\mathbf{a}}\right] \\ &= -2\mathbb{E}\left\{\tilde{\mathbf{R}}^{\mathcal{H}}\mathbf{W}_{M_1}\mathbf{T}(L_1)\mathbf{V}^T(L_1)\underbrace{\left[\mathbf{b} - \mathbf{V}(L_1)\mathbf{T}^T(L_1)\mathbf{W}_{M_1}^{\mathcal{H}}\tilde{\mathbf{R}}\tilde{\mathbf{a}}\right]}_{\mathbf{e}}\right\} \\ &= -2\mathbb{E}\left[\tilde{\mathbf{R}}^{\mathcal{H}}\mathbf{W}_{M_1}\mathbf{T}(L_1)\mathbf{V}^T(L_1)\mathbf{e}\right] \\ &= 0. \end{aligned} \tag{3-11}$$

Este resultado diz que, de acordo com o princípio da ortogonalidade [11], a entrada do filtro no instante i e o conjugado do i-ésimo vetor de erro obtido, são funções que, na média, são ortogonais entre si no $\mathbb{C}^{M_1}$.

No caso do CP, $L_1 = L$ e o vetor gradiente da função custo se resume à

$$\nabla_{\tilde{\mathbf{a}}} J\big|_{CP} = -2\mathbb{E}\left[\tilde{\mathbf{R}}^{\mathcal{H}}(i)\mathbf{W}_N\mathbf{e}_{CP}(i)\right]. \tag{3-12}$$

No caso do ZP, $L_1 = 0$, o que resulta em

$$\nabla_{\tilde{\mathbf{a}}} J\big|_{ZP} = -2\mathbb{E}\left[\tilde{\mathbf{R}}^{\mathcal{H}}(i)\mathbf{W}_{MN}\mathbf{e}_{ZP}(i)\right]. \tag{3-13}$$

Estes resultados são utilizados nas seções seguintes para encontrar as expressões de atualização dos filtros adaptativos.

3.2

LMS - Least Mean Square

O algoritmo LMS (*Least Mean Square*) é um dos mais difundidos e utilizados. Ele é em geral bem simples, fácil de se implementar e com baixo custo computacional se comparado com outros algoritmos adaptativos. Daí a sua larga utilização. Este equalizador adaptativo implementa a solução MMSE de forma recursiva, minimizando o quadrado da norma do erro instantâneo. Ou seja, retira-se o valor esperado de (3-2), o que significa dizer que a cada iteração tem-se uma estimativa da solução MMSE. Dessa forma, a função custo do LMS fica:

$$\begin{aligned} J_{LMS} &= \|\mathbf{b}(i) - \mathbf{V}(L_1)\mathbf{T}^T(L_1)\tilde{\mathbf{R}}(i)\tilde{\mathbf{a}}(i)\|^2 & \text{(3-14)} \\ &= \mathbf{e}^{\mathcal{H}}(i)\mathbf{e}(i). & \text{(3-15)} \end{aligned}$$

O gradiente, que neste caso também é uma estimativa do vetor gradiente da solução MMSE, é

$$\hat{\boldsymbol{\nabla}}_{\tilde{\mathbf{a}}} J_{LMS} = -2\tilde{\mathbf{R}}^{\mathcal{H}}\mathbf{W}_{M_1}\mathbf{T}(L_1)\mathbf{V}^T(L_1)\mathbf{e}(i). \quad \text{(3-16)}$$

Agora, vamos fazer com que o algoritmo caminhe no sentido oposto ao de maior crescimento da função custo, ou seja, na direção contrária ao do gradiente. Estabelece-se assim, o algoritmo iterativo LMS que é da forma

$$\begin{aligned} \tilde{\mathbf{a}}(i+1) &= \tilde{\mathbf{a}}(i) - \frac{1}{2}\mu\left[\hat{\boldsymbol{\nabla}}_{\tilde{\mathbf{a}}} J_{LMS}\right]^* \\ &= \tilde{\mathbf{a}}(i) + \mu\tilde{\mathbf{R}}^{\mathcal{H}}(i)\mathbf{W}_{M_1}\mathbf{T}(L_1)\mathbf{V}^T(L_1)\mathbf{e}(i). & \text{(3-17)} \end{aligned}$$

O uso do conjugado do gradiente na equação (3-17), se deve ao aparecimento de $\tilde{\mathbf{a}}^*(i)$ na expressão (3-10). Para o CP, tem-se então

$$\begin{aligned} \tilde{\mathbf{a}}_{LMS}\big|_{CP}(i+1) &= \tilde{\mathbf{a}}_{LMS}\big|_{CP}(i) - \frac{1}{2}\mu\left[\hat{\boldsymbol{\nabla}}_{\tilde{\mathbf{a}}} J_{LMS}\right]^* \\ &= \tilde{\mathbf{a}}_{LMS}\big|_{CP}(i) + \mu\tilde{\mathbf{R}}^{\mathcal{H}}(i)\mathbf{W}_N\mathbf{e}_{CP}(i), & \text{(3-18)} \end{aligned}$$

e para a transmissão ZP

$$\begin{aligned} \tilde{\mathbf{a}}_{LMS}\big|_{ZP}(i+1) &= \tilde{\mathbf{a}}\big|_{ZP}(i) - \frac{1}{2}\mu\left[\hat{\boldsymbol{\nabla}}_{\tilde{\mathbf{a}}} J_{LMS}\right]^* \\ &= \tilde{\mathbf{a}}\big|_{ZP}(i) + \mu\tilde{\mathbf{R}}^{\mathcal{H}}(i)\mathbf{W}_{MN}\mathbf{e}_{ZP}(i). & \text{(3-19)} \end{aligned}$$

3.3

NLMS - Normalized Least Mean Square

Como foi visto no algoritmo LMS, apesar de termos controle sobre o passo μ da atualização do filtro (que é escolhido de acordo com as características do canal e da RSR), não se tem qualquer controle sobre as excursões do bloco $\tilde{\mathbf{r}}(i)$ na entrada do filtro. Por conta disso, quando $\tilde{\mathbf{r}}(i)$ tem valores muito altos em suas componentes, o filtro LMS sofre da amplificação do gradiente do ruído.

Para contornar esta dificuldade, podemos usar o filtro LMS normalizado, ou NLMS. Em particular, o ajuste aplicado aos *taps* do filtro na iteração $i+1$ é normalizado com relação ao i-ésimo vetor observado na recepção $\tilde{\mathbf{r}}(i)$. Isso posto, a equação (3-17) pode ser rescrita da forma:

$$\begin{aligned} \tilde{\mathbf{a}}(i+1) &= \tilde{\mathbf{a}}(i) + \alpha \left[\tilde{\mathbf{R}}^{\mathcal{H}}(i)\tilde{\mathbf{R}}(i) + \delta \mathbf{I}_{M_1}\right]^{-1} \tilde{\mathbf{R}}^{\mathcal{H}}(i)\mathbf{W}_{M_1}\mathbf{T}(L_1)\mathbf{V}^T(L_1)\mathbf{e}(i) && \text{(3-20)} \\ &= \tilde{\mathbf{a}}(i) + \boldsymbol{\alpha}_{M_1}(i)\tilde{\mathbf{R}}^{\mathcal{H}}(i)\mathbf{W}_{M_1}\mathbf{T}(L_1)\mathbf{V}^T(L_1)\mathbf{e}(i). && \text{(3-21)} \end{aligned}$$

O parâmetro δ, uma constante de valor pequeno, foi introduzido para evitar problemas numéricos na inversão do produto $\tilde{\mathbf{R}}^{\mathcal{H}}(i)\tilde{\mathbf{R}}(i)$ contida em (3-20). Estes problemas são oriundos de nulos (ou valores muito baixos) em uma das componentes do espectro do canal. O controle de ganho é realizado pela introdução do escalar α. A matriz $\boldsymbol{\alpha}_{M_1}(i)$ em (3-21) é diagonal de dimensão $M_1 \times M_1$, e contém os ganhos (adaptativos) normalizados para o equalizador linear NLMS.

Para a transmissão CP, a equação (3-20) se torna

$$\begin{aligned} \tilde{\mathbf{a}}_{NLMS}\big|_{CP}(i+1) &= \tilde{\mathbf{a}}(i) + \alpha \left[\tilde{\mathbf{R}}^{\mathcal{H}}(i)\tilde{\mathbf{R}}(i) + \delta \mathbf{I}_{N}\right]^{-1} \tilde{\mathbf{R}}^{\mathcal{H}}(i)\mathbf{W}_{N}\mathbf{e}_{CP}(i) && \text{(3-22)} \\ &= \tilde{\mathbf{a}}(i) + \boldsymbol{\alpha}_{N}(i)\tilde{\mathbf{R}}^{\mathcal{H}}(i)\mathbf{W}_{N}\mathbf{e}_{CP}(i). && \text{(3-23)} \end{aligned}$$

Pode-se enxergar a solução CP-SC-FDE-NLMS como N equações desacopladas (independentes), em que o p-ésimo *tap* do filtro é atualizado fazendo

$$\begin{aligned} \tilde{a}_p\big|_{CP}(i+1) &= \tilde{a}_p\big|_{CP}(i) + \alpha \frac{\tilde{r}^*(i)}{\tilde{r}^*(i)\tilde{r}(i)}\mathbf{W}_p\mathbf{e}_{CP}(i) && \text{(3-24)} \\ &= \tilde{a}_p\big|_{CP}(i) + \alpha \frac{1}{\tilde{r}(i)}\mathbf{W}_p\mathbf{e}_{CP}(i), && \text{(3-25)} \end{aligned}$$

onde a constante δ em (3-24) foi suposta ser igual à zero apenas para efeito de ilustração do resultado e $\mathbf{W}_p$ é uma matriz de dimensão $1 \times N$ e que representa a p-ésima linha da matriz de DFT $\mathbf{W}_N$. Fica expresso na equação (3-24) que excursões altas de $\tilde{r}_p(i)$ representam um ganho menor na adaptação do algoritmo na i-ésima iteração. Já o LMS não faz qualquer distinção das

componentes da transformada do vetor recebido.
Para o sistema ZP, tem-se

$$\begin{aligned}\tilde{\mathbf{a}}_{NLMS}\big|_{ZP}(i+1) &= \tilde{\mathbf{a}}_{NLMS}\big|_{ZP}(i) + \alpha\left[\tilde{\mathbf{R}}^{\mathcal{H}}(i)\tilde{\mathbf{R}}(i) + \delta\mathbf{I}_M\right]^{-1}\tilde{\mathbf{R}}^{\mathcal{H}}(i)\mathbf{W}_{MN}\mathbf{e}_{ZP}(i) \quad (3\text{-}26)\\ &= \tilde{\mathbf{a}}_{NLMS}\big|_{ZP}(i) + \boldsymbol{\alpha}_M(i)\tilde{\mathbf{R}}^{\mathcal{H}}(i)\mathbf{W}_{MN}\mathbf{e}_{ZP}(i). \quad (3\text{-}27)\end{aligned}$$

3.4 RLS - Recursive Least Squares

Este algoritmo implementa de forma recursiva o algoritmo de mínimos quadrados [11]. Neste filtro adaptativo deve-se minimizar a soma dos erros médio quadráticos ponderados por uma constante com decaimento exponencial. Definindo-se então a função custo a ser minimizada como

$$\begin{aligned}J_{RLS} &= \sum_{l=1}^{i}\lambda^{i-l}\|\mathbf{e}(l)\|^2\\ &= \sum_{l=1}^{i}\lambda^{i-l}\mathbf{e}^{\mathcal{H}}(l)\mathbf{e}(l), \quad (3\text{-}28)\end{aligned}$$

onde o vetor de erro $\mathbf{e}(i)$ é definido em (3-4). No caso em que $\lambda = 1$, não fazemos nenhuma distinção dos erros anteriores, dando à eles o mesmo peso do presente na medida a ser minimizada. Por outro lado, podemos fazer com que λ assuma valores muito próximos da unidade (porém menores). Isto levará o algoritmo a "esquecer"um pouco do passado. O grau de esquecimento deve variar conforme for a velocidade do desvanescimento do canal. Os resultados de simulação na Seção 3.6 demonstram este fato.

Seguindo com o desenvolvimento do algoritmo feito na introdução do capítulo, percebe-se que no RLS há a substituição do valor esperado por somatórios ponderados. Assim, aproveitando os cálculos já realizados, é possível rescrever (3-11) como

$$\boldsymbol{\nabla}_{\tilde{\mathbf{a}}}J_{RLS} = -2\sum_{l=1}^{i}\lambda^{i-l}\left\{\tilde{\mathbf{R}}^{\mathcal{H}}(l)\mathbf{W}_{M_1}\mathbf{T}(L_1)\mathbf{V}^T(L_1)\left[\mathbf{b}(l) - \mathbf{V}(L_1)\mathbf{T}^T(L_1)\mathbf{W}_{M_1}^{\mathcal{H}}\tilde{\mathbf{R}}(l)\tilde{\mathbf{a}}(i)\right]\right\}. \quad (3\text{-}29)$$

Igualando-se à zero o gradiente da função custo do RLS, encontra-se

$$\underbrace{\sum_{l=1}^{i}\lambda^{i-l}\tilde{\mathbf{R}}^{\mathcal{H}}(l)\mathbf{W}_{M_1}\mathbf{T}(L_1)\mathbf{V}^T(L_1)\mathbf{b}(l)}_{\boldsymbol{\chi}(i)} = \underbrace{\sum_{l=1}^{i}\lambda^{i-l}\tilde{\mathbf{R}}^{\mathcal{H}}(l)\mathbf{W}_{M_1}\mathbf{T}(L_1)\mathbf{V}^T(L_1)\mathbf{V}(L_1)\mathbf{T}^T(L_1)\mathbf{W}_{M_1}^{\mathcal{H}}\tilde{\mathbf{R}}(l)}_{\boldsymbol{\Phi}(i)}\tilde{\mathbf{a}}(i), \quad (3\text{-}30)$$

onde $\boldsymbol{\chi}(i)$ é um vetor de dimensão $M_1 \times 1$ tal que

$$\boldsymbol{\chi}(i) = \sum_{l=1}^{i} \lambda^{i-l} \tilde{\mathbf{R}}^{\mathcal{H}}(l) \mathbf{W}_{M_1} \mathbf{T}(L_1) \mathbf{V}^T(L_1) \mathbf{b}(l), \tag{3-31}$$

e

$$\boldsymbol{\Phi}(i) = \sum_{l=1}^{i} \lambda^{i-l} \tilde{\mathbf{R}}^{\mathcal{H}}(l) \mathbf{W}_{M_1} \mathbf{T}(L_1) \mathbf{V}^T(L_1) \mathbf{V}(L_1) \mathbf{T}^T(L_1) \mathbf{W}_{M_1}^{\mathcal{H}} \tilde{\mathbf{R}}(l) + \delta \lambda^i \mathbf{I}_{M_1}, \tag{3-32}$$

que é quadrada, de dimensão $M_1 \times M_1$. A introdução da última parcela de regularização na equação (3-32) tem o efeito de evitar problemas numéricos (singularidade ou valores muito baixos em uma das componentes da transformada do bloco recebido), tornando-a não-singular desde a primeira iteração do algoritmo. Isso é equivalente a rescrever a função custo do RLS como

$$J_{RLS} = \sum_{l=1}^{i} \lambda^{i-l} \|\mathbf{e}(l)\|^2 + \delta \lambda^i \|\tilde{\mathbf{a}}(l)\|^2. \tag{3-33}$$

O parâmetro δ tem valor baixo e λ está definido no intervalo $(0, 1)$, o que significa que rapidamente a energia dos coeficientes do filtro vai perdendo peso na minimização do algoritmo. A solução para os M_1 parâmetros do filtro adaptativo RLS são encontrados calculando

$$\tilde{\mathbf{a}}(i) = \boldsymbol{\Phi}^{-1}(i) \boldsymbol{\chi}(i). \tag{3-34}$$

O vetor $\boldsymbol{\chi}(i)$ e a matriz $\boldsymbol{\Phi}(i)$ podem ser encontrados de maneira recursiva, repetindo a mesma estratégia em ambos os casos. Para isso, vamos isolar o termo em que $l = i$ na equação (3-32):

$$\begin{aligned} \boldsymbol{\Phi}(i) &= \lambda \left[\sum_{l=1}^{i-1} \lambda^{i-1-l} \tilde{\mathbf{R}}^{\mathcal{H}}(l) \mathbf{W}_{M_1} \mathbf{T}(L_1) \mathbf{V}^T(L_1) \mathbf{V}(L_1) \mathbf{T}^T(L_1) \mathbf{W}_{M_1}^{\mathcal{H}} \tilde{\mathbf{R}}(l) + \delta \lambda^{i-1} \right] \\ &+ \tilde{\mathbf{R}}^{\mathcal{H}}(i) \mathbf{W}_{M_1} \mathbf{T}(L_1) \mathbf{V}^T(L_1) \mathbf{V}(L_1) \mathbf{T}^T(L_1) \mathbf{W}_{M_1}^{\mathcal{H}} \tilde{\mathbf{R}}(i) && \text{(3-35)} \\ &= \lambda \boldsymbol{\Phi}(i-1) + \tilde{\mathbf{R}}^{\mathcal{H}}(i) \mathbf{W}_{M_1} \mathbf{T}(L_1) \mathbf{V}^T(L_1) \mathbf{V}(L_1) \mathbf{T}^T(L_1) \mathbf{W}_{M_1}^{\mathcal{H}} \tilde{\mathbf{R}}(i), && \text{(3-36)} \end{aligned}$$

e o mesmo vale para o vetor $\boldsymbol{\chi}(i)$ em (3-31), ou seja,

$$\boldsymbol{\chi}(i) = \lambda \boldsymbol{\chi}(i-1) + \tilde{\mathbf{R}}^{\mathcal{H}}(i) \mathbf{W}_{M_1} \mathbf{T}(L_1) \mathbf{V}^T(L_1) \mathbf{b}(i). \tag{3-37}$$

Assim, é possível realizar o algoritmo de forma recursiva.

Para o caso de transmissão CP, as equações de atualização se tornam

$$\begin{aligned} \boldsymbol{\Phi}\big|_{CP}(i) &= \lambda \boldsymbol{\Phi}\big|_{CP}(i-1) + \tilde{\mathbf{R}}^{\mathcal{H}}(i) \tilde{\mathbf{R}}(i) && \text{(3-38)} \\ \boldsymbol{\chi}\big|_{CP}(i) &= \lambda \boldsymbol{\chi}\big|_{CP}(i-1) + \tilde{\mathbf{R}}^{\mathcal{H}}(i) \mathbf{W}_N \mathbf{b}(i), && \text{(3-39)} \end{aligned}$$

onde percebe-se que a matriz $\mathbf{\Phi}(i)$ à ser invertida é diagonal, e sendo assim, não muito pesada computacionalmente para calcular os coeficientes do filtro em (3-34). Já para o ZP, tem-se

$$\mathbf{\Phi}\big|_{ZP}(i) = \lambda\mathbf{\Phi}\big|_{ZP}(i-1) + \tilde{\mathbf{R}}^{\mathcal{H}}(i)\mathbf{W}_{MN}\mathbf{W}_{MN}^{\mathcal{H}}\tilde{\mathbf{R}}(i) \quad (3\text{-}40)$$

$$\boldsymbol{\chi}\big|_{ZP}(i) = \lambda\boldsymbol{\chi}\big|_{ZP}(i-1) + \tilde{\mathbf{R}}^{\mathcal{H}}(i)\mathbf{W}_{MN}\mathbf{b}(i). \quad (3\text{-}41)$$

Seria natural, apartir dos resultados obtidos até o momento, fazer a aproximação $\mathbf{W}_{MN}\mathbf{W}_{MN}^{\mathcal{H}} \approx \frac{N}{M}\mathbf{I}_M$ em (3-40), e obter assim, uma matriz $\mathbf{\Phi}(i)$ diagonal. Resultados de simulação como ilustra a Figura 3.2, mostram , no entanto, que esta aproximação causa no ZP-SC-FDE-RLS uma queda de desempenho considerável.

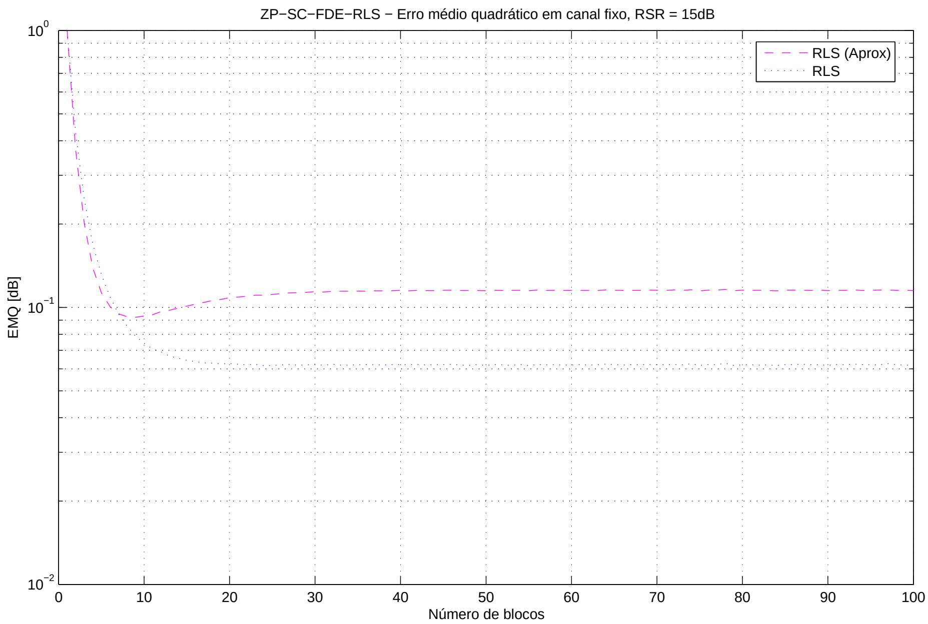

Figura 3.2: Erro médio quadrático dos algoritmos ZP-SC-FDE-RLS aproximado e original.

Ao não se fazer esta aproximação, o preço a ser pago é de um aumento no número de operações artiméticas no cálculo de (3-34) para o ZP. Todavia, o lema de inversão de matrizes possibilita calcular a inversa de $\mathbf{\Phi}(i)$ recursivamente, sem a necessidade da inversão de matrizes [11].

Assim, o algoritmo não desconsidera a correlação criada pelo produto da DFT truncada $\mathbf{W}_{MN}$ no bloco de dados observado na recepção. Ainda mais, o RLS é mais robusto à escolha dos parâmetros de atualização (no caso, o fator de esquecimento λ) e é o algoritmo escolhido para implementação da estrutura com laço de retorno apresentada no Capítulo 4. Estas estruturas são inerentementes instáveis, devido à realimentação de dados. A matriz $\mathbf{\Phi}(i)$ é inversível em todos os estágios devido ao fator de regularização introduzido em (3-32). Por conta da diferença de desempenho entre o RLS versão original e sua versão com a aproximação das DFTs truncadas, que leva o último a ter o pior desempenho entre os algoritmos recursivos, esta versão não é mais considerada nas análises que se seguem, adotando-se apenas o RLS original.

3.5 Modo orientado à decisão (Decision Directed)

Em canais variantes no tempo, é importante que haja um rastreamento das oscilações do canal, de forma a mitigar a perda de desempenho provocada pelo desvanecimento das componentes do canal. No modo orientado à decisão, as próprias estimativas $\hat{\mathbf{b}}(i)$ são utilizadas para realimentar o equalizador adaptativo. Mesmo sabendo que as estimativas $\hat{\mathbf{b}}(i)$ podem não ser réplicas

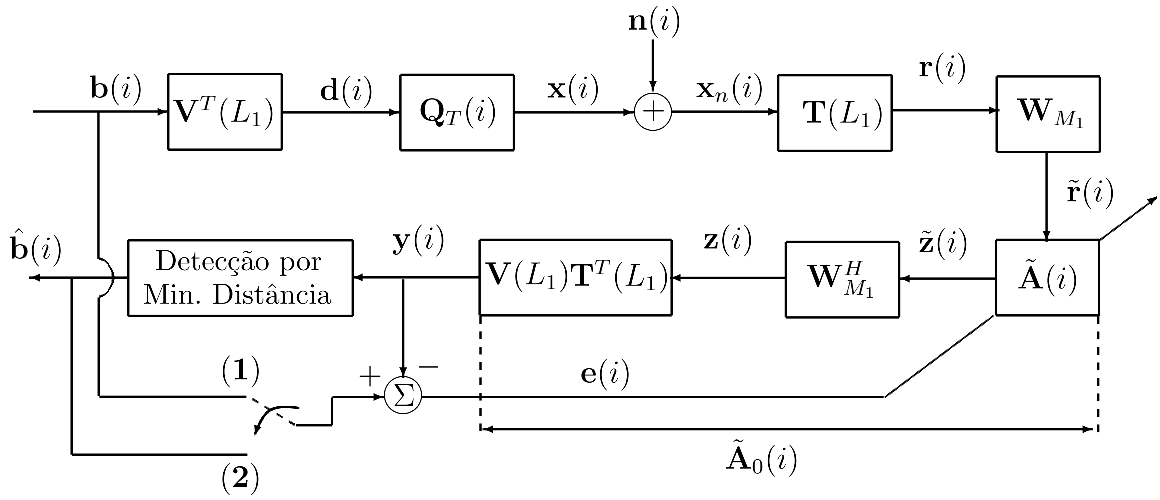

Figura 3.3: Estrutura da equalização adaptativa FDE com decisão direcionada.

fiéis dos dados enviados $\mathbf{b}(i)$ (o que faz com que algoritmo busque uma solução que ele considera certa, mas que na verdade não é). Observa-se nas curvas de desempenho que o *trade-off* ainda assim é consideravelmente favorável à esta técnica.

O estágio de treinamento não é descartado. Apenas agora, após este período inicial de treinamento com símbolos piloto, o sistema é chaveado (Figura 3.3) para alimentar o equalizador adaptativo com as estimativas do decisor. Durante o período inicial, o algoritmo trabalhará com mais precisão (treinamento, chave na posição **(1)**) utilizando-se de $\mathbf{b}(i)$, e depois, no modo orientado à decisão (chave na posição **(2)**), das estimativas $\hat{\mathbf{b}}(i)$.

3.6 Resultados de simulação

Nas figuras que se seguem, utiliza-se 100 blocos de treinamento (chave na posição **(1)**), e em seguida, mais 100 blocos na operação do sistema (chave na posição **(2)**).

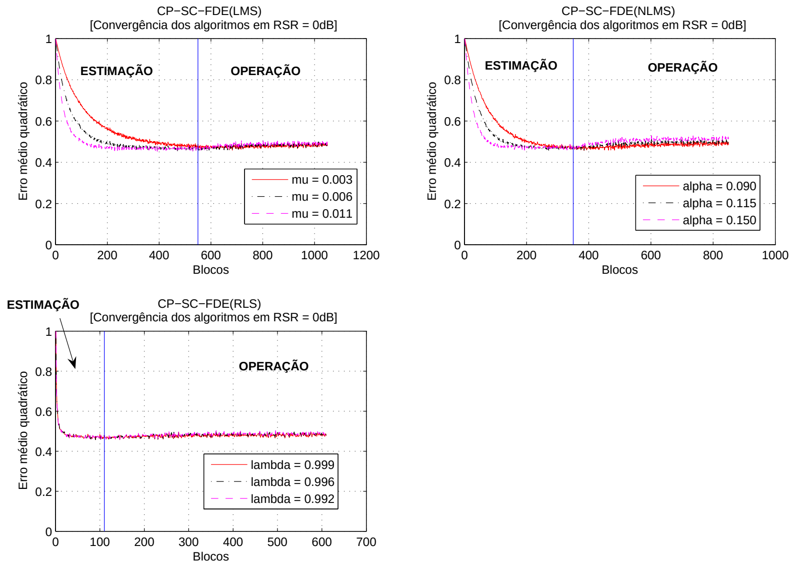

Figura 3.4: Erro médio quadrático dos algoritmos adaptativos nos sistemas CP-SC-FDE com RSR = 0dB

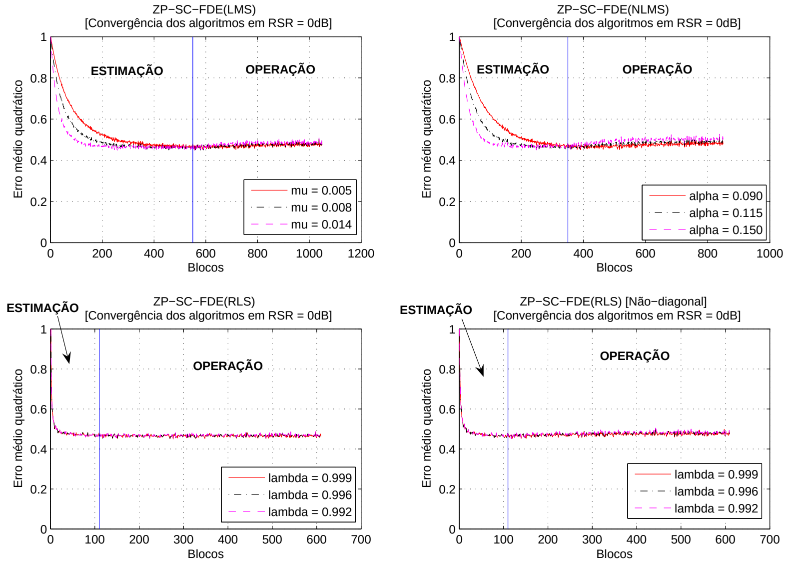

Figura 3.5: Erro médio quadrático dos algoritmos adaptativos nos sistemas ZP-SC-FDE com RSR = 0dB

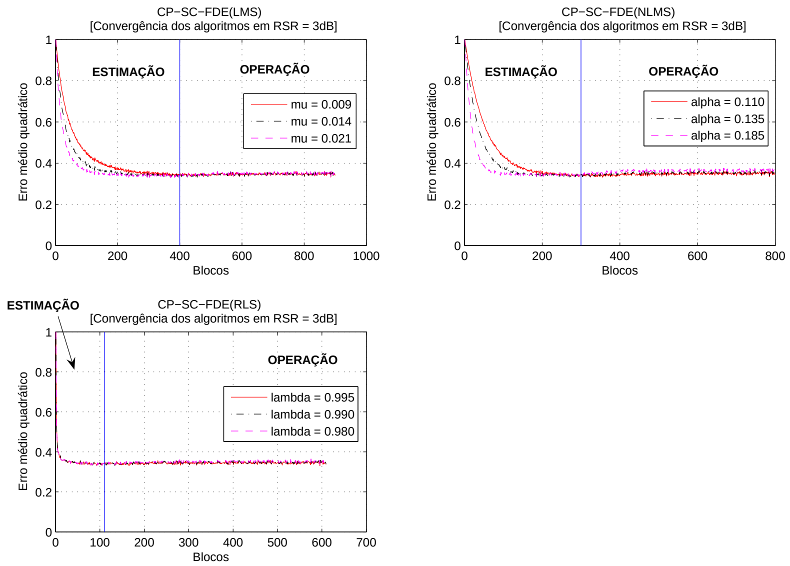

Figura 3.6: Erro médio quadrático dos algoritmos adaptativos nos sistemas CP-SC-FDE com RSR = 3dB

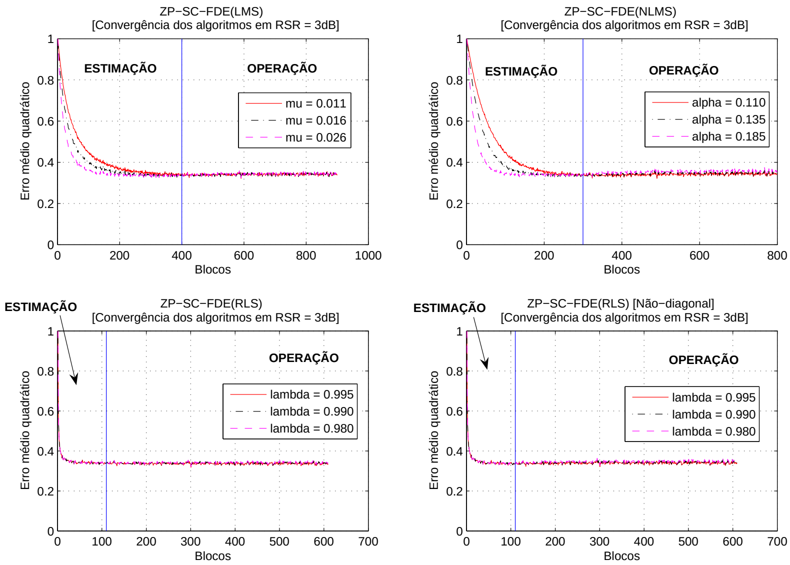

Figura 3.7: Erro médio quadrático dos algoritmos adaptativos nos sistemas ZP-SC-FDE com RSR = 3dB

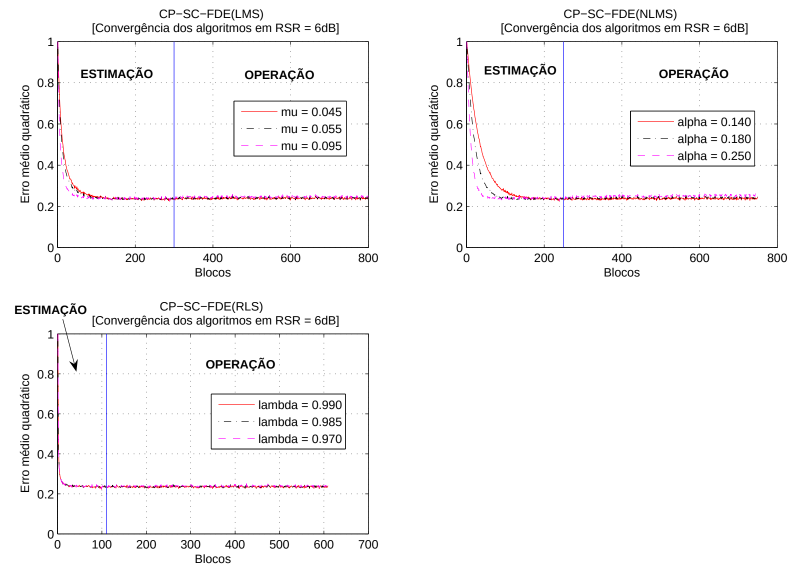

Figura 3.8: Erro médio quadrático dos algoritmos adaptativos nos sistemas CP-SC-FDE com RSR = 6dB

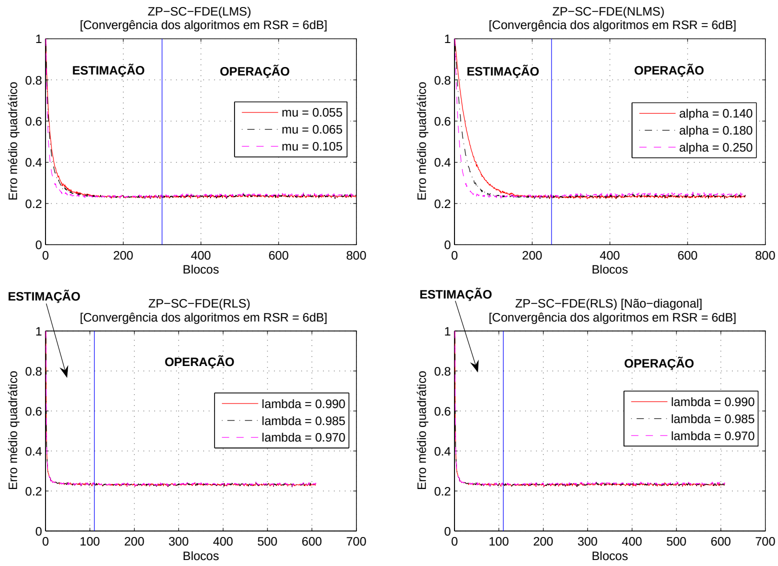

Figura 3.9: Erro médio quadrático dos algoritmos adaptativos nos sistemas ZP-SC-FDE com RSR = 6dB

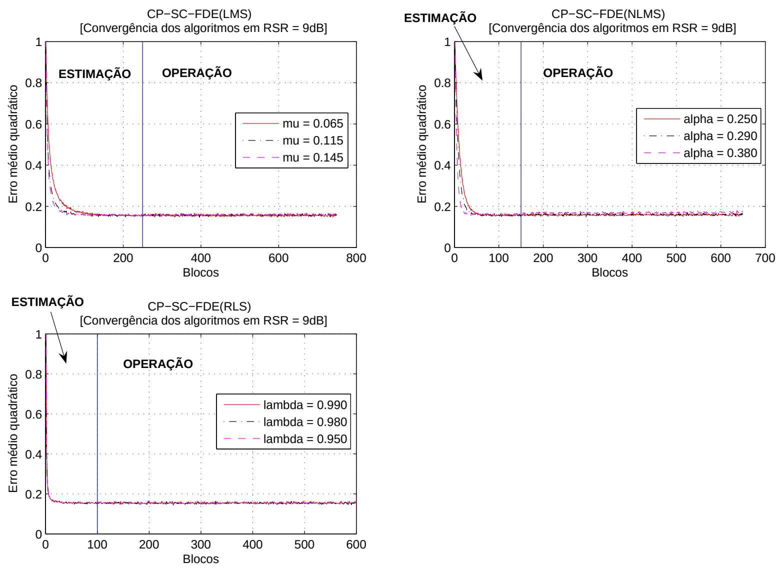

Figura 3.10: Erro médio quadrático dos algoritmos adaptativos nos sistemas CP-SC-FDE com RSR = 9dB

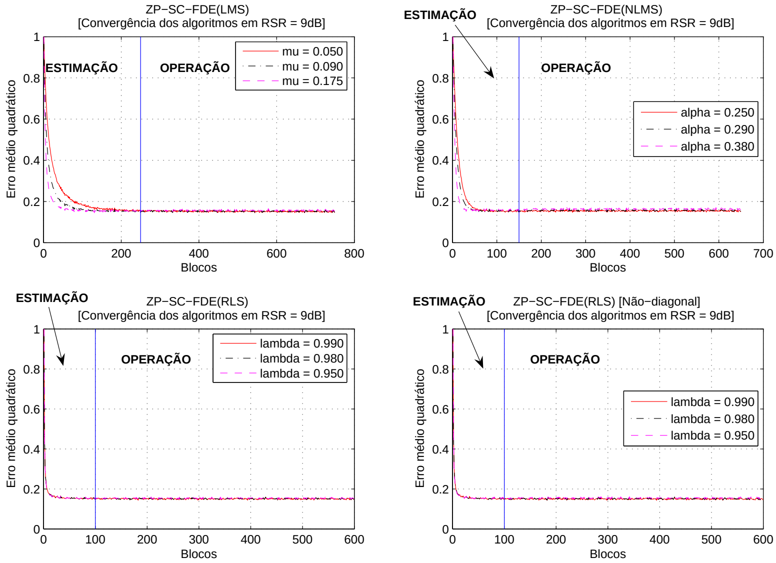

Figura 3.11: Erro médio quadrático dos algoritmos adaptativos nos sistemas ZP-SC-FDE com RSR = 9dB

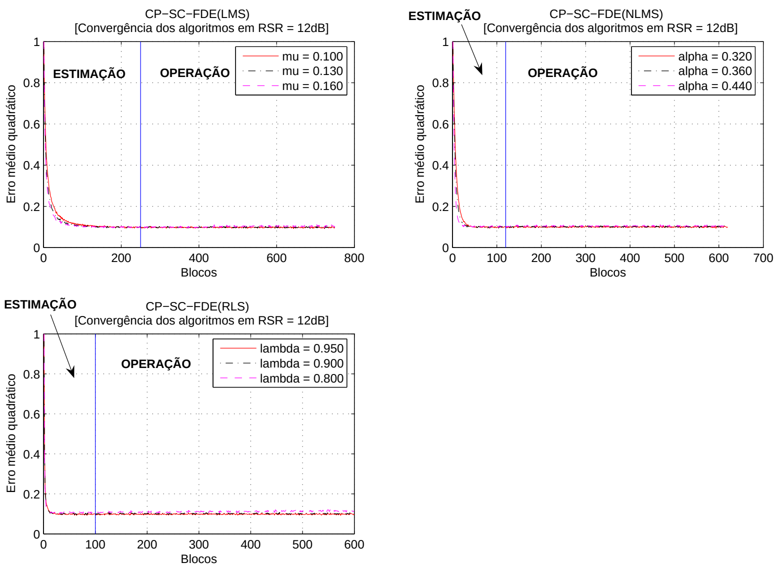

Figura 3.12: Erro médio quadrático dos algoritmos adaptativos nos sistemas CP-SC-FDE com RSR = 12dB

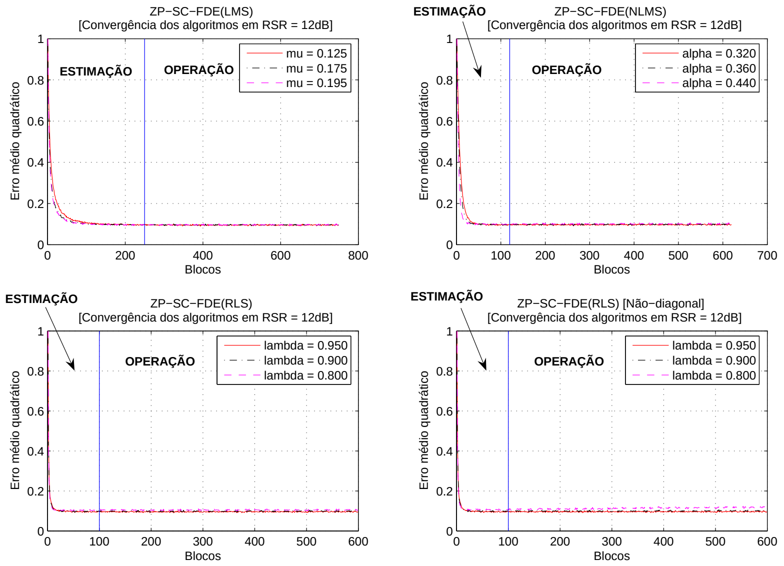

Figura 3.13: Erro médio quadrático dos algoritmos adaptativos nos sistemas ZP-SC-FDE com RSR = 12dB

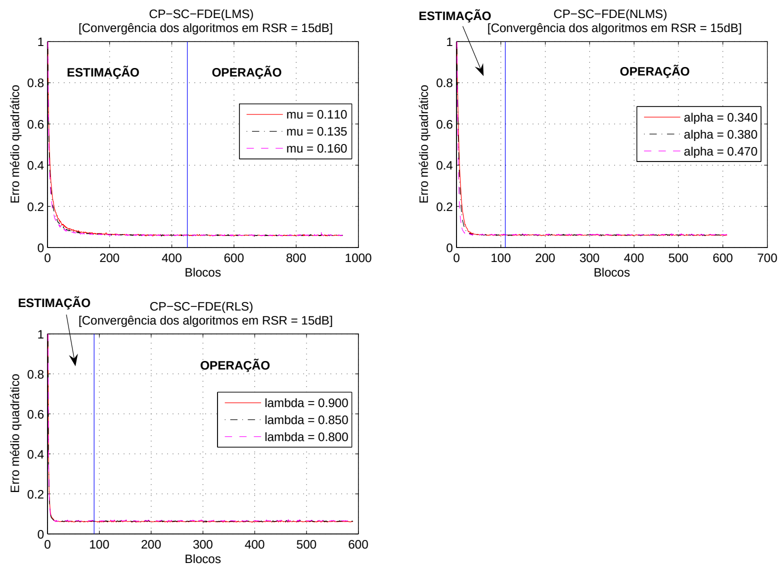

Figura 3.14: Erro médio quadrático dos algoritmos adaptativos nos sistemas CP-SC-FDE com RSR = 15dB

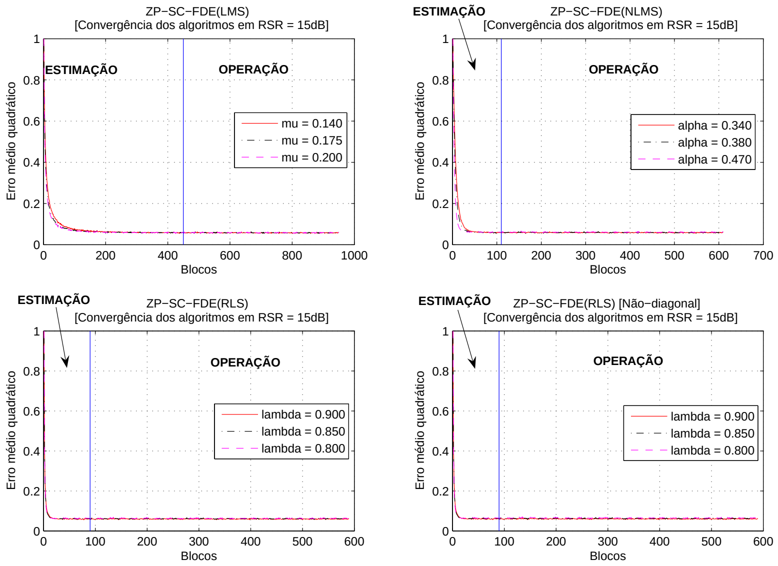

Figura 3.15: Erro médio quadrático dos algoritmos adaptativos nos sistemas ZP-SC-FDE com RSR = 15dB

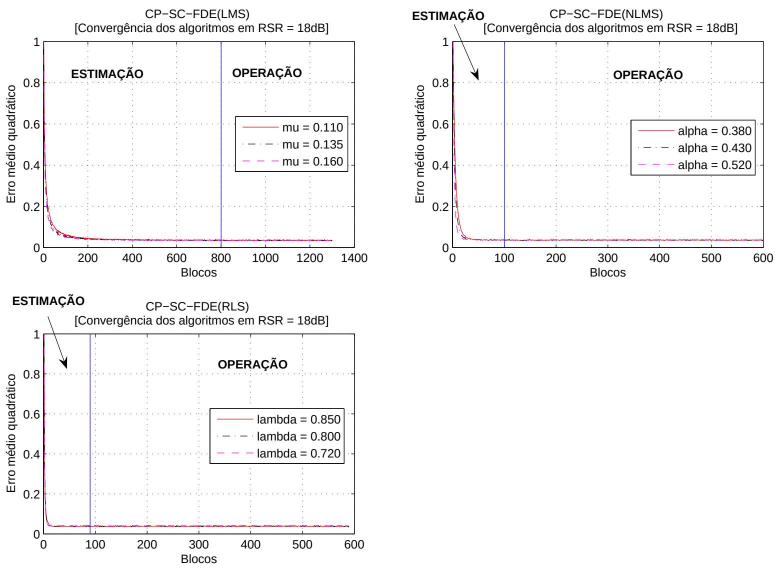

Figura 3.16: Erro médio quadrático dos algoritmos adaptativos nos sistemas CP-SC-FDE com RSR = 18dB

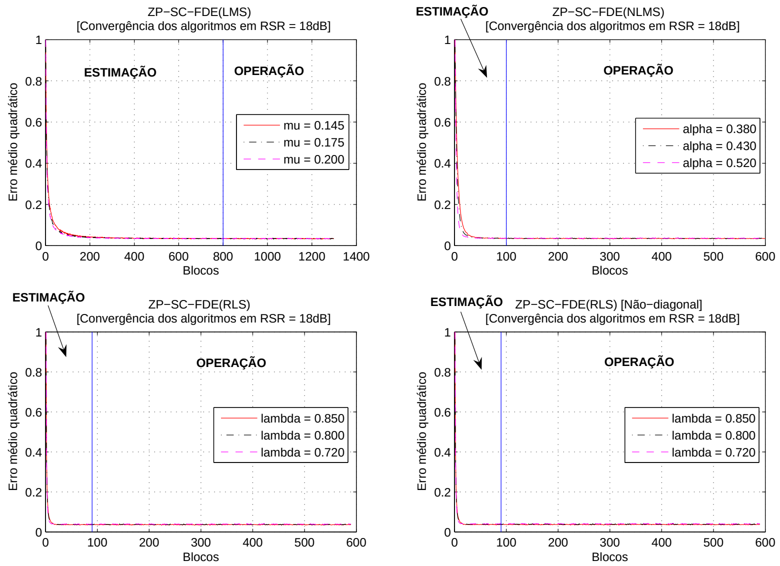

Figura 3.17: Erro médio quadrático dos algoritmos adaptativos nos sistemas ZP-SC-FDE com RSR = 18dB

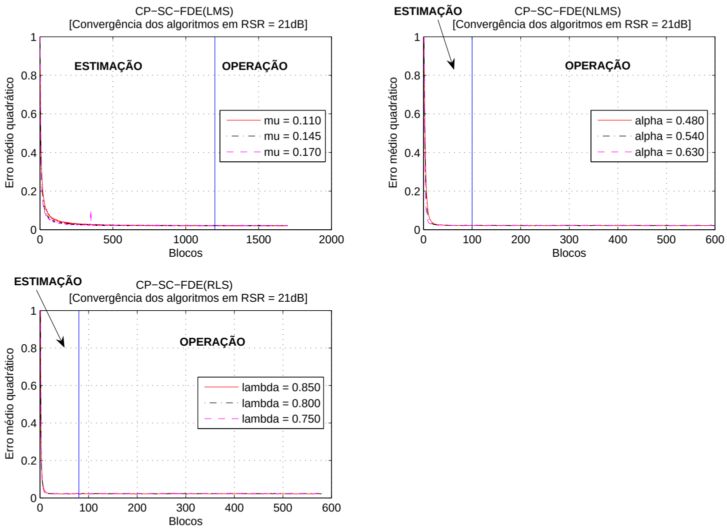

Figura 3.18: Erro médio quadrático dos algoritmos adaptativos nos sistemas CP-SC-FDE com RSR = 21dB

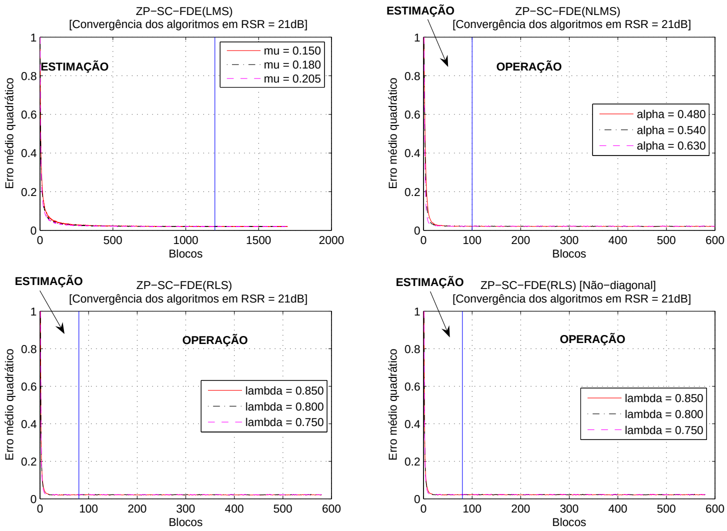

Figura 3.19: Erro médio quadrático dos algoritmos adaptativos nos sistemas ZP-SC-FDE com RSR = 21dB

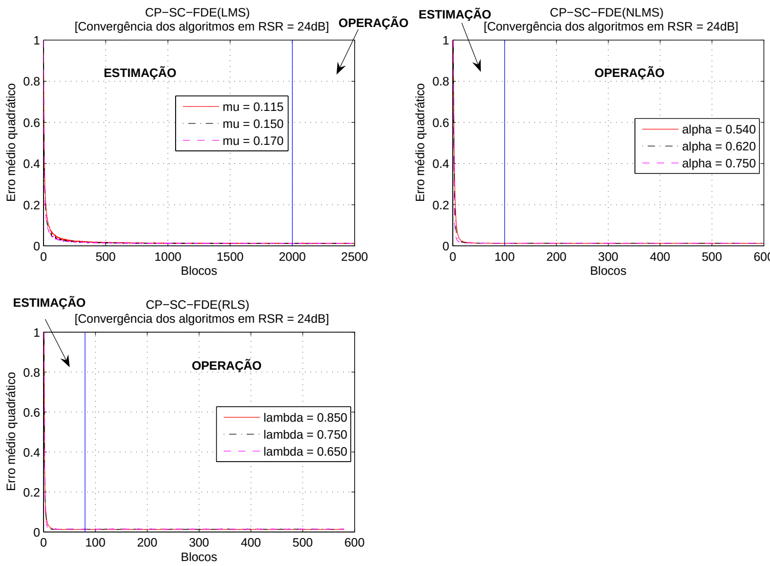

Figura 3.20: Erro médio quadrático dos algoritmos adaptativos nos sistemas CP-SC-FDE com RSR = 24dB

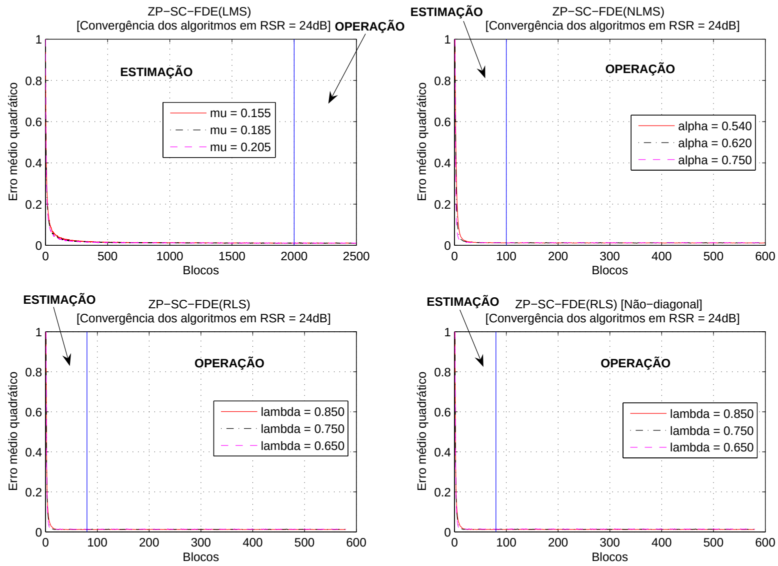

Figura 3.21: Erro médio quadrático dos algoritmos adaptativos nos sistemas ZP-SC-FDE com RSR = 24dB

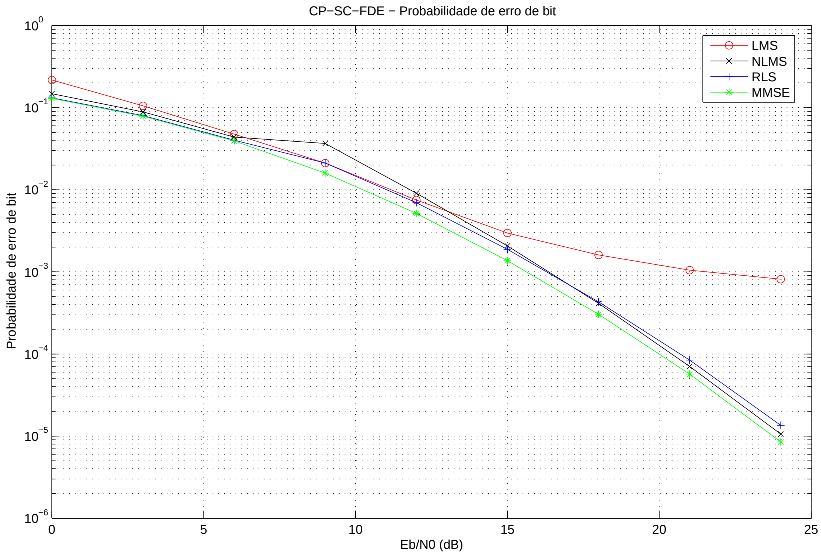

Figura 3.22: CP-SC-FDE: Comparativo entre a BER para os diferentes algoritmos adaptativos em canal fixo

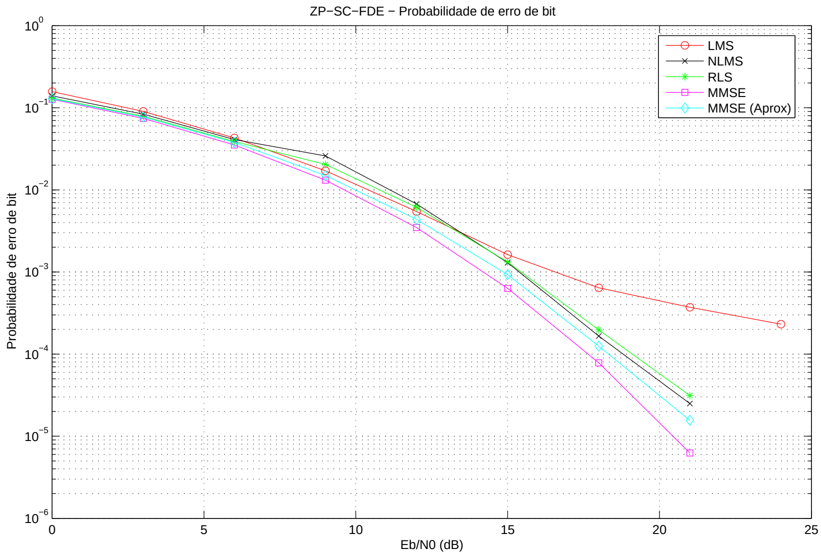

Figura 3.23: ZP-SC-FDE: Comparativo entre a BER para os diferentes algoritmos adaptativos em canal fixo

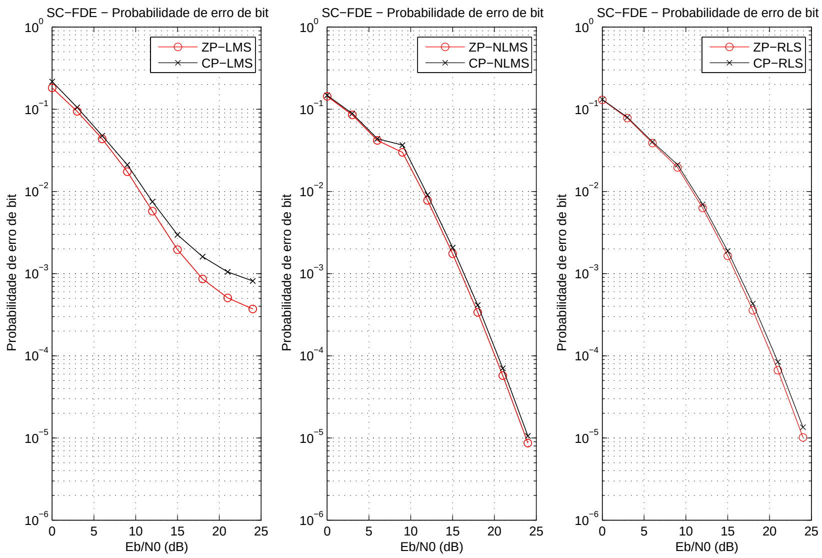

Figura 3.24: SC-FDE: Comparativo entre a BER do ZP e CP para os diferentes algoritmos em canal fixo

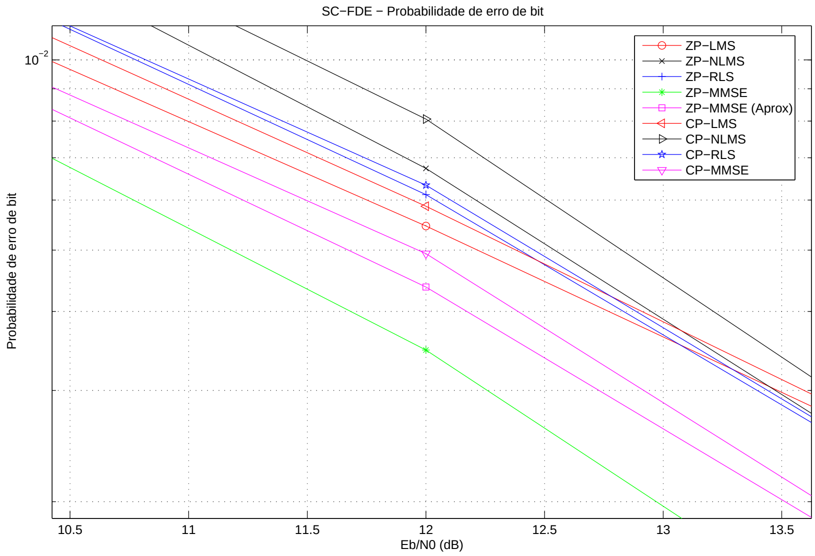

Figura 3.25: SC-FDE: Comparativo entre a BER do ZP e CP para os diferentes algoritmos em canal fixo (zoom em torno de 12dB)

3.6.1
Estimação com seqüência de treinamento reduzida

Nessa implementação alternativa, faz-se uso de um pequeno número de blocos de estimação, em que o algoritmo é chaveado para o modo *decision-directed* (chave na posição **(2)**), e contamos com que as estimativas $\hat{\mathbf{b}}(i)$ dos blocos transmitidos sejam boas o suficiente para que a convergência dos algoritmos prossiga com sucesso, utilizando apenas os símbolos detectados.
Na verdade, estamos apenas introduzindo um chute inicial com maior acuidade, enquanto o algoritmo estiver usando os verdadeiros símbolos enviados. O restante do trabalho é feito pelas estimativas $\hat{\mathbf{b}}(i)$ dos blocos transmitidos.
Esta tática se mostra ainda mais recomendada em canais que variam no tempo, onde não faz muito sentido fazer longas equalizações com muitos blocos-piloto, uma vez que há variações do canal de transmissão à taxa de blocos do sistema.

3.6.2
Canal fixo

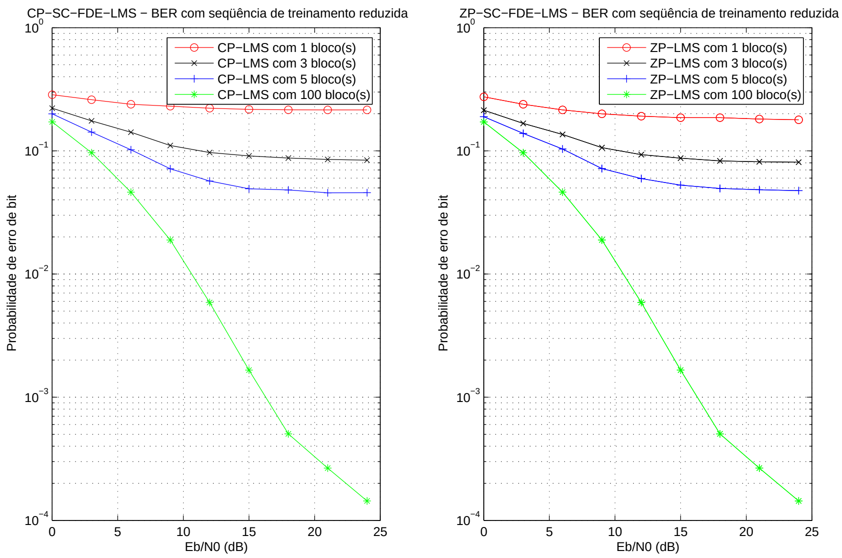

Figura 3.26: SC-FDE-LMS: Comparativo entre a BER do ZP e CP para os diferentes algoritmos em canal fixo com seqüências de treinamento reduzida.

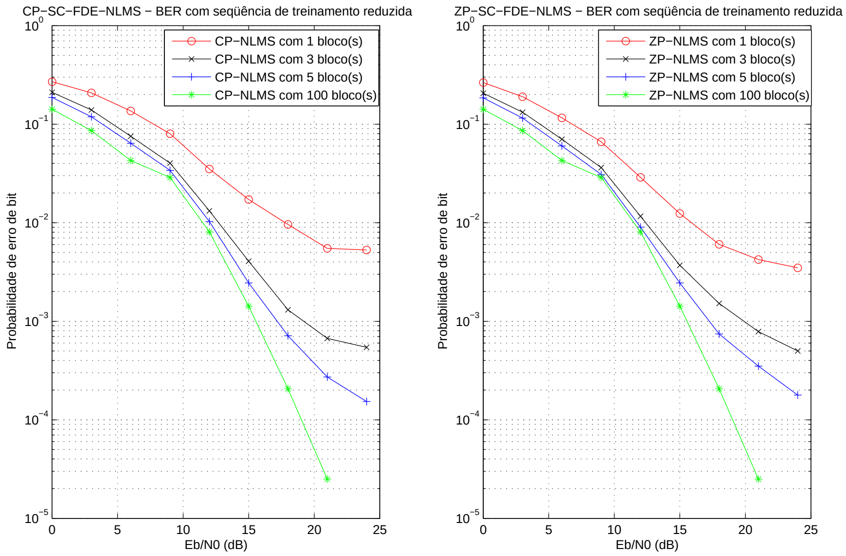

Figura 3.27: SC-FDE-NLMS: Comparativo entre a BER do ZP e CP para os diferentes algoritmos em canal fixo com seqüências de treinamento reduzida.

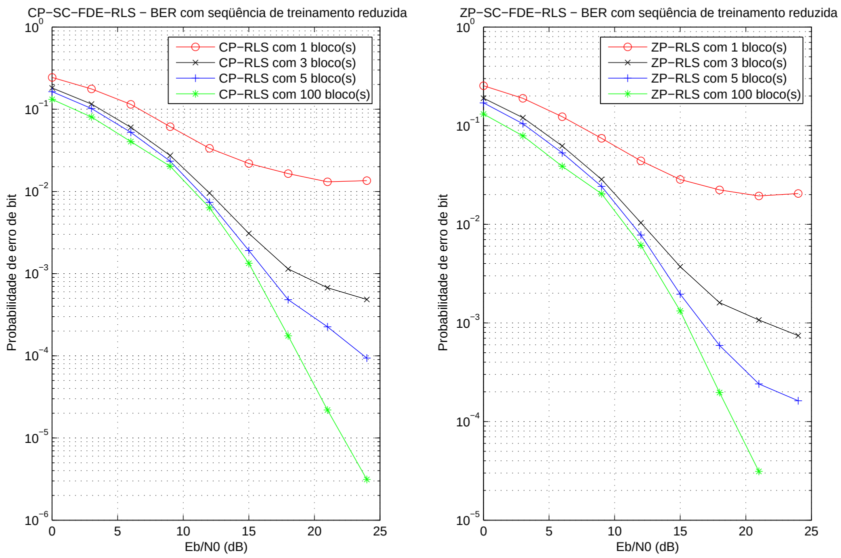

Figura 3.28: SC-FDE-RLS: Comparativo entre a BER do ZP e CP para os diferentes algoritmos em canal fixo com seqüências de treinamento reduzida.

3.7 Considerações Finais

Verifica-se neste capítulo que o algoritmo RLS apresenta desempenho superior quando comparado com o LMS e NLMS, sob qualquer razão sinal-ruído analisada e sob qualquer tipo de canal (fixo ou aleatório). Por conta disso, a partir de agora, não mais os consideraremos nas estruturas que se seguem, onde nos ateremos apenas ao algoritmo RLS.

É visto, também, que é possível iniciar os equalizadores com um número reduzido de blocos de treinamento, desde que a operação seja no modo orientado à decisão, onde as próprias estimativas do receptor são utilizadas para atualizar os filtros adaptativos (equalizadores). Estes resultados ficam razoáveis principalmente para sistemas SC-FDE-RLS, como apresentado nas figuras 3.26, 3.27, 3.28.