

Equalização Adaptativa Cega de Sinais com Kernel não-Gaussiano

Lucas Gois e Aline Neves

Resumo— A equalização baseada em aprendizagem por Teoria de Informação, por utilizar mais completamente a informação dos sinais envolvidos no processo, é capaz de resolver casos que técnicas clássicas não conseguem, como no caso da transmissão de sinais correlacionados. Estes métodos, no entanto, necessitam da estimação da função densidade de probabilidade dos sinais em tempo real. Normalmente, estimadores baseados no uso de *kernels* são utilizados. Na literatura, a função utilizada é sempre a Gaussiana. No entanto, esta função não é ótima. Neste trabalho aplicamos o kernel Epanechnikov para a estimação e mostramos como o desempenho dos algoritmos pode melhorar com esta nova função.

Palavras-Chave — Aprendizagem por Teoria da Informação, distância entre pdfs, kernel Epanechnikov

Abstract— Information Theoretic Learning based equalization is capable of solving problems where classic methods fail such as in the case of correlated signals. Nonetheless, these methods need the estimation of the signals probability density function. Usually, kernel based estimators are used, with a Gaussian kernel. However, such function is not optimum. In this work, we apply an Epanechnikov kernel to the estimation and show how the performance of the algorithms may improve

Keywords — Information Theoretic Learning, PDF Matching, Epanechnikov kernel

I. INTRODUÇÃO

O processamento digital de sinais desempenha um papel importante na atualidade, sendo um dos pilares da sociedade moderna. Um sinal, ao ser transmitido, passa por canais de transmissão não ideais que geram distorções no sinal. Visando minimizar o efeito da interferência intersimbólica sofrida pelo sinal original, é proposto algum tipo de processamento no receptor para compensar as distorções sofridas, como a equalização. As técnicas de equalização podem ser divididas em duas categorias: a equalização supervisionada e a não supervisionada.

Na equalização supervisionada há uma sequência de símbolos, conhecida pelo receptor, que é transmitida e usada como referência na adaptação do filtro equalizador. Na equalização não supervisionada ou cega, somente as características estatísticas do sinal transmitido são conhecidas pelo receptor e utilizadas no processo de equalização.

Recentemente, critérios baseados em teoria de informação têm sido utilizados para equalização não supervisionada [7]. Como tais critérios utilizam mais informação sobre os sinais disponíveis, acabam conseguindo resolver casos que as técnicas clássicas não conseguem, como o caso no qual o sinal

transmitido é correlacionado ou casos em que o ruído é impulsivo [7]. Os critérios e algoritmos desenvolvidos neste contexto são chamados de *aprendizagem por teoria da informação* (ITL - Information Theoretic Learning). Um dos critérios de equalização mais utilizados neste contexto consiste em se comparar as densidades de probabilidade dos sinais de saída do equalizador e do sinal transmitido (SQD – Stochastic Quadratic Distance) [1,6]. A estimação das densidades de probabilidade necessárias é feita utilizando-se a janela de Parzen, que nada mais é do que um estimador baseado no uso de kernels. Até o momento, os trabalhos presentes na literatura utilizam somente o kernel gaussiano para executar tal tarefa [3,5,7].

No entanto, conforme discutido em [2,4], o kernel gaussiano não é o melhor quando se trata de estimar densidades de probabilidade. Segundo [2], o kernel Epanechnikov é ótimo neste sentido. Sendo assim, este trabalho desenvolve o algoritmo SQD e SQD modificado [1,8] com este novo kernel, e analisa o seu desempenho quando comparado ao obtido com o uso do kernel gaussiano em diversos cenários de simulação.

Este artigo está organizado da seguinte forma. A seção II contempla os algoritmos baseados em ITL SQD e MSQD, assim como suas funções custo utilizando kernel gaussiano. Na seção III apresentamos o kernel Epanechnikov e os novos algoritmos gerados a partir dele, seguido da seção IV que apresenta os resultados de simulações. Por fim, na seção V é apresentada a conclusão do trabalho.

II. ALGORITMOS BASEADOS EM APRENDIZAGEM POR TEORIA DA INFORMAÇÃO

A Fig 1 abaixo ilustra, através de um diagrama de blocos, o sistema em estudo. O sinal, $s(n)$, é transmitido por um canal com resposta ao impulso $h(n)$, resultando no sinal $x(n)$. A fim de recuperar o sinal inicialmente transmitido, utilizamos um equalizador, $w(n)$. A saída deste filtro é dada por:

$$y(n) = \mathbf{w}^T(n)\mathbf{x}(n)$$

onde $\mathbf{w}(n)=[w_0 \ w_1 \ \dots \ w_N]$ é o vetor com os coeficientes do equalizador e $\mathbf{x}(n) = \mathbf{h}^T(n)\mathbf{s}(n) = [x(n) \ x(n-1) \ \dots \ x(n-N)]$.

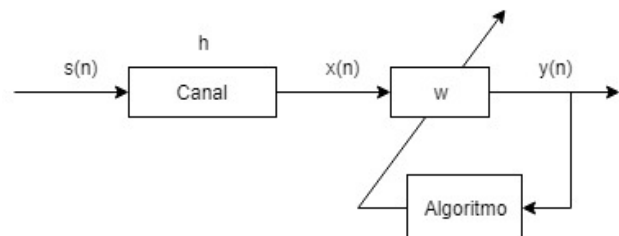


Fig.1 Diagrama de blocos do sistema em estudo.

Em seguida discutiremos os algoritmos a serem estudados neste trabalho.

A. Algoritmo da Distância Quadrática

Um dos critérios mais utilizados dentro do contexto de ITL é o critério baseado na distância quadrática entre a função densidade de probabilidade (pdf) do sinal na saída do equalizador e do sinal transmitido. Tal critério, proposto inicialmente em [1], foi denominado SQD – Stochastic Quadratic Distance. Sua função custo pode ser definida como [1,6]:

$$J_{SQD}(w) = \int_{-\infty}^{\infty} (f_y(z) - f_s(z))^2 dz \quad (1)$$

onde f_y e f_s são, respectivamente, as pdf's do sinal de saída do equalizador e do sinal transmitido inicialmente, $s(n)$. Para estimar f_y e f_s , é utilizado o método da janela de Parzen [1]:

$$\hat{f}_z(z) = \frac{1}{L} \sum_{i=1}^L K(z - z_i) \quad (2)$$

onde L é o número de símbolos utilizados para a estimação e $K(\cdot)$ é a função utilizada como kernel. Em [1,3,7] e nos trabalhos encontrados na literatura, a função utilizada é sempre a gaussiana. Assim, seguindo a proposta de [2], o kernel gaussiano é dado por:

$$K_G(A) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{A^2}{2\sigma^2}} \quad (3)$$

onde σ é o desvio padrão e A é uma variável aleatória qualquer. O parâmetro σ também é conhecido como tamanho ou largura do kernel e tem um papel crucial na convergência do método como será discutido na seção IV. Como o critério dado em (1) é uma função quadrática, é necessário calcular a convolução entre dois kernels. Neste processo, o kernel gaussiano apresenta uma importante vantagem, pois possui a seguinte propriedade [1]:

$$\int_{-\infty}^{\infty} K_G(y - A_1) K_G(y - A_2) dy = K_{G_{\sigma\sqrt{2}}}(A_1 - A_2) \quad (4)$$

onde $\sigma\sqrt{2}$ passa a ser o tamanho do kernel. Logo, estimando as pdfs em (1) através de (2), e levando (4) em consideração, chegamos em [1,6]:

$$\begin{aligned} J_{SQD}(w) = & \frac{1}{L^2} \sum_{i=0}^{L-1} \sum_{j=0}^{L-1} K_{G_{\sigma\sqrt{2}}}(|y(n-j)|^2 - |y(n-i)|^2) + \\ & \frac{1}{N_s^2} \sum_{i=1}^{N_s} \sum_{j=1}^{N_s} K_{G_{\sigma\sqrt{2}}}(|d_i|^2 - |d_j|^2) - \\ & \frac{2}{LN_s} \sum_{i=1}^{N_s} \sum_{j=0}^{L-1} K_{G_{\sigma\sqrt{2}}}(|y(n-j)|^2 - |d_i|^2) \end{aligned} \quad (5)$$

onde N_s é o número de símbolos da constelação do sinal transmitido, enquanto que d_i são os símbolos da constelação.

Para simplificar a notação, de agora em diante, sempre que tratarmos do kernel gaussiano, faremos $\sigma = \sigma\sqrt{2}$ e, portanto denotaremos o kernel simplesmente por K_G .

Utilizando o método do gradiente descendente, chegamos a equação de atualização dos pesos do equalizador:

$$w(n+1) = w(n) - \mu \nabla_w J_{SQD}(w) \quad (6)$$

onde μ representa o passo de adaptação e $\nabla_w J_{SQD}(w)$ é o gradiente de (5). Este algoritmo será denominado SQD-G, por utilizar o kernel gaussiano.

B. Algoritmo da Distância Quadrática Modificado

O algoritmo SQD Modificado (MSQD), leva em conta apenas o termo cruzado da função custo do algoritmo SQD, já que é este termo que efetivamente mede a distância entre as pdfs do sinal de saída do equalizador e do sinal transmitido [1,7,8]. Assim, sua função custo é dada pelo último termo de (5):

$$J_{MSQD}(w) = -\frac{2}{LN_s} \sum_{i=1}^{N_s} \sum_{j=0}^{L-1} K_G(|y(n-j)|^2 - |d_i|^2) \quad (8)$$

O gradiente de (8) é dado por:

$$\nabla_w J_{MSQD}(w) = -\frac{2}{LN_s} \sum_{i=1}^{N_s} \sum_{j=0}^{L-1} S_{yd} S_{yx} K_G(S_{yd}) \quad (9)$$

onde $S_{yd} = |y(n-j)|^2 - |d_i|^2$ e $S_{yx} = y(n-j)x^*(n-j)$ e a equação de atualização dos coeficientes do equalizador segue o que foi feito em (6). Este algoritmo será denotado MSQD-G.

III. KERNEL EPANECHNIKOV

Segundo o estudo realizado em [2], utilizando-se como medida a média da integral do erro médio quadrático (AMISE – Average Mean Integrated Squared Error) da estimação de pdfs, o kernel Epanechnikov é ótimo para a estimação de densidades de probabilidade. O kernel Epanechnikov é dado por:

$$K_E(A) = \frac{3}{4\sigma} \left(1 - \left(\frac{A}{\sigma} \right)^2 \right) \quad (10)$$

onde σ é a largura do kernel.

No entanto, para podermos aplicá-lo à estimação do critério SQD, primeiramente precisamos calcular o resultado da convolução entre dois kernels, de forma equivalente ao que foi feito em (4) para o kernel gaussiano. Utilizando-se, portanto, (10) e considerando o caso real, chegamos a:

$$\begin{aligned} \int_{-\sigma}^{\sigma} \frac{3}{4\sigma} \left(1 - \left(\frac{y - y(n-i)}{\sigma} \right)^2 \right) \frac{3}{4\sigma} \left(1 - \left(\frac{y - y(n-j)}{\sigma} \right)^2 \right) dy = \\ \frac{9}{8\sigma^3} \left[\frac{8\sigma^4}{15} - \frac{2\sigma^2}{3} (S_{yy})^2 + M_{yy}^4 \right] \end{aligned} \quad (11)$$

onde $S_{yy} = |y(n-j)|^2 - |y(n-i)|^2$ e $M_{yy} = y(n-i)y(n-j)$. Assim, poderíamos reescrever (5) usando a expressão acima:

$$\begin{aligned} J_{\text{SQD-Epa}}(w) = & \frac{1}{L^2} \sum_{i=0}^{L-1} \sum_{j=0}^{L-1} \frac{9}{8\sigma^3} \left[\frac{8\sigma^4}{15} - \frac{2\sigma^2}{3} (S_{yy})^2 + M_{yy}^4 \right] \\ & + \frac{1}{N_s^2} \sum_{i=1}^{N_s} \sum_{j=1}^{N_s} \frac{9}{8\sigma^3} \left[\frac{8\sigma^4}{15} - \frac{2\sigma^2}{3} (S_{dd})^2 + d_i^4 d_j^4 \right] \\ & - \frac{2}{LN_s} \sum_{i=1}^{N_s} \sum_{j=0}^{L-1} \frac{9}{8\sigma^3} \left[\frac{8\sigma^4}{15} - \frac{2\sigma^2}{3} (S_{yd})^2 + M_{yd}^4 \right] \end{aligned} \quad (12)$$

onde $S_{dd} = |d_j|^2 - |d_i|^2$, $M_{yd} = y(n-j)d_i$ e S_{yd} foi definido em (9). A atualização dos coeficientes do equalizador é feita usando-se o método gradiente descendente a partir da função acima, assim como foi feito em (6). Este algoritmo será denominado SQD-Epa.

Para desenvolvimento do MSQD com o kernel Epanechnikov, denominado MSQD-Epa, basta considerarmos somente o último termo de (12). O gradiente será dado então por:

$$\begin{aligned} \nabla_w J_{\text{MSQD-Epa}}(w) = & -\frac{2}{LN_s} \sum_{i=1}^{N_s} \sum_{j=0}^{L-1} \frac{9}{8\sigma^3} \left(\frac{-4\sigma^2}{3} S_{yd} S_{yx} \right) \\ & + 4y^3(n-j) d_i^4 x(n-j) \end{aligned} \quad (13)$$

Na próxima seção, apresentaremos alguns resultados de simulação comparando o SQD e o MSQD originais com suas versões obtidas com o kernel Epanechnikov.

IV. RESULTADOS DE SIMULAÇÃO

Foram utilizados dois canais para simulações de desempenho: um canal de fase mista $C_1(z) = 0.2 + z^{-1} + 0.5z^{-2}$ e um canal de fase mínima $C_2(z) = 1 + 0.5z^{-1} + 0.1z^{-2}$. Para todas as simulações, foi utilizado um vetor w com 21 coeficientes, inicializado com *center spike*, ou seja, o coeficiente $w_{11} = -1$ e todo o resto nulo.

Como medida de desempenho, utilizamos a interferência intersimbólica residual (IIS) definida como:

$$IIS(n) = \frac{\sum_j |\vartheta_j(n)|^2 - \max_j (|\vartheta_j(n)|^2)}{\max_j (|\vartheta_j(n)|^2)} \quad (14)$$

onde $\vartheta(n) = [\vartheta_0(n) \ \vartheta_1(n) \ \dots \ \vartheta_M(n)]$ é a resposta combinada entre o canal e o equalizador no instante de tempo n .

Iniciamos a análise do algoritmo utilizando um sinal $s(n)$ descorrelacionado, dado pela modulação BPSK (*Binary Phase Shift Keying*). Em seguida, passamos para o uso de um sinal correlacionado.

Para o caso do SQD-G e MSQD-G, como os algoritmos já foram bastante estudados [1], [4], foi considerada a escolha de parâmetros que levam ao melhor desempenho dos algoritmos para cada caso simulado. Os resultados mostrados são uma média de 20 simulações.

A. Sinais Descorrelacionados

A fim de analisar o desempenho do algoritmo com relação aos seus parâmetros, i.e., passo de adaptação μ e largura do kernel, σ , começamos variando estes parâmetros e observando o desempenho do algoritmo. Como o sinal $s(n)$ é descorrelacionado, mantivemos $L=2$ nestas primeiras simulações. Todas as figuras mostram também o desempenho obtido pelo algoritmo com o kernel gaussiano. Os melhores parâmetros, neste caso, foram obtidos do trabalho realizado em [3]. A notação Kg foi usada para simulações com o kernel gaussiano e Ke para o kernel Epanechnikov.

A Fig.2 mostra como a variação do passo afeta o desempenho do SQD para o canal $C_1(z)$, mantendo σ fixo igual a 20. Vemos que, em geral, o passo necessário para que o SQD-Epa convirja é bem menor do que o passo usado para o SQD-G. Ainda assim, a figura já mostra como o SQD-Epa pode apresentar um desempenho melhor do que o algoritmo com o kernel gaussiano. O passo $\mu = 1 \times 10^{-5}$ é o maior valor que pode ser usado para o SQD-Epa que ainda garante a convergência do algoritmo.

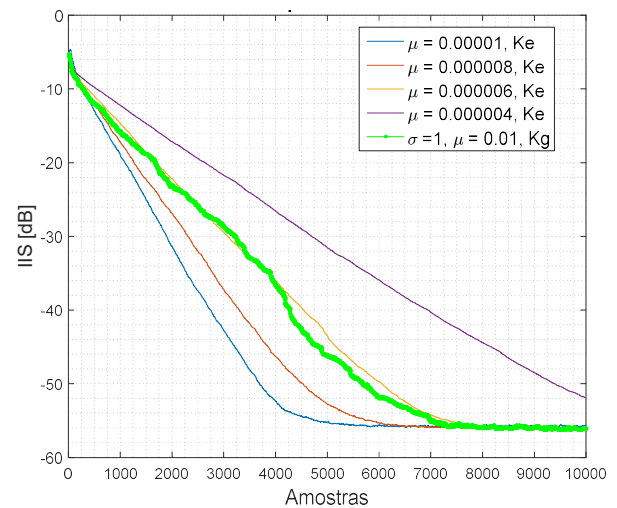


Fig. 2. Desempenho do algoritmo SQD variando o passo de adaptação, para um sinal BPSK utilizando o canal $C_1(z) = 0.2 + z^{-1} + 0.5z^{-2}$.

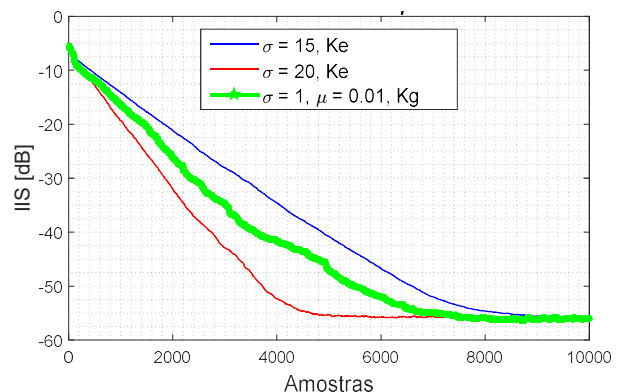


Fig. 3. Desempenho do algoritmo SQD variando o σ , para um sinal BPSK utilizando o canal $C_1(z) = 0.2 + z^{-1} + 0.5z^{-2}$.

Ainda mantendo $C_1(z)$, a Fig. 3 ilustra como o algoritmo se comporta conforme variamos o valor de σ , mantendo o passo fixo em $\mu = 1 \times 10^{-5}$. Diferentemente do SQD-G, para o qual $\sigma=1$ leva aos melhores resultados, para o SQD-Epa, o uso de valores bem maiores, chegando a $\sigma=20$, leva o

algoritmo a apresentar melhores desempenhos. Para valores abaixo desse, a velocidade de convergência do algoritmo tende a diminuir.

A Fig. 4 abaixo ilustra o desempenho do algoritmo SQD-Epa, conforme variamos o tamanho da janela temporal, L , para o caso do canal $C_2(z)$, $\mu = 1 \times 10^{-5}$ e $\sigma = 20$. Como o sinal utilizado aqui é decorrelacionado, é de se esperar que o algoritmo consiga bons desempenhos com valores baixos para L . Conforme mostra a Fig. 4, o melhor desempenho é obtido para $L=2$, tanto para o caso com kernel gaussiano como para o caso com kernel Epanechnikov. Ainda, por estarmos utilizando neste caso um canal de fase mínima, o desempenho das duas versões do algoritmo conseguem atingir desempenhos bastante semelhantes.

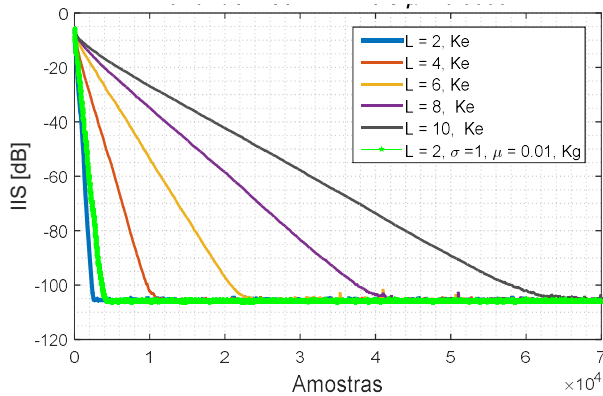


Fig. 4. Desempenho do algoritmo SQD variando o L , para um sinal BPSK utilizando o canal $C_2(z) = 1 + 0.5z^{-1} + 0.1z^{-2}$.

Já para o algoritmo MSQD, a Fig. 5 ilustra a variação do tamanho da janela, L , já considerando o melhor valor de passo para cada caso e $C_2(z)$. Para o MSQD-Epa, $\sigma = 20$ e mantivemos $\sigma = 1$ para o MSQD-G. A figura também inclui o melhor desempenho obtido pelo SQD-Epa, para facilitar a comparação.

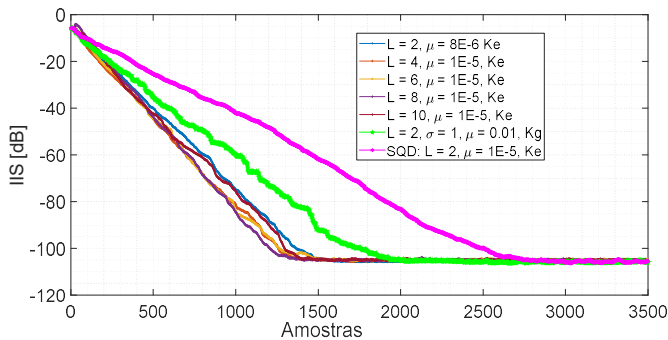


Fig. 5. Desempenho do algoritmo MSQD variando o L , para um sinal BPSK utilizando o canal $C_2(z) = 1 + 0.5z^{-1} + 0.1z^{-2}$, comparado com o desempenho do SQD para $L = 2$ da Fig. 4.

Na Fig. 5 podemos ver que o MSQD-Epa possui um desempenho superior ao MSQD-G independente do valor de L utilizado, convergindo mais rapidamente para o mesmo patamar de IIS. Além disso, também é possível perceber que o melhor caso do SQD-Epa, quando $L=2$, possui um desempenho inferior, em termos de velocidade de convergência, a todas as simulações do MSQD.

B. Sinais Correlacionados

Para o estudo dos algoritmos em um contexto com a transmissão de sinais correlacionados, utilizamos o diagrama de blocos mostrado na Fig. 6. Com o objetivo de controlar a correlação do sinal transmitido, filtramos o sinal BPSK por um pré-codificador, $f(n)$, a fim de gerar o sinal $s(n)$ correlacionado que, na sequência, será transmitido pelo canal. Assim, com o processo de equalização continuamos querendo recuperar o sinal $s(n)$.

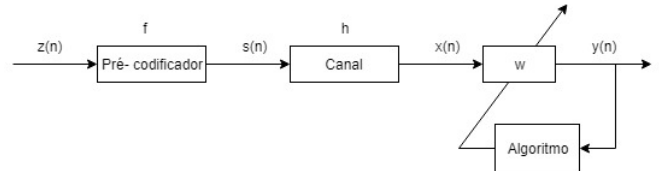


Fig. 6. Diagrama de blocos do sistema utilizando a pré-filtragem.

Os filtros utilizados como pré-codificadores foram $f_1(z) = 1 + 0.2z^{-1}$ e $f_2(z) = 1 + 0.5z^{-1} + 0.2z^{-2}$. É importante notar que o filtro $f_1(z)$ introduz correlação entre as amostras nos instantes n e $n-1$ de $s(n)$, enquanto que $f_2(z)$ introduz entre as amostras nos instantes n , $n-1$ e $n-2$. Para todas as simulações a seguir, foi utilizado o canal $C_2(z)$ e utilizamos o melhor valor de passo de adaptação encontrado. Para o caso em que utilizamos o pré-codificador f_1 , consideramos $L=2$ e para os casos com f_2 , utilizamos $L=4$.

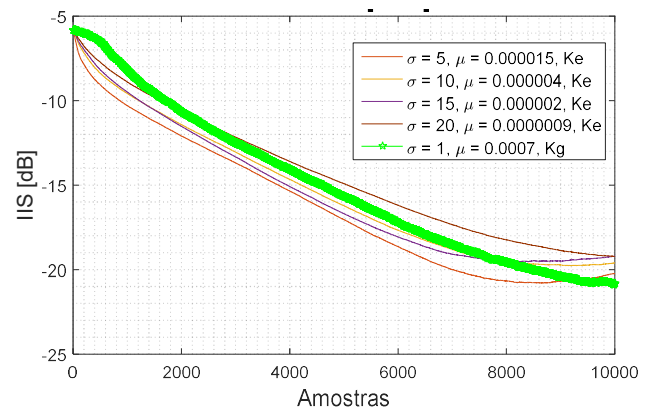


Fig. 7. Desempenho do algoritmo SQD para diversos valores de σ para um sinal BPSK pré-filtrado com $f_1 = 1 + 0.2z^{-1}$, utilizando canal o $C_2(z) = 1 + 0.5z^{-1} + 0.1z^{-2}$.

Considerando primeiramente o pré-codificador f_1 e o algoritmo SQD, a Fig. 7 mostra o resultado obtido considerando vários valores possíveis para σ no caso do kernel Epanechnikov. Podemos observar que as variações de σ não mudam muito o desempenho do algoritmo e praticamente temos o mesmo desempenho utilizando o SQD-G e o SQD-Epa. Este foi o único caso simulado no qual o SQD-Epa obteve um bom desempenho com valor baixo de σ . Aqui, $\sigma=5$ levou ao melhor desempenho, levando o algoritmo a convergir mais rápido do que o SQD-G, revertendo a tendência em obter melhores desempenhos com valores de σ mais altos apresentada na Fig.3.

A Fig. 8 abaixo ilustra o desempenho dos algoritmos com o uso do pré-codificador f_2 . Primeiramente, comparando com a Fig. 7, podemos ver que existe uma perda de desempenho neste caso. Na Fig. 7, com f_1 , havia sido possível obter IIS residuais abaixo de -20 dB. Neste caso, o menor valor

atingido fica em torno de -10dB. Assim, temos uma situação mais difícil de ser equalizada. Os melhores desempenhos obtidos pelo SQD-Epa foram com $\sigma=20$ e $\sigma=30$. Apesar do SQD-G apresentar uma oscilação maior, ele consegue atingir um patamar de IIS residual menor que o SQD-Epa.

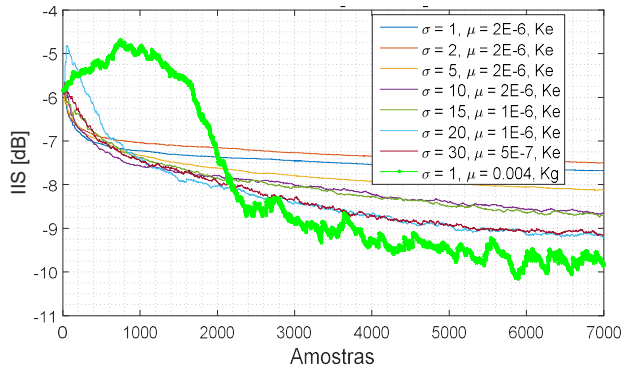


Fig. 8. Desempenho do algoritmo SQD para diversos valores de σ para um sinal BPSK pré-filtrado com $f_2 = 1 + 0.5z^{-1} + 0.2z^{-2}$ utilizando o canal $C_2(z) = 1 + 0.5z^{-1} + 0.1z^{-2}$.

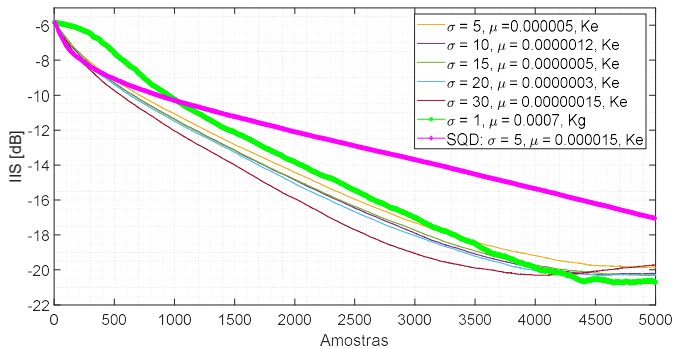


Fig. 9. Desempenho do algoritmo MSQD para diversos valores de σ para um sinal BPSK pré-filtrado $f_1 = 1 + 0.2z^{-1}$, utilizando o canal $C_2(z) = 1 + 0.5z^{-1} + 0.1z^{-2}$, comparado com o desempenho do SQD para $\sigma = 5$ da Fig. 7.

Considerando agora o MSQD, a Fig. 9 ilustra o desempenho dos algoritmos utilizando o pré-codificador f_1 . Podemos notar um comportamento bem semelhante ao obtido pelo SQD e mostrado na Fig. 7. A variação no valor de σ influi pouco no desempenho do algoritmo MSQD-Epa. Independentemente do valor de σ , o MSQD-Epa converge mais rapidamente do que o MSQD-G, sendo que o melhor desempenho é obtido com $\sigma=30$. Assim como no caso de sinais decorrelacionados (Fig. 5), aqui também podemos observar que o algoritmo MSQD possui um desempenho superior ao SQD, convergindo mais rapidamente.

Utilizando o pré-codificador f_2 , o resultado é mostrado na Fig. 10. Neste caso, independente do kernel utilizado, o MSQD não conseguiu atingir um bom desempenho. Ainda assim, o MSQD-Epa conseguiu convergir para um patamar de IIS residual cerca de 2 dB abaixo do atingido pelo MSQD-G.

V. CONCLUSÃO

Algoritmos de equalização não supervisionados baseados em ITL têm sido bastante utilizados para tratar situações nas quais as técnicas clássicas falham, como no caso da transmissão de sinais correlacionados. Neste trabalho, abordamos o algoritmo SQD, cujo critério busca igualar as funções densidade de probabilidade da saída do equalizador

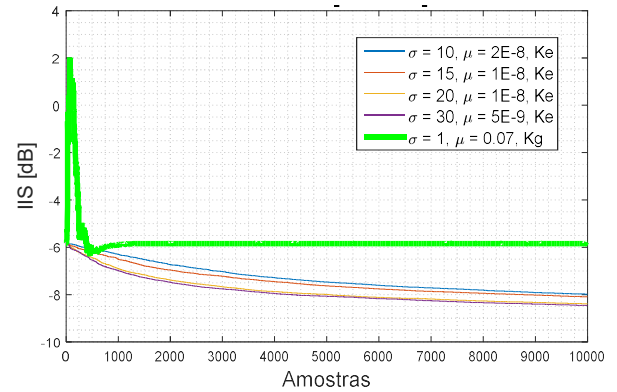


Fig. 10. Desempenho do algoritmo MSQD para diversos valores de σ para um sinal BPSK pré-filtrado $f_2 = 1 + 0.5z^{-1} + 0.2z^{-2}$, utilizando o canal $C_2(z) = 1 + 0.5z^{-1} + 0.1z^{-2}$.

com a do sinal transmitido. Para isso, é necessário se obter uma estimativa de tais pdfs em tempo real. Estimadores baseados em kernel têm sido muito utilizados com este fim, apresentando bons desempenhos. No entanto, dentro do contexto de equalização, a literatura apresenta somente o uso de kernels gaussianos. Como tal função não é ótima para a estimação de pdfs, neste trabalho, propomos o uso do kernel Epanechnikov. Deduzimos novamente os algoritmos SQD e MSQD a partir desta nova função kernel e mostramos, dentro de alguns contextos com sinais correlacionados e decorrelacionados, como o uso desta nova função pode melhorar o desempenho dos algoritmos.

AGRADECIMENTOS

Os autores agradecem à FAPESP pelo apoio financeiro.

REFERÊNCIAS

- [1] M. Lázaro, I. Santamaría and C. Pantaleón, “Matched PDF-Based Blind Equalization”, in IEEE International Conference on Acoustics, Speech and Signal Processing, 2003.
- [2] D. W. Scott, “Multivariate Density Estimation and Non Parametric Methods”. Willey, 2015.
- [3] E. Kraszczuk and A. Neves, “Análise de Critérios de Equalização Cega Baseados em Aprendizagem por Teoria da Informação” in XXXI Simpósio Brasileiro de Telecomunicações – SBTr2013, 2013.
- [4] E. T. S. Martins, “Análise do Desempenho de Algoritmos de Equalização Cega Baseados em ITL com Kernel Epanechnikov” in Dissertação apresentada ao Programa de Pós-graduação em Engenharia da Informação da Universidade Federal do ABC, 2018.
- [5] D. Fantinato; A. Neves; R. Attux. Analysis of a Novel Density Matching Criterion Within the ITL Framework for Blind Channel Equalization., Circuits Systems and Signal Processing, v. 37, p. 203-231, 2018.
- [6] M. Lázaro, I. Santamaría, D. Erdogmus, K. Hild, C. Pantaleón, J.C. Principe, Stochastic blind equalization based on PDF fitting using parzen estimator. IEEE Trans. Signal Process. 53(2), 696–704, 2005
- [7] J.C. Principe, Information Theoretic Learning Rényi’s Entropy and Kernel Perspectives, Springer, Berlin, 2010
- [8] M. Lázaro, I. Santamaría, C. Pantaleón, D. Erdogmus, K. Hild, J.C. Principe, Blind equalization by sampled PDF fitting, in Proceedings of the Fourth International Symposium Independent Component Analysis Blind Equalization, Nara, Japan, 2003, pp. 1041–1046