

Equalização Não-Linear usando *Single Channel ICA*

Gustavo Depieri e Aline Neves

Resumo— Este artigo propõe abordar a equalização de canais como um problema de separação cega de fontes através do *Single Channel ICA*. O problema é estendido para o caso não linear através do modelo *Post-Nonlinear* de misturas. Para a solução do problema, algoritmos bioinspirados são usados para recuperar a etapa não-linear da mistura, enquanto o *FastICA* é usado para separar a parte linear. As simulações computacionais avaliaram uma diversidade de cenários quanto ao tipo da fonte, canais e não linearidades, e os resultados atestaram o bom desempenho do algoritmo para estimativa da fonte e equalização do canal.

Palavras-Chave— Separação Cega de Fontes, Algoritmos Bioinspirados, SCICA PNL

Abstract— This paper proposes to model channel equalization as a blind source separation problem using the *Single Channel ICA*. The problem is extended to the nonlinear case using the *Post-Nonlinear Mixture Model*. To address it, we used bioinspired algorithms to recover the nonlinear stage of the mixture, while *FastICA* was used to separate the linear part. The computational simulations evaluated various scenarios regarding source type, channels, and nonlinearities, and the results attested to the good performance of the algorithm for source estimation and channel equalization.

Keywords— Blind Source Separation, Bioinspired algorithms, SCICA PNL

I. INTRODUÇÃO

A separação cega de fontes (BSS - *Blind Source Separation*) abarca um conjunto de técnicas que permitem a separação de misturas de sinais a partir de um conhecimento limitado das fontes e do processo misturador [1]. Dentre as técnicas de separação, a Análise de Componentes Independentes (ICA - do inglês *Independent Component Analysis*) baseia-se na premissa de independência estatística entre as fontes para estimar o separador [1].

A ICA é classicamente aplicada em modelos lineares nos quais o misturador pode ser representado por uma transformação linear invertível, sendo, portanto, os sinais observados gerados a partir de uma combinação linear das fontes. Nesse caso, a ICA estima uma matriz de separação das misturas. Já para misturas não-lineares gerais, não existe uma teoria consolidada que garanta a separação das fontes, uma vez que a flexibilidade do modelo tem potencial de produzir misturas que violem as condições de separabilidade da ICA. Por exemplo, é possível obter misturas de fontes independentes que também sejam independentes [1].

Uma extensão da ICA para misturas não-lineares pode ser feita a partir de uma restrição do modelo, dada pela introdução da não-linearidade invertível e sem memória posteriormente

ao misturador linear das fontes [2]. Tal caso de misturas em duas etapas é conhecido na literatura como misturas *Post-Nonlinear* (PNL). Para o sistema separador, primeiramente as não-linearidades são removidas das misturas pela estimativa de suas inversas e a mistura linear remanescente pode ser resolvida da mesma forma que no caso puramente linear [2].

Embora seja mais comum abordar a separação de misturas que envolvam diferentes fontes, também é possível aplicar as técnicas da ICA no caso particular de misturas (lineares ou PNL) com uma única fonte, denominado *Single Channel ICA* (SCICA) [3]. Esse caso surge no problema de deconvolução quando a fonte é filtrada por um sistema linear, o que introduz correlação ao sinal. Em seguida, uma não linearidade pode ser introduzida se o sistema for PNL. Um problema motivador para a aplicação desse modelo é o de equalização cega de canais lineares e não lineares, tendo em vista que as técnicas de BSS têm sido estudadas em aplicações de comunicações sem fio [4].

Na literatura, encontram-se trabalhos que modelam as não-linearidades por expansão em série de Taylor e redes neurais [5], sendo que a otimização dos parâmetros lineares e não-lineares é feita simultaneamente pelo método do gradiente para minimizar a função custo da informação mútua. Nesse artigo, por sua vez, será aplicado o algoritmo bioinspirado híbrido para uma única fonte para estimativa da parte não-linear da mistura, enquanto a matriz de separação será estimada pelo algoritmo *FastICA*. Este consiste na hibridização dos algoritmos de Otimização por Enxame de Partículas (OEP) com o algoritmo de Evolução Diferencial (ED).

O artigo está organizado da seguinte maneira: o problema do SCICA PNL é apresentado na Seção II. Os algoritmos bioinspirados OEP, ED e a sua versão híbrida aplicada ao SCICA PNL são explicados na Seção III. As Seções IV e V apresentam os cenários das simulações e os resultados obtidos. A conclusão do artigo é feita na Seção VI.

II. SINGLE CHANNEL ICA - PNL

No SCICA PNL é observado um único sinal de mistura, a partir do qual deve-se encontrar o sistema separador composto por uma não-linearidade e um filtro equalizador para recuperação da fonte [5].

As condições de separabilidade do SCICA são herdadas dos métodos clássicos da BSS baseados na ICA [2]. A única fonte na mistura deve ser um processo não gaussiano com amostras identicamente distribuídas e independentes (*i.i.d.*), denotado por $s[n]$ [5]. Caso a fonte seja um processo autoregressivo conhecido, é possível recuperá-la após a filtragem do processo *i.i.d.* na saída do SCICA [5].

A Fig. 1 mostra o esquemático de um misturador PNL. A etapa linear do misturador é representada por um filtro linear

C,

$$x = C * s, \quad (1)$$

onde s é o sinal transmitido e $*$ denota a operação de convolução, enquanto a não linear é uma função invertível e instantânea f aplicada posteriormente ao sinal x ,

$$z = f(x), \quad (2)$$

sendo z o sinal observado na saída do misturador.

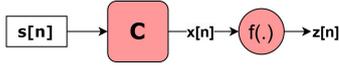


Fig. 1. Sistema misturador SCICA PNL. Fonte: Autoria própria.

O sistema separador deve estimar os parâmetros não-lineares e lineares apropriados para recuperar s dado que z foi observado. Para tanto, considera-se o sinal da mistura com N_B atrasos, para o qual deseja-se recuperar a independência estatística, $\mathbf{z} = (z[n], z[n-1], \dots, z[n-N_B+1])^T$, conforme mostra a Fig. 2, sendo $g \approx f^{-1}$ uma estimativa para a inversa

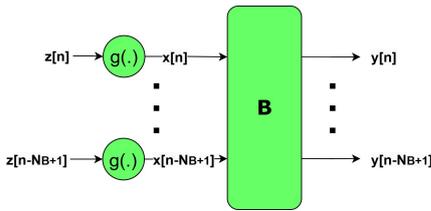


Fig. 2. Sistema separador SCICA PNL. Fonte: Autoria própria.

da função não-linear que o misturador introduziu no sinal e \mathbf{B} corresponde à matriz de separação das misturas lineares (pós-remoção da não-linearidade). A matriz \mathbf{B} recupera a fonte com diferentes atrasos, e apresenta, em cada linha, os coeficientes do filtro FIR que equaliza a etapa linear do sistema.

A. Sistema separador SCICA PNL

O sistema separador compreende um processo de duas etapas [2]. A etapa não-linear do separador estima a função g a partir da expansão em série de Taylor (3), com potências pares e ímpares até a ordem L ,

$$g(z_i) = \sum_{l=1}^L c_l z_i^l, \quad \text{com } i = 0, 1, \dots, N_B - 1, \quad (3)$$

em que $z_i = z[n-i]$ é o i -ésimo atraso do sinal observado. O conjunto de parâmetros $\mathbf{c} = (c_1, c_2, \dots, c_L)^T$, que corresponde aos coeficientes da série (3), é otimizado pelos algoritmos bioinspirados, conforme será detalhado na próxima seção. Aqui é feita uma modificação do algoritmo apresentado em [6] para incorporar termos de ordem par que estimam uma única g (3). A adição desses termos permite a modelagem de não-linearidades mais gerais, isto é, ainda invertíveis, porém sem a restrição de serem funções com simetria ímpar.

Como requisito de separação, durante o processo de otimização, deve-se atentar para soluções que sejam invertíveis; para tanto, será considerada a monotonicidade da função g a

partir da avaliação do sinal de sua derivada g' no intervalo das amostras de $z[n-i]$, isto é, no intervalo $z \leq \text{máx.}(|z_i|)$. A função g será considerada invertível se for monotonicamente crescente ou decrescente.

Seguindo a estimativa de g , a matriz de separação \mathbf{B} da etapa linear será obtida através do algoritmo clássico FastICA [1]. Esse algoritmo é baseado no método do gradiente de ponto fixo e maximiza a função custo da negentropia como medida da não-gaussianidade (medida de independência estatística das fontes) [1]. O número de atrasos N_B considerados determina as dimensões da matriz \mathbf{B} , o que, de forma equivalente, corresponde ao filtro equalizador com N_B taps.

Após as duas etapas supracitadas, pode-se representar a estimativa da fonte a partir de (4)

$$\mathbf{y} = \mathbf{B}g(\mathbf{z}) \approx \mathbf{s}, \quad (4)$$

sendo o vetor que contém a estimativa da fonte e seus atrasos dado por $\mathbf{y} = (y[n], y[n-1], \dots, y[n-N_B+1])^T$. Nesse momento, cabe pontuar que o SCICA carrega as mesmas ambiguidades dos métodos de BSS baseados na ICA quanto à amplitude, fase e ordem das versões atrasadas da fonte que são recuperadas [1]. Em seguida, será apresentada a função custo para o processo de otimização dos parâmetros do separador.

B. Função custo dada pela Informação Mútua

Conforme apresentado, a estimativa da fonte é obtida pela atuação conjunta dos algoritmos bioinspirados que otimizam os parâmetros da etapa não-linear (coeficientes da série de Taylor) e do FastICA que determina a matriz de separação. A função custo de todo esse processo é dada pela minimização da informação mútua (5), que pode ser tomada como uma medida da independência estatística entre os diferentes atrasos da fonte (processo *i.i.d.*),

$$I(\mathbf{y}) = \sum_{i=1}^n H(y_i) - H(\mathbf{z}) - E\left\{\sum_{i=1}^n \log(|g'_i(z_i)|)\right\} - \log |\det \mathbf{B}|, \quad (5)$$

sendo que o termo $H(\mathbf{z})$ é constante para os parâmetros do modelo e pode ser desconsiderado na otimização. A estimativa das entropias marginais $H(y_i)$ é feita pela expansão de Gram-Charlier, assim como em [6].

Nesse artigo também avaliaremos a estimativa da entropia marginal a partir do método KDE (*Kernel Density Estimation*) com o kernel Epanechnikov [7] $K_{epa.}(y; \sigma_e) = \frac{3}{4\sigma_e} (1 - \frac{y^2}{\sigma_e^2})$ se $-\sigma_e < y < \sigma_e$, e 0 fora desse intervalo, em uma abordagem similar para estimar a função custo que foi feita em [7]. A entropia marginal é calculada para N_w janelas de tamanho $W = 100$ amostras do sinal e com uma largura do kernel $\sigma_e = 0.3$, conforme (6),

$$H(y_i) \approx \frac{1}{N_w} \sum_{k=0}^{N_w-1} \hat{H}_k(\hat{p}_k(y_i)), \quad (6)$$

em que $\hat{p}_k(y_i[n]) = \frac{1}{W} \sum_{w=kW}^{(k+1)W-1} K_{epa.}(y_i[n] - y_i[w]; \sigma_e)$ é a estimativa da probabilidade da n -ésima amostra calculada na k -ésima janela, para $0 \leq k \leq N_w - 1$ (janelas) e $kW \leq n \leq (k+1)W - 1$ (amostras), e $\hat{H}_k(\hat{p}_k(y_i)) = \frac{1}{W} \sum_{w=kW}^{(k+1)W-1} -\log_2(\hat{p}_k(y_i[w]))$ corresponde à entropia marginal estimada de y_i na k -ésima janela.

III. ALGORITMOS BIOINSPIRADOS

Os algoritmos bioinspirados são algoritmos de busca heurísticos inspirados na natureza. Diante do problema de SCICA PNL, embora alguns trabalhos utilizem a otimização por gradiente, temos que a escolha desses algoritmos é mais adequada uma vez que a função custo (5) é multimodal e as soluções para o problema devem respeitar o critério de invertibilidade de g [2]. Esse trabalho utiliza a versão híbrida da Otimização por Enxame de Partículas (OEP) com a Evolução Diferencial (ED) em um algoritmo denominado OEP-ED-UF. Tal algoritmo segue a configuração apresentada em [6], mas é adaptado para uma única mistura do SCICA PNL.

O OEP é um algoritmo de busca inspirado na movimentação conjunta de pássaros em um enxame, que se influenciam tanto por decisões tomadas localmente quanto pelo grupo [8]. A interação entre as partículas se dá de acordo com a topologia do enxame, que define a vizinhança entre as partículas, isto é, com quais outras partículas cada uma pode trocar informações.

No problema do SCICA em questão, as coordenadas do vetor posição \mathbf{p} de cada uma das partículas correspondem aos coeficientes de otimização de (3), o que resulta em um espaço de busca em \mathbb{R}^L . As partículas percorrem o espaço de busca de maneira a minimizar a função custo do problema dada por (5).

No OEP, um conjunto de partículas é inicializado aleatoriamente no espaço de busca, tendo um vetor posição e um vetor velocidade associados [8]. A cada nova iteração $k + 1$ do algoritmo, os vetores velocidade $\mathbf{v}^{(k+1)}$ são atualizados conforme (7),

$$\mathbf{v}^{(k+1)} \leftarrow w^{(k+1)}\mathbf{v}^{(k)} + \alpha_1^{(k+1)}\Delta\mathbf{c}_{\text{cog}}^{(k)} + \alpha_2^{(k+1)}\Delta\mathbf{c}_{\text{soc}}^{(k)}, \quad (7)$$

e seus vetores posições $\mathbf{p}^{(k+1)}$ conforme (8),

$$\mathbf{p}^{(k+1)} \leftarrow \mathbf{p}^{(k)} + \mathbf{v}^{(k+1)}, \quad (8)$$

em que os fatores w , α_1 , α_2 correspondem, respectivamente, à inércia do movimento e aos coeficientes de aceleração ao atrator cognitivo \mathbf{c}_{cog} e ao social \mathbf{c}_{soc} . Os valores e as equações escolhidas para atualização de tais fatores são iguais às apresentadas em [6].

O termo $\Delta\mathbf{p}_{\text{cog}}^{(k)} = \mathbf{p}_{\text{cog}}^{(k)} - \mathbf{p}^{(k)}$ é o vetor de deslocamento da posição atual da partícula até o seu atrator cognitivo $\mathbf{p}_{\text{cog}}^{(k)}$, que corresponde à melhor posição que a partícula já visitou até a iteração k . Esse atrator pode ser atualizado ($\mathbf{p}_{\text{cog}}^{(k+1)} \leftarrow \mathbf{p}^{(k+1)}$) caso a nova posição apresente um valor de *fitness* maior e seja uma solução invertível. Já $\Delta\mathbf{p}_{\text{soc}}^{(k)}$ é o vetor de deslocamento em relação ao atrator social percebido por cada partícula, conforme a topologia do enxame [8]. O atrator social para cada partícula é definido como o atrator cognitivo com melhor *fitness* dentre os de sua vizinhança [8]. A topologia quadrada foi escolhida para o enxame, conforme os resultados apresentados em [6].

O algoritmo de Evolução Diferencial é um algoritmo de otimização inspirado nos princípios biológicos e aleatórios descritos na Teoria da Evolução [9]. No SCICA PNL, o ED atuará conjuntamente ao OEP na versão híbrida do algoritmo para otimizar os parâmetros em 3. As soluções candidatas são cromossomos com L genes, representados por vetores em \mathbb{R}^L .

O algoritmo ED produz novas soluções a partir da combinação pela diferença entre cromossomos na população [9]. Para cada cromossomo \mathbf{C}_{r_1} , outros dois (\mathbf{C}_{r_2} e \mathbf{C}_{r_3}) são sorteados para produzir um novo cromossomo \mathbf{C}_{r_n} , conforme (9),

$$\mathbf{C}_{r_n}^{(k+1)} \leftarrow \mathbf{C}_{r_1}^{(k)} + u_1(\mathbf{C}_{r_2}^{(k)} - \mathbf{C}_{r_1}^{(k)}) + u_2(\mathbf{C}_{r_3}^{(k)} - \mathbf{C}_{r_1}^{(k)}), \quad (9)$$

sendo u_1 e u_2 números aleatórios uniformemente distribuídos entre 0.75 e 1.00. Para incorporar a *crossover*, alguns genes de \mathbf{C}_{r_n} são aleatoriamente substituídos por seus respectivos em \mathbf{C}_{r_1} . A taxa de *crossover* adotada foi de 0.5. Caso o *fitness* de \mathbf{C}_{r_n} seja melhor do que o de \mathbf{C}_{r_1} , ele o substituirá na população; caso contrário, \mathbf{C}_{r_n} será descartado.

A. Algoritmo Híbrido OEP-ED-UF

O OEP-ED-UF utiliza o OEP, como algoritmo principal, em conjunto com a ED para estimar os parâmetros definidos em (3). Cada iteração do OEP consiste no deslocamento das partículas no espaço de busca, avaliação do *fitness* conforme (5), e da atualização dos atratores (se necessário) cognitivos e sociais [6]. A cada três iterações do OEP, o ED atua nas partículas do enxame que apresentam os atratores cognitivos com os piores valores de *fitness*. Uma nova solução candidata para o atrator cognitivo é produzida conforme o procedimento do algoritmo ED, podendo substituir o atual caso apresente uma melhoria relativa no *fitness* e seja invertível. O número de partículas que passam pelo ED corresponde a 1/4 do tamanho do enxame. O algoritmo interrompe a execução caso atinja o número limite de 300 iterações ou não verifique uma melhoria no *fitness* da melhor solução encontrada pelo enxame por 30 iterações seguidas. Tais parâmetros do algoritmo foram definidos por simulação e inspirados no trabalho [6].

IV. SIMULAÇÕES

As fontes escolhidas para simulação são processos *i.i.d* sendo $s_1 \sim \text{Unif.}(-1, 1)$ (uniformemente distribuídas no intervalo dado), e duas fontes amostradas de distribuições discretas e equiprováveis: $s_2[n] \in \{-1.0, -0.5, 0, +0.5, +1.0\}$, e $s_3[n] \in \{-1, +1\}$.

Para a avaliação dos resultados de simulação do OEP-ED-UF utilizaremos duas métricas principais: a SIR (*Signal to Interference Ratio*) (10) que é uma medida em dB da potência da fonte estimada e do sinal de erro em relação à fonte [5],

$$\mathbf{SIR} = 10 \log(E\{\mathbf{y}^2\}/E\{(\mathbf{y} - \mathbf{s})^2\}) \text{ [dB]}, \quad (10)$$

e a ISI (*Intersymbol Interference*) (12), que representa o erro residual de equalização. O vetor da convolução do canal com o filtro estimado é dado por

$$\mathbf{\Lambda}_i := \mathbf{C} * \mathbf{b}_i, \quad (11)$$

em que \mathbf{b}_i é a i -ésima linha da matriz de separação \mathbf{B} . A ISI pode ser calculada como,

$$\text{ISI}_i = 10 \log\left(\frac{\sum_{j=1}^M |\mathbf{\Lambda}_i(j)| - |\mathbf{\Lambda}_i|_{\text{máx}}}{|\mathbf{\Lambda}_i|_{\text{máx}}}\right) \text{ [dB]}, \quad (12)$$

em que M é o número de elementos de $\mathbf{\Lambda}_i$, $|\mathbf{\Lambda}_i|_{\text{máx}}$ representa o valor absoluto de seu maior elemento, e $|\mathbf{\Lambda}_i(j)|$ é o valor absoluto de seu j -ésimo elemento.

Para analisar os resultados da SIR, serão feitas as devidas correções das ambiguidades, anteriormente comentadas, de y e antes de comparar e calcular o erro com a fonte. Para a SIR, será considerado o valor máximo obtido dentre os diferentes atrasos, enquanto que para a ISI será considerado seu menor valor dentre os diferentes filtros equalizadores estimados pelo algoritmo.

Para a etapa linear das misturas serão explorados 3 canais com diferentes características de fase. A Fig. 3 apresenta a posição dos zeros no plano complexo, sendo¹: $C_1(Z) = 1 - 0.8Z^{-1} + 0.6Z^{-2} - 0.4Z^{-3}$ de fase mínima, $C_2(Z) = -0.3 + 0.5Z^{-1} - 0.7Z^{-2} + Z^{-3}$ de fase máxima, e o canal de fase não-mínima $C_3(Z) = 0.25 - 0.55Z^{-2} + Z^{-4} - 0.55Z^{-6} + 0.25Z^{-8}$.

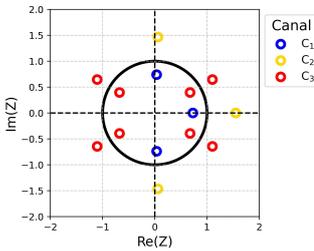


Fig. 3. Posição dos zeros dos canais na circunferência de raio unitário.

Na etapa não-linear da mistura PNL serão aplicadas as três funções da Fig. 4: a função f_1 , definida por intervalos,

$$f_1(ax) = \begin{cases} 0.5ax, & x \leq 0 \\ 1.5ax, & x > 0, \end{cases} \quad (13)$$

e as funções invertíveis com simetria ímpar $f_2(ax) = \text{sign}(ax)\sqrt{|a|x|}$ e $f_3(ax) = \tanh(5ax) + 0.1ax$, sendo esta última avaliada em [5]. Em todos os casos, o fator $a := \frac{2}{\max(|x|)}$ normaliza o sinal a partir de seu valor máximo absoluto. Dessa forma, é possível representar as não-linearidades em um mesmo intervalo, no caso escolhido entre $-2 \leq ax \leq +2$, independentemente do tipo da fonte ou do canal da mistura linear.

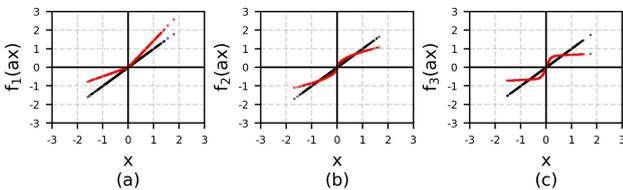


Fig. 4. Em preto, uma reta unitária para comparação e, em vermelho, as funções a) f_1 b) f_2 c) f_3 .

V. RESULTADOS OBTIDOS E ANÁLISE

O primeiro grupo de simulações foi realizado com a fonte s_2 com 1350 amostras, o canal de fase mínima $C_1(Z)$ e a não-linearidade f_1 . Nele foram avaliados o enxame com 49 partículas, tamanho do filtro $N_B = 15$, e $L = 5$ termos para

estimativa de g , dada por (3), e os dois métodos para estimar as entropias marginais dados pela expansão de Gram-Charlier e o KDE. Foram 10 repetições do experimento para a obtenção das estatísticas. Os resultados obtidos foram melhores para a expansão de Gram-Charlier, apresentando uma SIR média de 15.5 ± 1.8 e uma ISI de -22.7 ± 1.0 dB, contra os respectivos 10.4 ± 3.3 e -16.3 ± 4.6 dB do algoritmo com o KDE. Tendo em vista que o custo computacional desse último é mais elevado e os resultados foram piores, as demais simulações foram feitas considerando apenas o OEP-ED-UF com a expansão de Gram-Charlier no cálculo do *fitness* (5).

Mantendo o mesmo canal $C_1(Z)$ e a não-linearidade f_1 , foram simulados enxames com 25 e 49 partículas, tamanhos do filtro de $N_B = 5$ e $N_B = 15$ taps, e a expansão de g , dada por (3), com $L = 3$ e $L = 5$ termos. Ao todo, foram 8 cenários de combinações desses parâmetros para os quais foram simuladas 10 repetições. A Tabela I apresenta os valores médios e o intervalo de confiança para a SIR e a ISI.

TABELA I

VALORES MÉDIOS DA SIR E DA ISI COM SEUS RESPECTIVOS INTERVALOS DE CONFIANÇA (95%).

N_B	L	Nº Partículas	SIR	ISI
15	3	25	14.6 ± 0.4	-21.8 ± 1.0
15	3	49	14.2 ± 0.4	-21.1 ± 1.0
15	5	25	13.0 ± 4.0	-18.9 ± 5.3
15	5	49	15.5 ± 1.8	-22.7 ± 1.0
5	3	25	10.7 ± 0.3	-12.8 ± 0.2
5	3	49	10.2 ± 0.4	-12.5 ± 0.3
5	5	25	8.1 ± 2.4	-10.3 ± 3.0
5	5	49	10.4 ± 0.5	-12.5 ± 0.5

As 40 simulações realizadas com $N_B = 15$ apresentam um valor agregado de SIR e ISI iguais a 14.3 ± 1.1 e -21.1 ± 1.4 dB, respectivamente, enquanto que para $N_B = 5$ os resultados foram piores, e com significância estatística na diferença de ambas as medidas, sendo 9.8 ± 0.7 e -12.0 ± 0.8 dB.

Para os casos com $N_B = 15$ e analisando os valores médios, o melhor resultado foi obtido para $L = 5$ e 49 partículas. Para enxames com 25 partículas, um número menor de parâmetros $L = 3$ é mais adequado para estimar as não linearidades.

A Fig. 5 mostra gráficos de dispersão dos sinais em relação à sua versão atrasada de uma amostra para um caso simulado com os parâmetros: $N_B = 15$, $L = 5$ e 49 partículas, onde se observou uma SIR máxima de 17.1 dB. A partir da Fig. 5a pode-se inferir que a fonte é *i.i.d.*; na Fig. 5b é apresentada a mistura linear x na qual percebe-se a correlação introduzida pelo canal; na Fig. 5c verifica-se uma distorção não-linear da mistura dos sinais z , e por fim, na Fig. 5d temos a fonte estimada pelo algoritmo cujo padrão das amostras se assemelha à da fonte (original) onde se têm a recuperação da independência estatística do processo.

No segundo conjunto de simulações, foram consideradas as fontes s_1 e s_3 , os canais $C_1(Z)$, $C_2(Z)$ e $C_3(Z)$, que possuem diferentes características de fase, e a não-linearidade f_2 . Todos os 6 casos foram simulados com 10 repetições, $N_B = 15$, $L = 3$ e com enxame de 49 partículas.

Dos valores da SIR e da ISI, temos que o algoritmo conseguiu recuperar a fonte das misturas de maneira satisfatória,

¹Aqui a notação Z (maiúscula) representa a transformada Z dos canais.

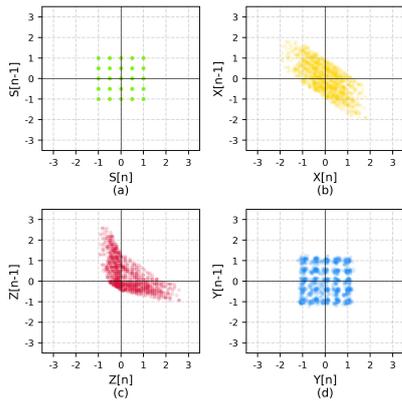


Fig. 5. Sinais e as suas versões atrasadas de uma amostra a) fonte b) mistura linear c) mistura PNL d) fonte estimada.

tendo maior facilidade com a fonte discreta s_3 para a qual obteve uma SIR média de 21.4 ± 1.0 contra 16.4 ± 1.6 dB do obtido para a s_1 . Os resultados agregados por canal estão apresentados na Tabela II, na qual se observa um melhor desempenho do algoritmo para os canais de fase mínima e máxima. A última coluna da Tabela II mostra o valor médio dos atrasos que, com base na SIR, foram melhor recuperados pelo algoritmo e que refletem as características de fase dos canais, lembrando que o filtro estimado apresenta 15 *taps*. Seus valores médios apresentam uma pequena divergência dos atrasos ótimos da teoria de filtragem adaptativa [1], o que pode justificar-se pelas misturas não serem lineares e o algoritmo recuperar mais de uma versão atrasada, próximas ao atraso ótimo, sem diferença significativa da SIR.

TABELA II
VALORES MÉDIOS DA SIR, DA ISI E DO MELHOR ATRASO COM SEUS RESPECTIVOS INTERVALOS DE CONFIANÇA (95%).

Canal	SIR	ISI	Atraso
$C_1(Z)$	21.0 ± 1.2	-24.1 ± 1.1	1.7 ± 0.8
$C_2(Z)$	18.9 ± 2.9	-21.8 ± 2.9	12.6 ± 0.7
$C_3(Z)$	16.8 ± 0.5	-20.0 ± 0.6	8.2 ± 0.6

Para atestar a utilidade do OEP-ED-UF nas misturas consideradas nesse trabalho, a Fig. 6 apresenta uma análise comparativa da SIR e ISI obtidas para os sinais estimados pelo algoritmo em todas as 60 simulações em relação aos valores dessas métricas que são obtidos ao se aplicar diretamente o FastICA na mistura não-linear. Como pode-se verificar na Fig. 6a para os valores da SIR e na Fig. 6b para a ISI, os resultados do OEP-ED-UF são significativamente superiores tanto na qualidade de recuperação da fonte das misturas como para a equalização dos canais. Dessa análise conclui-se que as misturas não são triviais para que se possa desprezar a não-linearidade.

No último conjunto de simulação foi considerado o canal de fase não-mínima $C_3(Z)$ com a não-linearidade f_3 . A fonte escolhida foi a s_1 , com $L = 9$ e $N_B = 15$, já o exame se deu com 49 partículas e foram realizadas 5 repetições. Obtivemos os valores de SIR e da ISI de 10.89 ± 2.90 dB e -14.79 ± 2.19 dB, respectivamente. Dessas, conclui-se que o algoritmo

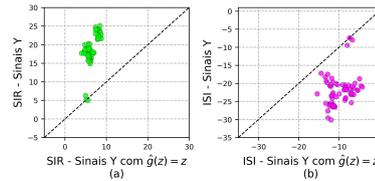


Fig. 6. Resultados comparativos entre os sinais estimados pelo OEP-ED-UF e ao se aplicar o FastICA diretamente na mistura não-linear para a) SIR b) ISI.

também foi capaz de recuperar a fonte quando submetida a uma não-linearidade mais forte e a um canal de fase não-mínima.

VI. CONCLUSÕES

Nesse trabalho, o algoritmo bioinspirado híbrido OEP-ED-UF foi aplicado ao problema de SCICA PNL e apresentou bons resultados em termos da SIR e da ISI nos diferentes cenários de não linearidades, canais e tipos de fonte. Mostrou-se também a capacidade do algoritmo de inverter não linearidades que não apresentam simetria ímpar, que são as mais comuns de serem abordadas na literatura. A avaliação da entropia marginal pelo método do KDE mostrou-se inferior à expansão de Gram-Charlier nos casos estudados. Contudo, a metodologia proposta, apesar de ter sido investigada com a motivação do problema de equalização não linear, apresenta elevado custo computacional para ser implementada em um sistema de telecomunicações em tempo real.

AGRADECIMENTOS

O presente trabalho foi realizado com apoio da CAPES, código de financiamento 001, e também da Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP) - projeto BIOS número 2020/09838-0.

REFERÊNCIAS

- [1] A. Hyvärinen and E. Oja. Independent component analysis: algorithms and applications. *Neural networks*, 13(4):411–430, 2000.
- [2] Yannick Deville and Leonardo Tomazeli Duarte. An overview of blind source separation methods for linear-quadratic and post-nonlinear mixtures. In *International Conference on Latent Variable Analysis and Signal Separation*, pages 155–167. Springer, 2015.
- [3] M.E. Davies and C.J. James. Source separation using single channel ica. *Signal Processing*, 87(8):1819–1832, 2007.
- [4] Luo Zhi-Quan, Li Chengjie, and Zhu Long. A comprehensive survey on blind source separation for wireless adaptive processing: Principles, perspectives, challenges and new research directions. *IEEE Access*, 6:66685–66708, 2018.
- [5] Jordi Sole, Christian Jutten, and Anisse Taleb. Parametric approach to blind deconvolution of nonlinear channels. *Neurocomputing*, 48(1-4):339–355, 2002.
- [6] Depieri F. Gustavo and Aline Neves. Algoritmo bioinspirado híbrido aplicado em separação cega de fontes no contexto post-nonlinear. *XLIII Simpósio Brasileiro de Telecomunicações e Processamento de Sinais*, 2024.
- [7] Aline Neves Caroline P. A. Moraes, Denis G. Fantinato. Epanechnikov kernel for pdf estimation applied to equalization and blind source separation. *Elsevier ScienceDirect Journals*, 189:108251–108251, 2021.
- [8] Saptarshi Sengupta, Sanchita Basak, and Richard Alan Peters. Particle swarm optimization: A survey of historical and recent developments with hybridization perspectives. *arXiv.org*, 2018.
- [9] Yujun Zheng, Xueqin Lu, Minxia Zhang, and Shengyong Chen. *Biogeography-Based Optimization: Algorithms and Applications*. Springer, 01 2019.