

Uma Melhoria na Convergência do Método ICA-MAP para Remoção de Ruído em Sinal de Voz

Fabio Louvatti do Carmo e Evandro Ottoni Teatini Salles

Resumo—Esse artigo apresenta uma modificação feita no algoritmo ICA-MAP a fim de melhorar sua convergência. Foi observado, através de testes, que é possível limitar a magnitude do vetor gradiente, usado para estimar os parâmetros do modelo de denoising, e assim melhorar a estabilidade do algoritmo. Tal adaptação pode ser entendida como uma restrição no problema de otimização original.

Palavras-Chave—Melhoramento de Voz, *Maximum a posteriori* (MAP), Análise das Componentes Independentes (ICA), Ruído Não-Gaussiano, Otimização.

Abstract—This paper presents a modification to the ICA-MAP algorithm in order to improve its convergence. It was observed through testing that it is possible to limit the magnitude of the gradient vector, used to estimate the denoising model, and thereby improve the stability of algorithm. Such adaptation can be understood as a restriction the original optimization problem.

Keywords—Speech Enhancement, *Maximum a posteriori* MAP, Independent Component Analysis (ICA), Non-Gaussian Noise, Optimization.

I. INTRODUÇÃO

Algoritmos de *denoising* em voz se preocupam em reduzir o nível do fundo ruidoso em sinais de voz capturados por um microfone a fim de melhorar a qualidade de sistemas de comunicações de voz.

Em especial, em [1], modificada em [2], é apresentada a técnica denominada Subtração Espectral, muito utilizada para tratar sinal de voz contaminado com ruído aditivo. Subtração Espectral utiliza amplitude espectral de tempo curto (STSA) e estima o sinal de voz limpo a partir da subtração do espectro de magnitude do ruído, mantendo a fase do sinal corrompido. Em [3] utiliza-se STSA e filtragem de Wiener de modo a minimizar o erro quadrático médio de cada componente espectral, assumindo-se que tanto o sinal limpo como o ruído tenham distribuição gaussiana.

Em [4],[5] empregou-se análise de componentes independentes (ICA - *Independent Component Analysis*) para separação de fontes, no caso sinal de voz e ruído, cuja estimação foi realizada pelo uso do algoritmo MAP (*maximum a posteriori*) em um domínio de transformação linear (ICA linear). Outros estudos sugerem o uso de DCT (Transformada Discreta de Cosseno) e KLT (Transformada de Karhunen-Loève) [6], transformada wavelet [7], além do uso de ICA [5] [8], também empregando MAP e assumindo distribuição não gaussiana para sinais de voz. Observou-se que esta modelagem

não gaussiana leva para uma estimativa não-linear, proporcionando bons resultados em muitos casos, sendo comumente utilizada a distribuição Laplaciana para voz e gaussiana para o ruído, conforme mostrado em [6]. Em [4] [5] [8], os sinais foram modelados por uma combinação de distribuições levemente super-gaussiana, Laplaciana e fortemente super-gaussiana, e o algoritmo obtido é referenciado como *Sparse Code Shrinkage* (SCS). Em [9] é mostrado que, utilizando-se o algoritmo MAP, podem ser empregadas transformações ICA individuais para o sinal limpo e ruído, e assim obter melhores estimativas do sinal para ruídos não gaussianos. Tal algoritmo é conhecido como ICA-MAP.

Nesse artigo, será mostrada uma modificação no algoritmo ICA-MAP, proposta em [9]. Foi notado que existem algumas dificuldades na implementação deste algoritmo devido a algumas particularidades da função densidade de probabilidade (*pdf*) esparsa do sinal limpo e ruído. A fim de melhorar a convergência do algoritmo, sugere-se uma modificação no cálculo do vetor gradiente usado na estimação do sinal melhorado originalmente proposta em [9].

Este artigo é organizado como segue. O algoritmo ICA-MAP [9] é mostrado na Seção II. A Seção III descreve a modificação proposta. Os testes e resultados são apresentados na Seção IV. E a Seção V dá as conclusões.

II. A ABORDAGEM ICA-MAP

A. Estimativa MAP

Considerando variáveis aleatórias escalares, x representa aqui o sinal limpo e v a parcela de ruído aditivo. Assume-se que o sinal observado y é uma versão ruidosa de x segundo

$$y = x + v. \quad (1)$$

O interesse é estimar x usando apenas o sinal observado y por meio de $\hat{x} = g(y)$. Para isso, utiliza-se o algoritmo de estimação MAP, que diferentemente da estimação ML (*Maximum Likelihood*), leva em consideração a distribuição condicional e a distribuição *a priori*. E, utilizando o logaritmo da distribuição, o algoritmo resulta em

$$\hat{x} \leftarrow \underset{x}{\operatorname{argmax}}(\ln p(y | x) + \ln p(x)), \quad (2)$$

onde o termo $p(y | x)$ é a distribuição condicional de y dado x , que é a probabilidade do ruído avaliada em $y - x$, ou seja, $p(y | x) = p_v(y - x)$, e $p(x)$ é a distribuição *a priori*, aqui denominada de $p_x(x)$.

Tomando-se o sinal de áudio $y(t)$ em pequenos segmentos (*frames*), cada um composto por N amostras, é possível escrever agora o seguinte vetor de observações

$$\mathbf{y} = \{\mathbf{y}(1), \mathbf{y}(2), \dots, \mathbf{y}(M - 1), \mathbf{y}(M)\},$$

onde $\mathbf{y}(t) = [y(t), \dots, y(t - N + 1)]^T$ e $(\cdot)^T$ significa o operador de transposição de matriz. Usando a mesma notação para o sinal limpo, a estimativa de $\hat{\mathbf{x}}$ para $M \rightarrow \infty$, de acordo com (2), é

$$\begin{aligned} & \{\hat{\mathbf{x}}(1), \dots, \hat{\mathbf{x}}(M)\} \leftarrow \\ & \leftarrow \operatorname{argmax}_{\hat{\mathbf{x}}} \left(\sum_t \ln p_v(\mathbf{y}(t) - \hat{\mathbf{x}}(t)) + \sum_t \ln p_x(\hat{\mathbf{x}}(t)) \right) \quad (3) \\ & \Leftrightarrow \operatorname{argmax}_{\hat{\mathbf{x}}} (E\{\ln p_v(\mathbf{y}(t) - \hat{\mathbf{x}}(t))\} + E\{\ln p_x(\hat{\mathbf{x}}(t))\}). \end{aligned}$$

Daqui para frente será omitido o índice t e se substituirá $\hat{\mathbf{x}}$ por \mathbf{x} .

Em [10] é mostrado que a maximização em (3) pode ser alcançada usando as componentes independentes através de uma abordagem ICA linear. Através do algoritmo ICA, é obtida uma matriz W , comumente chamada de matriz de desmistura, que leva as componentes do sinal para um espaço onde elas se tornam tão independentes quanto possível. Para realizar a estimação MAP utilizando essa transformação, usa-se a relação entre a função densidade de uma variável aleatória a e de uma versão transformada linearmente $b = Wa$, que é $p_a(\mathbf{a}) = p_b(W\mathbf{a})|\det(W)|$.

Considerando a estimativa de um único *frame*, o operador expectância pode ser removido e então o algoritmo MAP baseado em ICA resulta em

$$\begin{aligned} \mathbf{x} \leftarrow \operatorname{argmax}_{\mathbf{x}} (\ln p_{s_v}(W^v(\mathbf{y} - \mathbf{x})) + \ln p_{s_x}(W^x\mathbf{x}) + \\ + \ln |\det(W^v)| |\det(W^x)|), \quad (4) \end{aligned}$$

onde W^v e W^x são as matrizes de desmistura das componentes independentes do ruído e do sinal limpo, respectivamente. Visto que as matrizes são constantes para um determinado ruído e sinal limpo, o termo determinante em (4) pode ser omitido, segundo [9]. A estimativa de um *frame* do sinal $x(t)$ pode ser obtido usando o método do gradiente

$$\mathbf{x} \leftarrow \mathbf{x} + \lambda (W^{vT} f'_{s_v}(W^v(\mathbf{y} - \mathbf{x})) - W^{xT} f'_{s_x}(W^x\mathbf{x})), \quad (5)$$

onde λ é o tamanho do passo e $f'(s) = -\partial \ln p(s) / \partial s$.

A equação (5) é chamada de ICA-MAP(gen), que é o caso geral do algoritmo ICA-MAP. Deve-se notar que tal algoritmo não faz nenhuma suposição sobre o ruído, se ele é gaussiano ou não. Assumindo que o ruído é gaussiano, pode ser mostrado em [9] que (4) leva para o melhor estimador, isto é, filtro de Wiener, ou para o método *sparse code shrinkage*.

B. Modelo GGM

A *pdf* do algoritmo MAP em (4) é obtida através do modelo Gaussiano generalizado (GGM), usando em [11]. A escolha do modelo GGM é devido ao fato dele ser um bom ajuste para vários graus de não-gaussianidade, podendo modelar distribuições muito esparsas. A forma geral do modelo GGM é

$$p(x|\mu, \delta, \beta) = \frac{\omega(\beta)}{\delta} \exp \left[-c(\beta) \left| \frac{x - \mu}{\delta} \right|^{\frac{2}{1+\beta}} \right], \quad (6)$$

onde

$$c(\beta) = \left[\frac{\Gamma[3(1+\beta)/2]}{\Gamma[(1+\beta)/2]} \right]^{\frac{1}{1+\beta}} \quad (7)$$

e

$$\omega(\beta) = \frac{\Gamma[3(1+\beta)/2]^{\frac{1}{2}}}{(1+\beta)\Gamma[(1+\beta)/2]^{\frac{3}{2}}}, \quad (8)$$

onde $\Gamma[\cdot]$ é a função gama. μ e δ denotam a média e o desvio padrão dos dados, respectivamente. O parâmetro β controla a não-gaussianidade da distribuição. Esse modelo pode descrever distribuições gaussianas ($\beta = 0$), sub-gaussianas ($-1 < \beta < 0$) e super-gaussianas ($\beta > 0$). Por exemplo, quando $\beta = 1$, a distribuição é Laplaciana, quando $\beta \rightarrow -1$, a distribuição tende para a uniforme, e para $\beta \rightarrow \infty$, a distribuição tende para a função delta.

Para média zero, as funções f' em (5), que são o diferencial do logaritmo das *pdf*'s, podem ser expressadas como

$$f'(x) = \frac{2c(\beta)}{\delta(1+\beta)} \left| \frac{x}{\delta} \right|^{\frac{1-\beta}{1+\beta}}. \quad (9)$$

A fim de estimar o parâmetro β , pode ser usado o método MAP. Utilizando as amostras do sinal de treino, temos

$$\beta \leftarrow \operatorname{argmax}_{\beta} p(\beta|\mathbf{x}) \Leftrightarrow \operatorname{argmax}_{\beta} p(\mathbf{x}|\beta)p(\beta), \quad (10)$$

onde

$$p(\mathbf{x}|\beta) = \prod_t \omega(\beta) \exp \left[-c(\beta) |x(t)|^{\frac{2}{1+\beta}} \right] \quad (11)$$

e $p(\beta)$ é a distribuição *a priori* de β . Para modelar $p(\beta)$, é sugerido utilizar a distribuição gama deslocada

$$\begin{aligned} p(\beta) & \sim \text{Gama}(1 + \beta|a, b) = \\ & = \frac{b^a}{\Gamma[a]} (1 + \beta)^{a-1} \exp[-b(1 + \beta)], \quad (12) \end{aligned}$$

onde a e b são parâmetros ajustáveis. Foi usado $a = b = 2$, que dá uma distribuição *a priori* com 95% da densidade dentro da faixa $[-0, 5, 10, 5]$. A estimação pode ser realizada utilizando algum método do gradiente, ou empregando o método de otimização global DIRECT, utilizado em [9]

III. PROPOSTA

A implementação do algoritmo ICA-MAP, como está descrito em [9], é difícil de ser alcançada devido à forma das *pdf*'s modeladas. A modelagem das *pdf*'s é realizada no espaço da transformação ICA, onde os dados são mais esparsos, ou seja, a maior parte da energia do sinal está concentrada em poucas amostras, e as demais são aproximadamente zero. A Fig. 1 mostra um exemplo de distribuição esparsa $p(s)$.

O vetor gradiente no algoritmo em (5) utiliza o diferencial do logaritmo da *pdf*, como mostra os gráficos na Fig. 2.

É possível ver na Fig. 2 que o $\partial \ln p(s) / \partial s$ não é diferenciável em zero e retorna valores grandes próximos de zero. E como a maioria das amostras estão próximas de zero ou vão convergir para valores próximos de zero, isso é um problema. O algoritmo ICA-MAP, como está descrito em [9], não explicita essa dificuldade e nem a trata. Implementando o algoritmo ICA-MAP original, tem-se apenas o passo λ para

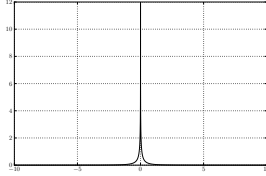


Fig. 1. Exemplo de uma distribuição esparsa usando o modelo GGM com média zero, variância unitária e $\beta = 5$.

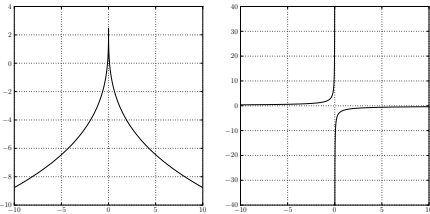


Fig. 2. O gráfico da esquerda mostra o $\ln p(s)$, e o da direita mostra $\partial \ln p(s) / \partial s$.

o controle do algoritmo. Usando valores usuais, como 10^{-2} ou 10^{-3} , a estimativa do sinal limpo retorna um nível alto de resíduo ruidoso, e à medida que diminui-se o passo, a possibilidade de retornar valores grandes no vetor gradiente em (5) aumenta.

Portanto, através de testes, foi verificado que uma solução para esse problema é a limitação da magnitude nas componentes g_i do vetor gradiente. A cada iteração, é verificado se a magnitude da i -ésima componente do vetor gradiente excede um certo limiar L . A função de limitação do vetor gradiente é

$$l(g_i) = \begin{cases} g_i & \text{se } |g_i| < L \\ \text{sgn}(g_i)L & \text{caso contrário} \end{cases}, \quad (13)$$

onde g_i é cada elemento do vetor gradiente, $\text{sgn}(\cdot)$ é a função sinal.

Com essa modificação é possível acelerar a convergência do algoritmo e eliminar o problema das amostras que convergem para valores próximos de zero. Pode-se ainda entender a introdução dessa modificação como uma restrição ao problema de otimização. O algoritmo fica

$$\begin{aligned} \mathbf{x} &\leftarrow \underset{\mathbf{x}}{\text{argmax}} (\ln p_{s_v}(W^v(\mathbf{y} - \mathbf{x})) + \ln p_{s_x}(W^x \mathbf{x})), \text{ com} \\ \text{abs}(W^{vT} f'_{s_v}(W^v(\mathbf{y} - \mathbf{x})) - W^{xT} f'_{s_x}(W^x \mathbf{x})) &\leq L, \end{aligned} \quad (14)$$

sendo $\text{abs}(\cdot)$ uma função que retorna o valor absoluto de cada componente de um vetor.

IV. TESTES E RESULTADOS

Os experimentos foram realizados usando sinais de voz da base de dados TIMIT [12], separados por gênero. Para o conjunto de treino para voz foram escolhidos aleatoriamente 30 sentenças de cada gênero, eliminando os silêncios no começo e final de cada sinal. Para o conjunto de testes foram tomadas 5 sentenças dos locutores fdaw0 (mulher) e mcpm0 (homem), do subconjunto dr1. Sentenças dos locutores de teste não foram usadas no conjunto de treino. Os sinais corrompidos

foram criados adicionando ruído ao sinal limpo com SNR em 0 dB e 5 dB. Foram usados quatro tipos de ruídos, o gaussiano e os outros três não-gaussianos retirados da base de dados DEMAND¹. Os ruídos não gaussianos escolhidos foram park, station e cafeter. Para o treino, foram utilizados 60 segundos de cada ruído. Os trechos usados para contaminar os sinais de voz não foram usados no treino. Todos os sinais foram reamostrados para 8kHz. Os *frames* foram gerados usando janela retangular. O algoritmo fastICA do pacote MDP² foi empregado para obter as transformações ICA. O valor utilizado para a limitação do gradiente foi $L = 500$. Os algoritmos foram implementados utilizando a linguagem de programação Python, juntamente com as bibliotecas Numpy, Scipy e Matplotlib. O algoritmo ICA-MAP em [9] é comparado com os algoritmos de subtração espectral em [2] e Sparse Code Shrinkage descrito em [9]. Para o algoritmo de subtração espectral, foi utilizado 0,5 segundo para a estimativa do espectro do ruído. As medidas de SNR e SD mostradas nas tabelas deste artigo são os valores médios dos resultados. As tabelas com todos os experimentos e todos os gráficos estão disponíveis em <http://www2.ele.ufes.br/~evandro/denoising>.

Denominando o sinal limpo como x e o sinal estimado como \hat{x} , o cálculo da SNR resultante é definida como

$$\text{SNR}_{\text{dB}} = 10 \log_{10} \frac{\sum x^2}{\sum (x - \hat{x})^2}. \quad (15)$$

Os resultados serão avaliados em termos da relação sinal-ruído (SNR) e distância espectral (SD). Como somente a informação da SNR é uma medida fraca para avaliar um algoritmo, também será usada a medida de distância espectral, que mede o quanto um espectro diverge de um outro, sendo desejável, se os espectros forem próximos, que a SD se aproxime de zero. A fim de calcular a SD entre x e \hat{x} , utilizar-se-á os sinais normalizados $x = x / \|x\|$ e $\hat{x} = \hat{x} / \|\hat{x}\|$, dessa forma a diferença de amplitude entre os sinais é desconsiderada. Então, cada sinal é dividido em *frames* não sobrepostos de 64 amostras, usando janela retangular e acrescentando 192 zeros em cada *frame*. O cálculo da SD entre dois espectros é

$$\text{SD} = \frac{1}{I} \sqrt{\sum_{i=0}^{I-1} \sum_{k=0}^{255} \left(10 \left(\log_{10} \frac{|X(k, i)|}{|\hat{X}(k, i)|} \right) \right)^2}, \quad (16)$$

onde k e i são os índices de frequência e de frame, respectivamente.

Primeiramente, são apresentados os gráficos de um resultado do experimento com ruído gaussiano. As Figs. 3 e 4 mostram os gráficos e espectrogramas, respectivamente, dos sinais limpo, corrompido com SNR de 0dB e estimados de cada algoritmo. Nas Tabelas I e II estão os valores médios de SNR e SD resultantes, respectivamente, dos experimentos com ruído gaussiano.

Pode-se notar que o ICA-MAP teve um resultado melhor tanto em SNR quanto em SD. Como o algoritmo SCS é implementado com uma fórmula fechada usando uma aproximação no problema de otimização, era esperado um desempenho pior, e acabou deixando um alto nível de resíduo ruidoso, como se

¹Disponível em <http://parole.loria.fr/DEMAND/>

²<http://mdp-toolkit.sourceforge.net/>

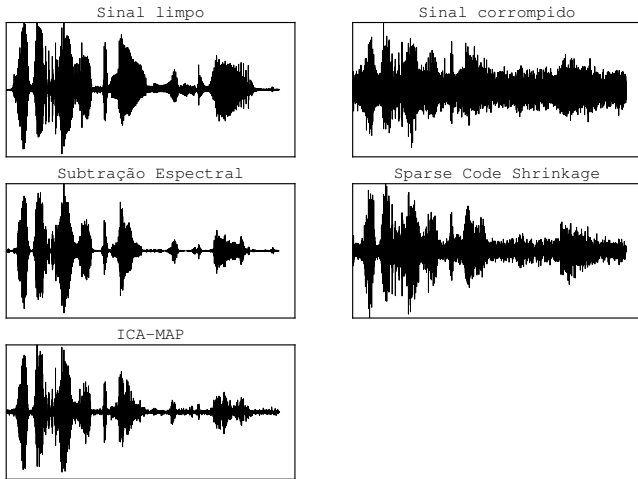


Fig. 3. Gráficos dos sinais limpo, corrompido por ruído gaussiano em 0dB e estimados por cada algoritmo.

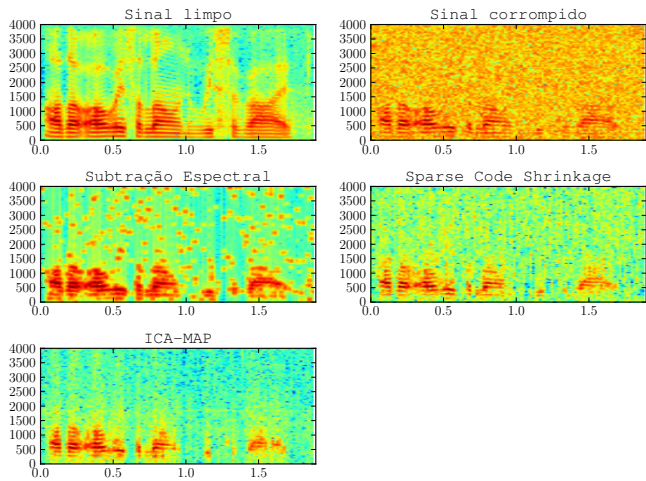


Fig. 4. Espectrogramas dos sinais limpo, corrompido por ruído gaussiano em 0dB e estimados por cada algoritmo.

TABELA I
COMPARAÇÃO DE SNR'S (EM dB) DOS SINAIS MELHORADOS CORROMPIDOS COM RUÍDO GAUSSIANO.

	Algoritmo	Ruído Gaussiano	
		0 dB	5 dB
homem	Subtração Espectral	5,41	7,52
	Sparse Code Shrinkage	5,25	8,73
	ICA-MAP (gen)	6,26	9,14
mulher	Subtração Espectral	5,70	8,07
	Sparse Code Shrinkage	4,97	8,53
	ICA-MAP (gen)	6,11	9,05

observa no sinal melhorado pelo SCS da Fig. 3. Lembrando que os valores tabelados são as médias de todos experimentos, é importante dizer que em alguns experimentos da subtração espectral, a SD piorou. Diferentemente, o resultado com subtração espectral e ICA-MAP sempre melhoraram.

Os resultados a seguir são de experimentos realizados com

TABELA II
COMPARAÇÃO DAS DISTORÇÕES ESPECTRAIS DOS SINAIS CORROMPIDOS (SD INICIAL) E DOS SINAIS CORROMPIDOS COM RUÍDO GAUSSIANO.

	Algoritmo	Ruído Gaussiano	
		0 dB	5 dB
homem	SD inicial	10,29	9,15
	Subtração Espectral	8,89	8,48
	Sparse Code Shrinkage	8,15	7,05
	ICA-MAP (gen)	5,83	5,68
mulher	SD inicial	10,26	9,19
	Subtração Espectral	7,70	6,65
	Sparse Code Shrinkage	8,18	7,15
	ICA-MAP (gen)	5,82	4,92

ruídos não-gaussianos. A Fig. 5 mostra um exemplo com o sinal limpo corrompido com o ruído park com SNR de 0dB.

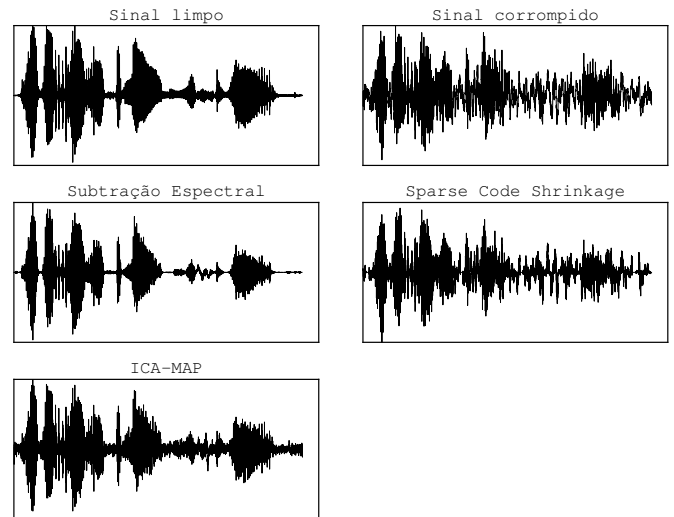


Fig. 5. Gráficos dos sinais limpo, corrompido por ruído não-gaussiano park em 0dB e estimados por cada algoritmo.

Nas Tabelas III e IV estão as medidas resultantes dos experimentos com ruído não-gaussiano.

O SCS teve o pior desempenho entre todos os algoritmos. Além de melhorar pouco a SNR, ele introduz artefatos no espectro, como pode ser visto na Fig.6, que faz piorar a SD. Já a subtração espectral melhora significativamente a SNR, mas em questão de SD, piora na maioria das vezes. O melhor resultado é do ICA-MAP, que não teve uma melhora na SNR tão grande quanto na subtração espectral, mas sempre melhora a medida de SD.

V. CONCLUSÕES

Neste artigo foi apresentada uma modificação na implementação do algoritmo ICA-MAP, limitando o vetor gradiente. Dessa forma, foi possível estabilizar melhor a evolução do algoritmo. Observou-se que o algoritmo ICA-MAP sempre melhora os sinais de voz tanto em relação a SNR tanto a SD. É importante salientar que o ICA-MAP depende de muitos fatores para ser executado, como uma boa estimativa das matrizes de desmistura do sinal limpo

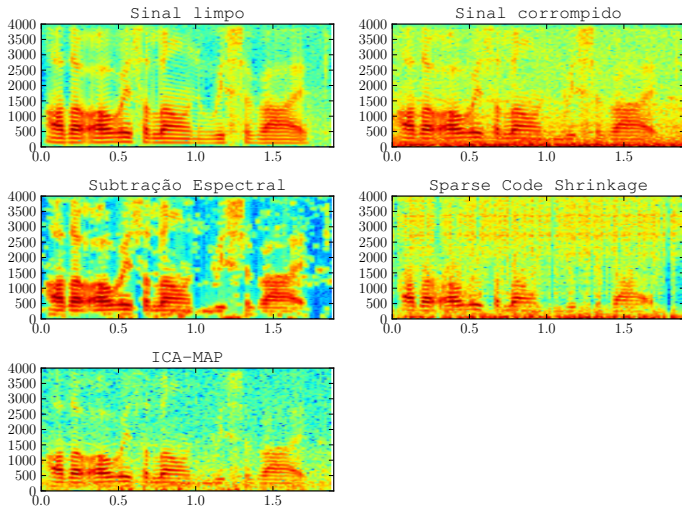


Fig. 6. Espectrogramas dos sinais limpo, corrompido por ruído não-gaussiano park em 0dB e estimados por cada algoritmo.

TABELA III
COMPARAÇÃO DE SNR'S (EM dB) DOS SINAIS MELHORADOS
CORROMPIDOS COM RUÍDOS NÃO-GAUSSIANOS.

	Algoritmo	park	
		0 dB	5 dB
homem	Subtração Espectral	9,21	11,46
	Sparse Code Shrinkage	1,94	6,28
	ICA-MAP (gen)	6,03	10,05
mulher	Subtração Espectral	10,76	12,82
	Sparse Code Shrinkage	1,77	6,23
	ICA-MAP (gen)	4,87	10,96
		station	
		0 dB	5 dB
homem	Subtração Espectral	5,70	7,92
	Sparse Code Shrinkage	2,91	6,95
	ICA-MAP (gen)	5,55	8,98
mulher	Subtração Espectral	6,88	9,22
	Sparse Code Shrinkage	2,57	6,76
	ICA-MAP (gen)	4,73	9,39
		cafeter	
		0 dB	5 dB
homem	Subtração Espectral	5,18	7,40
	Sparse Code Shrinkage	2,60	6,71
	ICA-MAP (gen)	4,34	8,54
mulher	Subtração Espectral	6,33	8,71
	Sparse Code Shrinkage	2,40	6,60
	ICA-MAP (gen)	3,76	8,46

e do ruído, que leve os sinais para um domínio esparso e uma estimativa dos modelos das pdf's do sinal limpo e do ruído que devem ser uma boa representação desses dados esparsos. Sabe-se que, através da técnica ICA, é possível obter uma representação esparsa, já que o algoritmo maximiza as não-gaussianidades dos dados, mas ainda seria possível utilizar outra técnica para obter essa transformação, por exemplo, utilizando algoritmo de otimização com norma ℓ_1 .

REFERÊNCIAS

[1] Boll, S., "Suppression of acoustic noise in speech using spectral subtraction," *Acoustics, Speech and Signal Processing*, IEEE Transactions on , vol.27, no.2, pp.113,120, Apr 1979.

TABELA IV
COMPARAÇÃO DE DISTORÇÃO ESPECTRAL DOS SINAIS CORROMPIDOS
(SD INICIAL) E DOS SINAIS MELHORADOS CORROMPIDOS COM RUÍDOS
NÃO-GAUSSIANOS.

	Algoritmo	park	
		0 dB	5 dB
homem	SD inicial	5,98	5,12
	Subtração Espectral	8,01	6,63
	Sparse Code Shrinkage	31,68	23,15
	ICA-MAP (gen)	5,24	4,38
mulher	SD inicial	6,20	5,39
	Subtração Espectral	5,70	4,52
	Sparse Code Shrinkage	27,52	19,76
	ICA-MAP (gen)	5,42	4,51
		station	
		0 dB	5 dB
homem	SD inicial	7,21	6,26
	Subtração Espectral	8,01	6,63
	Sparse Code Shrinkage	13,36	8,89
	ICA-MAP (gen)	6,45	5,69
mulher	SD inicial	7,41	6,53
	Subtração Espectral	7,74	6,29
	Sparse Code Shrinkage	11,07	6,94
	ICA-MAP (gen)	6,55	5,86
		cafeter	
		0 dB	5 dB
homem	SD inicial	6,83	5,91
	Subtração Espectral	14,03	9,11
	Sparse Code Shrinkage	12,10	7,74
	ICA-MAP (gen)	6,24	5,44
mulher	SD inicial	7,12	6,25
	Subtração Espectral	7,23	5,55
	Sparse Code Shrinkage	10,81	8,92
	ICA-MAP (gen)	6,45	5,72

[2] Y. Ephraim and D. Malah, "Speech enhancement using a minimum mean square error short-time spectral amplitude estimator," *IEEE Transactions in Acoust., Speech, Signal Process.*, vol. 32, no. 6, pp. 1109-1121, Dec. 1984.

[3] I. Y. Soon and S. N. Koh, "Low distortion speech enhancement," *Proc. Inst. Elec. Eng.*, vol. 147, no. 3, pp. 247-253, 2000.

[4] A. Hyvarinen, "Sparse code shrinkage: Denoising of nonGaussian data by maximum likelihood estimation," *Neural Computat.*, vol. 11, no. 7, pp. 1739-1768, 1999.

[5] J.-H. Lee, H.-Y. Jung, T.-W. Lee, and S.-Y. Lee, "Speech enhancement with MAP estimation and ICA-based speech features," *Electron. Lett.*, vol. 36, pp. 1506-1507, 2000.

[6] S. Gazor and W. Zhang, "Speech enhancement employing Laplacian-Gaussian mixture," *IEEE Trans. Speech Audio Process.*, vol. 13, no. 5, pp. 896-904, Sep. 2005.

[7] H. Xie, L. E. Pierce, and F. T. Ulaby, "SAR speckle reduction using wavelet denoising and Markov random field modeling," *IEEE Trans. Geosci. Remote Sens.*, vol. 40, no. 10, pp. 2196-2212, 2002.

[8] I. Potamitis, N. Fakotakis, G. Kokkinakis, and , "Speech enhancement using the sparse code shrinkage technique," in *Proc. Int. Conf. Acoustics, Speech, Signal Processing (ICASSP)*, Salt Lake City, UT, 2001, pp. 621-624.

[9] Xin Zou; Jancovic, P.; Ju Liu; Kokuer, M., "Speech Signal Enhancement Based on MAP Algorithm in the ICA Space," *Signal Processing, IEEE Transactions on* , vol.56, no.5, pp.1812-1820, May 2008.

[10] X. Zou, P. Jancovic, J. Liu. "The Efficiency of ICA-based Representation Analysis: Application to Speech Feature Extraction". *Chinese Journal of Electronics* Vol.20, No.2, pp.287-292, Apr. 2011.

[11] T. W. Lee and G. J. Lewicki, "The generalized Gaussian mixture model using ICA," in *Proc. Int. Workshop Independent Compon. Anal.*, Jun. 2000, pp. 239-244.

[12] Garofolo, John S., et al. "TIMIT Acoustic-Phonetic Continuous Speech Corpus. Linguistic Data Consortium," Philadelphia. 1993