

# Análisis y validación de algoritmos de separación de fuentes sonoras para aplicaciones en entornos industriales

A. Gómez<sup>1</sup> | A. Chacón-Rodríguez<sup>1</sup> | C. Lozano<sup>1,2</sup> | F. Merchán<sup>2</sup>

<sup>1</sup>DCILab, Escuela de Electrónica, Instituto Tecnológico de Costa Rica  
<sup>2</sup>Facultad de Ingeniería Eléctrica, Universidad Tecnológica de Panamá  
 agomez10010@gmail.com, {anaclevis.lozano, fernando.merchan}@utp.ac.pa

**Resumen:** este documento presenta la evaluación del costo computacional de las operaciones matemáticas básicas de tres algoritmos de separación de fuentes sonoras: FastICA, adaptativo basado en gradiente natural y adaptativo EASI basado en gradiente relativo. Estos algoritmos fueron seleccionados por su relativa simplicidad y la viabilidad de implementación en hardware de bajo costo en aplicaciones de localización acústica de agentes móviles en entornos industriales.

**Palabras claves:** separación ciega de fuentes (BSS), implementación en hardware, localización acústica.

**Title:** Analysis and Validation of Algorithms of Separation of Sound Sources for Applications in Industrial Environments

**Abstract:** This paper presents the evaluation of the computational cost of the basic mathematical operations of three algorithms separation of sound sources: FastICA, adaptive algorithm based on natural and adaptive EASI gradient based on relative gradient. These algorithms were chosen for their relative simplicity and viability of hardware implementation in low-cost applications of acoustic localization of mobile agents in industrial environments.

**Key words:** blind sources separation (BSS), hardware implementation, acoustic localization.

Tipo de artículo: original

Fecha de recepción: 5 de septiembre de 2014

Fecha de aceptación: 12 de noviembre de 2014

## 1. Introducción

Con las mediciones acústicas captadas de un entorno es posible obtener la distancia a la que se ubica un objeto emisor de un sonido [1], o utilizar la diferencia de tiempo y de fase entre las señales recibidas por una red de sensores para estimar la localización de una fuente (ver [2] para un estudio de fuentes bibliográficas al

respecto). No obstante, resulta claro que antes de poder localizar un objeto por el sonido que emite, es necesario saber si el sonido analizado proviene del objeto en cuestión. Para la extracción o separación ciega de fuentes BSS (del inglés "Blind Source Separation") se han desarrollado complejos algoritmos de procesamiento de señales, los cuales requieren cada vez mayor potencia en los cálculos matriciales y estadísticos, representando altas cargas computacionales [3]. Debido a las propiedades de las señales de audio tomadas de un ambiente real, donde las fuentes emiten sonidos simultáneamente y los obstáculos en el camino producen ecos y reverberación, la aplicación de las técnicas de BSS es un complejo problema [4]. Dichas técnicas utilizan los principios del procesamiento digital de señales y distribuciones probabilísticas basados en características supuestas de las señales, como la independencia estadística entre ellas [4, 5]. Considerando la viabilidad de implementar algoritmos de BSS en un circuito integrado, el primer paso es el análisis y la evaluación de las operaciones matemáticas básicas de las diferentes etapas de procesamiento de los algoritmos más comunes en aplicaciones de localización de agentes autónomos en movimiento en la industria [1], buscando soluciones óptimas de bajo costo por encima de altos estándares de robustez. En la sección II de este trabajo se presenta la descripción y el costo computacional de las tres estructuras implementadas en la plataforma de LabView de National Instruments. En la sección III se discuten los resultados de las pruebas de evaluación del comportamiento de cada algoritmo, usando la versión instantánea con mezclas construidas en el laboratorio. La sección IV presenta las conclusiones.

## 2. Descripción y Costo Computacional de los Algoritmos Diseñados

Con el fin de llevar a cabo la extracción de la fuente de interés, se asume que la mezcla capturada por cada micrófono tiene una independencia estadística entre las fuentes [5, 6]. Dichas mezclas, se clasifican como instantáneas o convolutivas, y pueden ser analizadas en el dominio del tiempo o de la frecuencia. También se ha demostrado la aplicación de algoritmos BSS en ambientes reales que utilizan técnicas de pre procesamiento para trabajar con las derivadas de las señales en lugar de las originales [7]. Por lo citado, se determinó que para la evaluación de los algoritmos en este trabajo, las muestras fueran instantáneas y en el dominio del tiempo.

En la implementación de los algoritmos se definen las siguientes convenciones simbólicas:

- $P$ : Número de micrófonos para las mediciones.
- $N$ : Tamaño de la muestra de datos.
- $J$ : Número de fuentes a estimar.
- $S$ : Matriz de fuentes de sonido del ambiente,  $J \times N$ .
- $A$ : Matriz de mezcla procesada,  $P \times J$ .
- $V$ : Matriz de mezclas observadas en los micrófonos,  $P \times N$ .
- $X$ : Matriz de mezclas después del pre-procesamiento,  $P \times N$ .
- $B$ : Matriz de separación final,  $J \times P$ .
- $Y$ : Matriz de estimación de las fuentes originales,  $J \times N$ .

## 2.1 Algoritmo FastICA

FastICA es uno de los algoritmos más conocidos y comúnmente utilizados para BSS en una gran cantidad de aplicaciones, por su robustez y velocidad de convergencia [4, 8]. Este algoritmo opera con datos en bloque, por lo que requiere de suficiente memoria para contener los bloques de  $N$  datos en cada etapa del procesamiento. Como muestra la Figura 1, el algoritmo FastICA consta de tres etapas.



Figura 1. Etapas del algoritmo FastICA. Las dos primeras etapas - centrado y blanqueo - son en realidad operaciones de pre procesamiento que optimizan el subsiguiente algoritmo de ICA.

El centrado es la media de los datos de la matriz de entrada  $v$ , calculada de la siguiente manera:

$$\bar{v}_i = v_i - \frac{1}{N} \sum_{j=1}^N v_{ij} \quad (1)$$

Las operaciones matemáticas básicas para centrar un vector fila se listan en la Tabla 1. A nivel de *hardware* este bloque puede replicarse  $P$  veces para calcular la media en paralelo de todas las filas, o insertar una lógica de secuencialización para utilizar los mismos recursos en  $P$  tiempos diferentes.

Tabla 1. Cantidad de operaciones para el centrado de datos.

Tipo	Cantidad por mezcla	Cantidad total
Sumas	$N$	$PN$
Multiplicaciones	1	$P$
Restas	$N$	$PN$
<b>Total</b>	$2N + 1$	$2PN + P$

El proceso de blanqueamiento de la señal conlleva el cálculo de las matrices de covarianza y de salida de datos, así como de los vectores y valores propios de la matriz covarianza. La Tabla 2 muestra la cantidad total de operaciones del proceso.

Tabla 2. Cantidad de operaciones para el blanqueamiento, donde  $k$  es el número de iteraciones, parámetro que determina la convergencia del algoritmo afectando la calidad de la separación.

Tipo	Cantidad total
Sumas	$\frac{1}{2}(3N-1)P^2 - \frac{1}{2}(N+1)P + k((2P-1)\sum_{j=1}^{P-1} j + P^3 - P)$
Multiplicaciones	$\frac{3}{2}(N+1)P^2 + \frac{1}{2}(N+3)P + k(P^3 + 3P^2 + 3P\sum_{j=1}^{P-1} j)$
Restas	$kP\sum_{j=1}^{P-1} j$
Divisiones	$kP + 1$
Raíz	$kP + P$
<b>Total</b>	$(3N+1)P^2 + 2P + 1 + k(2P^3 + 3P^2 + ((6P-1)\sum_{j=1}^{P-1} j) + P)$

Por último, se aplica el algoritmo central para la técnica FastICA, que tiene dos versiones: FastICA basado en curtosis y FastICA basado en negestropía [9]; para la evaluación del algoritmo central se eligió la versión basada en curtosis por su velocidad de convergencia cúbica que hace que sea entre 10-100 veces más rápido comparado con algoritmos basados en gradiente [9]. En la Tabla 3 se resume la cantidad de operaciones matemáticas que realiza el núcleo del algoritmo FastICA basado en curtosis.

Tabla 3. Cantidad de operaciones para el núcleo del algoritmo Fastica basado en curtosis, donde se observa la dependencia del costo computacional con número de iteraciones  $k$ .

Tipo	Cantidad total
Sumas	$k[(J-1)P^2 + 2NJ - J(N+2) + 1] + (J-1)\frac{P^2 + P}{2}$
Multiplicaciones	$k[(J-1)P^2 + (6J + 2NJ - 2)P + 2NJ] + J(J-1)\frac{P^2 + P}{2}$
Restas	$kP(2J - 1)$
Divisiones	$k(2J - 1)$
Raíz	$k(2J - 1)$
<b>Total</b>	$k[2(J-1)P^2 + (4NJ + 8J - 3)P + NJ + 2J - 1] + (2J-1)(J-1)\frac{P^2 + P}{2}$

## 2.2 Algoritmo adaptativo basado en gradiente natural

A diferencia del algoritmo FastICA que opera en bloque, este algoritmo opera dato a dato a medida que va ingresando en cada una de las mezclas, lo que representa una ventaja al no requerir grandes cantidades de memoria para almacenar grupos de datos. Además, por ser adaptativo, minimiza la dependencia estadística entre las fuentes estimadas tomando como base el parámetro de información mutua; no obstante, tiene la desventaja de que en condiciones iniciales se requeriría un tiempo apreciable para que el algoritmo empiece a converger a valores estables [10].

Debido a que este algoritmo solo requiere el centrado de los datos de entrada antes de ejecutar el núcleo central del mismo, se utiliza un filtro paso bajo (ecuación 2) para calcular la media  $m_i$  de cada mezcla  $x_i$ . Dicho cálculo realiza dos multiplicaciones, una suma y una resta (para extraer el dato centrado).

$$m_i(t) = \alpha x_i(t) + (1 - \alpha)m_i(t - 1) \quad (2)$$

El núcleo central del algoritmo adaptativo basado en gradiente natural se ejecuta en los pasos siguientes:

$$y(t) = B(t-1)x(t)$$

$$B(t) = B(t-1) - \mu[f(y)y^T - I]B(t-1)$$

donde B es la matriz de tamaño J x P y f(y) es la función de activación definida por:

$$f(y) = \frac{3}{4}y^{11} + \frac{25}{4}y^9 - \frac{14}{3}y^7 - \frac{47}{4}y^5 + \frac{29}{4}y^3$$

En la Tabla 4 se muestra el total de operaciones matemáticas de los dos pasos anteriores.

Tabla 4. Cantidad de operaciones para el núcleo del algoritmo adaptativo basado en gradiente natural

Tipo	Cantidad
Sumas	$PJ^2 - J$
Multiplicaciones	$(1 + P)J^2 + (2P)J$
Restas	$(1 + P)J$
<b>Total</b>	$(2P + 1)J^2 + 3PJ$

### 2.3 Algoritmo adaptativo EASI basado en gradiente relativo

Este algoritmo es de aprendizaje adaptativo también, que minimiza la dependencia estadística entre las fuentes estimadas. Al igual que el algoritmo basado en gradiente natural solo requiere de dos etapas, el centrado (dos multiplicaciones, una suma y una resta) y el núcleo central. La diferencia entre ellos es que algoritmo EASI incluye el gradiente relativo, representado por y que considera la matriz de separación como ortogonal. La cantidad de operaciones para el algoritmo EASI se muestra en la tabla 5, donde se observa que no hay gran variación en la cantidad de operaciones matemáticas ejecutadas.

Tabla 5. Cantidad de operaciones para el núcleo del algoritmo EASI.

Tipo	Cantidad
Sumas	$(1 + P)J^2 - J$
Multiplicaciones	$(2 + P)J^2 + (2P)J$
Restas	$J^2 + (1 + P)J$
<b>Total</b>	$2(P + 2)J^2 + (3P)J$

### 3. Análisis de resultados

Para evaluar el comportamiento de cada algoritmo implementado en LabView, se creó una matriz aleatoria de 20000 muestras a partir de la mezcla de tres fuentes de datos muestreados a una frecuencia de 22050 Hz. La selección de las fuentes se hizo con base en sonidos de elementos presentes en una ambiente industrial: el sonido repetido de la bocina de camión, la señal acústica de un motor de tractor en estado estacionario y una motosierra que acelera y desacelera (todas señales súper gaussianas). Con el fin de estimar la calidad de la separación, se utilizó la razón de señal a interferencia SIR (Signal to Interferences Ratio, por sus siglas en inglés).

Se tomó un bloque de  $N = 10000$  datos de cada fuente por separado antes de ser mezcladas, y con Matlab se calcularon los parámetros estadísticos de curtosis y de información mutua para cada una de ellas (resultados mostrados en la tabla VI), datos requeridos para el procesamiento del algoritmo central de FastICA.

Tabla 6. Parámetro de información n mutua entre cada par de fuentes y curtosis de cada una, parámetros estadísticos para maximizar FastICA.

Fuente	BocCamion	MotTractor	Motosierra	Curtosis
BocCamion	0,98	$8,13E-05$	$5,86E-04$	0,463
MotTractor	$8,13E-05$	1,00	$2,11E-05$	1,294
Motosierra	$5,86E-04$	$2,11E-05$	1,00	2,490

La primera prueba realizada fue observar el comportamiento del algoritmo FastICA con respecto a la cantidad de iteraciones ejecutadas. De los resultados que se muestran en la Figura 2, se puede resaltar que en doce iteraciones todas las fuentes han superado el 90% de su valor final; inclusive la fuente Motor Tractor demostró converger prácticamente a la primera iteración, cerca de 18 dB. El SIR promedio de la mezcla de las tres fuentes es aproximadamente 16 dB.

La segunda prueba se realizó con valores diferentes de la tasa de aprendizaje  $\mu$  en cada muestra, ejecutando los otros dos algoritmos (gradiente natural y EASI). Las simulaciones del algoritmo adaptativo basado en gradiente natural se muestran en las figuras 3 y 4, donde se observa que al aumentar el valor de la tasa de aprendizaje se acelera el proceso de convergencia: con  $\mu = 0.0005$  se requieren 20 iteraciones mientras que con  $\mu = 0.001$  solo ocho iteraciones. Para ambos casos el promedio final de la SIR es de alrededor de 27 dB. La mejor estimación se obtiene para MotorTractor con  $\pm 38$  dB y la peor para Motosierra con  $\pm 18$  dB.

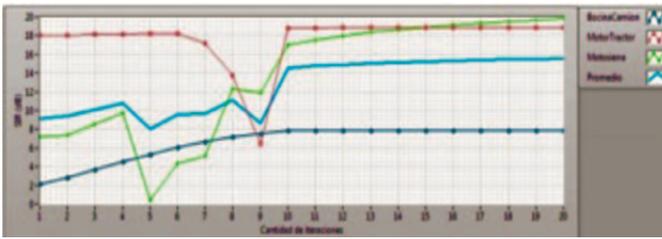


Figura 2. Influencia de la cantidad de iteraciones en el núcleo principal de FastICA sobre la calidad de la separación. Se resalta que casi todas las fuentes convergen a su valor final en  $\pm 12$  iteraciones; incluso la fuente MotorTractor muestra convergencia a la primera iteración, con un alto valor de la SIR (18 dB).

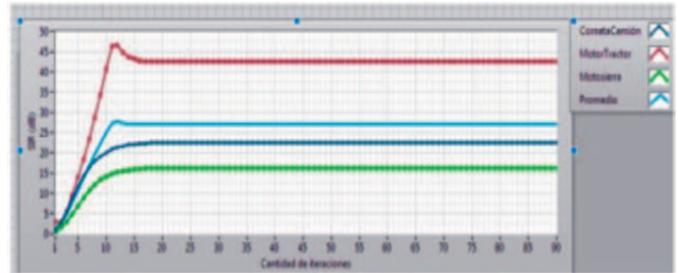


Figura 5. Influencia de la cantidad de iteraciones sobre la calidad de la separación con  $\mu = 0.0005$  del algoritmo EASI. No presenta gran diferencia entre éste y el gradiente natural, pues se reduce en  $\pm 5$  iteraciones.

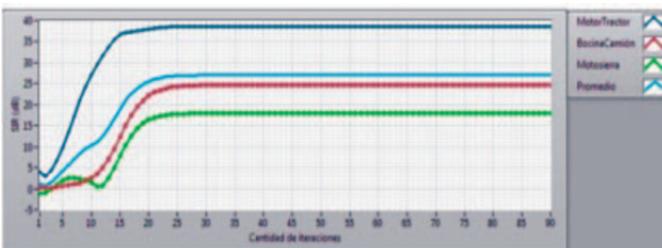


Figura 3. Influencia de la cantidad de iteraciones sobre la calidad de la separación con  $\mu = 0.0005$  con el algoritmo de gradiente natural. Para este valor de la tasa de aprendizaje el proceso de convergencia se lleva a cabo en  $\pm 20$  iteraciones.

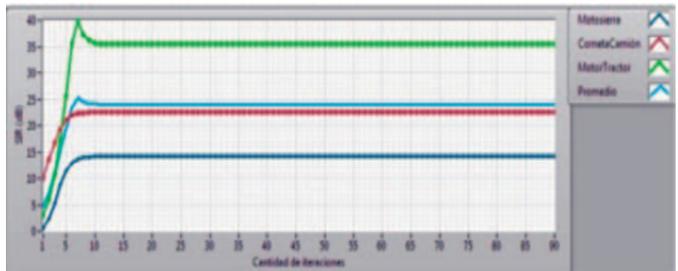


Figura 6. Influencia de la cantidad de iteraciones sobre la calidad de la separación con  $\mu = 0.001$  del algoritmo EASI. La diferencia entre éste y el de gradiente natural es en la reducción del promedio del SIR en aprox. 4 dB al aumentar  $\mu$ .

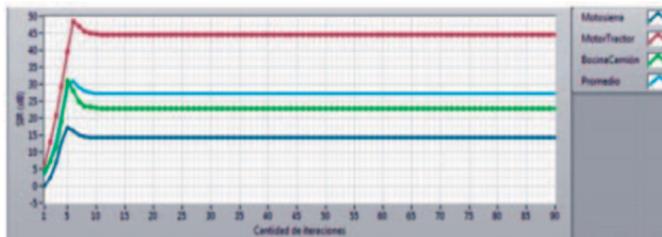


Figura 4. Influencia de la cantidad de iteraciones sobre la calidad de la separación con  $\mu = 0.001$  con el algoritmo de gradiente natural. Para este valor de la tasa de aprendizaje el proceso de convergencia se lleva a cabo en  $\pm 8$  iteraciones y el promedio de SIR se mantiene constante.

Se aplicaron los mismos parámetros de prueba al algoritmo adaptativo EASI basado en gradiente relativo. De los resultados presentados en las figuras 5 y 6, se observa que no hay una gran diferencia en el comportamiento de algoritmo EASI y el basado en gradiente natural en términos del número de iteraciones que toma para la convergencia. Sin embargo, el algoritmo EASI muestra que para una mayor tasa de aprendizaje hay una reducción en la SIR promedio y viceversa.

Por otro lado, cuantificando el total de operaciones matemáticas simples que realiza cada algoritmo (con  $P = 4$  mezclas y  $J = 3$  fuentes), el algoritmo FastICA es el que menor cantidad de operaciones ejecuta frente a los otros dos, como se muestra en la tabla VII. Esto se debe a que el algoritmo FastICA requiere efectuar varias ecuaciones diferentes pero que son independientes entre sí, lo que brinda más posibilidades para paralelizarlas, aunque ello implica utilizar mayores recursos de hardware.

Tabla 7. Cantidad de operaciones matemáticas básicas requeridas por cada algoritmo con  $\mu=0.001$ . A pesar de la gran cantidad de ecuaciones del algoritmo FastICA, requiere menor cantidad de operaciones.

Fuente	BocinaCamion	MotorTractor	Motosierra
BocinaCamion	0.98	$8.13E - 05$	$5.86E - 04$
MotorTractor	$8.13E - 05$	1.00	$2.11E - 05$
Motosierra	$5.86E - 04$	$2.11E - 05$	1.00

#### 4. Conclusiones

Se han analizado tres algoritmos de separación de fuentes sonoras basados en análisis de componentes independientes (ICA), estos fueron seleccionados debido a su relativa simplicidad, lo cual

permite una potencial implementación en un circuito integrado de bajo costo en aplicaciones de procesamiento acústico para ambientes industriales. Se cuantificó el costo computacional de cada uno en términos de la cantidad de operaciones matemáticas simples que se ejecutan, se propusieron estructuras lógicas que conceptualizan la posible implementación paralela de los algoritmos desarrollados y su aplicación en un sistema de localización de fuentes sonoras. Este análisis es extendible para la versión en frecuencia de los algoritmos con mezclas convolutivas.

## Referencias

- [1] J. Lanslots, F. Deblawe, K. Janssens, "Selecting Sound Source Localization Techniques for Industrial Applications", *The Noise and Vibration Control Magazine*, pp. 6-9, Junio 2010.
- [2] A. Chacon., F. Martin-Pirchio, S. Sanudo, and P. Julian, "A low-power integrated circuit for interaural time delay estimation without delay lines", *IEEE Trans. Circuits Systems II, Express Briefs*, 56, (7), 2009.
- [3] A. Van Schaik, S. Shamma, "A Neuromorphic Sound Localizer for a Smart MEMS System", *Analog Integrated Circuits and Signal Processing*, 39, pp. IV-864-867, 2004.
- [4] P. Common and C. Jutten. *Handbook of Blind Source Separation*, chapter 1: Introduction, pages 1–22. Elsevier Ltd, 1 edition, 2010.
- [5] A. Hyvarinen, J. Karhunen, E. Oja, "Independent component analysis", 2001.
- [6] V. Yan Fu Tan. *Blind audio source separation*, Mayo 2005. Fourth-year undergraduate project in Group F, 2004/2005.
- [7] M. Stanacevic and G. Cauwenberghs. Gradient flow adaptive beamforming and signal separation in a miniature microphone array. In *Proc. IEEE Int. Conf. Acoustic Speech and Signal Processing (ICASSP'2002)*, (Orlando FL, pages 13–17, 2002.
- [8] C. Alvarez, J. Monzón, "Aplicaciones de ICA con conceptos de estabilidad para separar señales", Departamento de Ingeniería, Facultad de Ciencias Exactas, Universidad Nacional del Nordeste, Corrientes – Argentina. Resumen: E-004, *Comunicaciones Científicas y tecnológicas*. 2006.
- [9] A. Hyvärinen. New approximations of differential entropy for independent component analysis and projection pursuit. In *Advances in Neural Information Processing Systems*, 10:273–279, 1998.
- [10] S. Amari, A. Cichocki, and H. H. Yang. A new learning algorithm for blind signal separation. In *Advances in Neural Information Processing Systems*, pages 757–763. MIT Press, 1996.