

**INTRODUCCIÓN AL  
RECONOCIMIENTO DE PATRONES**

**Trabajo Final**

**SEPARACIÓN DE AUDIO  
UTILIZANDO ICA**

**Mariana Díaz  
Ricardo Laureiro**

**Diciembre, 2008**

# ÍNDICE

<b>INTRODUCCIÓN</b>	<b>i</b>
<b>1 – MARCO TEÓRICO</b>	<b>1</b>
1.1 – Motivación para ICA	1
1.2 – Definiciones de ICA	1
1.3 – Ambigüedades de ICA	2
1.4 – Definiciones de independencia y propiedades fundamentales	3
1.5 – ¿Por qué variables gaussianas son prohibidas?	3
1.6 – Principios de la estimación ICA	3
1.7 – Minimización de la Información Mutua	6
1.8 – Estimación de Máxima Verosimilitud	6
1.9 – Preproceso para ICA	7
1.10 – Aplicaciones de ICA	8
<b>2 – IMPLEMENTACIONES DE ICA</b>	<b>9</b>
2.1 – FastICA	9
2.2 – EFICA	11
2.3 – EFICA + Clustering	13
<b>3 – EXPERIMENTACIÓN CON ALGORITMOS</b>	<b>16</b>
3.1 – Preproceso de señales y base de datos seleccionadas	16
3.2 – Condiciones de la simulación	17
3.3 – Acerca de la evaluación de desempeño	18
3.4 – Procesos de la simulación	20
3.5 – Resultados numéricos de las simulaciones	21
3.6 – Otras parametrizaciones y características de la recuperación	22
3.7 – Notas sobre sonidos particulares	22
<b>4 – COMENTARIOS FINALES</b>	<b>24</b>
<b>BIBLIOGRAFÍA</b>	<b>I</b>

## INTRODUCCIÓN

ICA (*Independent Component Analysis*), también conocida como BSS (*Blind Source Separation*), es una técnica utilizada para separar fuentes que se encuentran mezcladas. El clásico ejemplo es conocido como *cocktail party problem*. El mismo consiste en tratar de separar las conversaciones de varias personas que están hablando en una misma habitación al mismo tiempo y que son captadas por micrófonos ubicados en dicha habitación.

Se encuentran en la literatura varios algoritmos. En particular los algoritmos FastICA, EFICA y EFICA + Clustering. La realización de pruebas sobre este último algoritmo, es lo que motiva realizar el actual trabajo.

En el capítulo 1 se presenta un marco teórico de ICA, donde se discute sobre las características del algoritmo, así como las principales limitaciones. En el capítulo 2 se incluyen algunas particularidades de los algoritmos mencionados anteriormente.

Al final del capítulo 3 se presentan los resultados obtenidos al realizarse experimentación con diferentes variantes de ICA (FastICA, EFICA y EFICA + Clustering). Para llegar a ello fue necesario realizar las siguientes etapas. Conseguir mezclas de señales a separar, aplicarles los algoritmos de separación y medir luego el desempeño de dichas separaciones.

Se trabajó con dos voces mezcladas en diferentes idiomas, presentándose los resultados obtenidos especialmente para mezclas en idioma español.

Dado que no se disponían de mezclas para realizar las comparaciones de los algoritmos, fue necesario generar nuestras propias mezclas. Las mismas se realizaron en base a un simulador de habitación. Se mezclaron entonces voces en inglés, idiomas orientales y en español. Se trabajó con grabaciones *.wav* ya procesadas para tales efectos, las cuales fueron obtenidas de bases de datos públicas, a excepción de las de idioma español. Estas últimas fueron obtenidas de una base de datos pública, pero fue necesario realizar un acondicionamiento previo de las señales (nivel de continua, saturación, entre otros).

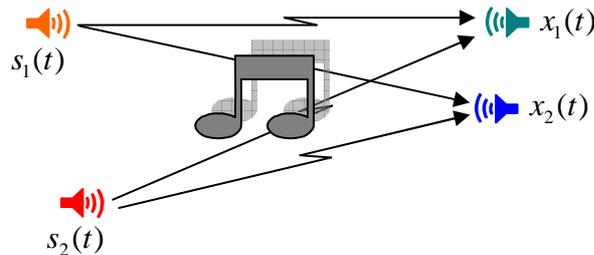
En cuanto a las medidas de desempeño, las comparaciones se realizaron en base a dos aspectos. Por un lado, a través de comparaciones subjetivas de la calidad de las grabaciones recuperadas y por el otro, en base a medidas objetivas. Como medida objetiva se seleccionó el SIR (*Source to Interference Ratio*) debido a que está ampliamente difundido en la literatura, en lo que se refiere a tratamiento de audio. De cualquier manera, se reconoce que es un indicador no totalmente fidedigno de la percepción del oído humano.

En el capítulo 4 se exponen las principales conclusiones obtenidas a partir de las simulaciones realizadas.

## 1 – MARCO TEÓRICO

### 1.1 – Motivación para ICA

ICA (*Independent Component Analysis*) es a veces denominado BSS (*Blind Source Separation*). Se presenta a continuación la aplicación típica que motiva el algoritmo, conocida como *cocktail party problem*.



Para este caso en particular, las ecuaciones que rigen el sistema son,

$$\begin{aligned} x_1(t) &= a_{11} \cdot s_1(t) + a_{12} \cdot s_2(t) \\ x_2(t) &= a_{21} \cdot s_1(t) + a_{22} \cdot s_2(t) \end{aligned} \quad (1)$$

Donde  $s_i(t)$  representan a las personas hablando (los parlantes) en el instante de tiempo  $t$  y los  $x_i(t)$  lo que captan los micrófonos (las mezclas de señales).

Los parámetros  $a_{ij}$  dependen de las distancias entre los micrófonos y los parlantes. No se consideran retardos ni reflexiones de las señales.

Para resolver dicho problema, alcanza con suponer que los  $s_i(t)$  son en cada instante de tiempo estadísticamente independientes.

### 1.2 – Definiciones de ICA

Supongamos ahora el problema extendido. Se consideran  $n$  fuentes ( $s$ ) y  $m$  mezclas ( $x$ ). Para la resolución del caso genérico es necesario suponer que  $m \geq n$ , pero en una primera instancia asumamos que tenemos igual número de fuentes que de receptores ( $m=n$ ).

En estas condiciones quedan planteadas  $n$  ecuaciones del tipo,

$$x_j = a_{j1} \cdot s_1 + \dots + a_{jn} \cdot s_{jn} \quad \forall j \quad (2)$$

El modelo de ICA asume que las fuentes son variables aleatorias independientes en cada instante de tiempo.

Se asume a su vez, sin perder generalidad, que tanto los  $s_i$  como los  $x_i$  tienen media cero.

En forma matricial, la ecuación del sistema queda planteada como,

$$x = A \cdot s \Rightarrow x = \sum_{i=1}^n a_i \cdot s_i \quad (3)$$

Donde  $a_i$  son las columnas de la matriz  $A$ .

Se debe asumir también que las distribuciones de probabilidad de los  $s_i$  no son gaussianas, siendo en principio densidades no conocidas.

Se buscará hallar una matriz  $W$ , próxima a  $A^{-1}$  de tal manera que podamos reconstruir  $s$  a partir de la siguiente operación,

$$s = W \cdot x \quad (4)$$

La asunción anterior de que  $m=n$ , lleva a que la matriz  $A$  sea cuadrada. También implica que no hay ruido en el sistema, es posible levantar con éxito esta hipótesis, pero no se abordará en este trabajo

Los elementos de  $s$ , se denominan según ICA, componentes independientes.

### 1.3 – Ambigüedades de ICA

- **No es posible determinar las varianzas (o energías) de las componentes independientes.**

Esto es debido a que tanto  $s$  como  $A$  son desconocidas, por ello se asume que  $E[s_i^2]=1$ , entonces  $A$  será adaptada para cumplir con esta restricción.

A pesar de esto, permanece la ambigüedad del signo, debido a que se puede multiplicar una componente independiente por  $-1$  sin afectar el modelo. En la mayoría de los problemas, este tema no es relevante.

Veremos este punto con un ejemplo.

Sea  $diag(\alpha_1, \dots, \alpha_n)$  una matriz diagonal cuyos elementos son  $\alpha_1, \dots, \alpha_n$  y el resto de sus elementos son cero. El sistema  $x = A \cdot s$  también se puede escribir como,  $x = [A \cdot diag(\alpha_1, \dots, \alpha_n)] [diag(\alpha_1^{-1}, \dots, \alpha_n^{-1}) \cdot s] = A' \cdot s'$ , por lo cual queda ejemplificado que no es posible recuperar la energía de  $s$ .

- **No es posible determinar el orden de las componentes independientes.** Esto también es debido a que tanto  $s$  como  $A$  son desconocidas.

Si se elige por ejemplo una matriz  $P$  de permutación, el sistema se puede escribir como  $x = [A \cdot P] [P^{-1} \cdot s] = A' \cdot s'$ , por lo cual queda claro que a menos de permutaciones se puede resolver el mismo sistema.

## 1.4 – Definiciones de independencia y propiedades fundamentales

$y_1, y_2$  son variables aleatorias independientes si y solo si  $p(y_1, y_2) = p_1(y_1) \cdot p_2(y_2)$

Si  $h_1, h_2$  son funciones, entonces  $E[h_1(y_1) \cdot h_2(y_2)] = E[h_1(y_1)] \cdot E[h_2(y_2)]$

Dos variables aleatorias  $y_1, y_2$  se dicen no correlacionadas si su covarianza es cero,  $Cov(y_1, y_2) = E[y_1, y_2] - E[y_1] \cdot E[y_2] = 0$ . Si las variables son independientes, son no correlacionadas, pero no es así a la inversa. Se pueden encontrar contraejemplos que verifican esto.

Muchos de los métodos de ICA restringen sus estimaciones a proporcionar estimaciones de las componentes independientes no correlacionadas. Esto permite reducir el número de parámetros libres (a determinar), simplificando el problema.

## 1.5 – ¿Por qué variables gaussianas son prohibidas?

Para ver esto, supongamos que  $A$  es una transformación ortogonal y las  $s_i$  son gaussianas no correlacionadas y de varianza unitaria. Entonces la distribución conjunta de densidades de probabilidad está dada por  $p(x_1, x_2) = \left(\frac{1}{2\pi}\right) e^{-\frac{x_1^2 + x_2^2}{2}}$ , esta distribución es completamente simétrica y por lo tanto no da información sobre la dirección de las columnas de  $A$ . Entonces,  $A$  no puede ser estimada.

Si sólo una de las componentes independiente es gaussiana, la estimación a través de ICA es posible.

## 1.6 – Principios de la estimación ICA

### Señales no gaussianas e independientes.

Por simplicidad se asumirá que todas las componentes independientes tienen la misma distribución de probabilidad.

Para estimar las componentes independientes consideramos  $y_i = w^T \cdot x = \sum_i w_i \cdot x_i$  donde  $w$  es el vector a estimar ( $W = (w_1, \dots, w_n)^T$ ). Si  $w$  fuese exactamente una fila de  $A^{-1}$  ( $x = A \cdot s$ ,  $s = W \cdot x$ ), obtendríamos una de las componentes independientes.

En la práctica no podemos determinar  $W$  exactamente, pues no tenemos conocimiento de la matriz  $A$ , pero se puede encontrar un estimador que dé una buena aproximación.

Considerando  $z = A^T \cdot w$  y recordando que  $y = w^T \cdot x$ , se obtiene que,

$$y = w^T \cdot x = w^T \cdot A \cdot s = z^T \cdot s \quad (5)$$

De esta manera se observa que  $y$  (estimación de  $s$ ) es una combinación lineal de los  $s_i$ . Como la suma de dos variables aleatorias es más gaussiana que las variables originales,  $z^T \cdot s$  es más gaussiana que cualquiera de las  $s_i$  (Teorema Central del Límite). A su vez  $y$  es menos gaussiana cuando iguala a algún  $s_i$ . En este último caso, solo un elemento de  $z$  ( $z_i$ ) es distinto de cero.

Entonces, se puede tomar  $w$  como el vector que maximiza la no-gaussianidad de  $y = w^T \cdot x$ . Este vector corresponde a  $z$  con un único elemento distinto de cero.

Un espacio de vectores  $w$  de dimensiones  $n$  tiene  $2n$  máximos locales, correspondientes a  $s_i$  y  $-s_i$ . Por lo tanto, es necesario encontrar todos estos máximos locales para encontrar los  $s_i$ .

### Medida de no gaussianidad.

Se indicarán a continuación, tres posibilidades, Kurtosis, Negentropía y aproximación a Negentropía

#### Kurtosis

Es un indicador clásico de no gaussianidad. El indicador propuesto es,

$$kurt(y) = E[y^4] - 3(E[y^2])^2 \quad (6)$$

Se asume que  $y$  tiene varianza unitaria ( $E[y^2]=1$ ). Kurtosis vale cero para variables aleatorias gaussianas. Distinguiéndose de acuerdo al signo,

$$\begin{aligned} kurt(y) < 0 &\Rightarrow \text{v.a. sub-gaussiana} \\ kurt(y) > 0 &\Rightarrow \text{v.a. super-gaussiana} \end{aligned} \quad (7)$$

La no gaussianidad es medida por el valor absoluto de Kurtosis, dicho indicador es muy utilizado debido a su simplicidad. Se puede estimar a partir del cuarto momento de los datos de las muestras ( $E[y^4]$ ) teniendo en cuenta que el segundo miembro de  $kurt(y)$  vale 3 por la hipótesis de varianza unitaria.

Kurtosis tiene las siguientes propiedades,

$$\begin{aligned} kurt(x_1 + x_2) &= kurt(x_1) + kurt(x_2) \\ kurt(\alpha \cdot x_1) &= \alpha^4 \cdot kurt(x_1) \end{aligned} \quad (8)$$

A modo de ejemplo consideremos dos componentes independientes ( $s_1$  y  $s_2$ ). Entonces,  $y = w^T \cdot x = w^T \cdot A \cdot s = z^T \cdot s = z_1 \cdot s_1 + z_2 \cdot s_2$

En estas condiciones,  $kurt(y) = kurt(z_1 \cdot s_1) + kurt(z_2 \cdot s_2) = z_1^4 \cdot kurt(s_1) + z_2^4 \cdot kurt(s_2)$

Considerando la restricción de varianza unitaria para  $y$ , y asumiendo que la misma vale para las  $s$ , obtenemos que,  $E[y^2] = z_1^2 + z_2^2 = 1$  (círculo unitario).

Por lo tanto, hay que maximizar en el círculo unitario,  $|kurt(y)| = |z_1^4 \cdot kurt(s_1) + z_2^4 \cdot kurt(s_2)|$ . El máximo se da donde uno de los elementos de  $z$  es distinto de cero (1 o -1) y el otro vale cero.

El principal problema que presenta Kuntosis es que es sensible a los outliers. No es una medida robusta.

### Negentropía

Se basa en la definición de entropía. Recordemos que la entropía para una variable aleatoria discreta es  $H(y) = -\sum_i p(y = a_i) \cdot \log(p(y = a_i))$  y para un vector con densidad de probabilidad  $f$ , se tiene que  $H(y) = -\int f(y) \cdot \log(f(y)) dy$ . Las variables gaussianas son las que tienen mayor entropía, entre las variables aleatorias de igual varianza (resultado de Teoría de la Información, [8]).

Para obtener una medida de no gaussianidad, como en el caso anterior que valga cero para variables aleatorias gaussianas, se define la negentropía como,

$$J(y) = H(y_{gauss}) - H(y) \quad (9)$$

Donde  $y_{gauss}$  es una variable aleatoria gaussiana que tiene la misma matriz de covarianza que  $y$ .

Negentropía tiene la propiedad de ser invariante frente a las transformaciones lineales invariantes. Si bien es una buena medida de no gaussianidad, es muy costoso computacionalmente, por ello se suele utilizar una aproximación a ella.

### Aproximación a Negentropía

Se basa en el principio de máxima entropía, la aproximación queda expresada como,

$$J(y) \approx \sum_{i=1}^p k_i [E[G_i(y)] - E[G_i(v)]]^2 \quad (10)$$

Donde  $k_i$  son constantes positivas,  $v$  son variables aleatorias de media nula y varianza unitaria y  $G_i$  son funciones no cuadráticas.

$J(y)$  es siempre positiva y vale cero para variables aleatorias gaussianas. Si se utiliza una única función no cuadrática  $G_i$ , la aproximación queda dada por,

$$J(y) \approx [E[G(y)] - E[G(v)]]^2 \quad (11)$$

Si  $G$  no crece demasiado rápido, se obtiene un estimador robusto. Dentro de las más utilizadas de las  $G$  están,

$$G_1(u) = \frac{1}{a} \log \cosh(a u);_{1 \leq a \leq 2} \quad G_2(u) = -e^{-u^2/2} \quad (12)$$

## 1.7 – Minimización de la Información Mutua

La minimización de la Información Mutua conduce a hallar la dirección de máxima no gaussianidad. La Información Mutua está dada por,

$$I(y_1, \dots, y_m) = \sum_{i=1}^m H(y_i) - H(y) \quad (13)$$

Donde las  $y_i$  son  $n$  variables aleatorias escalares e  $y$  es igual a  $y = [y_1, \dots, y_m]$ .

Es una medida natural de la independencia entre variables aleatorias. Es equivalente a la divergencia de Kullback-Leibler. Considera toda la estructura de dependencia entre las variables aleatorias y no solo la covarianza.

Es posible demostrar, que si se tiene una transformación lineal, se cumple,

$$y = W \cdot x \Rightarrow I(y_1, \dots, y_m) = \sum_{i=1}^m H(y_i) - H(x) - \log |\det W| \quad (14)$$

Si las  $y_i$  son no correlacionadas y de varianza unitaria, se cumple que  $E[yy^T] = WE[xx^T]W^T = I$

Por lo que  $\det(I) = 1 = \det(WE[xx^T]W^T) = \det(W) \det(E[xx^T]) \det(W^T)$

Con lo cual  $\det(W)$  debe ser una constante, por lo que,

$$I(y_1, \dots, y_m) = C - \sum_{i=1}^m J(y_i) \quad (15)$$

Donde  $C$  no depende de  $W$ .

Queda entonces mostrada la relación entre la Información Mutua y la negentropía. Por lo cual, encontrar la transformación invertible  $W$ , que minimiza la Información Mutua, es equivalente a encontrar las direcciones donde la negentropía es maximizada.

## 1.8 – Estimación de Máxima Verosimilitud

Se puede probar que maximizar la verosimilitud es equivalente a minimizar la Información Mutua. En estas condiciones se llega a que la log-verosimilitud ( $L$ ) es de la forma,

$$L = \sum_{t=1}^T \sum_{i=1}^n \log f_i(w_i^T x(t)) + T \log |\det W| \quad (16)$$

Donde  $f_i$  son las densidades de probabilidad de los  $s_i$ , las cuales se asumen conocidas.

Para ver la relación existente entre la Información Mutua y la log-verosimilitud, se toma la esperanza de la relación anterior,  $\frac{1}{T} E[L] = \sum_{i=1}^n E[\log f_i(w_i^T x(t))] + \log |\det W|$ . Si  $w_i^T x$  tuviese la misma distribución que  $s_i$  ( $f_i$ ), el segundo término de la expresión anterior quedaría  $-\sum_{i=1}^n H(w_i^T x)$ , por lo tanto, a menos de una constante, la máxima verosimilitud es igual a la Información Mutua.

Debido a los resultados alcanzados, es necesario estimar la densidad de probabilidad de  $w_i^T x$  como parte del método de máxima verosimilitud. Las mismas serán utilizadas para aproximar las densidades de  $s_i$ . En este caso, verosimilitud e Información Mutua son equivalentes.

## 1.9 – Preproceso para ICA

### Centrado

Sea  $m$  la media de  $x$ ,  $m = E[x]$ , se asigna  $x \leftarrow x - m$ , de esta manera  $s$  tiene media cero. Con este centrado no se pierde generalidad, dado que luego de estimar la matriz  $A$  con los datos centrados, es posible completar la estimación de  $s$ , a su media original, dada por  $A^{-1}m$ .

### Blanqueado (whitening)

Se le aplica al vector de observaciones  $x$  una transformación lineal de tal manera que el nuevo  $x$  ( $\tilde{x}$ ) tenga sus componentes no correlacionados y con varianzas unitarias. Es decir,  $E[\tilde{x} \tilde{x}^T] = I$ .

Una forma de blanquear a  $x$  es utilizar la descomposición en valores propios. Sea  $E[x x^T] = EDE^T$  donde  $E$  es la matriz ortogonal de vectores propios de  $E[x x^T]$  y  $D$  la matriz diagonal que contiene los valores propios. Si se define  $\tilde{x} = ED^{-1/2}E^T x$  con  $D^{-1/2} = \text{diag}(\alpha_1^{-1/2}, \dots, \alpha_n^{-1/2})$  se puede ver que  $\tilde{x}$  tiene varianzas unitarias. A su vez  $D^{-1/2}$  es fácil de calcular por ser diagonal.

Como  $x = As$  se puede ver que se cumple que  $\tilde{x} = ED^{-1/2}E^T Ax = \tilde{A}s$ . Este blanqueado trae como ventaja que la nueva matriz de mezcla ( $\tilde{A}$ ) es ortogonal ( $E[x x^T] = \tilde{A}E[x x^T]\tilde{A}^T = \tilde{A}\tilde{A}^T = I$ ). Esto reduce el número de parámetros a estimar ( $A \rightarrow n \times n$ ) y al ser  $\tilde{A}$  ortogonal tiene  $n(n-1)/2$  grados de libertad.

### Otros preprocesos

Si los datos consisten en señales en el tiempo, algún filtro pasabanda puede ser útil. Sea  $X$  la matriz donde cada columna corresponde a las muestras en cada tiempo, o sea  $X = x(1), \dots, x(T)$ . Si se nota  $M$  como el filtro pasabanda, se tiene que,

$$X^* = X M = A S M = A S^* \quad (17)$$

Con lo cual el modelo de ICA sigue siendo válido.

### **1.10 – Aplicaciones de ICA**

Por último, se resume a continuación algunas disciplinas donde se ha aplicado ICA con éxito. A saber,

- Aplicaciones médicas. Separación de señales de encefalogramas
- Aplicaciones financieras. Encontrar factores escondidos en datos financieros (drivers)
- Procesamiento de Imágenes. Reducción de ruido en imágenes naturales
- Telecomunicaciones. Separación de señales en comunicaciones celulares que utilizan la técnica CDMA
- Procesamiento de audio

## 2 – IMPLEMENTACIONES DE ICA

### 2.1 – FastICA

FastICA es una implementación de ICA. Se asume que los datos se encuentran previamente procesados, o sea centrados y blanqueados.

Dicho algoritmo está desarrollado de tal manera de encontrar una dirección, o sea un vector unitario  $w$ , cuya proyección maximice la no gaussianidad, dada por,

$$J(w^T x) \propto \left[ E[G(w^T x)] - E[G(v)] \right]^2 \quad (18)$$

#### Determinación de una componente independiente

Se detalla a continuación los pasos a seguir para hallar una de las componentes independientes. Para obtener todas las componentes, debe ejecutarse el algoritmo que se describirá a continuación, con las modificaciones que se detallan más adelante.

Se recuerda que la covarianza de  $w^T x$  debe ser 1 ( $E[s_i^2] = 1$ ). A su vez se asume que los datos están blanqueados, o sea,  $Cov(x) = I$ , por lo cual es equivalente a imponer que la norma de  $w$  debe ser 1.

Se consideran las funciones no cuadráticas  $G_1$  y  $G_2$  y sus derivadas  $g_1$  y  $g_2$  respectivamente, que se detallan a continuación.

$$\begin{aligned} G_1(u) &= \frac{1}{a} \log \cosh(a u);_{1 \leq a \leq 2} & G_2(u) &= -e^{-u^2/2} \\ g_1(u) &= \tanh(a u) & g_2(u) &= u e^{-u^2/2} \end{aligned} \quad (19)$$

#### Algoritmo:

1. Elegir un vector inicial de pesos  $w$  (por ejemplo randómico)
2. Sea  $w^+ = E[x g(w^T x)] - E[g'(w^T x)] w$
3. Sea  $w = w^+ / \|w^+\|$
4. Si no converge se vuelve al paso 2

Se describe a continuación el proceso que lleva a la determinación de la anterior algoritmia.

La convergencia del algoritmo implica que los valores viejos y nuevos de  $w$  apuntan a la misma dirección, es decir, su producto interno es casi 1.

Es pertinente recordar que  $w$  y  $-w$  definen la misma dirección, o sea que los componentes principales se pueden determinar a menos de un signo.

El máximo de la aproximación de negentropía de  $w^T x$  se obtiene en un cierto óptimo de  $E[G(w^T x)]$ . Dicho óptimo bajo la restricción de  $E[(w^T x)^2] = \|w\|^2 = 1$  está dado por la resolución de la ecuación,  $F = E[x g(w^T x)] - \beta w$

Para resolver la ecuación se utiliza el método de Newton. El jacobiano queda dado por  $J F(w) = E[xx^T g'(w^T x)] - \beta I = 0$ . Para simplificar el jacobiano, se aproxima el primer término por,  $E[xx^T g'(w^T x)] \approx E[xx^T] E[g'(w^T x)] = E[g'(w^T x)] I$

Con esta aproximación el jacobiano queda dado por una matriz diagonal, lo cual es fácilmente invertible.

De esta manera, la actualización de  $w$  queda dada por,

$$w^+ = w - \frac{E[x g(w^T x)] - \beta w}{E[g'(w^T x)] - \beta} \quad (20)$$

Para simplificar aún más la expresión, se multiplica ambos lados por  $\beta - E[g'(w^T x)]$ , quedando entonces una expresión para  $w$  igual a la planteada en el paso 2 del algoritmo, a menos de su norma.

Si bien lo ideal es utilizar todos los datos disponibles, en general no se hace debido al costo computacional. Usualmente se elige un conjunto menor de muestras, el cual se selecciona en forma independiente para cada iteración. Si el número de muestras considerado, no genera una convergencia satisfactoria, hay que aumentarlo.

### **Determinación de todas las componentes independientes**

Para estimar las demás componentes independientes, podríamos utilizar directamente el algoritmo descrito anteriormente. El problema radica es que pueden converger al mismo máximo, por ello es necesario descorrelacionar los vectores  $w^T x$  después de cada iteración.

Para realizar dicha descorrelación, se presentan dos de los métodos usuales.

#### **Método 1**

Supongamos que ya tenemos  $p$  componentes independientes estimadas. Entonces corremos el algoritmo anterior para hallar la componente  $w_{p+1}$ . Cada vez que el algoritmo termina el paso 3, se actualiza  $w_{p+1}$  de la siguiente manera,

$$3'. \quad w_{p+1} = w_{p+1} - \sum_{j=1}^p w_{p+1}^T w_j w_j$$

$$3''. \quad w_{p+1} = w_{p+1} / \sqrt{w_{p+1}^T w_{p+1}}$$

O sea que se descorrelaciona  $w_{p+1}$  de los valores hallados anteriormente, para posteriormente normalizarse nuevamente.

## Método 2

En este método (también conocido como de descorrelación simétrica) se van calculando todos los  $w_i$  al mismo tiempo (algoritmo visto anteriormente) y al final de cada iteración (paso 3) se descorrelacionan todos juntos, mediante la siguiente ecuación,

$$W = (WW^T)^{-1/2}W \quad (21)$$

Donde la matriz  $W = [w_1, \dots, w_n]^T$ .

$(WW^T)^{-1/2}$  se obtiene a partir de la descomposición en valores y vectores propios de  $WW^T$  ( $WW^T = FDF^T$ ) con  $D$  la matriz de los valores propios. De esta manera se calcula  $(WW^T)^{-1/2} = FD^{-1/2}F^T$ .

### **Propiedades del algoritmo FastICA**

- Convergencia cúbica (o al menos cuadrática). Los algoritmos ICA basados en descenso por gradiente tienen convergencia lineal
- El algoritmo es fácil de utilizar
- El algoritmo encuentra directamente los componentes independientes, sin necesidad de estimar las distribuciones de probabilidad como la mayoría de los algoritmos de ICA
- El desempeño del método puede ser optimizado eligiendo adecuadamente la función  $g$
- Los componentes independientes pueden ser hallados uno a uno lo que reduce la carga computacional
- FastICA tiene la mayoría de las ventajas de los algoritmos neuronales. Trabaja en paralelo, distribuido, es computacionalmente simple y requiere poca memoria

## **2.2 – EFICA**

Para dar paso a las principales características de EFICA (Efficient Fast ICA), recapitulemos las principales ideas manejadas hasta ahora, contextualizándolas para versiones particulares de ICA, pasando por fastICA para terminar luego con las modificaciones introducidas en EFICA.

El caso más simple de ICA, es el modelo de mezcla lineal instantáneo y sin ruido. El cual se expresa como hemos visto anteriormente por,  $x = A \cdot s$ .

Dentro de las hipótesis que se manejan es que, los  $s_{ij}$  son variables aleatorias mutuamente independientes, las densidades de probabilidad de los  $s_{ij}$  son no gaussianas (a menos de una) y que la matriz desconocida  $A$  es de rango completo.

Para resolver el problema básico de ICA se han implementado varios algoritmos, los cuales difieren en la forma de estimar las distribuciones de densidad de las señales originales o en como estas son remplazadas por funciones no lineales  $J$ .

La variante JADE está basada en la estimación de Kurtosis. NPICA utiliza un modelo no paramétrico de las funciones de densidad. La variante RADIAL utiliza una aproximación de la entropía de las densidades basada en orden estadístico.

El algoritmo fastICA utiliza tanto Kurtosis, como medida de no gaussianidad; así como otras, por ejemplo las aproximaciones por entropía mediante la utilización de un función no lineal  $G$ .

Uno de los cuestionamientos más importantes es la precisión teórica de los algoritmos desarrollados. Si bien muchos de los algoritmos convergen en teoría a la solución correcta, lo hacen bajo la hipótesis de que  $N$  (número de muestras de cada señal) tienda a infinito. Dado que esto es poco realista, genera como resultado que las fuentes no son completamente desmezcladas.

La literatura ha demostrado que fastICA está muy cerca de la cota inferior de Cramér-Rao, aunque no es igual. Se debe tener en cuenta que cuando el desempeño asintótico alcanza dicha cota inferior, se ha alcanzado la precisión absoluta y no puede ser mejorado.

Si bien fastICA es computacionalmente rápido, robusto y converge rápidamente, el problema que presenta este algoritmo es su error residual debido a la cantidad finita de muestras de las mezclas de señales.

El desarrollo de EFICA tuvo en cuenta las siguientes características de fastICA,

- El algoritmo simétrico fastICA puede ejecutarse con diferentes funciones no lineales  $G$  para diferentes fuentes
- En el paso de simetrización de cada interacción, es posible agregar constantes auxiliares, las cuales puedan ajustarse, con el objetivo de minimizar el error cuadrático medio en cada fila de  $W$
- El algoritmo se mantiene asintóticamente eficiente, si el óptimo teórico de las constantes auxiliares se reemplaza por sus estimaciones

EFICA combina las ideas del algoritmo simétrico fastICA generalizado, con la elección adaptativa de la función  $g$ .

El algoritmo consiste en tres pasos,

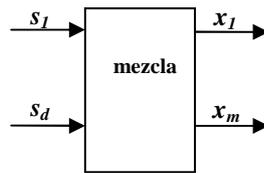
- Correr el algoritmo simétrico fastICA hasta que converja, utilizando para ello  $g(u) = \tanh(u)$  (la elección de esa  $g$  es debido a su universalidad, aunque es posible utilizar otras). Con ello se obtienen una estimación preliminar de las señales originales
- A partir de cada una de las salidas estimadas en el paso anterior, se realiza en este paso la selección de nuevas funciones  $g_k$  (una por cada señal a des-mezclar)
- A partir de la selección de las  $g_k$  se entra a una tercera etapa que realiza un ajuste fino para el cálculo de la matriz  $W$

### 2.3 – EFICA + Clustering

Se presenta a continuación, un nuevo algoritmo utilizado para separar mezclas de audio [1]. Forma parte de las técnicas de la familia BBS, desarrollándose en el dominio del tiempo y utilizando como base el algoritmo EFICA.

Este nuevo algoritmo contempla los retardos presentes al conformarse la señal en cada uno de los micrófonos de recepción, con el objetivo de recuperar de mejor manera las señales de audio originales.

El sistema planteado es el siguiente,



Se nota con  $n$  el tiempo de observación,  $s_i$  las fuentes de sonido ( $i=1,\dots,d$ ) y  $x_j$  la recepción del sonido en el  $j$ -ésimo micrófono ( $j=1,\dots,m$ ). De esta manera se define los correspondientes vectores  $s$  y  $x$  que contienen a todas las fuentes y micrófonos.

$$s(n) = [s_1(n), \dots, s_d(n)]^T \quad x(n) = [x_1(n), \dots, x_m(n)]^T \quad (22)$$

Sin perder generalidad se considera que  $d=m$ , esto es posible debido a que no hay ruido agregado.

El algoritmo ICA intenta resolver el problema  $x(n) = A s(n)$  donde las mezclas son instantáneas (sin retardo), siendo  $A$  la matriz de atenuaciones. A partir de hallar  $W$  (estimación de  $A^{-1}$ ) es posible recuperar  $s(n)$ .

En forma general en un sistema lineal MIMO (múltiples entrada, múltiples salidas), cada una de las salidas de un micrófono es representada como,

$$x_i(n) = \sum_{j=1}^d \sum_{\tau=0}^{M_{ij}} h_{ij}(\tau) s_j(n - \tau) \quad (23)$$

donde  $h_{ij}$  es la respuesta al impulso,  $n$  es el tiempo y  $\tau$  es el retardo. Dicha respuesta se acotó a un largo  $M_{ij}$ .

En estas condiciones es posible recuperar las señales de cada fuente a partir de un filtro de respuesta finita al impulso (FIR) que tendrá la forma,

$$\hat{s}_i(n) = \sum_{j=1}^m \sum_{\tau=0}^L w_{ij}(\tau) x_j(n - \tau) \quad (24)$$

Donde  $w_{ij}(\tau)$   $\tau = 0, \dots, L$  son los coeficientes de los filtros inversos.

Como se ha expresado anteriormente, el problema que presenta ICA es que son mezclas instantáneas, sin retardo ( $M_{ij} = 0 \forall i, j$ ).

Para representar el retardo existente que forma parte de la conformación del sonido de cada micrófono ( $1, \dots, L$ ), se considera un nuevo vector  $\tilde{x}(n)$  constituido y notado como,

$$\tilde{x}(n) = [x_1(n), x_1(n-1), \dots, x_1(n-L), \dots, x_m(n), x_m(n-1), \dots, x_m(n-L)]^T \quad (25)$$

Se puede mostrar que la descomposición en componentes independientes que se obtiene, son estimaciones reordenadas, atenuadas y retardadas de la copia original.

Sigue siendo necesario la hipótesis de que los datos sean secuencias no gaussianas e iid. A su vez para recuperar la señal de buena forma es necesario un  $L$  grande.

Bajo estas consideraciones el procedimiento a llevarse a cabo es el siguiente.

### Paso 1

$\tilde{x}(n)$  es un vector de  $m(L+1)$  componentes al cual se le aplica EFICA para obtener,

- las componentes independientes,  $m(L+1)$ , las cuales se denotan  $c(n)$
- la matriz de descomposición  $W$ , a partir de las cuales se obtiene  $c(n) = W \tilde{x}(n)$

### Paso 2

El aporte de este método consiste en que se puede agrupar los  $c(n)$  en m-cluster donde cada uno de ellos contiene las versiones filtradas de una única señal original.

El agrupamiento en los cluster implica la definición de una medida de distancia y la aplicación de una técnica de clustering. En [1] mencionan que la selección de la técnica de clustering más conveniente está actualmente en proceso de investigación. Para los resultados presentados en el artículo utilizan clustering jerárquico a través de la estrategia average-linkage.

### Paso 3

Cada cluster hallado se utilizará para recuperar la señal original. Se nota  $c^{(j)}(n)$  a un vector igual a  $c(n)$  donde fueron sustituidos por cero las filas que no pertenecen a el cluster  $j$ -ésimo. A partir de  $c^{(j)}(n)$  se reconstruye la señal en los micrófonos que son las versiones retrasadas de la fuente  $j$ -ésima.

$$\tilde{x}^{(j)}(n) = W^{-1} c^{(j)}(n) \quad (26)$$

Luego se halla la contribución de la fuente  $j$ -ésima a cada uno de los micrófonos ( $i$ ) lo cual se expresa como,

$$\tilde{s}_i^{(j)}(n) = \sum_{p=1}^{L+1} \tilde{x}_{(i-1)(L+1)+p}^{(j)}(n+p-1) \quad (27)$$

El estimado final de la fuente  $j$ -ésima,  $\hat{s}_j(n)$ , se obtiene como,

$$\hat{s}_j(n) = \sum_{i=1}^m \tilde{s}_i^{(j)}(n) \quad (28)$$

### 3 – EXPERIMENTACIÓN CON ALGORITMOS

#### 3.1 – Preproceso de señales y base de datos seleccionadas

Para poder realizar las pruebas y compararlas con los resultados presentados en [1], era necesario contar con bases de datos que tuviesen dos fuentes y dos mezclas de ellas. La información disponible en [9] si bien contaba con bases de datos de sonido, disponía sólo de una mezcla, lo que no se ajustaba a los requerimientos para nuestra simulación. Esta situación se reprodujo en otras bases de datos disponibles en Internet.

Dadas las circunstancias, para poder llevar a adelante las pruebas, disponíamos de tres opciones:

- a) utilizar una matriz de mezcla genérica
- b) generar nuestras propias grabaciones
- c) disponer de un simulador que nos permita realizar las mezclas de forma sintética

La opción a) es utilizada en varias bibliografías, pero nos pareció que podría distar mucho de una prueba real. Dados los exiguos plazos y falta de material/recursos que disponíamos para llevar adelante la opción b), optamos por la opción c). En [10] y [16] se pueden encontrar simuladores de ambiente para Matlab®, seleccionándose para estas pruebas el primero de ellos.

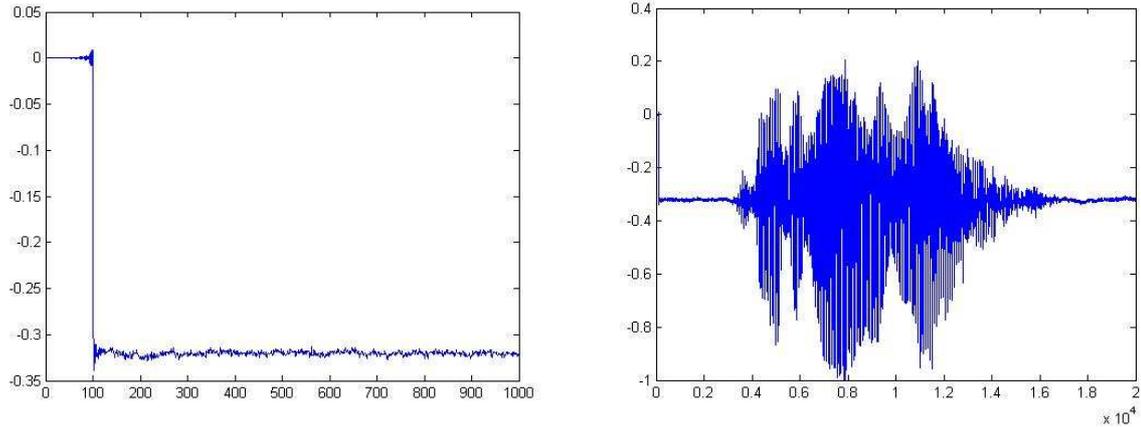
Dichos simuladores, permiten ubicar las fuentes de sonido y los puntos donde se quiere considerar las mezclas. Se compararon los resultados obtenidos con una mezcla real y los resultados fueron, desde un punto de vista auditivo, satisfactorios.

A partir del simulador de ambiente, generamos variadas mezclas, utilizando señales en inglés, en idiomas orientales y español. Dichas mezclas fueron generadas a partir de archivos *wav*. En particular las señales en español fueron obtenidas a través de grabaciones de sonidos recogidas por una tarjeta de sonido común de un PC comercial.

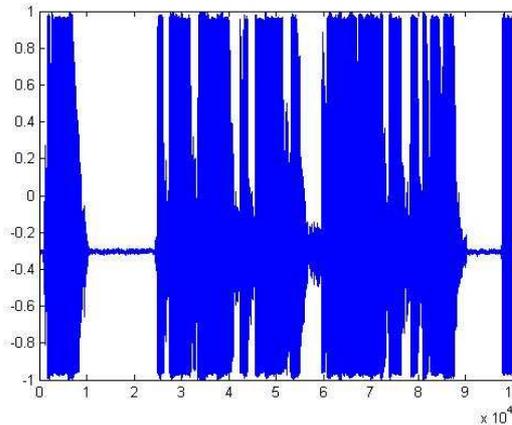
Es importante tener en cuenta, un par de situaciones que se presentan en algunas grabaciones que hace necesario realizarles un preproceso. La situación se ejemplifica en las gráficas de más abajo.

En la gráfica de la izquierda se observa que es usual que se genere un salto brusco de la señal en las primeras muestras, lo cual audiblemente es como un chasquido. Por lo cual, es necesario descartar estas primeras muestras.

En la gráfica de la derecha se puede ver que la señal a veces tiene un valor de continua que es necesario eliminando mediante un proceso previo a realizar la mezcla.



Otro fenómeno a tener en cuenta, ver gráfica de abajo, es la saturación que se presenta en algunas grabaciones. Es importante que esta situación no se de, dado que genera un sesgo importante de la señal de entrada.

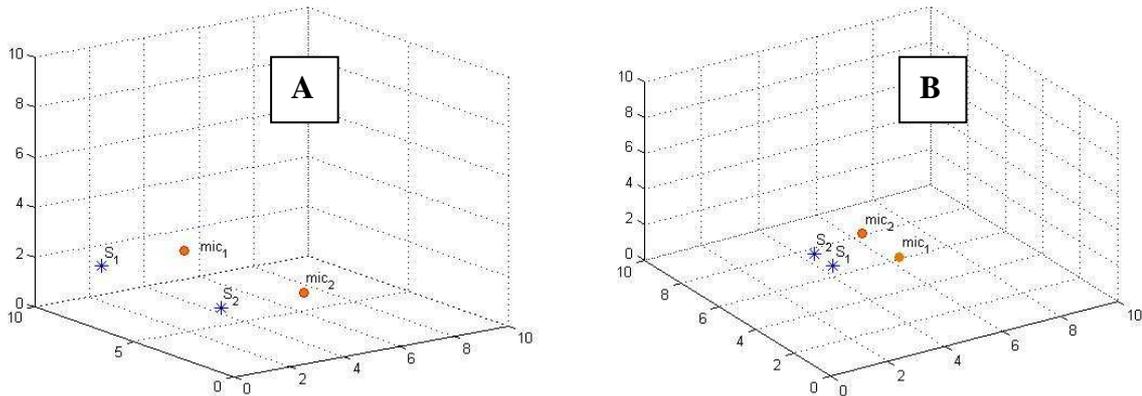


La frecuencia de muestreo seleccionada fue de 16 KHz. Las razones de la elección fueron que la mayoría de las bases de datos y varios artículos mencionan que se realizaron con estos valores. Por otra parte es una frecuencia razonable para señales de conversaciones como la que analizamos. Se utilizaron a su vez grabaciones mono a 16 bit de cuantificación.

### 3.2 – Condiciones de la simulación

Las simulaciones se realizaron teniendo en cuenta dos fuentes y dos mezclas. Se mezclaron personas hablando en varios idiomas y de ambos sexos. A su vez se realizaron algunas pruebas puntuales donde una de las fuentes era música.

También se ensayó con varias ubicaciones de las señales y de los puntos donde se obtenían las mezclas. Los resultados presentados fueron en base a las siguientes ubicaciones.



Como se indicó anteriormente, se trata del simulador referido en [10], el cual simula una habitación cúbica de 10x10x10 metros. La ubicación de cada uno de los puntos se hace en base a coordenadas (x,y,z), donde (x,y) dan la coordenada en la habitación y z da la altura en ella. Las coordenadas para los casos A y B son,

Caso A			
S <sub>1</sub>	S <sub>2</sub>	mic <sub>1</sub>	mic <sub>2</sub>
1,2,2	1,2,8	4,2,2	4,8,2
Caso B			
S <sub>1</sub>	S <sub>2</sub>	mic <sub>1</sub>	mic <sub>2</sub>
3,4,5,2	3,5,5,2	5,4,2	5,6,2

Las simulaciones se realizaron utilizando dos toolbox en Matlab® para lo cual se programaron un conjunto de rutinas las cuales incluían adaptaciones de las entradas y generaban las interfaces entre ellos. Ver [12].

En cuanto a la evaluación del desempeño de las señales recuperadas, nos basamos en rutinas propias y en un toolbox en Matlab®, [12], utilizado por [1] para realizar sus evaluaciones de desempeño. Nos pareció pertinente su utilización para tener un punto de comparación en común de los resultados alcanzados en [1].

### 3.3 – Acerca de la evaluación de desempeño

Para evaluar el desempeño de métodos de separación de audio, es necesario comparar las fuentes originales ( $s(t)$ ) con las señales estimadas ( $\hat{s}(t)$ ) después del proceso de separación. El problema radica en que no hay medidas objetivas para realizar dicha comparación.

Se define la señal  $s_{target}(t)$  como la señal original sometida a ciertas deformaciones que se asumen como permitidas, por ejemplo una ganancia constante. Las medidas más usuales de comparación de desempeño para audio radican en estimar la distancia entre la  $\hat{s}(t)$  y  $s_{target}(t)$ .

Este método supone que  $\hat{s}(t)$  se puede descomponer de la siguiente manera,

$$\hat{s}(t) = s_{target}(t) + e_{interf}(t) + e_{noise}(t) + e_{artif}(t)$$

donde:

$s_{target}$  es la deformación permitida de la fuente  $s(t)$

$e_{interf}$  es la deformación debido a otras fuentes

$e_{noise}$  es la deformación debido a fuentes de ruido

$e_{artif}$  es la deformación introducidas por sonidos de música o por el algoritmo de separación utilizado

La evaluación supone que, tanto las fuentes originales como el ruido que perturba la mezcla, están disponibles.

En estas condiciones, se pueden definir las siguientes medidas de desempeño,

$$\text{Source to Distortion Ratio: } SDR = 10 * \log \left( \frac{\|s_{target}\|^2}{\|e_{interf} + e_{noise} + e_{artif}\|^2} \right)$$

$$\text{Source to Interference Ratio: } SIR = 10 * \log \left( \frac{\|s_{target}\|^2}{\|e_{interf}\|^2} \right)$$

$$\text{Source to Noise Ratio: } SNR = 10 * \log \left( \frac{\|s_{target} + e_{interf}\|^2}{\|e_{noise}\|^2} \right)$$

$$\text{Source to Artifacts Ratio: } SAR = 10 * \log \left( \frac{\|s_{target} + e_{interf} + e_{noise}\|^2}{\|e_{artif}\|^2} \right)$$

Se consideró para evaluar las simulaciones realizadas que la deformación permitida sobre  $s(t)$  es una ganancia constante. Esta misma hipótesis fue llevada a cabo en [1].

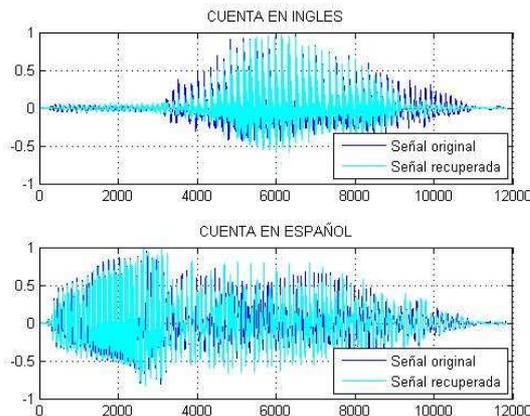
Al imponer que la deformación que se le permite a la señal estimada es una ganancia constante, la estimación de la componente  $e_{artif}$  tiene valores muy significativos lo que lleva a valores de  $SDR$  y  $SAR$  poco realistas. Al escuchar las señales  $s_{target}$ ,  $e_{interf}$  y  $e_{artif}$  se puede apreciar si la descomposición obtenida condice con la calidad apreciada en la señal estimada. Se pudo verificar en las simulaciones realizadas que los valores obtenidos de  $SDR$  y  $SAR$  no conciden con lo que se aprecia al escuchar las señales estimadas.

A su vez, no disponíamos de la componente ruido ( $e_{noise}(t)$ ), por lo que no se pudo obtener el  $SNR$ .

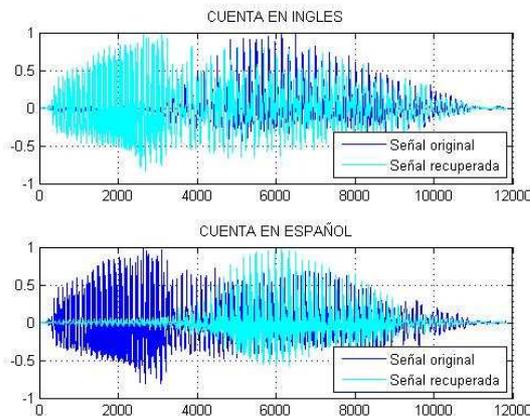
Por lo argumentado anteriormente, es que se optó por considerar como medida de desempeño y comparación el  $SIR$ .

### 3.4 – Procesos de la simulación

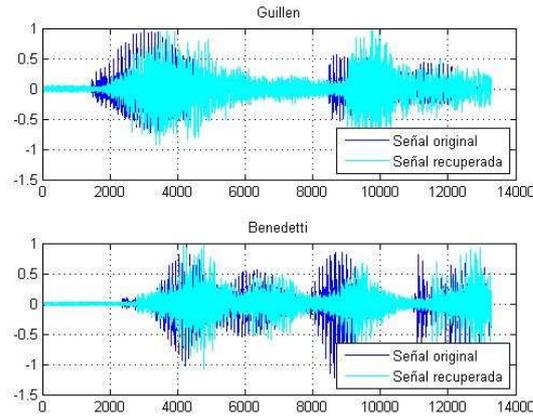
Se presenta a continuación un ejemplo que corresponden a la separación de dos señales. En él, se presenta la señal original y recuperada en una misma gráfica. El rango mostrado para cada una de las gráficas corresponde a la cuenta de un dígito en idioma inglés o español. Dicho ejemplo fue extraído de la simulación que más abajo se titula como “Conteo Inglés + Conteo Español”. Se trata de la cuenta de 1 hasta 10 en inglés y español, de forma simultánea.



Se muestra más abajo, para el mismo ejemplo indicado anteriormente, el caso cuando las señales recuperadas son dadas por el algoritmo en un orden diferente al que fueron presentadas las fuentes. El por qué fue analizado previamente en el Marco Teórico.



Se destaca a su vez que la señal estimada presenta un corrimiento respecto a la señal original, debido a las condiciones no instantáneas en las cuales se generó la mezcla. Dicho corrimiento se debe tener en cuenta para la comparación de las señales. Se presenta a continuación, un ejemplo típico de dicho retardo.



Para las simulaciones realizadas se utilizaron los siguientes parámetros.

FastICA:

Approach=symmetric

Stabilization=on

Nonlinearity=  $g(u) = u^3$

EFICA, EFICA + Clustering:

L=20

Largo del bloque 45000 (excepto conteo, =12000)

### 3.5 – Resultados numéricos de las simulaciones

#### Caso A

Voces en español		SIR					
		Fast ICA		EFICA		EFICA + Clustering	
1	N. Guillén + M. Benedetti	9.19	9.93	9.15	9.06	3.35	9.63
2	M. Benedetti + Rosalía de Castro	2.43	14.78	6.04	18.81	8.19	16.61
3	G. Mistral + Rosalía de Castro	6.77	7.40	13.58	14.86	14.61	8.44
Voces en Inglés							
4	Conteo Inglés + Conteo Español	14.61	7.58	7.91	7.99	14.45	10.69
5	Conteo Inglés + Música	3.54	7.90	5.73	11.18	6.34	9.78
6	2 voces masculinas en amb. Ruidoso	9.08	12.75	13.16	10.20	6.84	9.05

Al realizar una comparación auditiva entre EFICA y EFICA+Clustering, ambos presentan desempeños similares para los casos analizados. Presentando este último un menor tiempo de ejecución.

En general el desempeño de FAST-ICA en comparación con los otros dos algoritmos, desde un punto de vista auditivo, presenta un peor desempeño.

#### Caso B

Voces en español		SIR (EFICA + Clustering)	
1	N. Guillén + M. Benedetti	4.04	0.50

Para esta configuración, se pudo observar que las mezclas tenían mucha distorsión, lo cual dificultó la separación de las señales. Dichos resultados subjetivos se corroboran con un bajo valor de SIR.

Recordamos a su vez que estas mismas señales fueron mezcladas, como indica la planilla del Caso A, siendo los resultados mejores.

### 3.6 – Otras parametrizaciones y características de la recuperación

#### Largo de la ventana utilizada para ICA

Si bien se ha probado que cuanto más larga es la ventana de observación, el error cometido al recuperar las señales originales es menor, esto no resulta exactamente así desde el punto de vista auditivo.

Se varió el tamaño de la ventana de observación y el óptimo depende de cada timbre de voz, palabras, idiomas y entorno en particular. Incluso es usual que una voz quede más clara para un tamaño de ventana dado y la otra quede más clara para otro tamaño.

#### Largo del filtro L

Comentarios similares surgen de las pruebas realizadas para largo de filtro de 5, 10, 20 y 40. Se corrobora también que a medida que el largo del filtro crece el costo computacional también aumenta.

#### Sonidos metálicos y reverberaciones

En varias pruebas realizadas aparecen tonos metálicos en la voz, si bien desde el punto de vista de la calidad de una grabación no resulta un atributo deseable, para ciertos entornos sonoros resulta que mejora la comprensión del diálogo.

Las reverberaciones presentan un papel opuesto al mencionado anteriormente, su aparición está ligada a una menor comprensión del mensaje hablado.

### 3.7 – Notas sobre sonidos particulares

A partir de las simulaciones presentadas anteriormente, se analizó con más detalle la influencia de algunos sonidos particulares asociados al idioma. Para esto se tomó el caso indicado anteriormente como “Conteo Inglés + Conteo Español”.

Para generar dicho análisis nos valimos de las grabaciones de audio y de las medidas de SIR, calculadas en 10 ventanas correspondientes a los diez dígitos que se pronuncian en la grabación, tanto para el conteo en inglés como en español.

Los valores obtenidos de SIR son,

Dígito	1	2	3	4	5	6	7	8	9	10
SIR_ inglés	15,2	7,5	6,8	9,0	13,8	5,1	11,4	8,5	9,0	10,5
SIR_ español	7,7	10,4	7,1	9,1	7,0	12,4	8,0	9,7	9,5	8,7

Si se observan los valores obtenidos para el conteo en inglés, los dígitos 3 y 6 son los que presentan menor SIR, lo cual se corresponde a sonidos más asimilables a ruido. En dichos casos el desempeño del algoritmo de separación se ve degradado.

En español la degradación del SIR no se hace tan evidente para sonidos de la “s”, ya que estos están intercalados con vocales. Los casos en que se observa una pequeña degradación es para los dígitos 3 y 5.

## 4 – COMENTARIOS FINALES

Se resume a continuación los aspectos que se consideran a nuestro juicio más relevantes, surgidos a partir del proceso de elaboración del presente trabajo.

Si se desea comparar algoritmos de separación de señales, es imprescindible realizar las pruebas sobre las mismas mezclas, dada la diferencia significativa que se obtiene para diferentes grabaciones.

El desempeño de los algoritmos dependen mucho de las condiciones en las cuales se generaron las mezclas, por lo cual, la obtención de ellas a partir de micrófonos ubicados estratégicamente, es fundamental.

Si comparamos fastICA, EFICA y EFICA+Clustering a partir de los resultados obtenidos con el indicador SIR, estos presentan gran variabilidad dependiendo de la mezcla particular, por lo cual no es posible arribar a la conclusión que el algoritmo EFICA+Clustering tiene un desempeño superior a los otros.

Desde un punto de vista auditivo, se aprecia que el desempeño de EFICA y EFICA+Clustering es similar y a su vez superior a fastICA.

Se aprecia que el tiempo de ejecución de EFICA+Clustering es menor que el de EFICA.

Se analizó por separado el valor del SIR para los dígitos de conteo en español y en inglés, se observó que los valores de SIR más bajos correspondían a cuando se pronunciaban los números 3 y 6 en inglés. A su vez se identificó en las grabaciones que los mismos correspondían a sonidos más parecidos a ruido de fondo, lo que hizo más difícil la recuperación de dichos dígitos.

La aparición de voz metálica en las recuperaciones de señal no está necesariamente ligada a un peor desempeño desde el punto de vista del SIR o de la comprensión de los mensajes.

El variar el largo de la ventana de observación de ICA, así como el largo del filtro L, no generaron para los casos analizados mejoras cualitativas y cuantitativas en la calidad de las señales recuperadas.

En conclusión y a modo de comentarios finales, el proceso de investigación sobre el desempeño de los distintos algoritmos nos permite destacar que,

- Es necesario tener señales bien acondicionadas para evaluar los algoritmos, dado la variabilidad de los resultados a partir de diferentes señales
- De los casos analizados no es evidente un mejor desempeño del algoritmo EFICA+Clustering
- Si no se cuenta con un indicador objetivo de calidad, es preferible basarse en la percepción auditiva
- Es posible obtener inteligibilidad de la voz en las recuperaciones realizadas

## BIBLIOGRAFÍA

### Artículos proporcionados por los docentes

- [1] Algoritmo propuesto para mejorar el desempeño de EFICA en audio  
 Título: **Time-Domain Blind Audio Source Separation using Advanced ICA Methods**  
 Autores: Zbyněk Koldovský<sup>1</sup> and Petr Tichavský<sup>2</sup>  
 Institución:  
<sup>1</sup>Institute of Information Technology and Electronics, Technical University of Liberec, Czech Republic  
<sup>2</sup>Institute of Information Theory and Automation, Academy of Sciences of the Czech Republic, Czech Republic  
 Fecha: 2007
- [2] Totorial sobre ICA  
 Título: **A Review of Independent Component Analysis Techniques**  
 Autores: Bogdan Matei  
 Institución: Electrical and Computer Engineering Department Rutgers University, Piscataway, NJ, 08855-0909, USA

### Artículos obtenidos en Internet

- [3] Algoritmo propuesto para mejorar el desempeño de EFICA en audio  
 Título: **Time-Domain Blind Audio Source Separation Using Advanced Component Clustering And Reconstruction**  
 Autores: Zbyněk Koldovský<sup>1</sup> and Petr Tichavský<sup>2</sup>  
 Institución:  
<sup>1</sup>Institute of Information Technology and Electronics, Technical University of Liberec, Czech Republic  
<sup>2</sup>Institute of Information Theory and Automation, Academy of Sciences of the Czech Republic, Czech Republic  
 Fecha: 2008
- [4] Algoritmo EFICA  
 Título: **Efficient Variant of Algorithm FastICA for Independent Component Analysis Attaining the Cramér-Rao Lower Bound**  
 Autores: Zbyněk Koldovský, Member, IEEE, Petr Tichavský, Senior Member, IEEE, and Erkki Oja, Fellow, IEEE  
 Fecha: IEEE TRANSACTIONS ON NEURAL NETWORKS, VOL. 17, NO. 5, SEPTEMBER 2006
- [5] Algoritmos ICA y FICA  
 Título: **Independent Component Analysis: Algorithms and Applications**  
 Autores: Aapo Hyvärinen and Erkki Oja  
 Institución: Neural Networks Research Centre Helsinki University of Technology, Finland  
 Fecha: 2000
- [6] Algoritmos ICA  
 Título: **An introduction to independent component analysis**  
 Autores: Lieven De Lathauwer, Bart De Moor and Joos Vandewalle  
 Institución: JOURNAL OF CHEMOMETRICS  
 Fecha: 2000

### Libros

- [7] Generalidades de ICA  
 Título: **Pattern Classification**, Second Edition, Capítulo 10, **Unsupervised Learning and Clustering**  
 Autores: Duda, Hart y Stork
- [8] Teoría de la Información  
 Título: **Elements of Information Theory**  
 Autores: Cover y Thomas

### Páginas Web

- [9] Base de datos con sonidos  
<http://sisec.wiki.irisa.fr/tiki-index.php?page=Under-determined+speech+and+music+mixtures>
- [10] Base de datos con sonidos y simulador de ambiente  
[http://www.cnl.salk.edu/~tewon/Blind/blind\\_audio.html](http://www.cnl.salk.edu/~tewon/Blind/blind_audio.html)
- [11] Página de referencia sobre ICA y evaluación de desempeño de separación de señales  
<http://www.tsi.enst.fr/icacentral/>

[12] Página de Zbyněk Koldovský  
<http://itakura.kes.tul.cz/zbynek/index.htm>

[13] Página de Aapo Hyvärinen  
<http://www.cs.helsinki.fi/u/ahyvarin/>

[14] Página de referencia sobre ICA  
[http://www.cnl.salk.edu/~tewon/Blind/blind\\_audio.html](http://www.cnl.salk.edu/~tewon/Blind/blind_audio.html)

[15] Página de referencia sobre ICA  
<http://www.cis.hut.fi/projects/ica/>

[16] Simulador de ambiente  
<http://www.mathworks.com/matlabcentral/fileexchange/5184>