Departamento de Teoría de la Señal y Comunicaciones
Escuela Superior de Ingenieros
Universidad de Sevilla

PROYECTO FIN DE CARRERA

SEPARACIÓN CIEGA DE FUENTES EN MEZCLAS SINTÉTICAS DE VOZ

Autor: Óscar Muñoz Cueva
Tutor: Sergio Antonio Cruces Álvarez

Febrero de 2007

Notación

Variables.

A	Matriz de mezcla.
a_{ij}	Coeficiente de la matriz **A** situado en la fila *i*-ésima columna *j*-ésima.
B	Inversa de la matriz de mezcla **A**.
c	Velocidad de propagación del sonido.
$\mathbf{C}_{\mathbf{x}}$	Matriz de covarianzas del vector **x**.
d_i	*i*-ésimo autovalor de $\mathbf{C}_{\mathbf{x}}$.
$\mathbf{e}_i$	*i*-ésimo autovector de $\mathbf{C}_{\mathbf{x}}$.
d	Distancia entre micrófonos.
f	Variable que representa a la frecuencia.
f_s	Frecuencia de muestreo.
I	Matriz identidad.
P(*f*)	Matriz de permutación en la frecuencia *f*.
s_i	Variable aleatoria correspondiente a un componente independiente.
$s_i(t)$	Expresión temporal de una señal fuente (componente independiente).
$s(m)$	Expresión en tiempo discreto de $s(t)$.
s	Vector que contiene los componentes independientes.
$S(n,\omega)$	Transformada STFT de $s(m)$.

τ_{ij}	Retardo de propagación desde una fuente *j* hasta un sensor *i*
τ_{j}	Retardo de propagación de la fuente *j* entre los dos micrófonos en una grabación estereofónica.
μ_{x}	Media estadística de la variable aleatoria *x*.
σ_{x}	Varianza de la variable aleatoria *x*.
x_i	Variable aleatoria correspondiente a una mezcla de componentes independientes.
$x_i(t)$	Expresión temporal de una mezcla de componentes independientes.
$x(m)$	Expresión en tiempo discreto de $x(t)$.
x	Vector que contiene las mezclas de componentes independientes.
$X(n,\omega)$	Transformada STFT de $x(m)$.
$w(m)$	Expresión en tiempo discreto de una ventana.
W	Matriz de separación.
W(*f*)	Matriz de separación en el dominio de la frecuencia.
w_{ij}	Coeficiente de la matriz **W** situado en la fila *i*-ésima columna *j*-ésima.
$y_i(t)$	Expresión temporal de la estimación de un componente independiente.
$y(m)$	Expresión en tiempo discreto de $y(t)$.
$Y(n,\omega)$	Transformada STFT de $y(m)$.
$\delta(t)$	Función delta de Dirac.
θ_{j}	Ángulo de llegada del frente de ondas de la fuente *j*.
ω	Variable que representa a la frecuencia angular.

Operadores.

$(\cdot)^{T}$	Matriz o vector traspuesto.
$(\cdot)^{*}$	Matriz o vector conjugado.
$(\cdot)^{+}$	Matriz pseudoinversa.
$(\cdot)^{-1}$	Matriz inversa.
$(\cdot) \bullet (\cdot)$	Producto punto a punto de dos vectores o matrices de las mismas dimensiones (producto Hadamard).
$(\cdot)^{\bullet n}$	Matriz o vector donde sus elementos se elevan a la *n*-ésima potencia punto a punto (potencia Hadamard de orden *n*)
$(\cdot) * (\cdot)$	Convolución de dos señales.
$\lvert(\cdot)\rvert$	Magnitud de un número.
$\lVert(\cdot)\rVert$	Norma de un vector.
$E\{\ \}$	Media estadística.
$\mathrm{Cum}(\cdot,\cdot)$	Matriz de cumulantes.
$\det(\cdot)$	Determinante de una matriz.
$\mathrm{kurt}(\cdot)$	Kurtosis de una variable aleatoria.
$vec(\cdot)$	Agrupa las columnas de una matriz cuadrada formando un vector columna.
$\nabla(\cdot)$	Gradiente de una función.

Índice

1. Introducción.
1.1 Descripción del problema. 1
1.2Motivación 2
1.3 Estructura del proyecto. 3

2. Análisis de Componentes Principales, blanqueo y ortogonalización.
2.1 Introducción. 7
2.2 Representación lineal de datos multidimensionales 7
2.3 Análisis de componentes principales (PCA). 8
2.3.1 Interpretación de PCA como maximización de la varianza. 10
2.3.2 Interpretación de PCA como minimización del error cuadrático medio. 11
2.3.3 Elección del número de componentes principales. 12
2.3.4 Cálculo completo de PCA. 14
2.4 Blanqueo. 15
2.5 Ortogonalización. 16
2.6 Conclusiones. 16

3. Análisis de Componentes Independientes.
3.1 Introducción. 19
3.2 Separación Ciega de Fuentes. 20
3.2.1 Medidas de mezclas de señales desconocidas. 20
3.2.2 Separación de fuentes basada en la independencia. 21
3.3 Descripción formal de ICA. 23
3.3.1 Definición. 23
3.3.2 Cómo encontrar los componentes independientes. 24
3.3.3 Restricciones de ICA. 27
3.3.4 Ambigüedades de ICA. 28
3.3.5 Otras consideraciones. 29
3.3.5.1 Centrado de las variables. 29
3.3.5.2 ICA frente al blanqueo. 29
3.3.5.2 ICA y las variables gaussianas. 30
3.4 Criterios ICA y técnicas de optimización. 32
3.4.1 Maximización de la no-gaussianidad. 32
3.4.1.1 No gaussianidad e independencia. 32
3.4.1.2 Medida de la no-gaussianidad mediante kurtosis. 34
3.4.1.3 Algoritmo del gradiente usando kurtosis. 36
3.4.1.4 Algoritmo de punto fijo usando kurtosis. 37
3.4.1.5 Medida de la no-gaussianidad mediante la negentropía. 38
3.4.1.6 Aproximación de la negentropía. 39
3.4.1.7 Algoritmo del gradiente usando la negentropía. 40
3.4.1.8 Algoritmo de punto fijo usando negentropía. 40

3.4.2 Estimación de máxima verosimilitud. 41
3.4.2.1 La verosimilitud en el modelo ICA. 42
3.4.2.2 Algoritmos para la estimación de la máxima verosimilitud. 43
3.4.3 Minimización de la información mutua. 45
3.4.3.1 Definición de ICA usando la información mutua. 45
3.4.3.2 Información mutua y no-gaussianidad. 46
3.4.3.3 Información mutua y verosimilitud. 47
3.4.3.4 Algoritmos para la minimización de la información mutua. 47
3.5 Conclusiones. 47

4. La transformada de Fourier de corta duración (STFT).
4.1. Introducción. 49
4.2. Enventanado de señales. 49
4.3 Definición de la transformada STFT. 54
4.4 Transformada STFT inversa mediante la técnica de overlap-add. 58
4.5 Análisis de la voz usando la transformada STFT. 60
4.6 Conclusiones. 63

5. Modelos de Mezcla de la Voz.
5.1 Introducción. 65
5.2 Modelo de mezcla instantánea. 66
5.3 Modelo convolutivo de mezcla. 66
5.3.1 Descripción cualitativa del modelo. 66
5.3.2 Formulación del modelo en el dominio temporal. 67
5.4 Modelo de mezcla anecoica. 68
5.4.1 Formulación en el dominio del tiempo. 69
5.4.2 Formulación en el dominio de la frecuencia. 70
5.5 Caso sobredeterminado. 71
5.6 Conclusiones. 72

6. Separación ciega de señales de voz.
6.1 Introducción. 75
6.2 Separación ciega de señales de voz mediante enmascaramiento. 76
6.2.1 Planteamiento del problema. 76
6.2.2 Algoritmo de separación. 78
6.2.3 El algoritmo EM (Expectation Maximization). 79
6.3 Separación ciega de señales de voz mediante procesamiento adaptativo en el dominio de la frecuencia (método de Anemüller). 82
6.3.1 Planteamiento de la situación. 82
6.3.2 Algoritmo de optimización. 84
6.4 Separación ciega de mezclas convolutivas de voz en el dominio de la frecuencia. 87
6.4.1 Planteamiento del problema. 88
6.4.2 Solución ICA en cada subbanda de frecuencia. 91
6.4.3 Alineación de las permutaciones locales. 92
6.4.4 Ajuste del escalado. 96
6.5 Conclusiones. 97

7. Simulaciones.
7.1 Introducción. 99
7.2 Descripción de la situación de partida. 99
7.3 Separación usando enmascaramiento. 102
7.4 Separación usando el método de Anemüller. 107
7.5 Separación usando ICA independientemente en cada subbanda. 110
7.6 Casos no resueltos satisfactoriamente. 116
7.6.1 Fallos usando masking. 116
7.6.2 Fallos usando el método de Anemüller. 117
7.6.3 Fallos usando el método de subbandas. 117
7.7 Conclusiones. 118

8. Conclusiones y líneas futuras de investigación.
8.1 Conclusiones. 119
8.2 Líneas futuras de investigación. 120

Apéndices.
Apéndice 1 – Cumulantes. 121
Apéndice 2 – Método del gradiente. 123
Apéndice 3 – Guía para el usuario del programa realizado en Matlab. 126

Acrónimos. 139

Referencias Bibliográficas. 141

Capítulo 1

Introducción

1.1 Descripción del problema.

En este proyecto vamos a tratar de estudiar e implementar una serie de métodos que nos permitan, a partir de una grabación tomada por varios micrófonos simultáneamente donde hay varias personas hablando, extraer las voces de que está compuesta la mezcla.

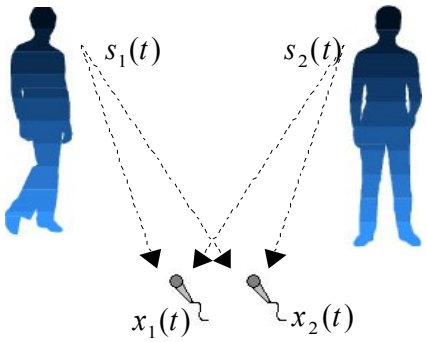


Figura 1.1 *Grabación de una mezcla de dos voces mediante dos micrófonos.*

Nuestro objetivo será tomar como datos únicamente las grabaciones de los micrófonos (aunque veremos que en una de las soluciones propuestas también hace falta conocer la distancia entre los micrófonos), a las que llamaremos observaciones, y a partir de estas ser capaces de estimar las voces que emitieron las personas que hablaban, es decir, las señales fuente.

Unas veces trataremos de estimar el modelo de mezcla para después invertirlo y recuperar las señales fuente, y otras lo que intentaremos será conocer a partir de los datos las direcciones con que llegan a los micros los frentes de onda correspondientes a cada fuente. Veremos que generalmente necesitaremos que las grabaciones se realicen con tantos micrófonos como fuentes pretendamos extraer de las mismas, ya que de otro modo la situación tiene difícil solución usando las técnicas que vamos a considerar en este texto, fundamentalmente el análisis de componentes independientes.

1.2 Motivación.

Extraer una sola voz de una grabación tomada por varios micrófonos donde hablen a la vez múltiples personas y posiblemente con ruido de fondo es una tarea que no es para nada trivial. Pues bien, el oído humano (o más concretamente el cerebro) es capaz de hacerlo por sí solo, sin embargo hay personas con deficiencias auditivas que no tienen esta capacidad, creándoles la necesidad de llevar sofisticados aparatos que puedan imitar el procesamiento de señales que lleva a cabo un sistema auditivo sano.

Se encuentran problemas similares cuando los sistemas de reconocimiento automático de voz deben trabajar en un entorno ruidoso. Esto quiere decir que para este tipo de sistemas ciertas capacidades del oído humano podrían ser deseables.

Se han desarrollado bastantes métodos de reducción de ruido que intentan suprimir los componentes de señal correspondientes a 'ruido' y realzar los componentes de voz explotando sus respectivas características, en un intento por imitar las habilidades del oído humano. Por ejemplo, en las aplicaciones de supresión de ruido espectral para enfatizar la voz, se supone que la señal de interés es la voz con sus típicas pausas, mientras que la señal de ruido es considerada como estacionaria e ininterrumpida. De esta forma es posible estimar el ruido espectral durante las pausas de la voz y posteriormente substraerlo del espectro de la voz contaminada con ruido para obtener la señal 'limpia' de voz.

Un punto de vista alternativo es considerar la escena acústica como si hubiera sido generada por fuentes de señal emitiendo simultáneamente en diferentes posiciones del espacio. Mediante la descomposición de el sonido grabado en sus componentes correspondientes a las diferentes fuentes, y posteriormente eligiendo las fuentes particulares que nos interesan, es posible suprimir las fuentes indeseadas de ruido. Sin embargo, operando de esta forma sólo se hace una distinción entre voz y ruido en el último paso, cuando seleccionamos la fuente de interés. En el primer y presumiblemente más difícil paso de descomponer la situación acústica en las fuentes de que consta, la noción de separación física es suficiente.

La separación ciega de fuentes (Blind Source Separation, BSS) constituye un enfoque que intenta conseguir esta descomposición partiendo de conocimientos a priori tan escasos como sea posible, de ahí el término 'ciega'. La formulación del problema como separación de las fuentes apunta a que será útil en muchas más posibles aplicaciones que simplemente en las de reducción de ruido, puesto que hay muchas situaciones en las que no es posible medir señales 'puras', correspondientes a una única fuente de forma independiente. En vez de eso se tendrá acceso a medidas de una superposición de bastantes fuentes a la vez. Ejemplos de esto que estamos diciendo podemos encontrarlos en el área de las comunicaciones inalámbricas cuando un equipo recibe señales procedentes de múltiples teléfonos móviles, el análisis de señales biomédicas obtenidas de un electroencefalograma, etc.

También se puede intentar llevar a cabo esta descomposición en aplicaciones donde no hay más información a priori que la de que los datos medidos están compuestos de partes mutuamente independientes, para facilitar análisis relacionados con las señales. Ejemplos de estas aplicaciones son el análisis de pequeños trozos de imágenes, segmentos cortos de sonido y datos financieros. Existen diversas opciones para definir cuándo las fuentes son mutuamente diferentes o independientes, y eso lo veremos en secciones posteriores.

En este texto vamos a centrarnos en una aplicación de entre todas las que hemos citado, que será la de llevar a cabo la separación ciega de señales de voz. Este problema suele llamarse en la literatura "cocktail-party problem", debido a que es similar a la situación que sucede en las fiestas en las que hay mucho ruido de fondo debido a música y otras personas hablando y tenemos que concentrarnos en una sola conversación.

1.3 Estructura del proyecto.

Este proyecto consta de dos partes diferentes aunque fuertemente relacionadas, por un lado el estudio teórico de las técnicas de separación que han sido propuestas para llevar a cabo la tarea de la separación ciega de fuentes, así como las herramientas teóricas y matemáticas necesarias para el tratamiento de las señales de voz, y por otro el desarrollo en Matlab de los algoritmos que realizan esta separación, comprobando su correcto funcionamiento y observando sus posibles limitaciones.

Concretando un poco, en el apartado práctico hemos implementado tres métodos diferentes de separación, aplicables cada uno de ellos a situaciones acústicas diferentes. El punto que tienen en común todos ellos es que los algoritmos de separación trabajan en el dominio de la frecuencia, concretamente en el dominio tiempo-frecuencia. Acudir a este dominio tiene ventajas que proceden de la propia naturaleza de la señal de voz, y que hacen imposible o muy difícil la separación de las fuentes operando sólo en el dominio temporal.

Pasamos a desglosar brevemente los contenidos del proyecto por capítulos. El segundo capítulo está dedicado al análisis de componentes principales (PCA), que si bien por sí solo no nos sería de gran utilidad para nuestro objetivo (elimina componentes prescindibles pero no separa los que quedan), es un paso interesante que se aplica como parte del preprocesado que le hacemos a los datos para llevar a cabo la separación. Lo que consigue PCA es reducir la dimensionalidad del problema en muchos casos, siendo útil por ejemplo en la reducción de ruido. Presentaremos tanto los conceptos como la forma de operar para implementar PCA, basándonos en estadísticos de segundo orden. También se habla en esta sección de los fundamentos de blanqueo y ortogonalización, que serán necesarios más adelante.

El tercer capítulo está dedicado la herramienta fundamental en torno a la que gira BSS, es el análisis de componentes independientes (ICA). Explicaremos con detalle los requisitos que requiere, fundamentalmente la no-gaussianidad e independencia de las fuentes, y detallaremos tanto los criterios como la

optimización de los diferentes algoritmos que pueden usarse para conseguir separar fuentes de muy diversa naturaleza a partir de observaciones de las mismas. Esta es la parte más importante del proyecto desde el punto de vista teórico, ya que en las secciones posteriores nos centramos ya en unos aspectos algo más enfocados a la puesta en práctica de la separación de mezclas de voz.

El capítulo 4 se centra exclusivamente en la transformada STFT, que es la que usaremos en todos los métodos de separación para pasar las señales al dominio de la frecuencia, y a partir de esta transformación de los datos se realizará la búsqueda de los componentes independientes. Veremos que ésta tiene forma matricial, ya que es una transformada de Fourier diferente a las más conocidas, de esta forma toma valores en dos dimensiones: espacio y tiempo. Presentaremos las expresiones matemáticas y las interpretaciones tanto de la transformada como de su inversa, que permite reconstruir las señales al dominio del tiempo. También describiremos en esa sección las características más importantes de la señal de voz, lo que nos ayudará a entender el por qué de la elección de la STFT como herramienta de trabajo fundamental en nuestros algoritmos.

En el quinto capítulo presentaremos los modelos matemáticos de mezcla de las señales de voz que adoptaremos en los sucesivos capítulos. Veremos fundamentalmente dos, tanto en el dominio del tiempo como en el dominio de la frecuencia. Uno describe la situación de una forma más general y por lo tanto es más difícil de tratar pero se adapta mejor a una gran variedad de situaciones acústicas. Este tiene como elemento distintivo que usa la convolución para describir la interacción de las señales con el entorno, debido a que las señales de audio se propagan por el espacio con una velocidad finita, sufriendo retardos y atenuaciones. El segundo modelo importante que veremos es un caso particular del primero, ya que sigue siendo realista pero describe la mezcla de voces en un entorno anecoico, esto es, sin ecos, como puede ser el espacio libre.

El capítulo 6 es el que hemos dedicado a hablar de los tres algoritmos prácticos que hemos desarrollado para separar las señales de voz a partir de las observaciones de mezclas de las mismas. El primero se basa en tratar de estimar el ángulo con que llegan los frentes de onda de las diferentes señales fuente, usando estos datos para extraer una estimación de dichas señales a partir de las grabaciones. El segundo es para el espacio libre e intenta estimar una serie de parámetros que nos permitirán estimar el sistema de mezcla, para después invertirlo. El tercero es el que a priori es más robusto de los tres, ya que no simplifica la situación en exceso. En este se ejecuta la separación mediante ICA en las diferentes subbandas de frecuencia de las señales por separado, teniendo en cuenta posteriormente los resultados de todos los problemas individuales para reconstruir las señales en el dominio del tiempo. La descomposición del objetivo total en pequeños problemas independientes tendrá una serie de connotaciones que resultarán problemáticas para poner en común los resultados obtenidos en cada banda de frecuencia. Hablaremos con detalle de esto y de las soluciones adoptadas.

El séptimo capítulo romperá un poco con el enfoque teórico de toda la información tratada hasta ese momento y se dedica a mostrar los resultados obtenidos en la simulación de los métodos de separación desarrollados en

Matlab. Presentaremos varios ejemplos para ser capaces de comprobar que todo lo contado hasta ahora tiene utilidad práctica, exponiendo los pros y los contras de cada técnica de separación y hablando también de los resultados fallidos que se han encontrado en las simulaciones, buscando además una explicación.

Por último, se exponen en el capítulo 8 las conclusiones que hemos podido extraer de todos los experimentos realizados en el desarrollo del proyecto. También esbozamos las líneas de investigación sobre las que se puede seguir trabajando en el futuro en campos relacionados con el que tratamos en este proyecto.

Capítulo 2

Análisis de Componentes Principales, blanqueo y ortogonalización

2.1 Introducción.

Un problema central en el análisis de datos multivariantes es la reducción de la dimensionalidad: si es posible describir con precisión los valores de n variables por un pequeño subconjunto $m < n$ de ellas, se habrá reducido la dimensión del problema a costa de una pequeña pérdida de información.

El análisis de componentes principales tiene este objetivo: dadas T observaciones de n variables, se analiza si es posible representar adecuadamente esta información con un número menor de variables construidas como combinaciones lineales de las originales. Por ejemplo, con variables de alta dependencia, es frecuente que un pequeño número de nuevas variables (menos del 20 por ciento de las originales) expliquen la mayor parte (más del 80 por ciento) de la variabilidad original.

La técnica de componentes principales es debida a Hotelling (1933), aunque sus orígenes se encuentran en los ajustes ortogonales por mínimos cuadrados introducidos por K. Pearson (1901). La utilidad de PCA es doble:

1. Permite representar óptimamente en un espacio de dimensión pequeña observaciones de un espacio general n-dimensional. En este sentido, el análisis de componentes principales es el primer paso para identificar las posibles variables *latentes*, o no observadas que generan los datos.

2. Permite transformar las variables originales, en general correladas, en nuevas variables incorreladas, facilitando la interpretación de los datos.

2.2 Representación lineal de datos multidimensionales.

Supongamos que tenemos unos datos que consisten en un número de variables que hemos observado de forma conjunta. Denotaremos el número de variables por n y el número de observaciones por T. Podemos representar los datos como $x_i(t)$, donde los índices toman los valores $i = 1,...,n$ y $t = 1,...,T$. De momento no hay restricción en cuanto a las dimensiones n y T.

Una formulación muy general del problema puede ser enunciada así: se trata de pasar de un espacio *n*-dimensional a otro *m*-dimensional tal que las variables transformadas contengan información sobre los datos que nos interesan y que de otro modo están ocultos en el conjunto de medidas. Así, las variables transformadas deberían ser los *factores* subyacentes o *componentes* que describen la estructura esencial de los datos. Es deseable que estos componentes se correspondan con alguna causa física involucrada en el proceso de generación de los datos.

Si consideramos funciones lineales, cada componente, llamémoslo y_i, puede expresarse como una combinación lineal de las variables observadas:

$$y_i(t) = \sum_j w_{ij} x_j(t), \; i = 1,...,m, \; j = 1,...,n \tag{2.1}$$

donde los w_{ij} son los coeficientes que definen la representación. Así, el problema puede replantearse como la determinación de los coeficientes w_{ij}. Usando álgebra lineal, podemos expresar la transformación lineal de la ecuación (2.1) como una multiplicación de matrices. Reuniendo los coefiecientes w_{ij} en una matriz **W**, la ecuación queda como:

$$\begin{bmatrix} y_1(t) \\ y_2(t) \\ ... \\ y_m(t) \end{bmatrix} = \mathbf{W} \begin{bmatrix} x_1(t) \\ x_2(t) \\ ... \\ x_n(t) \end{bmatrix} \tag{2.2}$$

Una aproximación estadística básica consiste en considerar los $x_i(t)$ como un conjunto de T realizaciones de n variables aleatorias. Así cada conjunto $x_i(t)$, $t = 1,...T$ es un conjunto de muestras de una variable aleatoria, que denotaremos por x_i.

Un principio estadístico para elegir la matriz **W** es limitar el número de componentes y_i para que sea pequeño, y determinar **W** de forma que y_i contenga tanta información como sea posible. Esto nos lleva a una familia de técnicas llamadas análisis de componentes principales (PCA) o análisis de factores.

2.3 Análisis de componentes principales (PCA).

El análisis de componentes principales está relacionado estrechamente con algunas de las técnicas clásicas usadas en análisis de datos estadísticos y compresión de datos.

Dado un conjunto multidimensional de medidas, el objetivo será encontrar un conjunto más pequeño de variables con menor redundancia, lo que nos

aportaría una mejor representación de los datos. En PCA la redundancia es medida por correlaciones entre los datos. Usar sólo las correlaciones como se hace en PCA tiene la ventaja de que el análisis puede basarse exclusivamente en estadísticos de segundo orden.

Para motivar el problema, usaremos un ejemplo práctico. Supongamos que hemos colocado varios sensores, digamos *n*, que han tomado medidas simultáneamente de *m* señales que nos interesan. No importa ahora la naturaleza de estas señales, pues no cambiará ni el planteamiento ni la solución adoptada. Bien, de esta forma las señales observadas serán una mezcla de las señales que realmente nos interesan más un cierto ruido que no podemos evitar que aparezca. De esta forma el sistema tomará la siguiente forma:

$$\begin{bmatrix} x_1(t) \\ \vdots \\ x_n(t) \end{bmatrix} = \begin{bmatrix} a_{11} & \cdots & a_{1m} \\ \vdots & \ddots & \vdots \\ a_{n1} & \cdots & a_{nm} \end{bmatrix} \begin{bmatrix} s_1(t) \\ \vdots \\ s_n(t) \end{bmatrix} + \begin{bmatrix} e_1(t) \\ \vdots \\ e_n(t) \end{bmatrix} \tag{2.3}$$

donde $n>m$, $x_i(t)$ son las señales tomadas por los sensores, $s_i(t)$ son las señales que nos interesan y $e_i(t)$ son las señales de ruido. Expresando esto de forma vectorial para mayor comodidad, nos queda:

$$\mathbf{x} = \mathbf{A}\mathbf{s} + \mathbf{e} \tag{2.4}$$

El objetivo entonces sería hacer alguna transformación al vector **x** de forma que redujésemos su dimensión a *m*, ya que sólo nos interesa tener esos valores en función de las *m* señales que nos interesan, eliminando tanto como sea posible los componentes correspondientes al ruido. Esto equivale por lo tanto a eliminar la redundancia existente en las *n* grabaciones de los sensores y quedarnos sólo con la información verdaderamente importante. Intentaremos explicar a continuación cómo conseguiremos esta meta haciendo uso del análisis de componentes principales.

El punto de partida entonces de PCA es un vector aleatorio **x** con *n* elementos. Los vectores que se usarán a continuación tienen el mismo significado y siguen la misma notación que sus homónimos descritos anteriormente. Tenemos disponible una muestra **x**(1),…,**x**(*T*) de este vector. En PCA no se supone que la densidad de probabilidad de los vectores es gaussiana, no obstante los estadísticos de primer y segundo orden son conocidos o pueden estimarse a través de las muestras. Tampoco suponemos ningún modelo generativo para el vector **x**. Es esencial en PCA que los elementos de **x** estén mutuamente correlados, y así haya cierta redundancia en **x**, haciendo posible la compresión. Es destacable el hecho de que si los elementos son independientes, no se puede conseguir nada con PCA.

En PCA, lo primero que se hace es centrar el vector restándole su media.

$$\mathbf{x} \leftarrow \mathbf{x} - E\{\mathbf{x}\} \tag{2.5}$$

La media en la práctica se estima a partir de las muestras disponibles **x**(1),…, **x**(*T*). Vamos a suponer en todo lo siguiente que ya se ha hecho el centrado, y por lo tanto, $E\{\mathbf{x}\} = 0$. A continuación, **x** será transformado linealmente en otro vector **y** con *m* elementos, *m*<*n*, de forma que se elimina la redundancia introducida por las correlaciones. Esto se hace encontrando un sistema coordenado ortogonal tal que los elementos de **x** proyectados en las nuevas coordenadas estén incorrelados. Al mismo tiempo, las varianzas de las proyecciones de **x** en los nuevos ejes coordenados son maximizadas de forma que el primer eje corresponde a la varianza máxima, el segundo eje corresponde a la varianza máxima en la dirección ortogonal al primer eje, y así sucesivamente.

A continuación veremos diferentes criterios para hallar la solución PCA.

2.3.1 Interpretación de PCA como maximización de la varianza.

En términos matemáticos, consideramos una combinación lineal

$$y_1 = \sum_{k=1}^{n} w_{k1} x_k = \mathbf{w}_1^T \mathbf{x} \tag{2.6}$$

de los elementos $x_1,\ldots,x_n$ del vector **x**. Los $w_{11},\ldots,w_{n1}$ son coeficientes escalares o pesos, elementos a su vez de un vector *n*-dimensional $\mathbf{w}_1$, con traspuesta $\mathbf{w}_1^T$.

El factor y_1 es llamado componente principal primero de **x** si la varianza de y_1 es máxima. Ese es el criterio de máxima varianza. Debido a que la varianza depende tanto de de la norma como de la orientación del vector de peso $\mathbf{w}_1$, y crece sin límite según crece la norma, tenemos que imponer la restricción de que la norma de $\mathbf{w}_1$ sea constante, normalmente igual a la unidad. Así evitamos el crecimiento desmesurado de la varianza. Entonces lo que haremos será buscar un vector peso $\mathbf{w}_1$ que maximice este criterio PCA:

$$J_1^{PCA}(\mathbf{w}_1) = E\{y_1^2\} = E\left\{\left(\mathbf{w}_1^T\mathbf{x}\right)^2\right\} = \mathbf{w}_1^T E\{\mathbf{x}\mathbf{x}^T\}\mathbf{w}_1 = \mathbf{w}_1^T\mathbf{C}_\mathbf{X}\mathbf{w}_1 \text{ tal que } \|\mathbf{w}_1\| = 1 \tag{2.7}$$

donde la norma de $\mathbf{w}_1$ se calcula de la forma habitual $\|\mathbf{w}_1\| = \left(\mathbf{w}_1\mathbf{w}_1^T\right)^{1/2}$. $\mathbf{C}_\mathbf{X}$ es la matriz de covarianzas $n \times n$ de **x**, calculada como $\mathbf{C}_\mathbf{X} = E\{\mathbf{x}\mathbf{x}^T\}$.

La solución al problema PCA se da en términos de los autovectores unitarios $e_1,\ldots,e_n$ de la matriz $\mathbf{C}_\mathbf{X}$. Para que no haya ambigüedades, la ordenación de los autovectores se corresponde con la ordenación creciente de sus respectivos autovalores, es decir, $d_1 \geq d_2 \geq \cdots \geq d_n$. La solución que maximiza la expresión (2.7) es $\mathbf{w}_1 = \mathbf{e}_1$, por lo tanto el primer componente principal de **x** será $y_1 = \mathbf{e}_1^T\mathbf{x}$. El criterio J_1^{PCA} de la expresión (2.7) puede generalizarse para *m* componentes principales, siendo *m* un número comprendido entre 1 y *n*. Denotando al *m*-ésimo componente principal por $y_m = \mathbf{w}_m^T\mathbf{x}$, y siendo $\mathbf{w}_m$ el correspondiente

vector peso de norma unidad, la varianza de y_m es maximizada bajo la restricción de que y_m esté incorrelada con todos los componentes principales encontrados previamente, es decir:

$$E\{y_m y_k\} = 0,\ k < m \tag{2.8}$$

Los componentes principales también tienen media cero porque $E\{y_m\} = \mathbf{w}_m^T E\{\mathbf{x}\} = 0$. De la condición (2.8) sigue:

$$E\{y_m y_k\} = E\{(\mathbf{w}_m^T x)(\mathbf{w}_k^T x)\} = \mathbf{w}_m^T \mathbf{C}_\mathbf{x} \mathbf{w}_k = 0 \tag{2.9}$$

Para el segundo componente principal, la condición es:

$$\mathbf{w}_2^T \mathbf{C} \mathbf{w}_1 = d_1 \mathbf{w}_2^T \mathbf{e}_1 = 0 \tag{2.10}$$

porque ya sabemos que $\mathbf{w}_1 = \mathbf{e}_1$. Estamos así buscando la máxima varianza $E\{y_2^2\} = E\{(\mathbf{w}_2^T \mathbf{x})^2\}$ en el subespacio ortogonal para el primer autovector de $\mathbf{C}_\mathbf{x}$. La solución viene dada por $\mathbf{w}_2 = \mathbf{e}_2$. Si seguimos iterando de esta manera, llegamos a la conclusión de que $\mathbf{w}_k = \mathbf{e}_k$. Así, el componente principal *k*-ésimo será $y_k = \mathbf{e}_k^T \mathbf{x}$.

Exactamente el mismo resultado para los w_i se obtiene si las varianzas de y_i son maximizadas bajo la restricción de que los vectores componentes principales son ortonormales, esto es, $\mathbf{w}_i^T \mathbf{w}_j = \delta_{ij}$. Eso no lo demostraremos.

2.3.2 Interpretación de PCA como minimización del error cuadrático medio.

En el punto anterior, habíamos definido los componentes principales como sumas ponderadas de los elementos de **x** con varianza máxima bajo las restricciones de que los pesos se normalicen y los componentes principales estén incorrelados unos con otros. Resulta que esto está fuertemente relacionado con la minimización del error cuadrático medio de **x**, que es otra forma de plantear el problema PCA. Ahora buscaremos un conjunto de *m* vectores base ortonormales, abarcando un subespacio *m*-dimensional, tal que el error cuadrático medio entre **x** y su proyección en el subespacio sea mínimo. Denotando de nuevo los vectores base por $\mathbf{w}_1, \ldots, \mathbf{w}_m$, para los que supondremos $\mathbf{w}_i^T \mathbf{w}_j = \delta_{ij}$, la proyección de **x** sobre el subespacio abarcado por ellos es $\sum_{i=1}^{m} (\mathbf{w}_i^T \mathbf{x}) \mathbf{w}_i$.

El criterio del error cuadrático medio (MSE), que será minimizado por las bases ortonormales $\mathbf{w}_1, \ldots, \mathbf{w}_m$, es:

$$J_{MSE}^{PCA} = E\left\{\left\|\mathbf{x} - \sum_{i=1}^{m}\left(\mathbf{w}_i^T\mathbf{x}\right)\mathbf{w}_i\right\|^2\right\} \tag{2.11}$$

Es fácil demostrar que debido a la ortogonalidad de los vectores $\mathbf{w}_i$, este criterio puede ser rescrito como:

$$J_{MSE}^{PCA} = E\left\{\|\mathbf{x}\|^2\right\} - E\left\{\sum_{j=1}^{m}\left(\mathbf{w}_j^T\mathbf{x}\right)^2\right\} = trace(\mathbf{C}_\mathbf{X}) - \sum_{j=1}^{m}\mathbf{w}_j^T\mathbf{C}_\mathbf{X}\mathbf{w}_j \tag{2.12}$$

Puede demostrarse también que el mínimo de (2.12) bajo la condición de ortonormalidad de los $\mathbf{w}_i$ viene dado por alguna base ortonormal del subespacio PCA abarcado por los m primeros autovectores $\mathbf{e}_1,\ldots,\mathbf{e}_m$. Sin embargo, el criterio no especifica las bases de este subespacio. Cualquier base ortonormal del subespacio nos dará la misma compresión óptima. Mientras que esta ambigüedad puede verse como una desventaja, debería caerse en la cuenta de que podría haber algún otro criterio por el cual se prefiera una determinada base en el subespacio PCA frente a las demás.

Puede demostrarse que el valor del error cuadrático medio mínimo de (2.11) es

$$J_{MSE}^{PCA} = \sum_{i=m+1}^{n} d_i \tag{2.13}$$

es decir, la suma de los autovalores correspondientes a los autovectores descartados $\mathbf{e}_{m+1},\ldots,\mathbf{e}_n$. Si cambiamos la restricción de ortonormalidad por $\mathbf{w}_j^T\mathbf{w}_k = \omega_k\delta_{jk}$, donde todos los números ω_k son positivos y diferentes, entonces el problema del error cuadrático medio tendrá una solución única dada por los autovectores escalados.

2.3.3 Elección del número de componentes principales.

Del resultado de que los componentes principales vectores base $\mathbf{w}_i$ son los autovalores $\mathbf{e}_i$ de $\mathbf{C}_\mathbf{X}$, se deduce que

$$E\left\{y_m^2\right\} = E\left\{\mathbf{e}_m^T\mathbf{x}\mathbf{x}^T\mathbf{e}_m\right\} = \mathbf{e}_m^T\mathbf{C}_\mathbf{X}\mathbf{e}_m = d_m \tag{2.14}$$

Las varianzas de los componentes principales vienen dadas directamente entonces por los autovalores de $\mathbf{C}_\mathbf{X}$. Reseñar que, por el hecho de que los componentes principales tengan media cero, un autovalor de pequeña varianza d_m indica que el valor del correspondiente componente principal y_m es cercano a cero.

Una aplicación importante de PCA es la compresión de datos. Los vectores **x** del conjunto de datos original (que ya han sido centrados restándoles su media) son aproximados por la expansión PCA truncada

$$\hat{\mathbf{x}} = \sum_{i=1}^{m} y_i \mathbf{e}_i \tag{2.15}$$

Sabemos de (2.13) que el error cuadrático medio $E\left\{\|\mathbf{x} - \hat{\mathbf{x}}\|^2\right\}$ es igual a $\sum_{i=m+1}^{n} d_i$. Como los autovalores son todos positivos, el error decrece si se van incluyendo más términos en (2.15), hasta que el error se hace cero cuando $m=n$ o todos los componentes principales hayan sido incluidos. Un problema muy importante en la práctica es cómo encontrar el valor de m en (2.15), porque hay una compensación entre el error y la cantidad de datos que se necesitan para la expansión.

Si conocemos los autovalores, a menudo puede usarse la condición (2.14) para determinar el número de componentes principales m. La secuencia de autovalores $d_1, d_2, \cdots, d_n$ de una matriz de covarianzas para medidas de datos provenientes de medidas reales normalmente es decreciente, y por consiguiente es posible fijar un límite por debajo del cual los autovalores, de hecho componentes principales, son insignificantes. Fijando ese límite podemos saber cuántos componentes principales usar.

A veces el umbral puede determinarse a partir de alguna información a priori de los vectores **x**. Por ejemplo, suponiendo que **x** sigue un modelo de señal más ruido como el siguiente:

$$\mathbf{x} = \sum_{i=1}^{m} \mathbf{s}_i a_i + \mathbf{n} \tag{2.16}$$

Los vectores $\mathbf{s}_i$ son fijos y los coeficientes a_i son número aleatorios incorrelados y de media cero. Podemos suponer que sus varianzas han sido absorbidas en los vectores $\mathbf{s}_i$ y por lo tanto tienen varianza unidad. El término **n** es ruido blanco, para el que $E\left\{\mathbf{n}\mathbf{n}^T\right\} = \sigma^2\mathbf{I}$. Entonces los vectores $\mathbf{s}_i$ representan el subespacio de señal, que tiene una dimensionalidad más baja que el espacio completo de vectores **x**. El subespacio ortogonal al subespacio de señal representa a puro ruido y lo llamaremos subespacio de ruido.

Es fácil mostrar que en este caso la matriz de covarianza de **x** tiene una forma especial:

$$\mathbf{C}_\mathbf{X} = \sum_{i=1}^{m} \mathbf{s}_i \mathbf{s}_i^T + \sigma^2\mathbf{I} \tag{2.17}$$

Los autovalores son ahora los autovalores de $\sum_{i=1}^{m} \mathbf{s}_i \mathbf{s}_i^T$ sumados a la constante σ^2. Pero la matriz $\sum_{i=1}^{m} \mathbf{s}_i \mathbf{s}_i^T$ tiene como máximo m autovalores no nulos, y estos se corresponden con los autovectores que generan el subespacio de señal.

Cuando calculamos los autovalores de $\mathbf{C_x}$, los primeros *m* forman una secuencia decreciente y el resto son pequeñas constantes, iguales a σ^2:

$$d_1 > d_2 > \cdots > d_m > d_{m+1} = d_{m+1} = \cdots = d_n = \sigma^2 \qquad (2.18)$$

Normalmente es posible detectar dónde los autovalores se vuelven constantes, y colocando un umbral en ese índice, *m*, eliminamos los autovalores y autovectores correspondientes a ruido puro y sólo nos quedamos con la parte de señal.

2.3.4 Cálculo completo de PCA.

Para usar la solución en forma cerrada calculada en apartados anteriores $\mathbf{w}_i = \mathbf{e}_i$ para los vectores base PCA, debemos conocer los autovectores de la matriz de covarianzas $\mathbf{C_x}$. En los casos prácticos convencionales en los que se usa PCA, hay una muestra suficientemente grande de vectores **x** disponible, a partir de los cuales podemos estimar la media y la matriz de covarianzas $\mathbf{C_x}$ por métodos estándar. Así cuando se soluciona el problema de los autovalores y los autovectores para $\mathbf{C_x}$, tenemos también la estima para $\mathbf{e}_1$.

Afortunadamente, hay bastantes métodos numéricos eficientes disponibles para encontrar los autovectores, sin embargo, no siempre es factible hallarlos por métodos numéricos estándar. En aplicaciones on-line de compresión de datos como imágenes o audio, los datos llegan a una velocidad muy alta, y no siempre es posible estimar la matriz de covarianzas y solucionar el problema de los autovalores y autovectores una vez para todos. Por un lado está el motivo computacional: la solución de los autovectores es numéricamente exigente si la dimensión *n* es grande y la tasa de muestreo elevada, y por otro el hecho de que la matriz $\mathbf{C_x}$ podría no ser estacionaria, debido a fluctuaciones de los estadísticos en la secuencia de muestras **x**(*t*), de forma que la estima tendría que ir actualizándose. Por consiguiente, la solución PCA a menudo es reemplazada por transformaciones subóptimas no adaptativas como la transformada discreta del coseno.

Hay también otras alternativas, como derivar algoritmos de optimización del gradiente ascendente u otros métodos on-line. Estos algoritmos deben converger a las soluciones del problema, es decir, a los autovectores. La ventaja que tienen estos métodos es que trabajan on-line, usando cada vector de entrada **x**(*t*) una vez y haciendo actualizaciones en cada iteración, pero sin calcular para nada la matriz de covarianzas. En el apéndice 2 se describe la esencia del citado método del gradiente.

2.4 Blanqueo.

Veremos más adelante que el problema ICA se simplifica bastante si los vectores mezcla observados son primero blanqueados. Ahora hablaremos brevemente de en qué consiste el citado blanqueo.

Un vector de media cero $\mathbf{z} = (z_1,\ldots,z_n)^T$ se dice que es *blanco* si sus elementos z_i son incorrelados y tienen varianza unidad:

$$E\{z_i z_j\} = \delta_{ij} \tag{2.19}$$

En términos de la matriz de covarianza, esto significa que $E\{\mathbf{z}\mathbf{z}^T\} = \mathbf{I}$. Un ejemplo simple y fácil de entender sería el del ruido blanco, donde los elementos z_i serían las intensidades del ruido en cada instante de tiempo $i = 1,2,\ldots$ y no habría correlaciones temporales en el proceso ruidoso. El término *blanco* procede del hecho de que el espectro densidad de potencia del ruido blanco es constante durante todas las frecuencias, como ya sabemos.

El blanqueo puede hacerse mediante una operación lineal. Entonces podemos definir el problema del blanqueo como: dado un vector aleatorio **x** con *n* elementos, queremos encontrar una transformación lineal que nos lo convierta en otro vector **z** que sea blanco, es decir, encontrar **V** tal que **z=Vx** sea blanco.

El problema tiene una solución sencilla en términos de la expansión PCA. Nombraremos $\mathbf{E}=(\mathbf{e}_1\ldots\mathbf{e}_n)$ a la matriz cuyas columnas son los autovectores (ya con norma unidad) de la matriz de covarianzas $\mathbf{C_X} = E\{\mathbf{x}\mathbf{x}^T\}$. Éstos pueden calcularse a partir de una muestra de los vectores **x** o directamente por una de las reglas de aprendizaje on-line de PCA. Llamemos $\mathbf{D} = diag(d_1,\ldots,d_n)$ a la matriz diagonal con los autovalores de $\mathbf{C_X}$. Entonces una transformación lineal de blanqueo vendrá dada por

$$\mathbf{V} = \mathbf{E}\mathbf{D}^{-1/2}\mathbf{E}^T \tag{2.20}$$

Esta matriz existirá siempre que los autovalores d_i sean positivos, lo que sucederá en los casos prácticos en general. Es fácil demostrar que la matriz **V** de (2.20) es una transformación de blanqueo. Rescribiendo $\mathbf{C_X}$ en función de sus matrices de autovalores y autovectores **E** y **D** como $\mathbf{C_X} = \mathbf{E}\mathbf{D}\mathbf{E}^T$, siendo **E** una matriz ortogonal que satisface $\mathbf{E}^T\mathbf{E} = \mathbf{E}\mathbf{E}^T = \mathbf{I}$, llegamos a la siguiente conclusión:

$$E\{\mathbf{z}\mathbf{z}^T\} = \mathbf{V}\mathbf{E}\{\mathbf{x}\mathbf{x}^T\}\mathbf{V}^T = \mathbf{D}^{-1/2}\mathbf{E}^T\mathbf{E}\mathbf{D}\mathbf{E}^T\mathbf{E}\mathbf{D}^{-1/2} = \mathbf{I} \tag{2.21}$$

La covarianza de **z** es la matriz identidad, por lo tanto **z** es blanco.

Entonces llegamos a la conclusión de que el operador lineal **V** de (2.20) es la única matriz de blanqueo. De todas forma, no es difícil observar que cualquier matriz **UV**, donde **U** sea una matriz ortogonal, también es una matriz de blanqueo. Esto es debido a que del hecho de que **z=UVx** se sigue:

$$E\{\mathbf{z}\mathbf{z}^T\} = \mathbf{U}\mathbf{V}\mathbf{E}\{\mathbf{x}\mathbf{x}^T\}\mathbf{V}^T\mathbf{U}^T = \mathbf{U}\mathbf{I}\mathbf{U}^T = \mathbf{I} \tag{2.22}$$

2.5 Ortogonalización.

En algunos algoritmos ICA (ver capítulo 3) y PCA, sabemos que teóricamente los vectores solución (vectores base) son ortogonales u ortonormales, sin embargo los algoritmos iterativos no siempre producen esta ortogonalidad automáticamente. En esos casos es necesario ortogonalizar los vectores después de cada paso de la iteración, o al menos en momentos del algoritmo en los que sea conveniente. Para conseguir esto se puede acudir a varios métodos bien conocidos de ortogonalización como el de Gram-Schmidt. Puede encontrarse información detallada de este método en la mayoría de libros de álgebra lineal, como por ejemplo [Grossman95].

2.6 Conclusiones.

En este primer capítulo hemos visto una técnica para reducir la dimensión en problemas de datos donde tenemos varias variables. Para ello lo que hacemos es escoger un nuevo sistema de coordenadas para el conjunto original de datos en el cual la varianza de mayor valore del conjunto de datos es capturada en el primer eje (primer componente principal), a partir de este elegimos un segundo eje ortogonal al primero que capture la segunda varianza más grande, y así sucesivamente. La ventaja que tiene usar esta técnica es que al reducir la dimensionalidad de los datos conservamos las características del conjunto de datos que contribuyen más a la varianza, es decir, que no perdemos la información más importante, sacrificando solamente aquella información que aporta menos al error cuadrático medio o que incluso es completamente prescindible. Para ello hacemos uso de los autovalores y autovectores de la matriz de correlación de los datos.

Para conseguir este objetivo, hemos visto dos criterios, el de maximización de la varianza que acabamos de nombrar, y el de minimización del error cuadrático medio, aunque ambos son equivalentes en su esencia y lo único que cambia es la forma de operar para llegar al resultado. A veces conocemos el número de componentes principales que nos van a interesar, pero en otras ocasiones esto no sucede, y para esos casos vimos cómo imponer un umbral que determinara cuando debíamos detenernos y despreciar los componentes sucesivos, determinando así la dimensión del conjunto de datos resultante. Los posibles componentes que se desprecian se correspondían con los valores más bajos de los autovalores.

Por último, hemos expuesto brevemente el concepto de blanqueo, que usaremos en los siguientes capítulos para conseguir que los elementos de un vector de datos sean incorrelados y de varianza unidad, así como el de ortogonalización de los vectores base de un subespacio.

Capítulo 3

Análisis de Componentes Independientes (ICA)

3.1 Introducción.

El análisis de componentes independientes (ICA) es una técnica estadística y computacional que se usa para revelar factores ocultos que subyacen en conjuntos de variables aleatorias, medidas o señales. En el modelo de generación de los datos, el sistema de mezcla es desconocido para nosotros y las variables latentes se suponen no gaussianas y mutuamente independientes, por eso las llamaremos componentes independientes de los datos observados. Estos componentes, también llamados fuentes o factores, serán encontrados por ICA sin usar ninguna información adicional.

ICA puede verse como una extensión del análisis de componentes principales (PCA) o análisis de factores, del que ya hemos hablado en el capítulo 2. Sin embargo, ICA es una técnica mucho más potente, capaz de encontrar los factores escondidos aun cuando los métodos clásicos fallan. Lo que distingue ICA de otros métodos es que busca componentes que son al mismo tiempo independientes y no gaussianos, pero ya presentaremos ampliamente estos detalles más adelante.

El origen de los datos analizados por ICA puede venir de multitud de aplicaciones diferentes, incluyendo, por citar algunas, imágenes digitales, bases de datos o medidas de la actividad eléctrica del cerebro humano.

La técnica de ICA es relativamente nueva. Se presentó por primera vez a principio de los años ochenta en el contexto del modelado de redes neurales. A mediados de los noventa, bastantes grupos de investigación introdujeron con éxito algunos algoritmos novedosos, acompañados de impresionantes demostraciones de su eficacia en problemas hasta entonces difíciles de solucionar. Durante los últimos veinte años se han publicado muchos artículos sobre ICA en los campos del tratamiento de señales, redes neurales artificiales, estadística, teoría de la información, etc.

Nos disponemos ahora a desglosar los aspectos fundamentales de esta técnica, presentando los conceptos básicos, aplicaciones y principios de estimación de ICA.

3.2 Separación Ciega de Fuentes.

3.2.1 Medidas de mezclas de señales desconocidas.

Consideremos una situación donde hay un conjunto de señales emitidas por alguna fuente física. Estas fuentes podrían ser, por ejemplo, personas hablando en la misma habitación o teléfonos móviles emitiendo sus ondas de radio. Supongamos también que hay varios sensores o receptores situados en diferentes posiciones, de modo que cada uno graba una mezcla de las señales originales con ligeras diferencias. Una situación como esta será nuestro punto de partida.

Para simplificar, hablaremos ahora de tres señales fuente, y también de tres señales observadas. Denotaremos por $x_1(t), x_2(t)$ y $x_3(t)$ las señales observadas, que son las amplitudes de las señales grabadas en el instante de tiempo *t*, y por $s_1(t), s_2(t)$ y $s_3(t)$ las señales originales. Los $x_i(t)$ son entonces sumas ponderadas de los $s_i(t)$, donde los coeficientes dependen de las distancias entre las fuentes y los sensores:

$$\begin{aligned} x_1(t) &= a_{11}s_1(t) + a_{12}s_2(t) + a_{13}s_3(t) \\ x_2(t) &= a_{21}s_1(t) + a_{22}s_2(t) + a_{23}s_3(t) \\ x_3(t) &= a_{31}s_1(t) + a_{32}s_2(t) + a_{33}s_3(t) \end{aligned} \qquad (3.1)$$

Los a_{ij} son coeficientes constantes que nos dan el peso de cada señal en la mezcla. Estos coeficientes se suponen desconocidos, puesto que no podemos saber sus valores sin conocer antes todas las propiedades del sistema físico de mezcla, lo que será extremadamente difícil en general. Las señales fuente s_i son desconocidas también, puesto que lo problemático es que no podemos grabarlas directa e independientemente, sino que tan sólo podemos llegar a conocerlas a través de las grabaciones de las mezclas, que es lo que estamos intentando hacer.

Para ilustrar lo que estamos diciendo, consideraremos que las señales observadas por tres sensores son las de la figura (3.1). Corresponden a tres mezclas lineales x_i de tres señales fuente desconocidas para nosotros. Aunque aparentemente no sean más que ruido, hay tres señales claramente estructuradas ocultas en esas mezclas.

Lo que nos gustaría entonces es encontrar las señales originales a partir de las mezclas $x_1(t), x_2(t)$ y $x_3(t)$. Ese es el problema de la separación ciega de fuentes (BSS). El término *ciega* aquí significa que tenemos muy poca información (si es que tenemos alguna) de las fuentes originales.

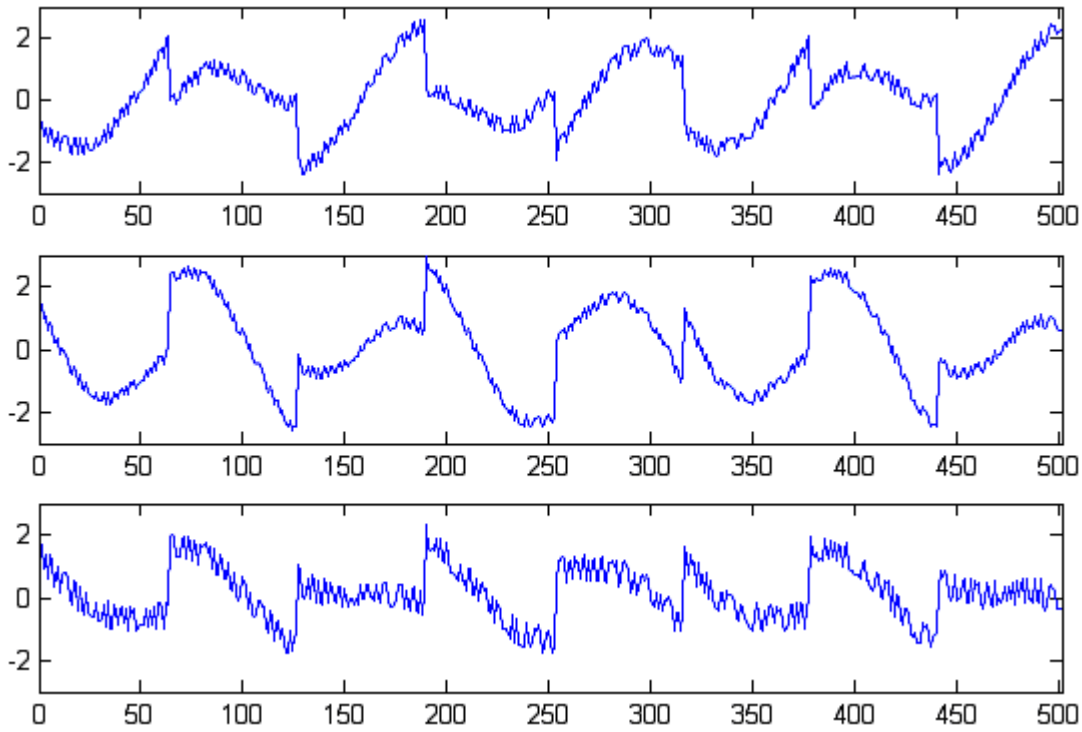


Figura 3.1 *Señales observadas provenientes de la mezcla de tres fuentes desconocidas*

Podemos suponer que los coeficientes de la mezcla, a_{ij}, son lo suficientemente diferentes para que la matriz que forman sea invertible. Entonces existe una matriz **W** con coeficientes w_{ij}, tales que pueden separar los s_i como:

$$\begin{aligned} s_1(t) &= w_{11}x_1(t) + w_{12}x_2(t) + w_{13}x_3(t) \\ s_2(t) &= w_{21}x_1(t) + w_{22}x_2(t) + w_{23}x_3(t) \\ s_3(t) &= w_{31}x_1(t) + w_{32}x_2(t) + w_{33}x_3(t) \end{aligned} \tag{3.2}$$

Podríamos encontrar la matriz **W** como la inversa de la matriz formada por los coeficientes de mezcla a_{ij}, si conociéramos esos coeficientes.

3.2.2 Separación de fuentes basada en la independencia.

La pregunta ahora es: ¿cómo podemos estimar los coeficientes w_{ij}? Queremos obtener un método general que trabaje en circunstancias variadas, por lo tanto, debemos usar propiedades estadísticas muy generales. Todo lo que podemos conocer son las señales x_1, x_2 y x_3, y queremos encontrar una matriz **W** que nos ayude a obtener una buena aproximación a las señales originales s_1, s_2 y s_3.

Una solución sorprendentemente simple al problema puede encontrarse considerando solamente la independencia estadística de las señales. De hecho,

si las señales son no gaussianas, es suficiente determinar los coeficientes w_{ij} de forma que las señales:

$$\begin{aligned} y_1(t) &= w_{11}x_1(t) + w_{12}x_2(t) + w_{13}x_3(t) \\ y_2(t) &= w_{21}x_1(t) + w_{22}x_2(t) + w_{23}x_3(t) \\ y_3(t) &= w_{31}x_1(t) + w_{32}x_2(t) + w_{33}x_3(t) \end{aligned} \tag{3.3}$$

sean estadísticamente independientes. Si las señales y_1, y_2 e y_3 son independientes, entonces son iguales a las señales originales s_1, s_2 y s_3, quizás multiplicadas por alguna constante escalar, pero eso tiene poca importancia.

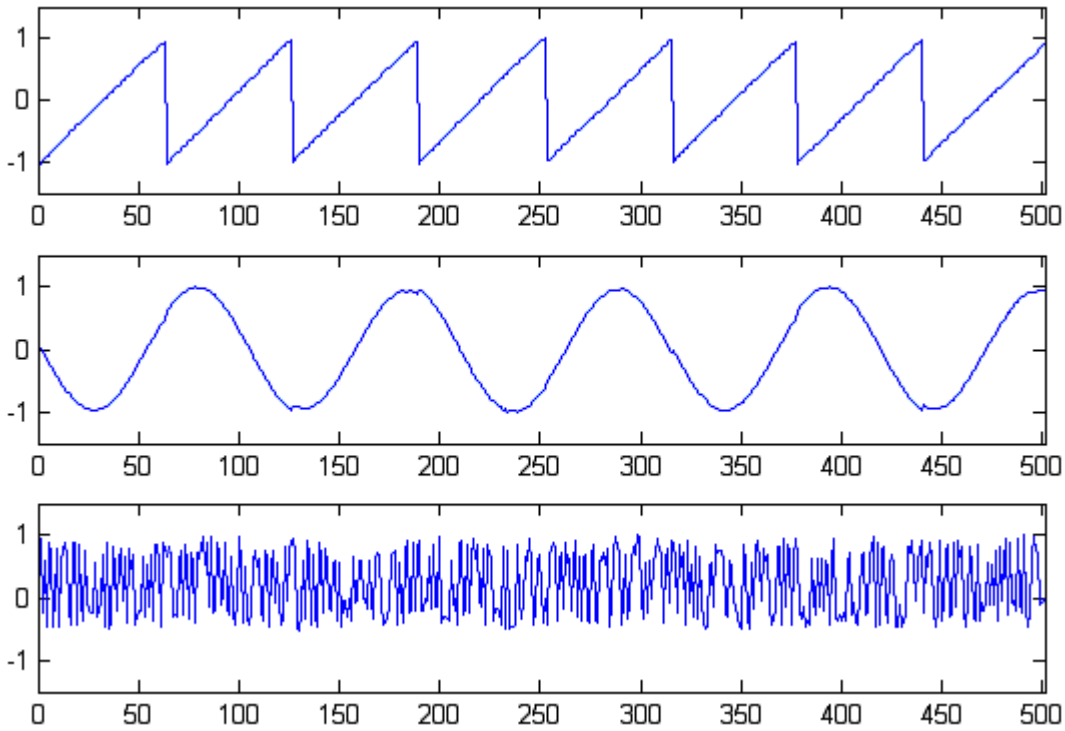


Figura 3.2 *Señales fuente estimadas usando exclusivamente las tres mezclas*

Tan sólo usando la información de la independencia estadística, podemos estimar la matriz de coeficientes **W** para hallar las señales originales.

Volviendo al ejemplo recientemente planteado, en la figura (3.2) se muestran las estimas de las tres señales fuente obtenidas usando el algoritmo ThinICA. Vemos que a partir de unos datos que parecían carentes de toda estructura, hemos sido capaces de estimar con bastante precisión las tres señales fuente, usando simplemente un algoritmo que busca la independencia de las señales estimadas. No mostramos aquí las señales fuente originales por el simple hecho de que son prácticamente idénticas a las que ha obtenido el algoritmo de separación, correspondiendo las mismas como puede observarse a un diente de sierra, un seno y un ruido aleatorio.

3.3 Descripción formal de ICA.

3.3.1 Definición.

El problema de la separación ciega de fuentes, como vimos antes, se reduce a la representación lineal de una mezcla cuyos componentes son estadísticamente independientes. En la práctica no podemos encontrar una representación en la que los componentes sean verdaderamente independientes, pero al menos sí podemos encontrar componentes que sean tan independientes como sea posible.

Esto no lleva a la siguiente definición de ICA:

Dado un conjunto de observaciones de variables aleatorias: $x_1(t), x_2(t), \ldots, x_n(t)$, donde t es normalmente el tiempo, vamos a suponer que han sido generadas como una mezcla lineal de componentes independientes:

$$\begin{bmatrix} x_1(t) \\ x_2(t) \\ \ldots \\ x_n(t) \end{bmatrix} = \mathbf{A} \begin{bmatrix} s_1(t) \\ s_2(t) \\ \ldots \\ s_m(t) \end{bmatrix} \tag{3.4}$$

siendo **A** una matriz desconocida. El análisis de componentes independientes consiste ahora en estimar tanto la matriz **A** como los $s_i(t)$, usando para ello sólo los $x_i(t)$. Es importante darse cuenta de que el número de componentes independientes s_i es igual al número de variables observadas, aunque ésta es una suposición que, si bien simplifica las cosas, no es absolutamente necesaria.

Alternativamente, podemos definir ICA como sigue: encontrar una transformación lineal dada por una matriz **W** tal que las variables aleatorias y_i, $i = 1, \ldots, n$ sean tan independientes como sea posible. Esta formulación no es muy diferente de la anterior, puesto que después de estimar **A**, simplemente calculando su inversa tenemos **W**.

Puede demostrarse que el problema está bien definido y que el modelo puede ser estimado si y sólo si los componentes s_i son no gaussianos. Este es un requisito fundamental que también explica la principal diferencia entre ICA y el análisis de factores, donde la no gaussianidad de los datos no se tiene en cuenta. De hecho, ICA puede considerarse análisis de factores no gaussianos, puesto que en el análisis de factores también modelamos los datos como mezclas lineales. Hablaremos de este requisito con más detalle posteriormente.

3.3.2 Cómo encontrar los componentes independientes.

Resulta sorprendente que los componentes independientes puedan ser estimados a partir de mezclas lineales sin ninguna suposición más que su

independencia. Ahora trataremos de explicar brevemente por qué y cómo es esto posible.

La incorrelación no es suficiente. Lo primero que diremos es que la independencia es una propiedad mucho más fuerte que la incorrelación. Considerando el problema de la separación ciega, podemos encontrar muchas representaciones incorreladas de las señales que no serían independientes y no separarían las fuentes. Por tanto, la incorrelación por sí misma no es suficiente para separar los componentes. Ésta es también la razón por la que el análisis de componentes principales (PCA) o el análisis de factores no pueden separar las señales, pues dan como resultado componentes que son incorrelados, pero poco más.

Para comprender intuitivamente lo que estamos diciendo, observemos la figura (3.3). En ella se representan dos señales aleatorias uniformemente distribuidas, es decir, pueden tomar valores dentro de un determinado intervalo con igual probabilidad. Hemos representado una en el eje de abcisas y otra en el de ordenadas. La figura toma la forma de un cuadrado debido a la independencia de los componentes.

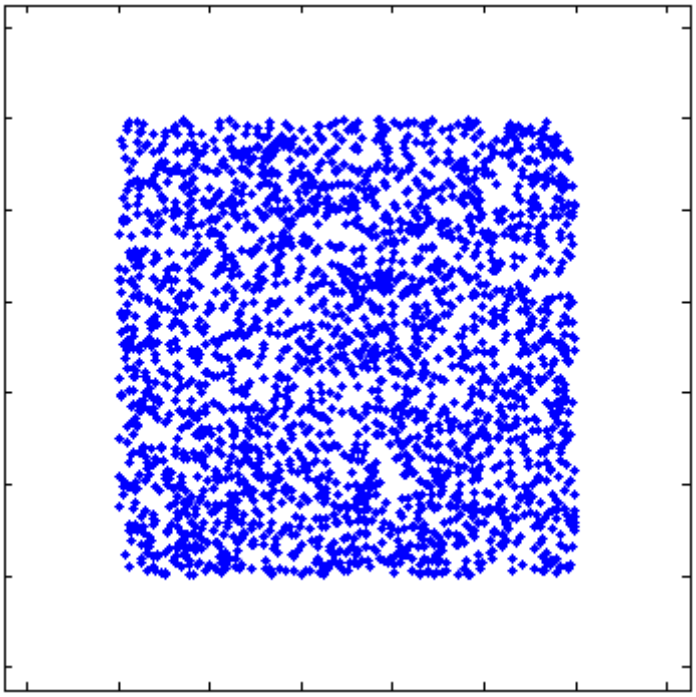


Figura 3.3 Muestras de dos componentes independientes s_1 y s_2 con distribuciones uniformes. *Eje horizontal:* s_1 *; Eje vertical:* s_2

Ahora vamos a fijarnos en la figura (3.4). En ella mostramos dos mezclas incorreladas de los componentes independientes de la figura (3.3). La mezcla se ha realizado usando una matriz ortogonal, que corresponde a una rotación del plano. Podemos observar claramente que, auque las mezclas son incorreladas, las distribuciones no son ya las mismas. Se puede ver fácilmente que las mezclas

no son independientes, a pesar de que los componentes a partir de los que han sido generadas sí lo eran: si el componente del eje horizontal toma un valor cercano a la esquina del extremo derecho, esto claramente restringe los posibles valores que puede tomar el componente del eje vertical. De hecho, usando los métodos conocidos de decorrelación, podemos transformar cualquier mezcla de componentes independientes en componentes incorrelados, en cuyo caso la mezcla es ortogonal. La figura (3.4) entonces nos da un indicio de por qué ICA es posible. Localizando los filos del cuadrado, podemos computar la rotación que nos dará los componentes originales. Ahora veremos un par de métodos mas sofisticados para la estimación ICA.

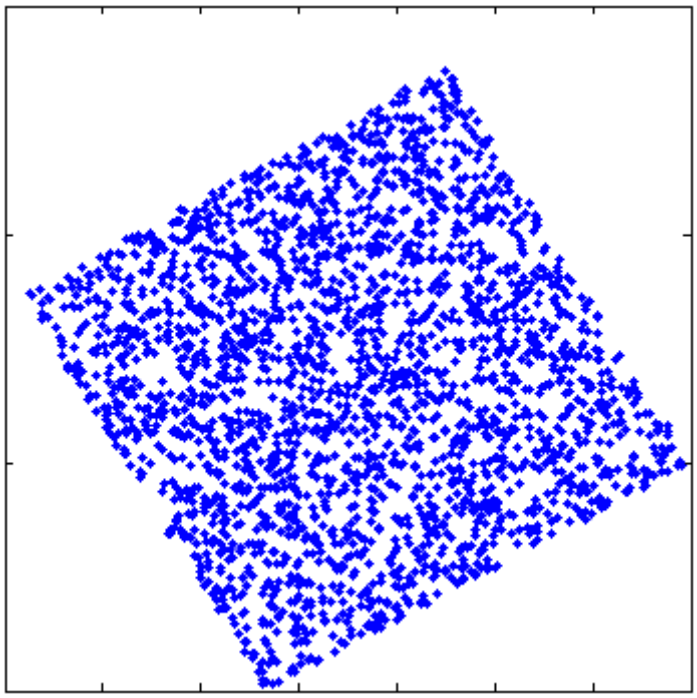


Figura 3.4 *Mezclas incorreladas* x_1 *y* x_2.
Eje horizontal: x_1*; Eje vertical:* x_2

La decorrelación no lineal es el método ICA básico. Una manera de demostrar cómo de fuerte es la independencia frente a la incorrelación es decir que la independencia implica incorrelación no lineal. Si s_1, s_2 son independientes, entonces sus transformaciones no lineales $g(s_1)$ y $h(s_2)$ son incorreladas (en el sentido en que su covarianza es nula). En contraste, para dos variables aleatorias que son meramente incorreladas, tales transformaciones no lineales no tendrán covarianza cero en general.

De esta forma podríamos intentar implementar ICA para una forma más fuerte de decorrelación, encontrando una representación donde los y_i sean incorrelados incluso después de alguna transformación no lineal. Esto nos da un simple principio para la estimación de la matriz **W**: *encontrar la matriz* ***W*** *tal que para cualquier* $i \neq j$*, los componentes* y_i *e* y_j *sean incorrelados, y los*

componentes transformados $g(y_i)$ *y* $h(y_j)$ *sean también incorrelados, con* ***g*** *y* ***h*** *funciones no lineales adecuadas.*

Esta es una aproximación válida para la estimación ICA: si las no-linealidades son escogidas apropiadamente, el método es capaz de encontrar los componentes independientes. De hecho, computando correlaciones no lineales entre las dos mezclas de la figura (3.4), uno inmediatamente ve que las mezclas no son independientes. Y aunque este principio es muy intuitivo, deja en el aire una pregunta: ¿cómo deben ser elegidas las no-linealidades **g** y **h**? La respuesta a esta pregunta se encuentra acudiendo a los principios de la teoría de la estimación y la teoría de la información. La teoría de la estimación provee el método clásico de estimar cualquier modelo estadístico: el método de máxima verosimilitud. Por otro lado, la teoría de la información nos da medidas exactas de la independencia, tales como la información mutua. Usando cualquiera de estos métodos podríamos determinar las funciones no lineales **g** y **h** de un modo satisfactorio.

Los componentes independientes son los componentes máximamente gaussianos. Otro principio importante de la estimación ICA es la no-gaussianidad máxima. La idea es que de acuerdo al teorema central del límite (ver sección 3.4.1.1), las sumas de variables aleatorias no gaussianas están más cerca de ser gaussianas que las originales. Por lo tanto, si tomamos una combinación lineal $y = \sum_i b_i x_i$ de las variables observadas, ésta será máximamente no gaussiana si es igual a uno de los componentes independientes. Esto se debe a que si fuera una mezcla real de dos o más componentes, estaría más cerca de una distribución gaussiana (por el teorema central del límite). Este principio puede enunciarse como sigue: se trata de *encontrar el máximo local de no-gaussianidad de una combinación lineal* $y = \sum_i b_i x_i$ *bajo la restricción de que la covarianza de y es constante. Cada máximo local nos da un componente independiente.*

En la práctica, para realizar medidas de la no-gaussianidad podríamos usar, por ejemplo, el kurtosis. Kurtosis es un cumulante de orden superior, una especie de generalización de la varianza usando polinomios de orden superior. Los cumulantes tienen propiedades estadísticas y algebraicas interesantes y por eso juegan un importante papel en la teoría de la estimación ICA.

La estimación ICA necesita algo más que las covarianzas. Lo que tienen en común todos los diferentes métodos de estimación ICA es que toman en consideración estadísticos que no están contenidos en la matriz de covarianzas. Esto se debe a que usando la matriz de covarianzas, podemos decorrelar los componentes en el sentido lineal, pero no en otro sentido más fuerte. Por eso todos los métodos basados en ICA usan algún tipo de estadísticos de orden superior, lo que significa concretamente que la información usada no está contenida en la matriz de covarianzas. Un ejemplo de esto es el kurtosis antes nombrado, del que hablaremos más adelante.

Gran importancia de los métodos numéricos. Además del principio de estimación, hay que encontrar un algoritmo adecuado que implemente las

necesidades de cómputo. Al hacer uso de funciones no cuadráticas en la estimación, los cálculos necesarios no pueden hacerse en general usando simplemente álgebra lineal, y por lo tanto es más exigente. Por ese motivo los algoritmos numéricos son una parte inseparable de los métodos de estimación ICA.

Los métodos numéricos más usados suelen estar basados en la optimización de funciones objetivo. El método de optimización básico es el método del gradiente. Se han desarrollado bastantes algoritmos que se adaptan muy bien a las necesidades concretas de ICA, como por ejemplo FastICA y muchos otros.

3.3.3 Restricciones de ICA.

Para asegurarnos de que el modelo básico ICA ya planteado puede estimarse, tenemos que hacer que se cumplan ciertas suposiciones y restricciones, de las que hablaremos a continuación:

1. Los componentes independientes tienen que ser estadísticamente independientes.

Éste es el principio sobre el que se sustenta ICA. Sorprendentemente, ya hemos visto que no se necesita mucho más aparte de esta suposición para que el modelo pueda ser estimado. Es por eso que ICA es un método tan poderoso para muchas aplicaciones en diferentes áreas. Básicamente, diremos que dado un conjunto de variables aleatorias $y_1, y_2, \ldots, y_n$, éstas son independientes si la información de los valores de y_i no nos da ninguna información sobre los valores de las demás variables y_j. Técnicamente, la independencia puede definirse partiendo de las densidades de probabilidad. Denotaremos por $p(y_1, y_2, \ldots, y_n)$ la función densidad de probabilidad (fdp) conjunta de los y_i, y por $p_i(y_i)$, la fdp marginal de y_i, es decir, la fdp de y_i cuando se considera sola. Entonces decimos que los y_i son independientes si y sólo si la fdp conjunta es factorizable de la siguiente forma:

$$p(y_1, y_2, \ldots, y_n) = p_1(y_1) p_2(y_2) \ldots p_n(y_n) \tag{3.5}$$

2. Los componentes independientes deben tener distribuciones no gaussianas.

Los cumulantes de orden superior de las distribuciones gaussianas son cero, sin embargo, esta información de orden superior es esencial en el modelo de estimación ICA, como se verá posteriormente. Por lo tanto, ICA es imposible si las variables observadas tienen distribuciones gaussianas. Cabe destacar que en el modelo básico no estamos suponiendo en ningún momento que conozcamos las distribuciones (no gaussianas) de los componentes independientes, sin embargo, si éstas fueran conocidas, el problema se simplificaría considerablemente.

3. Por simplicidad, supondremos que la matriz de mezcla es cuadrada.

Dicho de otro modo, el número de componentes independientes es igual al número de mezclas observadas. Esta suposición puede no cumplirse a veces, pero la hacemos porque simplifica mucho la estimación. Así, después de estimar la matriz **A**, podemos calcular su inversa y obtener los componentes independientes simplemente como **s=Bx.**

También estamos suponiendo entonces que la matriz de mezcla es invertible. Si ese no fuera el caso, significa que hay mezclas redundantes que pueden ser omitidas, en cuyo caso la matriz no sería cuadrada.

Así, bajo las tres hipótesis precendentes (o al menos las dos primeras), podemos asegurar que el modelo ICA es identificable, significando esto que la matriz de mezcla y los componentes independientes pueden ser estimados, salvo quizás algunas indeterminaciones triviales que discutiremos a continuación.

3.3.4 Ambigüedades de ICA.

En el modelo ICA ya descrito, no es difícil ver que habrá que solucionar las siguientes ambigüedades o indeterminaciones:

1. No podemos determinar las varianzas (energías) de los componentes independientes.

La razón es que tanto **s** como **A** son desconocidos, y por tanto cualquier multiplicación por un escalar de una de las fuentes s_i, puede cancelarse siempre dividiendo la correspondiente columna a_i de **A** por el mismo escalar, llamémosle α_i:

$$\mathbf{x} = \sum_i \left(\frac{1}{\alpha_i} \mathbf{a}_i \right) (s_i \alpha_i) \tag{3.6}$$

Como consecuencia, podemos fijar las magnitudes de los componentes independientes. Puesto que son variables aleatorias, la forma más natural de hacerlo es suponer que cada una tiene varianza unidad: $E\{s_i^2\} = 1$. Por tanto la matriz **A** será adaptada en los métodos de solución ICA para tener en cuenta esta restricción. Notar que aún continúa la ambigüedad del signo: podemos multiplicar un componente independiente por -1 sin afectar al modelo. Esta ambigüedad, afortunadamente, es insignificante en la mayoría de aplicaciones prácticas.

2. No podemos determinar la ordenación de los componentes independientes.

La razón es de nuevo que ambos **s** y **A** son desconocidos, y por lo tanto podemos cambiar el orden de los términos libremente sin que tenga consecuencias en los resultados. Formalmente, una matriz de permutación **P** y su inversa pueden ser sustituidas en el modelo resultando $\mathbf{x} = \mathbf{A}\mathbf{P}^{-1}\mathbf{P}\mathbf{s}$. Los elementos de **Ps** son las variables independientes originales s_i, pero en otro orden.

3.3.5 Otras consideraciones.

3.3.5.1 Centrado de las variables.

Sin pérdida de generalidad, podemos suponer que las variables mezcla y los componentes independientes tienen media cero. Esta suposición simplifica mucho tanto la teoría como los algoritmos, y así lo haremos de aquí en adelante.

Si la suposición de media cero no se cumple, podemos hacer un preprocesado de las señales para forzar que así sea, ya que es posible centrar las variables observadas simplemente restándole su media muestral antes de hacer la separación ICA. Si llamamos **x**' a los datos antes del preprocesado y **x** a los mismos después:

$$\mathbf{x} = \mathbf{x}' - E\{\mathbf{x}'\} \tag{3.7}$$

De este modo los componentes independientes también tendrán media cero. La matriz de mezcla sin embargo, permanecerá igual tras este preprocesado, así que podemos hacerlo siempre sin afectar a la estimación de la matriz de la misma. Después de estimar la matriz de mezcla y los componentes independientes para los datos de media cero, la media sustraída puede ser reconstruida simplemente añadiendo $\mathbf{A}^{-1}E\{\mathbf{x}'\}$ a los componentes independientes de media cero.

3.3.5.2 ICA frente al blanqueo.

Debido a que si tenemos algunas variables aleatorias es sencillo transformarlas linealmente en variables incorreladas, sería tentador intentar estimar los componentes independientes mediante ese método, que es llamado típicamente *blanqueo*, y que a menudo es implementado por PCA, como dijimos anteriormente. Ahora mostraremos que eso no es posible, y discutiremos la relación entre ICA y los métodos basados en la decorrelación.

Una forma más débil de independencia es la incorrelación. Definiremos brevemente la incorrelación: se dice que dos variables aleatorias y_1 e y_2 son incorreladas si su covarianza es cero:

$$\mathrm{cov}(y_1, y_2) = E\{y_1 y_2\} - E\{y_1\}E\{y_2\} \tag{3.8}$$

Siempre supondremos que las variables tienen media cero a no ser que indiquemos explícitamente lo contrario. De esta forma, la covarianza es igual a la correlación $\mathrm{corr}(y_1, y_2) = E\{y_1 y_2\}$, y la incorrelación es lo mismo que la correlación cero.

Si las variables aleatorias son independientes, son incorreladas. Esto es debido a que si y_1 e y_2 son independientes, entonces para dos funciones cualesquiera h_1 y h_2, tenemos

$$E\{h_1(y_1)h_2(y_2)\} = E\{h_1\}E\{h_2\} \tag{3.9}$$

Tomando $h_1(y_1) = y_1$, y $h_2(y_2) = y_2$, vemos que esto implica la incorrelación.

Sin embargo no sucede lo contrario, esto es, la incorrelación no implica independencia. Por ejemplo, supongamos que (y_1, y_2) toman valores discretos y siguen una distribución tal que el par toma cualquiera de los siguientes valores: (0,1), (0,-1), (1,0) y (-1,0) con probabilidad 1/4. Se puede afirmar entonces que y_1 e y_2 son incorrelados, lo que puede calcularse fácilmente. Por otro lado,

$$E\{y_1^2 y_2^2\} = 0 \neq \frac{1}{4} = E\{y_1^2\}E\{y_2^2\} \tag{3.10}$$

así que la condición dada en (3.9) es violada, y las variables no pueden ser independientes.

Una propiedad ligeramente más fuerte que la *incorrelación* es la blancura. Recordamos que si un vector aleatorio de media cero, llamémosle *y*, posee esta propiedad, significa que sus componentes son incorrelados y sus varianzas iguales a la unidad. En otras palabras, la matriz de covarianzas (y la matriz de correlación también) de **y** es igual a la matriz identidad: $E\{\mathbf{y}\mathbf{y}^T\} = \mathbf{I}$

En consecuencia, el blanqueo significaba que transformamos linealmente el vector de datos observados **x** multiplicándolo linealmente por alguna matriz **V:** **z**=**Vx** de forma que obtenemos un nuevo vector **z** que es blanco.

3.3.5.2 ICA y las variables gaussianas.

El blanqueo puede ayudarnos a comprender por qué las variables aleatorias gaussianas están prohibidas en ICA. Supongamos que la distribución conjunta de dos componentes independientes, s_1 y s_2, es gaussiana. Esto significa que su función densidad de probabilidad conjunta viene dada por:

$$p(s_1, s_2) = \frac{1}{2\pi}\exp\left(-\frac{s_1^2 + s_2^2}{2}\right) = \frac{1}{2\pi}\exp\left(-\frac{\|\mathbf{s}\|^2}{2}\right) \tag{3.11}$$

Supongamos ahora que la matriz de mezcla **A** es ortogonal, por ejemplo, porque los datos han sido blanqueados. Fijándonos en que para una matriz ortogonal se cumple que $\mathbf{A}^{-1} = \mathbf{A}^T$, se puede demostrar que la densidad conjunta de las mezclas x_1 y x_2 es:

$$p(x_1, x_2) = \frac{1}{2\pi}\exp\left(-\frac{\|\mathbf{A}^T\mathbf{x}\|^2}{2}\right)\left|\det \mathbf{A}^T\right| \tag{3.12}$$

Debido a la ortogonalidad de **A**, tenemos que $\|\mathbf{A}^T\mathbf{x}\|^2 = \|\mathbf{x}\|^2$ y que $|\det \mathbf{A}| = 1$. Lo mismo sucede para la traspuesta de **A**, por lo tanto llegamos a la expresión:

$$p(x_1, x_2) = \frac{1}{2\pi} \exp\left(- \frac{\|\mathbf{x}\|^2}{2} \right) \tag{3.13}$$

Mirando la expresión anterior nos damos cuenta rápidamente de que la matriz de mezcla no cambia para nada la función densidad de probabilidad. Las distribuciones de las señales originales y la de las mezclas son idénticas, y por consiguiente, no hay manera de inferir la matriz de mezcla a partir de las mezclas.

Esto que acabamos de decir puede entenderse mejor gráficamente, observando la figura (3.5), en la que se representan muestras de dos mezclas ortogonales. La figura muestra que la densidad es rotacionalmente simétrica, por lo tanto, no contiene ninguna información sobre las direcciones de las columnas de la matriz de mezcla **A**, y sin esa información no puede estimarse dicha matriz.

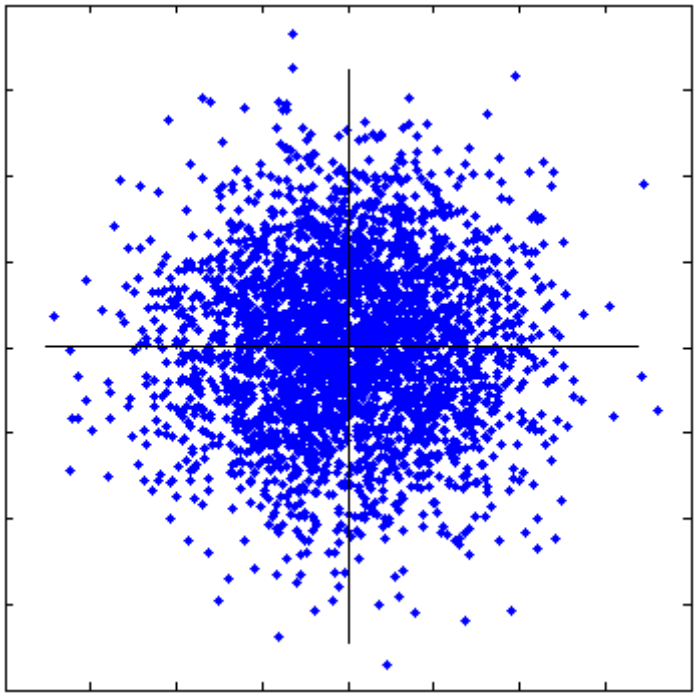


Figura 3.5 *Distribución de dos variables independientes gaussianas*

Así, en el caso de componentes independientes gaussianos, lo máximo que podemos hacer es blanquear los datos, pero no podemos llevar a cabo una separación mediante ICA.

Una pregunta interesante que se nos podría ocurrir es: ¿qué pasaría si intentáramos estimar el modelo ICA cuando algunos de los componentes son gaussianos y otros no gaussianos? La respuesta es que, en ese caso, podemos estimar todos los componentes no gaussianos, pero no podemos separar los componentes gaussianos unos de otros. En otras palabras, algunos de los componentes estimados serán combinaciones lineales arbitrarias de los

componentes gaussianos. Se puede deducir por lo tanto, que si hay solamente un componente gaussiano, este sí que puede ser estimado, porque no podría mezclarse con ningún otro componente con la misma distribución.

3.4 Criterios ICA y técnicas de optimización.

3.4.1 Maximización de la no-gaussianidad.

El primer principio que veremos para estimar el modelo ICA es el basado en la maximización de la no-gaussianidad.

Como ya se ha visto en apartados anteriores, la no-gaussianidad es un aspecto con una importancia primordial en ICA, sin ella, la estimación de los componentes independientes no sería posible. Por lo tanto no debe sorprendernos que pueda ser usada como criterio en la estimación ICA. Posiblemente esta es la razón principal por la que ICA ha tenido tanto protagonismo en la investigación últimamente, ya que en la mayoría de las teorías estadísticas clásicas, se suponía que las variables aleatorias tenían distribuciones gaussianas.

3.4.1.1 No gaussianidad e independencia.

Comenzaremos este apartado recordando el teorema central del límite, ya que tiene una importancia fundamental en este criterio. Supongamos que tenemos una variable aleatoria

$$x_k = \sum_{i=1}^{k} z_i \tag{3.14}$$

consistente en la suma de una secuencia de variables aleatorias z_i independientes e idénticamente distribuidas. Antes de seguir, definiremos $y_k = \frac{x_k - m_{x_k}}{\sigma_{x_k}}$, donde m_{x_k} y σ_{x_k} son la media y la varianza de x_k. Hacemos esta transformación debido a que la media y la varianza de x_k pueden crecer sin límite según $k \to \infty$. Dicho esto, puede demostrarse que la distribución de y_k converge a una distribución gaussiana con media cero y varianza unidad si $k \to \infty$. Este resultado se conoce como teorema central del límite. Hay muchas formas diferentes de enunciar este teorema, alguna de ellas relajan algunas suposiciones como las de la independencia y las distribuciones idénticas de las variables aleatorias. Diremos también que se debe al teorema central del límite el hecho de que muchos y muy diversos fenómenos aleatorios se modelen estadísticamente como variables aleatorias gaussianas. Por ejemplo, el ruido aditivo, que procede de la suma de un gran número de pequeños efectos, se simplifica mucho siendo modelado como una variable gaussiana, evitándonos lo engorroso de vernos obligados a tener en cuenta muchas variables diferentes.

Una vez recordado el teorema central del límite, podemos quedarnos con una idea fundamental, y es que, si hablamos de forma aproximada, podríamos decir que la suma de dos variables aleatorias independientes tendrá una distribución que está más cerca de ser gaussiana que cualquiera de las distribuciones de las dos variables aleatorias originales.

Vamos a suponer para el vector de datos **x** el modelo ICA que nos es familiar: **x=As**. **x** será entonces una mezcla de componentes independientes. Por motivos de simplicidad, supondremos en esta sección que todos los componentes independientes están idénticamente distribuidos. La estimación de los componentes independientes puede hacerse encontrando las combinaciones lineales adecuadas de las variables de mezcla, puesto que podemos invertir la mezcla como: $\mathbf{s} = \mathbf{A}^{-1}\mathbf{x}$. Así, para estimar los componentes independientes, podemos considerar una combinación lineal de los x_i. Denotaremos ésta como $y = \mathbf{b}^T\mathbf{x} = \sum_i b_i x_i$, donde **b** es un vector aún por determinar. Hay que fijarse en que también se cumple $y = \mathbf{b}^T\mathbf{A}\mathbf{s}$. Así, **y** es una cierta combinación lineal de los s_i, con coeficientes dados por $\mathbf{b}^T\mathbf{A}$. Denotaremos esto como el vector **q**. Entonces tenemos:

$$y = \mathbf{b}^T\mathbf{x} = \mathbf{q}^T\mathbf{s} = \sum_i q_i s_i \tag{3.15}$$

Si **b** fuera una de las filas de la inversa de **A**, esta combinación lineal $\mathbf{b}^T\mathbf{x}$ sería igual a uno de los componentes independientes. En ese caso, el correspondiente **q** sería tal que sólo uno de sus elementos sería 1 los demás cero.

La cuestión ahora es: ¿cómo podríamos usar el teorema central del límite para determinar **b** de forma que iguale a una de las filas de la inversa de **A**? En la práctica, no podemos hallar ese valor exacto de **b**, porque no tenemos conocimiento de la matriz **A**, pero sí podemos encontrar un estimador que nos de una buena aproximación.

Variaremos ahora los coeficientes en **q**, y veremos cómo cambia la distribución de $y = \mathbf{q}^T\mathbf{s}$. La idea fundamental es que, puesto que una suma de incluso dos variables aleatorias independientes es más gaussiana que las variables originales, $y = \mathbf{q}^T\mathbf{s}$ normalmente es más gaussiana que cualquiera de los s_i y se vuelve menos gaussiana cuando de hecho es uno de los s_i. En este caso, obviamente sólo uno de los elementos q_i de **q** es no nulo.

En la práctica no conocemos los valores de **q**, pero no los necesitamos porque $\mathbf{q}^T\mathbf{s} = \mathbf{b}^T\mathbf{x}$ por la definición de **q**. Podemos simplemente dejar que **b** varíe y mirar la distribución de $\mathbf{b}^T\mathbf{x}$. Por lo tanto, podríamos tomar **b** como un vector que maximizase la no-gaussianidad de $\mathbf{b}^T\mathbf{x}$. Tal vector necesariamente correspondería a $\mathbf{q} = \mathbf{A}^T\mathbf{b}$, que tiene sólo un componente no nulo. Esto significa que $y = \mathbf{b}^T\mathbf{x} = \mathbf{q}^T\mathbf{s}$ es igual a uno de los componentes independientes. Maximizar la no-gaussianidad de $\mathbf{b}^T\mathbf{x}$ nos da entonces uno de los componentes independientes. De hecho, la optimización de la no-gaussianidad en el espacio

n-dimensional de vectores **b** tiene 2*n* máximos locales, dos para cada componente independiente, correspondientes a s_i y a $-s_i$ (ya que como ya dijimos los componentes independientes pueden ser estimados salvo la ambigüedad del signo).

Recapitulando, hemos formulado la estimación ICA como la búsqueda de direcciones que son máximamente no-gaussianas: cada máximo local nos da un componente independiente. Nuestro acercamiento a la solución aquí ha sido un poco heurística, pero se verá en una sección posterior con una justificación rigurosa.

Desde un punto de vista práctico debemos responder a las siguientes preguntas: ¿cómo se mide la no-gaussianidad de $\mathbf{b}^T\mathbf{x}$? y ¿cómo podemos calcular los valores que maximizan localmente tal medida de la no-gaussianidad? El resto de este apartado lo dedicaremos a intentar responder estas cuestiones.

3.4.1.2 Medida de la no-gaussianidad mediante kurtosis.

Para usar la no-gaussianidad en la estimación ICA, debemos tener una medida cuantitativa de la no-gaussianidad de una variable aleatoria, digamos *y*. Ahora mostraremos cómo usar kurtosis, una medida clásica de la no-gaussianidad, para la estimación ICA. Kurtosis no es otra cosa que el nombre que se le ha dado al cumulante de cuarto orden de una variable aleatoria. Para más información de los cumulantes, ver apéndice 1. Así obtenemos un método de estimación que puede considerarse una variante del clásico método de los momentos.

El kurtosis de *y*, denotado por kurt(*y*), se define como:

$$\text{kurt}(y) = E\{y^4\} - 3\left(E\{y^2\}\right)^2 \tag{3.16}$$

Recordar que todas las variables aleatorias que aquí usamos tienen media cero; en el caso general, la definición de kurtosis es ligeramente más complicada. Para simplificar las cosas, podemos suponer también que hemos normalizado la variable *y* para que su varianza sea igual a la unidad: $E\{y^2\} = 1$. De esa manera la ecuación anterior queda como $\text{kurt}(y) = E\{y^4\} - 3$. Esto muestra que kurtosis es simplemente una versión normalizada del momento de cuarto orden $E\{y^4\}$. Para un vector gaussiano *y*, el momento de cuarto orden es igual a $3\left(E\{y^2\}\right)^2$, y por lo tanto kurtosis es cero para las variables aleatorias gaussianas. Para la mayoría del resto de variables aleatorias, kurtosis no es cero.
El kurtosis puede ser positivo o negativo, en este último caso la variable aleatoria en cuestión es llamada subgaussiana, y aquellas con kurtosis positivos serán supergaussianas. Las variables supergaussianas suelen tener una función densidad de probabilidad puntiaguda, donde la probabilidad es relativamente grande en cero y en valores altos de la variable, mientras que es pequeña para valores intermedios. Un ejemplo típico es la distribución laplaciana, que podemos observar en la figura (3.6).

Las variables aleatorias subgaussianas, por otro lado, tienen típicamente una distribución "plana", que es aproximadamente constante cerca de cero, y toma valores grandes de la variable con muy poca probabilidad. Un ejemplo de esto se muestra en la figura (3.7).

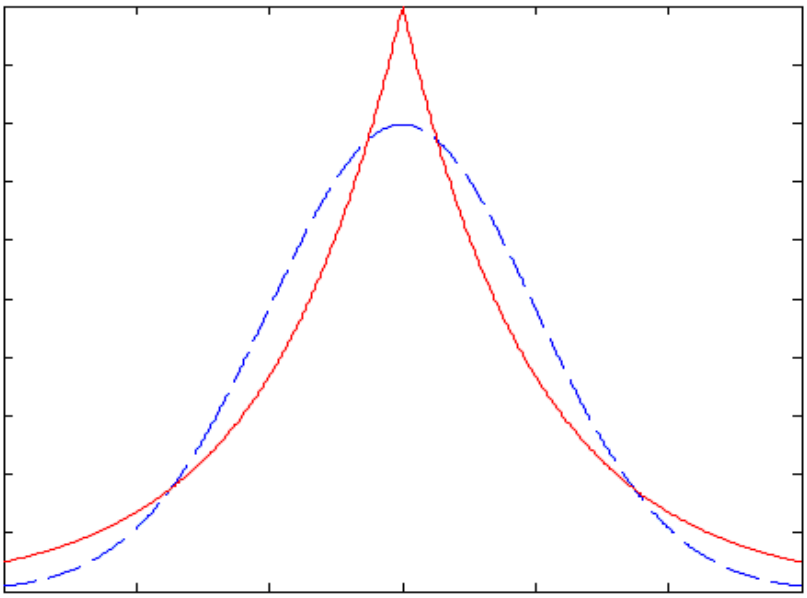


Figura 3.6 *Función densidad de probabilidad de la distribución laplaciana (rojo), que es un ejemplo típico de distribución supergaussiana. También se muestra la distribución gaussiana (azul).*

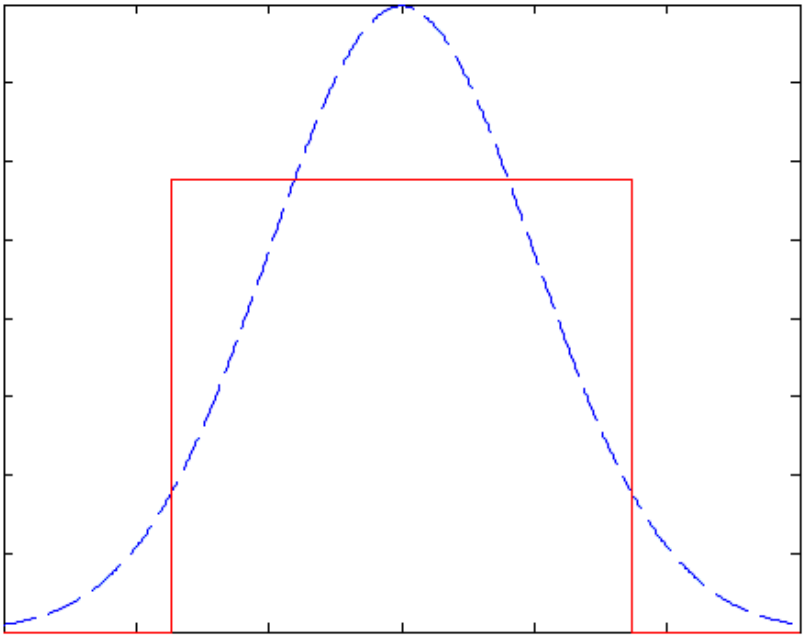


Figura 3.7 *Función densidad de probabilidad de la distribución uniforme (rojo), que es un ejemplo típico de distribución subgaussiana. También se muestra la distribución gaussiana (azul).*

Típicamente la no-gaussianidad es medida por el valor absoluto de kurtosis. También puede usarse el kurtosis elevado al cuadrado. Ambas medidas son nulas para una variable gaussiana, y mayores que cero para la mayoría de las variables no gaussianas. También existen variables aleatorias que, sin ser gaussianas, tienen kurtosis cero, pero rara vez aparecen en la práctica. La razón por la que kurtosis (o su valor absoluto) ha sido tan usado para medir la no-gaussianidad en ICA y campos relacionados es la simplicidad, tanto computacional como teórica. Computacionalmente, kurtosis puede estimarse simplemente usando el momento de cuarto orden de los datos (si la varianza se mantiene constante). El análisis teórico también se simplifica debido a la siguiente propiedad lineal: si x_1, x_2 son dos variables aleatorias independientes, se sigue que $\text{kurt}(x_1 + x_2) = \text{kurt}(x_1) + \text{kurt}(x_2)$ y $\text{kurt}(\alpha x_1) = \alpha^4 \text{kurt}(x_1)$, donde α es una constante. Estas propiedades pueden demostrarse fácilmente viendo las propiedades de los cumulantes (apéndice 1).

3.4.1.3 Algoritmo del gradiente usando kurtosis.

En la práctica, para maximizar el valor absoluto de kurtosis, comenzaríamos con algún vector **w**, calcularíamos la dirección en la cual el valor absoluto del kurtosis de $y = \mathbf{w}^T\mathbf{z}$ crece más fuertemente, basándonos en las muestras disponibles **z**(1),…,**z**(*T*) del vector de mezcla **z**, y luego moveríamos el vector **w** en esa dirección. Esta idea se implementa mediante los métodos basados en el gradiente y sus extensiones.

Usando los principios expuestos en el apéndice 2, el gradiente del valor absoluto del kurtosis de $\mathbf{w}^T\mathbf{z}$ puede calcularse como:

$$\frac{\partial \left|\text{kurt}\left(\mathbf{w}^T\mathbf{z}\right)\right|}{\partial \mathbf{w}} = 4\text{sign}\left(\text{kurt}\left(\mathbf{w}^T\mathbf{z}\right)\right)\left[E\left\{\mathbf{z}\left(\mathbf{w}^T\mathbf{z}\right)^3\right\} - 3\mathbf{w}\|\mathbf{w}\|^2\right] \tag{3.17}$$

puesto que para los datos blanqueados tenemos que $E\left\{\left(\mathbf{w}^T\mathbf{z}\right)^2\right\} = \|\mathbf{w}\|^2$. Ya que estamos optimizando esta función en la esfera unidad correspondiente a $\|\mathbf{w}\|^2 = 1$, el método del gradiente debe complementarse con la proyección de **w** en la esfera unidad tras cada paso. Esto se hará simplemente dividiendo **w** por su norma.

Operando de esa forma obtenemos el siguiente algoritmo del gradiente:

$$\begin{aligned} \Delta \mathbf{w} &\propto \text{sign}\left(\text{kurt}\left(\mathbf{w}^T\mathbf{z}\right)\right)E\left\{\mathbf{z}\left(\mathbf{w}^T\mathbf{z}\right)^3\right\} \\ \mathbf{w} &\leftarrow \mathbf{w}/\|\mathbf{w}\| \end{aligned} \tag{3.18}$$

También podemos obtener una versión on-line (adaptativa) de este algoritmo, esto es posible omitiendo el segundo operador esperanza en el algoritmo:

$$\begin{aligned} \Delta \mathbf{w} &\propto \text{sign}\left(\text{kurt}\left(\mathbf{w}^T\mathbf{z}\right)\right)\mathbf{z}\left(\mathbf{w}^T\mathbf{z}\right)^3 \\ \mathbf{w} &\leftarrow \mathbf{w}/\|\mathbf{w}\| \end{aligned} \tag{3.19}$$

En este caso cada observación $\mathbf{z}(t)$ puede ser usada en el algoritmo una sola vez. Sin embargo, conviene darse cuenta de que al calcular $\text{sign}\left(\text{kurt}\left(\mathbf{w}^T\mathbf{z}\right)\right)$, el operador esperanza en la definición de kurtosis no puede ser omitido. En vez de eso, el kurtosis deber ser apropiadamente estimado haciendo un promedio temporal, que se hará on-line. Denotando por γ la estima del kurtosis, podríamos usar:

$$\Delta\gamma \propto \left(\left(\mathbf{w}^T\mathbf{z}\right)^4 - 3\right) - \gamma \tag{3.20}$$

La verdad es que en muchos casos se conoce de antemano la naturaleza de las distribuciones de los componentes independientes, por ejemplo si son subgaussianos o supergaussianos. En esa situación uno puede simplemente colocar el signo adecuado al kurtosis en el algoritmo y evitar su estimación.

Una vez dicho todo esto, cabe destacar que el algoritmo del gradiente no es ni mucho menos el más eficiente para maximizar el valor absoluto del kurtosis. La ventaja de los métodos basados en el gradiente es que las entradas $\mathbf{z}(t)$ pueden usarse una sola vez en el algoritmo, permitiendo una rápida adaptación en entornos no estacionarios. Sin embargo la convergencia puede volverse muy lenta según la elección de la tasa de aprendizaje. Para hacer el aprendizaje mucho más rápido y fiable aparecen los algoritmos de iteración de punto fijo. Un ejemplo de los métodos de ese tipo es el algoritmo FastICA.

3.4.1.4 Algoritmo de punto fijo usando kurtosis.

Para derivar una iteración de punto fijo más eficiente que la del método del gradiente, debemos darnos cuenta que en un punto estable del algoritmo del gradiente, éste tiene que apuntar a la dirección de **w**, es decir, el gradiente debe ser igual a **w** multiplicado por alguna constante escalar. Sólo en tal caso, al sumarle el gradiente a **w**, éste mantendrá intacta su dirección, y podremos tener la deseada convergencia. Llamando al gradiente de la expresión (3.17) **w**, tendríamos:

$$\mathbf{w} \propto \left[E\left\{\mathbf{z}\left(\mathbf{w}^T\mathbf{z}\right)^3\right\} - 3\|\mathbf{w}\|^2\mathbf{w}\right] \tag{3.21}$$

Esta ecuación sugiere inmediatamente un algoritmo de punto fijo donde primero calculamos el lado derecho, y le damos un nuevo valor a **w**:

$$\mathbf{w} \leftarrow E\left\{\mathbf{z}\left(\mathbf{w}^T\mathbf{z}\right)^3\right\} - 3\mathbf{w} \tag{3.22}$$

En cada iteración, **w** se divide por su norma para seguir cumpliendo las restricciones. El vector **w** final da uno de los componentes independientes mediante la combinación lineal $\mathbf{w}^T\mathbf{z}$. En la práctica las medias estadísticas pueden remplazarse por sus estimas.

Con esto conseguimos que los antiguos y nuevos valores de **w** apunten en la misma dirección. Este algoritmo se llama FastICA, y funciona muy bien en la práctica, además con una rápida convergencia.

3.4.1.5 Medida de la no-gaussianidad mediante la negentropía.

En las secciones anteriores, hemos estado mostrando cómo medir la no-gaussianidad mediante kurtosis, y obtuvimos un método de estimación ICA bastante simple. Sin embargo, kurtosis también tiene algunos puntos que lo desaconsejan en la práctica, cuando tenemos que estimar su valor de una muestra medida. El principal problema que presenta kurtosis es que es muy sensible a los "outliers". En estadística, un outlier es una observación aislada cuyo valor es muy diferente al resto de los datos. Por ejemplo, si tenemos una muestra de 1000 valores de una variable aleatoria con media cero y varianza 1 que contiene un valor igual a 10, entonces el kurtosis será igual, al menos, a $10^4/1000 - 3 = 7$, lo que significa que un solo valor hace que el kurtosis sea grande. Entonces podemos deducir que el valor de kurtosis puede estar condicionado por muy pocas muestras que pueden ser erróneas o irrelevantes. En otras palabras, kurtosis no es una medida robusta de la no-gaussianidad en determinados casos.

Podría haber otras medidas de la no-gaussianidad que fueran mejores que kurtosis en alguna situaciones. En este apartado vamos a considerar una de ellas, la negentropía, que es la segunda medida más importante de la no-gaussianidad. Sus propiedades son de algún modo las opuestas a las de kurtosis, es un método robusto pero computacionalmente complicado.

La negentropía está basada en la entropía diferencial. La entropía es el concepto básico de la teoría de la información, y diremos que la entropía de una variable aleatoria está relacionada con la información que da la observación de las variables dadas. Cuanto más "aleatoria", es decir, más impredecible y falta de estructura sea la variable, mayor será su entropía. Definimos la entropía diferencial *H* de un vector aleatorio **y** con densidad $p_y(\boldsymbol{\eta})$ como:

$$H(\mathbf{y}) = -\int p_y(\boldsymbol{\eta}) \log p_y(\boldsymbol{\eta}) d\boldsymbol{\eta} \tag{3.23}$$

Un resultado fundamental de la teoría de la información es que una variable gaussiana tiene la mayor entropía de entre todas las variables aleatorias con igual varianza. Esto tiene una consecuencia importantísima para nosotros: la negentropía puede ser usada como medida de la no-gaussianidad. De hecho, muestra de que la distribución gaussiana es la "más aleatoria" o la menos estructurada de todas las distribuciones. La entropía es pequeña para distribuciones que están claramente concentradas en torno a ciertos valores, por ejemplo, aquellas cuya función densidad de probabilidad es muy picuda.

Para obtener una medida de la no-gaussianidad que sea cero para una variable gaussiana y que nunca sea negativa, a menudo se usa una versión normalizada de la entropía diferencial, llamada negentropía. La negentropía *J* se define como:

$$J(\mathbf{y}) = H(\mathbf{y}_{gauss}) - H(\mathbf{y}) \tag{3.24}$$

donde $\mathbf{y}_{gauss}$ es un vector aleatorio gaussiano con la misma matriz de correlación y de covarianza que **y**. Debido a las propiedades ya mencionadas, la negentropía es siempre no negativa, y es cero si y sólo si **y** es gaussiana. Además tiene la interesante propiedad de que es invariante para transformaciones lineales invertibles.

La ventaja de usar la negentropía como medida de la no-gaussianidad es que está bien justificada por la teoría estadística. De hecho, la negentropía es de algún modo el estimador óptimo de la no-gaussianidad. El problema viene por el alto coste computacional, puesto que si usamos la definición de negentropía necesitamos la estimación de la función densidad de probabilidad. Por esto encontrar una aproximación de la negentropía nos sería muy útil, como veremos a continuación.

3.4.1.6 Aproximación de la negentropía.

En la práctica sólo se necesita una aproximación unidimensional de la negentropía, por eso sólo consideraremos aquí el caso escalar. El método clásico para aproximar la negentropía hace uso de los cumulantes de orden superior, de esta forma se llega a la siguiente aproximación:

$$J(y) \approx \frac{1}{12} E\{y^3\}^2 + \frac{1}{48} kurt(y)^2 \qquad (3.25)$$

Suponemos como siempre que la variable aleatoria *y* tiene media cero y varianza unidad. La verdad es que esta aproximación a menudo nos lleva a usar el kurtosis como en el método presentado con anterioridad. Esto se debe a que el primer término del lado derecho de la expresión (3.25) es cero para variables aleatorias con una distribución aproximadamente simétrica, lo que es muy común en la práctica. Si eso sucede, la expresión (3.25) es equivalente al cuadrado de kurtosis. La maximización del cuadrado de kurtosis por supuesto también maximiza su valor absoluto, por lo tanto este método nos estaría llevando más o menos al mismo que ya hemos descrito. En particular, esta aproximación sufre la falta de robustez del kurtosis que estamos intentando evitar con este método.

Entonces habrá que desarrollar alguna aproximación más sofisticada de la negentropía. Una técnica útil consiste en generalizar la aproximación de cumulantes de orden superior para que use las medias estadísticas de funciones no cuadráticas generales. En general, podemos remplazar los polinomios y^3 e y^4 por otras funciones cualquiera G^i (donde *i* es un índice, no un exponente). Entonces el método nos da una forma sencilla de aproximar la negentropía basándose en las esperanzas $E\{G^i(y)\}$. Como caso especial, podemos tomar dos funciones no cuadráticas G^1 y G^2 siendo G^1 impar y G^2 par, obteniendo la siguiente aproximación:

$$J(y) \approx k_1(E\{G^1(y)\})^2 + k_2(E\{G^2(y)\} - E\{G^2(v)\})^2 \qquad (3.26)$$

siendo k_1 y k_2 constantes positivas, y υ una variable gaussiana de media cero y varianza 1. Esta es una generalización de la aproximación basada en los momentos dada en (3.25), que se obtiene tomando $G^1(y) = y^3$ y $G^2(y) = y^4$. En el caso en que solamente usemos una función no cuadrática *G*, la aproximación es:

$$J(y) \propto [E\{G(y)\} - E\{G(\upsilon)\}]^2 \tag{3.27}$$

Esta expresión es válida para prácticamente cualquier función no cuadrática *G*. Podemos darnos cuenta de que tomando $G(y) = y^4$ obtenemos la aproximación basada en kurtosis.

Operando de esta forma podemos obtener aproximaciones de la negentropía con un buen compromiso entre las propiedades de las dos medidas clásicas de la no-gaussianidad dadas por kurtosis y la negentropía. Además son conceptualmente simples, rápidas de calcular y especialmente robustas, por todo esto son apropiadas para ICA, que es lo que estábamos buscando.

3.4.1.7 Algoritmo del gradiente usando la negentropía.

Al igual que con kurtosis, podemos derivar un algoritmo sencillo del gradiente para maximizar la negentropía. Tomando el gradiente de la aproximación de la negentropía dada en (3.25) con respecto a **w**, y teniendo en cuenta la normalización $E\{(\mathbf{w}^T\mathbf{z})^2\} = \|\mathbf{w}\|^2 = 1$, se obtiene el siguiente algoritmo:

$$\begin{aligned} \Delta \mathbf{w} &\propto \gamma E\{\mathbf{z} g(\mathbf{w}^T\mathbf{z})\} \\ \mathbf{w} &\leftarrow \mathbf{w}/\|\mathbf{w}\| \end{aligned} \tag{3.28}$$

donde $\gamma = E\{G(\mathbf{w}^T\mathbf{z})\} - E\{G(\upsilon)\}$, siendo υ una variable aleatoria gaussiana estandarizada. La normalización es necesaria para proyectar **w** en la esfera unidad y así mantener la varianza $\mathbf{w}^T\mathbf{z}$ constante. La función *g* es la derivada de la función *G* usada en la aproximación de la negentropía. La esperanza podría omitirse para obtener la versión adaptativa (on-line) del algoritmo. El parámetro γ puede ser estimado fácilmente como $\Delta\gamma \propto (G(\mathbf{w}^T\mathbf{z}) - E\{G(\upsilon)\}) - \gamma$. Este parámetro, γ, corresponde al signo de kurtosis.

3.4.1.8 Algoritmo de punto fijo usando negentropía.

Al igual que sucedía con kurtosis, existe un método mucho más rápido para maximizar la negentropía que el dado por el método del gradiente, y puede encontrarse usando un algoritmo de punto fijo. El algoritmo FastICA resultante encuentra un dirección, es decir, un vector unitario **w**, tal que la proyección de $\mathbf{w}^T\mathbf{z}$ maximice la no-gaussianidad. Aquí la no-gaussianidad es medida por la aproximación de la negentropía $J(\mathbf{w}^T\mathbf{z})$ dada en (3.27). Hay que recalcar que la varianza de $\mathbf{w}^T\mathbf{z}$ debe ser 1; para datos blancos, esto es equivalente a fijar la norma de **w** a 1.

Si miramos la expresión (2.45), ésta nos sugiere la siguiente iteración de punto fijo:

$$\mathbf{w} \leftarrow E\{\mathbf{z}g(\mathbf{w}^T\mathbf{z})\} \tag{3.29}$$

que tendría que ser seguida naturalmente de la normalización de **w**. El coeficiente γ no aparece porque de todas formas sería eliminado por la normalización.

La iteración descrita por (3.29) no tiene tan buenas propiedades de convergencia como las de FastICA usando kurtosis, porque los momentos no polinómicos no tienen las mismas propiedades algebraicas que tenían los cumulantes reales como kurtosis y que facilitaban la convergencia. Por lo tanto tenemos que modificar esa iteración, lo hacemos porque podemos añadir **w**, multiplicado por alguna constante α , a ambos lados de (2.46) sin alterar los puntos fijos. Operando de esa forma tenemos:

$$\mathbf{w} = E\{\mathbf{z}g(\mathbf{w}^T\mathbf{z})\} \Leftrightarrow (1+\alpha)\mathbf{w} = E\{\mathbf{z}g(\mathbf{w}^T\mathbf{z})\} + \alpha\mathbf{w} \tag{3.30}$$

Entonces escogiendo α apropiadamente, puede ser posible obtener un algoritmo que converja tan rápido como el algoritmo de punto fijo usando kurtosis. De hecho tal valor de α encontrarse usando el método de aproximación de Newton. No detallaremos el proceso completo, pero operando de esta forma se llega a la iteración de punto fijo básica FastICA, que es lo que nos interesa:

$$\mathbf{w} \leftarrow E\{\mathbf{z}g(\mathbf{w}^T\mathbf{z})\} - E\{g'(\mathbf{w}^T\mathbf{z})\}\mathbf{w} \tag{3.31}$$

3.4.2 Estimación de máxima verosimilitud.

Una forma muy popular de estimar el modelo ICA es la estimación de máxima verosimilitud (ML). Es éste un método fundamental en la estimación estadística, como ya sabemos. Primero veremos brevemente en qué consiste el método de máxima verosimilitud y luego pasaremos a su aplicación en ICA.

El estimador de máxima verosimilitud supone que los parámetros desconocidos $\boldsymbol{\theta}$ son constantes o no hay información a priori disponible sobre ellos. El estimador ML tiene una serie de propiedades que lo convierten en una elección deseable, especialmente cuando el número de muestras es grande. Ha sido aplicado con éxito a una gran variedad de problemas en muchas áreas de aplicación.

El estimador $\hat{\boldsymbol{\theta}}_{ML}$ del vector $\boldsymbol{\theta}$ es el valor que maximiza la función de verosimilitud

$$p(\mathbf{x}_T \mid \boldsymbol{\theta}) = p(x(1), x(2), \ldots, x(T) \mid \boldsymbol{\theta}) \tag{3.32}$$

de las medidas $x(1), x(2), \ldots, x(T)$. Ese valor del estimador corresponde al que hace las medidas obtenidas más probables. Debido a que muchas funciones de

densidad contienen una función exponencial, normalmente es más conveniente trabajar con el logaritmo de la función de verosimilitud: $\ln p(\mathbf{x}_T \mid \boldsymbol{\theta})$. Claramente, el estimador $\hat{\boldsymbol{\theta}}_{ML}$ también maximiza la función de verosimilitud logarítmica. El estimador de máxima verosimilitud se encuentra normalmente en las soluciones de la ecuación de verosimilitud

$$\frac{\partial}{\partial \boldsymbol{\theta}} \ln p(\mathbf{x}_T \mid \boldsymbol{\theta}) \Big|_{\theta = \theta_{ML}} = 0 \tag{3.33}$$

La ecuación de verosimilitud da los valores de $\boldsymbol{\theta}$ que maximizan (o minimizan) la función de verosimilitud.

3.4.2.1 La verosimilitud en el modelo ICA.

La densidad de probabilidad p_x del vector de mezclas puede hallarse sin demasiada dificultad en el modelo ICA. Para hacerlo nos basamos en el conocido método para derivar la densidad de una transformación lineal, sabiendo la de sus componentes. En este caso, el modelo es el habitual **x=As**. Si conocemos la distribución de los componentes de **s**, la densidad de **x** puede formularse como:

$$p_x(\mathbf{x}) = |\det \mathbf{B}| p_s(\mathbf{s}) = |\det \mathbf{B}| \prod_i p_i(s_i) \tag{3.34}$$

donde $\mathbf{B} = \mathbf{A}^{-1}$, y p_i denota las densidades de los componentes independientes. Esto puede expresarse en función de las filas de **B** y de **x**, como:

$$p_x(\mathbf{x}) = |\det \mathbf{B}| \prod_i p_i(\mathbf{b}_i^T \mathbf{x}) \tag{3.35}$$

Supongamos que tenemos *T* observaciones de **x**, denotadas por **x**(1),**x**(2),…,**x**(*T*). Entonces la verosimilitud puede obtenerse como el producto de estas densidades evaluadas en los *T* puntos. Aquí la llamaremos *L* y la consideraremos función de **B**:

$$L(\mathbf{B}) = \prod_{t=1}^{T} \prod_{i=1}^{n} p_i(\mathbf{b}_i^T \mathbf{x}(t)) |\det \mathbf{B}| \tag{3.36}$$

La verosimilitud logarítmica es:

$$\log L(\mathbf{B}) = \sum_{t=1}^{T} \sum_{i=1}^{n} \log p_i(\mathbf{b}_i^T \mathbf{x}(t)) + T \log |\det \mathbf{B}| \tag{3.37}$$

La única diferencia es que en muchas ocasiones esta última expresión es algebraicamente más simple, pero no cambia la localización de los máximos, por lo que se podría trabajar indistintamente con ambas formas sin alterar los resultados.

Para simplificar la notación y hacerla más acorde a la que venimos usando podemos sustituir las sumas sobre el índice *t* por operadores esperanza, y dividir por *T*, así obtenemos:

$$\frac{1}{T}\log L(\mathbf{B}) = E\left\{\sum_{i=1}^{n} \log p_i\left(\mathbf{b}_i^T \mathbf{x}\right)\right\} + \log|\det \mathbf{B}| \tag{3.38}$$

La esperanza aquí no es la teórica sino el promedio calculado a partir de las muestras observadas. De todas formas, en los algoritmos todas las medias estadísticas se sustituyen por promedios muestrales, por lo tanto la diferencia es puramente teórica.

Hasta ahora hemos expresado la verosimilitud en función de los parámetros del modelo, que son los elementos de la matriz de mezcla. Puesto que hemos supuesto que esta matriz será invertible, podemos calcular **B** fácilmente como su inversa. Pero hay otra cosa que usamos y que no hemos tenido en cuenta hasta ahora, es la densidad de los componentes independientes. La verosimilitud es función también de estas densidades, y esto complica bastante el problema porque la estimación de las densidades es, en general, un problema no paramétrico. No paramétrico significa que no puede reducirse a estimar un conjunto finito de parámetros. De hecho, el número de parámetros que habría que estimar sería infinito o muy grande en la práctica. Entonces hay dos formas de evitar esta estimación.

La primera es simplemente conocer de antemano las densidades de probabilidad de los componentes independientes, usando algún tipo de información a priori. En ese caso la función de verosimilitud sería función solamente de **B**. Si los errores en la especificación de las densidades tienen poca importancia en el estimador, este procedimiento nos dará resultados razonablemente buenos.

La otra posibilidad es aproximar las densidades de los componentes independientes por una familia de densidades que estén definidas por un número limitado de parámetros. Se el número de parámetros de la familia de densidades es muy grande, la verdad es que ganamos muy poco, puesto que la meta es justo no tener que hacer eso. Afortunadamente, es posible usar una familia muy simple de densidades para estimar el modelo ICA y obtener una solución sencilla. No hablaremos con más profundidad de cómo hacer esto puesto que podríamos desviarnos del propósito general de este texto.

3.4.2.2 Algoritmos para la estimación de la máxima verosimilitud.

Presentaremos ahora con brevedad algunos algoritmos que serán útiles para conseguir la maximización numérica de la verosimilitud.

Algoritmo de Bell-Sejnowski. Los algoritmos más simples para maximizar la verosimilitud se obtienen por métodos basados en el gradiente. Uno de esto es el que referenciamos aquí. Fácilmente podemos derivar el gradiente estocástico de la verosimilitud logarítmica de la expresión (3.36) como:

$$\frac{1}{T}\frac{\partial \log L}{\partial \mathbf{B}} = \left[\mathbf{B}^T\right]^{-1} + E\left\{\mathbf{g}(\mathbf{Bx})\mathbf{x}^T\right\} \tag{3.39}$$

$\mathbf{g}(\mathbf{y}) = (g_1(y_1),\ldots,g_n(y_n))$ es una función vectorial que consiste en las funciones g_i de las distribuciones de s_i, y se definen como:

$$g_i = (\log p_i)' = \frac{p'_i}{p_i} \tag{3.40}$$

Inmediatamente se obtiene el siguiente algoritmo para la estimación ML:

$$\Delta \mathbf{B} \propto \left[\mathbf{B}^T\right]^{-1} + E\left\{\mathbf{g}(\mathbf{Bx})\mathbf{x}^T\right\} \tag{3.41}$$

Se puede usar también una versión estocástica de este algoritmo. Esto quiere decir que omitimos la media estadística, y sólo usamos un punto de los datos en cada paso del algoritmo:

$$\Delta \mathbf{B} \propto \left[\mathbf{B}^T\right]^{-1} + \mathbf{g}(\mathbf{Bx})\mathbf{x}^T \tag{3.42}$$

Para más información sobre este algoritmo se puede consultar en [Bell95]. El algoritmo de (3.41) converge muy lentamente, sobre todo debido a que se necesita invertir la matriz **B** en cada paso. De todas formas la convergencia puede mejorarse blanqueando los datos, y sobre todo usando el gradiente natural, del que hablamos a continuación.

Algoritmo del gradiente natural. El método del gradiente natural o relativo simplifica la maximización de la verosimilitud considerablemente. Podemos partir de la expresión (3.41) multiplicando el lado derecho por $\mathbf{BB}^T$, para obtener:

$$\Delta \mathbf{B} \propto \left(\mathbf{I} + E\left\{\mathbf{g}(\mathbf{y})\mathbf{y}^T\right\}\right)\mathbf{B} \tag{3.43}$$

Este algoritmo puede interpretarse como decorrelación no lineal. La idea es que el algoritmo converge cuando $E\left\{\mathbf{g}(\mathbf{y})\mathbf{y}^T\right\} = -\mathbf{I}$, lo que significa que los y_i y los $g_j(y_j)$ están incorrelados para $i \neq j$. Como siempre, para obtener la versión on-line de este algoritmo basta con quitar el operador esperanza y vamos iterando punto a punto.

Algoritmo rápido de punto fijo. La verosimilitud también puede maximizarse usando un método de punto fijo. El algoritmo FastICA que presentamos en apartados anteriores puede aplicarse directamente a la maximización de la verosimilitud. Haciendo las transformaciones oportunas y que aquí no mostraremos para no distraernos en desarrollos matemáticos engorrosos, se obtiene la iteración básica de FastICA:

$$\mathbf{B} \leftarrow \mathbf{B} + diag(\alpha_i)\left[diag(\beta_i) + E\left\{\mathbf{g}(\mathbf{y})\mathbf{y}^T\right\}\right]\mathbf{B} \tag{3.44}$$

donde tenemos que **y=Bx**, $\beta_i = -E\{y_i g(y_i)\}$, y $\alpha_i = -1/(\beta_i + E\{g'(y_i)\})$.

Después de cada paso, hay que actualizar la matriz **B** como

$$\mathbf{B} \leftarrow \left(\mathbf{B}\mathbf{C}\mathbf{B}^T\right)^{-1/2}\mathbf{B} \tag{3.45}$$

donde $\mathbf{C} = E\{\mathbf{x}\mathbf{x}^T\}$ es la matriz de correlación de los datos.

3.4.3 Minimización de la información mutua.

Un enfoque importante para el análisis de componentes independientes, inspirado en la teoría de la información, es la minimización de la información mutua. Este será el último criterio que veremos para llevar a cabo ICA.

La motivación de este enfoque es que a veces podría ser un poco irreal suponer que los datos siguen el modelo ICA. Por lo tanto, estaría bien desarrollar una solución que no suponga nada sobre los datos. Entonces lo que queremos es una medida general de la dependencia de los componentes de un vector aleatorio. Usando esta medida, podríamos definir ICA como la descomposición que minimiza tal dependencia, es decir, la que los hace lo más independientes que sea posible. Bien, pues para desarrollar esta idea haremos uso de la información mutua, que es la forma de medir la dependencia en la teoría de la información. Una de las principales utilidades de la información mutua es que sirve como marco unificador para muchos principios de estimación, en particular los ya vistos de máxima verosimilitud y maximización de la no-gaussianidad.

3.4.3.1 Definición de ICA usando la información mutua.

Los lectores no familiarizados con la teoría de la información pueden encontrar interesante leer el capítulo 5 del libro [ICA01]. Daremos aquí brevemente las definiciones básicas de la teoría de la información. La entropía diferencial *H* de un vector aleatorio **y** con densidad $p(\mathbf{y})$ se define como:

$$H(\mathbf{y}) = -\int p(\mathbf{y}) \log p(\mathbf{y}) \mathrm{d}\mathbf{y} \tag{3.46}$$

La entropía está estrechamente relacionada con la longitud del código del vector aleatorio. Una versión normalizada de la entropía viene dada por la negentropía *J*, que ya mostramos en la expresión (3.24). La información mutua *I* entre *m* variables aleatorias escalares $y_i, i = 1 \ldots m$, se define como sigue:

$$I(y_1, y_2, \ldots, y_m) = \sum_{i=1}^{m} H(y_i) - H(\mathbf{y}) \tag{3.47}$$

La información mutua es una medida natural de la independencia entre variables aleatorias, como dijimos antes, además nunca es negativa, y sólo es cero si las variables son estadísticamente independientes. La información mutua tiene en cuenta la completa dependencia de las variables, y no sólo la covarianza, como era el caso de PCA y otros métodos parecidos. Por lo tanto

podemos usar la información mutua como criterio para usar la representación ICA. Definiremos el ICA de un vector aleatorio **x** de la forma habitual, es decir, como una transformación invertible **s=Bx**, donde la matriz **B** es determinada de forma que la información mutua de los componentes transformados s_i sea minimizada. Si los datos siguen el modelo ICA, esto permite la estimación del modelo de los datos. Pero por otro lado no necesitamos suponer que los datos siguen el modelo en esta definición. En cualquier caso, la minimización de la información mutua puede interpretarse como el hallazgo de los componentes máximamente independientes.

3.4.3.2 Información mutua y no-gaussianidad.

Para una transformación invertible **y=Bx** tenemos:

$$I(y_1, y_2, \ldots, y_n) = \sum_i H(y_i) - H(\mathbf{x}) - \log|\det \mathbf{B}| \tag{3.48}$$

Ahora vamos a ver qué ocurre si forzamos a los y_i a que sean incorrelados y de varianza unidad. Esto significa que $E\{\mathbf{y}\mathbf{y}^T\} = \mathbf{B}E\{\mathbf{x}\mathbf{x}^T\}\mathbf{B}^T = \mathbf{I}$, lo que implica:

$$\det \mathbf{I} = 1 = \det\left(\mathbf{B}E\{\mathbf{x}\mathbf{x}^T\}\mathbf{B}^T\right) = \left(\det \mathbf{B}\right)\left(\det E\{\mathbf{x}\mathbf{x}^T\}\right)\left(\det \mathbf{B}^T\right) \tag{3.49}$$

y esto implica que $\det \mathbf{B}$ debe ser constante puesto que $\det E\{\mathbf{x}\mathbf{x}^T\}$ no depende de **B**. Más aún, para los y_i de varianza unidad, la entropía y la negentropía disienten sólo en una constante y el signo, como puede verse en (3.24). Así obtenemos:

$$I(y_1, y_2, \ldots, y_n) = \text{const} - \sum_i J(y_i) \tag{3.50}$$

donde el término constante no depende de **B**. Esto muestra la relación fundamental entre la negentropía y la información mutua.

Podemos observar en (3.50) que encontrar una transformación lineal invertible **B** que minimice la información mutua equivale a encontrar las direcciones en las que la negentropía es maximizada. Ya vimos que la negentropía es una medida de la no-gaussianidad. Así, (3.50) demuestra que la formulación de la estimación ICA mediante la minimización de la información mutua es equivalente a maximizar la suma de las no-gausssianidades de las estimas de los componentes independientes, cuando las estimas son forzadas a ser incorreladas.

Entonces se puede ver que la formulación de ICA como minimización de la información mutua da otra justificación rigurosa de la idea de encontrar las direcciones máximamente no-gaussianas que vimos en apartados anteriores. En la práctica, sin embargo, hay algunas diferencias entre estos dos criterios.

3.4.3.3 Información mutua y verosimilitud.

La información mutua y la verosimilitud están íntimamente relacionadas. Para ver mejor esta conexión existente entre ambas, consideremos la esperanza de la verosimilitud logarítmica que mostramos en (3.38):

$$\frac{1}{T} E\{\log L(\mathbf{B})\} = \sum_{i=1}^{n} E\{\log p_i(\mathbf{b}_i^T \mathbf{x})\} + \log|\det \mathbf{B}| \tag{3.51}$$

Si los p_i fueran iguales a las verdaderas funciones densidad de probabilidad de $\mathbf{b}_i^T \mathbf{x}$, el primer término sería igual a $-\sum_i H(\mathbf{b}_i^T \mathbf{x})$. Así la verosimilitud sería igual a la información muta dada en (3.48) negada, salvo por una constante aditiva dada por la entropía total de **x**. En la práctica, la relación puede ser aún más fuerte. Esto se debe a que no conocemos las distribuciones de los componentes independientes. Una solución razonable es estimar la desidad de $\mathbf{b}_i^T \mathbf{x}$ como una parte del método de estimación ML, y usarla como aproximación de la densidad de s_i. Entonces, los p_i en esta aproximación de la verosimilitud son de hecho iguales a las verdaderas *fdp* de $\mathbf{b}_i^T \mathbf{x}$.

En cambio, para aproximar la información mutua, podríamos tomar una aproximación fija de las densidades de y_i, e introducirlas en la definición de la entropía. Si denotamos las *fdp* por $G_i(y_i) = \log p_i(y_i)$, podemos aproximar (3.48) por:

$$I(y_1, y_2, \ldots, y_n) = -\sum_i E\{G_i(y_i)\} - \log|\det \mathbf{B}| - H(\mathbf{x}) \tag{3.52}$$

Esto también nos da un método alternativo para aproximar la información mutua diferente de la aproximación que usa la negentropía.

3.4.3.4 Algoritmos para la minimización de la información mutua.

Para usar la información mutua en la práctica, necesitamos algún método para estimarla o aproximarla a partir de los datos reales. Antes ya vimos dos métodos para aproximar la entropía mutua. Si usamos la información mutua, esto nos lleva a los mismos algoritmos que usamos para maximizar la no-gaussianidad o para estimar la máxima verosimilitud. Por ello no presentaremos ningún algoritmo nuevo en esta sección, basta con repasar los que ya mostramos.

3.5 Conclusiones.

Hemos intentado en este capítulo presentar al lector los fundamentos básicos del análisis de componentes independientes de una forma general. Empezamos hablando un poco de la separación ciega de fuentes (BSS), que es el problema más importante al que da solución ICA, para pasar a describir cómo se lleva a cabo la misma. Destacamos que lo más importante y principal novedad es que

se basa en la independencia de las fuentes que se pretenden separar, y otro punto importante es que supone la no-gaussianidad de las mismas, siendo imposible separarlas en caso contrario. Para medir la no-gaussianidad de las fuentes hemos visto que había varios criterios que se pueden emplear, haciendo uso de estadísticos de orden superior. Veíamos que la estimación de las fuentes era posible si se cumplían las dos suposiciones de independencia y no gaussianidad pero seguía habiendo dos ambigüedades: el orden de los componentes independientes estimados y el escalado de los mismos. Podemos intercambiar los componentes de orden y multiplicarlos por constantes sin que esto afecte al modelo, y por lo tanto ICA no puede solucionar estas indeterminaciones.

Hemos presentado tres métodos de llevar a cabo el análisis de componentes independientes, basados respectivamente los criterios de maximización de la no-gaussianidad, maximización de la verosimilitud y minimización de la información mutua. Para ello hemos introducido conceptos necesarios como el kurtosis y la negentropía, y hemos presentado algunos algoritmos que son utilizados frecuentemente para optimizar el éxito de estas técnicas, centrándonos en el método del gradiente y los algoritmos de punto fijo, ya que estos últimos son fácilmente implementables en computadores digitales.

Capítulo 4

La Transformada de Fourier de corta duración (STFT)

4.1. Introducción.

En esta sección vamos a centrarnos en un tipo especial de transformada de Fourier para señales discretas. Suponemos que el lector conoce los fundamentos del análisis en frecuencia de señales mediante estos métodos, si no fuera así, podría dirigirse al libro [Oppenheim89].

Como sabemos, la transformada discreta de Fourier (DFT) es una representación de la transformada de Fourier para señales de tiempo discreto y longitud finita. Debido a que ésta puede calcularse explícitamente mediante varios algoritmos eficientes, podemos afirmar que juega un papel central en una gran variedad de aplicaciones en el procesado de señales, incluyendo filtrado y análisis espectral.

En aplicaciones y algoritmos basados en la evaluación explícita de la transformada de Fourier, a menudo se desea trabajar con la transformada de Fourier de tiempo discreto (DTFT), mientras que la puede calcularse es la DFT. Para señales de longitud finita, la DFT provee muestras en el dominio de la frecuencia de la DTFT, y por lo tanto deben comprenderse las implicaciones de este muestreo. Además de esto, en muchas aplicaciones de filtrado y análisis espectral las señales no tienen longitud finita, o bien su longitud es mucho mayor que aquella que necesitaríamos que tuviera para trabajar con estas transformadas. Como veremos, esta inconsistencia entre el requerimiento de longitud finita de la DFT y la realidad de la longitud indefinida de las señales en la práctica puede solucionarse acudiendo a conceptos como el enventanado, el procesamiento de los datos por bloques y las transformadas de Fourier dependientes del tiempo.

4.2. Enventanado de señales.

Ya hemos dicho que a veces las señales son demasiado largas para aplicarles la DFT o incluso tienen una duración infinita o indeterminada a priori. Esto se soluciona mediante el análisis por tramas. Intuitivamente, esto significa dividir la señal en trozos más pequeños de la misma y tratarlos por separado como si de señales diferentes se tratase. Para “trocear” las señales se hace uso de las ventanas. Las ventanas no son más que señales con una duración finita que nos

servirán para ver solo una parte de la señal. Sea $w(m)$ la ventana que vamos a usar y $x(m)$ la señal que queremos enventanar. La señal enventanada será:

$$v(m) = x(m)w(m) \tag{4.1}$$

La señal $v(m)$ será entonces un "trozo" de $x(m)$ multiplicado por $w(m)$. El caso más sencillo en pricipio sería aquel en que

$$w(m) = \begin{cases} 1, & 0 \le m \le N-1 \\ 0 & eoc \end{cases} \tag{4.2}$$

Entonces $v(m)$ sería equivalente simplemente a tomar N muestras de la señal $x(m)$. Si fuéramos trasladando la ventana en tiempo podríamos obtener todos los puntos de $x(m)$ como $v(m) = x(m)w(m-n)$ y tendríamos la posibilidad de ir procesándolos por separado.

La importancia de las ventanas radica en que las características de inicio y finalización de las mismas permiten disminuir los efectos de las discontinuidades que se producen al enventanar las señales. Por este motivo habría que plantearse cúal es la ventana más adecuada que habrá que usar en cada caso concreto. Vamos a presentar ahora las ventanas más conocidas.

Rectangular:

$$w(m) = \begin{cases} 1 & 0 \le m \le N-1 \\ 0 & eoc \end{cases} \tag{4.2}$$

Hamming:

$$w(m) = \begin{cases} 0.54 - 0.46\cos\left(\dfrac{2\pi m}{N-1}\right) & 0 \le m \le N-1 \\ 0 & eoc \end{cases} \tag{4.3}$$

Hanning:

$$w(m) = \begin{cases} 0.5\left(1 - \cos\left(\dfrac{2\pi m}{N-1}\right)\right) & 0 \le m \le N-1 \\ 0 & eoc \end{cases} \tag{4.4}$$

Blackman:

$$w(m) = \begin{cases} 0.42 - 0.5\cos\left(\dfrac{2\pi m}{N-1}\right) + 0.08\cos\left(\dfrac{4\pi m}{N-1}\right) & 0 \le m \le N-1 \\ 0 & eoc \end{cases} \tag{4.5}$$

Barlett (triangular):

$$w(m) = \begin{cases} 2m/N & 0 \le m \le N/2 \\ a(N - m) & N/2 < m \le N - 1 \end{cases} \tag{4.6}$$

En la siguiente figura podemos ver la representación temporal de todas las ventanas que acabamos de presentar:

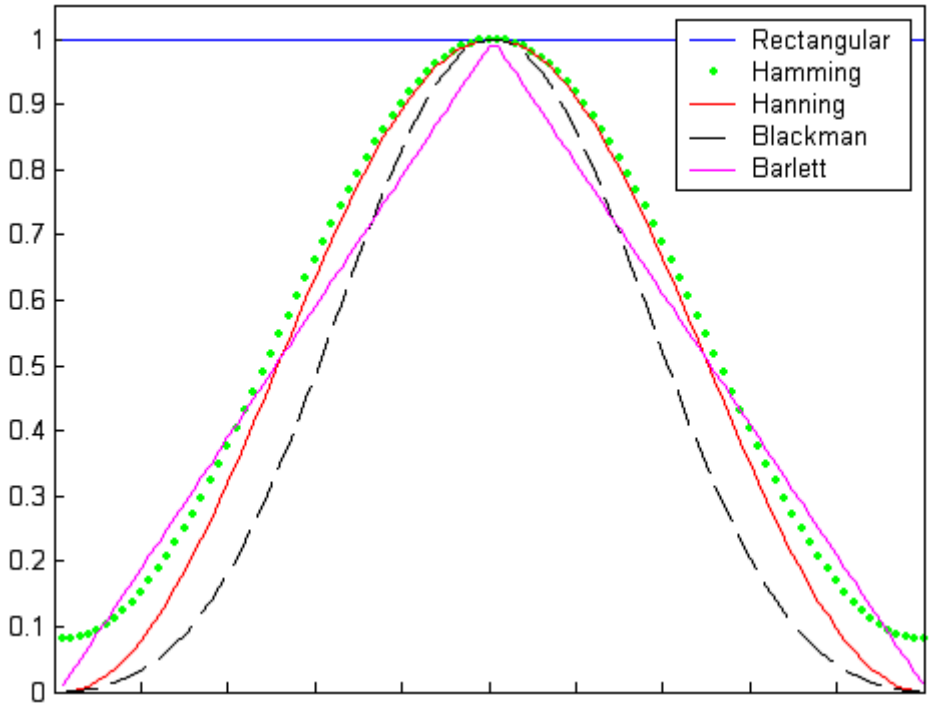


Figura 4.1 *Ventanas en el dominio del tiempo.*

Pero no sólo es importante la forma de las ventanas en el dominio del tiempo, sino que aún tiene más importancia su espectro, es decir, sus características frecuenciales, ya que si no son adecuadas en este dominio, deformarán las señales que se enventanen, haciendo difícil o imposible su correcta reconstrucción. Podemos observar el espectro de estas ventanas (en escala logarítmica) en las figuras (4.2) a (4.6).

Podemos fijarnos en dos características del espectro de las ventanas que serán fundamentales para decantarnos por una u otra en nuestra elección en cada caso particular: estas son la anchura del lóbulo principal, que determinará la resolución en frecuencia, y la buena atenuación de los lóbulos laterales frente al principal, que evitará la distorsión en la forma y envolvente del espectro de la señal enventanada.

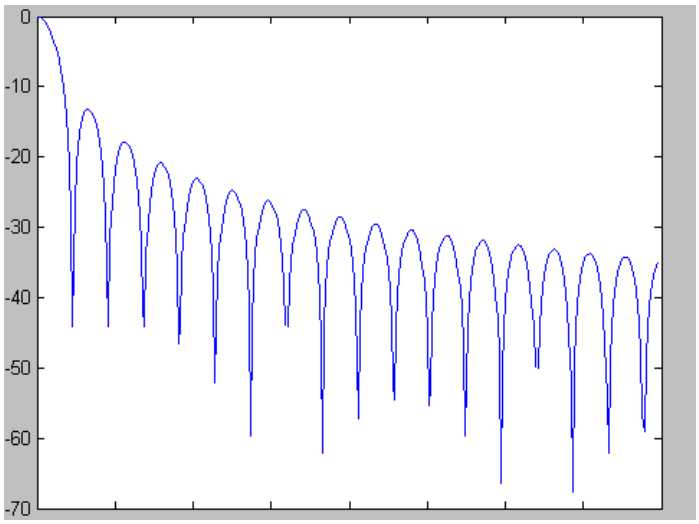


Figura 4.2 *Espectro de la ventana rectangular.*

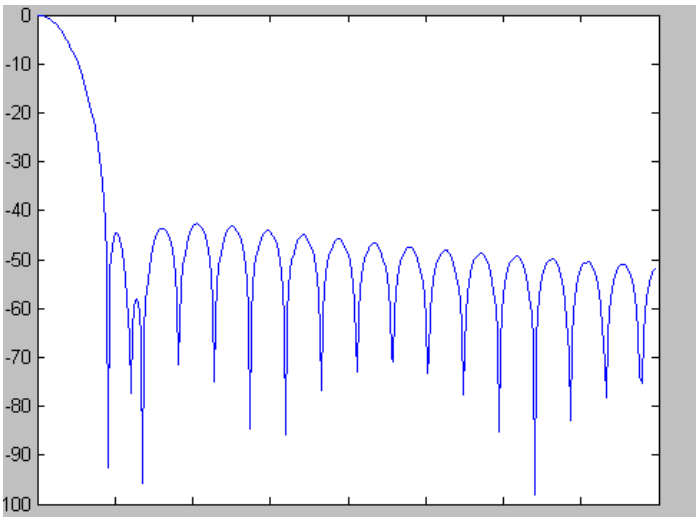


Figura 4.3 *Espectro de la ventana hamming.*

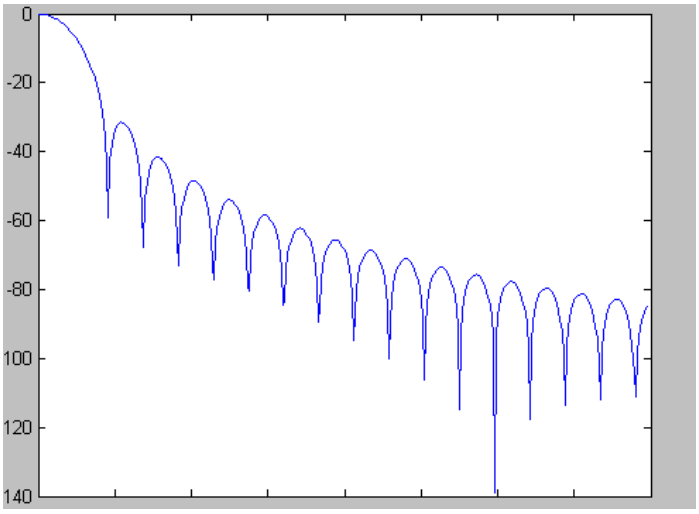


Figura 4.4 *Espectro de la ventana hanning*

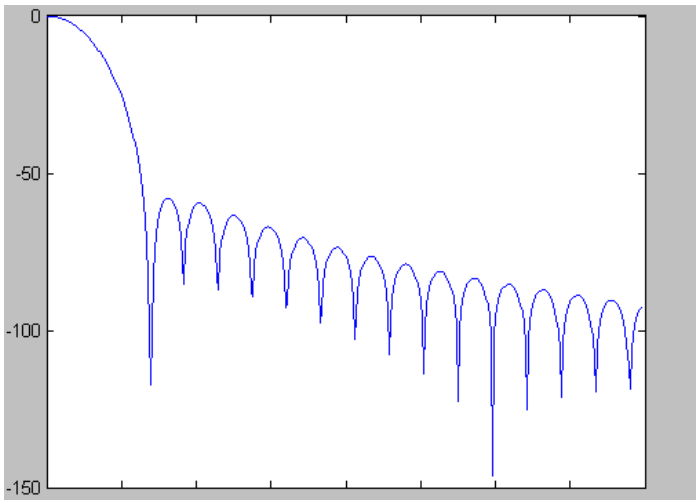


Figura 4.5 *Espectro de la ventana blackman.*

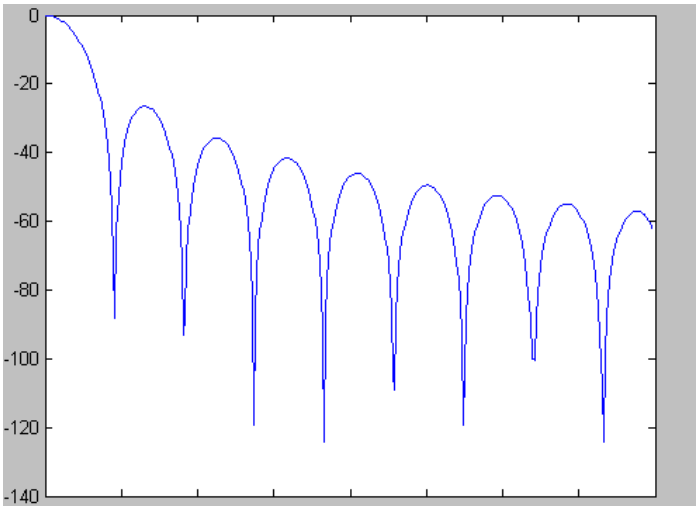


Figura 4.6 *Espectro de la ventana triangular.*

Los valores de estos dos parámetros de interés (que pueden verse en la figura 4.7) se muestran en el la tabla (4.1) para las diferentes ventanas que hemos visto (N es la longitud de la ventana y f_m la frecuencia de mu
estreo).

Ventana	**Δf**	**ΔL(dB)**
Rectangular	$2f_m / N$	-13
Hamming	$4f_m /(N - 1)$	-41
Hanning	$4f_m /(N - 1)$	-31
Blackman	$6f_m /(N - 1)$	-57
Triangular	$4f_m /(N - 1)$	-25

Tabla 4.1 *Anchura del lóbulo principal y atenuación del lóbulo secundario para las principales ventanas.*

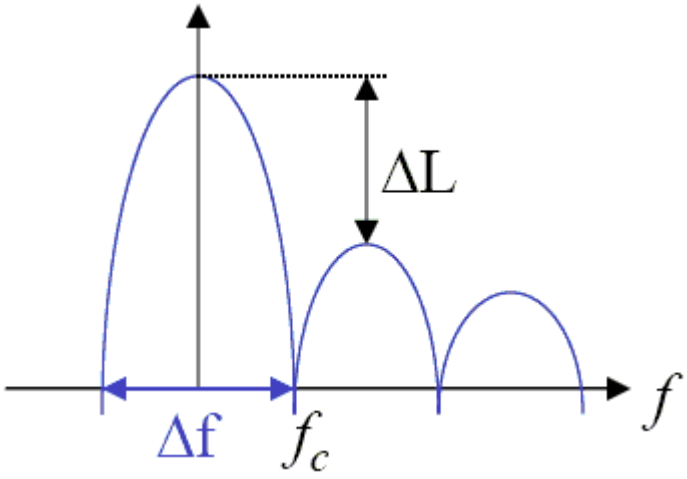


Figura 4.7 *Anchura del lóbulo principal y atenuación del lóbulo secundario.*

4.3 Definición de la transformada STFT.

Una vez vista de forma general la base del enventanado vamos a utilizar esto en una nueva transformada de Fourier basada en la DFT. En la práctica, hay muchas aplicaciones en las que las propiedades de la señal que se trata cambian muy rápidamente con el tiempo. Por ejemplo, esto sucede con señales no estacionarias tales como las de radar, sonar, voz y señales de comunicaciones. Pues bien, en estos casos calcular una única DFT para toda la señal no es suficiente, además de la dificultad añadida de que ésta podría ser larguísima siendo imposible de tratar en la práctica, ya que suelen usarse computadores digitales con una capacidad de cálculo y almacenamiento limitados. Todo ello nos guía hacia el concepto de transformada de Fourier de corta duración o STFT (Short-Time Fourier Transform).

La STFT de una señal $x(m)$ se define como:

$$X(n,\omega) = \sum_{m=-\infty}^{\infty} x(m)w(n-m)e^{-j\omega m} \tag{4.7}$$

donde $w(m)$ es la ventana. En la STFT, la secuencia unidimensional $x(m)$, función de una variable discreta, es transformada en una función bidimensional de la variable n, que es discreta, y de la frecuencia ω, que es continua. Hay que darse cuenta de que la STFT es periódica en ω con periodo 2π, y por lo tanto sólo tendremos que considerar los valores incluidos en $0 \le \omega \le 2\pi$, o cualquier otro intervalo de longitud 2π.

Teniendo en cuenta la simetría de las ventanas, la ecuación (4.7) puede rescribirse como:

$$X(n,\omega) = \sum_{m=-\infty}^{\infty} x(n+m)w(m)e^{-j\omega m} \tag{4.8}$$

De esta forma, (4.8) puede interpretarse como la transformada de Fourier de la señal desplazada $x(m+n)$, y vista a través de la ventana $w(m)$. La ventana tendría un origen fijo, y según n va cambiando, la señal se desliza pasando a través de la ventana de forma que para cada valor de n vemos una porción diferente de la señal.

Vamos a ver ahora un ejemplo para una señal con una modulación lineal en frecuencia. Esta señal suele llamarse "chirp", y es de la forma:

$$x(m) = \cos(\omega_0 m^2) \qquad (4.9)$$

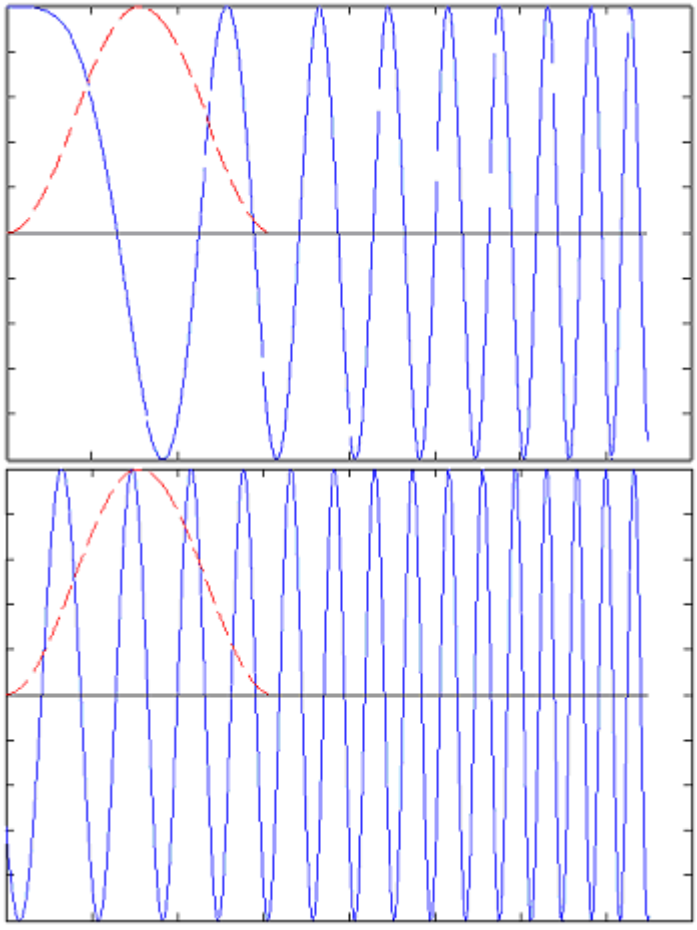


Figura 4.8 *Dos segmentos de la señal chirp con la ventana (en rojo) superpuesta.*

Podemos observar claramente el crecimiento lineal de la frecuencia según transcurre el tiempo en la figura (4.9), que se corresponde a la magnitud de la transformada STFT de la señal que estamos tratando. El eje vertical es proporcional a la frecuencia y el horizontal al tiempo. La magnitud de la transformada STFT se representa por la oscuridad del color.

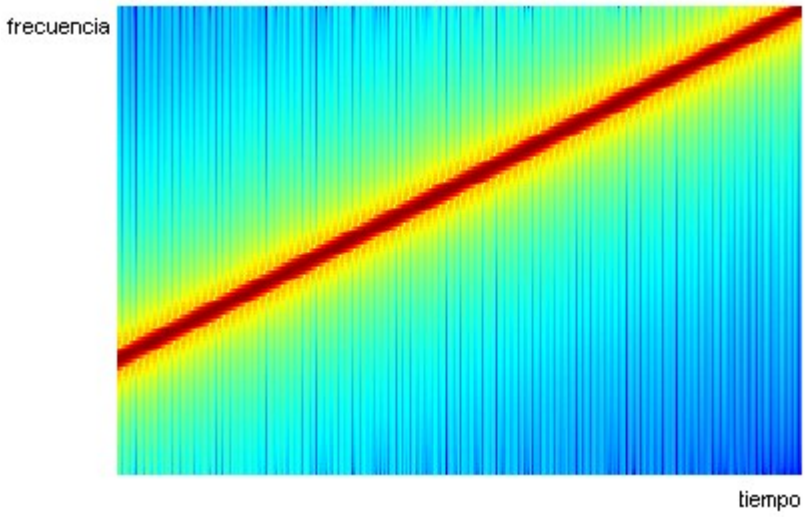


Figura 4.9 *Magnitud de la transformada STFT de* $x(m) = \cos(\omega_0 m^2)$ *usando una ventana Hamming.*

Vamos a ver ahora las dos posibles interpretaciones de esta transformada, que quizás sean de ayuda para su mejor comprensión.

Primera interpretación: n fijo, ω varía.

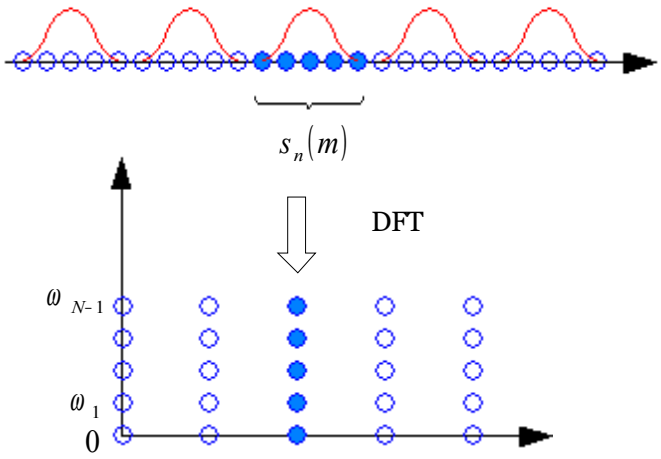


Figura 4.10 *Primera interpretación de la STFT.*

En este primer caso podemos ver la STFT como la transformada de Fourier de la señal $s_n(m) = s(m)w(n-m)$. De forma que tendríamos:

$$S(n,\omega) = \sum_{m=-\infty}^{\infty} s_n(m)e^{-j\omega m} \tag{4.10}$$

Esto nos lleva a darnos cuenta de que entonces podríamos intentar hacer la transformada inversa de esa expresión y estaríamos hallando la señal $s_n(m)$.

La frecuencia, que hasta ahora hemos considerado continua, en la práctica no lo es, ya que si queremos calcular esta transformada en computadores digitales es imposible que lo sea. Si muestreamos $S(n,\omega)$ en N frecuencias igualmente espaciadas $\omega_k = k2\pi/N$, siendo N mayor que la longitud de la ventana, entonces podemos recuperar la señal original partiendo de la transformada STFT muestreada.

Sabiendo esto podemos intentar reconstruir la señal de la siguiente forma:

$$S(n,\omega_k) = \sum_{m=-\infty}^{\infty} s_n(m)e^{-j\omega_k m} \leftrightarrow s_n(m) = \frac{1}{N}\sum_{k=0}^{N-1} S(n,\omega_k)e^{j\omega_k m} \tag{4.11}$$

$$\text{en } n = m \quad s_n(n) = w(0)s(n) \Rightarrow s(n) = \frac{1}{w(0)N}\sum_{k=0}^{N-1} S(n,\omega_k)e^{j\omega_k n} \tag{4.12}$$

En este caso $S(n,\omega_k)$ es la DFT de la señal enventanada $s_n(m) = s(m)w(n-m)$. Usando la transformada inversa obtenemos la expresión (4.12). Lo más importante de este punto de vista es que la longitud de la ventana es finita y por eso hay que tomar al menos tantas muestras en frecuencia como muestras no nulas tenga la ventana.

Una ilustración de esta interpretación se muestra en la figura (4.10).

Segunda interpretación: n varía, ω fijo.

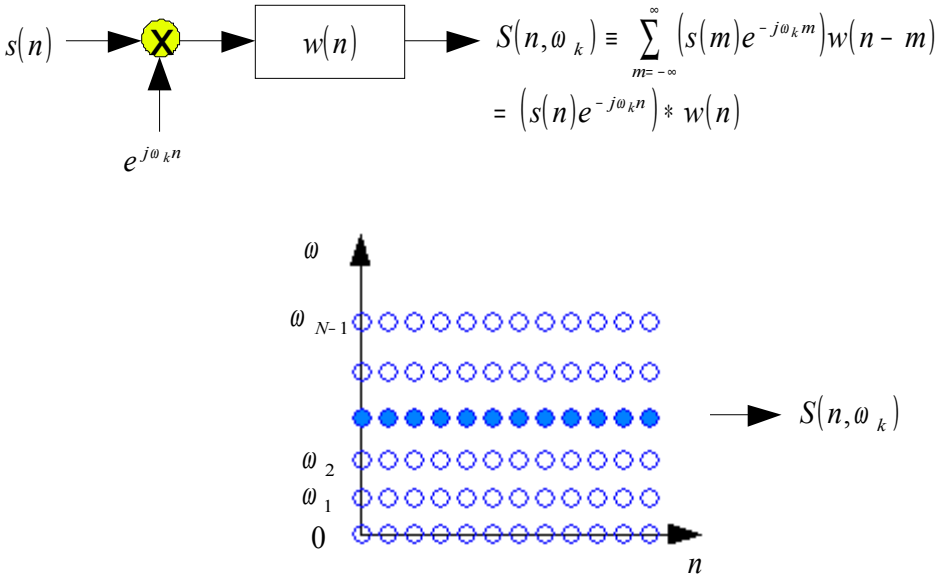


Figura 4.11 *Segunda interpretación de la STFT.*

La segunda forma de interpretar la STFT equivale a ver la transformada para cada frecuencia como el paso de la señal (multiplicada por una exponencial compleja) por un filtro que tendría como respuesta impulsiva la expresión de la

ventana elegida. Implementando entonces un banco de *N* filtros como el mostrado en la figura (4.11) obtendríamos $S(n,\omega_k)$.

4.4 Transformada STFT inversa mediante la técnica de overlap-add.

Ya hemos visto la expresión de la transformada STFT. En la práctica, es decir, cuando trabajamos en computadores digitales, se suele calcular usando la FFT (Fast Fourier Transform), que es una forma más rápida de operar, ya que las convoluciones en tiempo se pueden implementar como multiplicaciones en frecuencia. Para más información sobre la FFT ver el libro [Oppenheim89].

Pues bien, lo que se hace es coger cada tramo enventanado de la señal que queremos transformar y aplicarle la FFT de *NFFT* puntos. Para hacer esto requerimos que el número de puntos *NFFT* sea mayor o igual que la longitud de la ventana, es decir, si llamamos *L* a la longitud de dicha ventana: $NFFT \geq L$. De esta forma si el número de puntos de la FFT no es igual al número de puntos de la ventana tendremos que aplicar zero-padding al tramo de señal enventanado, que consiste en añadir (*NFFT-L*) ceros al final del mismo y luego aplicar la FFT. Cabe destacar que la ventana puede seleccionar tramos de señal que solapen, es decir, puede haber varios puntos de la señal que sean seleccionados más de una vez al realizar la transformada. Esto se ilustra en la figura (4.12).

Entonces para calcular la STFT obtendríamos una columna de la misma por cada STFT, teniendo la matriz resultante *NFFT* filas para el caso general, aunque si la señal es real suelen tomarse sólo la mitad más una, porque las demás podrían obtenerse a partir de éstas mediante su compleja conjugada.

Una vez dicho todo esto, vamos a ver una serie de condiciones que nos facilitarán la reconstrucción perfecta de la señal original a partir de su transformada STFT. Sabemos, como ya hemos dicho, que la convolución en el dominio del tiempo equivale a una multiplicación en el dominio de la frecuencia. Sin embargo, el único problema es que la multiplicación en el dominio de la frecuencia se corresponde a una convolución cíclica en el dominio del tiempo cuando lo que nos gustaría es que fuera simplemente una convolución lineal. En la convolución cíclica, los puntos del final del bloque de muestras se suman a las del principio. Esto es conocido como aliasing temporal. El método “solapa y suma” (overlap-add) soluciona este problema usando una longitud de transformada que asegura que no habrá problemas de aliasing temporal, y la convolución cíclica se comportará como una convolución lineal.

La condición clave para que sea posible la reconstrucción de la señal partiendo de la STFT es que la suma de las ventanas desplazadas según se han usado para realizar la STFT sea exactamente igual a 1 en todos los puntos, es decir, que la señal $\sum_k w(m - kn)$ tenga valor 1 en todos sus puntos, siendo *n* el número de puntos que se desplaza la ventana para tomar el siguiente segmento de la señal

al calcular la STFT. Esto equivale a decir que las ventanas (de longitud *L*) tienen que solapar en (*L-n*) puntos.

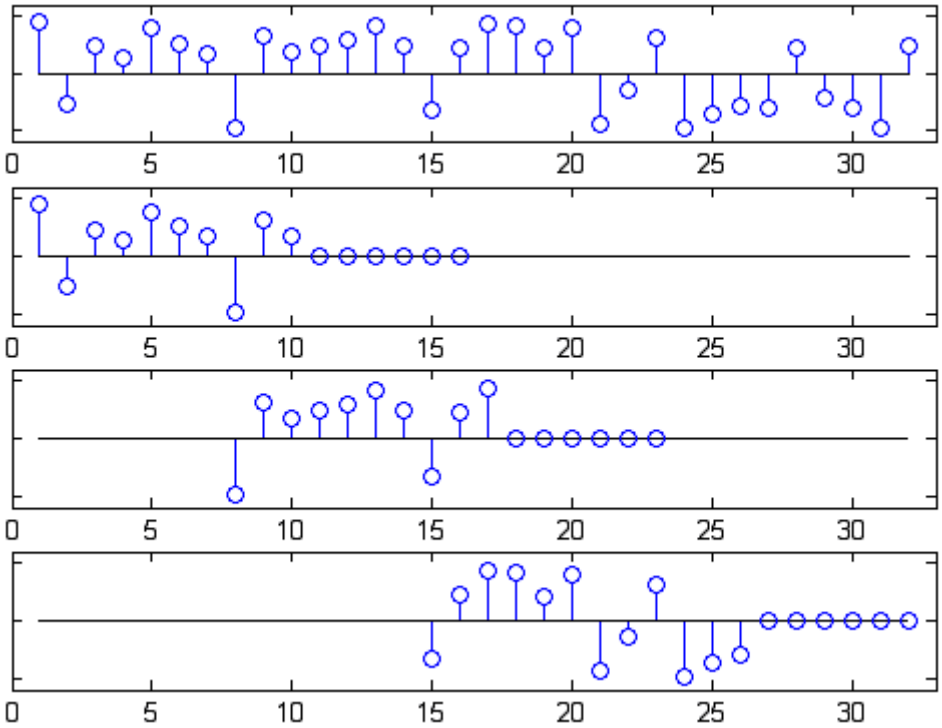


Figura 4.12 *Descomposición de una señal (1ª fila) en secciones de longitud 10 que solapan en 3 muestras y a las que se le ha aplicado zero-padding para calcular una FFT de 16 puntos.*

Vemos en la siguiente tabla cuál es la el número de puntos en que deben solapar cada tipo de ventanas para que cumplan la condición de overlap-add. Damos este dato como un porcentaje de la longitud de la ventana:

Ventana	*Solapamiento*
Rectangular	**0%**
Hamming	**50%**
Hanning	**50%**
Blackman	**66%**
Triangular	**50%**

Tabla 4.2 *Solapamiento necesario para conseguir la reconstrucción perfecta mediante overlap-add en cada tipo de ventana.*

De esta forma sólo tendremos que aplicar la transformada FFT inversa a las columnas de la matriz del espectrograma y sumar los tramos de señal obtenidos con el debido desplazamiento. Así obtendremos la señal original reconstruida en el dominio del tiempo.

Recordamos que la fórmula general para el cálculo de la STFT inversa viene dada por la expresión (4.12), que mostramos de nuevo a continuación:

$$s(n) = \frac{1}{w(0)N}\sum_{k=0}^{N-1} S(n,\omega_k)e^{j\omega_k n}$$

4.5 Análisis de la voz usando la transformada STFT.

La voz es una señal claramente no estacionaria. Una señal no estacionaria es una señal cuyas propiedades varían con el tiempo, por ejemplo, una suma de componentes senoidales con amplitudes, frecuencias o fases variantes en el tiempo. Sin embargo, para la voz, puede suponerse que las características de la señal permanecen en su esencia constantes en intervalos de tiempo del orden de 30 o 40 milisegundos. El contenido en frecuencia de la señal de voz puede extenderse en un rango de 15 kHz o aún más, pero la voz es claramente inteligible incluso si nos limitamos a una banda de frecuencias por debajo de los 3 kHz. Los sistemas de telefonía digital, por ejemplo, limitan la frecuencia más alta que se transmite a frecuencias alrededor de los 3 kHz. Usando una tasa de muestreo estándar de 8000 muestras/segundo, un intervalo de 40 ms equivale a 320 muestras.

La voz es producida por la excitación de un tubo acústico, el tracto vocal, que termina por un lado en los labios y por otro en la glotis. Cuando hablamos emitimos dos tipos fundamentales de sonidos diferentes:

- Sonoros: son producidos por la excitación del tracto vocal con pulsos cuasiperiódicos de aire a causa de abrir y cerrar la glotis. Se produce una vibración de las cuerdas vocales, dando lugar a una señal con una estructura periódica en tiempo. Este tipo de señal tiene una energía elevada debido a que el aire encuentra poca obstrucción en su salida al exterior. Su espectro decae hacia las altas frecuencias, como podemos ver en la figura (4.13).

- Sordos: en este caso las cuerdas vocales no vibran, por ello el aire encuentra bastante obstrucción en su salida hacia el exterior, dando lugar a una estructura aleatoria sin periodicidades marcadas, es decir, con aspecto de ruido. Su espectro, sin embargo, es más compensado en frecuencia. En la figura (4.14) mostramos la expresión temporal, así como el espectro de un ejemplo de este tipo de sonidos.

Hay más clasificaciones dentro de estos dos tipos pero no vamos a hablar de ellas aquí porque no nos serán de mucha utilidad.

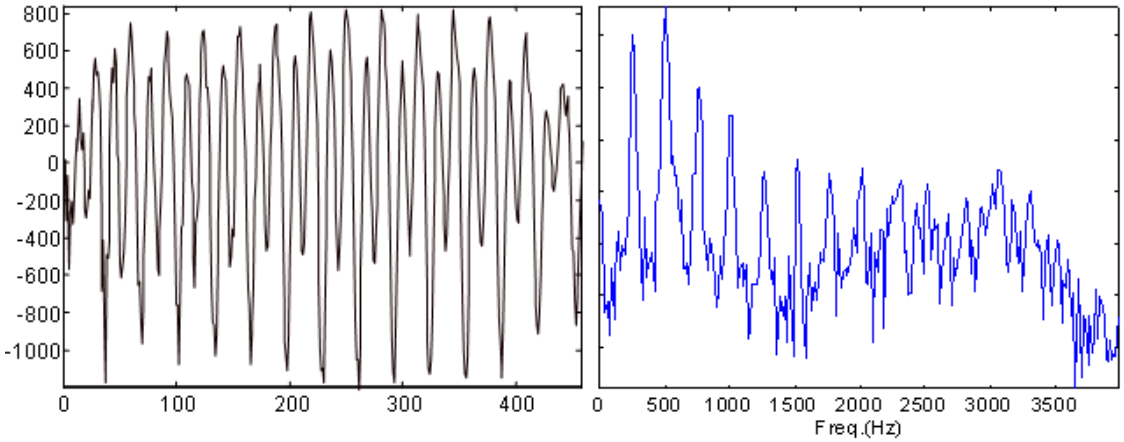


Figura 4.13 *Tramo sonoro de voz correspondiente al fonema /e/ y su correspondiente espectro tomado con una tasa de muestreo de 460 muestras/segundo usando una ventana hamming y sin solapamiento.*

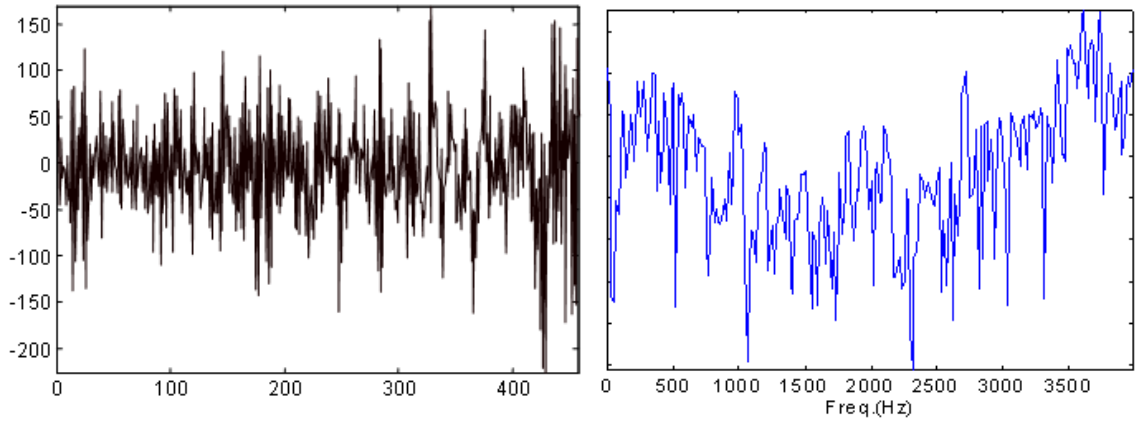


Figura 4.14 *Tramo sordo de voz correspondiente al fonema /s/ y su correspondiente espectro tomado con una tasa de muestreo de 460 muestras/segundo usando una ventana hamming y sin solapamiento.*

La figura (4.15) muestra que la forma de onda consiste en una secuencia de segmentos de voz cuasi-periódicos intercalados con segmentos *sordos* de voz con apariencia de ruido. La figura sugiere que si la longitud de la ventana no es demasiado larga, las propiedades de la señal no cambiarán apreciablemente desde el principio del segmento de voz hasta su final. Así, la DFT de un segmento de voz enventanado mostraría las propiedades en el dominio de la frecuencia de la señal correspondiente al intervalo de tiempo en que esté situada la ventana. Por ejemplo, si la longitud de la ventana es suficiente para que los armónicos sean observados, la DFT de un segmento de voz debería mostrar una serie de picos en múltiplos enteros de la frecuencia fundamental de la señal en ese intervalo. Por el contrario, si la ventana es demasiado corta, los armónicos no podrían observarse pero aun así la forma general del espectro sería evidente. Esto es típicamente un dilema entre la resolución en frecuencia y la resolución en tiempo en el análisis de señales no estacionarias. Pero tampoco podemos usar una ventana demasiado larga, pues si esto ocurre, las propiedades de la

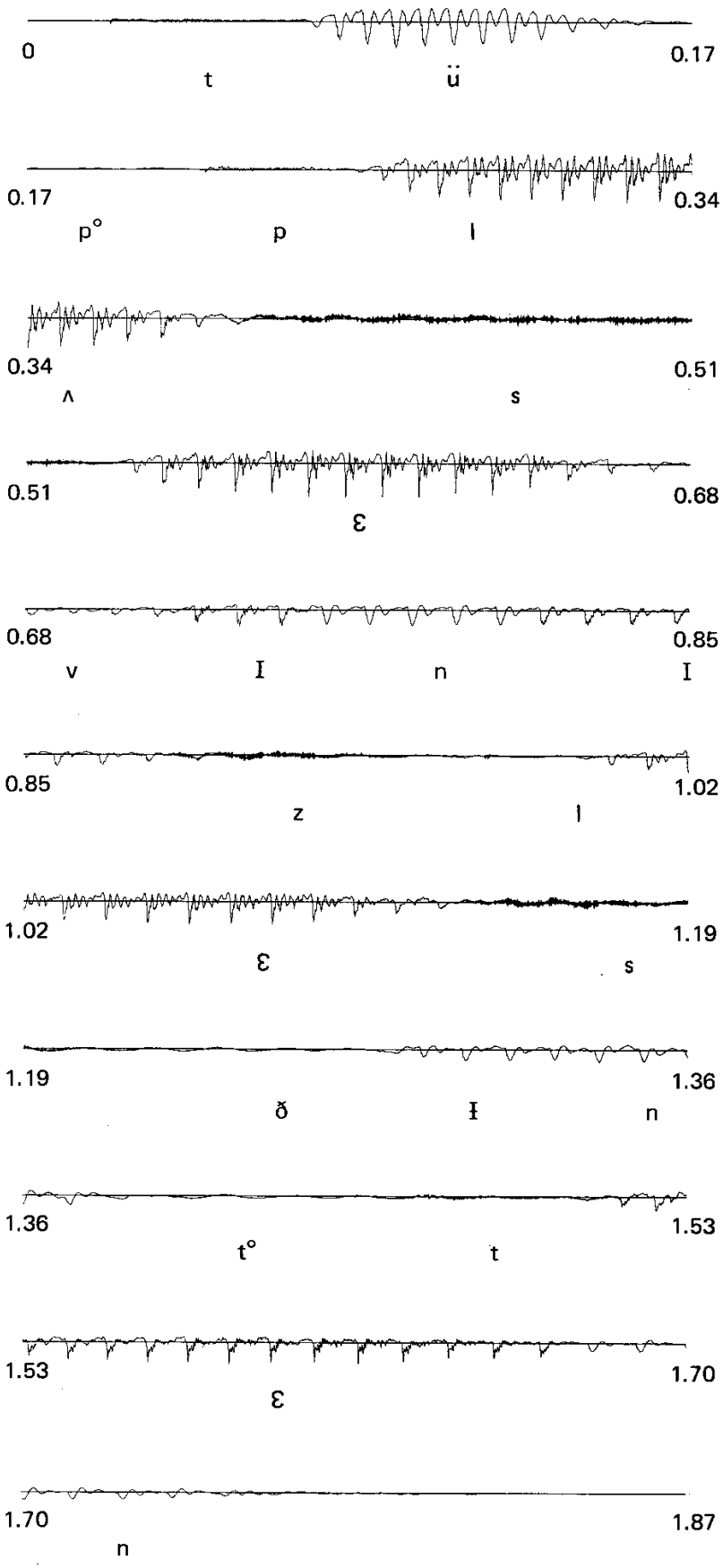


Figura 4.15 *Forma de onda de la frase "Two plus seven is less than ten". Cada línea dura 0.17 segundos. © Oppenheim.*

señal podrían cambiar demasiado durante el transcurso de la ventana. Si la ventana es demasiado corta, habrá que sacrificar la resolución de los componentes de banda estrecha.

Para ilustrar cómo sería la apariencia de la transformada STFT de una señal de voz, mostramos en la figura (4.16) el espectrograma de una grabación vocal. El espectrograma no es otra cosa que la magnitud al cuadrado de la STFT.

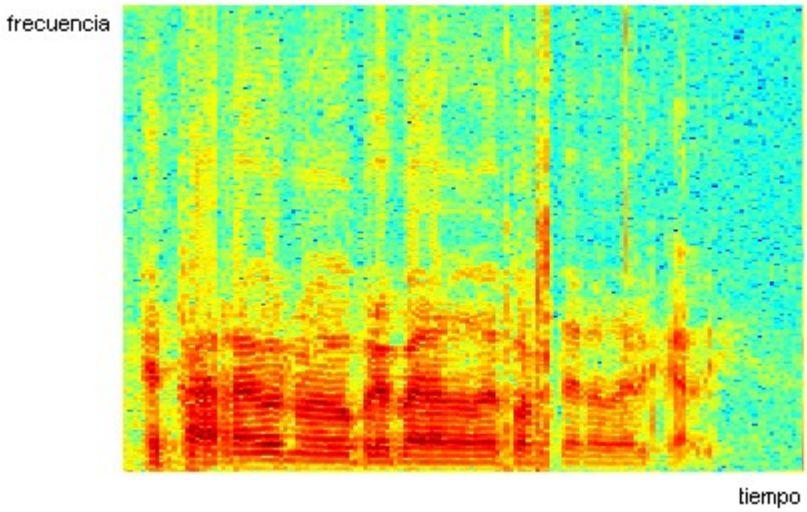


Figura 4.16 *Espectrograma de una señal de voz grabada con una tasa de muestreo de 16 kHz. La ventana usada es de 30 ms de duración. Eje vertical: frecuencia. Eje horizontal: tiempo.*

4.6 Conclusiones.

En este apartado hemos visto los conceptos necesarios para conocer y usar la STFT. Para ello hemos comenzado introduciendo el concepto de ventanas, resaltando su utilidad y presentando las más importantes, ya que son imprescindibles para calcular esta transformada. Hemos visto la expresión general de la STFT, que daba lugar a una señal bidimensional en el espacio tiempo-frecuencia:

$$X(n,\omega) = \sum_{m=-\infty}^{\infty} x(m)w(n-m)e^{-j\omega m}$$

interpretándola de dos formas diferentes para su mejor comprensión. Posteriormente hemos explicado de forma cualitativa cómo llevar a cabo la reconstrucción de las señales transformadas al dominio temporal mediante el método de overlap-add, presentando también las condiciones y expresiones necesarias para llevar a cabo este fin.

Finalmente hemos enumerado las propiedades de la señal de voz, centrándonos en las características que hacen que la transformada STFT sea la idónea para su tratamiento, debido principalmente a que es una señal estacionaria en periodos cortos de tiempo.

Capítulo 5

Modelos de mezcla de la voz

5.1 Introducción.

Durante la sección dedicada a ICA, nos hemos referido continuamente a conjuntos de variables aleatorias, que hemos usado para hallar los componentes independientes subyacentes, o simplemente para hacer algún tipo de transformación sobre las mismas que pudiera resultarnos útil de cara al propósito que estuviéramos buscando en ese momento. Sin embargo, nunca hemos concretado el origen de esas variables aleatorias. Hemos tratado el tema de la forma más general posible viendo problemas y soluciones para los casos genéricos más habituales. Expresábamos los datos como vectores, y las mezclas se hacían mediante combinaciones lineales que venían dadas por una matriz invertible que multiplicaba a esos vectores de datos proporcionando otros vectores diferentes, que eran a los que podíamos tener acceso mediante medidas de los mismos.

Pues bien, ahora vamos a centrarnos en un tipo de señal muy específico: la voz humana. Veremos brevemente algunas de las características más importantes de la misma, ya que luego nos serán de utilidad, y expondremos los modelos matemáticos que usaremos para simular el proceso físico de mezcla de estas señales en particular. La situación que vamos a tratar de describir de distintas formas es la que corresponde a varias personas situadas en distintos puntos del espacio emitiendo sonidos simultáneamente, que se graban mediante sensores (micrófonos) situados también en diferentes lugares. Las grabaciones resultantes serán en apariencia iguales entre si, pero tendrán leves diferencias, y se corresponderán con superposiciones de las voces de los interlocutores presentes, ya que no hay ninguna forma de grabar estas voces independientemente sin aislar físicamente en habitáculos distintos a las personas que las emiten.

Nos disponemos ahora a intentar expresar esas grabaciones de las mezclas en función de las voces independientes. El modelo matemático que habrá que adoptar dependerá en gran medida del entorno en que se realicen las grabaciones, puesto que, como veremos, influyen diversos factores tales como la presencia o no de paredes, el tipo de suelo, el movimiento de los interlocutores, etc.

5.2 Modelo de mezcla instantánea.

Para empezar con los modelos matemáticos de mezclas de señales de voz, comenzaremos con el modelo más sencillo que podemos plantear, es el de mezcla instantánea. Si planteamos el modelo de mezcla como un sistema donde las entradas son las señales acústicas que pronuncian las personas que hablan y las salidas son las grabaciones de los micrófonos, este modelo se implementaría tomando simplemente como entrada las señales de voz en el dominio temporal (vector **s**), siendo la salida una combinación lineal de las mismas (vector **x**). Los coeficientes de esta combinación lineal vendrían dados por una matriz **A**, de forma que si tenemos N señales fuente y M sensores (micrófonos), la matriz de mezcla **A** tendría dimensión $M \times N$. Esto es:

$$\mathbf{x} = \mathbf{As} \Leftrightarrow \begin{bmatrix} x_1 \\ \vdots \\ x_M \end{bmatrix} = \begin{bmatrix} a_{11} & \cdots & a_{1N} \\ \vdots & \ddots & \vdots \\ a_{M1} & \cdots & a_{MN} \end{bmatrix} \begin{bmatrix} s_1 \\ \vdots \\ s_N \end{bmatrix} \tag{5.1}$$

donde los s_j, $j = 1,\ldots,N$ y x_i, $i = 1,\ldots,M$ son respectivamente las muestras de las señales de voz $s_j(t)$ y de las mezclas $x_i(t)$, tomadas en los instantes de muestreo. Así pues, los vectores **s** y **x** contienen, para cada instante de tiempo (muestreado) las muestras de todas las señales fuente y de todas las mezclas resultantes.

Ni que decir tiene que este modelo no se corresponde con ninguna situación real, puesto que para buscar un caso teórico que se ajustase a ese modelo habría que suponer que las señales acústicas se propagan a una velocidad infinita, suposición que por supuesto nunca se cumplirá en la práctica.

5.3 Modelo convolutivo de mezcla.

5.3.1 Descripción cualitativa del modelo.

En este apartado proponemos la situación más general posible. Es un modelo que se adaptaría a cualquier situación acústica, ya sea al aire libre o en cualquier habitación y sin ninguna restricción en cuanto a la presencia de barreras físicas.

Sabemos que la mezcla de las fuentes sonoras en el aire es lineal y conlleva velocidad de propagación finita y reverberación. En ese caso el modelo instantáneo no es para nada el adecuado y hay que recurrir al modelo de mezcla convolutiva.

Supongamos una situación hipotética en la que haya una sola persona hablando en una habitación y pretenda grabarse con un micrófono. Trataremos de entender de una forma intuitiva y no rigurosa de qué señales estaría compuesta dicha grabación. Según esta persona emite sonidos, estos se propagarían en todas direcciones por la habitación con una determinada velocidad, hasta que se encuentren con obstáculos en su propagación (paredes por ejemplo). Estamos

entonces imaginándonos la voz como una serie de "rayos" que salen desde la localización de la persona que habla en variadas direcciones. Al encontrarse con obstáculos en su camino los rayos van a "rebotar" en éstos, cambiando su dirección. Esto se irá repitiendo hasta que la señal quede muy atenuada por estas "reflexiones" en los obstáculos y su amplitud ya sea despreciable. Bien, pues suponemos que van a darse muchísimos de estos rebotes, de forma que podemos pensar que la señal que el micrófono graba en cada instante de tiempo será equivalente a la suma de muchas réplicas de la voz del hablante con ligeras diferencias, es decir, habiendo recorrido cada una de esas réplicas un camino diferente hacia el micrófono, y llegando por lo tanto con una atenuación y un retraso diferente. Esto no lo percibimos al escuchar la grabación, ya que los retardos y las atenuaciones son pequeños, pero sí debemos tenerlo en cuenta para hacer un modelo matemático riguroso de la mezcla, y es precisamente eso lo que vamos a ver a continuación.

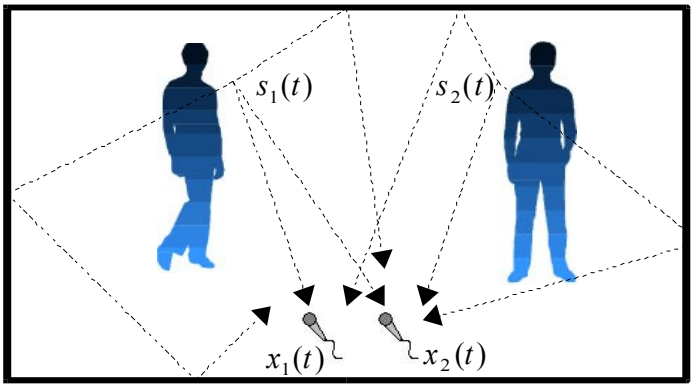


Figura 5.2 *Modelo de rayos en un entorno convolutivo.*

5.3.2 Formulación del modelo en el dominio temporal.

Como ya hemos dicho, ahora tendremos que tomar en consideración retrasos temporales y ecos, que no se contemplaban en el caso instantáneo. Entonces lo que se grabará será la suma de las señales de voz con versiones atenuadas y retrasadas de ellas mismas y de las demás voces presentes. Así pues, si comparamos este caso con el anterior de mezcla instantánea, los coeficientes de la matriz de mezcla ya no pueden ser constantes, sino que tienen que depender del tiempo, es decir, ahora el elemento de la fila *i*-ésima y la columna *j*-ésima será $a_{ij}(t)$, además ya no podemos hallar la salida del sistema de mezcla como la multiplicación de la matriz de mezcla por el vector de entrada, si no que habrá que convolucionar dichas matrices en cada instante de tiempo:

$$\mathbf{x}(t) = \mathbf{A}(t) * \mathbf{s}(t) \Leftrightarrow \begin{bmatrix} x_1(t) \\ \vdots \\ x_M(t) \end{bmatrix} = \begin{bmatrix} a_{11}(t) & \cdots & a_{1N}(t) \\ \vdots & \ddots & \vdots \\ a_{M1}(t) & \cdots & a_{MN}(t) \end{bmatrix} * \begin{bmatrix} s_1(t) \\ \vdots \\ s_N(t) \end{bmatrix} \tag{5.2}$$

El operador $*$ denota la convolución y no el producto, siendo la convolución entre dos funciones $f(x)$ y $g(x)$ la función calculada como $h(x) = f(x) * g(x) = \int_{-\infty}^{\infty} f(z)g(x-z)dz$. En el caso de secuencias discretas, ésta se calcula como $h(n) = f(n) * g(n) = \sum_{m=-\infty}^{\infty} f(n-m)g(m)$. Las propiedades de la operación de convolución se muestran en multitud de libros, por ejemplo, [Oppenheim89].

De esa forma, la señal grabada por el micrófono *i*, se obtendrá como la suma de las convoluciones de los $s_j(t)$ con la respuestas impulsivas $a_{ij}(t)$ correspondientes al camino entre la fuente *j* y el micrófono *i*, que dependerá de la geometría del entorno en que se realice la grabación.

$$x_i(t) = \sum_j \int a_{ij}(\tau)s_j(t-\tau)d\tau \tag{5.3}$$

En la figura (5.3) se muestra cómo sería el modelo que acabamos de plantear para el caso en que halla solamente dos fuentes y se usen dos micrófonos para hacer las grabaciones.

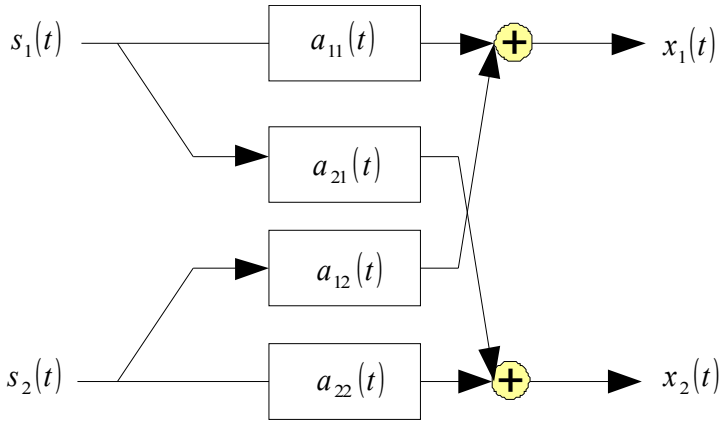


Figura 5.3 *Sistema de mezcla convolutiva en el caso de dos fuentes y dos micrófonos.*

5.4 Modelo de mezcla anecoica.

En el apartado 5.3 hemos visto el caso general de mezcla de voz. Ese caso sería válido para cualquier situación, sin embargo, no siempre el entorno es tan "hostil" matemáticamente hablando, y por lo tanto habrá ocasiones en que podamos hacer algunas simplificaciones razonables para evitar tener que trabajar con ese problema matemático, simplificando los cálculos.

Una de esas situaciones es precisamente el caso que vamos a estudiar ahora. Se corresponde a una situación anecoica, es decir, sin ecos. Esta situación se produciría por ejemplo en el interior de una cámara anecoica. También puede aproximarse en muchas ocasiones como entorno anecoico el espacio libre, es decir, si realizamos grabaciones sin presencia de paredes u otros obstáculos que produzcan ecos y reflexiones del sonido.

5.4.1 Formulación en el dominio del tiempo.

En el espacio libre, que es el caso que tratamos en este apartado, el sonido propagándose desde un punto *j* hasta otro punto *i* es atenuado por un factor a_{ij} y se retrasa un tiempo τ_{ij}. Esto se corresponde a modelar las voces como rayos que se propagan desde la fuente que las emite (las personas) hasta los micrófonos en los que son grabadas.

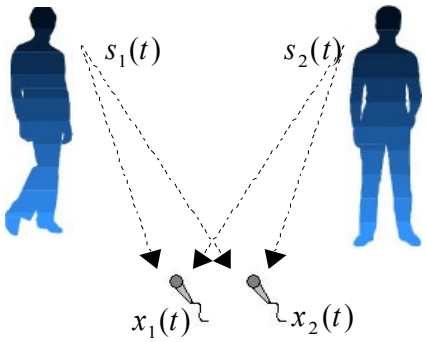


Figura 5.4 *Modelo de rayos en un entorno anecoico.*

Partiendo del modelo propuesto en la ecuación (5.2), la respuesta al impulso de cada camino que recorren las señales para el modelo de atenuación y retardo se simplificaría como:

$$a_{ij}(t) = a_{ij}\delta(t - \tau_{ij}) \tag{5.4}$$

donde $\delta(t)$ es la conocida función delta de Dirac. Aquí los coeficientes τ_{ij} están referidos a los micrófonos, es decir, tomamos los retrasos que sufren las fuentes al propagarse desde que se graba en un micrófono hasta que el frente de ondas de dicha fuente llega a otro, ya que podemos tomar el punto de referencia donde queramos. Ya sabemos como es la convolución de una función cualquiera con una delta de Dirac:

$$s_j(t) * \delta(t - \tau_{ij}) = s_j(t - \tau_{ij}) \tag{5.5}$$

Por lo tanto, las señales mezcladas en el espacio libre serían:

$$x_i(t) = \sum_j a_{ij} s_j(t - \tau_{ij}) \tag{5.6}$$

Puesto que los valores absolutos de amplitud y retardos de propagación son difíciles de identificar, es suficiente con conocer las diferencias entre tiempo y atenuación entre las diferentes señales fuente recibidas en los diferentes micrófonos. De esta forma, podemos normalizar los elementos de la diagonal de $a_{ij}(t)$ a la unidad. El sistema de mezcla correspondiente a la situación de una

mezcla de dos fuentes que son grabadas por dos micrófonos en el espacio libre sería:

$$\begin{aligned} x_1(t) &= s_1(t) + a_{12}s_2(t - \tau_{12}) \\ x_2(t) &= s_2(t) + a_{21}s_1(t - \tau_{21}) \end{aligned} \tag{5.7}$$

Esto se corresponde con el sistema de mezcla que vemos en la figura (5.5).

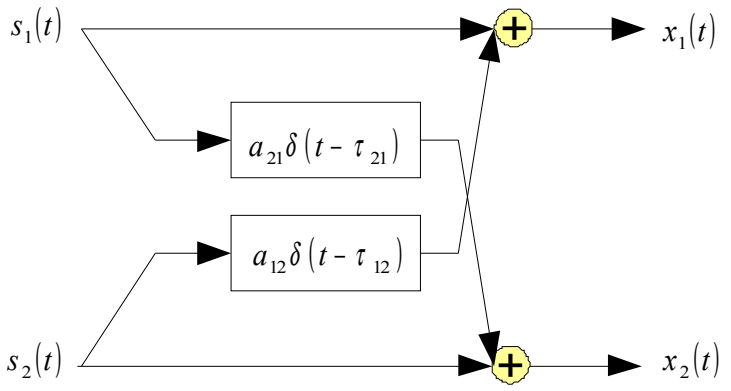


Figura 5.5 *Sistema de mezcla para la aproximación anecoica.*

5.4.2 Formulación en el dominio de la frecuencia.

Ya hemos hablado en la sección anterior de la transformada STFT. Bien pues este tipo de transformada puede aplicarse a las señales que estamos tratando, y veremos inmediatamente que simplificará bastante la situación. La transformada STFT $X_i(n,\omega)$ correspondiente a la señal $x_i(m)$ que obtenemos del muestreo de $x_i(t)$ es:

$$X_i(n,\omega) = \sum_{m=-\infty}^{\infty} x_i(m)w(n - m)e^{-j\omega m} \tag{5.8}$$

Trabajamos aquí con la versión muestreada $x_i(m)$ porque así será como se trabaje en los computadores digitales en los que se almacenarán estas señales. Así mismo trabajamos con la versión discreta de la transformada STFT, de modo que:

$$X_i(n,\omega) = X_i(ln_0, k\omega_0) \tag{5.9}$$

Los parámetros n_0 y ω_0 son las dimensiones temporal y frecuencial de las celdas de la rejilla del espectrograma, de manera que *k* representa el índice de frecuencia y *l* el índice del tiempo. Entonces considerando la expresión temporal de $x_j(t)$ en función de las fuentes $s_i(t)$ que acabábamos de ver, podemos pasar al dominio de la frecuencia como:

$$X_i(n,\omega) = \sum_j a_{ij}(\omega) S_j(n,\omega) \tag{5.10}$$

Al ser conocida para nosotros la forma que tendrán los coeficientes $a_{ij}(t) = a_{ij}\delta(t - \tau_{ij})$, podemos saber fácilmente su transformada en frecuencia, que es:

$$a_{ij}(\omega) = a_{ij} e^{-j\omega\tau_{ij}} \tag{5.10}$$

y de esta forma la formulación de la situación en el dominio de la frecuencia (concretamente en el dominio tiempo-frecuencia) sí podría expresarse de forma matricial como:

$$\begin{bmatrix} X_1(n,\omega) \\ \vdots \\ X_M(n,\omega) \end{bmatrix} = \begin{bmatrix} a_{11}(\omega) & \cdots & a_{1N}(\omega) \\ \vdots & \ddots & \vdots \\ a_{M1}(\omega) & \cdots & a_{MN}(\omega) \end{bmatrix} \begin{bmatrix} S_1(n,\omega) \\ \vdots \\ S_N(n,\omega) \end{bmatrix} \tag{5.11}$$

Hemos llegado entonces a un sistema lineal que describe un problema que en el dominio del tiempo era no lineal. Entonces podemos vislumbrar que en la mayoría de ocasiones será más conveniente trabajar con esta representación de los datos, ya que no es difícil pasar las señales al dominio de la frecuencia, y este paso nos simplificará bastante el problema.

5.5 Caso sobredeterminado.

Hasta ahora no habíamos concretado en número de fuentes y de micrófonos presentes en la mezcla, sino que los habíamos tratado de forma genérica como constantes *N* y *M*.

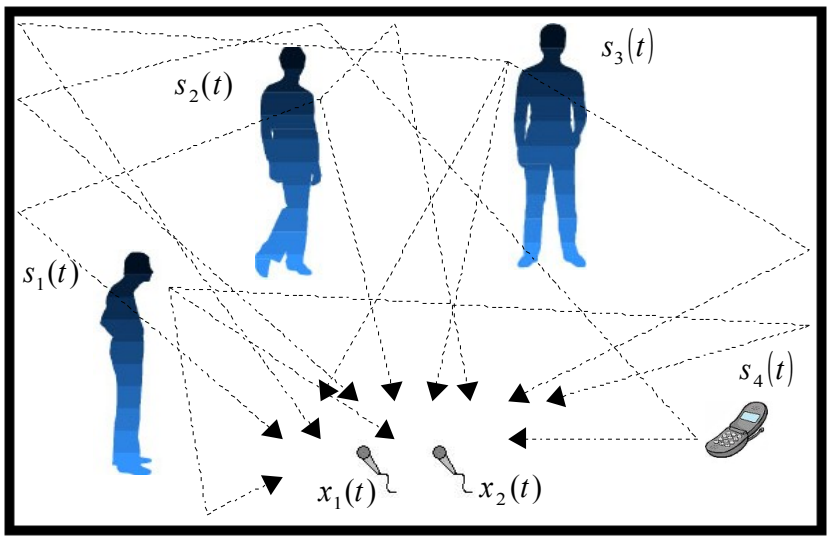


Figura 5.6 *Modelo de rayos en el caso sobredeterminado.*

En la figura (5.6) se muestra un modelo similar al ya planteado de mezcla convolutiva pero en el caso en que haya más fuentes que grabaciones. Podemos intuir sin necesidad de hacer cálculos que el modelo matemático se complica mucho, ya que aparecen muchos más parámetros desconocidos que conocidos. Matemáticamente este caso se trataría aplicando el mismo modelo que hemos visto en el apartado (5.3), siendo un caso particular de éste.

Anticipamos ahora que estamos viendo todos estos modelos debidos a que luego intentaremos aplicarles los conceptos y técnicas en el capítulo dedicado al análisis de componentes independientes para intentar extraer las señales fuente a partir de las grabaciones de los micrófonos. Y si recordamos los principios de ICA, siempre se suponía en el modelo que había tantas señales observadas como señales fuente, puesto que esto era algo necesario para llevar a cabo la separación. Si esto no se cumplía el sistema no era invertible y por lo tanto no podía conseguirse nada. Bien, en la situación que aquí hemos planteado no se cumple esa condición, y por lo tanto no podría usarse esa familia de técnicas para tratar los datos generados en una situación que se ajustase a este modelo. Por este motivo no vamos a prestar más atención a este caso y simplemente nos hemos limitado a nombrarla, dejando claro que es una situación a la que no se le ha dado aún una solución satisfactoria, aunque en el siguiente capítulo veremos que en un caso particular de la misma sí que se podrá tratar, aunque es cierto que simplificaremos bastante el modelo suponiendo que las grabaciones se realizan en un entorno anecoico.

5.6 Conclusiones.

En el capítulo anterior dedicado a la STFT ya hablamos de las características más importantes de la señal de voz, pues bien, en este capítulo nos hemos dedicado a presentar los modelos matemáticos de mezcla de señales de voz que nos harán falta conocer en la siguiente sección. Cada uno de estos métodos los hemos descrito en el dominio del tiempo y también en el de la frecuencia, ya que a veces simplifica bastante las cosas.

Comenzamos con el modelo más simple, el instantáneo, que sólo se presenta de forma introductoria a los demás, ya que no se ajusta a ninguna situación práctica. El que sí que se ajusta a una gran variedad de situaciones es el modelo de mezcla convolutiva, que es el más completo. Éste se basa en modelar la voz como una serie de rayos que se propagan por el espacio y llegan hasta los sensores con un retraso y una atenuación debidas a la velocidad de propagación finita y a la reverberación. Esto lo expresamos mediante la notación usada en los capítulos precedentes donde representábamos el sistema de mezcla por una matriz, pues ahora la matriz depende del instante de tiempo y además no se multiplica por las señales, sino que hay que realizar la integral de convolución. Veíamos que si transformamos las señales al dominio tiempo-frecuencia las convoluciones en el dominio del tiempo se transforman en multiplicaciones en el dominio transformado. Este tratamiento hace la situación bastante más sencilla.

Tras ver el caso más general, hemos hecho una simplificación del mismo tomando en consideración un solo retardo en el modelo, lo que se corresponde

con una situación al aire libre. Esto hacía que las convoluciones se realizaran con deltas de Dirac y por lo tanto había muchos menos parámetros en el sistema, estando compuesta la matriz de mezcla por coeficientes constantes en cada banda de frecuencia, reduciéndose cada uno de ellos a una constante multiplicada por una exponencial compleja dependiente de la frecuencia.

Por último hemos nombrado el caso sobredeterminado en el que hay más señales fuente que sensores tomando medidas, sin entrar en muchos detalles ya que este caso es muy complicado y todavía no tiene una solución convincente en el propósito que nos ocupa.

Capítulo 6

Separación ciega de señales de voz

6.1 Introducción.

Hemos esbozado ya la descripción de todas las herramientas que nos harán falta para el verdadero propósito de este texto, que no es otro que plantear una serie de métodos que nos permitan llevar a cabo la separación de señales de voz partiendo de grabaciones en las que se tenga una mezcla de ellas. El punto que tienen en común todas ellas es que la separación la realizaremos en el dominio de la frecuencia, y para ello usaremos la transformada de Fourier de corta duración. Plantearemos tres métodos distintos que serán útiles en diferentes situaciones y que explicaremos con detalle en los siguientes apartados:

- Separación mediante enmascaramiento. Este método de separación parte del supuesto de que las señales fuente son disjuntas en el dominio tiempo-frecuencia, y por lo tanto se basa en analizar el espectrograma de las mezclas grabadas para determinar a qué fuente corresponde cada punto de los mismos. Esto se hará mediante la estimación de los ángulos de llegada de los frentes de onda correspondientes a cada fuente, formando máscaras que posteriormente serán aplicadas a los espectrogramas de las grabaciones con el fin de separar cada fuente.
- Separación mediante procesamiento adaptativo en el dominio de la frecuencia. En este método aplicaremos un algoritmo adaptativo que irá iterando tomando como entrada los puntos de la transformada STFT para lograr la estimación de los parámetros más importantes de la matriz de separación. Propondremos un algoritmo basado en cumulantes. Una vez estimados dichos parámetros, se forma la matriz de separación, que será distinta en cada banda de frecuencia, y se realiza la separación conforme al modelo ICA.
- Separación mediante análisis independiente de las subbandas de frecuencia. Este último método es quizás el que puede dar buenos resultados en una gama de situaciones más variada. Se basa en separar mediante ICA todas las subbandas de frecuencia (filas del espectrograma) de las mezclas observadas de forma independiente. Se trata por consiguiente de descomponer el problema total en muchos problemas independientes de menor envergadura. Como veremos después, también será importante solucionar de forma conjunta las ambigüedades inherentes a ICA, que son la correcta ordenación de los componentes independientes y el ajuste del escalado tras la separación.

6.2 Separación ciega de señales de voz mediante enmascaramiento.

En este apartado vamos a presentar una solución sencilla para resolver la separación ciega de fuentes (BSS) de voz que puede ser útil en el caso de sistemas sobredeterminados, en el que el número de observaciones es menor que el número de fuentes, particularizando para el caso en que la grabación se realice con 2 micrófonos (situación que se corresponde con sensores estereofónicos). La solución propuesta se basa en la técnica de enmascaramiento (masking) en el dominio tiempo-frecuencia bajo la suposición de que las señales fuente son disjuntas en dicho dominio.

6.2.1 Planteamiento del problema.

Sean N fuentes estadísticamente independientes distribuidas espacialmente y dos micrófonos idénticos situados a una distancia conocida d, tal como se aprecia en la figura (6.1). Bajo la hipótesis de campo lejano, podemos afirmar que la localización de la fuente i-ésima es función del ángulo de llegada θ_i, definido como el ángulo que forma el frente de ondas de la fuente i-ésima respecto al sensor de referencia. Sin pérdida de generalidad, podemos tomar como sensor de referencia el micrófono 1, de manera que la fuente i-ésima llegará retardada al micrófono 2, y dicho retardo dependerá del ángulo de llegada θ_i de dicha fuente.

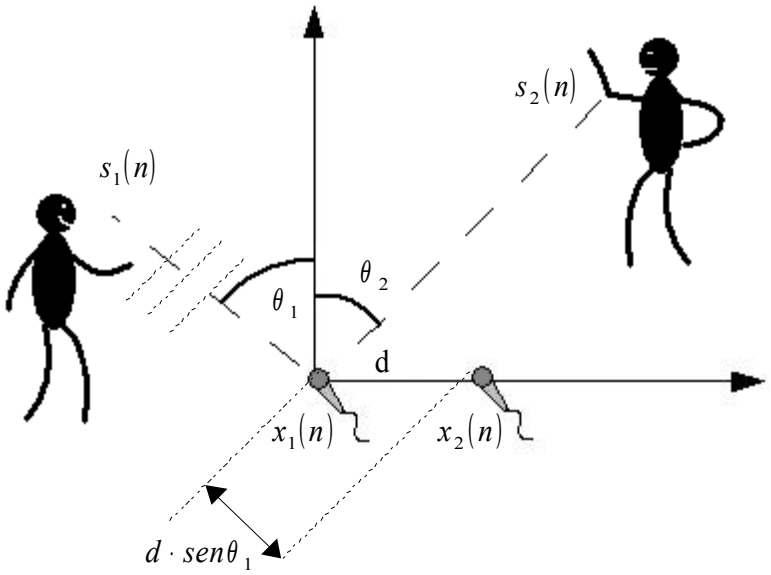


Figura 6.1 *Modelos del sistema de estudio. Dos micrófonos separados una distancia d, graban señales bajo la hipótesis de campo lejano.*

Vamos a plantear el sistema para el caso general sobredeterminado en que haya N señales fuente. Luego podrá particularizarse para el número de fuentes presentes en el sistema (normalmente dos). De esta forma, y para el modelo

planteado, las señales obtenidas en los micrófonos pueden expresarse en función de las fuentes $s_1(n),\ldots,s_N(n)$ a través del siguiente sistema de mezcla:

$$x_1(n) = \sum_{j=1}^{N} s_j(n) \tag{6.1}$$

$$x_2(n) = \sum_{j=1}^{N} a_j s_j(n - \tau_j) \tag{6.2}$$

donde a_j es la amplitud de la fuente *j* en el micrófono 2 en relación al micrófono de referencia 1, y τ_j es el retardo de propagación de la fuente *j* entre los dos micrófonos. El retardo τ_j puede expresarse fácilmente en función del ángulo de llegada θ_j de la fuente *j*:

$$\tau_j = \frac{d \cdot sen(\theta_j)}{c} \tag{6.3}$$

donde *c* es la velocidad de propagación del frente de ondas, que para el caso del sonido en el aire es de 344 m/s.

La solución que proponemos trabaja en el dominio trasformado tiempo-frecuencia. Para ello usaremos la transformada localizada de Fourier (STFT) de las señales discretas que ya definimos en la sección 3. De esta forma, la STFT de $s_j(n)$, como ya estaba definida, es:

$$S_j(n,\omega) = \sum_{m=-\infty}^{\infty} s(m)w(n-m)e^{-j\omega m} \tag{6.4}$$

Ya que queremos que este algoritmo pueda implementarse en un computador digital, trabajaremos con la versión discreta de la STFT:

$$S_j(n,\omega) = S_j(ln_0, k\omega_0) \tag{6.5}$$

Los parámetros n_0 y ω_0 son las dimensiones temporal y frecuencial de las celdas de la rejilla del espectrograma. Esta equivalencia no es trivial, y será sólo verdadera para ventanas con valores suficientemente pequeños de n_0 y ω_0.

Vamos ahora entonces a enunciar la suposición fundamental de este método, y sobre la que se sustenta la separación mediante enmascaramiento. Dicha suposición es que las señales fuente son disjuntas en el dominio tiempo-frecuencia, esto es:

$$S_i(n,\omega) \cdot S_j(n,\omega) = 0 \quad \forall\, n,\omega, i \neq j \tag{6.6}$$

Esto quiere decir que el valor de cada punto del espectrograma de las grabaciones de los micrófonos, sólo será debido a una de las fuentes que forman parte de la mezcla, es decir, sólo una fuente está "activa" en cada punto del

espectrograma, siendo nula la contribución del resto de fuentes a ese punto de la matriz STFT. Si esa suposición se cumple podemos expresar la mezcla de la siguiente manera:

$$X_1(n,\omega) = \sum_{j=1}^{N} S_j(n,\omega) \tag{6.7}$$

$$X_2(n,\omega) = \sum_{j=1}^{N} a_j S_j(k,l) e^{-j\omega\tau_j} \tag{6.8}$$

6.2.2 Algoritmo de separación.

La técnica de enmascaramiento estima el espectrograma de las fuentes aplicando máscaras al espectrograma de las observaciones. El algoritmo de separación estima la fuente que está activa en cada punto del espectrograma de las observaciones clasificando los puntos en función del ángulo de llegada o del retardo. Posteriormente, se construyen las máscaras, un por cada fuente, que aplicadas sobre el espectrograma de la observación de referencia, estiman el espectrograma de las fuentes. Finalmente se realiza la transformación del dominio tiempo-frecuencia al dominio temporal.

Bajo la suposición de que las señales fuente son disjuntas en el dominio tiempo-frecuencia, se puede decir que cada punto (n,ω) del espectrograma corresponde al menos a una única fuente. Cada punto de las observaciones en este dominio tomará entonces la siguiente forma:

$$\begin{bmatrix} X_1(n,\omega) \\ X_2(n,\omega) \end{bmatrix} = \begin{bmatrix} 1 \\ a_j e^{-j\omega\tau_j} \end{bmatrix} S_j(n,\omega) \tag{6.9}$$

Por lo tanto, analizando cada punto del espectrograma, podemos obtener el retardo asociado a la fuente que está activa en dicho punto a través de la fase de la división de los espectrogramas de las observaciones:

$$\tau(n,\omega) = \frac{1}{\omega} \angle \frac{X_1(n,\omega)}{X_2(n,\omega)} \tag{6.10}$$

Dado que el retardo depende del ángulo de llegada de la fuente a través de la ecuación (6.3), se puede conseguir la separación de las fuentes clasificando los puntos del espectrograma en función del retardo asociado a cada punto.

La clasificación puede realizarse mediante diversas técnicas de clustering. Se considera que los retardos asociados a cada fuente son variables aleatorias con función densidad de probabilidad gaussiana, de manera que la totalidad de los retardos del espectrograma corresponde a una mezcla de gaussianas. Para determinar los parámetros del modelo de mezcla de gaussianas se maximiza la verosimilitud de los parámetros para los datos de los retardos. La solución adoptada se basa en el algoritmo de maximización del valor esperado, EM,

utilizando el criterio MDL para la estimación del número de fuentes. Expondremos esto con algo más de detalle en el subapartado siguiente.

Una vez clasificados todos los puntos del espectrograma, se construyen las máscaras de extracción de cada fuente *j*, $M_j(n,\omega)$. Estas máscaras se aplican sobre el espectrograma del micrófono de referencia $X_1(n,\omega)$, y se obtienen los N espectrogramas estimados de las señales fuente como:

$$Y_j(n,\omega) = M_j(n,\omega) \bullet X_1(n,\omega) \quad \forall\, n,\omega\,,\, j = 1,\ldots,N \tag{6.11}$$

El operador $\bullet$ denota multiplicación punto a punto de las matrices. Las máscaras podrían ser binarias o continuas, aunque aquí sólo vamos a considerar el primer caso. La máscara binaria para la extracción de la fuente *j* será nula en aquellos puntos del espectrograma en los que el ángulo de llegada estimado no pertenezca a la fuente *j* y 1 en caso contrario. Para evitar puntos aislados en el espectrograma de las salidas, que pueden ser causantes de ruido musical, se puede aplicar un filtrado de mediana a las máscaras.

Finalmente, para obtener las señales fuente estimadas en el dominio del tiempo, basta con reconstruirlas a partir de los espectrogramas obtenidos tras aplicar las máscaras, esto lo haremos mediante la inversa de la transformada STFT usando la técnica de overlap-add.

6.2.3 El algoritmo EM (Expectation Maximization).

El algoritmo EM [Bishop96] presenta una técnica iterativa para realizar una estimación de máxima verosimilitud de los parámetros de un modelo probabilístico determinado, dado un conjunto de datos. En el caso que nos ocupa los parámetros que queremos estimar son los retardos del modelo y partiremos tomando como datos de entrada los espectrogramas de las observaciones.

La función densidad de probabilidad de los retardos $f(\tau)$ se aproxima por una suma ponderada de *K* componentes, donde cada una es una función densidad parametrizada cuyos parámetros hay que averiguar.

$$f(\tau) = \sum_{j=1}^{K} f(\tau \mid j) P(j) \tag{6.12}$$

$$\sum_{j=1}^{K} P(j) = 1 \text{ con } 0 \le P(j) \le 1 \tag{6.13}$$

$P(j)$ son las probabilidades a priori de los datos, que también forman parte de la solución buscada. Para el problema que nos ocupa resultan adecuadas las gaussianas, de manera que los parámetros a buscar son la media μ_j y la varianza σ_j^2 de cada una de ellas. La probabilidad condicionada viene dada entonces por:

$$f(\tau \mid j) = \frac{1}{\sqrt{2\pi}\,\sigma_j} \exp\left\{ - \frac{\left\| \tau - \mu_j \right\|^2}{2\sigma_j^2} \right\} \tag{6.14}$$

El ajuste de los parámetros del modelo requiere alguna medida de su bondad, es decir, cómo de bien encajan los datos sobre la distribución que los representa. Este valor de bondad se conoce como la verosimilitud *L* de los datos. Se trataría entonces de estimar los parámetros buscados maximizando la verosimilitud, criterio que se conoce como *Maximum Likelihood* (ML). Por simplicidad analítica se calcula el logaritmo de la verosimilitud, conocido como verosimilitud logarítmica, que denotamos por *l*, obteniéndose la misma solución gracias a la propiedad de monotonicidad del logaritmo. La verosimilitud logarítmica negativa se puede interpretar como una función error *E*, de forma que maximizar *L* es equivalente a minimizar *E*.

$$E = -\ln L = -\sum_{n=1}^{N} \ln f(\tau^n) = -\sum_{n=1}^{N} \ln\left\{ \sum_{j=1}^{K} f(\tau^n \mid j) P(j) \right\} \tag{6.15}$$

donde *N* es el número de instancia del algoritmo EM. Minimizar la función *E* respecto a los parámetros, como veremos, no es algo trivial. En el caso de mezclas gaussianas, los parámetros toman los siguientes valores cuando alcanzamos el mínimo de *E*:

$$\hat{\mu}_j = \frac{\sum_n P(j \mid \tau^n) \tau^n}{\sum_n P(j \mid \tau^n)} \tag{6.16}$$

$$\hat{\sigma}_j^2 = \frac{\sum_n P(j \mid \tau^n) \left\| \tau^n - \hat{\mu}_j \right\|^2}{\sum_n P(j \mid \tau^n)} \tag{6.17}$$

$$\hat{P}(j) = \frac{1}{N} \sum_{n=1}^{N} P(j \mid \tau^n) \tag{6.18}$$

El algoritmo EM, se nutre con unos parámetros iniciales que convergen después de una serie de iteraciones a un mínimo local de la función *E*. Para una iteración del algoritmo denotamos por $f^{new}(\tau)$ a la densidad de probabilidad evaluada con los parámetros nuevos y $f^{old}(\tau)$ a la densidad de probabilidad evaluada con los parámetros de la anterior iteración. La disminución en el error *E* será pues:

$$E^{new} - E^{old} = -\sum_n \ln\left\{ \frac{f^{new}(\tau^n)}{f^{old}(\tau^n)} \right\} \tag{6.19}$$

que puede ser escrita de la siguiente forma utilizando la desigualdad de Jensen:

$$E^{new} - E^{old} \leq -\sum_n \sum_j P^{old}\left(j \mid \tau^n\right) \ln\left\{ \frac{P^{new}(j) f^{new}\left(\tau^n \mid j\right)}{P^{old}\left(j \mid \tau^n\right) f^{old}\left(\tau^n\right)} \right\} = Q \qquad (6.20)$$

Nuestro objetivo es minimizar E^{new} respecto a los nuevos parámetros. Dado que $E^{new} \leq E^{old} + Q$, resulta que $E^{old} + Q$ constituye una cota inferior de E^{new}. El algoritmo EM minimiza esta cota inferior respecto a los nuevos parámetros en cada iteración, tal como puede verse en la figura (6.2).

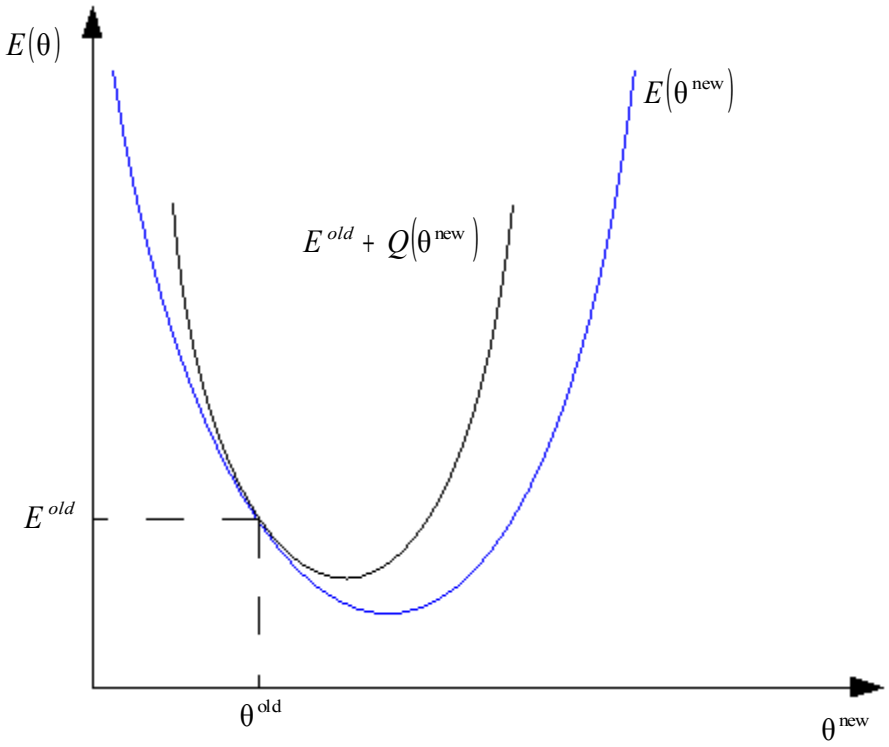


Figura 6.2 *Gráfica de la función error E en función del nuevo valor* θ^{new} *de uno de los parámetros del modelo de mezcla.*

Para una mezcla de gaussianas, los parámetros nuevos en cada iteración vienen dados por:

$$\mu_j^{new} = \frac{\sum_n P^{old}\left(j \mid \tau^n\right) \tau^n}{\sum_n P^{old}\left(j \mid \tau^n\right)} \qquad (6.21)$$

$$(\sigma_j^{new})^2 = \frac{\sum_n P^{old}\left(j \mid \tau^n\right) \left\| \tau^n - m\mu_j^{new} \right\|^2}{\sum_n P^{old}\left(j \mid \tau^n\right)} \qquad (6.22)$$

$$P(j)^{new} = \frac{1}{N} \sum_n P^{old}\left(j \mid \tau^n\right) \qquad (6.23)$$

Para utilizar el algoritmo EM es necesario conocer el número de componentes de la mezcla. Si desconocemos a priori el número de señales de voz presente en las observaciones, no podemos utilizar el algoritmo EM tal y como está propuesto. Un posible criterio para estimar el número de componentes de la mezcla podría ser incrementar el número de gaussianas hasta que el error de modelado sea minimizado. Es razonable pensar que el error va a decrecer de forma monótona a medida que se incrementa el orden del modelo. Podemos monitorizar la velocidad de decrecimiento del error y decidir terminar el proceso cuando ésta se hace relativamente lenta. Sin embargo, esta aproximación puede ser imprecisa y mal condicionada, por lo que resultan necesarios otros métodos. De todas formas nosotros siempre supondremos conocido el dato del número de fuentes presentes en la mezcla por lo que no seguiremos desarrollando estos métodos. Para inicializar las medias de las gaussianas, se construye un modelo autorregresivo del histograma, y se utilizan los polos del modelo como estimación inicial. En el caso de las varianzas, lo más aconsejable es utilizar varianzas fijas con valores suficientemente elevados para evitar mínimos locales. Las probabilidades a priori, $P(j)$ se calculan usando los valores del histograma en las medias de las gaussianas calculadas anteriormente, y normalizando para que la suma de las probabilidades sea igual a 1.

6.3 Separación ciega de señales de voz mediante procesamiento adaptativo en el dominio de la frecuencia (método de Anemüller).

6.3.1 Planteamiento de la situación.

En este segundo método de separación que vamos a presentar nos basaremos en el modelo de mezcla que presentamos en la sección 4 para un entorno anecoico. La idea de este método está expuesta de forma muy detallada en la publicación [Anemüller01].

Recordamos que según el modelo de mezcla planteado, la señal grabada por el micrófono *j* en función de las señales fuente venía dada por la siguiente expresión:

$$x_i(t) = \sum_j a_{ij} s_j(t - \tau_{ij}) \tag{6.24}$$

Esa era la expresión en el dominio del tiempo, donde a_{ij} eran los coeficientes que representaban la atenuación y τ_{ij} eran los retardos de propagación, ambos referidos al camino de propagación desde una fuente *j* hasta el micrófono *i*.

También llegamos a la conclusión de que no era difícil transformar esa expresión al dominio de la frecuencia haciendo uso de la transformada STFT, resultando un modelo lineal que representábamos en notación matricial:

$$\begin{bmatrix} X_1(n,\omega) \\ \vdots \\ X_M(n,\omega) \end{bmatrix} = \begin{bmatrix} a_{11}(\omega) & \cdots & a_{1N}(\omega) \\ \vdots & \ddots & \vdots \\ a_{M1}(\omega) & \cdots & a_{MN}(\omega) \end{bmatrix} \begin{bmatrix} S_1(n,\omega) \\ \vdots \\ S_N(n,\omega) \end{bmatrix} \tag{6.25}$$

donde los coeficientes en el dominio de la frecuencia son $a_{ij}(\omega) = a_{ij} e^{-j\omega\tau_{ij}}$. Pues bien, partiendo de este sistema bien definido el objetivo ahora es, con del conocimiento sólo de las grabaciones de los sensores $x_i(t)$, $i = 1,\ldots,M$, ser capaces de obtener las señales fuente $s_j(t)$, $j = 1,\ldots,N$.

A partir de ahora vamos a suponer en este modelo que tenemos el mismo número de fuentes que de grabaciones, digamos *N*. La solución que vamos a plantear necesita al menos una grabación por cada fuente, por lo que no será válida para sistemas sobredeterminados. Así mismo, en el caso en que hubiera más grabaciones que fuentes, alguna de ella sería redundante y reduciríamos el número de las mismas hasta quedarnos sólo con una por cada fuente. Por lo tanto no tiene sentido considerar esos casos que no podemos resolver. De esta forma la matriz de mezcla ahora es cuadrada y de dimensión $N \times N$.

Para solucionar este modelo vamos a usar técnicas de análisis de componentes independientes. Utilizaremos por lo tanto el modelo ampliamente comentado en el capítulo 2 donde los datos y los componentes independientes vienen dados mediante sendos vectores y los segundos se obtienen a partir de los primeros haciendo uso de una matriz de mezcla. Por supuesto, y como ya que hemos planteado este modelo, para representar esta mezcla por medio de una matriz tenemos que transformar las señales al dominio de la frecuencia, concretamente al dominio tiempo-frecuencia, mediante la ya definida y usada anteriormente STFT.

Nuestro sistema queda como se muestra a continuación:

$$\begin{bmatrix} X_1(n,\omega) \\ \vdots \\ X_N(n,\omega) \end{bmatrix} = \begin{bmatrix} 1 & \cdots & a_{1N}(\omega) \\ \vdots & \ddots & \vdots \\ a_{N1}(\omega) & \cdots & 1 \end{bmatrix} \begin{bmatrix} S_1(n,\omega) \\ \vdots \\ S_N(n,\omega) \end{bmatrix} \tag{6.26}$$

Hemos normalizado las señales de forma que la diagonal principal de la matriz de mezcla esté formada por unos, ya que como se vio en el capítulo dedicado a ICA, la multiplicación por cualquier constante de los todos los componentes independientes no afecta al modelo, y de hecho era una de las ambigüedades que nos encontrábamos en la solución. Hemos hecho esto porque así tenemos menos parámetros que estimar, y por lo tanto será más sencillo llegar a la solución.

Si somos capaces de estimar los parámetros de la matriz de mezcla, seremos capaces de reconstruir las señales fuente a partir de la inversa de la misma, transformando por último al dominio del tiempo. Si consideramos el caso en que tenemos dos señales fuente y dos grabaciones, llamando $\mathbf{W}(\omega)$ a la matriz de separación en el dominio de la frecuencia, tenemos que, para cada valor de la

frecuencia, dicha matriz puede expresarse en función de los parámetros de la matriz de mezcla como:

$$\mathbf{W}(\omega) = \begin{bmatrix} w_{11}(\omega) & w_{12}(\omega) \\ w_{21}(\omega) & w_{22}(\omega) \end{bmatrix} = \frac{1}{1 - a_{12}(\omega)a_{21}(\omega)} \begin{bmatrix} 1 & -a_{12}(\omega) \\ -a_{21}(\omega) & 1 \end{bmatrix} \qquad (6.27)$$

De esta forma podríamos hallar la transformada STFT de los componentes independientes, y sería equivalente al siguiente sistema:

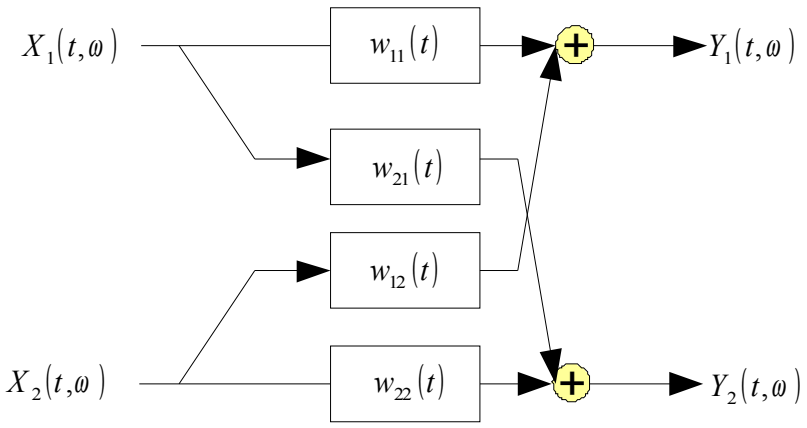


Figura 6.3 *Sistema de separación para obtener las señales originales a partir de las mezclas.*

6.3.2 Algoritmo de optimización.

Ya sabemos cómo actuar para conseguir la separación de las fuentes a partir de las mezclas. Entonces ya tenemos claro que tenemos que estimar los parámetros fundamentales de la matriz de separación, éstos serán las magnitudes y los retardos, es decir, w_{ij} y τ_{ij} respectivamente. Conocidos esos coeficientes, ya podríamos construir la matriz de separación para cada banda de frecuencia, y con ella obtener la correspondiente banda del espectrograma de las señales fuente estimadas. Vamos a presentar ahora el algoritmo que tenemos que implementar para encontrar esos coeficientes.

Lo primero que diremos es que la solución que hemos adoptado consiste básicamente en la determinación de una función para la que la solución que nos lleva a la separación de las fuentes, sin considerar las densidades de las mismas, sea siempre un tipo especial de punto crítico cuyas propiedades puedan ser explotadas posteriormente para su localización. El método propuesto es radicalmente diferente a los que suelen usarse para BSS puesto que el punto crítico deseado no es ni un máximo ni un mínimo de la función, sino un punto silla. Para obtener más información tanto del criterio como de los algoritmos que vamos a exponer se puede acudir a la publicación [Cruces02].

Puesto que la suposición clave para conseguir la separación de las fuentes es su independencia mutua, el principio más natural para BSS es el principio de información mutua (MMI), que afirma que la separación puede encontrarse

siempre como el mínimo global de la información mutua de las salidas (del sistema de separación). Pero la minimización de la información mutua no es trivial porque necesitamos estimar la función densidad de probabilidad de las salidas a partir de un conjunto finito de datos. Para evitar los problemas derivados de trabajar con la información mutua, en este texto explotaremos otra posibilidad menos usual. Usaremos una medida de la no-gaussianidad más simple de evaluar que la que se usa en MMI, pero que conserva su esencia. Lo que hacemos es sustituir la entropía diferencial por el respectivo cumulante de orden $(1+\beta)$ de las salidas, $C_{y_i}^{(1+\beta)}$ (ver apéndice 1 para más información sobre los cumulantes).

Aplicando esta transformación podemos expresar la función de la cual buscamos el punto silla como:

$$\boldsymbol{\Psi}(\mathbf{G}) = \sum_{i=1}^{N} \frac{\left|C_{y_i}^{(1+\beta)}\right|}{1+\beta} - \log\left|\det(\mathbf{G})\right| - h(\mathbf{s}) \tag{6.28}$$

donde hemos denotado por **G** la matriz producto de las matrices de separación y de mezcla respectivamente, $\mathbf{G} = \mathbf{WA}$. Podemos ver claramente que lo que buscamos es que esta matriz sea igual a la unidad, puesto que en el caso de separación ideal tendríamos que $\mathbf{W} = \mathbf{A}^{-1}$. $h(\mathbf{s})$ es una constante que representa a la entropía diferencial de **s,** siendo **s** el vector que contiene los componentes independientes, según la notación que hemos seguido en el capítulo 2.

Puesto que la separación de las fuentes no se encuentra ni buscando un máximo ni un mínimo de $\boldsymbol{\Psi}(\mathbf{G})$, sino un punto silla, no podemos usar métodos de optimización basados en el gradiente para hallar los parámetros del sistema de separación. En lugar de eso, proponemos encontrar la solución BSS usando un método precondicionado que emplea la información de segundo orden disponible en la separación. Para encontrar los ceros del gradiente, proponemos el uso de la iteración precondicionada de forma:

$$vec\mathbf{G}^{(k+1)} = vec\mathbf{G}^{(k)} - \mu^{(k)}\left(\hat{\mathrm{H}}\,\boldsymbol{\Psi}\right)^{-1} vec\left(\frac{\partial \boldsymbol{\Psi}}{\partial \mathbf{G}}\right) \tag{6.29}$$

donde $\hat{\mathrm{H}}\,\boldsymbol{\Psi}$ es una aproximación de la matriz hessiana en las cercanías de la separación, $vec(\cdot)$ es un operador que apila las columnas de una vector cuadrado de dimensión $N \times N$ en un vector de dimensión $N^2 \times 1$ y k es una constante que representa el número de la iteración. $\mu^{(k)}$ es el paso de adaptación. Este algoritmo es una variante de los métodos *chord cuasi-Newton*.

La aproximación para la matriz hessiana que usaremos es:

$$\hat{\mathrm{H}}\,\boldsymbol{\Psi}(\mathbf{G}) = \mathrm{H}_N\left(\left(\mathbf{G}^{-1}\right)^{\mathrm{T}} \otimes \mathbf{G}^{-1}\right) \tag{6.30}$$

que sólo difiere de la matriz hessiana verdadera en los términos de la diagonal. Sustituyendo (6.30) en la iteración (6.29) y volviendo a la notación matricial se llega al siguiente algoritmo:

$$\mathbf{G}^{(k+1)} = \mathbf{G}^{(k)} - \mu^{(k)}\left(\mathbf{C}_{y,y}^{1,\beta}\mathbf{S}_y^{\beta} - \mathbf{I}\right)\mathbf{G}^{(k)} \tag{6.31}$$

Multiplicando (6.31) por $\mathbf{A}^{-1}$ desde la derecha obtenemos la iteración en términos de la matriz de separación:

$$\mathbf{W}^{(k+1)} = \mathbf{W}^{(k)} - \mu^{(k)}\left(\mathbf{C}_{y,y}^{1,\beta}\mathbf{S}_y^{\beta} - \mathbf{I}\right)\mathbf{W}^{(k)} \tag{6.32}$$

que es denotada como algoritmo de inversión iterativa basada en cumulantes (CII).

Al implementar este algoritmo, es útil tener en cuenta que $\mathbf{S}_y^{\beta} = diag\left(diag\left(\mathbf{C}_{y,y}^{1,\beta}\right)\right)$ y que la matriz de cumulantes $\mathbf{C}_{y,y}^{1,\beta}$ puede obtenerse en términos de los momentos de las salidas usando las expresiones mostradas en el apéndice 1. Por ejemplo, si consideramos señales reales y $\beta = 3$ el algoritmo CII puede escribirse de la siguiente forma:

$$\mathbf{W}^{(k+1)} = \mathbf{W}^{(k)} - \mu^{(k)}\left(\left(E\left[\mathbf{y}\left(\mathbf{y}^{\bullet 3}\right)^{\mathrm{T}}\right] - 3E\left[\mathbf{y}\mathbf{y}^{\mathrm{T}}\right]diag\left(E\left[\mathbf{y}^{\bullet 2}\right]\right)\right)\mathbf{S}_y^{3} - \mathbf{I}\right)\mathbf{W}^{(k)} \tag{6.33}$$

donde los elementos de la matriz $\mathbf{S}_y^3$ se definen como $\left[\mathbf{S}_y^3\right]_{ii} = \mathrm{sign}\left(E\left[y_i^4\right] - 3\left(E\left[y_i^2\right]\right)^2\right)$.

Puesto que el algoritmo CII es del tipo cuasi-Newton, deberíamos asegurarnos de que siempre trabaja en la región donde $\boldsymbol{\Psi}(\mathbf{G})$ es continua. Esto significa que no deberíamos alcanzar o cruzar las discontinuidades que suceden cuando la matriz $\mathbf{W}$ se vuelve singular. Puesto que una condición necesaria para que $\mathbf{W}^{(k+1)} = \left(\mathbf{I} - \Delta^{(k)}\right)\mathbf{W}^{(k)}$ sea singular es que $\left\|\Delta^{(n)}\right\| \geq 1$, sólo debemos asegurarnos de que eso nunca suceda. Por lo tanto, y teniendo en cuenta la desigualdad triangular:

$$\left\|\mathbf{C}_{y,y}^{1,\beta}\mathbf{S}_y^{\beta} - \mathbf{I}\right\| \leq 1 + \left\|\mathbf{C}_{y,y}^{1,\beta}\mathbf{S}_y^{\beta}\right\| = 1 + \left\|\mathbf{C}_{y,y}^{1,\beta}\right\| \tag{6.34}$$

basta con escoger:

$$\mu^{(k)} = \min\left(\frac{2\eta}{1+\eta\beta}, \frac{\eta}{1+\eta\left\|\mathbf{C}_{y,y}^{1,\beta}\right\|}\right) \tag{6.35}$$

para evitar que $\mathbf{W}^{(k+1)}$. η es una constante menor que 1 y la expresión $2\eta/(1+\eta\beta)$ procede de las propiedades de convergencia del algoritmo.

Ya tenemos el algoritmo que convergerá a la solución óptima de separación, y por lo tanto podremos hallar los parámetros que buscábamos. Cada iteración *k* se corresponde con una banda de frecuencias del espectrograma de las señales. De este modo iremos iterando según los siguientes pasos:

1) Comenzamos con la primera banda de frecuencias del espectrograma, tomando la primera fila del mismo, que se corresponde con la frecuencia más baja, suponiendo inicialmente que la matriz de separación es igual a la matriz unidad, es decir,

$$\mathbf{W}^{(0)} = \begin{bmatrix} w_{11}(\omega_0) & w_{12}(\omega_0) \\ w_{21}(\omega_0) & w_{22}(\omega_0) \end{bmatrix} = \begin{bmatrix} w_{11} & -w_{12}e^{-j\omega_0\tau_{12}} \\ -w_{21}e^{-j\omega_0\tau_{21}} & w_{22} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

por lo tanto en principio tomamos $w_{11} = w_{22} = 1$, $w_{12} = w_{21} = 0$, vemos que esa situación equivale a suponer que no ha habido mezcla y las señales fuente son exactamente igual a las de mezcla.

2) Calculamos el paso de adaptación μ según la expresión (6.35), así como la matriz de cumulantes $\mathbf{C}_{y,y}^{1,\beta}$, y a partir de ésta $\mathbf{S}_y^{\beta}$.

3) Avanzamos un paso en la iteración según la expresión (6.33).

4) Calculamos las señales de salida como:

$$\begin{bmatrix} Y_1(n,\omega_k) \\ Y_1(n,\omega_k) \end{bmatrix} = \mathbf{W}^{(k)} \begin{bmatrix} X_1(n,\omega_k) \\ X_2(n,\omega_k) \end{bmatrix}$$

5) Extraemos los parámetros w_{ij} y τ_{ij}, tomamos la siguiente frecuencia y actualizamos la matriz de separación $\mathbf{W}^{(k+1)}(\omega_{k+1})$.

6) Volver a (2).

6.4 Separación ciega de mezclas convolutivas de voz en el dominio de la frecuencia.

Hemos visto ya dos métodos diferentes para llevar a cabo la separación ciega de fuentes. Sin embargo, en ambos se adoptaba un modelo de rayos que podía ser poco real en situaciones prácticas, puesto que se adaptaban más a un entorno anecoico, mientras que normalmente la superposición acústica implica reverberación, resultando una mezcla convolutiva.

Para separar señales fuente mezcladas convolutivamente, se debe llevar a cabo un filtrado de las señales grabadas por los micrófonos en vez de una multiplicación, como se hacía en el caso de mezclas no convolutivas. Dependiendo del dominio en el que se implementen los filtros, los algoritmos propuestos en la literatura se clasifican en algoritmos en el dominio del tiempo y algoritmos en el dominio de la frecuencia. Algunos algoritmos pueden considerarse híbridos puesto que implementan la estructura de separación y la

función de optimización en el dominio del tiempo pero acuden al dominio de la frecuencia durante la fase de adaptación de parámetros (por ejemplo Lambert, 1996; Amari, 1997).

Los algoritmos en el dominio del tiempo (Weinstein, 1993; Yellin y Weinstein, 1996; Lee, 1997) tienen que solucionar un problema de optimización no trivial en el que todos los coeficientes de los filtros de separación están acoplados. Lindaren y Broman (1998) informaron de que esto conduce a un mínimo local que hace difícil encontrar el óptimo global.

Los algoritmos en el dominio de la frecuencia (por ejemplo Capdevielle, 1995; Murata, 1998; Parra y Spence, 2000), en contraste, están basados en la propiedad de la transformada de Fourier que dice que la convolución en el dominio del tiempo resulta en una multiplicación en el dominio de la frecuencia. Por lo tanto, el problema de la separación convolutiva de fuentes en el dominio del tiempo es transformado en *K* problemas de separación de fuentes desacoplados, uno por cada frecuencia, $f = f_1,\ldots,f_K$. Tras llevarse a cabo la separación en el dominio de la frecuencia, las fuentes separadas son transformadas de nuevo al dominio del tiempo usando, por ejemplo, la técnica de overlap-add (ver capítulo 4).

El inconveniente de los métodos en el dominio de la frecuencia es que en general aparecen permutaciones locales, esto es, los componentes espectrales de las fuentes son recuperados en un orden diferente (y desconocido) en los diferentes canales de frecuencia, haciendo así imposible una reconstrucción de las señales al dominio del tiempo. Esto se debe a que la separación se realiza de forma independiente en cada frecuencia f_k. Hay varias soluciones propuestas para tratar de solucionar el problema de las permutaciones locales.

Entonces recapitulando, sabemos ya que todos los algoritmos basados en el dominio de la frecuencia tienen que implementar dos etapas de procesado para obtener las señales separadas. En la primera, se busca una solución para el problema BSS en un canal de frecuencia teniendo en cuenta los componentes de señal en la misma frecuencia exclusivamente. Tras hacer esto en todos los canales de frecuencia, el objetivo es reordenar los filtros de separación y los componentes espectrales de las señales separadas de forma que se eviten las permutaciones locales. Hay un tercer problema, relacionado con el escalado de las distintas bandas de frecuencia de las señales obtenidas, pero que tiene una solución relativamente sencilla que veremos más adelante.

Vamos pues a abordar la situación y explicar cómo intentaremos llegar a la separación usando el método descrito en el dominio de la frecuencia. Para una información más amplia del método aquí desarrollado se puede leer [Makino05].

6.4.1 Planteamiento del problema.

El sistema de mezcla es el mismo que planteamos en la sección 4.4, donde las señales grabadas por los micrófonos vienen dadas por:

$$x_i(t) = \sum_{j=1}^{N} \sum_{\tau} a_{ij}(\tau) s_j(t-\tau) \tag{6.36}$$

como ya sabemos, $a_{ij}(t)$ es la respuesta impulsiva de la habitación desde la localización de la fuente *j* hasta el micrófono *i*; $s_j(t)$, $j = 1,\ldots,N$ son las fuentes independientes y $x_i(t)$, $i = 1,\ldots,M$ son las fuentes grabadas por los micrófonos.

La meta de BSS es recuperar las señales fuente conociendo sólo las señales grabadas por los micrófonos, aproximando las señales obtenidas en la separación mediante las señales $y_j(t)$. A partir de ahora consideraremos que tenemos el mismo número de grabaciones que de fuentes, digamos *N*. Por lo tanto, la solución buscada sería:

$$y_i(t) = \sum_{j=1}^{N} \sum_{\tau} w_{ij}(\tau) x_j(t-\tau) \tag{6.37}$$

lo que nos daría una estimación de las señales fuente a partir de las señales grabadas por los sensores. Lo que tendremos que estimar de alguna forma son los coeficientes de los filtros, $w_{ij}(\tau)$, aunque como ya veremos no nos va a hacer falta conocer la expresión temporal de dichos filtros, puesto que lo resolveremos todo en el dominio de la frecuencia.

Usaremos la representación tiempo-frecuencia dada por la transformada de Fourier de corta duración (STFT) para ejecutar la separación. De esa forma, cada señal procedente de los micrófonos dará lugar a una matriz:

$$X_i(n,f) = \sum_{m=-\infty}^{\infty} x_i(m) w(n-m) e^{-j2\pi fm} \tag{6.38}$$

Hemos optado ahora por trabajar con la frecuencia en hertzios en vez de con la frecuencia angular, que veníamos usando hasta ahora. Cabe destacar que esto no cambia nada, ya que tienen una relación proporcional dada por $\omega = 2\pi f$. Usando una transformada de Fourier de *L* puntos, la variable de la frecuencia tomará los valores $f \in \left\{0, \frac{1}{L} f_s, \ldots, \frac{L-1}{L} f_s\right\}$, donde f_s es la frecuencia de muestreo de las señales en el dominio del tiempo.

En este dominio las mezclas convolutivas pueden aproximarse en cada canal de frecuencia como mezclas instantáneas de la siguiente forma:

$$X_i(n,f) = \sum_{i=1}^{N} h_{ji}(f) S_i(n,f) \tag{6.39}$$

donde $h_{ji}(f)$ es la respuesta en frecuencia desde la fuente *i* al sensor *j*, y $S_i(n,f)$ es la STFT de la versión muestreada de $s_i(t)$, obtenida de la forma que se muestra en (6.38). La notación vectorial para este sistema de mezcla es:

$$\mathbf{X}(n,f) = \sum_{i=1}^{N} \mathbf{h}_i(f) S_i(n,f) \tag{6.40}$$

donde $\mathbf{X} = [X_1,\ldots,X_N]^{\mathrm{T}}$ es un vector de muestras de los puntos de la transformada STFT de las señales grabadas por los sensores y $\mathbf{h}_i = [h_{1i},\ldots,h_{Ni}]^{\mathrm{T}}$ es el vector de las respuestas en frecuencia de los caminos existentes entre las fuentes s_i hacia los N sensores.

Para obtener las respuestas en frecuencia $w_{ij}(f)$ de los filtros, hay que solucionar el modelo ICA de valores complejos

$$\mathbf{Y}(n,f) = \mathbf{W}(f)\mathbf{X}(n,f) \tag{6.41}$$

en cada banda de frecuencia, donde $\mathbf{Y} = [Y_1,\ldots,Y_N]^{\mathrm{T}}$ es un vector que contiene las señales separadas, $\mathbf{W} = [\mathbf{w}_1,\ldots,\mathbf{w}_N]^{\mathrm{H}}$ es una matriz de separación $N \times N$, $\mathbf{w}_i = [w_{i1},\ldots,w_{iN}]^{\mathrm{H}}$, y $w_{ij} = [\mathbf{W}]_{ij}$. Estos vectores serán diferentes en cada banda de frecuencia, por supuesto.

Calcular la pseudoinversa $\mathbf{W}^{+}$ (que en el caso de matrices cuadradas se reduce a la inversa $\mathbf{W}^{-1}$) de $\mathbf{W}$ como:

$$\mathbf{W}^{+} = [\mathbf{b}_1,\ldots,\mathbf{b}_N] \tag{6.42}$$

$$\mathbf{b}_i = [b_{1i},\ldots,b_{Ni}]^{\mathrm{T}} \tag{6.43}$$

será muy útil para ajustar el escalado, como veremos posteriormente. También nos servirá para darnos cuenta de lo siguiente. No es difícil hacer que **W** sea invertible usando ICA. Si multiplicando ambos lados de (6.41) por $\mathbf{W}^{+}$, el vector de muestras de los sensores $\mathbf{X}(n,f)$ queda representado por una combinación lineal de los vectores base $\mathbf{b}_1,\ldots,\mathbf{b}_N$:

$$\mathbf{X}(n,f) = \sum_{i=1}^{N} \mathbf{b}_i(f) Y_i(n,f) \tag{6.44}$$

Como ya dijimos antes, la solución ICA tiene ambigüedades de permutación y escalado, esto es, incluso si permutamos las filas de **W**(*f*) o multiplicamos una fila por una constante, todavía sigue siendo una solución ICA válida. Esto significa que

$$\mathbf{W}(f) \leftarrow \mathbf{\Lambda}(f)\mathbf{P}(f)\mathbf{W}(f) \tag{6.45}$$

es también una solución ICA para cualquier matriz de permutación $\mathbf{P}(f)$ y cualquier matriz diagonal $\mathbf{\Lambda}(f)$. Tendremos que decidir entonces cuales son las matrices $\mathbf{P}(f)$ tal que las señales separadas en el dominio del tiempo contengan todos los componentes en frecuencia de la misma fuente. Solucionar el escalado consiste en decidir $\mathbf{\Lambda}(f)$ en cada frecuencia para que no se amplifiquen aleatoriamente los distintos canales de frecuencia en la señal separada, resultando en una pobre calidad de la señal reconstruida en el dominio del tiempo.

6.4.2 Solución ICA en cada subbanda de frecuencia.

Ya hemos dicho cuál será el primer paso para realizar la separación de las fuentes en el dominio de la frecuencia. Se basa en descomponer en problema global en muchos problemas de separación aislados entre sí. Entonces si tras calcular los espectrogramas, las matrices resultantes tienen *K* filas, correspondientes a *K* valores de la frecuencia, tendremos que solucionar ICA para todos y cada uno de los *K* canales de frecuencia por separado.

Vamos así iterando, comenzando por la frecuencia más baja (lo hacemos así por llevar una ordenación conocida, pero podría hacerse desordenadamente) formando $\mathbf{X}(n, f_k)$, compuesto por las filas *k*-ésimas de $X_i(n, f)$, $i = 1,...,N$ y estimando $\mathbf{W}(f_k)$ mediante algún algoritmo ICA que trabaje con valores complejos, ya que los puntos de la transformada STFT serán en general números complejos. Así calculamos la estima de las filas del espectrograma de las señales fuente correspondientes a esos valores de la frecuencia como $\mathbf{Y}(n, f_k) = \mathbf{W}(f_k)\mathbf{X}(n, f_k)$.

Podríamos pensar que el problema ya está solucionado, pero no es así puesto que desconocemos la ordenación de las fuentes en el dominio de la frecuencia, esto es así porque al haberse realizado la separación de forma independiente para cada frecuencia, la ordenación de los componentes independientes obtenidos no será la misma en general. Además, el escalado de las matrices $\mathbf{W}(f)$ también será aleatorio, y tendremos que buscar alguna forma de fijar una referencia igual para todas las frecuencias.

Observando la figura (6.4) podemos ver gráficamente a lo que nos referimos cuando hablamos de permutaciones locales. Si no solucionamos ese problema, es decir, si reconstruimos las señales al dominio del tiempo con las permutaciones locales, la transformada inversa de Fourier mezcla componentes espectrales pertenecientes a distintas fuentes. Incluso si la separación mediante ICA ha sido perfecta en cada canal de frecuencia, el efecto de las permutaciones locales resulta en una calidad de separación muy pobre o incluso puede provocar que no se aprecie separación en absoluto.

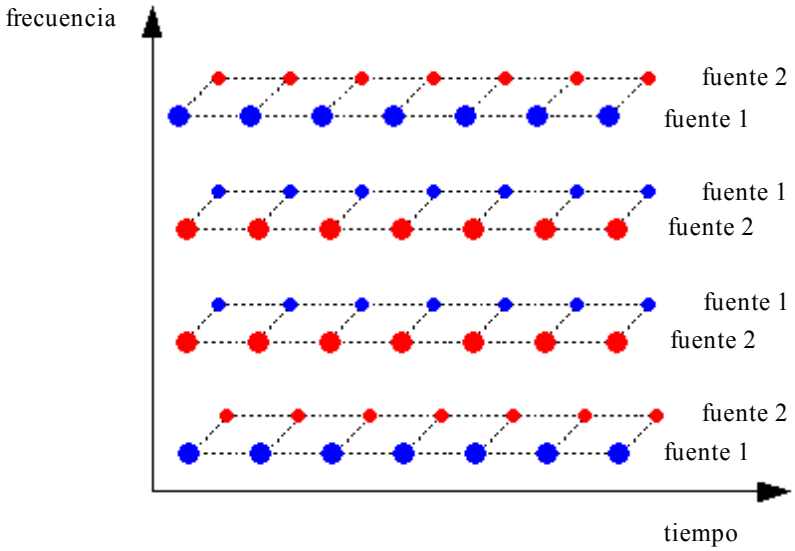


Figura 6.4 *Efecto de las permutaciones locales. Vemos que la ordenación de los componentes está permutada en las distintas bandas de frecuencia.*

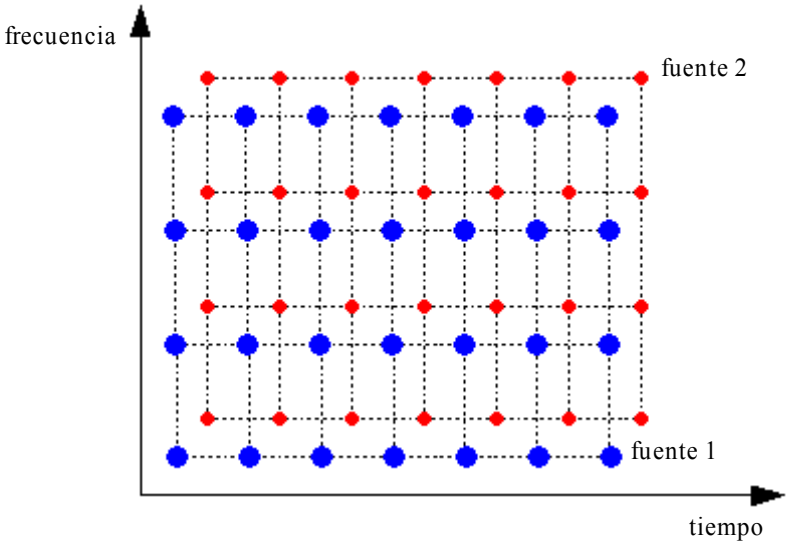


Figura 6.5 *Misma ordenación de los componentes en todos los canales de frecuencia.*

6.4.3 Alineación de las permutaciones locales.

En esta sección vamos a discutir cómo solucionar el problema de las permutaciones. Se han propuesto variadas soluciones a este problema, algunas de las cuales nombramos a continuación:

- La primera estrategia implica aplicar una operación a la matriz de separación $\mathbf{W}(f)$, "suavizándolas" en el dominio de la frecuencia. Esto se hace reduciendo la longitud del filtro mediante enventanado rectangular en el dominio del tiempo, o promediando las matrices de separación en frecuencias adyacentes. Sin embargo, esta operación hace que la matriz

$\mathbf{W}(f)$ sea diferente de la obtenida en la solución ICA, lo que puede ir en detrimento de la calidad de separación.

- La segunda estrategia se basa en utilizar la información contenida en la propia matriz $\mathbf{W}(f)$ para aproximar un modelo de rayos, donde los patrones de directividad sean analizados para identificar la dirección de llegada de cada fuente. Esto se conoce como DOA (Direction Of Arrival).
- En tercer lugar, se puede usar la localización de las fuentes con los vectores base $\mathbf{b}_1(f),\ldots,\mathbf{b}_N(f)$. Este método usa básicamente la misma información que el segundo método, puesto que la matriz de separación $\mathbf{W}(f)$ y los vectores base $\mathbf{b}_1(f),\ldots,\mathbf{b}_N(f)$ están directamente relacionados por la operación de la matriz pseudoinversa. No obstante, la información que se usa en este método es más fácil de manipular puesto que representa directamente el sistema de mezcla.
- La última categoría usa la información de las señales separadas $Y_1(n,f),\ldots,Y_N(n,f)$, empleando las correlaciones entre frecuencias de las mismas. Esta técnica es particularmente efectiva en el caso de señales no estacionarias tales como la voz.

Una vez dicho esto, hemos optado por presentar el último método de los que acabamos de enunciar. Antes de entrar a describirlo en detalle hablaremos un poco de la estructura de la señal de voz, ya que esto ayudará a comprender por qué nos hemos decantado por este método.

Se ha encontrado en la distribución tiempo-frecuencia de la voz, que ésta tiene una fuerte modulación en amplitud, y que ni la información transmitida ni la citada modulación de amplitud son independientes en las diferentes bandas espectrales. Se sabe de la literatura existente sobre procesamiento de voz que la amplitudes en diferentes frecuencias están interrelacionadas.

Así mismo, para las señales de voz, la estructura semántica y la fisiología del mecanismo de producción están consideradas como el origen del parecido observado en diferentes frecuencias. La composición del habla a partir de pequeños elementos (fonemas, sílabas y palabras) contribuye directamente a que la modulación esté interrelacionada en diferentes frecuencias. Las vocales, a su vez, están caracterizadas por picos espectrales simultáneos en las frecuencias de los formantes. Se puede afirmar que la principal fuente de energía en la producción de la voz es la glotis, que emite un sonido de banda ancha con picos espectrales en los armónicos de la frecuencia de pitch de la persona que habla. Por lo tanto, cualquier modulación de la excitación de la glotis afecta a todas las frecuencias simultáneamente. El posterior filtrado por parte del tracto vocal también altera la amplitud de la señal en múltiples frecuencias de forma simultánea.

El algoritmo que vamos a presentar tiene sentido debido precisamente a todas las propiedades que acabamos de decir, debido a la modulación de amplitud ampliamente interrelacionada en canales de frecuencia diferentes e incluso distantes.

Una forma de medir "el parecido" de la modulación de amplitud entre dos canales de frecuencia de dos señales (posiblemente diferentes) es calcular la correlación entre los correspondientes canales de frecuencia. De esta forma tendríamos una manera de saber cuánto se parecen dos señales, o dos bandas de frecuencia entre ellas. La correlación debemos calcularla a partir de la magnitud de las amplitudes $|Y_i(n,f)|$ de las señales separadas, y dicha correlación entre dos secuencias $x(n)$ e $y(n)$ se mide normalmente mediante el coeficiente de correlación:

$$\mathrm{cor}(x,y) = (\mu_{x\cdot y} - \mu_x \mu_y)/(\sigma_x \cdot \sigma_y) \tag{6.46}$$

donde μ_x es la media y σ_x es la desviación típica de *x*. Basándose en la definición, se puede comprobar que cor(*x*,*x*)=1, y cor(*x*,*y*)=0 si *x* e *y* son incorrelados.

Si usamos la amplitud de los espectrogramas de las señales de salida en las distintas bandas de frecuencia y las renombramos como:

$$v_i^f(n) = |Y_i(n,f)| \tag{6.47}$$

podemos observar cómo las amplitudes de una misma señal en frecuencias cercanas se parecen mucho (ver figura 6.6 gráficas superior e inferior) y por lo tanto tendrán una correlación alta. Sin embargo, si comparamos las amplitudes de dos señales separadas pertenecientes a dos personas diferentes en una misma frecuencia vemos (figura 6.6 segunda fila) que no se parecen, y en consecuencia el valor del coeficiente de correlación entre ambas será bajo.

Por lo tanto haciendo uso de las correlaciones entre las subbandas de frecuencia de las señales separadas, un criterio para alinear las permutaciones es el de maximizar la suma de las correlaciones entre los pares de señales de salida para frecuencias cercanas, esto en el caso de dos señales fuente correspondería a evaluar la siguiente expresión:

$$\mathrm{cor}(v_1^f, v_2^{f'}) + \mathrm{cor}(v_2^f, v_1^{f'}) \tag{6.48}$$

tomando todas las posibles matrices permutación $\mathbf{P}(f)$ (en este caso concreto significaría hacer ese cálculo dos veces) y quedarnos con la matriz $\mathbf{P}(f)$ que nos proporcione el valor máximo de dicha expresión.

En el caso general en que tengamos N señales fuente, habría que formar N matrices de permutación, evaluar la suma de las correlaciones de las amplitudes de las subbandas de frecuencia de todas las señales separadas dos a dos, y quedarnos aquella que ofrezca el valor máximo. Así iremos alineando las permutaciones una a una hasta llegar a la última, donde tendríamos todas las componentes espectrales de las señales fuente bien alineadas en las señales separadas.

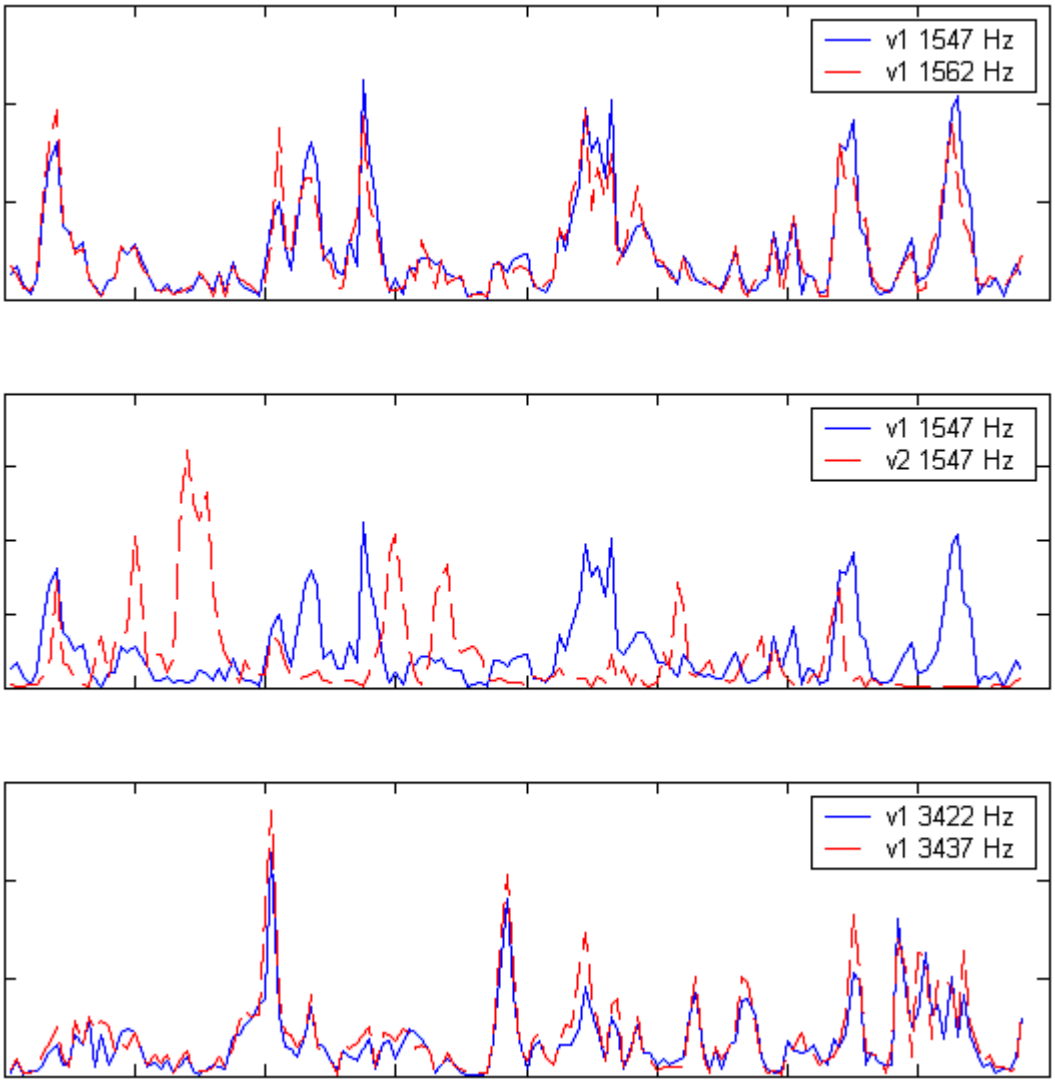


Figura 6.6 *Comparación entre las bandas de frecuencia de varias señales*

En la práctica hemos observado que a veces no es suficiente con evaluar la correlación de las amplitudes de la señal en la frecuencia que queremos alinear tomando la correlación sólo con la frecuencia adyacente que se acaba de alinear, sino que para no cometer errores hace falta correlacionar con más subbandas de frecuencia. Esto aumenta la eficacia del algoritmo pero también aumenta el tiempo de cómputo, por lo que hay que llegar a una solución de compromiso entre ambos, de forma que el algoritmo tenga una garantía de éxito aceptable sin demorarse demasiado en llevar a cabo la separación.

Igualmente también tenemos que mencionar el problema que puede tener este método de solucionar las permutaciones locales, y es que una sola frecuencia mal alineada suele provocar que la separación no sea apreciable, y por lo tanto todo el algoritmo fracase. Esto se debe a que si sucede una permutación en una determinada frecuencia, es de esperar que las posteriores alineaciones se hagan conforme a esa banda de frecuencias mal alineada, y por lo tanto estemos mezclando componentes espectrales de varias señales. Esto puede evitarse en gran medida calculando las correlaciones no sólo entre dos bandas de frecuencia

adyacentes, sino tomando en consideración un conjunto de frecuencias ya alineadas, como acabábamos de comentar.

6.4.4 Ajuste del escalado.

Llegados a este punto, podemos pensar que ya se ha ejecutado con éxito ICA en todos los canales de frecuencia y que las posibles permutaciones de los mismos han sido alineadas. Pues bien, lo único que nos falta antes de poder reconstruir al dominio del tiempo es ajustar el escalado de las subbandas de frecuencia de las señales separadas.

Como ya vimos en la sección dedicada a ICA, en la matriz de separación, además de poder intercambiar las filas sin que esto afectara al éxito del algoritmo, también había otra indeterminación relacionada con la energía de las señales separadas. Podemos multiplicar los componentes independientes por constantes sin que estos dejen de ser indendientes, y esto es lo que sucede en cada subbanda de frecuencia de las señales separadas. Cada canal de frecuencia ha sido hallado aplicando un algoritmo basado en ICA de forma independiente, y por lo tanto el escalado aplicado a las matrices de separación y por consiguiente a las señales separadas es diferente en cada frecuencia. Por otra parte, esto no afectaba en nada a las correlaciones, y este problema no ha sido solucionado al alinear las permutaciones.

Entonces si pensáramos en aplicar la transformada de Fourier inversa a las señales que tenemos ahora mismo para retornar al dominio del tiempo, nos encontraremos con que unos canales de frecuencia están amplificados y otros atenuados dentro de la misma señal, y por lo tanto aunque la forma de onda de la señal sea la correcta en todas las frecuencias, la expresión temporal de la misma tendrá poco o nada que ver con la señal que estamos intentando estimar, es decir, con la señal fuente.

Sin embargo esta ambigüedad se resuelve fácilmente haciendo uso de la pseudoinversa de la matriz de separación $\mathbf{W}(f)$. El objetivo de BSS en el dominio de la frecuencia es que se cumpla para todo *i*:

$$Y_i(n,f) = h_{J_i i}(f) S_i(n,f) \tag{6.49}$$

donde J_i puede seleccionarse de acuerdo a cada salida *i* pero debe ser el mismo para todas las frecuencias *f*. Si tanto ICA como el problema de la permutación han sido solucionados, entonces el término $\mathbf{b}_i$ de (6.44) es parecido al término $\mathbf{h}_i$ de (6.40), es decir:

$$\mathbf{h}_i(f) S_i(n,f) \approx \mathbf{b}_i(f) Y_i(n,f) \tag{6.50}$$

Sustituyendo (6.49) en (6.50), tenemos la condición para la alineación del escalado:

$$\mathbf{h}_i(f) \approx \mathbf{b}_i(f) h_{J_i i}(f) \Leftrightarrow \; b_{J_i i}(f) \approx 1 \tag{6.51}$$

Esta condición, $b_{J_{ii}}(f) = 1$, es alcanzada mediante:

$$\begin{aligned} \mathbf{W}(f) &\leftarrow \mathbf{\Lambda}(f)\mathbf{W}(f) \\ \mathbf{\Lambda}(f) &= \operatorname{diag}\left(b_{J_{11}}(f),\ldots,b_{J_{NN}}(f)\right) \end{aligned} \tag{6.52}$$

donde $b_{ji}(f) = \left[\mathbf{W}^{+}(f)\right]_{ji}$ es un elemento de la pseudoinversa de $\mathbf{W}(f)$.

Tras hacer esta transformación sí que podemos ya aplicar la STFT inversa a las señales $Y_i(n,f)$ y obtener la expresión temporal de las señales separadas, que deben ser una buena estimación de las señales fuente.

6.5 Conclusiones.

Hemos expuesto tres métodos diferentes de separación, aplicables cada uno en situaciones diferentes. El primer método que vimos, el de separación mediante enmascaramiento, dejaba abierta la posibilidad de estimar las fuentes independientes en grabaciones estereofónicas donde puede que haya más de dos personas hablando, esto es, sistemas sobredeterminados. El concepto en el que se basa este método es la estimación del ángulo con que llegan los frentes de onda de voz a los micrófonos y la suposición que debe cumplirse es que las señales fuente sean disjuntas en el dominio tiempo-frecuencia.

El segundo método es quizás el más débil de los tres, pues como veremos en la sección de simulaciones, necesita de la existencia de tantas grabaciones como fuentes estén presentes en la mezcla, y además está pensado exclusivamente para entornos anecoicos, porque ésta es la simplificación que se hace al definir el modelo de mezcla que luego será estimado. Para la estimación de dicho modelo hemos hecho uso de los cumulantes.

Por último, hemos visto una técnica de separación que, en teoría, debe funcionar con éxito en situaciones acústicas más adversas que las dos anteriores, pues no se realiza simplificación alguna en el modelo de mezcla, suponiendo que ésta se ha hecho de forma convolutiva. Separábamos el problema global en pequeños problemas que solucionamos por separado, y luego aplicamos los conocimientos que tenemos sobre la señal de voz para unificar todos esos resultados. Este método se separación también requiere tantas grabaciones como voces se quieran separar, pero si se cumple esto, cualquier mezcla que haya podido separarse usando alguno de los métodos anteriores debe separarse también con éxito aplicando este, cosa que en general no sucederá a la inversa.

Capítulo 7

Simulaciones

7.1 Introducción.

Este capítulo del proyecto está dedicado exclusivamente al aspecto práctico de la realización del mismo. Presentaremos varias simulaciones de las muchas que hemos realizado para comprobar los resultados que ofrecen los métodos de separación descritos en el capítulo 6.

Todos los algoritmos que hemos usado para la realización de las simulaciones han sido desarrollados en Matlab 6.5. Los resultados que presentamos también han sido obtenidos con el mismo programa. Para hacer más sencilla la interacción con las funciones escritas en Matlab, y además facilitar que una persona ajena al proyecto pueda también realizar sus propias pruebas y comprobar el éxito o fracaso de los métodos de separación descritos, hemos generado un entorno gráfico para cargar los datos, elegir los parámetros de las simulaciones y ver los resultados. Para hacer esto recurrimos a una herramienta que nos proporciona Matlab denominada *GUI* (Graphical User Interface) o interfaz gráfica de usuario.

Las *GUI*'s son interfaces de usuario cuyos componentes son objetos tales como botones, textos, barras deslizantes, menús, etc. Una *GUI* proporciona una interfaz entre una aplicación y un usuario, que permite a este último manejar la aplicación sin la necesidad de conocer las instrucciones que se ejecutarán en la ventana de comandos de Matlab, ocultando al mismo la complejidad de la programación que se esconde detrás de los algoritmos. Para más información sobre la herramienta gráfica que hemos creado se puede acudir al apéndice 3.

7.2 Descripción de la situación de partida.

Vamos a ver primero los resultados obtenidos en la separación de dos señales de voz a partir de una mezcla de las mismas generada digitalmente simulando una situación anecoica a partir de las señales grabadas individualmente. Partiendo de esta mezcla en la que hay presentes dos fuentes de voz y dos grabaciones de las mismas, vamos a presentar los resultados de la separación empleando dos de los tres métodos descritos en la sección anterior, simulando luego una situación en la que hay tres fuentes para ver que estos algoritmos pueden extenderse a casos con más fuentes.

Las mezclas digitales se han realizado siguiendo un modelo anecoico, por lo que han sido realizadas en el dominio de la frecuencia y posteriormente transformadas al dominio del tiempo.

La situación que simularemos primero se corresponde con la de la figura (7.1), donde la separación entre los micrófonos que se ha tomado es $d = 1.1\,\text{cm}$, y los ángulos de llegada de las fuentes son $\theta_1 = 14.4^\circ$ y $\theta_2 = 86.4^\circ$, con la referencia que se muestra en la ilustración.

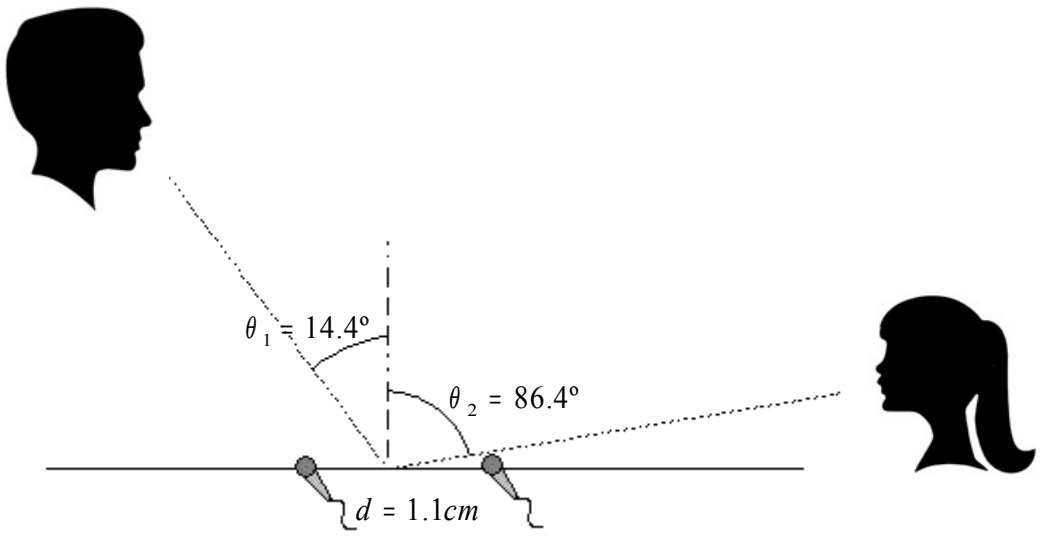


Figura 7.1 *Descripción de la situación en que se genera la mezcla.*

Para calcular los espectrogramas en las simulaciones hemos escogido los siguientes parámetros:

- Tipo de ventana: Hanning
- Duración de la ventana: 30 milisegundos
- Número de puntos de la FFT: 1024

En las figuras (7.2) y (7.3) mostramos los dos espectrogramas de las mezclas, cada uno correspondiente a un micrófono. En la parte inferior de las figuras puede observarse también la expresión temporal de las señales grabadas.

A simple vista no podemos observar grandes diferencias entre los espectrogramas salvo pequeños detalles, y es que como ya comentamos en la sección dedicada a los modelos de mezcla de la voz, las diferencias son muy sutiles debido a que los retardos entre los dos micrófonos son muy pequeños dada su cercanía y la alta velocidad de propagación del sonido, y lo mismo sucede con las atenuaciones. Esto provoca que aparentemente las grabaciones sean iguales, pero en realidad no los son, por que si fueran exactamente iguales sería imposible la separación mediante estos métodos.

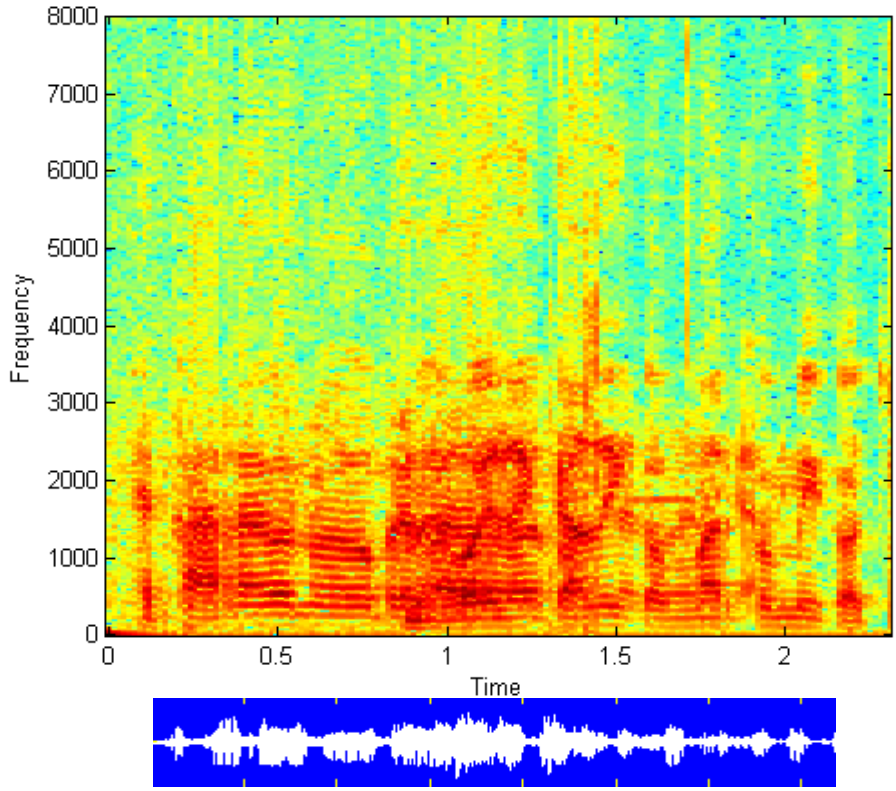


Figura 7.2 *Espectrograma de la señal grabada por el primer micrófono y su expresión temporal.*

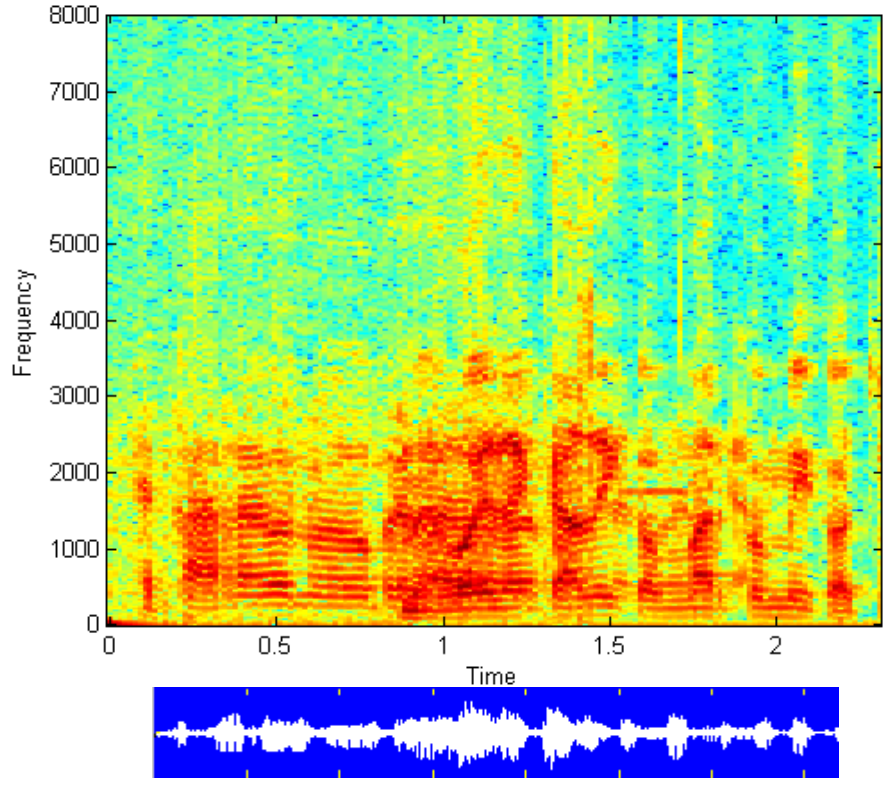


Figura 7.3 *Espectrograma y expresión temporal de la señal grabada por el segundo micrófono.*

7.3 Separación usando enmascaramiento.

Partiendo de la situación que acabamos de describir, vamos a ver ahora los resultados obtenidos empleando *masking* como método de separación.

Tras operar de la forma detallada en la sección (6.2.2), el histograma de retardos obtenidos se muestra en la figura (7.5). Esto se ha conseguido tomando los datos de los espectrogramas de las grabaciones y haciendo uso del algoritmo EM que presentamos en la sección (6.2.3). Vemos en la figura que se pueden distinguir claramente cómo los valores se concentran en torno a dos picos que se corresponden con los retardos debidos a las dos direcciones principales de propagación de los dos frentes de onda de las fuentes. A partir de ese histograma se obtienen las medias y las varianzas de los retardos, que son usadas para hallar las máscaras que serán aplicadas al espectrograma del micrófono de referencia y así extraer las estimaciones de las señales fuente.

En la figura (7.5) se muestran los dos ángulos de llegada, los cuales son calculados directamente a partir de los datos de los retardos. Las máscaras que nos ha proporcionado el algoritmo y que aplicamos al espectrograma del micrófono de referencia son las dibujadas en la figura (7.6).

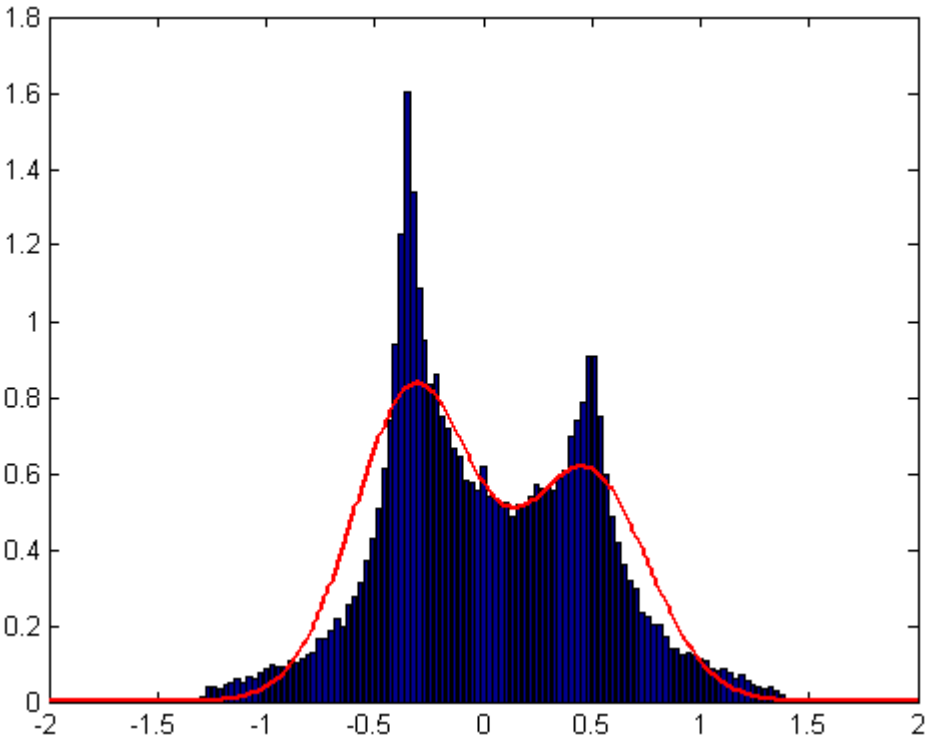


Figura 7.4 *Histograma de retardos estimados a partir de los espectrogramas de las grabaciones.*

Y tras aplicar esas máscaras, obtenemos los espectrogramas que mostramos en las figuras (7.7) y (7.9). También hemos querido representar la visualización del espectrograma de las señales fuente originales para que puedan ser comparados

con los obtenidos por el algoritmo de separación y valorar sus similitudes y diferencias.

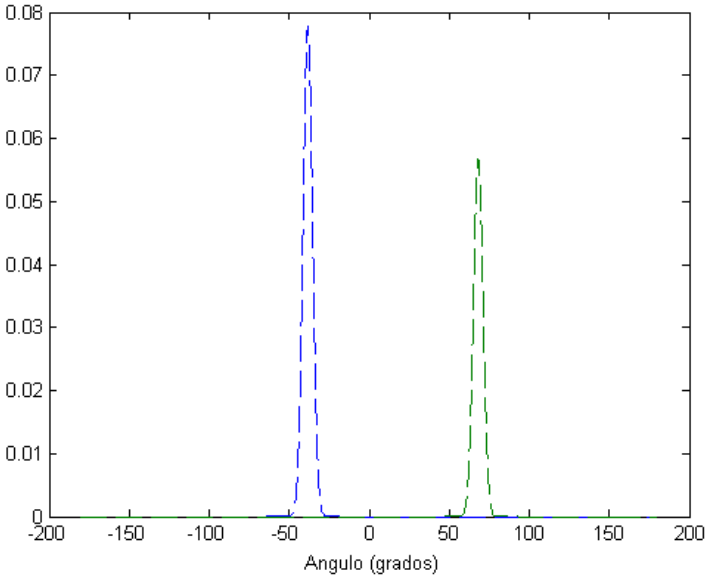


Figura 7.5 *Ángulos estimados para las dos fuentes.*

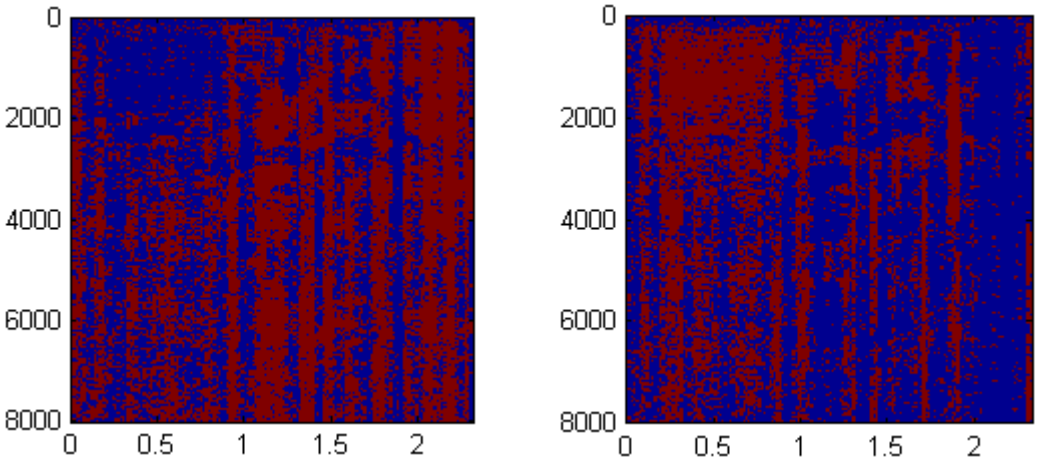


Figura 7.6 *Máscaras aplicadas a los espectrogramas del micrófono de referencia para hallar las estimaciones de las señales fuente. El eje vertical es la frecuencia (Hz) y el horizontal el tiempo (s).*

Finalmente, las figuras (7.11) y (7.12) muestran las señales correspondientes a las reconstrucciones de los espectrogramas al dominio del tiempo, junto con las señales fuente originales a las que queríamos que se parecieran lo más posible. Al escuchar las señales obtenidas se puede percibir que la separación se ha llevado a cabo de manera muy satisfactoria.

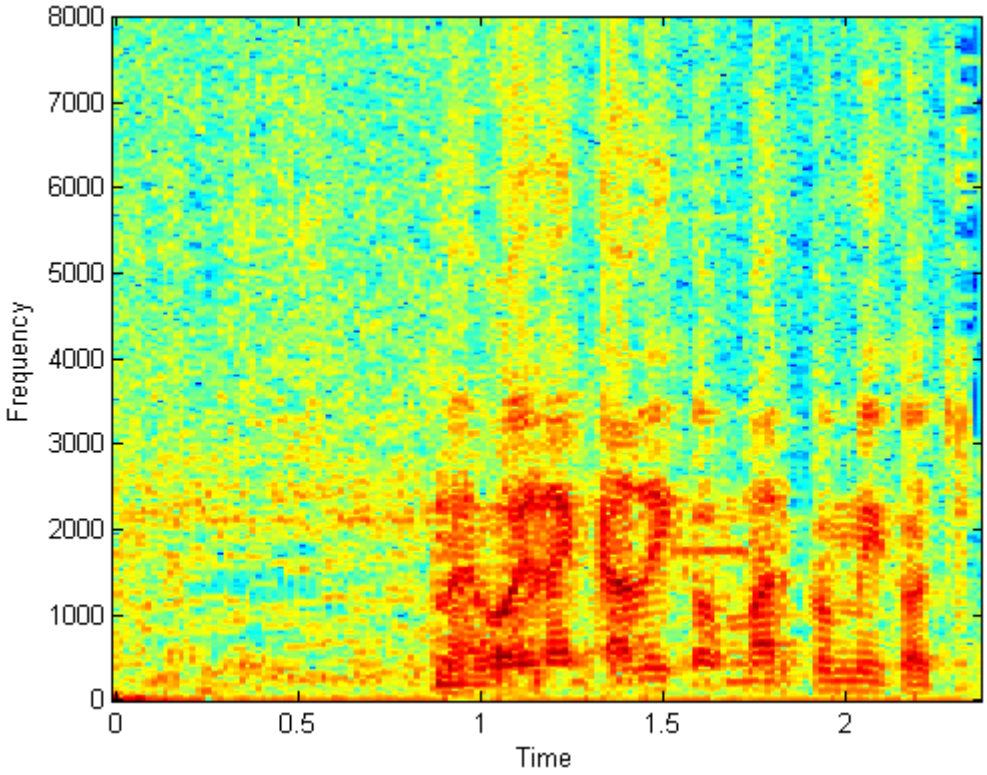


Figura 7.7 *Espectrograma de la primera señal obtenida mediante masking.*

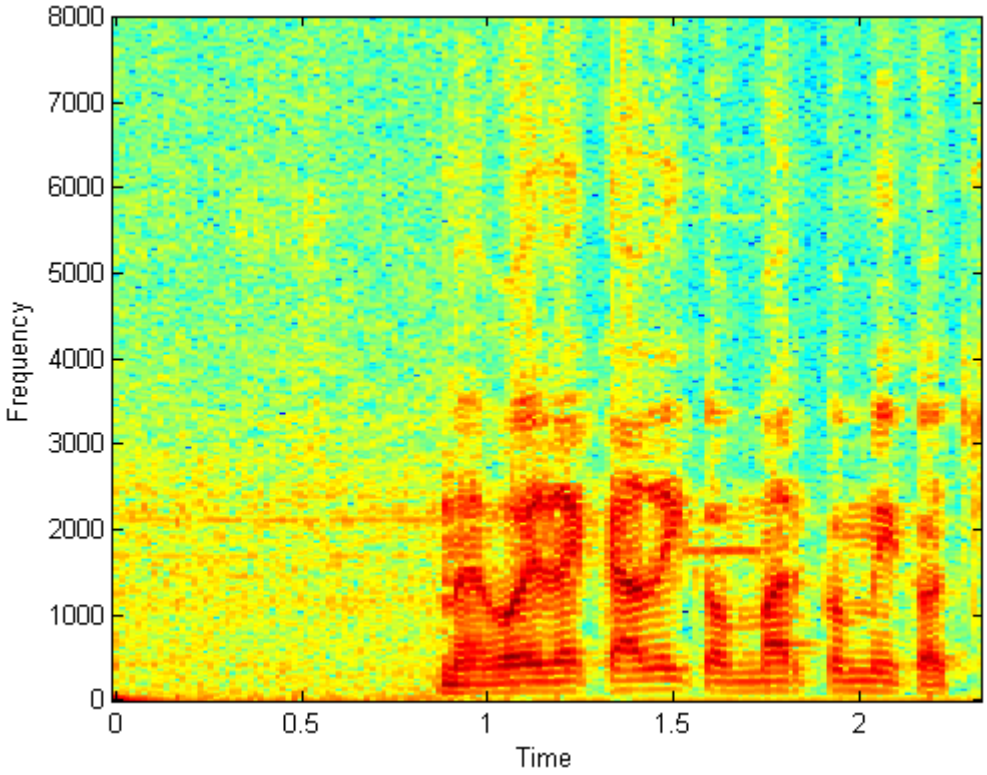


Figura 7.8 *Espectrograma de la señal fuente original.*

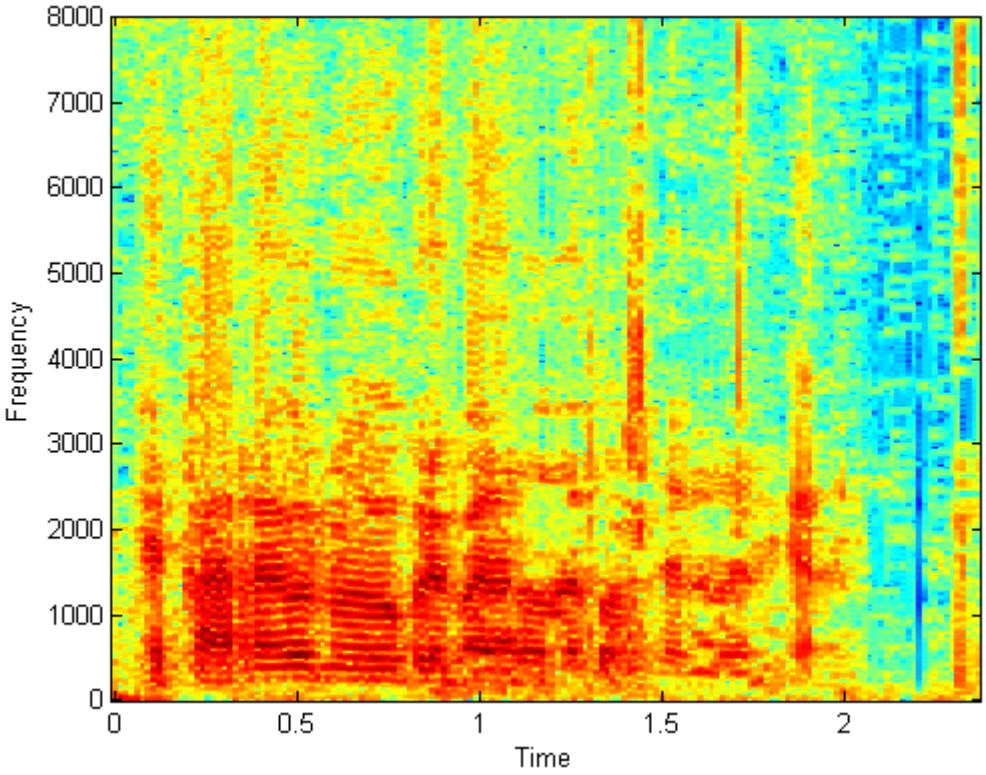


Figura 7.9 *Espectrograma de la segunda señal obtenida mediante masking.*

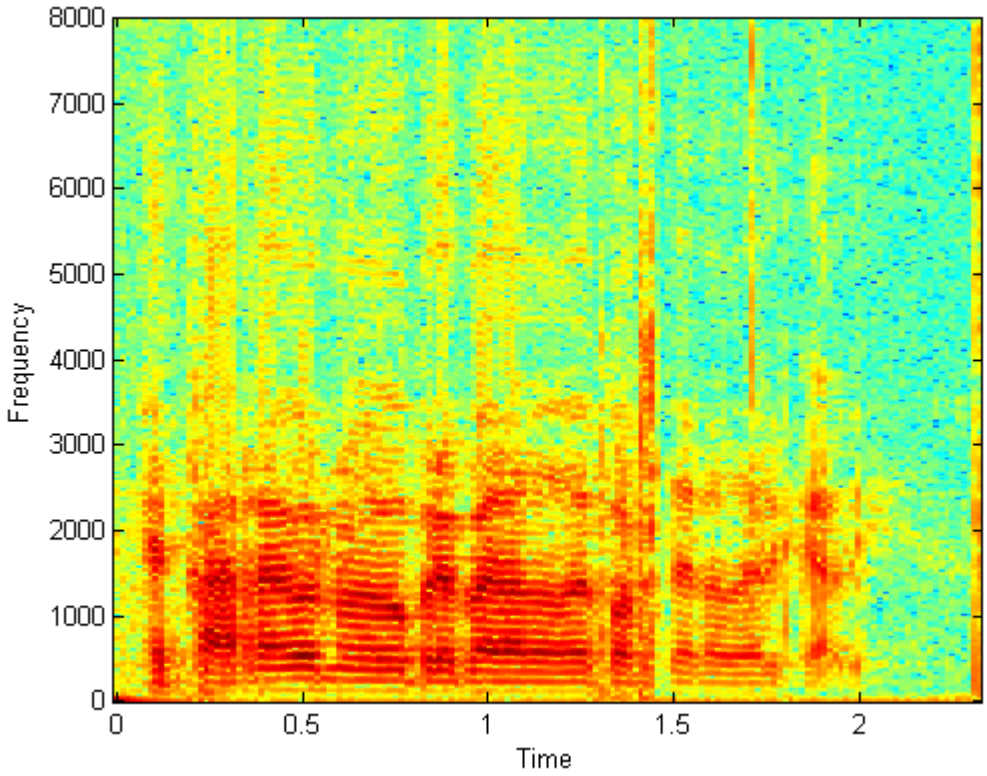


Figura 7.10 *Espectrograma de la señal fuente original.*

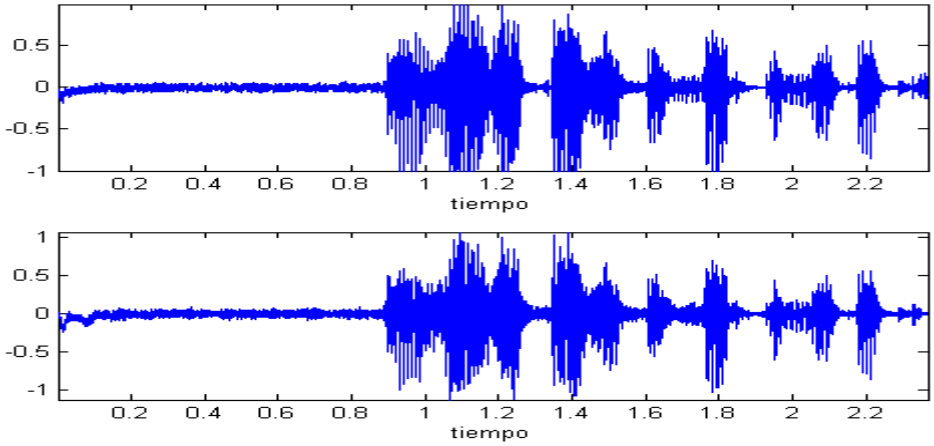


Figura 7.11 *Señal fuente original (arriba) y señal fuente estimada mediante masking (abajo).*

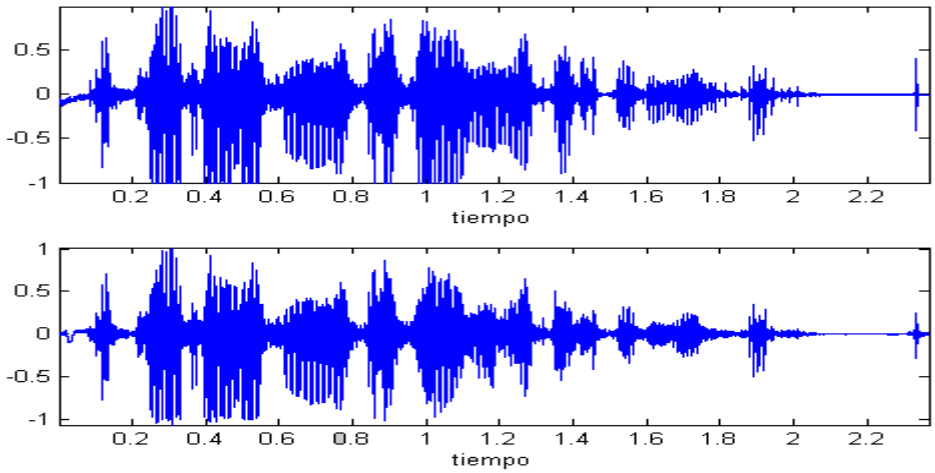


Figura 7.12 *Señal fuente original (arriba) y señal fuente estimada mediante masking (abajo).*

Por último, haremos una observación sobre una de las hipótesis que era clave para el correcto funcionamiento de este método de separación. Recordamos que postulamos como requisito imprescindible que los espectrogramas de las señales fuente fueran disjuntos. Esto se puede comprobar matemáticamente calculando el producto de ambos espectrogramas en cada punto como vimos en la ecuación (6.6). Si hacemos esto debemos obtener valores nulos o muy cercanos a 0 para todos los puntos, en caso contrario los espectrogramas no serán disjuntos en el dominio tiempo-frecuencia y el algoritmo no tiene por qué funcionar. Pues bien, hemos hecho esta operación y representamos la magnitud de los productos para todas las bandas de frecuencia en la figura (7.13). Inmediatamente nos damos cuenta de que los espectrogramas no son disjuntos en absoluto, solapando en multitud de puntos comprendidos en frecuencias bajas. Aun así, los resultados de la separación son convincentes, lo que pone de manifiesto la robustez del algoritmo, así como su posible mejoría cuando los espectrogramas de las fuentes sí sean disjuntos.

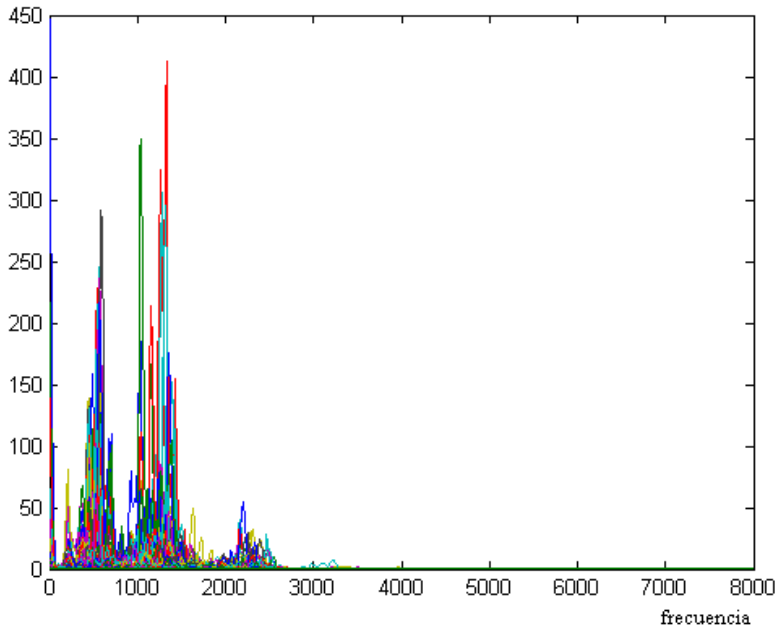


Figura 7.13 *Magnitud del producto de los espectrogramas. Se representa la superposición de todas las bandas de frecuencia.*

7.4 Separación usando el método de Anemüller.

Simularemos ahora el mismo ejemplo que en el apartado anterior pero ejecutando el método de separación de Anemüller. Las expresiones temporales de las señales obtenidas, así como la de las fuentes originales, se muestran en las figuras (7.14) y (7.15). Se puede percibir claramente la separación y al escucharlas nos damos cuenta de que son perfectamente inteligibles.

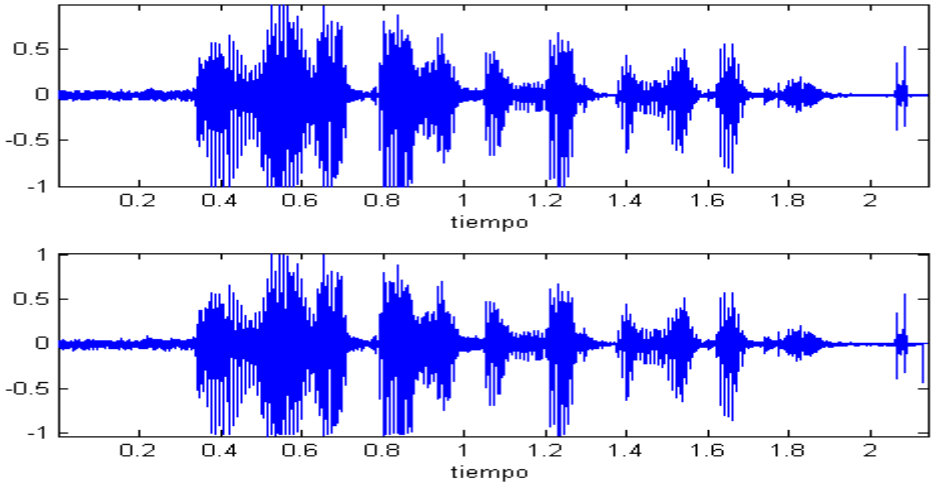


Figura 7.14 *Señal fuente original (arriba) y señal fuente estimada mediante el método de Anemüller (abajo).*

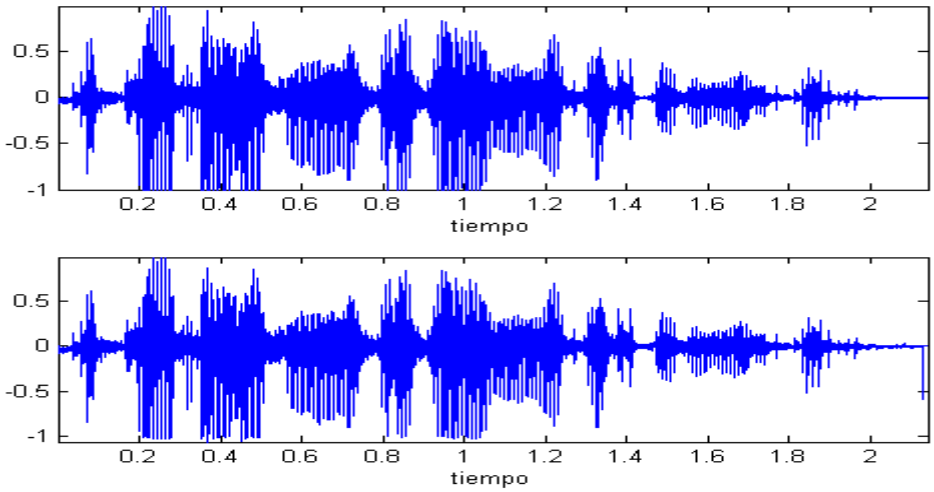


Figura 7.15 *Señal fuente original (arriba) y señal fuente estimada mediante el método de Anemüller (abajo).*

En las figuras (7.16) y (7.17) podemos ver los espectrogramas de las señales estimadas. No hemos representado los de las señales fuente originales de nuevo puesto que son los mismos de las figuras (7.8) y (7.10), y puede acudirse a ellas para observarlos.

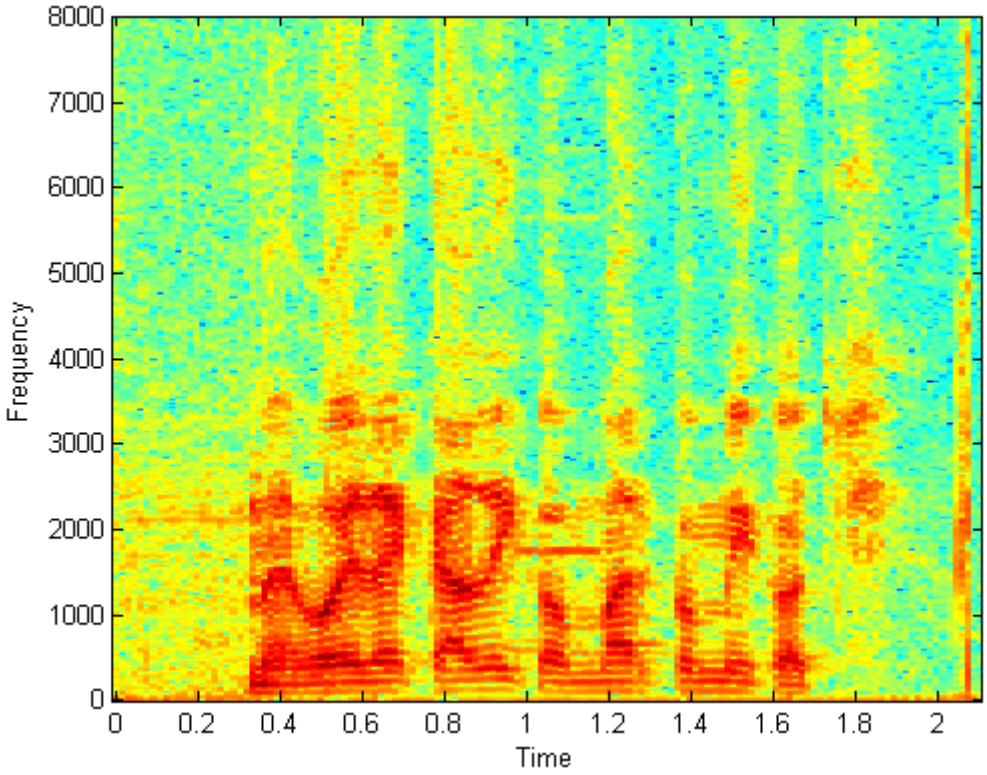


Figura 7.16 *Espectrograma de la primera señal obtenida mediante Anemüller.*

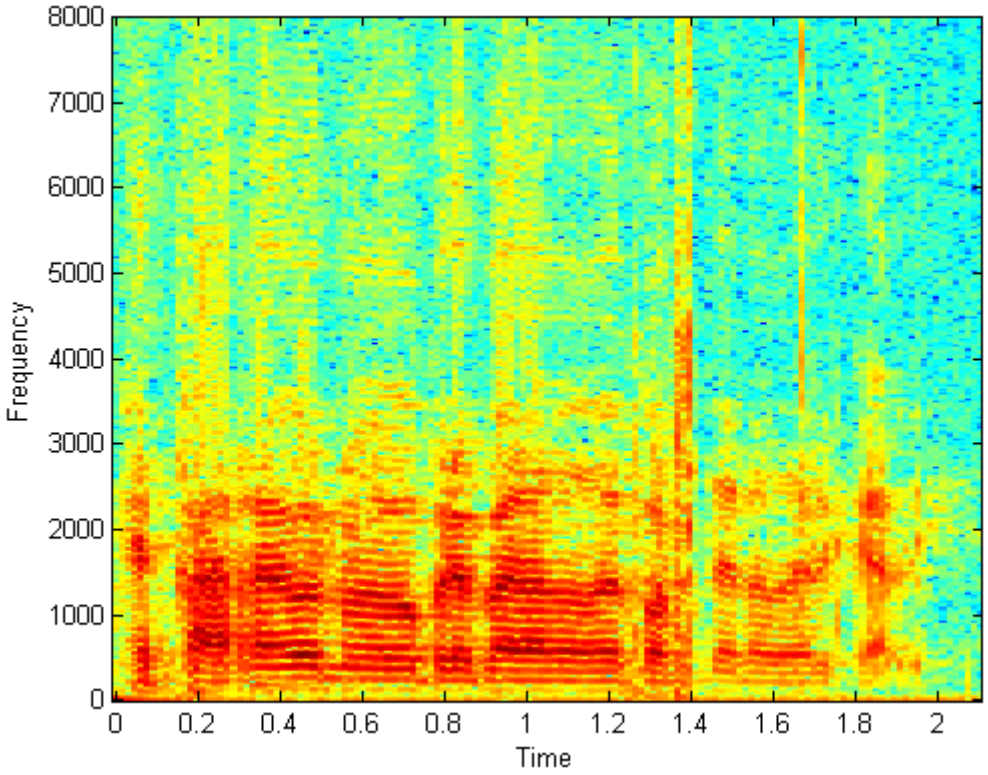


Figura 7.17 *Espectrograma de la segunda señal obtenida número uno mediante Anemüller.*

La matriz de mezcla usada para generar las señales que tomamos como entrada del sistema de separación tenía los siguientes coeficientes de amplitud:

$a_{11} = 1$ $\quad$ $a_{12} = 0.9786$

$a_{21} = 0.9691$ $\quad$ $a_{22} = 1$

Los retardos de las fuentes correspondientes a esos ángulos de llegada y esa distancia entre los micrófonos eran:

$\tau_{12} = 0.2189 \cdot 10^{-4} s \quad \tau_{21} = 0.3191 \cdot 10^{-4} s$.

Pues bien, los coeficientes de la matriz de mezcla que ha estimado el algoritmo son:

$\hat{a}_{12} = 0.9682$ $\quad$ $\hat{a}_{21} = 0.9573$

mientras que los retardos estimados han sido:

$\hat{\tau}_{12} = 0.2229 \cdot 10^{-4} s \quad \hat{\tau}_{12} = 0.3142 \cdot 10^{-4} s$.

Podemos comprobar que la estima ha sido muy buena, y hemos usado la versión batch del algoritmo basado en cumulantes que planteamos en el capítulo 6, por lo que la convergencia a la solución ha sido muy rápida.

7.5 Separación usando ICA independientemente en cada subbanda.

En este apartado no vamos a mostrar la simulación del mismo ejemplo que en los dos anteriores, aunque también lo hemos resuelto y los resultados son muy satisfactorios, sino que vamos a plantear una situación algo más compleja, tratando ahora de separar una mezcla de tres señales fuente realizada (digitalmente) por tres micrófonos.

Los datos de la situación son: la distancia entre los micrófonos es de 1.5 cm, tanto entre el primer y segundo micrófono, como entre el segundo y el tercero, por lo tanto hay tres centímetros de separación entre el primer y el tercer micrófono. Los ángulos de llegada de cada una de las tres señales son $\theta_1 = -25.2^{\circ}$, $\theta_2 = 46.8^{\circ}$ y $\theta_3 = 82.8^{\circ}$. Para realizar el cálculo de las transformadas STFT hemos escogido los siguientes parámetros:

- Tipo de ventana: Hanning
- Duración de la ventana: 30 milisegundos
- Número de puntos de la FFT: 512

Hemos tomado una longitud menor de la FFT para disminuir el volumen de datos que hay que tratar y por lo tanto hacer que el algoritmo llegue a la solución más rápidamente, ya que éste es el más costoso de los tres algoritmos computacionalmente hablando.

Un esquema de la situación que se simula en la mezcla y que acabamos de describir numéricamente puede verse en la figura (7.18).

En la figura (7.19) se muestran el espectrograma y la señal en tiempo grabada por el primero de los tres micrófonos. No mostramos las otras dos grabaciones puesto que no se observan diferencias significativas entre ellas y la que sí mostramos, por lo que no resultan interesantes.

En las figuras (7.20), (7.21) y (7.22) se comparan las señales obtenidas en la separación con las fuentes originales, y puede comprobarse su gran parecido. La calidad de las mismas al ser escuchadas es muy alta, de hecho este método consigue posiblemente los mejores resultados en cuanto a calidad de señal de los tres métodos propuestos en este proyecto y en una variedad mayor de situaciones. Sin embargo, emplea más tiempo para llegar a la solución, debido principalmente al algoritmo que alinea correctamente las permutaciones. Este tiempo puede disminuirse haciendo correlaciones entre menos bandas de frecuencia, pero también la fiabilidad del mismo podría resentirse. En las figuras (7.23) a (7.28) se pueden observar los espectrogramas obtenidos y los de las fuentes originales que fueron usadas para realizar la mezcla y puede contemplarse su gran parecido.

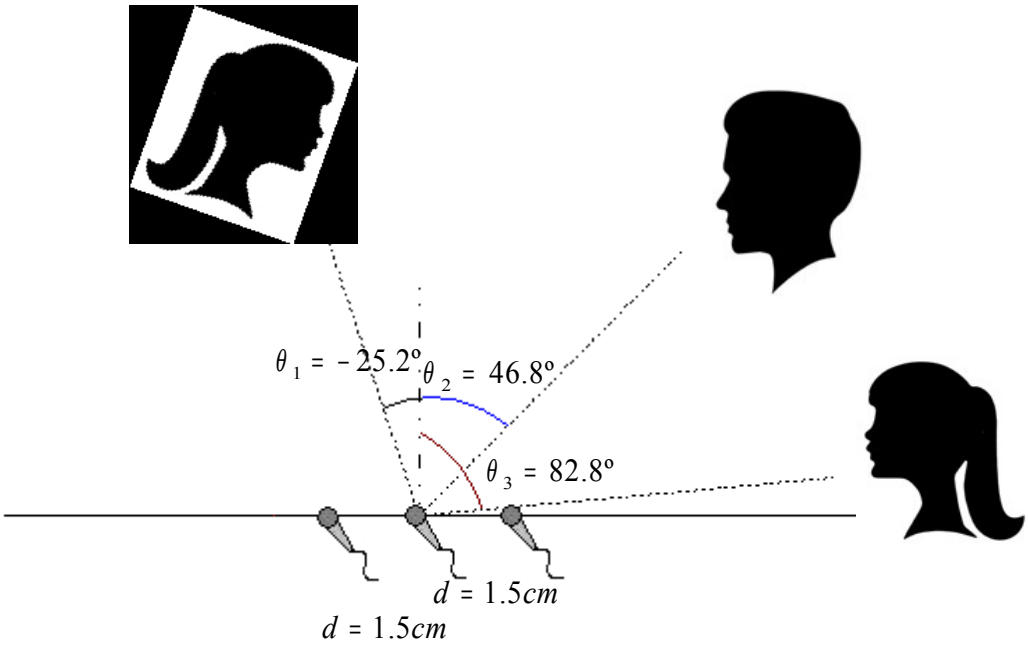


Figura 7.18 *Descripción de la situación en que se genera la mezcla en este caso.*

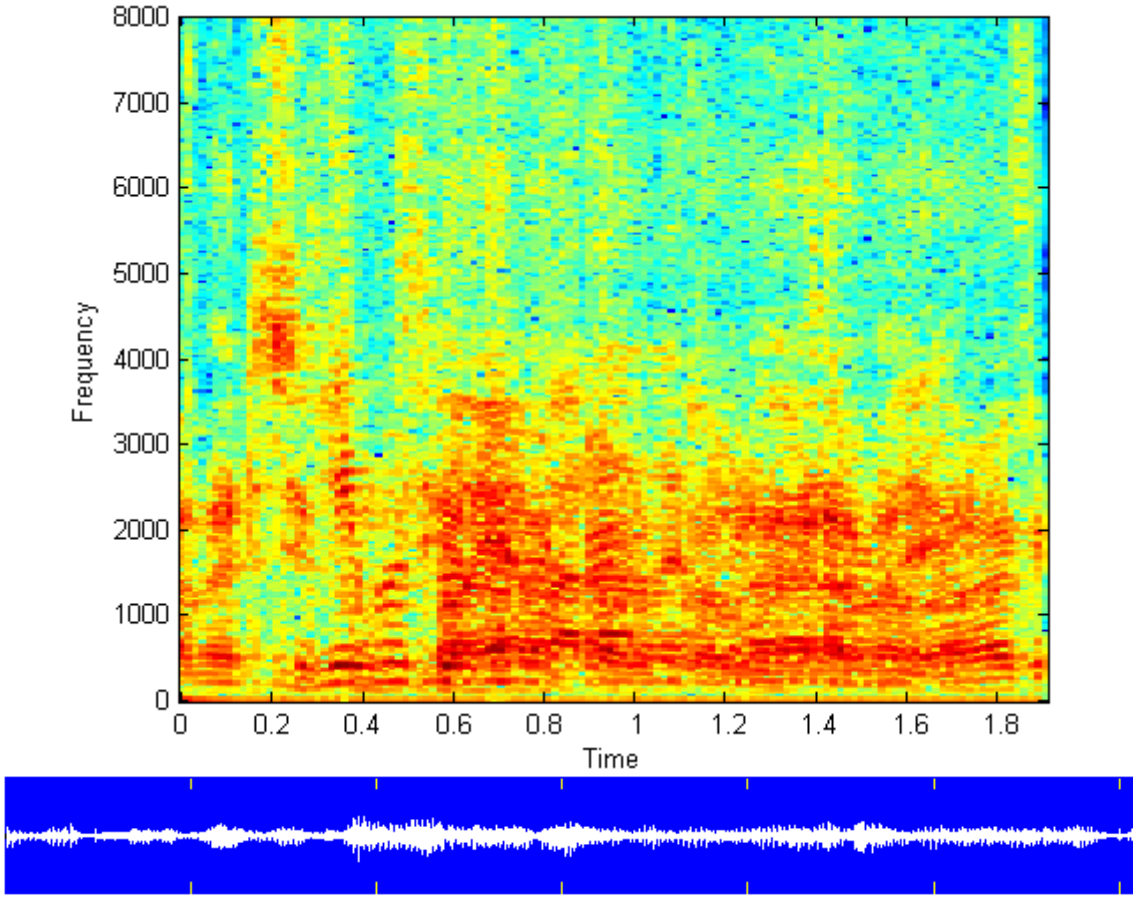


Figura 7.19 *Espectrograma y expresión temporal de la señal grabada por uno de los micrófonos.*

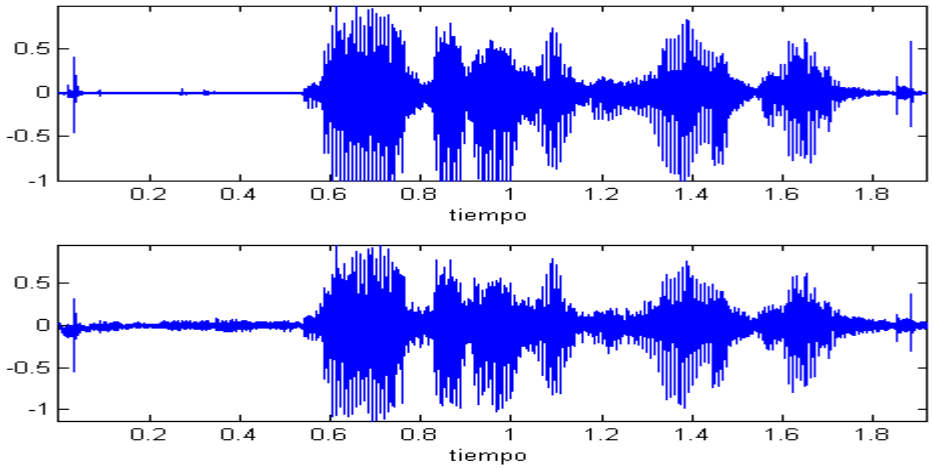


Figura 7.20 *Señal fuente original (arriba) y señal fuente estimada (abajo).*

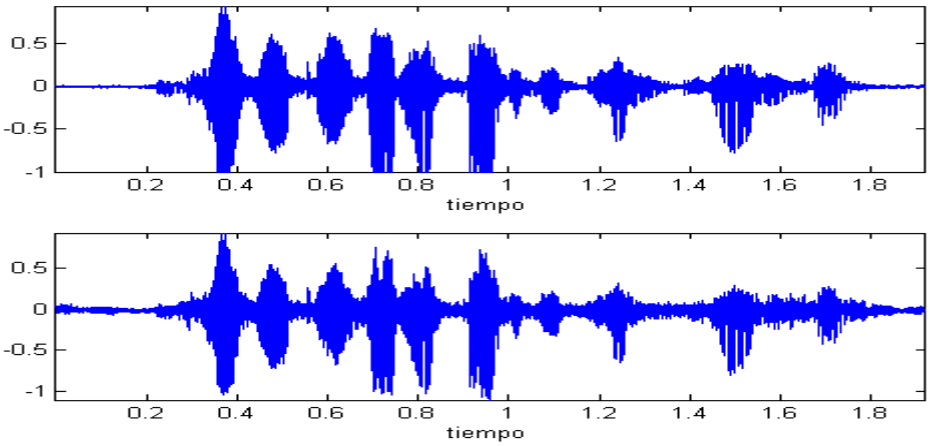


Figura 7.21 *Señal fuente original (arriba) y señal fuente estimada (abajo).*

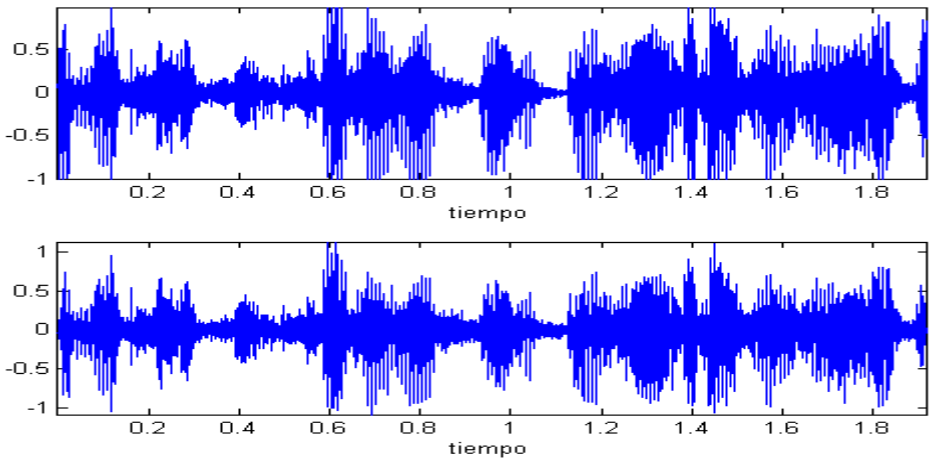


Figura 7.22 *Señal fuente original (arriba) y señal fuente estimada (abajo).*

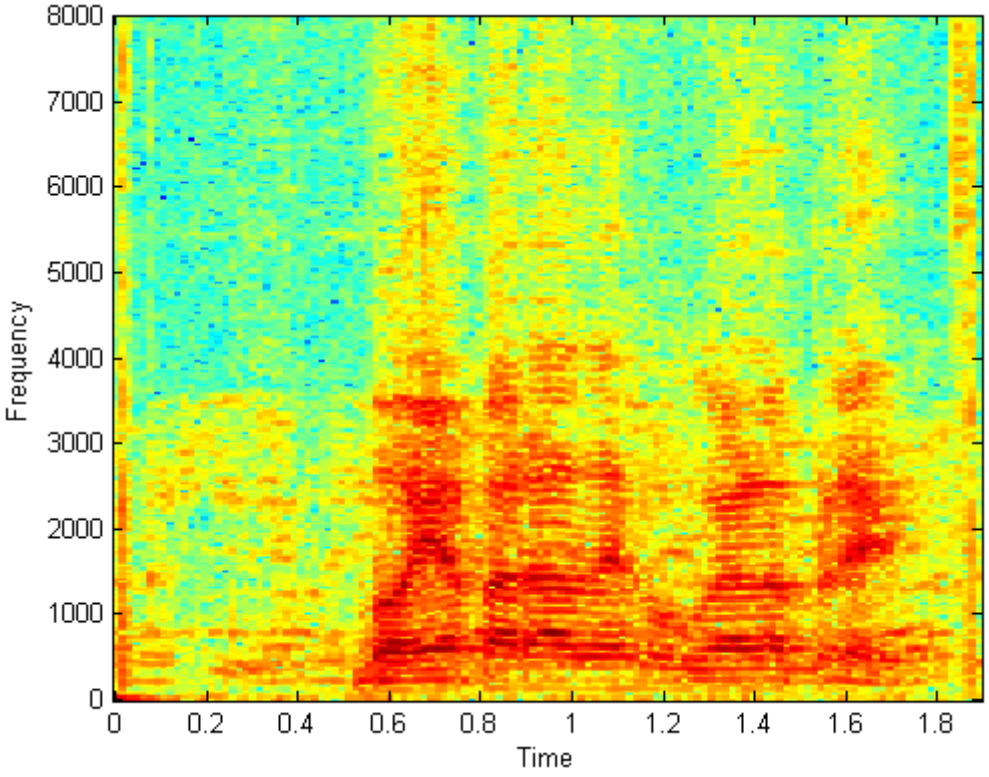


Figura 7.23 *Espectrograma de la primera señal obtenida mediante Anemüller.*

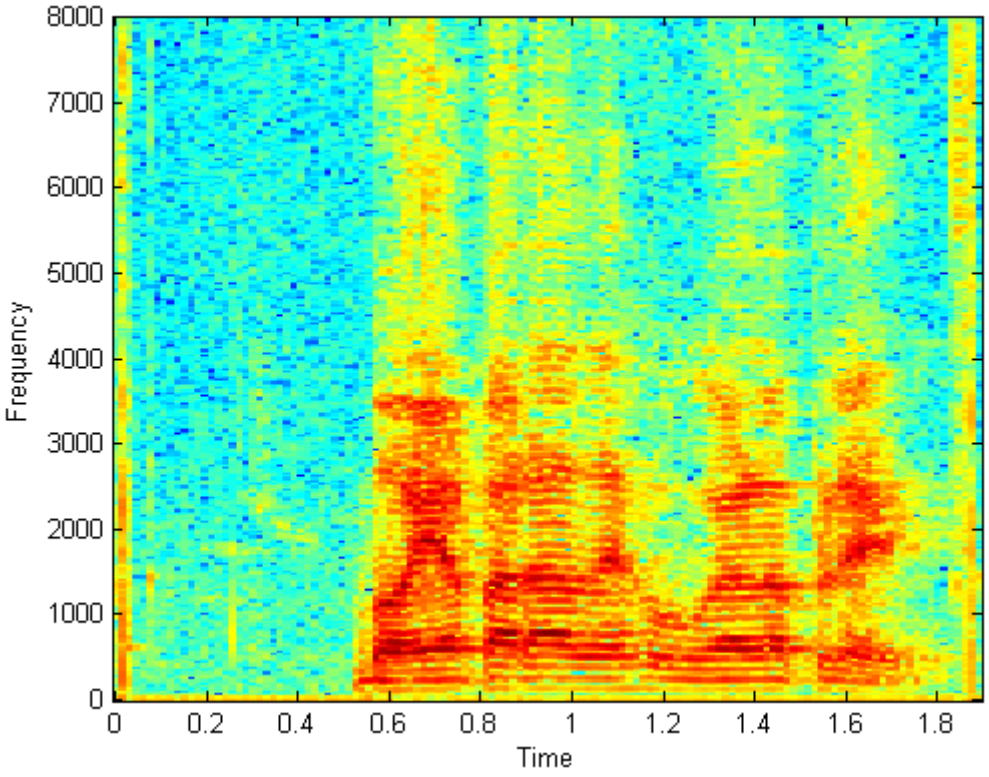


Figura 7.24 *Espectrograma de la primera señal fuente original.*

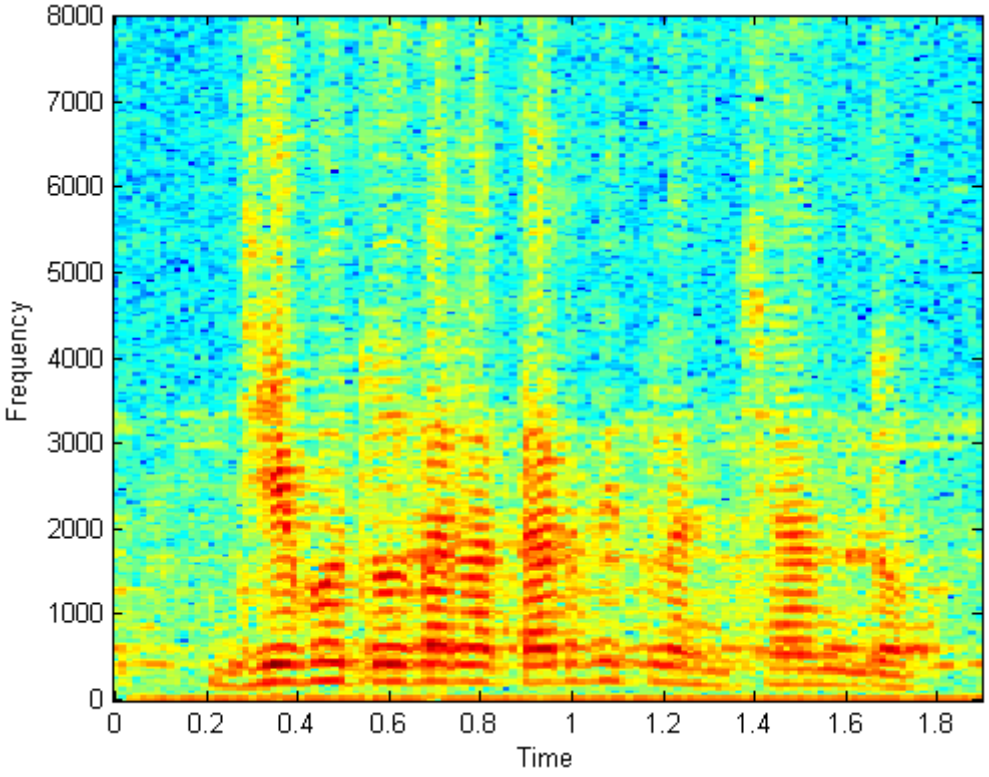


Figura 7.25 *Espectrograma de la segunda señal estimada.*

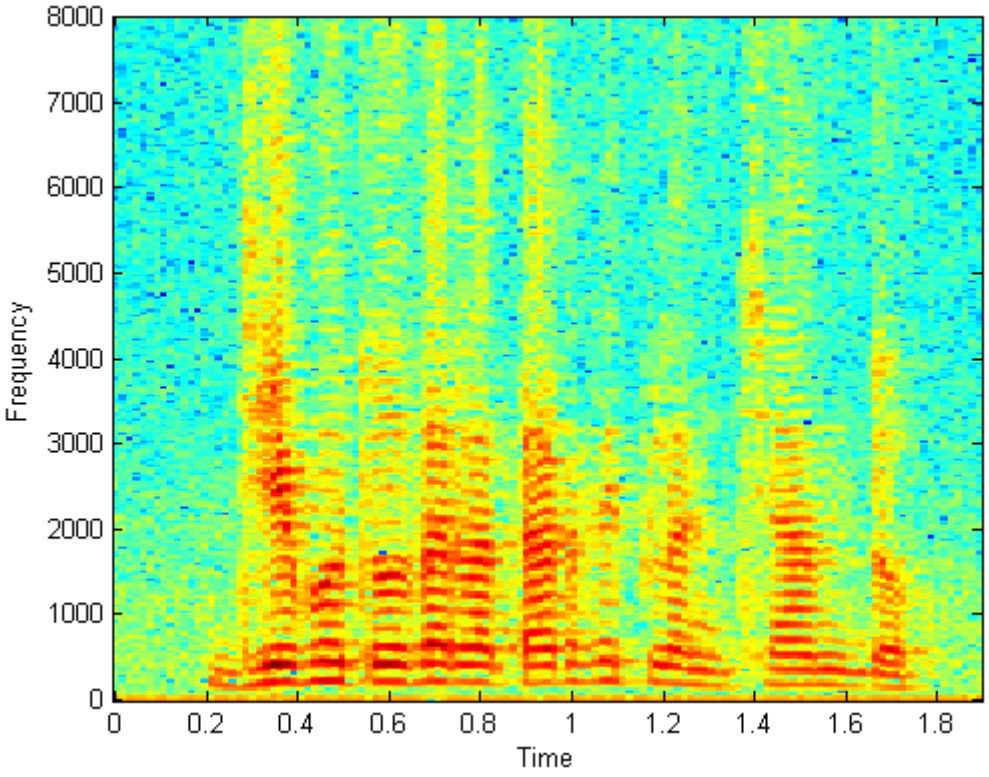


Figura 7.26 *Espectrograma de la segunda señal fuente original.*

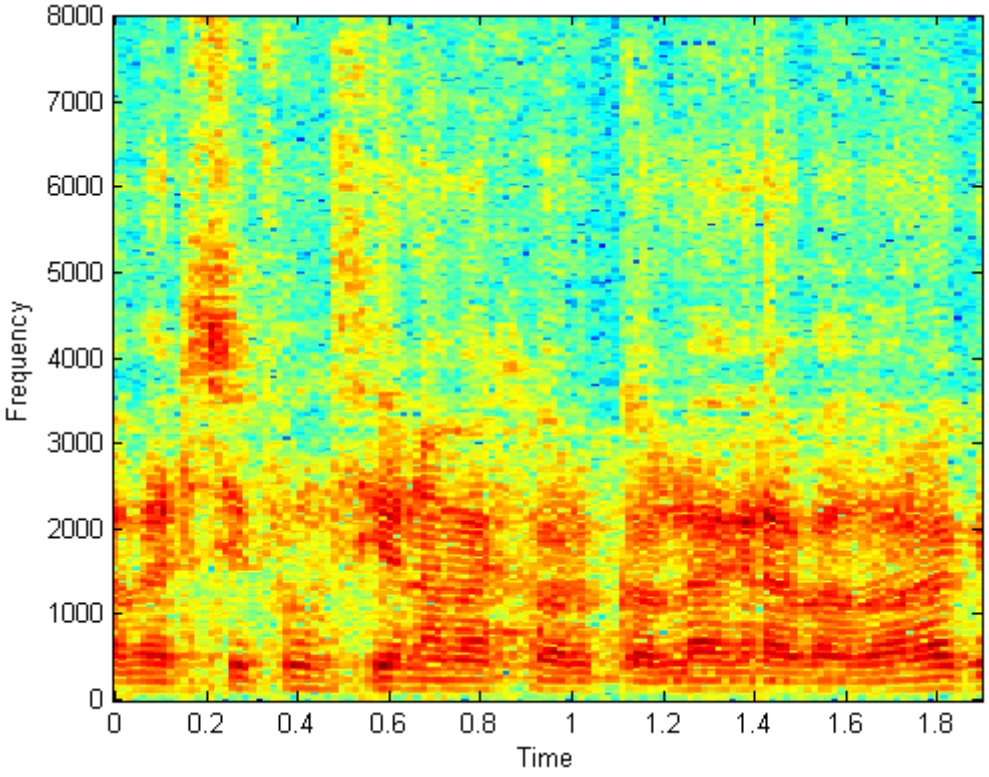


Figura 7.27 *Espectrograma de la tercera señal estimada.*

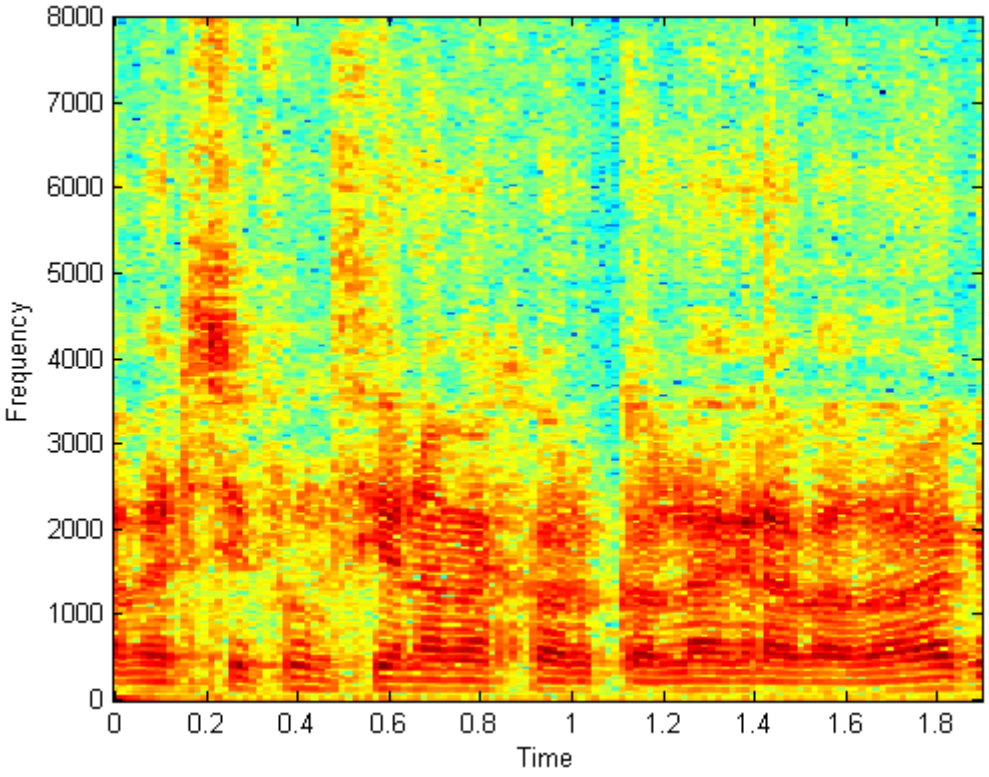


Figura 7.28 *Espectrograma de la tercera señal fuente original.*

Por último, y a modo de ejemplo, mostramos en la figura (7.29) la comparación entre las subbandas de frecuencia de las tres señales fuente con las obtenidas tras llevar a cabo ICA, alinear las permutaciones y ajustar el escalado, en la fila del espectrograma correspondiente a la frecuencia de 2156 Hz.

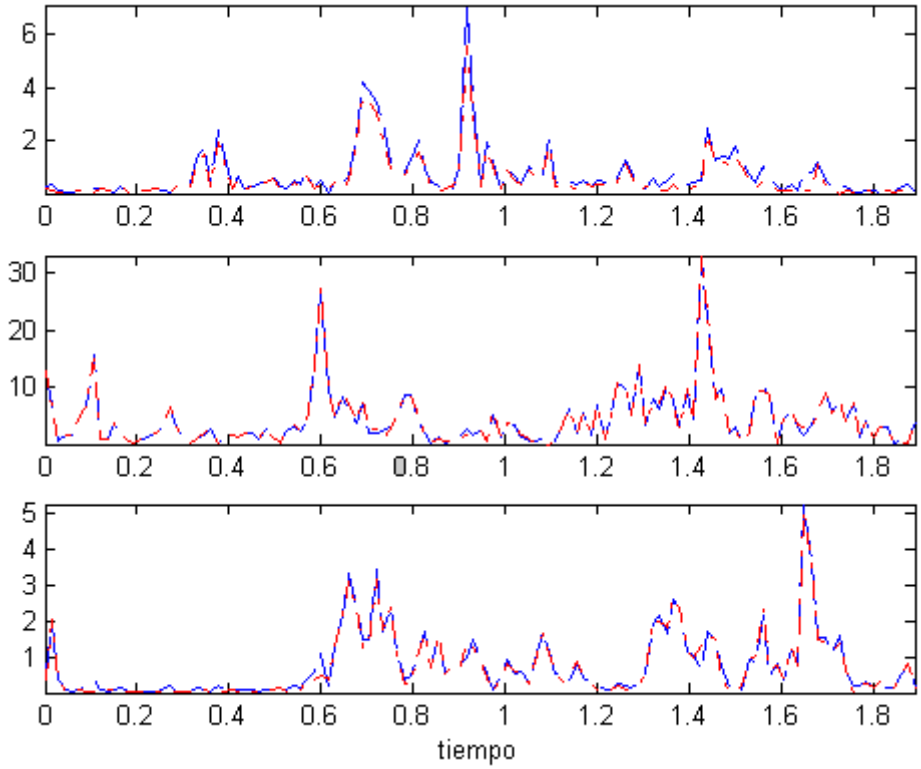


Figura 7.29 *Magnitud de las filas de los espectrogramas de las señales estimadas (azul), y de las señales fuente originales (rojo) correspondientes a una frecuencia de 2156 Hz.*

7.6 Casos no resueltos satisfactoriamente.

En este capítulo hemos presentado ya varios ejemplos de simulaciones que se han llevado a cabo con éxito. Bien, ahora vamos a hablar de las limitaciones que tienen los métodos de separación que hemos desarrollado, y que provocan que éstos fallen en determinadas situaciones.

7.6.1 Fallos usando masking.

El primer caso en el que hemos podido comprobar que este método de separación fracasa es cuando tenemos más de dos voces presentes en la mezcla. El motivo al que se deben los fallos es posiblemente la dificultad de encontrar señales de voz en la práctica que sean disjuntas en el dominio tiempo-frecuencia. Veíamos anteriormente que en el caso de dos señales fuente, aunque estas no fueran totalmente disjuntas, el algoritmo separaba las fuentes, y si bien la calidad de las señales de voz resultantes no era perfecta, éstas sí que eran perfectamente inteligibles. Sin embargo, cuando no son dos las señales que solapan en cada punto sino un número mayor de ellas, la estimación de los

ángulos de llegada de las fuentes se complica bastante puesto que en muchos puntos del espectrograma se suman las contribuciones de varios frentes de onda pertenecientes a varias fuentes, siendo errónea la extracción del retardo a partir de la fase.

Por otro lado, en los casos en que hay dos fuentes de voz presentes en las mezclas, la separación se lleva a cabo de forma exitosa para un gran rango de ángulos de llegada, siempre con valores razonables para la distancia entre los micrófonos. El algoritmo sólo falla cuando los frentes de onda llegan con direcciones muy similares, esto es, con una separación menor de 15º aproximadamente, aunque con distancias de separación entre los micrófonos de varios centímetros, sí que hemos conseguido separar señales cuyos ángulos de llegada diferían en muy pocos grados, por ejemplo 3º. La calidad de las señales resultante es aceptable, aunque si se cumpliera estrictamente la condición de que los espectrogramas son disjuntos sería mejor aún.

Cabe mencionar que a veces el algoritmo EM no funciona correctamente, ya que estima mal la media o la varianza de algunos de los retardos. Este caso es fácilmente identificable, ya que en el histograma de retardos similar la mostrado en la figura (7.4) podrían observarse los dos picos correspondientes a los valores correctos de los retardos pero alguna de las envolventes no se ajustarían bien a la forma del histograma. Cuando éste sea el problema, la solución es simplemente repetir la separación y ésta ya será ejecutada correctamente, ya que al ser EM un algoritmo estadístico, no opera de forma determinista aunque los datos sean exactamente los mismos, y a veces ofrecer resultados mejorables sin que eso significa que no pueda hallarse una estima mejor a partir de esos datos de entrada.

Por último reseñar que hemos probado este algoritmo para mezclas reales de voz sin obtener resultados satisfactorios. Esto puede deberse a que este modelo es válido para una situación anecoica y la simple presencia del suelo y otros obstáculos hace que el entorno en que se realiza la grabación no se ajuste a este modelo.

7.6.2 Fallos usando el método de Anemüller.

Este método funciona con mezclas digitales de voz que se realicen según el modelo matemático de mezcla anecoica. Por lo tanto no funcionará para grabaciones de mezclas reales de voz. Al igual que en el caso de separación por enmascaramiento, requiere que las señales de voz estén razonablemente separadas en el espacio, ya que si provienen de lugares muy parecidos la matriz de mezcla tendrá valores de retardos y atenuaciones muy parecidos para ambas grabaciones y será difícil llevar a cabo la separación. En el artículo de Jörn Anemüller que presenta este método de separación se indica desde un principio que está enfocado exclusivamente al caso anecoico, y por tanto no hay razones para pensar que deba funcionar en otros casos más complejos.

7.6.3 Fallos usando el método de subbandas.

Este es sin duda el método de separación más fiable que hemos desarrollado, podemos afirmar que funciona para cualquier mezcla digital anecoica sin

ninguna restricción. Hemos comprobado que tiene éxito incluso con diferencias entre los ángulos de llegada de las fuentes de menos de 1º. Además la calidad de las señales obtenidas es excelente y el único límite en cuanto al número de fuentes que se pueden incluir en las mezclas viene dado por el tiempo de cómputo que necesita para ejecutar el algoritmo, que obviamente aumenta cuantas más grabaciones tengamos.

Dicho esto diremos que el único punto flaco que presenta este método de separación es que no funciona para mezclas reales de voz. Este método está pensado para separar grabaciones realizadas en entornos convolutivos, es decir, en cualquier entorno real, y si la mezcla se realiza digitalmente entonces el algoritmo es totalmente efectivo, pero si la mezcla viene del 'mundo real' no se consigue la extracción de las fuentes.

7.7 Conclusiones.

En este capítulo dedicado a simulaciones hemos comenzado describiendo la situación que íbamos a simular en primer lugar, pasando después a aplicar a las mezclas obtenidas los algoritmos de separación basados en enmascaramiento y el método de Anemüller. Presentamos los resultados obtenidos dándolos por válidos, y recalcamos el importante resultado de que a pesar del hecho de que los espectrogramas no eran disjuntos la separación mediante masking separaba con éxito las señales fuente de la mezcla.

Posteriormente hemos expuesto la situación que simulamos para probar la bondad del algoritmo de separación ICA en cada subbanda de frecuencia de forma independiente. Para simular este caso tomábamos una mezcla de tres señales fuente grabadas con tres micrófonos, y veíamos que los resultados eran muy satisfactorios.

Por último hemos dedicado una sección a comentar los casos en que fallan los distintos algoritmos de separación, trantando de explicar por qué suceden dichos fallos y cómo pueden evitarse cara al futuro, y llegábamos a la conclusión de que el método más robusto de los tres es el último que hemos presentado, y por tanto es quizás el que tiene un mayor interés para seguir perfeccionándolo.

Capítulo 8

Conclusiones y líneas futuras de investigación

8.1 Conclusiones.

Durante la realización del proyecto hemos llevado a cabo un estudio teórico de varios temas enfocados a tener los conocimientos suficientes para afrontar la separación ciega de fuentes de señales de voz, que por supuesto hemos puesto en práctica. La parte teórica del proyecto tenía como objetivo principal llegar a ser capaces de usar del análisis de componentes independientes (ICA) para posteriormente aplicarlo a la separación de señales de voz. Hemos pretendido presentar aquí una visión general del modo en que se trabaja en ICA, planteando problemas y ofreciendo las soluciones que se han ido adoptando, exponiendo los criterios que se usan para lograr la separación y mostrando también los algoritmos fundamentales mediante los que se intenta llegar al objetivo. Posteriormente hemos visto cómo aplicar todos estos conceptos generales al caso concreto que nos ocupa, y para ello hemos tenido que estudiar un poco la señal de voz, presentando las características fundamentales de la misma que debemos conocer para trabajar con ella. Con este fin, el de tratar de la forma más adecuada la señal de voz hemos introducido la *transformada de Fourier dependiente del tiempo* (STFT), que ha sido la principal herramienta matemática que hemos usado para obtener una representación adecuada de los datos que nos permitiera llegar a nuestra meta.

Todos esos conocimientos los hemos aplicado en la presentación de los tres métodos que abarca nuestro estudio. Primero hemos hecho una descripción teórica de los mismos, explicando el modo en que se trabaja en cada uno de ellos y las suposiciones que a priori deben cumplirse para esperar su correcto funcionamiento. Tras la explicación teórica de los tres métodos, hemos contrastado en el capítulo de simulaciones que los algoritmos expuestos teóricamente pueden implementarse en un computador. Así, mediante varios ejemplos, hemos llegado a la conclusión de que el algoritmo que proporciona los mejores resultados en cuanto a la calidad de separación de las voces y a su adaptación a una variedad mayor de situaciones es el de separación independiente por subbandas. Recordamos que este método realizaba separaciones independientes mediante ICA en cada subbanda, usando luego el conocimiento de que las diferentes bandas de frecuencia de una misma señal de voz no son totalmente independientes entre sí para poner en común los resultados obtenidos por separado y hallar una estimación de las señales fuente. Para el caso de separación mediante enmascaramiento y Anemüller

concluíamos que sólo tenían éxito en el caso de mezclas anecoicas (puesto que para eso fueron diseñados) y necesitaban de cierta separación entre las direcciones de llegada de los frentes de onda correspondientes a las señales fuente. La calidad de la separación es bastante aceptable aunque sin llegar a la conseguida con la separación por subbandas. El asunto de todos los algoritmos presentados es conseguir que se adapten a mezclas reales de voz, y no sólo a mezclas digitales, siendo el que se muestra más cercano a conseguir este objetivo el último de los tres métodos presentados, ya que parte de que la mezcla ha sido realizada siguiendo un modelo convolutivo.

8.2 Líneas futuras de investigación.

- Una de las posibles líneas de investigación a desarrollar en el futuro puede ser hacer un estudio similar al realizado en este proyecto pero teniendo en cuenta la presencia de ruido, ya que la introducción de ruido en el modelo hace la situación más complicada y en las mezclas reales de voz es inevitable su aparición.
- Estudiar las limitaciones de estos algoritmos que hacen que no funciones con mezclas reales de voz, e implementar unos algoritmos que sí sean capaces de separar voces a partir de mezclas reales.
- Estudiar otros posibles modelos de mezcla de la voz que se adapten mejor a situaciones reales, principalmente teniendo en cuenta la reverberación, que hace que la respuesta en frecuencia del canal sea muy larga. Cuando la respuesta en frecuencia del canal dura más que la ventana que se usa, aparecen problemas sobre los que se podría investigar.
- Otro caso que aquí ha quedado sin una solución satisfactoria es el que se da cuando en la grabación están presentes más fuentes de voz que micrófonos. Podría ampliarse la información sobre el método de enmascaramiento para que no sea tan sensible a la condición de disjunción de los espectrogramas o investigar si es posible desarrollar otros métodos basados en ICA que contemplen esta posibilidad.
- Un campo interesante de estudio sería el de la viabilidad o no de incorporar un segundo micrófono en los teléfonos móviles para eliminar ruido de fondo, y comprobar cuánto mejoraría la calidad de la señal para ver si merece la pena su incorporación, así como el incremento que tendría en el coste económico y computacional.

Apéndices

Apéndice 1- Cumulantes.

Supongamos que x es una variable aleatoria escalar, continua, real y de media cero, y $p_x(x)$ su función densidad de probabilidad.

La primera *función característica* $\varphi(\omega)$ de x se define como la transformada continua de Fourier de la fdp $p_x(x)$:

$$\varphi(\omega) = E\{\exp(j\omega x)\} = \int_{-\infty}^{\infty} \exp(j\omega x)\, p_x(x)\, dx$$

donde $j = \sqrt{-1}$ y ω es la variable transformada correspondiente a x. Toda distribución de probabilidad viene especificada de forma única por su función característica, y viceversa. Expandiendo la función característica $\varphi(\omega)$ en series de Taylor sigue:

$$\varphi(\omega) = \int_{-\infty}^{\infty} \left(\sum_{k=0}^{\infty} \frac{x^k (j\omega)^k}{k!} \right) p_x(x)\, dx = \sum_{k=0}^{\infty} E\{x^k\} \frac{(j\omega)^k}{k!}$$

Así los coeficientes de esta expansión son momentos $E\{x^k\}$ de x (suponiendo que existan). Por este motivo, la función característica $\varphi(\omega)$ también se llama *función generadora de momentos*.

A menudo es desable usar la *segunda función característica* $\phi(\omega)$ de x, o *función generadora de cumulantes*. Esta función viene dada por el logaritmo natural de la primera función característica:

$$\phi(\omega) = \ln(\varphi(\omega)) = \ln(E\{\exp(j\omega x)\})$$

Los cumulantes κ_k de x se definen de una forma similar a sus respectivos momentos como los coeficientes de la expansión en serie de Taylor de la segunda función característica:

$$\phi(\omega) = \sum_{k=0}^{\infty} \kappa_k \frac{(j\omega)^k}{k!}$$

donde el k-ésimo cumulante se obtiene como la derivada

$$\kappa_k = (-j)^k \left. \frac{d^k \phi(\omega)}{d\omega^k} \right|_{\omega=0}$$

Para una variable aleatoria x de media cero, los cuatro primeros cumulantes son:

$$\kappa_1 = 0,\ \kappa_2 = E\{x^2\},\ \kappa_3 = E\{x^3\} \text{ y } \kappa_4 = E\{x^4\} - 3[E\{x^2\}]^2$$

De ahíque los tres primeros cumulantes son iguales a los respectivos momentos, y el cuarto cumulante κ_4 es igual a kurtosis.

Cuando la media de x no es cero, los cuatro primeros cumulantes toman las siguientes expresiones:

$$\kappa_1 = E\{x\}$$
$$\kappa_2 = E\{x^2\} - [E\{x\}]^2$$
$$\kappa_3 = E\{x^3\} - 3E\{x^2\}E\{x\} + 2[E\{x\}]^3$$
$$\kappa_4 = E\{x^4\} - 3[E\{x^2\}]^2 - 4E\{x^3\}E\{x\} + 12E\{x^2\}[E\{x\}]^2 - 6[E\{x\}]^4$$

Las complejidad de las expresiones para los cumulantes de orden superior irá incrementándose.

En el caso en que **x** sea un vector aleatorio y $p_{\mathbf{x}}(\mathbf{x})$ su función densidad de probabilidad, la función característica de **x** es nuevamente:

$$\varphi(\omega) = E\{\exp(j\omega\,\mathbf{x})\} = \int_{-\infty}^{\infty} \exp(j\omega\,\mathbf{x}) p_{\mathbf{x}}(\mathbf{x}) d\mathbf{x}$$

donde ahora ω es un vector fila de la misma dimensión de **x**, y la integral se calcula sobre todos los componentes de **x**. Los momentos y los cumulantes de **x** se obtienen de forma similar al caso escalar. En el caso vectorial, loas cumulantes son llamamados a menudo cumulantes cruzados (cross-cumulants) en analogía con las covarianzas cruzadas (cross-covariances). Puede demostrarse que el segundo, tercer, y cuarto cumulante para un vector aleatorio **x** de media cero son:

$$\mathrm{Cum}(x_i, x_j) = E\{x_i x_j\}$$
$$\mathrm{Cum}(x_i, x_j, x_k) = E\{x_i x_j x_k\}$$
$$\mathrm{Cum}(x_i, x_j, x_k x_l) = E\{x_i x_j x_k x_l\} - E\{x_i x_j\}E\{x_k x_l\} - E\{x_i x_k\}E\{x_j x_l\} - E\{x_i x_l\}E\{x_j x_k\}$$

De esas expresiones podemos deducir que el segundo cumulante es igual al segundo momento $E\{x_i x_j\}$, lo que se convierte en la correlación o la covarianza entre las variables x_i y x_j. El tercer cumulante también es igual al tercer momento. Ambos momentos y cumulantes poseen la misma información estadística, porque los cumulantes pueden expresarse en términos de sumas de

productos de los momentos. Normalmente es preferible trabajar con cumulantes porque presentan de una forma más clara la información adicional proporcionada por los estadísticos de orden superior.

Para hallar el cumulante de cualquier orden en función de los momentos se usa la siguiente fórmula:

$$\mathrm{Cum}(x_1,\ldots,x_n) = \sum_{(p_1,\ldots,p_m)} (-1)^{m-1}(m-1)!\,E\left[\prod_{i\in p_1} y_i\right]E\left[\prod_{i\in p_2} y_i\right]\cdots\left[\prod_{i\in p_m} y_i\right]$$

donde la suma se extiende a todas las posibles particiones $(p_1,\ldots,p_m)$, $m = 1,\ldots,n$ del conjunto de números naturales $(1,\ldots n)$.

A continuación presentamos las expresiones correspondientes a las matrices de cumulantes $\mathbf{C}_{y,y}^{1,\beta}$ para señales *y* reales y de media cero. Por simplicidad en la escritura de las expresiones, denotaremos el momento de orden α de *y* como $\mathbf{M}_y^{\alpha} = E\left[\mathbf{y}^{\bullet\alpha}\right]$ y las matrices de momentos cruzados como $\mathbf{M}_{y,y}^{1,\beta} = E\left[\mathbf{y}\left(\mathbf{y}^{\bullet\beta}\right)^{\mathrm{T}}\right]$. Vamos a mostrar dichas matrices de cumulantes cruzados para valores de β comprendidos entre 1 y 5.

$$\mathbf{C}_{y,y}^{1,1} = \mathbf{M}_{y,y}^{1,1} = E\left[\mathbf{y}\mathbf{y}^{\mathrm{T}}\right]$$
$$\mathbf{C}_{y,y}^{1,2} = \mathbf{M}_{y,y}^{1,2} = E\left[\mathbf{y}\left(\mathbf{y}^{\bullet 2}\right)^{\mathrm{T}}\right]$$
$$\mathbf{C}_{y,y}^{1,3} = \mathbf{M}_{y,y}^{1,3} - 3\mathbf{M}_{y,y}^{1,1}\,diag\left(\mathbf{M}_y^2\right)$$
$$\mathbf{C}_{y,y}^{1,4} = \mathbf{M}_{y,y}^{1,4} - 4\mathbf{M}_{y,y}^{1,1}\,diag\left(\mathbf{M}_y^3\right) - 6\mathbf{M}_{y,y}^{1,2}\,diag\left(\mathbf{M}_y^2\right)$$
$$\mathbf{C}_{y,y}^{1,5} = \mathbf{M}_{y,y}^{1,5} - 5\mathbf{M}_{y,y}^{1,1}\,diag\left(\mathbf{M}_y^4\right) - 10\mathbf{M}_{y,y}^{1,2}\,diag\left(\mathbf{M}_y^3\right) - 10\mathbf{M}_{y,y}^{1,3}\,diag\left(\mathbf{M}_y^2\right) + 30\mathbf{M}_{y,y}^{1,1}\,diag\left(\mathbf{M}_y^2\right)^2$$

En el caso de que las señales sean complejas las expresiones se complican bastante. Por ejemplo, para $\beta = 3$, la matriz de cumulantes sería:

$$\mathbf{C}_{y,y}^{1,3} = E\left[\mathbf{y}\left(\mathbf{y}\bullet\mathbf{y}^{*}\bullet\mathbf{y}^{*}\right)^{\mathrm{T}}\right] - E\left[\mathbf{y}\mathbf{y}^{\mathrm{T}}\right]diag\left(E\left[\mathbf{y}^{*}\bullet\mathbf{y}^{*}\right]\right) - 2E\left[\mathbf{y}\mathbf{y}^{\mathrm{H}}\right]diag\left(E\left[\mathbf{y}\bullet\mathbf{y}^{*}\right]\right)$$

Apéndice 2- Método del gradiente.

El método del gradiente es un algoritmo de optimización. Describiremos aquí el método de ascenso del gradiente, aunque en esencia es igual al método de descenso del gradiente. Este algoritmo nos servirá para encontrar el máximo local de una función, y para ello iremos iterando dando pasos en la dirección proporcional al gradiente. Si quisiéramos buscar el mínimo local de esa función, deberíamos aplicar el algoritmo de gradiente descendente.

Consideremos una función de valor escalar de *m* variables:

$$F = F(x_1, \ldots, x_m) = F(\mathbf{x})$$

Suponiendo que dicha función F es diferenciable, su vector gradiente con respecto a **x** es el vector *m*-dimensional que contiene las derivadas parciales:

$$\frac{\partial F}{\partial \mathbf{x}} = \begin{bmatrix} \frac{\partial F}{\partial x_1} \\ \vdots \\ \frac{\partial F}{\partial x_m} \end{bmatrix}$$

Usaremos la notación ∇F para designar el gradiente. El ascenso del gradiente está basado en la observación de que, si la función $F(\mathbf{x})$ está definida y es diferenciable en las cercanías de un punto **a**, entonces $F(\mathbf{x})$ se incrementa más rápidamente si uno se mueve desde **a** en la dirección del gradiente de F en **a**, $\nabla F(\mathbf{a})$. Se sigue que si

$$\mathbf{b} = \mathbf{a} + \gamma \nabla F(\mathbf{a})$$

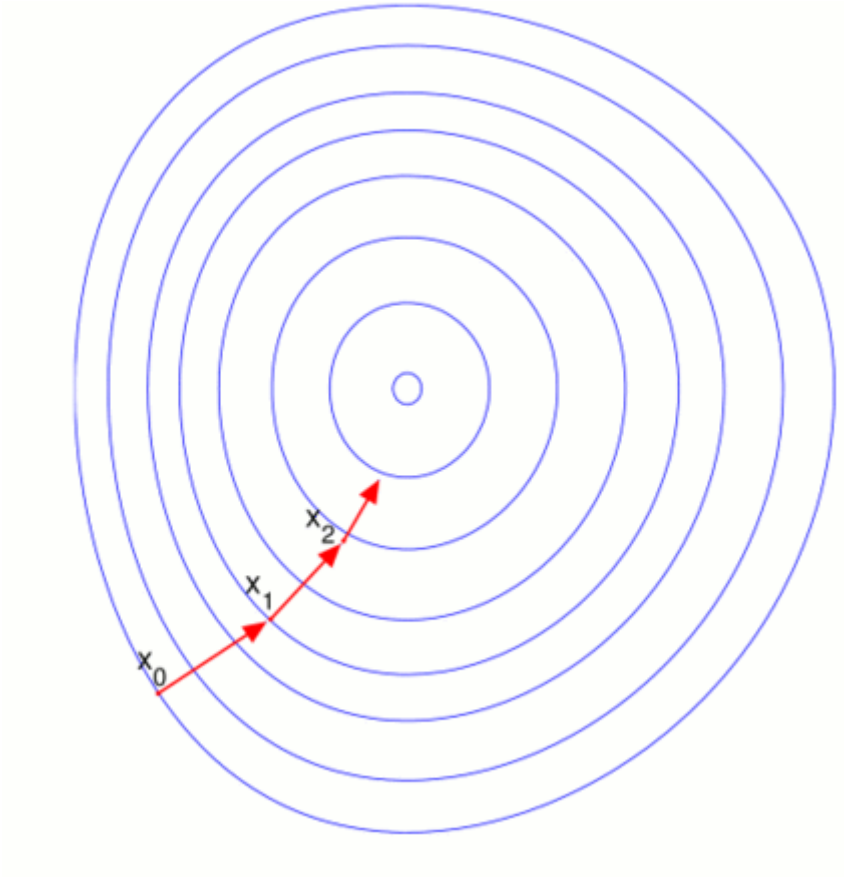
siendo $\gamma > 0$ un número pequeño, entonces $F(\mathbf{a}) \leq F(\mathbf{b})$. Teniendo esto en mente, se comienza por una estimación inicial $\mathbf{x}_0$ para un máximo local de F, y consideramos la secuencia $\mathbf{x}_0, \mathbf{x}_1, \mathbf{x}_2, \ldots$ tal que

$$\mathbf{x}_{n+1} = \mathbf{x}_n + \gamma_n \nabla F(\mathbf{x}_n), \; n \geq 0$$

De esta forma tenemos que se cumple $F(\mathbf{x}_0) \leq F(\mathbf{x}_1) \leq F(\mathbf{x}_2) \leq \cdots$, y la secuencia $(\mathbf{x}_n)$ debe converger al máximo local. El valor del paso de adaptación γ puede variar en cada iteración.

Ilustramos este procedimiento con la figura que hay debajo. En este caso F está definida en dos dimensiones, y la gráfica se asemeja a una colina. Las curvas azules se corresponden con las regiones en que F es constante, y las flechas rojas indican la dirección del gradiente en ese punto. Observar que el gradiente es siempre el punto perpendicular a las líneas azules. Vemos que el ascenso del

gradiente nos lleva a la cima de esa colina, esto es, el punto donde el valor de la función *F* es máximo.



Apéndice 3- Guía para el usuario del programa realizado en Matlab.

Antes de empezar, sólo decir que este programa ha sido realizado en Matlab 6.5, y por lo tanto es posible que no funcione en otras versiones de Matlab.

Una vez dicho esto, arrancamos el Matlab 6.5, seleccionamos el directorio en que se encuentren los ficheros fuente como directorio de trabajo y tecleamos en la ventana de comandos de Matlab:

>> programa_separa

Carga de datos de entrada.

Se abrirá entonces la siguiente ventana:

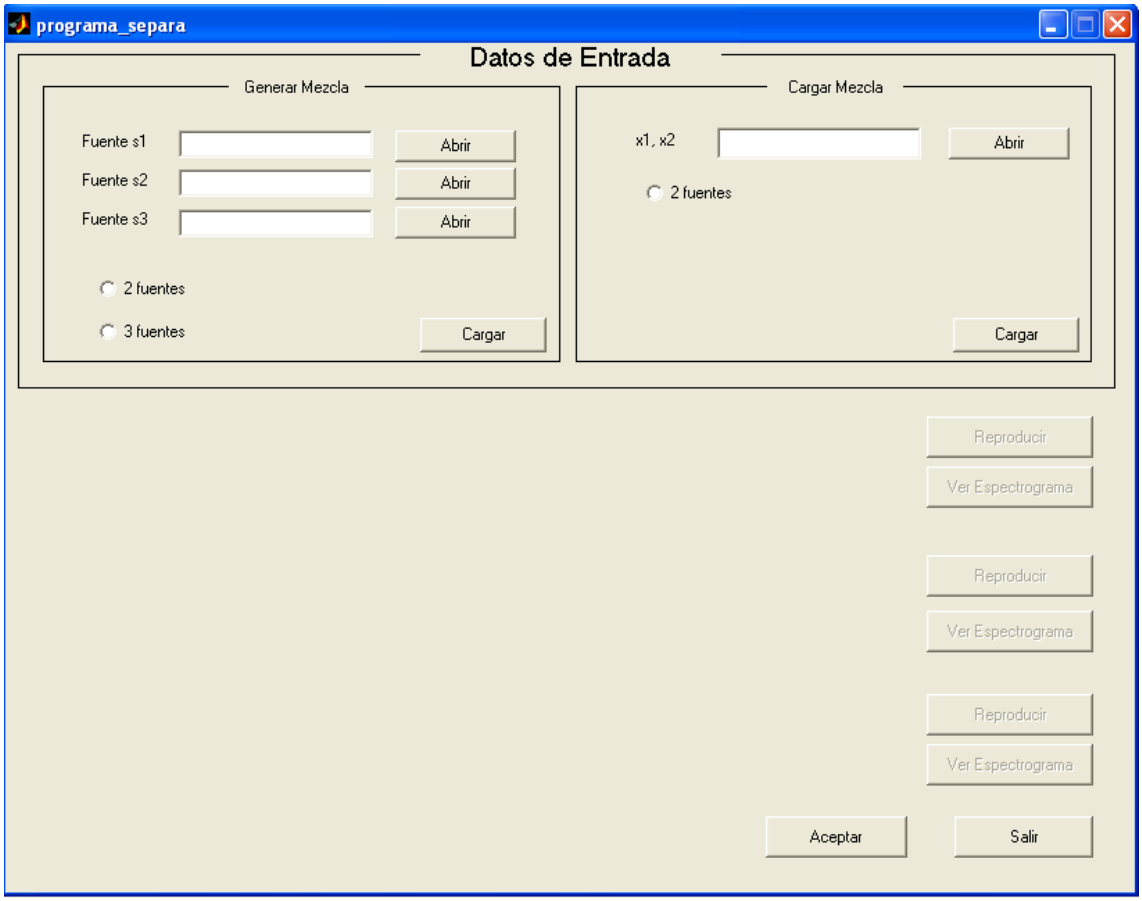


Figura A1

Vemos que hay varios botones que se pueden pulsar. El botón “Salir” finalizará el programa. De los demás hablaremos a continuación.

Esta ventana es para seleccionar los archivos de voz que usaremos en la simulación. Tenemos dos posibilidades, la de seleccionar las señales fuente que posteriormente mezclaremos para generar las señales observadas, o la de cargar una mezcla de voces ya generada o grabada con anterioridad.

En la parte superior izquierda de la ventana mostrada en la figura A1 tenemos el cuadro correspondiente a la primera posibilidad, es decir, queremos generar una mezcla digital de señales de voz para después simular su separación. Vemos que tenemos tres etiquetas que muestran las tres posibles señales fuente, con tres botones de "Abrir" a sus respectivas derechas. Si pulsamos en estos botones se nos abre un cuadro donde podremos elegir los ficheros fuente en la jerarquía de directorios. Estos archivos tendrán que ser grabaciones con la extensión ".wav" para poder abrirlos. Si pulsamos el botón "Abrir" el cuadro que nos aparece será algo como lo siguiente:

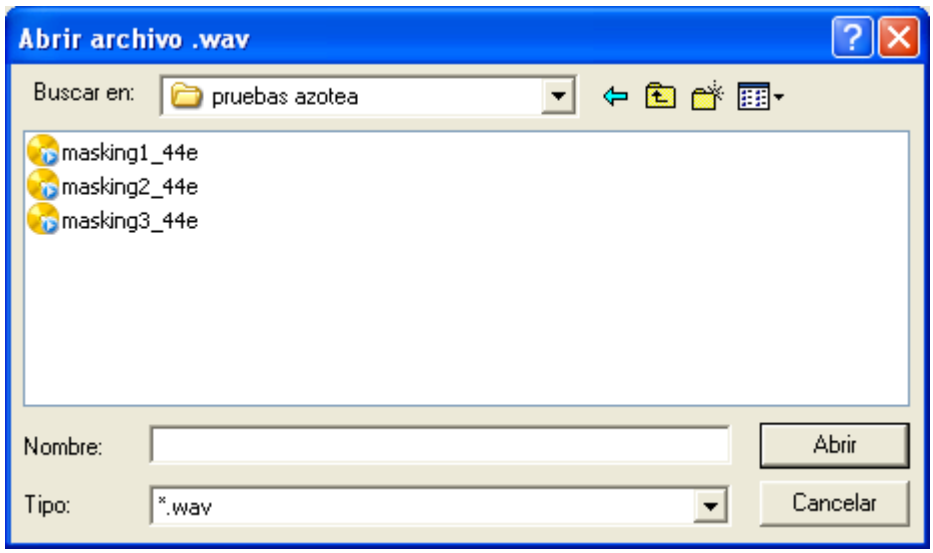


Figura A2

Podemos seleccionar de esta forma dos o tres fuentes, marcando previamente la opción elegida o simplemente dejar que el programa lo haga automáticamente. Una vez hecho esto, veremos cómo si pulsamos el botón "Cargar" las señales de voz elegidas se cargan en la parte inferior de la ventana, quedando la apariencia de la misma como se muestra en la figura A3.

Si picamos en el botón reproducir junto a la forma de onda de cada señal escucharemos la correspondiente grabación de voz. Así mismo podemos ver su espectrograma pulsando el botón correspondiente, que provocará que se abra una ventana como la mostrada en la figura A4.

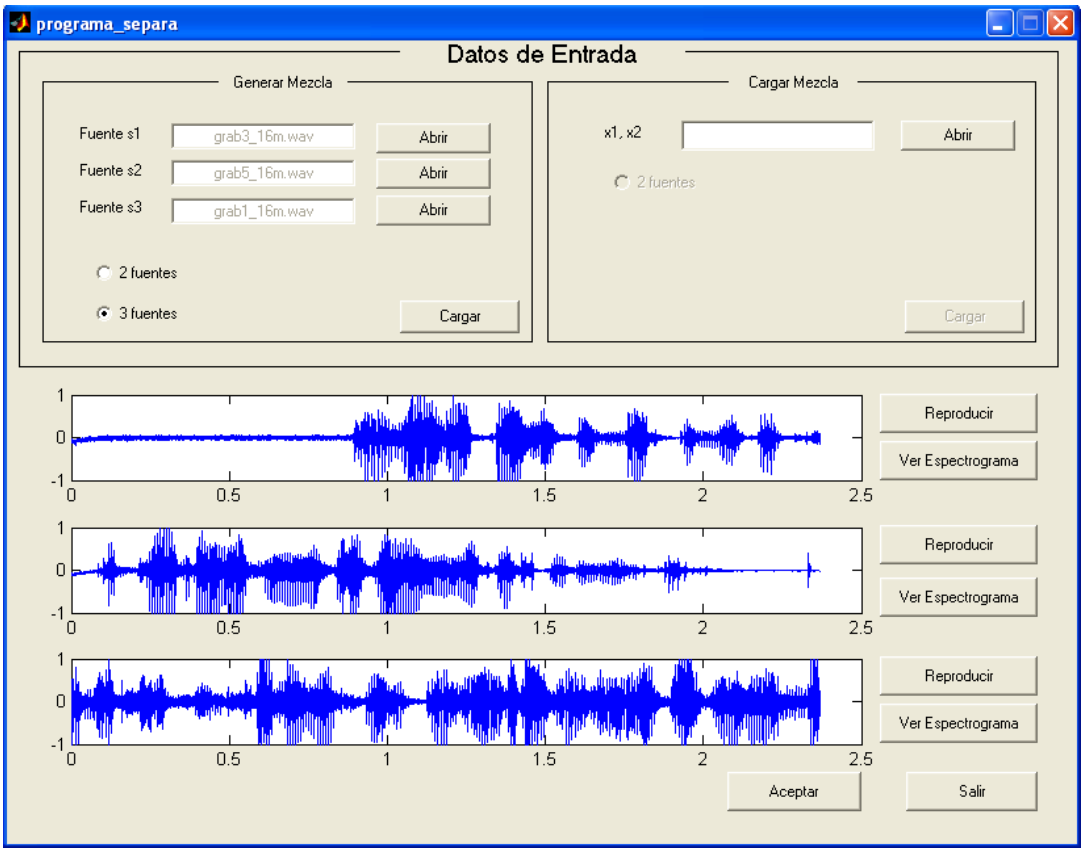


Figura A3

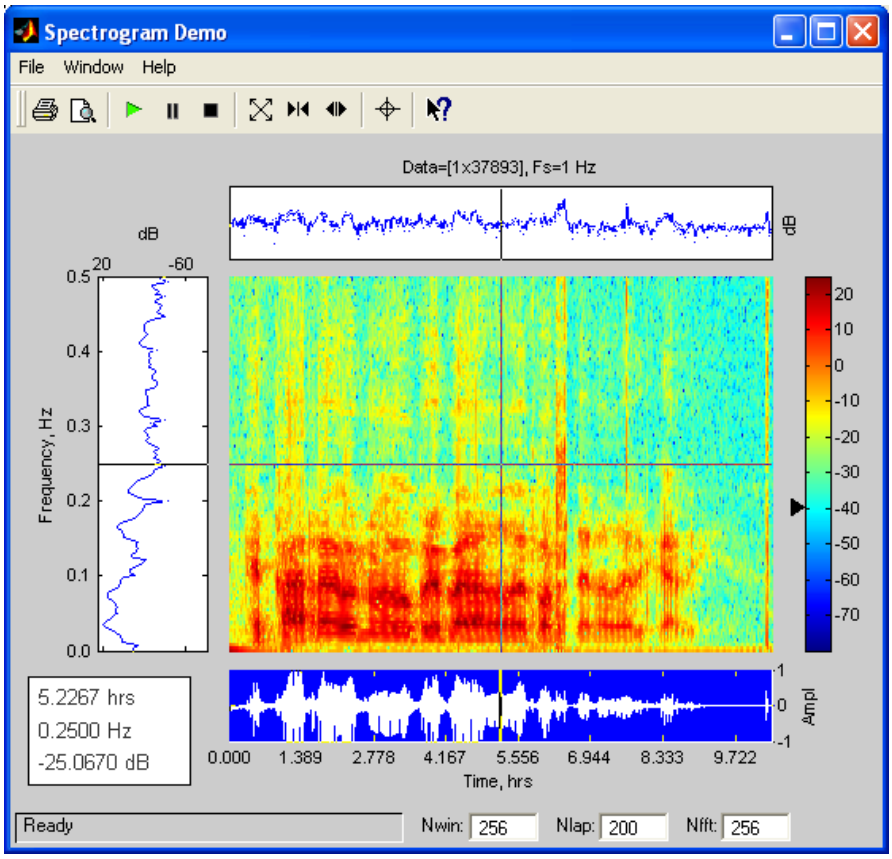


Figura A4

En la representación del espectrograma podemos mover dos cursores, uno horizontal y otro vertical. El horizontal se corresponde con las distintas bandas de frecuencia, ordenadas en orden creciente de abajo a arriba. Si movemos este cursor vemos cómo cambia la gráfica situada encima de la del espectrograma. Ésta se corresponde a la evolución temporal de la señal en la frecuencia seleccionada por el cursor.

Por otro lado, el cursor vertical se corresponde con el tiempo, y podemos observar el espectro correspondiente a ese instante de tiempo en la gráfica vertical situada a la izquierda del espectrograma.

Podemos cambiar los parámetros con que se realiza el espectrograma a nuestro gusto mediante los cuadros de texto situados en el pie de la ventana. El parámetro Nwin se corresponde con la longitud de la ventana, Nlap es el número de muestras en que solapan las ventanas y Nfft es el número de puntos de la transformada FFT que se aplica a cada segmento enventanado de la señal para calcular el espectrograma. Cambiando estos valores estamos cambiando la resolución del espectrograma.

En el cuadro que se encuentra en la zona inferior izquierda de la figura podemos ver en cada momento los valores tanto de la frecuencia y el tiempo en que estamos situados como la amplitud del espectrograma en ese punto (en decibelios). Por último, en la paleta de colores de la derecha se indica con una flecha la escala usada. También destacar que en la barra de herramientas se pueden realizar acciones como zoom y otras utilidades.

Volviendo a la ventana de generación de datos de entrada, la otra opción que se puede escoger es cargar una mezcla ya grabada con una grabadora de sonidos estereofónica (en formato wav) o cargar un documento ".mat" que contenga una mezcla de señales de voz ya hecha. En ese caso el fichero debe contener al menos dos variables, una matriz de dos filas con nombre *s* que contenga la grabación de un micrófono en cada fila y una variable llamada *fs* que represente la frecuencia de muestreo de las señales de entrada. Estos datos se introducirán de la misma forma que vimos antes pero en el cuadro superior derecho de la ventana.

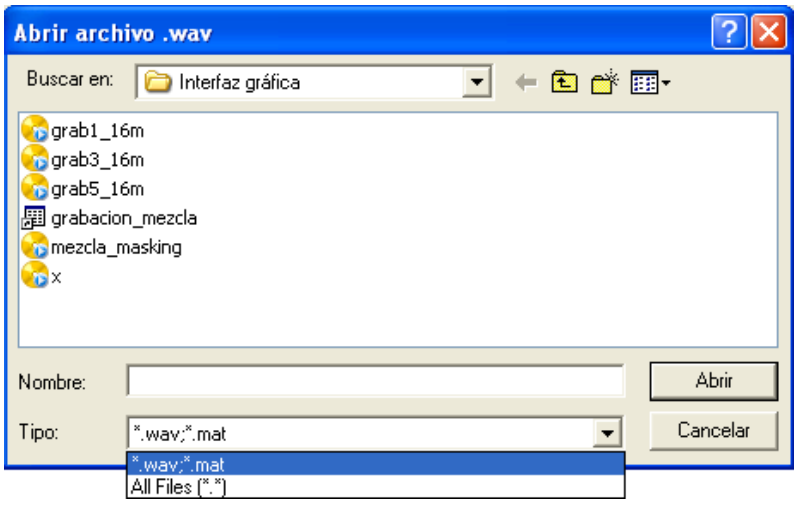


Figura A5

Elección de los parámetros de la simulación.

Bien, vamos a centrarnos en lo que sucede si nos decidimos por la opción de generar mezcla. Una vez cargados los datos pulsamos en el botón “Aceptar”. Entonces debe aparecer la siguiente ventana:

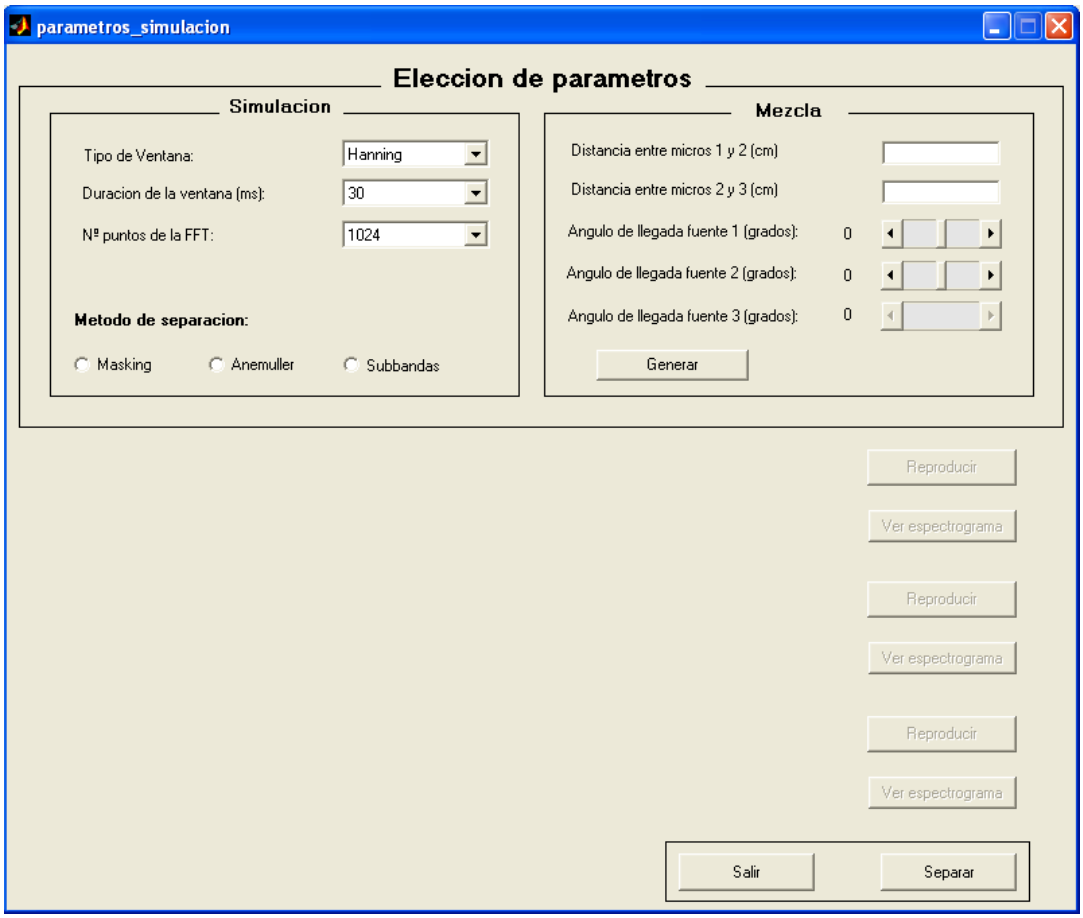


Figura A6

En esta ventana elegiremos los parámetros de la simulación. Vemos que en la parte superior izquierda de la ventana hay tres listas desplegables donde escogemos datos relacionados con la transformada STFT que se aplicará a las señales de voz. Debemos elegir el tipo de ventana, la duración de la misma y el número de puntos de la FFT que se aplicará a los segmentos de voz enventanados para calcular las columnas del espectrograma.

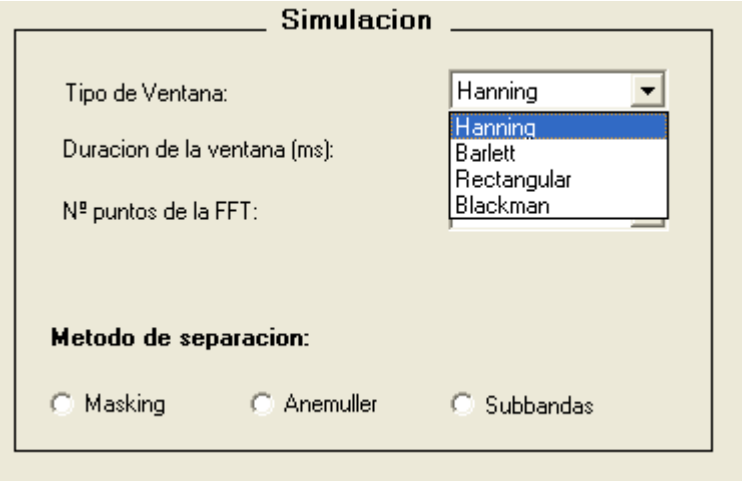


Figura A7

También debemos seleccionar uno de los tres métodos de separación, que será el que se use posteriormente para realizar la misma. Las tres posibilidades son las explicadas en la sección 5 de este documento, es decir, enmascaramiento o masking, estimación de los parámetros de la matriz de separación, que se corresponde con Anemuller, y el de separación mediante ICA en cada subbanda de frecuencia por separado.

Una vez hecho esto nos dispondremos a introducir los datos correspondientes a la descripción del entorno en que queremos que se realice la mezcla, esto lo haremos en el cuadro de la parte superior derecha de la ventana. La situación se corresponde con la siguiente: habrá dos o tres micrófonos (según el número de fuentes y el método de separación) situados en línea. La separación entre los micrófonos se introduce mediante el teclado en los campos de texto correspondientes. Si sólo hay dos micrófonos no hay que introducir el dato de la separación entre los micros 2 y 3, y si se hace este dato será ignorado. Los ángulos de llegada de las fuentes se corresponden a la situación de las personas que hablan, es decir, las que emiten las fuentes. Estos ángulos se miden con respecto a la perpendicular a la línea imaginaria que une los micrófonos. Podemos observar esto en la figura (A8).

Una vez introducidos todos los datos hacemos click en el botón “Generar” y se nos generará la grabación que habría sido realizada por los micrófonos en esa situación.

Nuevamente podemos escuchar las grabaciones pulsando el botón “Reproducir”, así como visualizar el espectrograma de las mezclas.

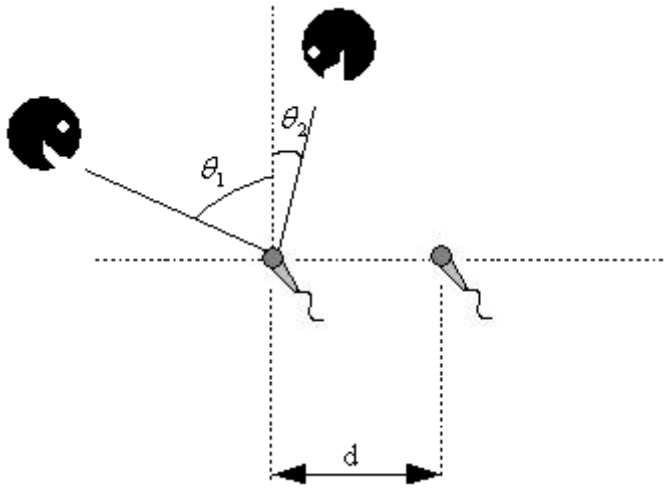


Figura A8

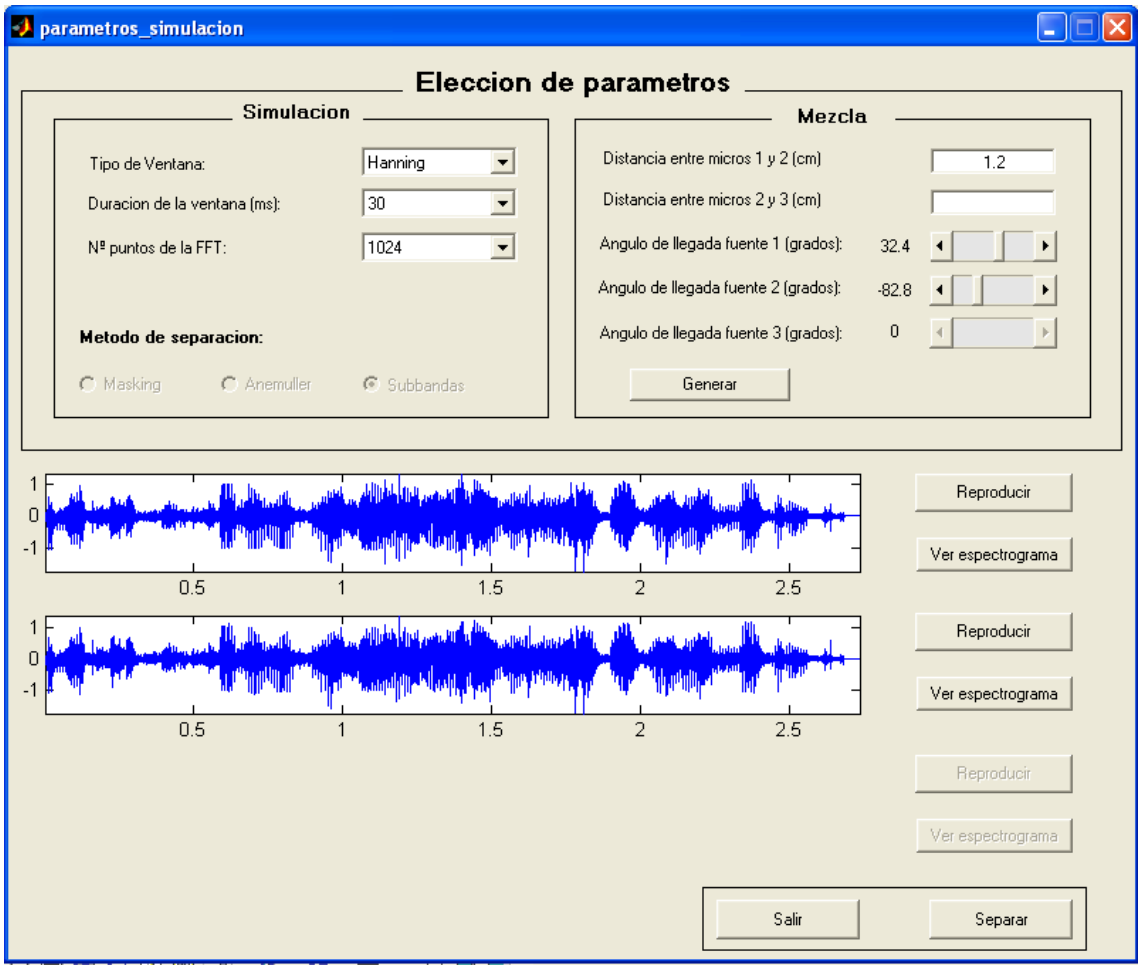


Figura A9

Ahora sólo nos queda comprobar si la separación se ejecuta con éxito en esas condiciones que hemos introducido. Los datos que se visualizarán serán diferentes según el método de separación que hayamos escogido. Si pulsamos el botón “Separar”, veremos aparecer la siguiente ventana:

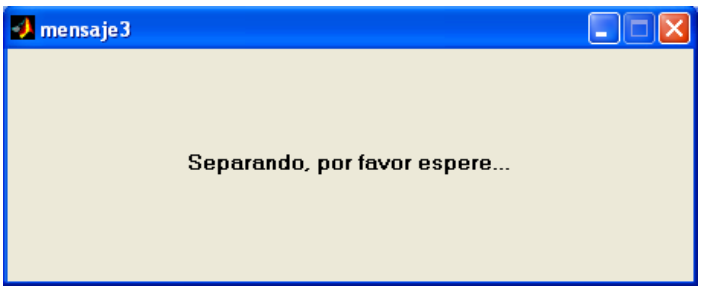


Figura A10

Tras lo cual se abrirá la ventana de presentación de resultados que será diferente para cada método de separación elegido.

Presentación de resultados.

Vamos a distinguir entre los tres métodos de separación:

Subbandas.

Se mostrará una ventana como la mostrada en la figura A11.

En ella podemos escuchar las señales obtenidas de la separación y ver su espectrograma, como viene siendo habitual. Pero también aparece una novedad, ya que, como sabemos, este método de separación se basa en llevar a cabo una separación en cada canal de frecuencia de forma independiente, podemos visualizar cómo es el canal de frecuencia de cualquier señal separada y compararlo con la señal original, es decir, antes de realizar la mezcla. Esto se hace eligiendo una señal en la lista desplegable e introduciendo el número de la subbanda que deseamos visualizar. A continuación pulsaremos el botón “Representar” y obtendremos algo como lo que vemos en la figura A12.

Vemos en la figura A12 que se pueden comparar tanto el módulo como la fase de la evolución temporal de la señal en la frecuencia escogida. En rojo se muestra la grabación original y en azul la obtenida en la separación.

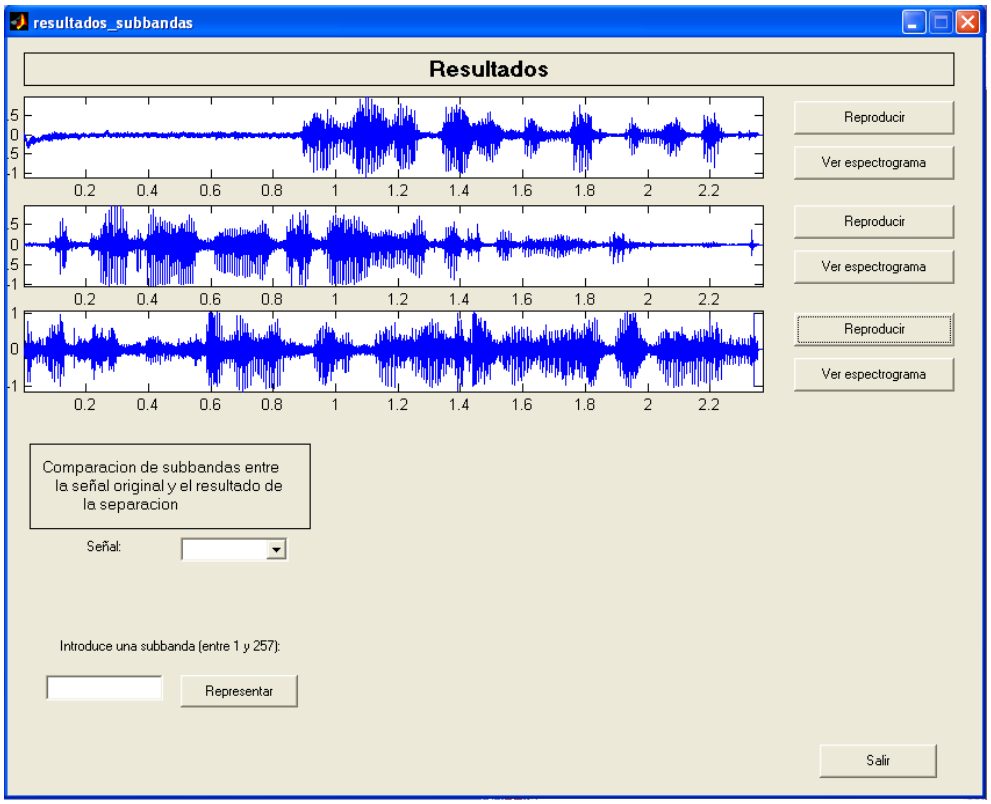


Figura A11

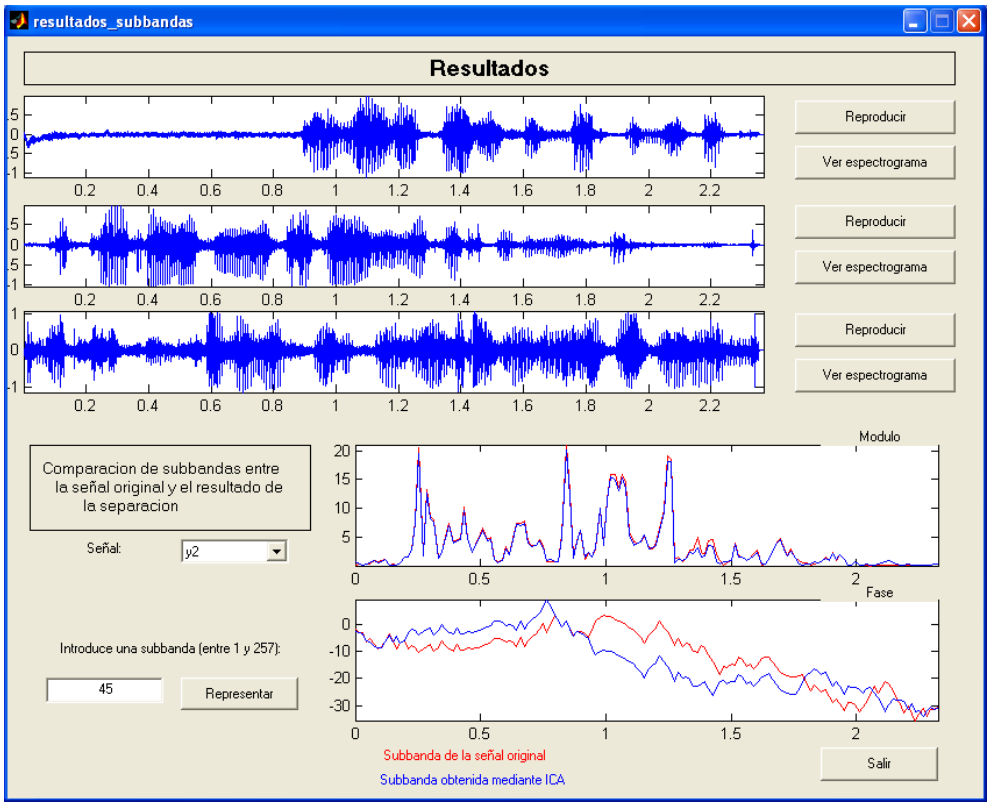


Figura A12

Masking.

La ventana de presentación de resultados tiene el aspecto que se muestra a continuación. En ella podemos observar la expresión temporal de las señales de voz resultantes de la separación, que podremos escuchar pulsando el botón correspondiente, así como visualizar su espectrograma. Debajo de estas gráficas se pueden ver las máscaras correspondientes a cada señal separada. En ellas se representan los puntos del espectrograma de la mezcla grabada por el micrófono de referencia que se ha estimado que corresponden a esa fuente. Por último, en la parte inferior de la pantalla se muestran las gráficas correspondientes a los retardos encontrados y a los ángulos de llegada estimados. En la de la izquierda se ve el histograma de retardos obtenidos a partir de los espectrogramas de las grabaciones, donde si todo ha ido bien deben observarse picos acusados correspondientes a las fuentes presentes en la mezcla. En la derecha se distinguen los ángulos con los que el programa ha estimado que llegaron los frentes de onda de las fuentes.

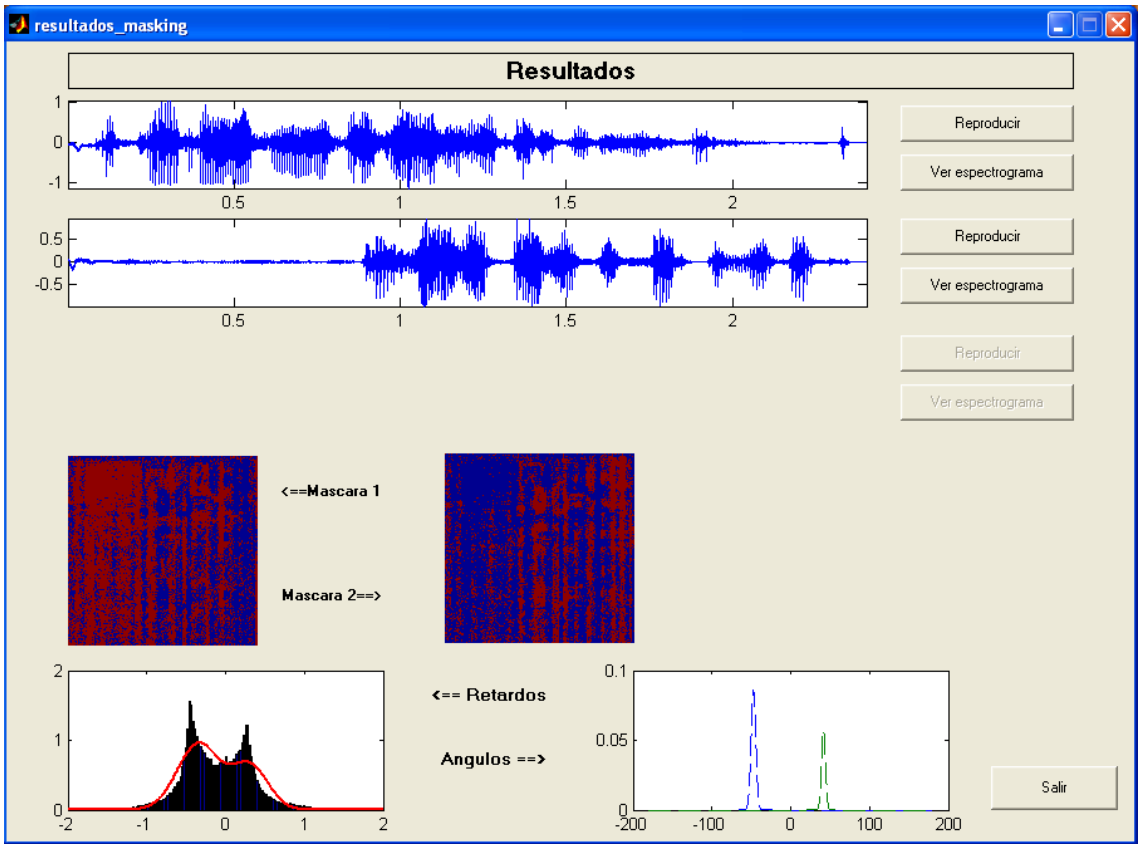


Figura A13

Anemuller.

En la ventana de la figura A14 se muestran las señales obtenidas de la separación. Puede comprobarse si el algoritmo ha tenido éxito escuchando su reproducción.

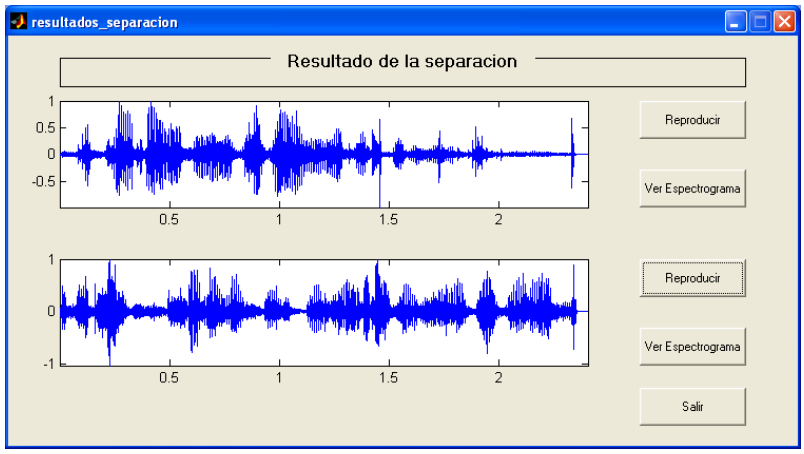


Figura A14

Carga de una mezcla.

Vimos que en la primera ventana que aparecía al ejecutar el programa teníamos la posibilidad de generar una mezcla a partir de las grabaciones independientes de las señales fuente o de cargar una mezcla ya realizada para proseguir con su separación. Vamos ahora a ver lo que sucede en este último caso.

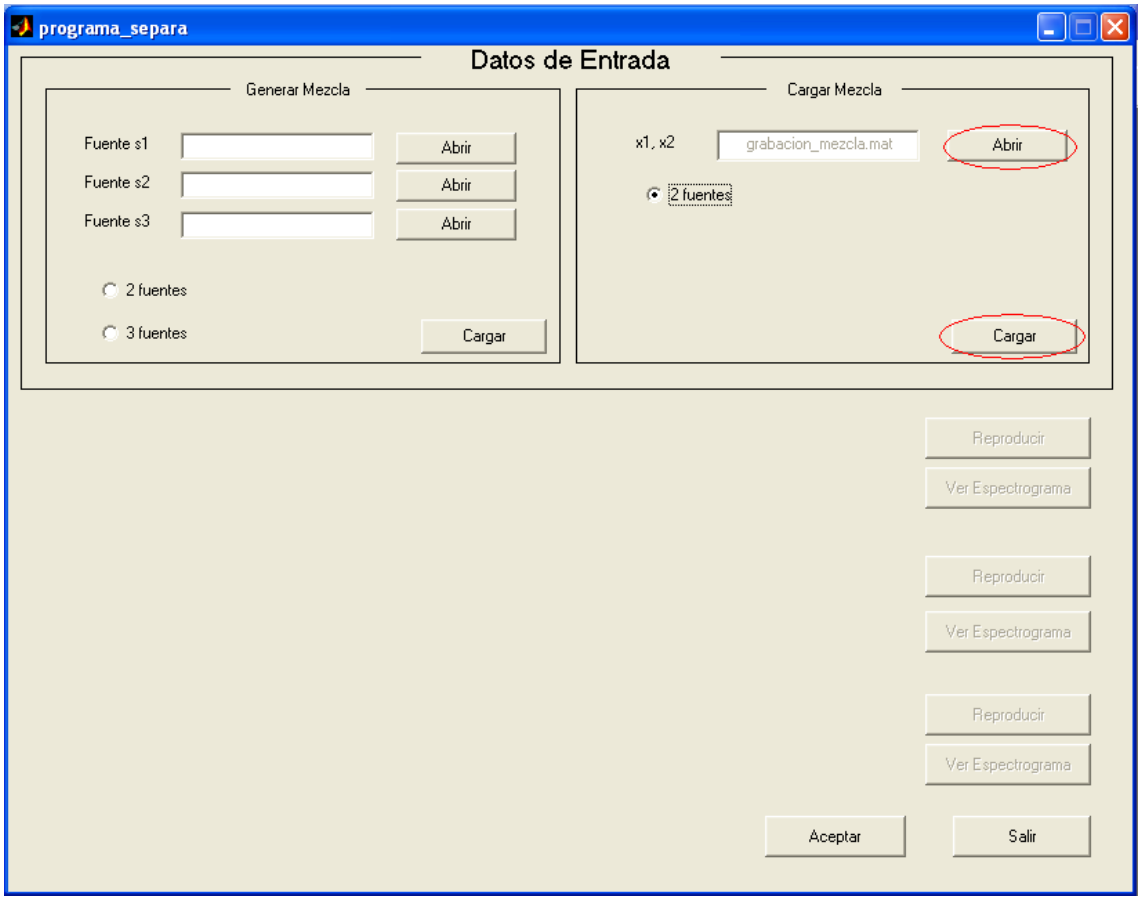


Figura A15

Ahora ya no tienen sentido algunos de los parámetros que elegíamos antes en la ventana de elección de los parámetros de simulación, puesto que teníamos que introducir datos sobre la situación geométrica del entorno en que se iba a realizar la mezcla para simularla. Como la mezcla ya está generada, sólo tendremos que elegir los valores de los datos que se usarán para calcular los espectrogramas, y que determinarán la resolución de los mismos, así como la elección del método que queremos usar para separar las señales fuente.

Figura A16

Una vez introducidos los datos, pulsamos el botón “Separar” y se pasará a la presentación de resultados, que será igual que la que ya hemos presentado anteriormente.

Acrónimos

BSS	Blind Source Separation (Separación ciega de fuentes)
DFT	Discrete Fourier Transform (Transformada discreta de Fourier)
DOA	Direction of Arrival (Dirección de llegada)
DTFT	Discrete-Time Fourier Transform (Transformada de Fourier de tiempo discreto)
EM	Expectation Maximization (Maximización de la esperanza)
FFT	Fast Fourier Transform (Transformada rápida de Fourier)
GUI	Graphical User Interface (Interfaz gráfica de usuario)
ICA	Independent Component Analysis (Análisis de componentes independientes)
ISTFT	Inverse Short-Time Fourier Transform (Transformada de Fourier de corta duración inversa)
ML	Maximum Likelihood (Máxima verosimilitud)
MMI	Minimum Mutual Information (Mínima información mutua)
MSE	Mean Square Error (Error cuadrático medio)
PCA	Principal Component Analysis (Análisis de componentes principales)
STFT	Short-Time Fourier Transform (Transformada de Fourier de corta duración)

Referencias Bibliográficas

[Anemüller01] **Across-Frequency Processing in Convolutive Blind Source Separation.**
Jörn Anemüller.
Universidad de Oldenburgo, 2001.

[Grossman95] **Álgebra Lineal**
Stanley I. Grossman.
McGraw-Hill, 1995.

[Anemüller00] **Amplitude Modulation Decorrelation for Convolutive Blind Source Separation.**
Jörn Anemüller.
Birger Kollmeier.
Proceedings of the second international workshop on independent component analysis and blind signal separation, 2000.

[Murata98] **An Approach to Blind Source Separation Based on Temporal Structure of Speech Signals.**
Noboru Murata.
Shiro Ikeda.
Andreas Ziehe.
Technical Reports nº98-2, RIKEN Brain Science Institute, 1998.

[Bell95] **A non-linear information maximization algorithm that performs blind separation.**
A. J. Bell.
T.J. Sejnowski.
The MIT Press, Cambridge, MA, 1995.

[Mitianoudis02] **Audio Source Separation: Solutions and Problems.**
Nikolaos Mitianoudis.
Mike E. Davies.
International Journal of Adaptive Control and Signal Processing, 2002.

[Yilmaz04] **Blind Separation of Speech Mixtures via Time-Frequency Masking.**
Özgür Yilmaz.
Scott Rickard.
IEEE Transactions on Signal Processing, vol. 52, nº7, Julio 2004.

[Makino05] **Blind Source Separation of Convolutive Mixtures of Speech in Frequency Domain.**
Shoji Makino.
Hiroshi Sawada.
Ryo Mukai.
Shoko Araki.
IEICE Trans. Fundamentals, vol. E88-A, nº 7, Julio 2005.

[Oppenheim89] **Discrete-Time Signal Processing.**
Alan V. Oppenheim.
Ronald W. Schafer.
Prentice Hall Signal Processing Series, 1989.

[ICA01] **Independent Component Analysis.**
Aapo Hyvärinen.
Juha Karhunen.
Erkki Oja.
Wiley Inter-Science, 2001.

[Bishop96] **Neural Network Pattern Recognition.**
C.M. Bishop.
Oxford University Press, 1996.

[Cruces02] **Robust blind source separation algorithms using cumulants.**
Sergio Cruces.
Luis Castedo.
Andrzej Cichocki.
Neurocomputing, vol. 49, nº 1, 2002.

[Makino06] **Solving the permutation problem of frequency-domain BSS when spatial aliasing occurs with wide sensor spacing.**
Hiroshi Sawada.
Shoko Araki.
Ryo Mukai.
Shoji Makino.
IEEE, 2006.