

CAPÍTULO 5

SIMULACIONES Y ESTUDIO DEL COMPORTAMIENTO DEL ALGORITMO

5.1 Ilustración del funcionamiento de la versión básica del algoritmo

5.1.1 Introducción

El algoritmo propuesto en el capítulo anterior para la separación de fuentes con pequeños retardos se implementó en el lenguaje de programación MATLAB con el objeto de estudiar el correcto funcionamiento del mismo. En este capítulo, se analizará el comportamiento del algoritmo y la validez de las aproximaciones realizadas en el capítulo anterior. Además se presentarán los resultados obtenidos en las simulaciones de aquellos esquemas de separación que se han considerado más significativos, para mostrar cómo se comporta el método de separación propuesto, al variar los diversos parámetros que intervienen en el proceso de mezcla y separación de las fuentes.

5.1.2 Ejemplo de separación de mezclas de fuentes con retardos

Para mostrar el funcionamiento del algoritmo, se procedió a separar una mezcla de cuatro fuentes de audio procedentes de cuatro hablantes distintos.

Para la separación se emplearon un total de $M = 9$ receptores. Según se vio en el capítulo anterior, el número de receptores debe ser, en general, mayor que $2N$ siendo, en este caso, $N = 4$.

La información recibida en cada uno de los receptores consistía en una combinación lineal de las fuentes transmitidas con un cierto retraso en cada uno de ellos. Se asume que las cuatro fuentes están en la misma región del espacio y que los receptores se encuentran en una misma región del espacio, aunque separados una pequeña distancia entre sí, por lo que las señales serán recibidas en ellos con distintos retardos porque la distancia de cada fuente a cada receptor es distinta.

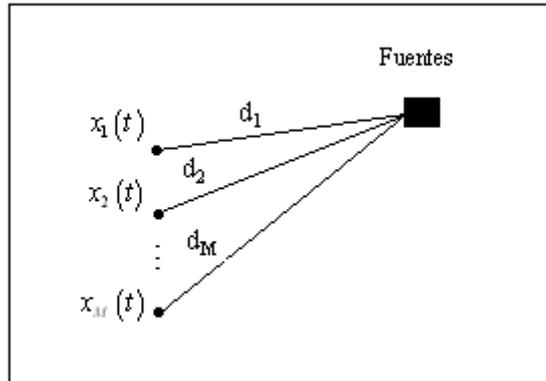


Figura 5.1: Situación de las fuentes y de los receptores.

Para generar la información recibida en los receptores, se introdujeron en cada receptor unos retardos distintos y aleatorios en las fuentes originales y posteriormente se mezclaron las fuentes retardadas con una matriz de mezclas de dimensión $N \times 1$. La matriz de mezclas se genera de forma aleatoria en cada simulación y sus elementos son números reales uniformemente distribuidos en el intervalo $[-1,1]$.

Según lo visto en el tema anterior, para que el retardo desde la fuente j hasta el receptor i , que denotamos como τ_{ij} se considere pequeño debe verificarse:

$$\tau_{ij}^2 \ll \frac{1}{2\pi^2 f_{\max}^2} \quad \forall i, j \quad (5.1)$$

Donde f_{\max} es el ancho de banda máximo de las señales que se transmiten.

Puesto que se trabaja con señales de audio, se escogió una $f_{\max} = 2\text{KHz}$. Se tomó este valor para respetar la condición anterior y a la vez permitir que las señales de

audio con las que se trabaja sean perfectamente inteligibles, a fin de poder comprobar fácilmente su calidad tras el proceso de separación.

Para el filtrado de las señales de audio, se utilizó un filtro LP elíptico. Se utilizó un filtro elíptico porque es el que necesita menor orden para satisfacer unas especificaciones dadas. Sus características son las siguientes:

- Extremo de la banda de paso $f_p = 2\text{KHz}$
- Extremo de la banda de rechazo $f_s = 2.08\text{KHz}$
- Rizado en la banda de paso $R_p = 0.5\text{dB}$
- Rizado en la banda de rechazo $R_s = 30\text{dB}$

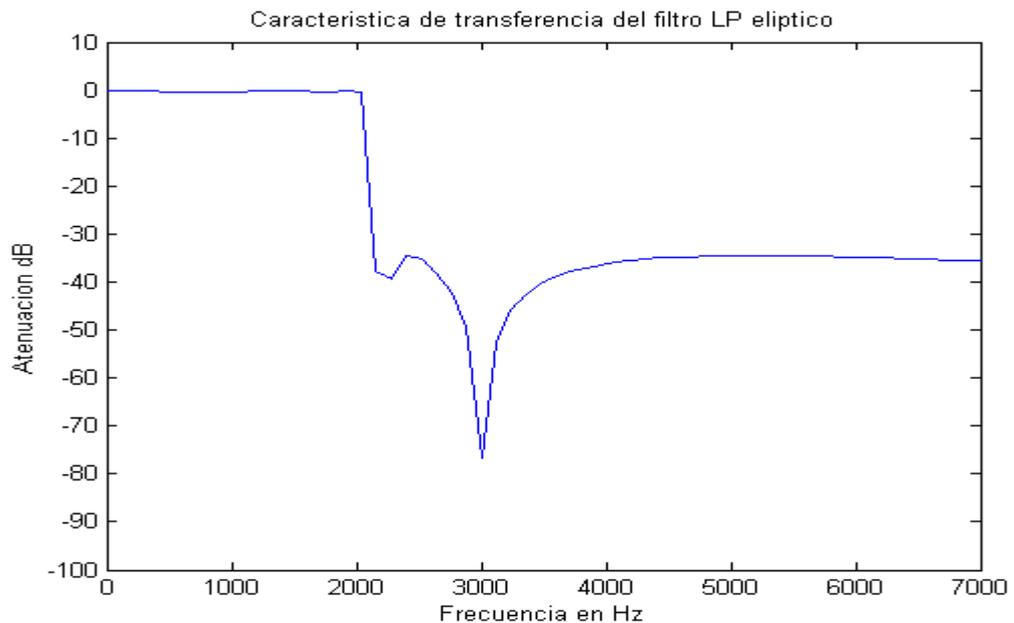


Figura 5.2: Característica de transferencia del filtro LP utilizado

Si en la desigualdad (5.1) imponemos que el término de la izquierda sea un orden de magnitud menor que el de la derecha, se obtendrá la siguiente ecuación:

$$10\tau_{ij}^2 = \frac{1}{2\pi^2 f_{\max}^2} \quad (5.2)$$

Sustituyendo f_{\max} , se obtiene el retraso máximo admisible que podemos tener:

$$\tau_{ij} = \frac{1}{\sqrt{20\pi} f_{\max}} = 3.6 \cdot 10^{-5} s \quad (5.3)$$

Las señales de audio utilizadas fueron grabadas con una tasa de muestreo de 48KHz. Se escogió una tasa de muestreo elevada para obtener un margen más amplio en los retardos que se pueden introducir y a la vez cumplir la condición de pequeños retardos.

Así, con los valores que se tomaron el número máximo de muestras que podemos retrasar cumpliendo la desigualdad (5.1) es:

$$3.6 \cdot 10^{-5} s \times \frac{48000 \text{muestras}}{s} = 1.75 \approx 2 \text{muestras} \quad (5.4)$$

En la simulación se asignaron retardos a las fuentes de una o dos muestras a las fuentes de forma arbitraria. Dichos retardos se introdujeron filtrando las fuentes con un filtro de respuesta impulsiva $h(t) = \delta(t - n_0)$, siendo n_0 el número de muestras que se decidió retardar.

Por último a las señales recibidas en los receptores se les añadió ruido gaussiano¹ de media cero, de forma que se obtuvo una relación señal a ruido aproximada de 30dB.

A continuación se presentan las señales originales, sus mezclas retardadas y las estimaciones de las señales originales que realiza el algoritmo:

¹ Posteriormente se estudiará el comportamiento del algoritmo frente al ruido.

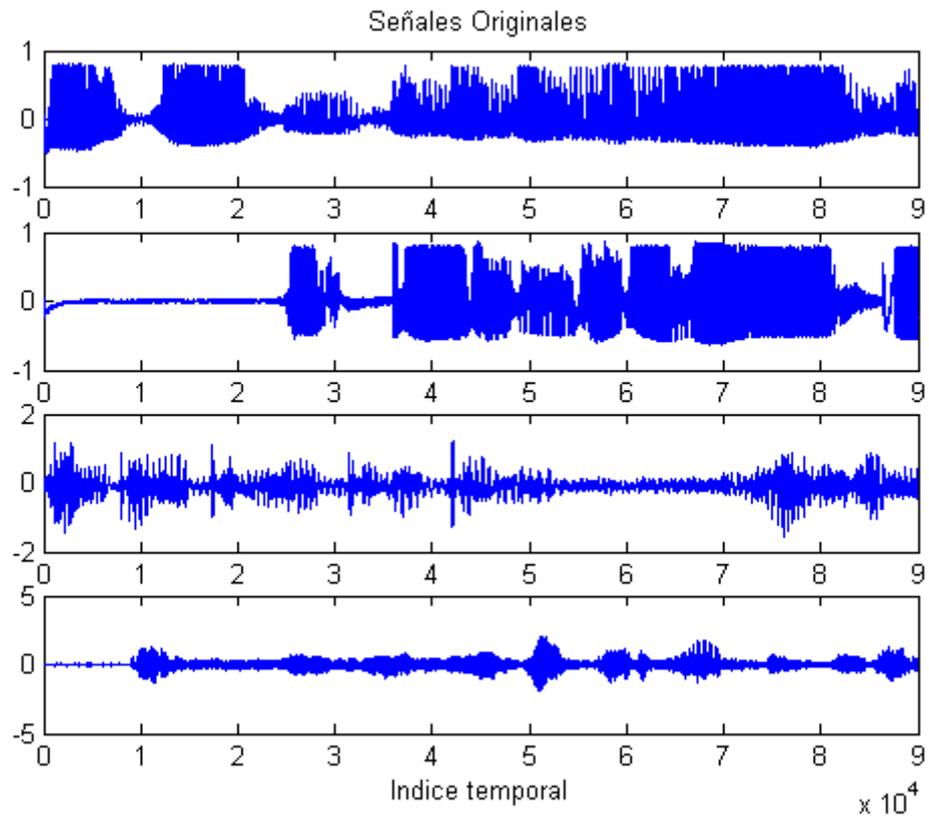


Figura 5.3: Señales de audio emitidas por las fuentes

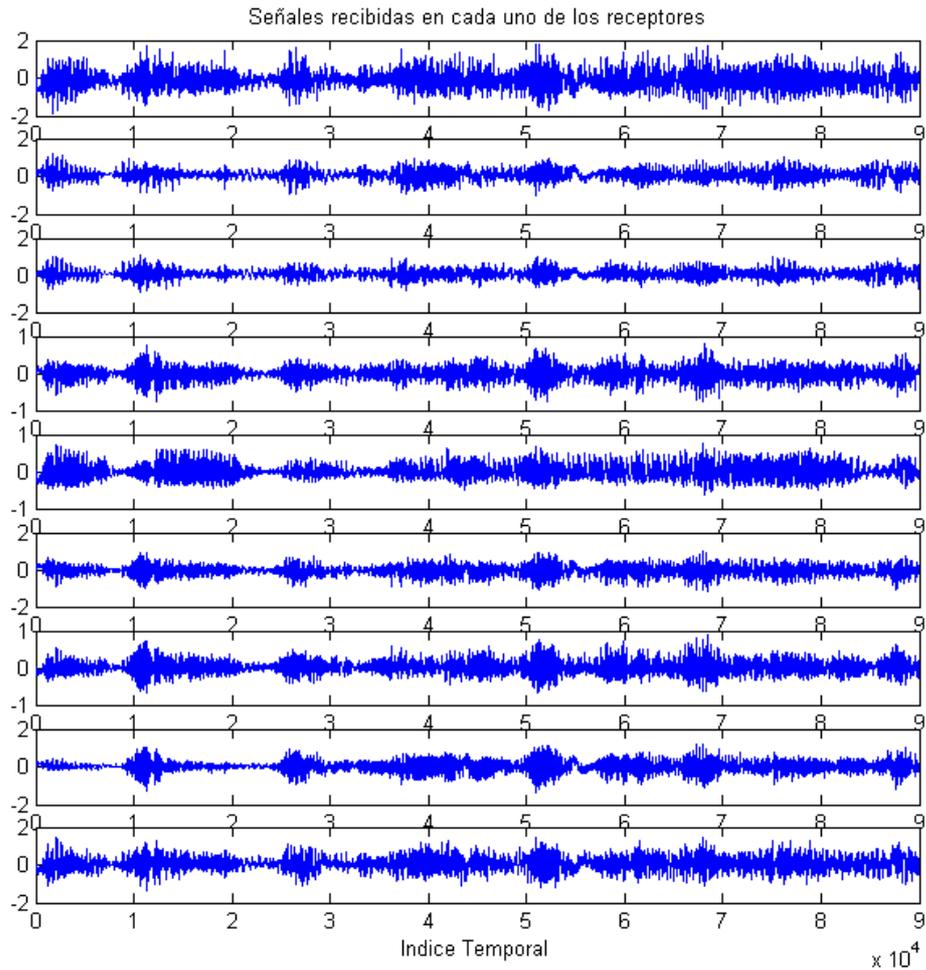


Figura 5.4: Mezclas de señales recibidas en los receptores

Tal como puede verse en la *figura 5.4*, en cada receptor se recibe una combinación lineal de las fuentes originales con un cierto retardo, más el ruido de tipo gaussiano que se ha añadido para hacer modelar con más precisión las señales que se reciben en los sensores. Tal como se vio en el capítulo anterior, las señales recibidas en los receptores pueden expresarse de la forma:

$$x_i(t) = \sum_{j=1}^N c_{ij} s_j(t - \tau_{ij}) + n_i(t) \quad (5.5)$$

Con $i = 1, \dots, M$.

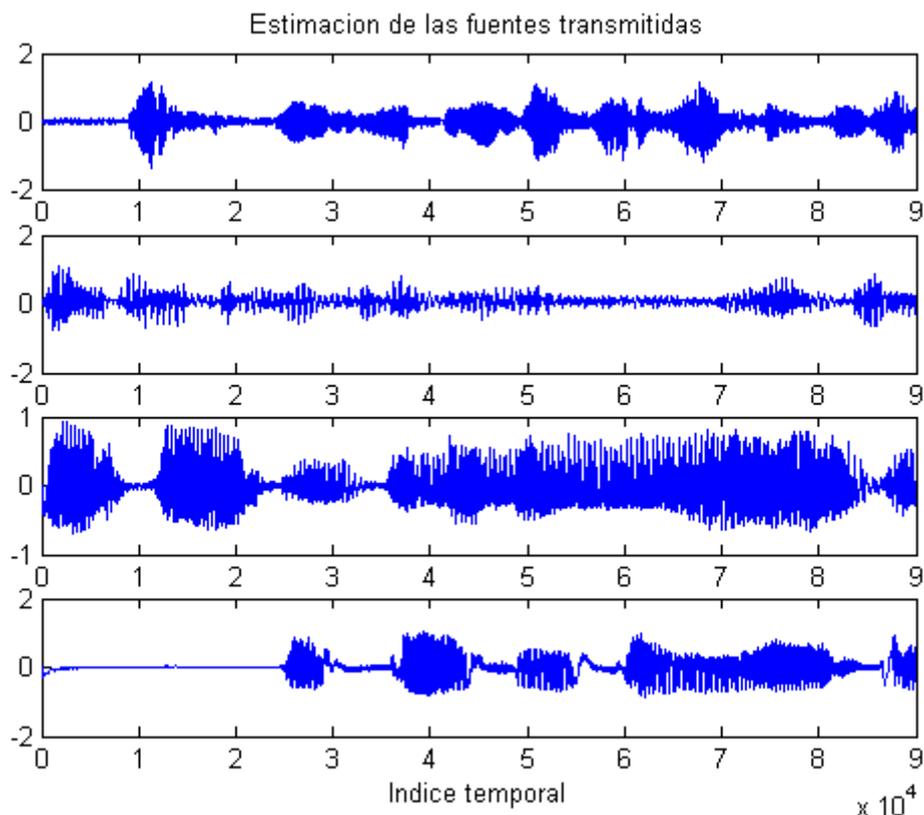


Figura 5.5: Estimación de las fuentes que se transmitieron

En la *figura 5.5*, se aprecia la estimación realizada por el algoritmo de las fuentes originales a partir de las mezclas. Las señales obtenidas son bastante parecidas a las originales. A simple vista, puede asociarse cada estima con la fuente correspondiente. También puede verse que el algoritmo es incapaz el orden y la amplitud de las señales que se mezclaron, tal como se comentó en el capítulo 2 en el apartado dedicado a las restricciones de ICA.

5.1.3 Obtención de las derivadas de las fuentes originales

Según el estudio realizado en el apartado anterior, la señal recibida en el receptor *i*-ésimo puede escribirse de la siguiente forma:

$$x_i(t) = \sum_{j=1}^N c_{ij} s_j(t - \tau_{ij}) + n_i(t) \quad (5.6)$$

Teniendo en cuenta que los términos de la forma $s(t-\tau)$ pueden expresarse de manera aproximada como $s(t-\tau) \approx s(t) - \tau \dot{s}(t)$, la señal recibida en el receptor viene dada por:

$$x_i(t) \approx c_{i,1}s_1(t) - c_{i,1}\tau_{i,1} \frac{ds_1(t)}{dt} + c_{i,2}s_2(t) - c_{i,2}\tau_{i,2} \frac{ds_2(t)}{dt} + \dots \quad (5.7)$$

$$\dots + c_{i,N}s_N(t) - c_{i,N}\tau_{i,N} \frac{ds_N(t)}{dt} + n_i(t)$$

El algoritmo de separación propuesto considera las derivadas primeras de las fuentes transmitidas como nuevas fuentes a separar. Tal como puede verse en la expresión anterior, dichas derivadas aparecen multiplicadas por el retardo τ_{ij} que se produce desde la fuente j hasta el receptor i . Este término de retardo es, por definición, pequeño y hará que la amplitud de las derivadas de las fuentes sea varios órdenes de magnitud más pequeña que la amplitud de las fuentes originales. En el caso particular de la simulación realizada, el valor máximo admisible que se obtuvo para el retardo desde cualquier fuente a cualquier receptor fue de $\tau_{ij_{\max}} = 3.6 \cdot 10^{-5} s$.

Es de esperar que, siempre que se cumpla la condición de pequeños retardos, la estimación que realiza el algoritmo de las derivadas no sea adecuada, puesto que su amplitud es mucho menor que la de las fuentes originales. Además si existe ruido, puede ocurrir que la amplitud del mismo sea del mismo orden de magnitud que la de las derivadas o incluso superior, por lo que la tarea de estimación de las derivadas se dificulta aún más. Cuando no se cumpla la desigualdad (5.1), la estimación de las derivadas será correcta puesto que su amplitud será comparable a la de las fuentes.

De cualquier manera, el objetivo del algoritmo es estimar las fuentes originales y no sus derivadas. Si se quisiera hacer una estimación de las mismas, se obtendrían resultados mucho más exactos si simplemente se calculara la derivada primera de las estimaciones de las fuentes.

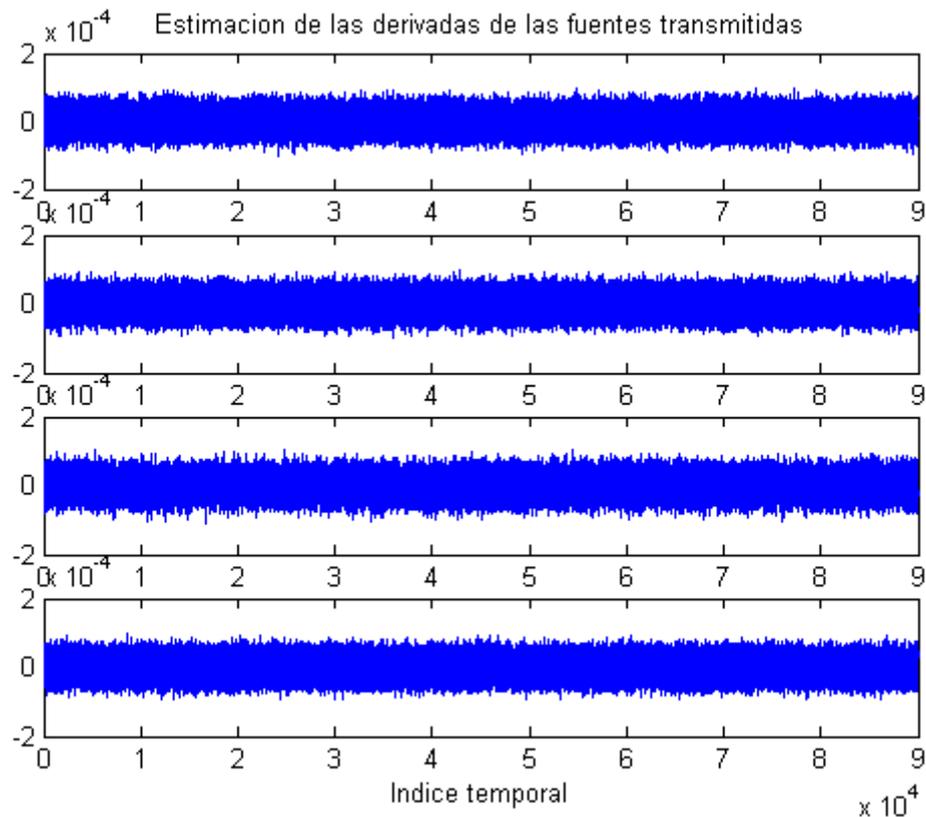


Figura 5.6: Estimación de las derivadas primeras de las fuentes.

Tal como puede verse en las gráficas de arriba, el algoritmo es incapaz de determinar con exactitud las derivadas en condiciones normales de funcionamiento, ya que éstas aparecen enmascaradas por el propio ruido que se ha introducido en la simulación, ya que su energía es menor que la del ruido. Esta será la situación que se tendrá siempre que se introduzca un nivel de ruido apreciable.

Para mostrar la estimación que el algoritmo hace de las derivadas, se realizó una simulación en las mismas condiciones del apartado anterior, pero con una salvedad, se eliminó el ruido (condiciones ideales) para que éste no enmascare a las derivadas.

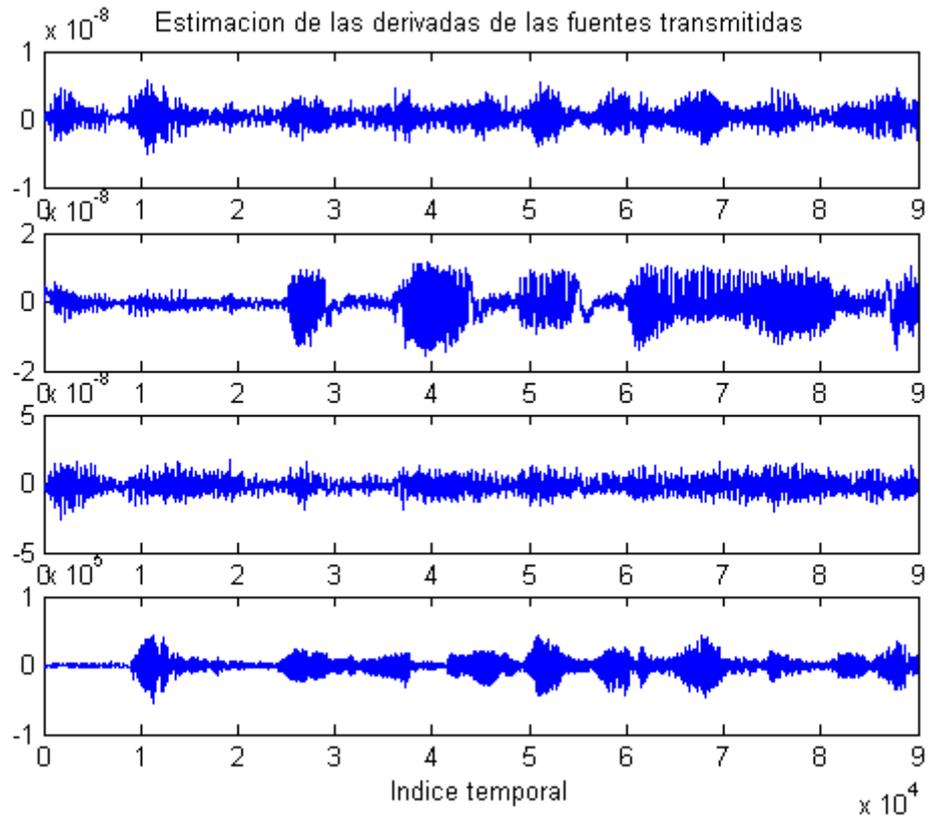


Figura 5.7: Estimación realizada de las derivadas en ausencia de ruido

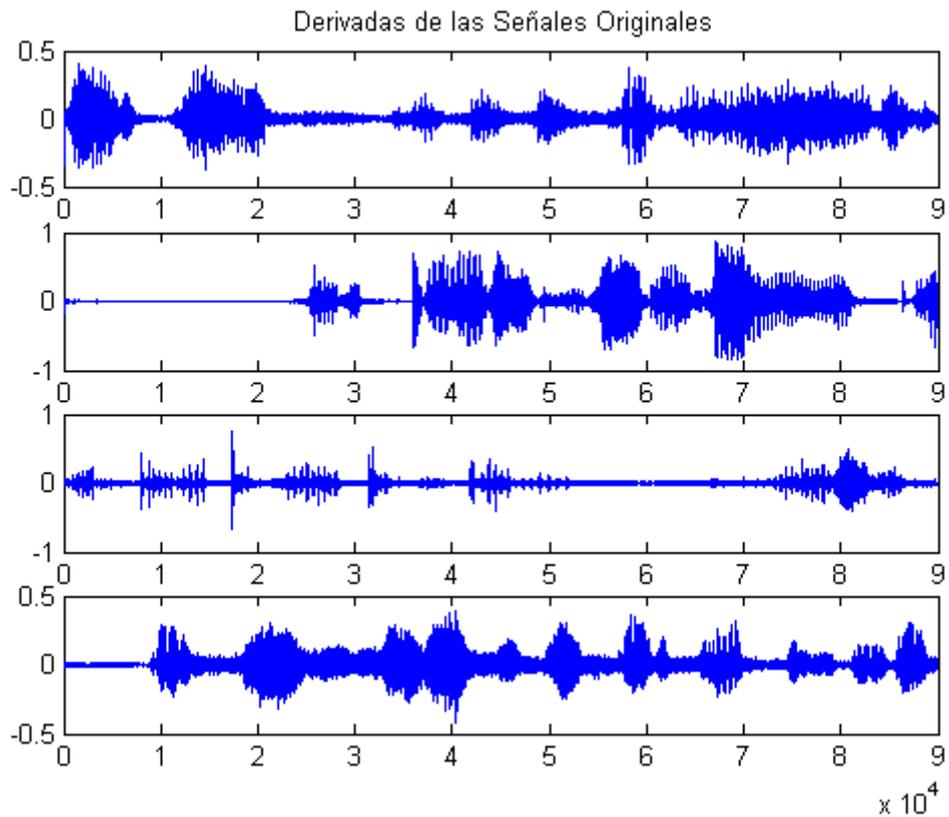


Figura 5.8: Valores reales de las derivadas de las fuentes transmitidas

Comparando las estimaciones de las derivadas con y sin ruido, puede verse que mejora de forma notable cuando se elimina el ruido. Pese a todo, si se comparan las estimaciones realizadas de las derivadas con respecto a las derivadas de cada una de las fuentes, vemos que la estimación realizada por el algoritmo sigue siendo bastante burda.

A pesar de ello, es posible identificar a simple vista a qué derivada corresponde cada una de las derivadas estimadas por el algoritmo. La primera gráfica de la *figura 5.7* es la estimación de la primera gráfica de la *figura 5.8* y así sucesivamente.

En resumen, la estimación de las derivadas que hace el algoritmo no es buena debido a la pequeña amplitud de éstas frente a la de las fuentes. Esto es algo lógico, ya que si unas señales poseen mucha más energía que otras, es habitual que los algoritmos ICA fallen. Pese a todo, esto no supone un problema puesto que el objetivo del algoritmo no es la obtención de las derivadas, sino la de las fuentes originales.

En general, el algoritmo devuelve las señales estimadas (derivadas y fuentes) desordenadas. La separación puede hacer simplemente, escogiendo las N señales de mayor energía. Estas se corresponderán con las señales de interés.

Posteriormente se verá cómo las ampliaciones realizadas sobre el algoritmo básico permiten obtener de manera directa las fuentes de interés, sin tener que hacer una separación entre fuentes y derivadas.

También se mostrará cómo al introducir retardos por encima de los admisibles, la estima de las derivadas se hace de forma correcta, ya que estos casos la amplitud de las derivadas no es despreciable frente a la de las fuentes y a la del propio ruido.

5.1.4 Comportamiento del algoritmo frente al ruido

5.1.4.1 Respuesta ante el ruido

Tal como se introdujo en el capítulo anterior, supondremos que el ruido recibido en todos los receptores es blanco y gaussiano. Por simplicidad, se supondrá que el ruido recibido en todos los receptores tiene la misma varianza σ^2 .

El cálculo de la relación señal a ruido en cada receptor es complejo. Si a cada receptor se le asocia una fuente de interés, en él se recibe además del ruido y de la propia señal de interés, una combinación lineal del resto de señales. Esta combinación lineal puede considerarse como un ruido adicional o interferencia.

En las simulaciones se emplearon señales de audio con un nivel de energía similar. A la hora de estimar la relación señal a ruido, se utilizó el siguiente criterio:

$$SNR = \frac{E_{señales}}{E_{ruido}} \quad (5.8)$$

Siendo $E_{señales}$ la energía de las señales y E_{ruido} la energía de la señal de ruido que se introdujo. Es un criterio que arroja resultados pesimistas porque en cada receptor se recibe ruido adicional debido a combinaciones lineales de señales distintas a la de interés, que pueden considerarse como interferencia, por lo que la SNR será más baja en realidad que los resultados que nos dé la expresión anterior.

Para evaluar la robustez del algoritmo frente al ruido, se realizaron varias simulaciones con un número de fuentes comprendido entre 3 y 5. En ellas, se fue aumentando de forma progresiva el nivel de ruido para observar las degradaciones producidas en las señales estimadas².

- Para relaciones señal a ruido $SNR > 18dB$ el comportamiento del algoritmo es bastante acertado y las señales de audio son extraídas sin que se perciba la influencia del ruido. Para relaciones en el rango $14dB < SNR < 18dB$, las señales son extraídas de forma correcta, sin embargo, al oírlas puede apreciarse la influencia de un cierto nivel de ruido de fondo.
- Si bajamos de una $SNR = 14dB$, el algoritmo dejará de comportarse correctamente y en las señales comenzarán a apreciarse pequeños lóbulos debido a las otras señales distintas a la de interés. Por lo que al escuchar las señales, se oirán fuentes superpuestas a la de interés.
- Para $SNR < 10dB$, el algoritmo deja de funcionar y en las estimas se obtiene una superposición de las señales de interés.

² En dichas simulaciones se eligió el filtro y el número de receptores de forma óptima

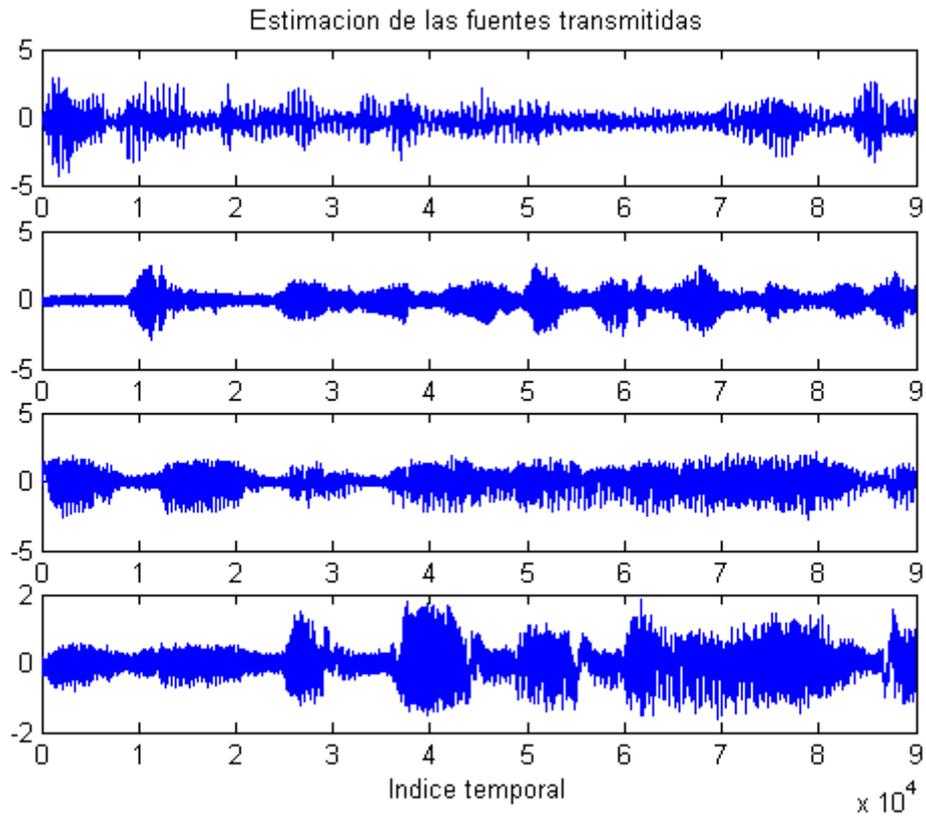


Figura 5.9: Estimaciones de las señales para una $SNR \approx 8dB$

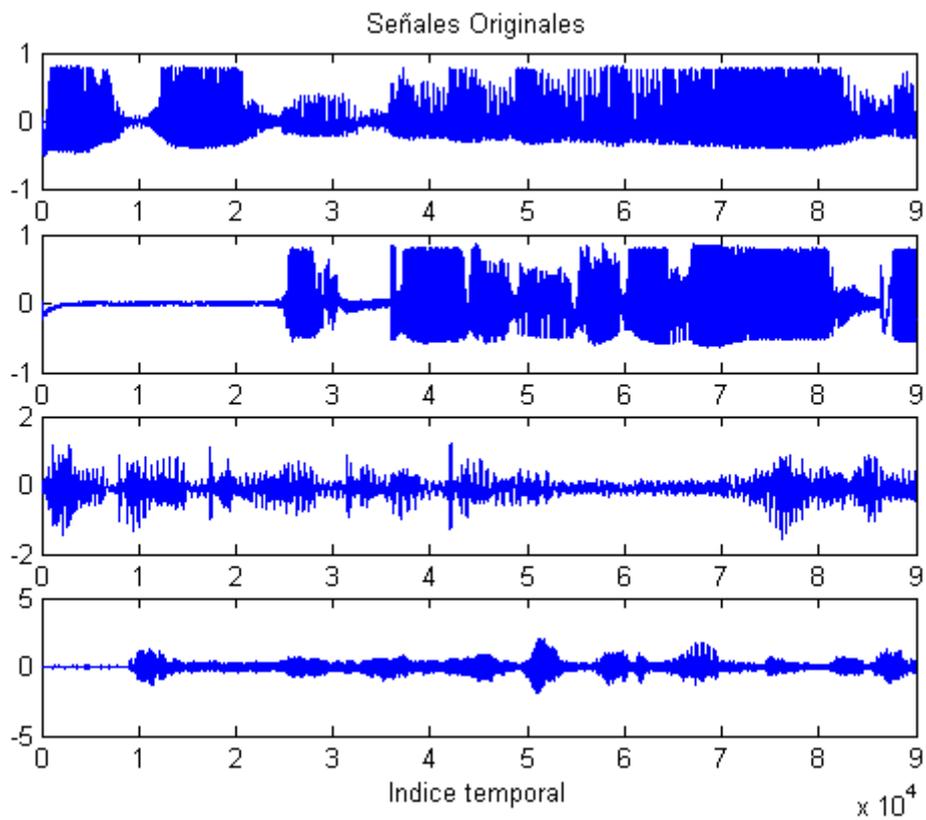


Figura 5.10: Señales originales

En las figuras anteriores puede verse lo que ocurre cuando empiezan a introducirse relaciones señal a ruido por debajo de los $10dB$, al intentar extraer las fuentes originales, se obtienen combinaciones lineales de éstas porque el algoritmo confunde el espacio de las fuentes originales con el del ruido, ya que cuando su energía se hace comparable a la de las fuentes, puede considerarse el ruido como una nueva fuente estadísticamente independiente del resto.

5.1.4.2 Elección del filtro utilizado para la obtención de la segunda matriz de autocovarianzas

Según vimos en el capítulo anterior el algoritmo de separación se basa en el uso de dos matrices de autocovarianzas para la obtención de la matriz de separación. La matriz de autocovarianzas de las observaciones y la matriz de autocovarianzas de las observaciones filtrada con un filtro con respuesta impulsiva $h(t)$:

$$\mathbf{R}_{xx}(0) = \mathbf{A}\mathbf{D}_0\mathbf{A}^T + \sigma^2\mathbf{I}_M \quad (5.9)$$

$$\mathbf{R}_{xx}^h(0) = \mathbf{A}\mathbf{D}_0^h\mathbf{A}^T + \sigma^2h(0)\mathbf{I}_M \quad (5.10)$$

Para facilitar resolver el problema según el método propuesto el filtro ha de escogerse de forma que su respuesta impulsiva sea nula en cero, de forma que las ecuaciones anteriores queden así:

$$\mathbf{R}_{xx}(0) = \mathbf{A}\mathbf{D}_0\mathbf{A}^T + \sigma^2\mathbf{I}_M \quad (5.11)$$

$$\mathbf{R}_{xx}^h(0) = \mathbf{A}\mathbf{D}_0^h\mathbf{A}^T \quad (5.12)$$

En principio, cualquier filtro no nulo, con respuesta impulsiva nula en cero, será válido para que el algoritmo funcione. Sin embargo, el comportamiento frente al ruido depende mucho del filtro que se elija. Una correcta elección del filtro hará mucho más robusto al algoritmo frente al ruido.

Una elección incorrecta del filtro hará que la relación señal a ruido mínima para la cual el efecto del ruido es imperceptible aumente de forma considerable. Así en las

simulaciones realizadas con un filtro adecuado, se obtuvo una relación señal a ruido mínima 20dB más pequeña que cuando el filtro se elige de forma arbitraria.

El filtro $h(t)$ escogido para todas las simulaciones es el siguiente:

$$h(t) = \delta(t-1) + \delta(t-2) + \dots + \delta(t-n_0) \quad (5.13)$$

Siendo n_0 un entero comprendido en el rango $[10,15]$. El filtro presenta respuesta impulsiva nula en cero para cancelar el término debido al ruido y el resto de sus coeficientes son iguales a la unidad. En las señales de voz, las distintas muestras están parcialmente correladas para valores o distancias de hasta unas 15 muestras. Se trata de escoger un filtro que altere lo menos posible las propiedades de la señal de audio.

5.1.4.3 Comportamiento del algoritmo al variar el número de receptores

Según lo visto en el capítulo anterior, el número mínimo de receptores que debe emplearse es de $M = 2N$ en el caso de que no haya ruido y de $M = 2N + 1$ en el caso con ruido. Nos centraremos, a partir de ahora, en este último caso por ser el más general.

Tal como se vio, el algoritmo para calcular la matriz de separación realiza la descomposición en valores singulares de la matriz de autocovarianzas de las observaciones:

$$\mathbf{R}_{xx}(0) = [\mathbf{u}_1, \dots, \mathbf{u}_M] \mathbf{L} [\mathbf{u}_1, \dots, \mathbf{u}_M]^T \quad (5.14)$$

$$\text{Siendo } \mathbf{L} = \text{diag}(\lambda_1^2 + \sigma^2, \dots, \lambda_{2N}^2 + \sigma^2, \sigma^2, \dots, \sigma^2) \quad (5.15)$$

Y a partir de la media de los últimos $M - 2N$ elementos de la diagonal de la matriz \mathbf{L} se realiza una estimación de la varianza del ruido σ^2 . Lógicamente mientras más próxima sea esta estimación al valor real mejor será el comportamiento del algoritmo. Por tanto, es de esperar que si se aumenta el número de receptores, los resultados dados por el algoritmo mejoren en presencia de ruido.

Los resultados obtenidos utilizando un número adecuado de filtros son mucho mejores que si se utiliza el número mínimo permitido.

Para mostrar el comportamiento frente al ruido al variar el número de receptores, se realizó una simulación con tres fuentes y se analizó la calidad de las estimaciones del algoritmo usando el número mínimo de receptores $M_1 = 2N + 1$ y un número suficientemente elevado (se tomó $M_2 = 14$).

En las simulaciones se obtuvo que la relación señal a ruido mínima para la que el ruido es imperceptible con un número adecuado de receptores es de $SNR_{\min} = 18dB$, mientras que si el número de receptores se reduce al mínimo, $SNR_{\min} = 28dB$, por lo que tomando un número suficientemente elevado de receptores, se obtiene una mejora de $10dB$.

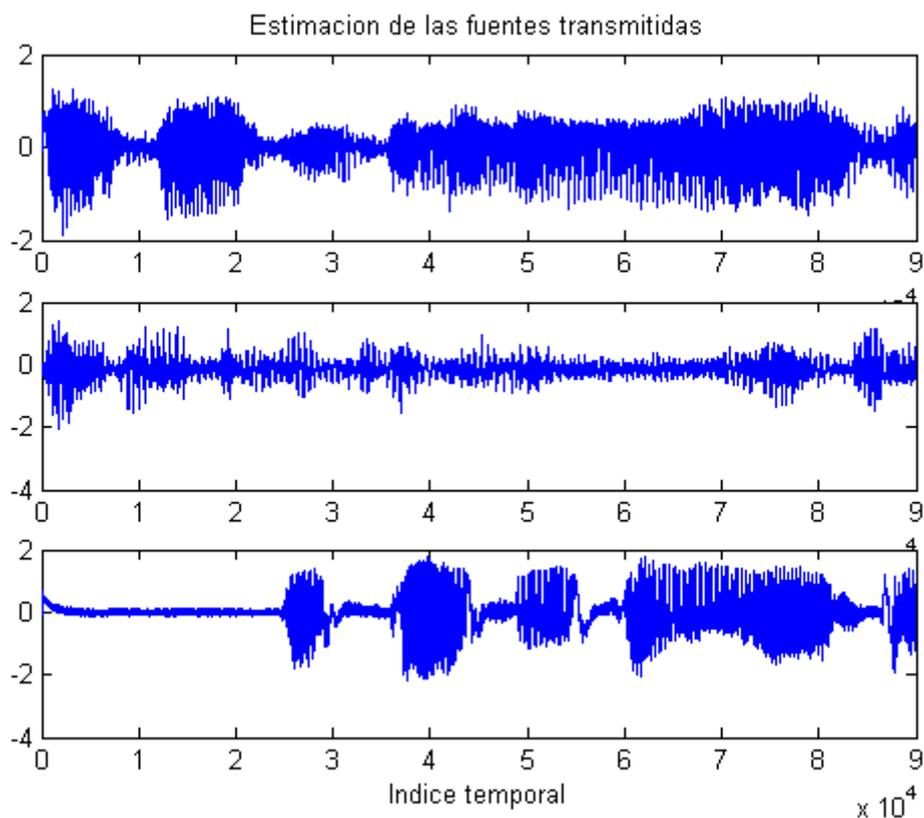


Figura 5.11: Estimación de las fuentes con una $SNR \approx 12dB$ y $M_2 = 14$

En la figura de arriba puede verse la estimación de las fuentes realizada por el algoritmo para una relación señal a ruido de $12dB$. Si comparamos las estimaciones de

las fuentes con las fuentes originales de la *figura 5.3*, vemos que el algoritmo se comporta de forma correcta.

A continuación se redujo el número de receptores al mínimo. Tal como puede verse en la gráfica de abajo, el algoritmo es incapaz de estimar correctamente las fuentes que se transmitieron.

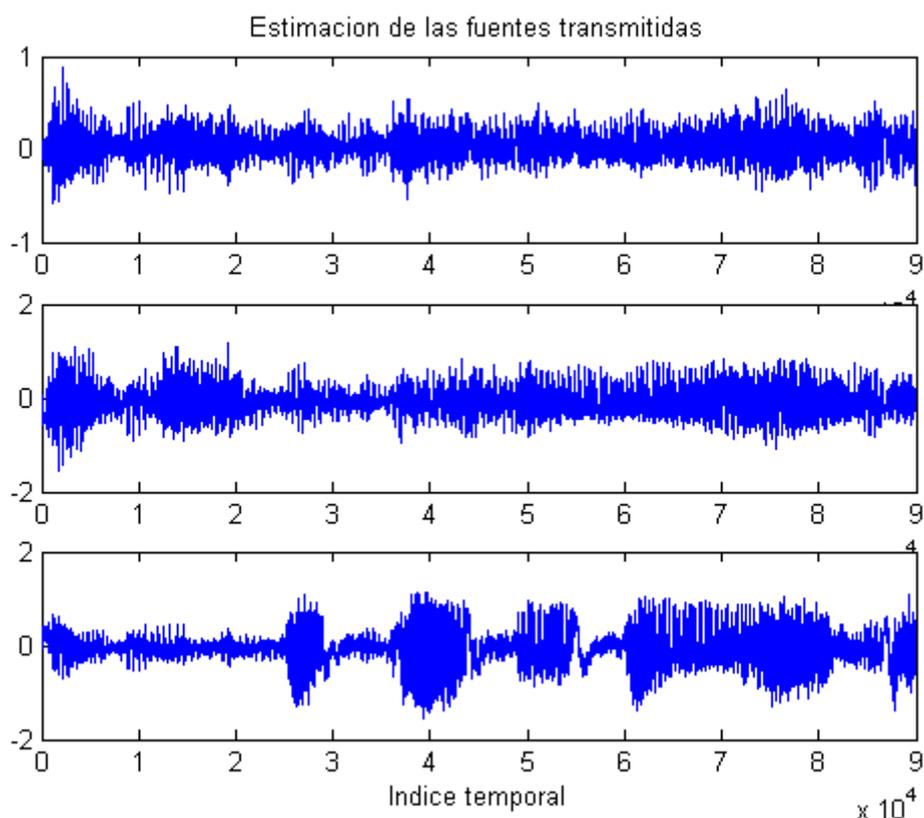


Figura 5.12: Estimación de las fuentes transmitidas para una $SNR \approx 12dB$ y $M_1 = 7$

En la práctica, al aumentar el número de receptores por encima de un cierto umbral, no se obtienen mejoras en el comportamiento del algoritmo.

A la vista de los resultados obtenidos en varias simulaciones, un número bastante acertado para el número de receptores resultó ser el doble del número mínimo que podemos tener, es decir, $M = 2*(2N + 1)$.

Es preciso remarcar que el algoritmo maneja matrices de hasta dimensión $M \times M$, por lo que si se aumenta mucho el número de receptores, porque queramos mejorar el comportamiento ante el ruido o porque el número de fuentes a separar sea elevado, el tiempo de respuesta aumentará considerablemente.

5.1.5 Comportamiento frente a los retardos de las fuentes

Para mostrar el comportamiento del algoritmo frente a los retardos, se realizaron varias simulaciones con las mismas condiciones que en el apartado 5.1.2. Tal como se estableció anteriormente, de acuerdo con la ecuación (5.2), el número máximo de retardos admisibles entre las fuentes recibidas es de dos muestras. Para retardos mayores, se estará violando la condición de pequeños retardos. En este apartado, se hicieron simulaciones introduciendo retardos mayores que los admisibles y se estudió el comportamiento del método de separación en estos casos. Los resultados obtenidos fueron los siguientes:

- Para retardos de 1 a 2 muestras (tal como se mostró) el comportamiento es el esperado.
- Cuando se introducen retardos de 3 a 4 muestras (ya estamos fuera de los límites establecidos) las fuentes se estiman de forma correcta también. En el proceso de extracción, ya empiezan a estimarse de forma correcta algunas de las derivadas de las fuentes. Ello se debe a que en la ecuación (5.7) los términos correspondientes a las derivadas ya no son despreciables frente a las fuentes porque los términos debidos a los retrasos τ_{ij} han aumentado considerablemente.
- Para retrasos en torno a las 8 muestras (las suposiciones realizadas ya no son correctas) el algoritmo deja de funcionar de forma adecuada. Las fuentes extraídas aunque todavía se reconocen bien, tienen una amplitud bastante pequeña comparable a la de las propias derivadas. En estos casos, ya la estima de las derivadas es correcta, aunque en algunos casos el algoritmo omite alguna de ellas, especialmente, si el nivel de ruido que se introduce es elevado. Decir que bajo estas condiciones, el algoritmo se vuelve mucho más sensible al ruido. Para relaciones señal a ruido por debajo de los $20dB$, el algoritmo ya no es capaz de estimar de forma correcta las fuentes.

- Para retrasos mayores de las 10 muestras, el comportamiento es inaceptable. La amplitud de las fuentes estimadas es muy baja. Además éstas aparecen distorsionadas y muy afectadas por el ruido.

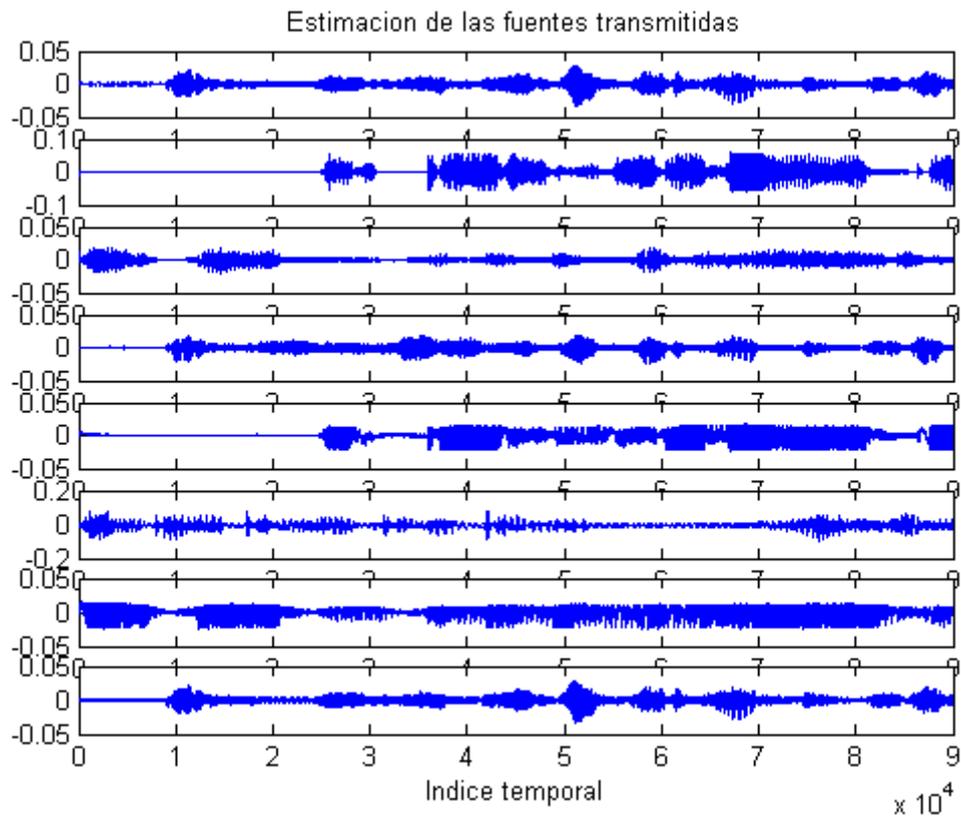


Figura 5.13: Estimación de las fuentes y sus derivadas para retrasos 8 muestras y una $SNR \approx 30dB$

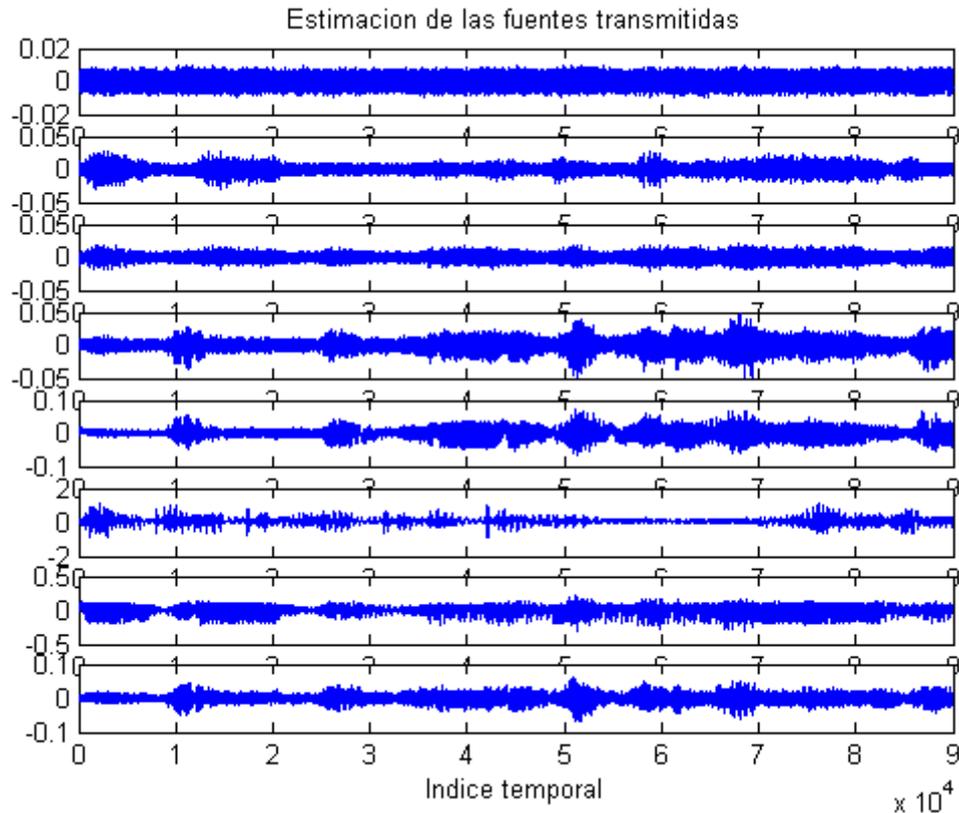


Figura 5.14: Estimación de las fuentes y sus derivadas para retrasos en torno a 11 muestras y una $SNR \approx 30dB$

En la *figura 5.13*, puede verse la estimación de las fuentes originales y sus derivadas cuando se introducen retardos de hasta 8 muestras. Es fácil identificar a simple vista las fuentes obtenidas, si las comparamos con las señales que aparecen en la *figura 5.3* (fuentes originales) y las de la *figura 5.8* (derivadas primeras de las fuentes originales).

Para estos retardos tan elevados la amplitud de las fuentes deseadas se hace muy pequeña. Además su calidad es baja debido a la distorsión.

En la *figura 5.14* se observan los resultados obtenidos para retrasos de hasta 11 muestras. Claramente el algoritmo no funciona bien y tanto las fuentes como sus derivadas no se estiman de forma correcta.

Un caso particular en el que también podemos tener retardos, es el de la *figura 5.15*. Se trata de la situación contraria considerada hasta ahora, las fuentes se encuentran separadas una pequeña distancia entre sí y los receptores se encuentran juntos en una misma región del espacio.

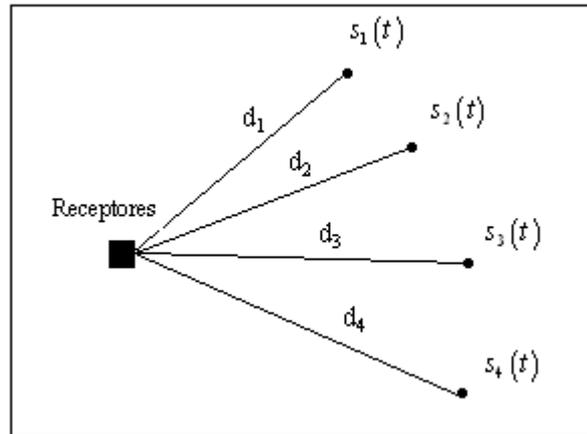


Figura 5.15: Segundo esquema de simulación posible

Las señales recibidas en los receptores pueden expresarse matemáticamente de la forma:

$$x_i(t) = \sum_{j=1}^N c_{ij} s_j(t - \tau_j) + n_i(t) \quad (5.16)$$

En este caso, las señales de interés son recibidas en cada uno de los receptores con el mismo retardo puesto que se consideran que la distancia que los separa es nula. Podemos considerar el problema de separación como un problema de separación de mezclas instantáneas en el que las fuentes de interés son las señales retrasadas. Es por ello, que al simular esta situación, el algoritmo se mostró inmune a los retrasos introducidos. Se introdujeron retrasos muy por encima de los límites admisibles y la separación de las fuentes siempre fue perfecta. Por ello, en las diversas simulaciones realizadas se simuló la situación propuesta en la *figura 5.1*, que es la más restrictiva frente a los retardos y la que se propone en el artículo original, en el que se describe el algoritmo, véase [1].

Un tercer esquema de simulación más complejo consiste en una combinación de los dos anteriores, es decir, tanto las fuentes como los receptores se encuentran separados una pequeña distancia entre sí. En este caso, el comportamiento del algoritmo frente a los retardos fue exactamente el mismo que el obtenido con el primer esquema de simulación.

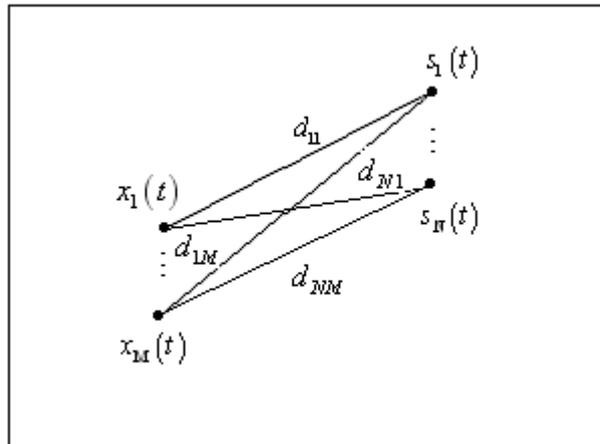


Figura 5.16: Tercer esquema de simulación posible

A modo de resumen, podemos decir que el comportamiento del algoritmo frente a los retardos es el esperado y que las hipótesis realizadas en el capítulo anterior a cerca de los retardos se cumplen. Además se ha visto que para retardos de hasta 4 muestras (el doble del valor establecido para los retardos máximos admisibles) el algoritmo sigue funcionando correctamente. A la vista de esto, si en la desigualdad (5.1) imponemos que el término de la izquierda sea unas cinco veces mayor que el de la derecha, en lugar de un orden de magnitud como se estableció inicialmente, obtendremos una expresión empírica para el retardo máximo que se puede introducir, acorde con las simulaciones realizadas. Así el retardo máximo admisible puede determinarse de forma aproximada mediante la siguiente expresión³:

$$\tau_{ij_{\max}} \approx \frac{1}{\sqrt{10\pi} f_{\max}} \quad (5.17)$$

³ Para relaciones señal a ruido por encima de los 20dB

5.2 Estudio del funcionamiento de la versión ampliada del algoritmo

5.2.1 Introducción

En este apartado se presentarán los resultados obtenidos en las diversas simulaciones de la versión ampliada del algoritmo que se expuso en el capítulo anterior. Dicha ampliación consistía básicamente en el uso de varios filtros en lugar de uno solo y en el uso de la operación de deflación para extraer las fuentes una a una.

A la vez que se vayan presentando los resultados obtenidos, se irán haciendo comparaciones con los resultados obtenidos con la otra versión del algoritmo, con el fin de evaluar las prestaciones de cada una de las dos implementaciones.

5.2.2 Ejemplo de separación de fuentes con retardos

Para mostrar el funcionamiento de la versión ampliada del algoritmo, se realizó una simulación en las mismas condiciones que en el apartado 5.1.2. Al igual que antes, se pretendía separar las voces de cuatro hablantes distintos, $N = 4$, usando un total de $M = 9$ receptores.

Las voces utilizadas fueron las mismas que en el apartado anterior. Además se filtraron de la misma forma que antes utilizando el filtro con la característica de transferencia presentada en la *figura 5.2* y el nivel de ruido que se introdujo no se varió.

Los retrasos se introdujeron de la misma forma que en la simulación anterior. Al igual que antes, se obtuvieron unos retardos máximos admisibles de dos muestras entre las señales recibidas.

La novedad de esta simulación consistió en el uso de varios filtros, en lugar de uno, para mejorar la estimación de las fuentes. Se decidió utilizar un total de cuatro filtros⁴, $N_{Fil} = 4$. Los filtros utilizados presentaban todos respuesta impulsiva nula en cero, tenían una longitud de unas 15 muestras y presentaban pequeñas variaciones entre

⁴ Posteriormente se presentará un análisis del comportamiento frente al número de filtros.

ellos. Básicamente estos filtros eran secuencias formadas por un número inicial de ceros pequeño en las primeras muestras, seguidos por un número mayor de unos en el resto de muestras. La razón de la elección de este tipo de filtros es que fueron los que mostraron mejor comportamiento por separado en la versión básica del algoritmo.

A continuación en la *figura 5.17*, se presenta la estimación realizada de las fuentes originales. Si se comparan a simple vista con las fuentes originales de la *figura 5.18*, apenas se observan diferencias entre las estimaciones y las fuentes reales.

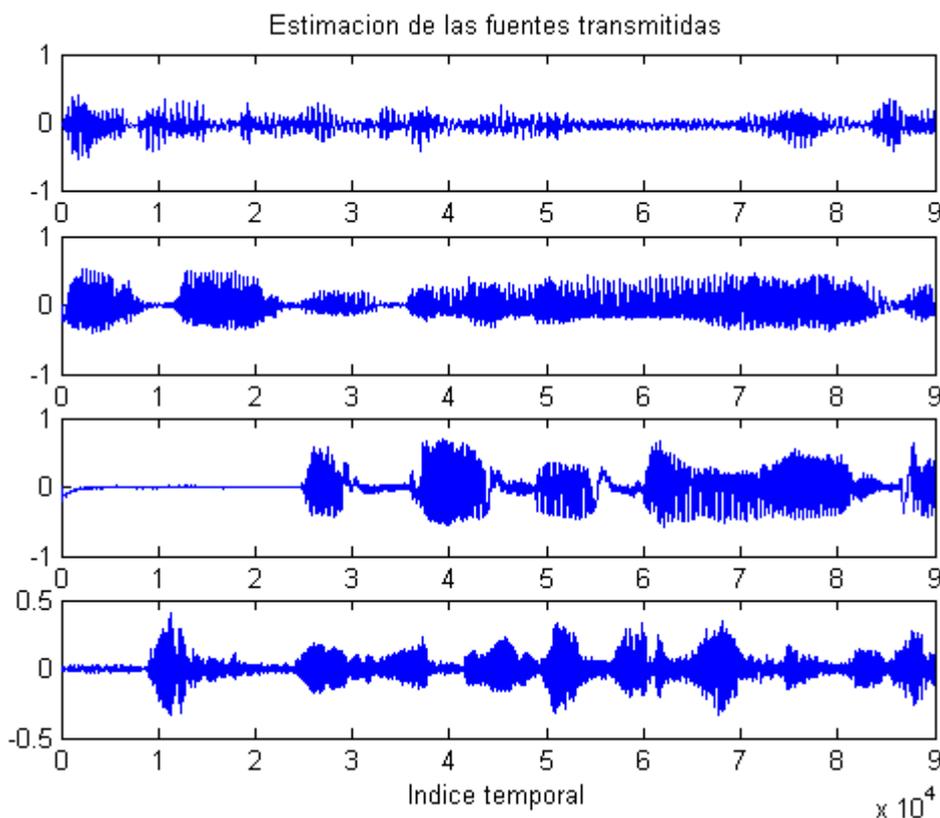


Figura 5.17: Estimación de las fuentes transmitidas en la versión mejorada del algoritmo

Como novedad, al ejecutar el algoritmo dentro de los límites admisibles para los retardos, las N primeras fuentes que se extraigan se corresponderán con las señales de interés, por lo que ya no tendremos que preocuparnos por la identificación de las fuentes y las derivadas al ejecutar el algoritmo, como ocurría en el caso anterior. Además, si decidimos extraer sólo las fuentes de interés, obtendremos un ahorro en el coste computacional. Pese a ello, esta versión es mucho más lenta que la anterior porque todos los cálculos matriciales con los vectores de observación deben repetirse cada

vez que se realiza la operación de deflación y, en este caso, se realizan cuatro veces más que en el ejemplo anterior.

Si se comparan los resultados obtenidos por las dos versiones del algoritmo en las mismas condiciones, observando los resultados obtenidos, a simple vista no se aprecian diferencias, ya que ambas versiones funcionan de forma acertada dentro de los límites establecidos. Tan sólo se aprecian algunas pequeñas diferencias al escuchar las señales extraídas. En la versión ampliada, la calidad de las señales de audio es mejor, ya que la influencia del ruido que se añadió apenas se nota.

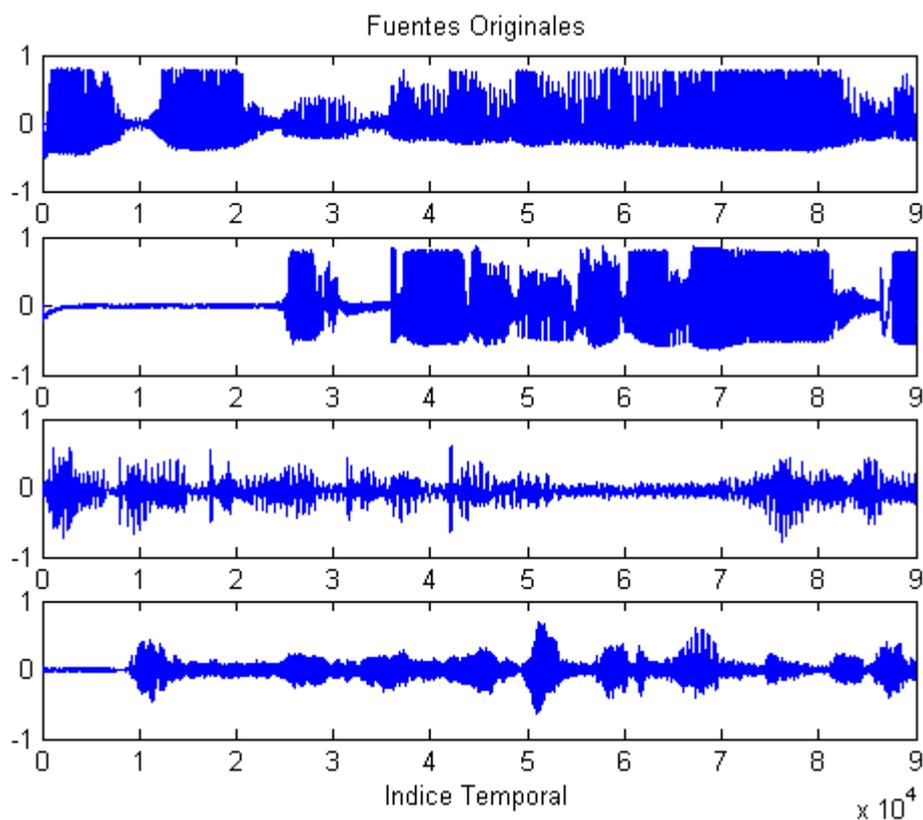


Figura 5.18: Fuentes Originales

5.2.3 Obtención de las derivadas de las fuentes originales

Al igual que en la versión anterior, la estimación de las derivadas, dentro de los límites admisibles para los retrasos, no es buena. Tal como se demostró anteriormente, las señales recibidas en el i -ésimo receptor pueden expresarse de la forma:

$$\begin{aligned}
 x_i(t) \approx & c_{i,1}s_1(t) - c_{i,1}\tau_{i,1} \frac{ds_1(t)}{dt} + c_{i,2}s_2(t) - c_{i,2}\tau_{i,2} \frac{ds_2(t)}{dt} + \dots \\
 & \dots + c_{i,N}s_N(t) - c_{i,N}\tau_{i,N} \frac{ds_N(t)}{dt} + n_i(t)
 \end{aligned} \tag{5.18}$$

Si los retardos son pequeños (del orden de microsegundos en la simulación anterior), la amplitud de las fuentes originales será varios órdenes de magnitud menor que la de las derivadas, lo cual dificulta el buen funcionamiento del algoritmo, ya que la operación de deflación siempre deja una parte residual de la señal que queremos eliminar en la mezcla. Si las derivadas son muy pequeñas, estas porciones residuales de las señales de interés que previamente se han extraído y que aún permanecen en la mezcla pueden ser dominantes sobre ellas, lo cual hará que la estimación sea peor que en la versión básica del algoritmo. Además, si se añade ruido, tal como se ha hecho en todas las simulaciones, en muchos casos la energía de la señal de ruido puede ser mayor que la de las derivadas, por lo que el algoritmo confundirá el espacio de las derivadas con el de la señal de ruido.

Si los retardos introducidos están dentro de los límites, la operación de deflación nos permitirá discernir entre las derivadas y las fuentes de interés. Las N primeras fuentes extraídas serán las de interés y las últimas N fuentes se corresponderán con las derivadas. Generalmente sólo se extraerán las N primeras fuentes, puesto que las derivadas no interesan.

En el caso de que se introduzcan retardos muy por encima de los admisibles, es conveniente realizar la extracción de las $2N$ fuentes, puesto que la derivada de alguna fuente puede hacerse mayor que otra fuente distinta y, si sólo se extraen las N primeras fuentes, puede perderse la estimación de alguna de las señales de interés porque en su lugar se extraiga la derivada de otra fuente.

5.2.4 Comportamiento frente al ruido

5.2.4.1 Respuesta frente al ruido

Para analizar el comportamiento de la versión ampliada del algoritmo, se realizaron varias simulaciones en las que se fue variando la relación señal a ruido. Los

retardos introducidos estaban dentro de los límites admisibles y se mantuvieron constantes en las diversas simulaciones.

Al igual que en el estudio que se hizo del comportamiento frente al ruido en la versión básica del algoritmo, en las simulaciones se introdujo ruido blanco y gaussiano. Además, el nivel de ruido que se introdujo en todos los receptores fue el mismo, de varianza σ^2 , pues ésta es una de las hipótesis en las que se fundamenta el método de separación propuesto.

La estimación de la relación señal a ruido se hizo de la misma forma que en el apartado 5.1.4.1, la SNR se determinó de la siguiente forma:

$$SNR = \frac{E_{señales}}{E_{ruido}} \quad (5.19)$$

Siendo $E_{señales}$ el nivel de energía de las fuentes, que, por defecto, es el mismo para todas, y E_{ruido} es la energía de la señal de ruido introducida. Tal como se argumentó en el apartado 5.1.4.1, es una estimación que arroja resultados pesimistas. Además, en este caso, este hecho se acentúa debido a que la operación de deflación siempre deja un residuo, de la señal que queremos extraer, en la mezcla. Estas señales residuales que quedan en la mezcla, se comportarán como ruido adicional que dificultará la extracción de las señales siguientes.

Las simulaciones se realizaron, al igual que en la versión básica, con un número de fuentes comprendido entre 3 y 5 y se fue variando el número de filtros usados. En ellas se fue disminuyendo el nivel la SNR de forma progresiva para ver la robustez del algoritmo.

En general, los resultados obtenidos fueron similares a los de la versión básica del algoritmo. Pese al uso de varios filtros, la relación señal a ruido mínima a partir de la cual el algoritmo dejaba de funcionar correctamente, resultó ser aproximadamente la misma que en la versión básica. Sin embargo, la calidad de las señales de audio estimadas en presencia de ruido resultó ser mayor, ya que el efecto del ruido aparece mucho más mitigado. Esto es lógico, pues si se hace uso de varios filtros y éstos han sido elegidos de forma adecuada, la estimación de la matriz de separación debe ser más precisa. Los resultados obtenidos en las diversas simulaciones, al ir variando la SNR , fueron los siguientes:

- Para relaciones señal a ruido $SNR > 17dB$, el comportamiento es bastante acertado y las señales de audio son extraídas sin que se aprecie la influencia del ruido.

- Para relaciones en el rango $14dB < SNR < 17dB$, las señales son extraídas de forma correcta, sin embargo, al oírlas puede apreciarse la influencia de un cierto nivel de ruido de fondo. En esta versión del algoritmo, la influencia que se percibe del ruido es menor que en la anterior.

Para relaciones señal a ruido menores de 17dB, la operación de deflación puede fallar en alguna de las iteraciones del algoritmo, por lo que el algoritmo puede devolver dos o más estimaciones consecutivas de una misma señal. Si esto ocurre, ya las N primeras estimaciones no se corresponderán con las señales de interés y será preciso examinar todas las $2N$ fuentes estimadas. La identificación de las señales de interés será fácil, puesto que las N fuentes de interés se extraen correctamente, aunque no en orden.

- Si bajamos de una $SNR = 14dB$, el algoritmo dejará de comportarse correctamente y en las señales comenzarán a apreciarse pequeños lóbulos debido a las otras señales distintas a la de interés. Por lo que al escuchar las señales extraídas, se oirán fuentes superpuestas. Al igual que antes, puede ocurrir que algunas señales de interés estimadas no aparezca entre las N primeras fuentes devueltas por el algoritmo.

- Para $SNR < 10dB$, el algoritmo deja de funcionar y en las estimas se obtiene una superposición de las señales de interés.

A continuación se muestran los resultados obtenidos en una simulación con una $SNR = 15dB$, tres fuentes y tres filtros distintos. Para estas relaciones señal a ruido relativamente bajas, a veces, la operación de deflación falla en alguna de las iteraciones del algoritmo, tal como se muestra en la *figura 5.19*. Este fenómeno se verá acentuado

si una de las fuentes es claramente dominante frente al resto, es decir, su energía es mucho mayor que la de las demás.

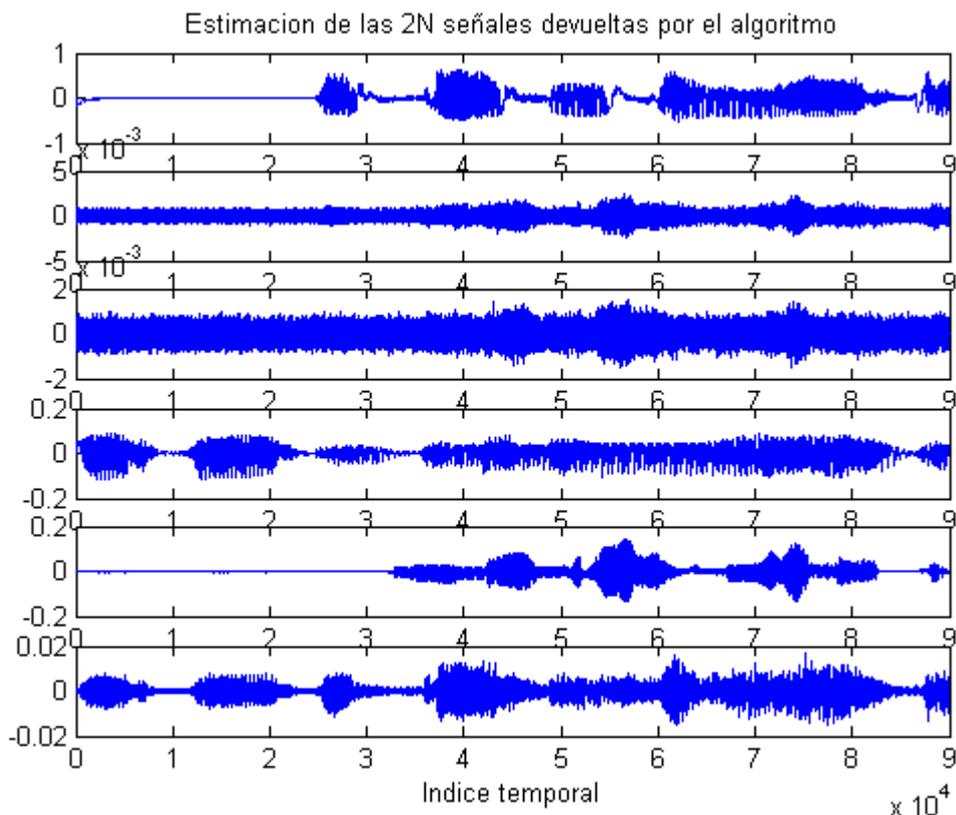


Figura 5.19: Salidas del algoritmo obtenidas al mezclar tres fuentes con una $SNR \approx 15dB$ y usando tres filtros distintos.

Si se observan las estimaciones realizadas por el algoritmo y se comparan con las fuentes originales mostradas en la *figura 5.20*, puede verse a simple vista que la primera estimación se corresponde con la segunda fuente, la cuarta estimación con la primera y la quinta estimación con la tercera fuente. Las N primeras estimaciones no se corresponden con las fuentes originales porque cuando se trabaja con relaciones señal a ruido bajas, a veces, la operación de deflación falla. En la *figura 5.19* puede verse cómo la operación de deflación de la primera fuente no se realiza de forma correcta dos veces consecutivas. Una vez que ésta se ha realizado completamente, el algoritmo continúa con la extracción del resto de fuentes, que ahora sí se realiza correctamente.

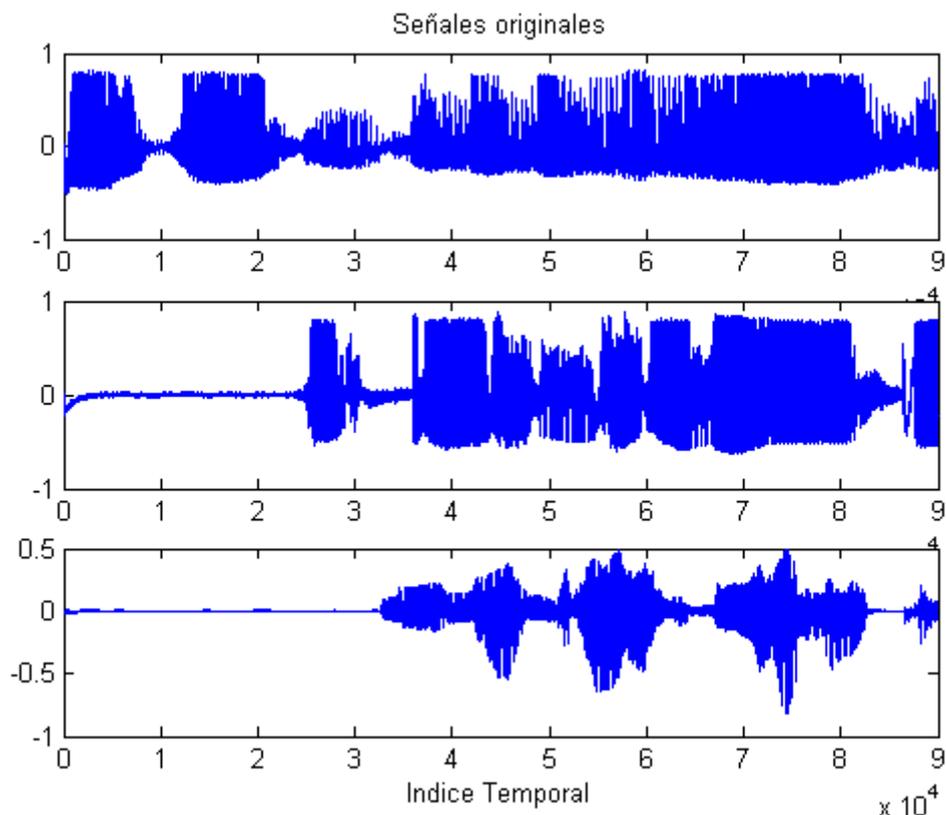


Figura 5.20: Fuentes originales utilizadas para la simulación de la *figura 5.19*

5.2.4.2 Comportamiento del algoritmo al variar el número de receptores

Los resultados obtenidos en este apartado fueron similares a los obtenidos en la versión básica del algoritmo. Tal como se comentó anteriormente, en general, la calidad de las señales extraídas mejora si se aumenta el número de receptores, puesto que la estimación de la varianza del ruido es más precisa. El número mínimo de receptores que puede usarse en presencia de ruido es de $M = 2N + 1$, por lo que habrá de elegirse un número mayor o igual a éste en las simulaciones.

En esta versión del algoritmo, es especialmente importante la elección correcta del número de receptores. Eligiéndolo de forma adecuada, puede obtenerse una mejora significativa del comportamiento del algoritmo frente al ruido. En el caso de que el número de receptores no se elija correctamente, tanto por exceso como por defecto, el rendimiento del algoritmo se verá mermado.

Si el número de receptores es escaso, el algoritmo se volverá mucho menos robusto frente al ruido. Al igual que en la versión básica, en las simulaciones se obtuvo que la

relación señal a ruido mínima para la que el ruido es imperceptible, con un número adecuado de receptores, es de $SNR_{\min} = 17dB$, mientras que si el número de receptores se reduce al mínimo, $SNR_{\min} = 27dB$, por lo que tomando un número suficientemente elevado de receptores, se obtiene una mejora de $10dB$.

Si el número de receptores es excesivo, el tiempo de respuesta del algoritmo aumentará de forma vertiginosa. Esto se debe a que para obtener las matrices de autocorrelación filtradas correspondientes a cada filtro, hemos de filtrar cada una de las filas de la matriz de autocorrelación de las observaciones $\mathbf{R}_{xx}(0)$ de dimensión $M \times M$ con el filtro que corresponda. Lógicamente si se hace uso de un número de filtros elevado, hemos de cuidar no aumentar excesivamente el número de receptores.

Además cuando el número de receptores es muy elevado, del orden $4N$, la operación de deflación, a veces, no se realiza de forma correcta, si una de las fuentes es dominante frente al resto, ya que si la señal no se elimina correctamente de la señal de mezcla de alguno de los receptores, puede ocurrir que el algoritmo realice dos extracciones de la misma señal.

A partir de las diversas simulaciones que se realizaron, un valor adecuado para el número de receptores resultó ser $M = M_{\min} + 2$, siendo $M_{\min} = 2N + 1$.

5.2.5 Elección de los filtros usados y estudio del comportamiento del algoritmo al variar el número de filtros

En esta implementación del algoritmo, se hace uso de varios filtros elegidos por el usuario para la obtención de varias matrices de autocovarianzas filtradas, en lugar de una como en la versión anterior. Cada una de estas matrices de autocovarianzas puede expresarse de la forma:

$$\mathbf{R}_{xx}^{h_i}(0) = \mathbf{A}\mathbf{D}_0^{h_i}\mathbf{A}^T + \sigma^2 h_i(0)\mathbf{I}_M \quad \text{Con } i = 1, \dots, N_{Fil} \quad (5.20)$$

Siendo $h_i(t)$ la respuesta impulsiva de cada uno de los filtros utilizados. Tal como se argumentó anteriormente, la respuesta impulsiva de los filtros debe ser nula en cero y así poder expresar la ecuación anterior de la forma:

$$\mathbf{R}_{xx}^{h_i}(0) = \mathbf{A}\mathbf{D}_0^{h_i}\mathbf{A}^T \quad (5.21)$$

En la versión anterior del algoritmo, la elección correcta del filtro a utilizar era clave. Así en las simulaciones realizadas se obtuvo que eligiendo el filtro de forma adecuada, puede obtenerse una relación señal a ruido mínima, para la que el ruido es imperceptible, del orden de $20dB$ menor que si el filtro se elige de forma arbitraria.

En esta versión la dependencia con los tipos de filtros escogidos no es tan grande. Si se escogen varios filtros de forma correcta y otros de forma arbitraria, el algoritmo devolverá resultados aceptables. Esto es así porque, tal como se explicó anteriormente, el algoritmo realiza un promedio entre todos los autovectores correspondientes a una misma fuente, que han sido obtenidos mediante el uso de diversas matrices de autocovarianzas filtradas. Por tanto, si alguno de los filtros se elige de forma incorrecta, no influirá tan negativamente como en la versión anterior.

Los filtros $h_i(t)$ escogidos en las simulaciones fueron similares al filtro utilizado en la versión básica del algoritmo, ya que de todos los filtros que se probaron, fue el que mejor comportamiento tuvo. Dicho filtro tenía una respuesta impulsiva de la forma:

$$h(t) = \delta(t-1) + \delta(t-2) + \dots + \delta(t-n_0) \quad (5.22)$$

Siendo n_0 un entero comprendido en el rango $[10,15]$.

Los filtros $h_i(t)$ utilizados y que dieron mejores resultados fueron conjuntos de filtros como el utilizado en la versión básica, con pequeñas diferencias, como retardos de algunas muestras o variaciones de su longitud en algunas muestras.

Con el propósito de determinar el número adecuado de filtros a utilizar, se realizaron varias simulaciones con una $SNR \approx 20dB$ en las que se fueron variando el número de filtros usados. El número de filtros que se fue incrementando desde 1 hasta 8. Todos los filtros que se utilizaron cumplían con los requisitos necesarios y fueron probados por separado para comprobar que su comportamiento era adecuado.

También se realizaron varias simulaciones en las mismas condiciones anteriores en las que se utilizó un conjunto de filtros elegidos de forma adecuada y uno o dos filtros con un funcionamiento inadecuado.

A partir de las diversas simulaciones, se obtuvieron las siguientes conclusiones:

- En general, si los filtros utilizados se han elegido correctamente, la calidad de las señales de audio es algo mayor que en la versión con un solo filtro, aunque estas diferencias no son demasiado significativas.
- Al aumentar el número de filtros, el comportamiento del algoritmo mejora si éstos han sido elegidos de forma correcta. Para conjuntos de filtros mayores de tres, las diferencias en el comportamiento no son significativas.
- Es preferible hacer uso de número reducido de filtros (uno o dos), que hayan sido elegidos de forma óptima, a hacer uso de un número elevado de filtros entre los que haya algunos con comportamiento mediocre.
- El tiempo de respuesta del algoritmo aumenta de forma considerable a medida que se introducen filtros, puesto que los procesos de filtrado son los que más recursos consumen.

5.2.6 Comportamiento frente a los retardos de las fuentes

Para estudiar el comportamiento frente al retardo, se realizaron varias simulaciones en las que se fue variando el número de retardos que sufrían las fuentes en su trayecto hacia los receptores. Se consideró la situación descrita en la *figura 5.1*, en la que las distintas fuentes procedentes desde un foco puntual se propagan hacia los receptores, separados pequeñas distancias entre sí. En la simulación se emplearon un total del $M = 8$ receptores para separar mezclas de $N = 3$ fuentes. Se utilizó una $SNR = 30dB$, para que el ruido no influyera mucho en el proceso de separación de las fuentes. El número de filtros utilizados fue de $N_{Fil} = 3$, todos ellos elegidos de forma adecuada. Por último, el resto de parámetros se mantuvieron de la misma forma que en el apartado 5.1.2.

Tal como se estableció en dicho apartado, los retardos máximos admisibles, bajo las condiciones que en él se especifican, son de 2 muestras según la condición (5.2).

A modo de resumen, los resultados obtenidos en las simulaciones fueron los siguientes:

- Para retrasos de hasta dos muestras (estamos dentro de los límites admisibles) el comportamiento del algoritmo fue el esperado.
- Para retrasos de tres muestras (ya estamos por encima de los límites del algoritmo) el comportamiento es bueno, aunque, a veces, las fuentes de interés no aparecen entre las N primeras devueltas por el algoritmo porque la operación de deflación no se realiza de forma correcta.
- Para retrasos desde 5 hasta 10 muestras, las fuentes son extraídas de forma correcta, sin embargo, a veces, el algoritmo omite alguna de ellas y en muchos casos, el orden de las estimaciones no es el esperado.
- A partir de retrasos de 13 muestras, el algoritmo deja de funcionar y sólo se obtienen mezclas de las fuentes de interés. A veces, el algoritmo realiza correctamente la estima de una fuente (la de mayor energía).
- Para los otros dos esquemas de simulación propuestos, los resultados obtenidos fueron los mismos que con la versión básica.

A partir de los resultados obtenidos, podemos decir que el comportamiento frente a los retardos es similar al de la versión básica. Al igual antes, la condición para los retardos máximos admisibles dada por (5.2) resultó ser demasiado restrictiva. Una expresión más próxima a la realidad para los retardos máximos que se pueden introducir, es la dada por la expresión empírica que se determinó en el apartado anterior:

$$\tau_{j_{\max}} \approx \frac{1}{\sqrt{10\pi} f_{\max}} \quad (5.23)$$

5.3 Comparación entre las dos versiones del algoritmo

A la vista de los resultados obtenidos en las simulaciones, el comportamiento de ambas versiones es similar.

La versión ampliada del algoritmo proporciona señales con algo de más calidad y no exige una elección óptima del filtro para proporcionar resultados aceptables. Además tiene la ventaja de permitirnos extraer sólo las señales de interés. Como desventaja su coste computacional es mucho más elevado.

La versión básica presenta resultados bastante similares a los de la versión ampliada si el filtro se escoge de forma adecuada y su tiempo de respuesta es mucho menor.

Comparando ambas versiones, puede decirse que la versión básica presenta resultados bastante similares a los de la versión avanzada, si el filtro se elige de forma adecuada (esto no supondrá un problema, puesto que ya se han descrito los tipos de filtro que proporcionan mejores resultados). Además los resultados son estimados en mucho menor tiempo, lo cual es una gran ventaja de cara a una posible implementación física de un sistema de separación en tiempo real que haga uso del algoritmo. La gran desventaja de la versión básica es que devuelve las señales de interés y sus derivadas mezcladas. La amplitud de las derivadas es despreciable frente a la de las fuentes, por lo que pueden idearse métodos basados en la energía de las señales para separar unas de otras. Por tanto, de cara a una posible implementación física, la versión básica, con una pequeña ampliación que nos permita separar las fuentes originales de sus derivadas, quizás sea la opción más acertada.

5.4 Resultados obtenidos al utilizar la expansión en serie de Taylor de segundo orden

5.4.1 Introducción

Tal como se expuso en el capítulo anterior, en el caso particular de que todas las fuentes sufran la misma atenuación en su propagación hacia los receptores, es posible realizar una optimización en el algoritmo, operando con las señales recibidas, que nos permite introducir retrasos mayores de los impuestos por la condición (5.1).

En el capítulo anterior, vimos que las observaciones en los diversos receptores distintos al de referencia con $i = 2, \dots, M$ pueden expresarse con respecto a la observación del sensor de referencia, con $i = 1$, de la forma:

$$x_i(t) \approx s_1(t) - \tau_{i,1} \frac{ds_1(t)}{dt} + \frac{\tau_{i,1}^2}{2} \frac{d^2s_1(t)}{dt^2} + s_2(t) - \tau_{i,2} \frac{ds_2(t)}{dt} + \frac{\tau_{i,2}^2}{2} \frac{d^2s_2(t)}{dt^2} + \dots \quad (5.24)$$

$$\dots + s_N(t) - \tau_{i,N} \frac{ds_N(t)}{dt} + \frac{\tau_{i,N}^2}{2} \frac{d^2s_N(t)}{dt^2} + n_i(t)$$

Y la observación en el sensor de referencia, $i = 1$, es igual a:

$$x_1(t) = s_1(t) + s_2(t) + \dots + s_N(t) + n_1(t) \quad (5.25)$$

Si a las observaciones en los receptores distintos al de referencia les restamos las observaciones del de referencia, tendremos $M - 1$ observaciones, con $i = 2, \dots, M$, cuyo valor aproximado es:

$$\tilde{x}_i(t) \approx -\tau_{i,1} \frac{ds_1(t)}{dt} + \frac{\tau_{i,1}^2}{2} \frac{d^2s_1(t)}{dt^2} - \tau_{i,2} \frac{ds_2(t)}{dt} + \frac{\tau_{i,2}^2}{2} \frac{d^2s_2(t)}{dt^2} + \dots \quad (5.26)$$

$$\dots - \tau_{i,N} \frac{ds_N(t)}{dt} + \frac{\tau_{i,N}^2}{2} \frac{d^2s_N(t)}{dt^2} + n_i(t)$$

La idea es aplicar el algoritmo de separación a las $M-1$ observaciones anteriores para obtener las derivadas de las fuentes originales. Con ello conseguimos que la condición de pequeños retardos, en este caso particular, pueda ponerse de la forma:

$$\tau_{i,j}^2 \ll \frac{3}{2\pi^2 v_M^2} \quad \forall i, j \quad (5.27)$$

Comparando la nueva condición para los retardos con la anterior dada en (5.1) podemos deducir que, operando con las mezclas de la forma y en las condiciones descritas, pueden introducirse retardos mayores de los que en condiciones normales el algoritmo es capaz de soportar. A costa de ello, el algoritmo devolverá las derivadas de las fuentes y se tendrá que emplear un receptor más. Esto no supondrá un problema en muchas aplicaciones y siempre será posible estimar las fuentes originales a partir de las derivadas mediante integración.

5.4.2 Simulaciones realizadas y resultados obtenidos

Para comprobar la validez del método de separación propuesto, se realizaron varias simulaciones, respetando las condiciones descritas en el apartado anterior. En dichas simulaciones, se mezclaron dos fuentes con el propósito de comparar los resultados obtenidos mediante simulación con los resultados obtenidos experimentalmente por los desarrolladores del algoritmo y descritos en [1], al mezclar también dos fuentes. Téngase en cuenta que tal como se demuestra en apéndice A.2, este procedimiento es incapaz de separar mezclas de más de tres fuentes, si se utiliza en una implementación física real.

En las simulaciones se mantuvieron el resto de condiciones comentadas en el apartado 5.1.2. Las mezclas se generaron introduciendo retardos en las señales recibidas en los receptores distintos al de referencia y sumando simplemente las señales retardadas en cada receptor. Al contrario que en las simulaciones anteriores, ahora las señales retardadas no se multiplicaron por un coeficiente generado de forma aleatoria para modelar las distintas atenuaciones que sufren las fuentes en su propagación hacia los receptores, pues se asume que todas sufren la misma atenuación.

En este caso los retardos máximos admisibles vienen dados por la condición (5.27). Si en ella, imponemos que el término de la izquierda sea un orden de magnitud menor que el de la derecha, tendremos la siguiente expresión para los retardos máximos admisibles con respecto al receptor de referencia:

$$10\tau_{\max}^2 = \frac{3}{2\pi^2 v_M^2} \quad (5.28)$$

Despejando obtenemos que el retardo máximo es igual a:

$$\tau_{\max} = \frac{\sqrt{3}}{\sqrt{20\pi} f_{\max}} \quad (5.29)$$

En las condiciones del apartado 5.1.2, $f_{\max} = 2\text{KHz}$. Sustituyendo se obtiene que el retardo máximo admisible es igual a:

$$\tau_{\max} = 6.164 \cdot 10^{-5} \text{ s} \quad (5.30)$$

Las señales de audio utilizadas, al igual que antes, fueron grabadas con una tasa de muestreo de 48KHz , por lo que el número máximo de retraso en muestras que podemos tener es de:

$$6.164 \cdot 10^{-5} \text{ s} \times \frac{48000 \text{ muestras}}{\text{s}} \approx 3 \text{ muestras} \quad (5.31)$$

Comparando con los retrasos máximos admisibles que se obtuvieron en el apartado 5.1.2, vemos que en este caso, particular pueden obtenerse retardos $\sqrt{3}$ mayores.

En las diversas simulaciones realizadas, se fueron ampliando el número de retardos entre las fuentes recibidas. Además se utilizó una $SNR = 30\text{dB}$. Los resultados obtenidos fueron bastante similares a los obtenidos experimentalmente por los desarrolladores del algoritmo en [1].

A continuación se muestran la estimación de las derivadas y las derivadas de las fuentes, para una simulación en la que se introdujeron retardos de hasta tres muestras. Comparando ambas gráficas, vemos que la calidad de las estimaciones es bastante buena en las condiciones descritas.

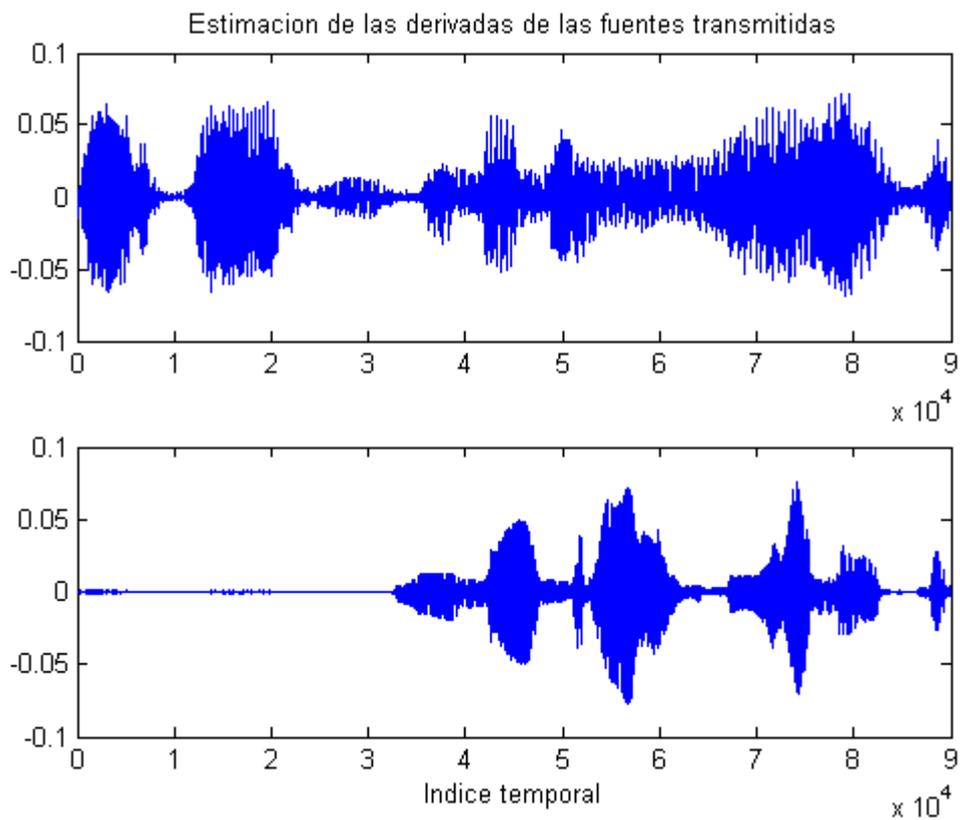


Figura 5.21: Estimación de las derivadas de las fuentes transmitidas

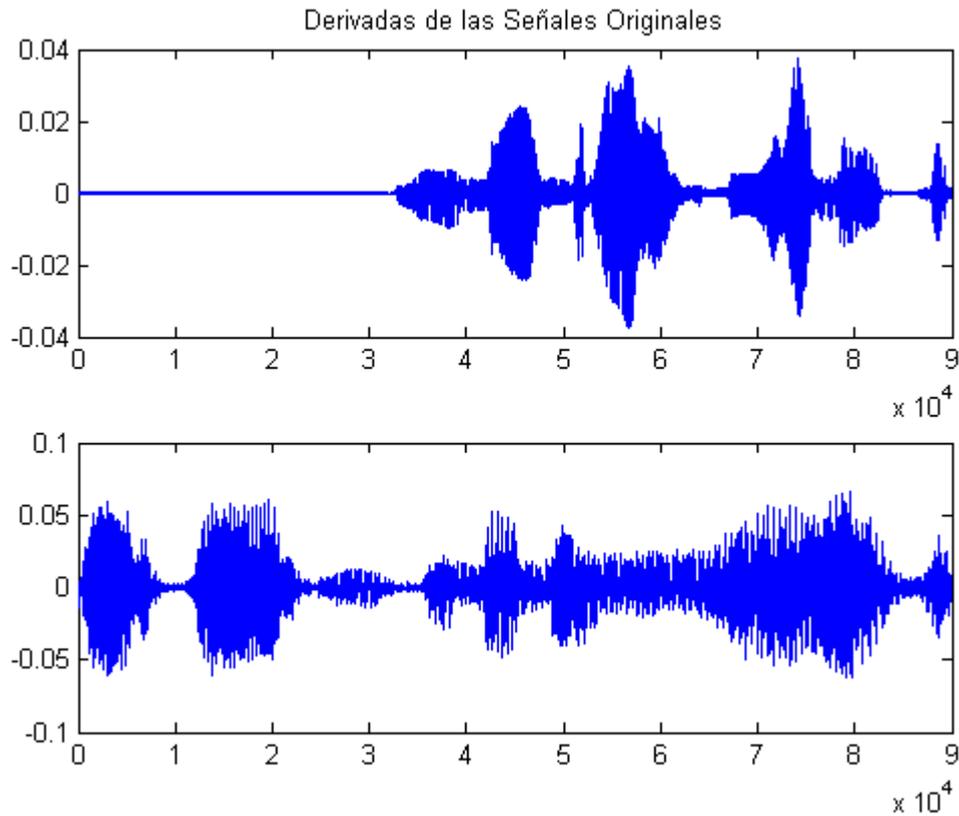


Figura 5.22: Derivadas de las fuentes transmitidas

En simulaciones sucesivas, se aumentaron poco a poco los retardos introducidos. Los resultados obtenidos fueron los siguientes:

- Para retardos de hasta 5 muestras (el límite teórico era de 3) los resultados obtenidos fueron aceptables.
- Para retardos de 6 a 8 muestras, empieza a observarse degradación en las señales estimadas. En ellas se aprecia la influencia de la otra señal que formaba parte de la mezcla.
- Para retardos por encima de las 10 muestras, el algoritmo deja de funcionar correctamente y sólo devuelve mezclas de las señales de interés.

A la hora de utilizar esta optimización del algoritmo, hemos de tener en cuenta que si se introducen retardos nulos o muy pequeños, las estimaciones no serán correctas puesto que tal como puede verse en la expresión (5.26), las señales de interés van multiplicadas por los retrasos introducidos. Si estos retrasos se hacen nulos o muy pequeños, en las observaciones dominará el término correspondiente al ruido y la separación no será posible. En las simulaciones realizadas con las condiciones descritas, a partir de retrasos de una muestra, se obtuvieron resultados aceptables.

Hemos de destacar que el comportamiento frente al ruido resultó ser mucho peor que en las versiones anteriores del algoritmo. La razón de esto es la misma de antes, las fuentes de interés están multiplicados por los retrasos que se introduce, que por definición, son pequeños. En la práctica para relaciones señal a ruido por debajo de los $25dB$, los resultados no son aceptables. Si se desea que las señales de audio no se vean muy degradadas, es conveniente trabajar con relaciones señal a ruido no inferiores a los $30dB$.

5.4.3 Conclusiones sobre el comportamiento del método de separación propuesto

A partir de los resultados obtenidos en las simulaciones, puede verse que el método propuesto puede ser una buena alternativa cuando se deseen introducir retardos algo mayores que los que soporta el método de separación general, el número de fuentes que se desean separar no es superior a tres y todas las fuentes sufran la misma atenuación en su propagación hacia los receptores.

El método tiene las limitaciones de que necesita funcionar con relaciones señal a ruido relativamente altas y de que no devuelve las señales de interés, sino sus derivadas. Aunque siempre se podrán estimar las fuentes originales integrando las estimaciones de las derivadas. En el caso de señales de voz, sus derivadas son perfectamente inteligibles, por lo que si no se desea una alta calidad en las estimaciones de las señales, podemos conformarnos con las estimaciones de las derivadas que devuelve el algoritmo.

Por último, se ha visto que en este caso la condición de pequeños retardos también resulta ser algo pesimista y puede trabajarse con retardos mayores de los previstos.

5.5 Resumen del capítulo y conclusiones

En el presente capítulo se ha estudiado el funcionamiento de una implementación en el lenguaje de programación MATLAB del algoritmo propuesto en el capítulo anterior para la separación de fuentes con retardos. Además de su correcto funcionamiento, se han comprobado la validez de las hipótesis y aproximaciones que se hicieron en el capítulo anterior. También se han recogido los resultados de numerosas simulaciones para analizar el comportamiento del algoritmo al variar los diversos parámetros que intervienen en él: número de receptores, nivel de ruido, magnitud de los retrasos, tipo de filtro empleado, etc.

Todos estos estudios se han repetido para la versión ampliada del algoritmo y, a partir de ellos, se ha hecho una comparativa entre ambas versiones en la que se han analizado las ventajas y desventajas de cada una frente a la otra.

Finalmente se ha estudiado un caso particular, en el que realizando ciertas operaciones con las observaciones, pueden obtenerse mejoras en el comportamiento frente a los retardos introducidos.

A modo de conclusión podemos decir que las hipótesis realizadas en el capítulo anterior son acertadas, puesto que los resultados obtenidos han sido los esperados. Además se ha visto cómo variando de forma adecuada ciertos parámetros, como el número de receptores, y eligiendo de forma adecuada el filtro auxiliar, el comportamiento del algoritmo mejora.