

5 Aplicaciones

En este capítulo se presentan algunas de las aplicaciones de la teoría del Compressive Sensing. En primer lugar se estudiará la reconstrucción de un canal de comunicaciones sparse empleando tanto el algoritmo Matching Pursuit como el Ortogonal Matching Pursuit. A continuación se utilizará la teoría del CS para la detección de sinusoides, se reconstruirá una señal suma de varias sinusoides a distintas frecuencias usando ambos algoritmos.

5.1 Identificación del canal en comunicaciones

Los canales con respuesta impulsiva sparse, como el representado en la figura 38, aparecen en un gran número de aplicaciones en comunicaciones. Aprovechando la escasez del canal, mostramos en este capítulo cómo se puede obtener un estimado del canal usando los algoritmos estudiados, Matching Pursuit y Orthogonal Matching Pursuit.

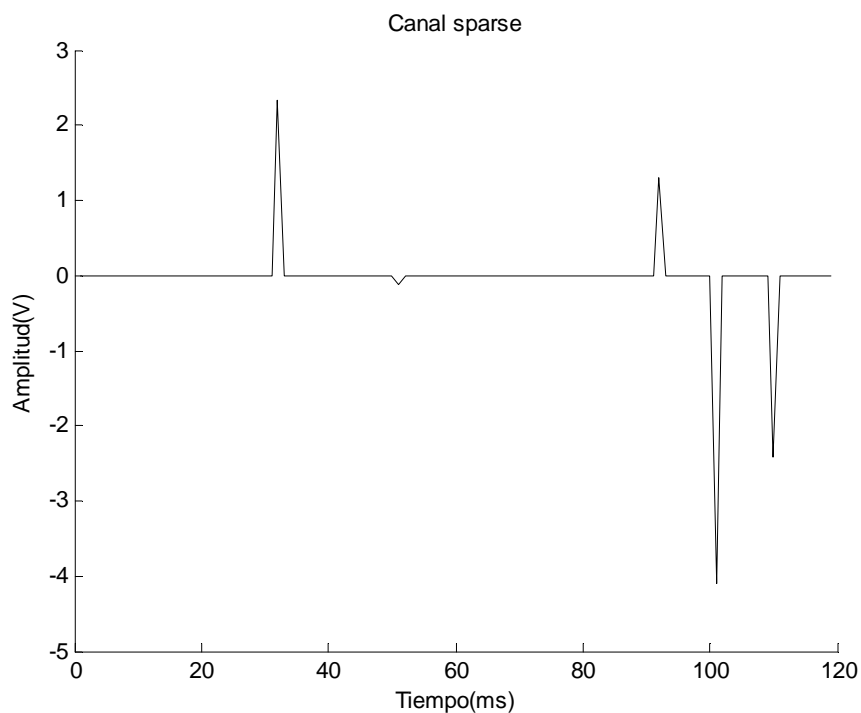


Fig38. Ejemplo de canal sparse de longitud $m = 119$ con sólo $k = 5$ muestras distintas de cero.

En primer lugar, se propone explotar la escasez natural del canal mediante el uso del algoritmo MP para obtener una estimación precisa del canal. Esta estimación obtenida a partir de MP es inherentemente sparse a diferencia de la estimación de mínimos cuadrados (LS) del canal donde cada uno de los valores de los pulsos serán generalmente distintos de cero.

En el algoritmo MP, como cada iteración de optimización se realiza sobre todos los vectores del diccionario, es posible reelegir un vector seleccionado en una iteración anterior. Este hecho disminuye la velocidad de convergencia hacia la solución sparse. Por eso, se propone el uso del algoritmo OMP para la estimación del canal. En OMP el problema de reelección se elimina ya que en cada iteración vamos almacenando el vector seleccionado, gracias a esto, OMP obtiene una velocidad de convergencia hacia la solución sparse mucho mayor.

5.1.1 Modelo del sistema.

El problema de la estimación de canales sparse se puede describir de la siguiente forma [18] y [19]: la secuencia de prueba $t(j)$ para $j = 0, 1, \dots, n-1$, se transmite a través del canal, y las muestras de la señal recibida son:

$$(81) \quad r(k) = \sum_{i=0}^{m-1} t(k-i)c(i) + n(k)$$

donde $c(i)$ representa la respuesta impulsiva del canal para $i = 1, \dots, m-1$. Asumimos que el canal es sparse, es decir, que $c(i) \neq 0$ para pocos valores de i . $n(k)$ representa el ruido blanco gaussiano aditivo de media cero y varianza σ_n^2 . La potencia de la señal de entrada y de la señal recibida se denotan por σ_t^2 y σ_r^2 , respectivamente. Los términos $t(j)$ con $j < 0$ se pueden tomar igual a cero si es el primer paquete recibido o se pueden obtener de una secuencia de prueba enviada anteriormente. Podemos escribir la ecuación anterior en forma de matrices:

$$(82) \quad \begin{bmatrix} t(0) & t(-1) & \dots & t(-m+1) \\ t(1) & t(0) & \dots & t(-m+2) \\ \vdots & \vdots & \vdots & \vdots \\ t(n-1) & t(n-2) & \dots & t(n-m) \end{bmatrix} \begin{bmatrix} c(0) \\ c(1) \\ \vdots \\ c(m-1) \end{bmatrix} = \begin{bmatrix} r(0) \\ r(1) \\ \vdots \\ r(n-1) \end{bmatrix} + \begin{bmatrix} n(0) \\ n(1) \\ \vdots \\ n(n-1) \end{bmatrix}$$

que puede abreviarse como:

$$(83) \quad Ac = r + n$$

Como el canal es sparse, la mayoría de las componentes del canal son cero. El problema consiste en aproximar el vector r en términos de combinación lineal de un pequeño número de columnas de la matriz A , es decir, debemos encontrar c tal que $Ac \approx r$.

Sabemos que para encontrar la solución sparse de (83) en los casos donde $m \ll n$, podemos aplicar los algoritmos de selección secuencial de base (SBS, Sequential Basis Selection) como el MP o el OMP. Aunque en el problema de estimación de canales sparse tenemos $m \geq n$, los algoritmos SBS se pueden aplicar para encontrar una solución sparse a la estimación del canal.

5.1.2 Simulaciones

En el artículo de *Karabulut y Yongacoglu [19]*, los dos algoritmos de estimación de canal sparse, MP y OMP, se comparan en términos de precisión de la estimación y rendimiento. Los algoritmos para la estimación de canales se obtienen sustituyendo en el MP y el OMP la señal original x por las muestras de la señal recibida r , es decir, $x = r$.

El canal sparse usado en la simulación se muestra en la Fig38. La longitud del canal es $m = 119$ y la de la señal de la secuencia de prueba la tomamos igual, es decir, $n = 119$, con el fin de hacer hincapié en el hecho de que los algoritmos de selección secuencial de base (SBS) trabajan bien con señales de prueba de pequeña longitud. Existen dos razones para considerar una secuencia de prueba corta. Primero, manteniendo n pequeño no se introduce una diferencia computacional considerable entre los dos algoritmos, MP y OMP. Segundo, una secuencia de prueba de longitud más corta mejora la eficiencia de procesamiento para los sistemas donde la longitud de los paquetes transmitidos es pequeña.

Primero comparamos la precisión de la estimación de los algoritmos MP y OMP. En las siguientes figuras se muestran los histogramas del número de componentes detectadas correctamente para valores de la relación señal a ruido de 5, 10 y 15 dB.

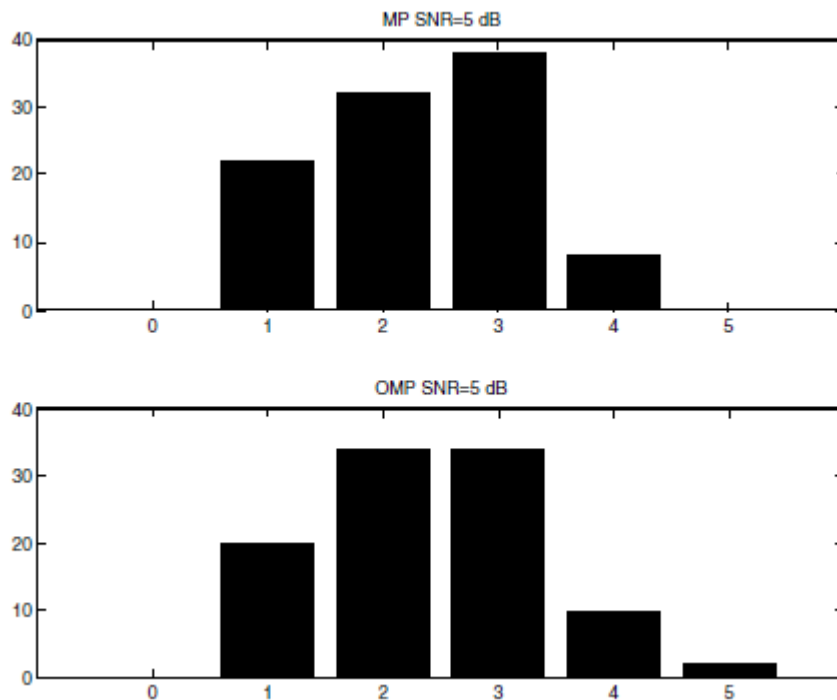


Fig39. Histograma del número de componentes detectadas correctamente con los algoritmos MP y OMP para SNR = 5dB

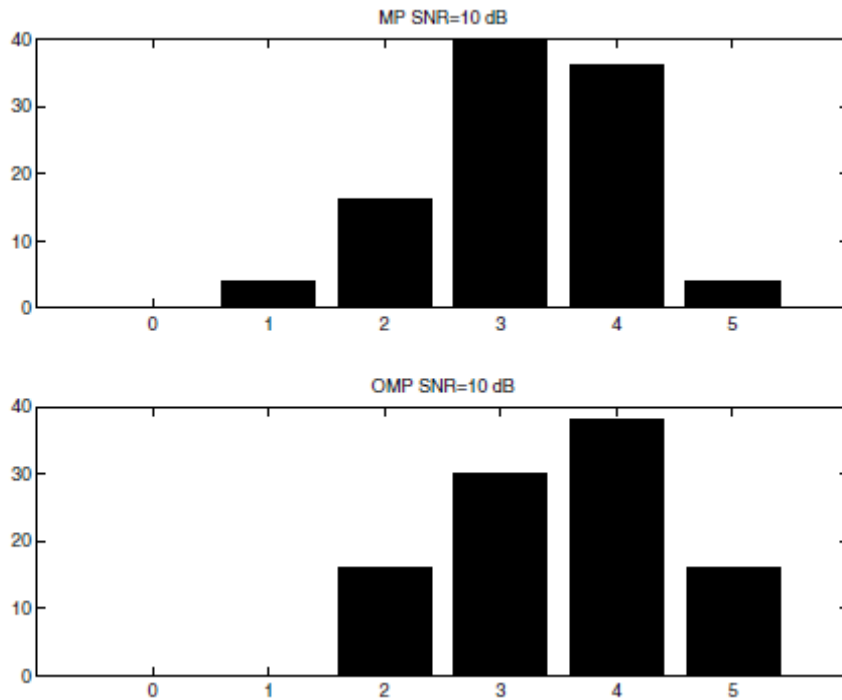


Fig40. Histograma del número de componentes detectadas correctamente con los algoritmos MP y OMP para SNR = 10 dB

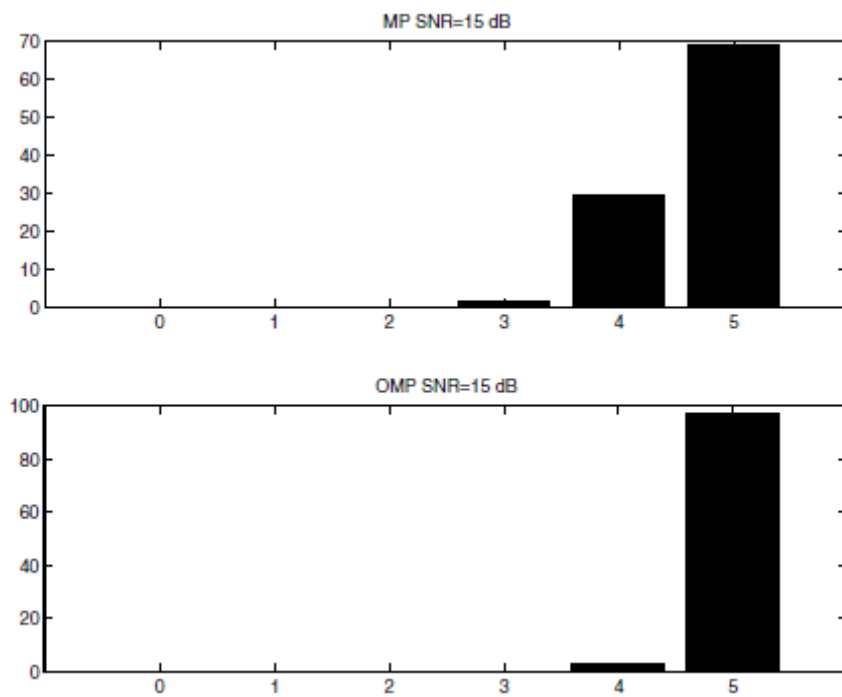


Fig41. Histograma del número de componentes detectadas correctamente con los algoritmos MP y OMP para SNR = 15 dB

Cada histograma muestra el porcentaje de veces que se recupera correctamente el conjunto de componentes (de 1 a 5 componentes) distintas de cero para un valor determinado de la SNR. De estos histogramas concluimos que el algoritmo OMP nos da una estimación más coherente y más precisa del canal que el algoritmo MP.

A modo de ejemplo se ha realizado la estimación de un canal sparse de longitud $m = 119$ con una señal de prueba de la misma longitud. Aunque en el problema de estimación de canales sparse tenemos $m \geq n$, ya vimos en el apartado (5.2) que los algoritmos MP y OMP se pueden aplicar para encontrar una solución sparse a la estimación del canal.

El canal a estimar se muestra en la figura 42:

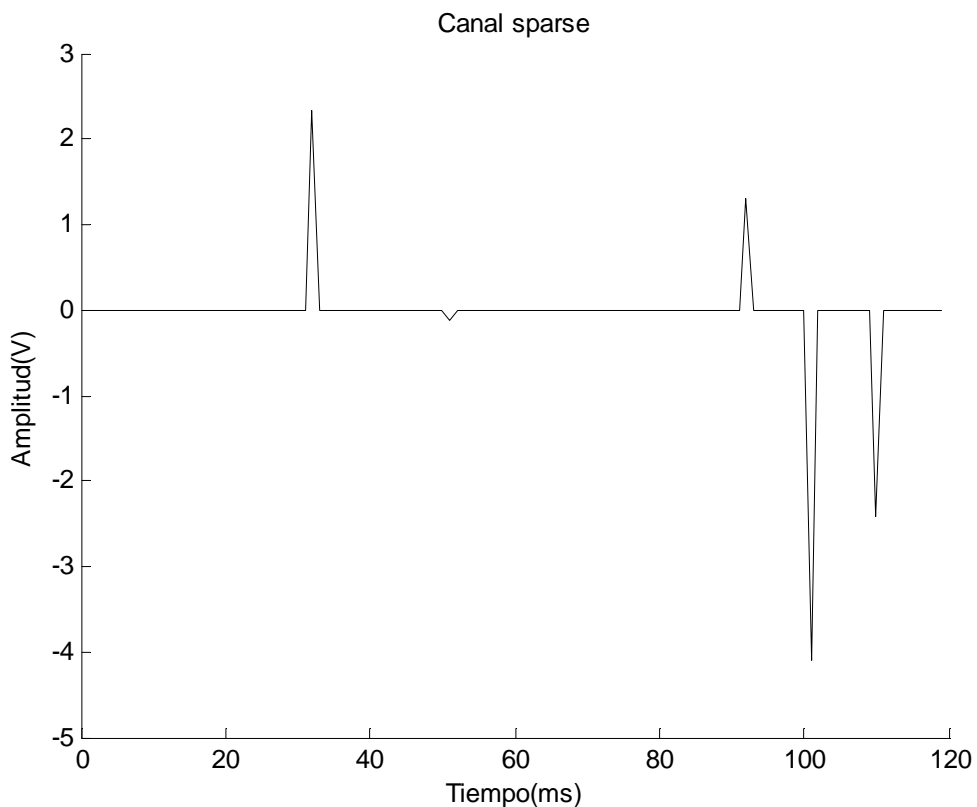


Fig42. Canal sparse de longitud $m = 119$ con sólo $k = 5$ muestras distintas de cero.

Para la simulación se han utilizado los algoritmos MP y OMP desarrollados en el apartado (3) estableciendo $r = x$, siendo r la secuencia recibida que se obtiene al transmitir la señal de prueba a través del canal sparse, ambas señales se muestran en la figura siguiente:

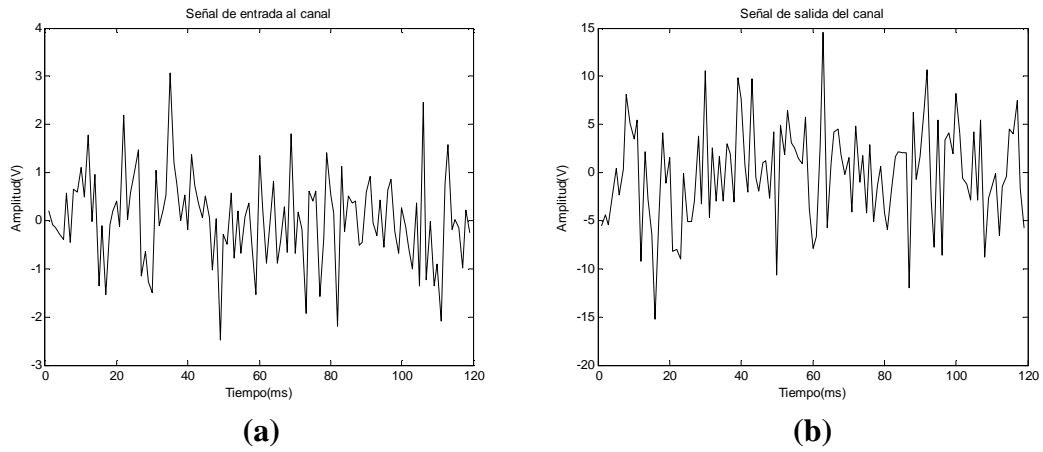


Fig43.(a) Señal de prueba enviada por el canal sparse. (b) Señal a la salida del canal sparse.

A continuación se muestra el canal estimado y el error cuadrático medio producido en la estimación del canal utilizando ambos algoritmos:

Algoritmo MP

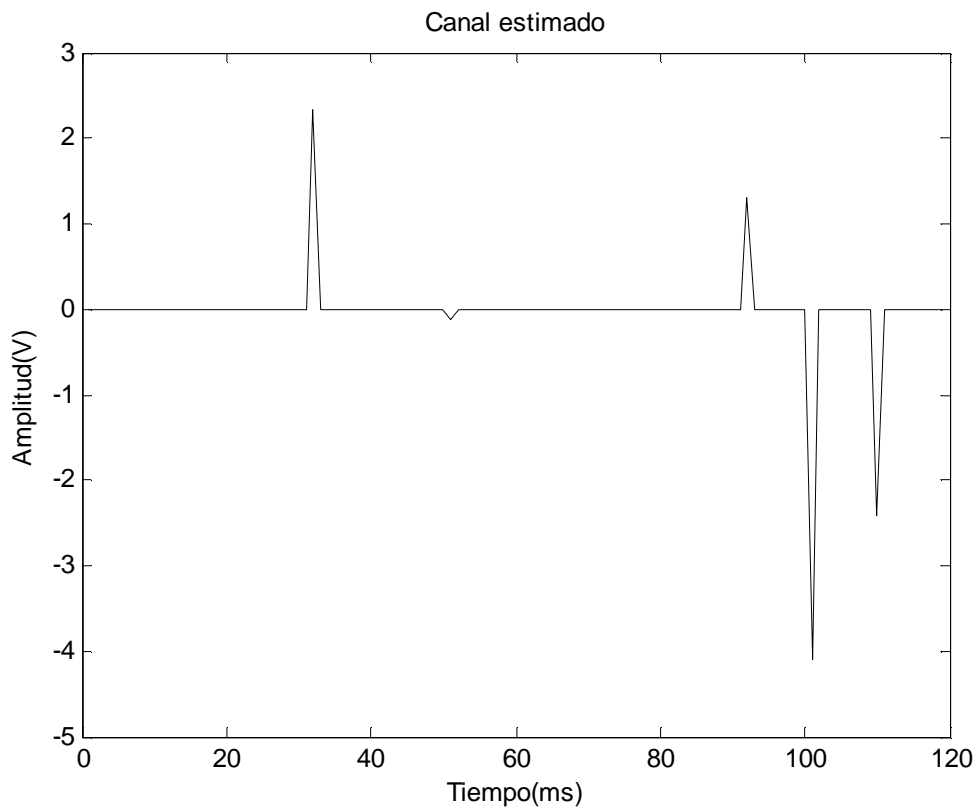


Fig44. Canal sparse estimado mediante el algoritmo Matching Pursuit.

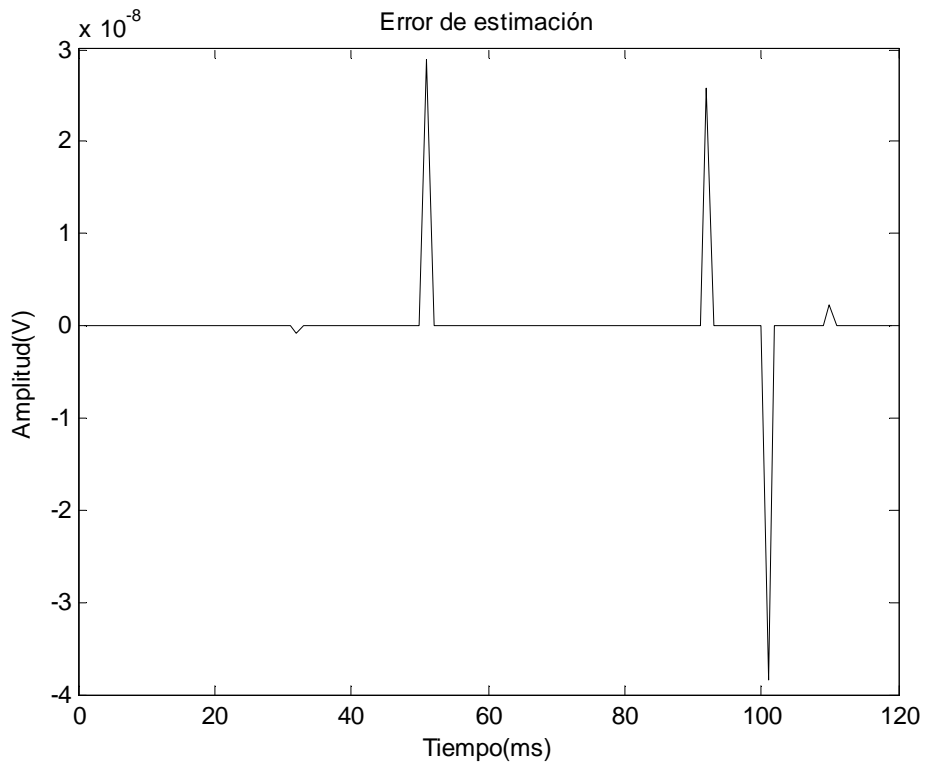


Fig45. Error cuadrático medio obtenido en la estimación del canal sparse empleando el algoritmo Matching Pursuit.

Algoritmo OMP

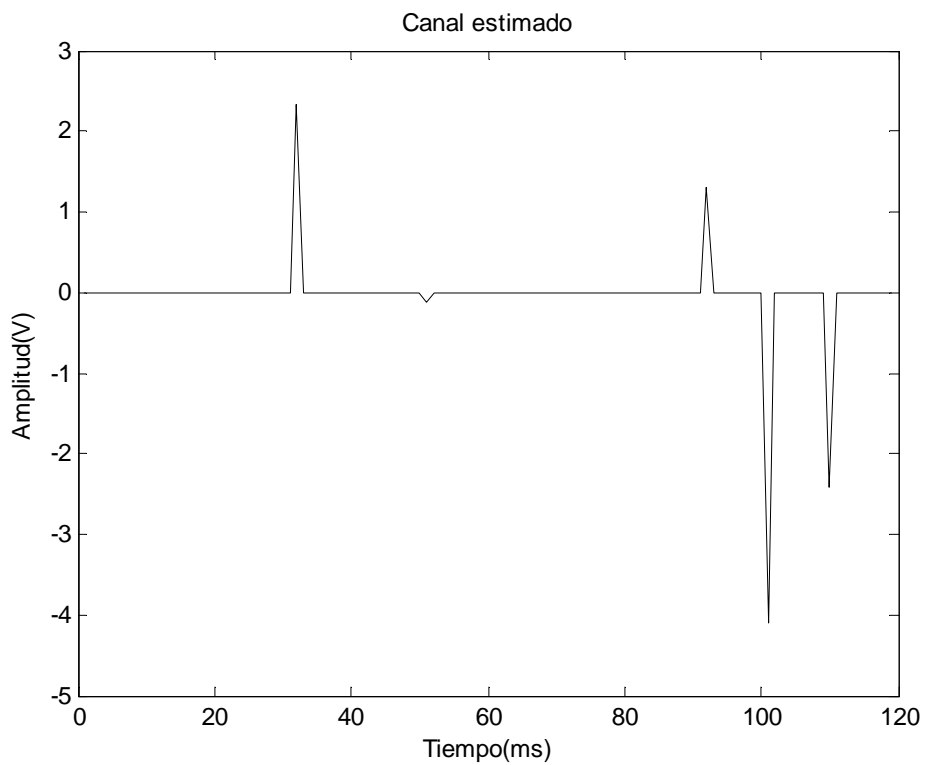


Fig46. Canal sparse estimado mediante el algoritmo Orthogonal Matching Pursuit.

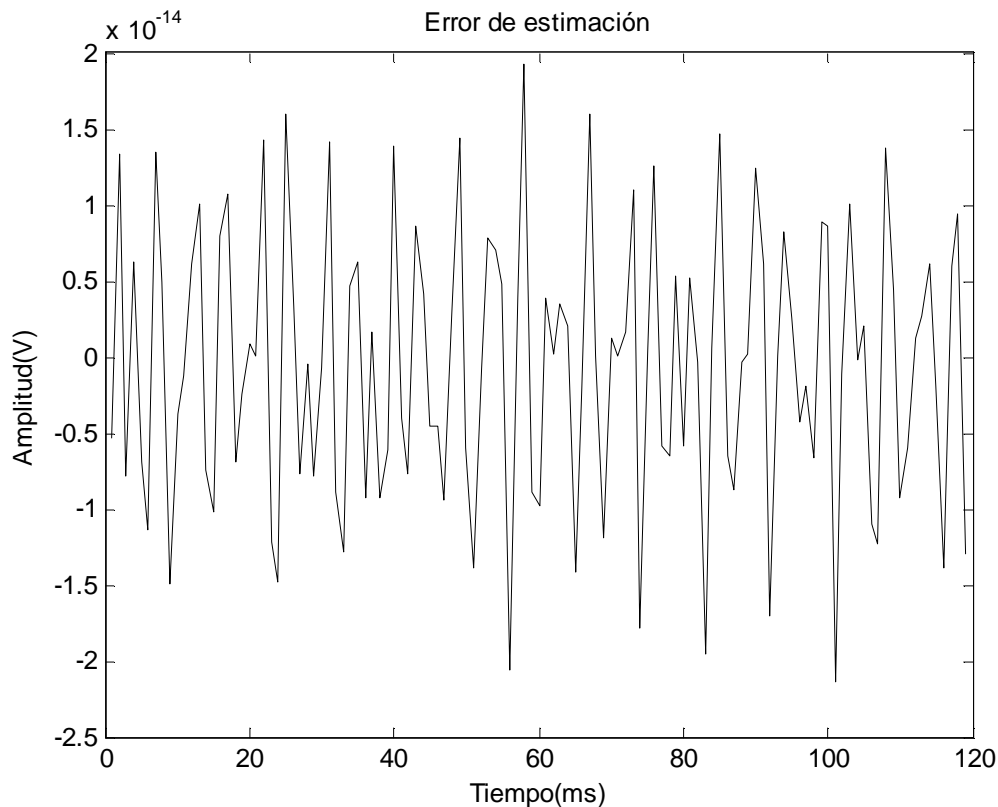


Fig47. Error cuadrático medio obtenido en la estimación del canal sparse empleando el algoritmo Orthogonal Matching Pursuit.

Algoritmo	Error cuadrático medio
MP	$2,9871 \times 10^{-15}$
OMP	$9,7812 \times 10^{-27}$

Tabla 6. Error cuadrático medio obtenido en la reconstrucción del canal sparse empleando los algoritmos MP y OMP.

Como podemos comprobar, los algoritmos MP y OMP se pueden emplear en la reconstrucción de un canal sparse a pesar de que en este caso no se cumpla $m \ll n$, si no que el número de mediciones m será mayor o igual a la longitud de la señal n .

Para comprobar el efecto del ruido gaussiano blanco aditivo que aparece en la mayoría de los canales de comunicaciones, se realizarán una serie de simulaciones de estimación del canal sparse con distintos niveles de relación señal a ruido empleando ambos algoritmos, MP y OMP. El canal sparse que se empleará en estas simulaciones tiene longitud $n = 119$ y escasez $k = 5$.

Se muestran en primer lugar las simulaciones realizadas con el algoritmo Matching Pursuit (MP).

SNR = 5dB

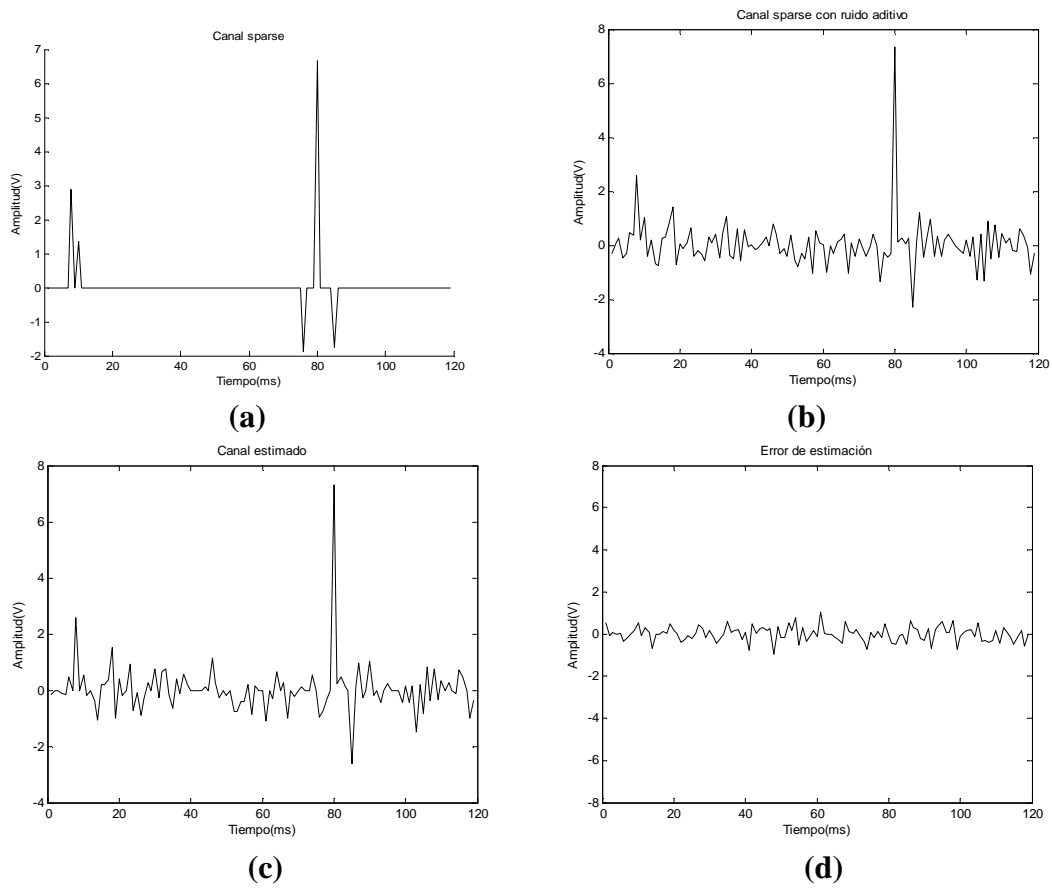


Fig48. Estimación del canal sparse con ruido blanco gaussiano aditivo con SNR = 5dB utilizando el algoritmo MP. **(a)** Canal sparse. **(b)** Canal con ruido, SNR = 5dB. **(c)** Canal estimado a partir de MP. **(d)** Error de reconstrucción.

SNR = 20dB

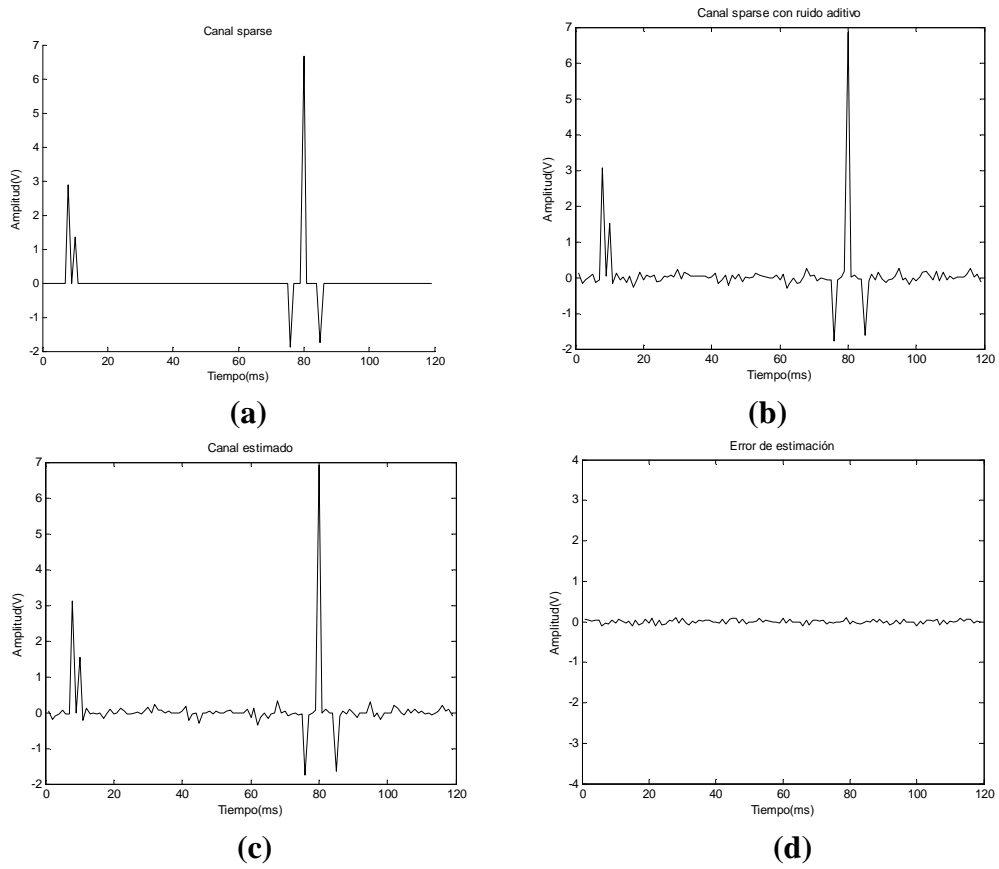


Fig49. Estimación del canal sparse con ruido blanco gaussiano aditivo con SNR = 20dB utilizando el algoritmo MP. **(a)** Canal sparse. **(b)** Canal con ruido, SNR = 20dB. **(c)** Canal estimado a partir de MP. **(d)** Error de reconstrucción.

SNR = 40dB

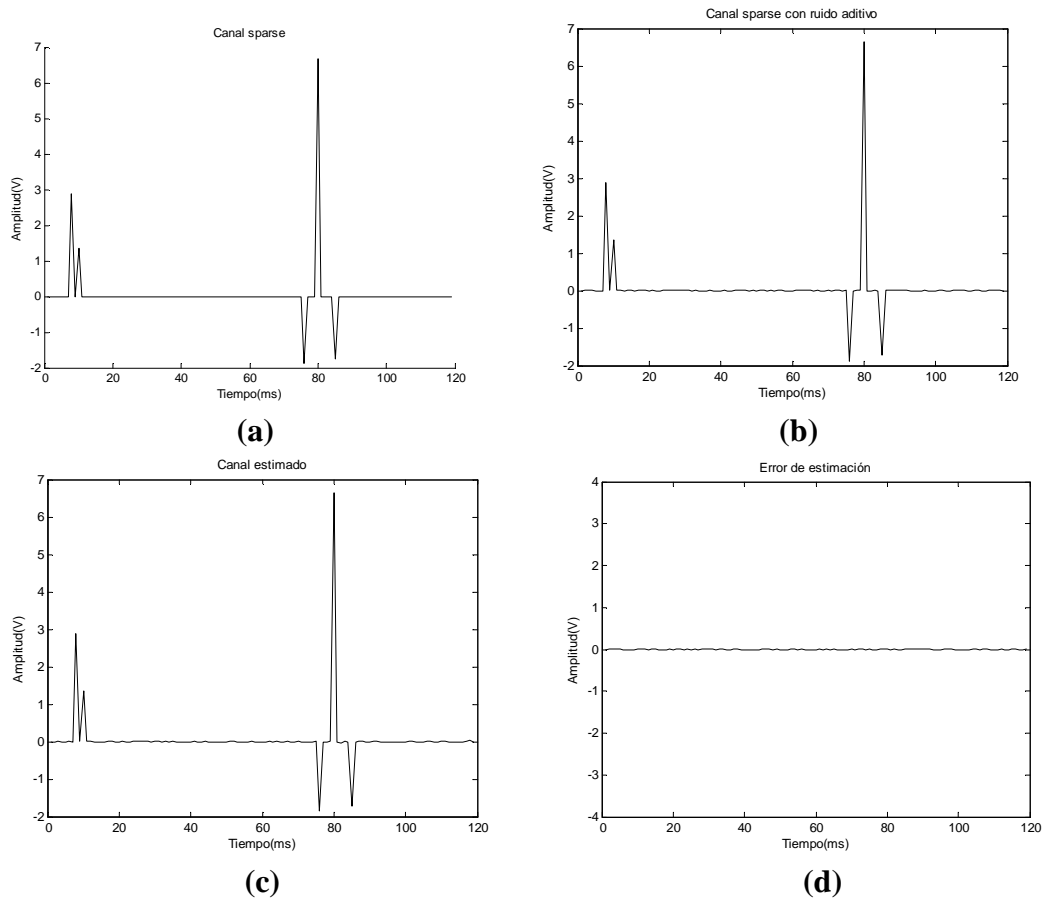


Fig50. Estimación del canal sparse con ruido blanco gaussiano aditivo con SNR = 40dB utilizando el algoritmo MP. **(a)** Canal sparse. **(b)** Canal con ruido, SNR = 40dB. **(c)** Canal estimado a partir de MP. **(d)** Error de reconstrucción.

Podemos comprobar que el error de estimación del canal sparse con ruido blanco gaussiano aditivo es bastante elevado, al igual que ocurría en la reconstrucción de señales sparse en el dominio del tiempo con este tipo de ruido. Se observa también que a medida que disminuye la potencia del ruido (es decir, aumenta la relación señal a ruido), disminuye el error de reconstrucción.

A continuación se realizan las mismas reconstrucciones empleando el algoritmo Orthogonal Matching Pursuit.

SNR = 5dB

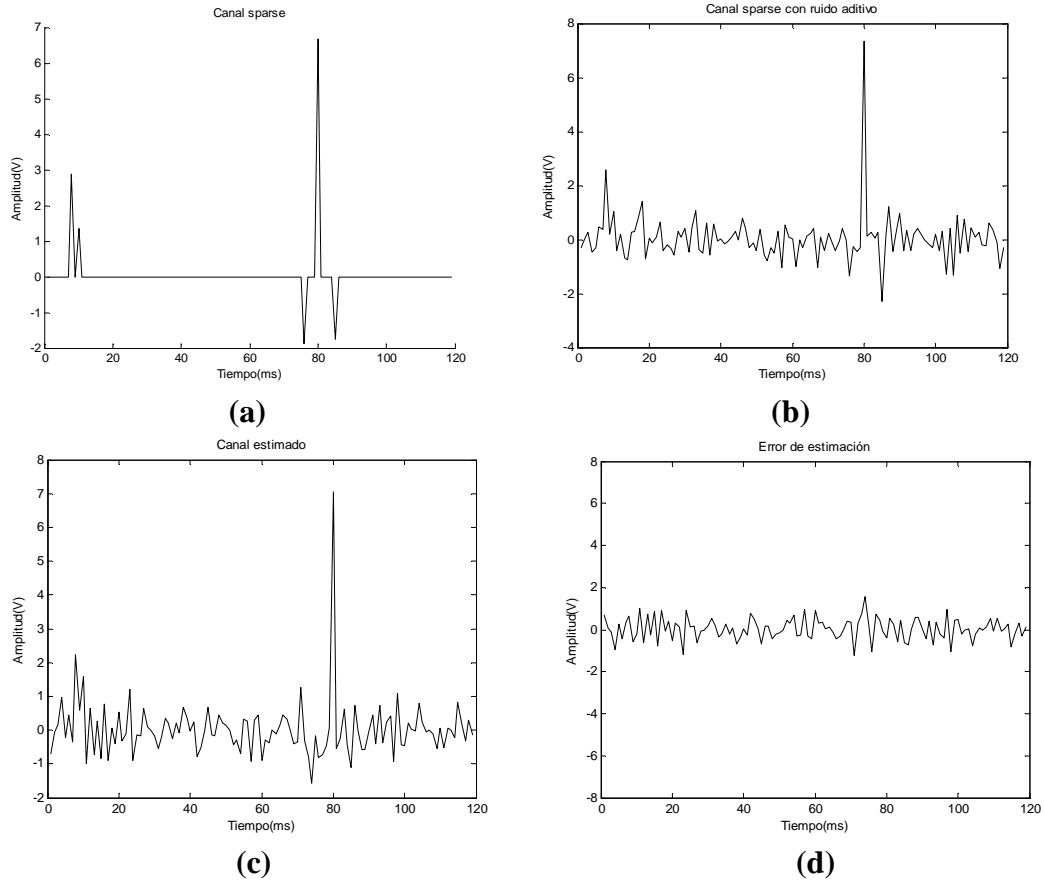


Fig51. Estimación del canal sparse con ruido blanco gaussiano aditivo con SNR = 5dB utilizando el algoritmo OMP. **(a)** Canal sparse. **(b)** Canal con ruido, SNR = 5dB. **(c)** Canal estimado a partir de OMP. **(d)** Error de reconstrucción.

SNR = 20dB

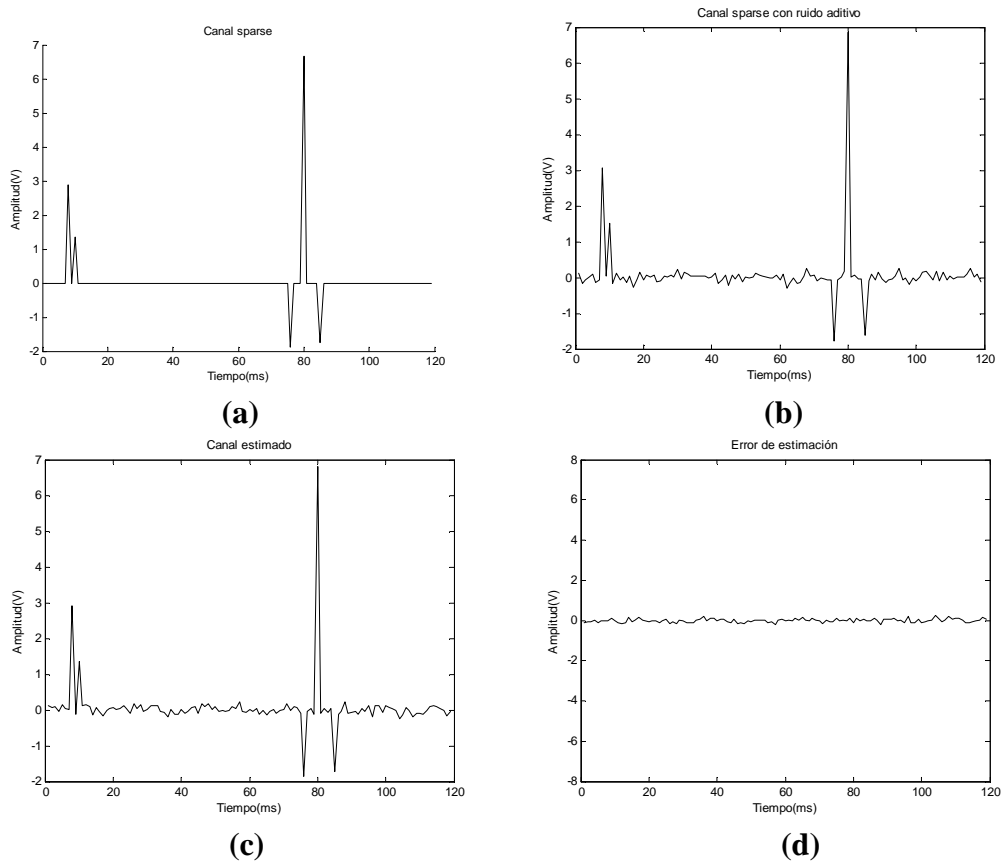


Fig52. Estimación del canal sparse con ruido blanco gaussiano aditivo con SNR = 20dB utilizando el algoritmo OMP. (a) Canal sparse.(b) Canal con ruido, SNR = 20dB. (c) Canal estimado a partir de OMP. (d) Error de reconstrucción.

SNR = 40dB

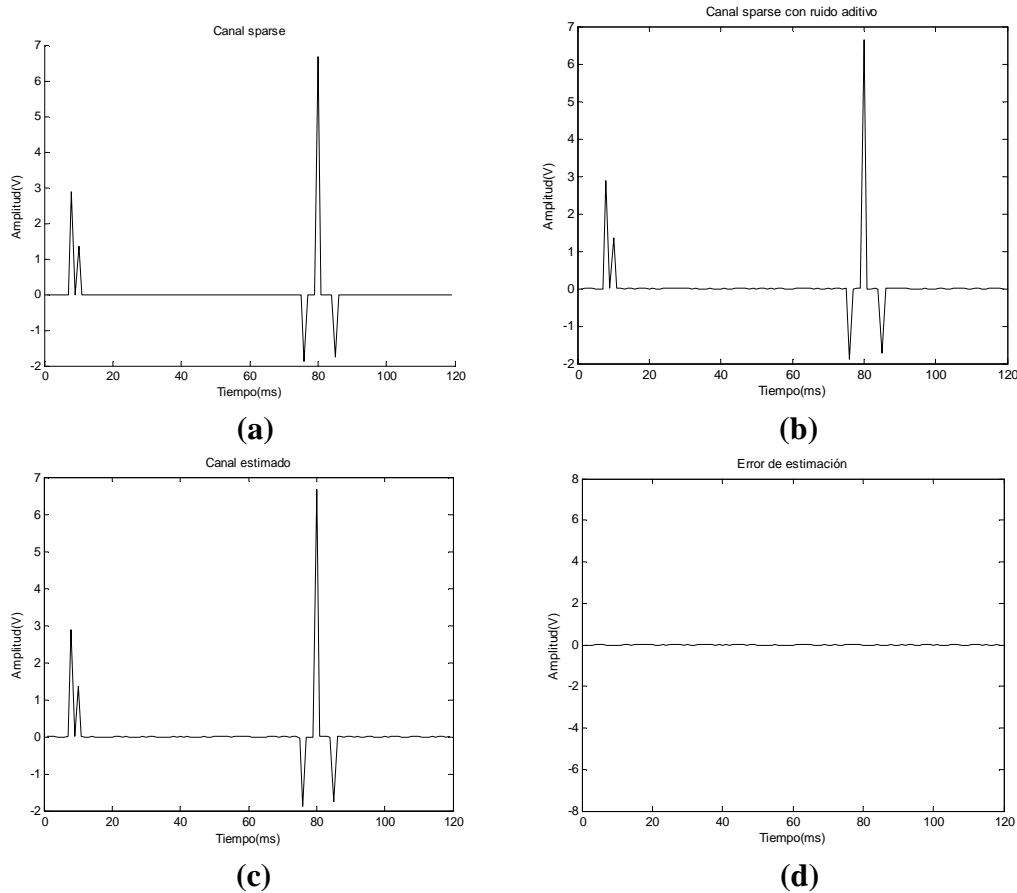


Fig53. Estimación del canal sparse con ruido blanco gaussiano aditivo con SNR = 40dB utilizando el algoritmo OMP. **(a)** Canal sparse. **(b)** Canal con ruido, SNR = 40dB. **(c)** Canal estimado a partir de OMP. **(d)** Error de reconstrucción.

Al igual que en el caso anterior, podemos comprobar que el error de estimación del canal sparse con ruido blanco gaussiano aditivo es bastante elevado, al igual que ocurría en la reconstrucción de señales sparse en el dominio del tiempo. Se observa también que a medida que disminuye la potencia del ruido (es decir, aumenta la relación señal a ruido), disminuye el error de reconstrucción.

5.2 Detección de sinusoides en frecuencia

En este capítulo se realizará una comparación entre el análisis frecuencial de señales basado en la DFT (Discrete Fourier Transform) y el análisis frecuencial basado en la teoría del Compressive Sensing, obteniendo el error producido por cada uno en la reconstrucción de una señal sparse en el dominio de la frecuencia. También se realizará un estudio comparativo a nivel de software del potencial y la capacidad de los dos algoritmos de reconstrucción estudiados, Matching Pursuit y Orthogonal Matching Pursuit, en la reconstrucción de señales sparse en el dominio de la frecuencia.

5.2.1 Análisis frecuencial de señales usando la DFT

La transformada discreta de Fourier (DFT), constituye una poderosa herramienta para el análisis frecuencial de señales en tiempo discreto. Una de las razones fundamentales de su importancia radica es la existencia de algoritmos eficientes para su cálculo, como la FFT.

Sea una secuencia $x(n)$. Definimos su transformada discreta de Fourier (DFT, Discrete Fourier Transform) de N puntos

$$(83) \quad X(k) = \sum_{n=0}^{N-1} x(n)W_N^{kn} \quad k = 0, 1, \dots, N-1$$

En principio no existe ninguna relación forzosa entre la duración de la secuencia $x(n)$ y el número de puntos N de su DFT. La longitud puede ser en principio mayor, menor o igual que N , concretamente:

- Si la longitud de $x(n)$ es mayor que N , se desprecian directamente las muestras que caigan fuera del rango del sumatorio, es decir, las situadas a partir de la posición N . Dichas muestras se pierden a todos los efectos.
- Si la longitud de $x(n)$ es menor que N , se tomarán tantas muestras nulas como sea necesario, rellenando con ceros el sumatorio hasta completar la cantidad N .

Debido a lo que acabamos de comentar, por defecto se entiende que el número de puntos de la DFT de $x(n)$, es decir, N , coincide exactamente con la longitud de dicha secuencia. Esto es así pues ese es el mínimo número de puntos para que la DFT represente unívocamente a la señal.

Los pasos a seguir son:

1. Se enventana la señal $x(n)$, es decir, se recorta para que tenga longitud finita. De no ser así, no podría aplicarse correctamente la DFT. El enventanado consiste en multiplicar por una ventana, es decir, por una secuencia $w(n)$ de duración finita N (esto es, es cero fuera del intervalo $0 \leq n \leq N-1$). Obtenemos así una secuencia enventanada $v(n)$, dada por:

$$(84) \quad v(n) = x(n)w(n)$$

Si queremos reducir $x(n)$ a un conjunto de N muestras, lo más inmediato es tomar

$$(85) \quad v(n) = \begin{cases} x(n), & 0 \leq n \leq N-1 \\ 0, & \text{en otro caso} \end{cases}$$

Esto equivaldría a tomar lo que se denomina una ventana rectangular, esto es,

$$(86) \quad w(n) = \begin{cases} 1, & 0 \leq n \leq N-1 \\ 0, & \text{en otro caso} \end{cases} \quad \Leftrightarrow \quad W(e^{j\omega}) = \frac{\text{sen}(\omega N / 2)}{\text{sen}(\omega / 2)} e^{-j\omega(N-1)/2}$$

2. Se muestra el espectro a las frecuencias

$$(87) \quad w_k = \frac{2\pi}{M}k \quad k = 0, 1, \dots, M-1$$

siendo $M \geq N$, para que así la longitud de la DFT sea al menos igual a la de la secuencia en el tiempo.

Más concretamente, vemos qué ocurre con la DFT para una $x(n)$ sinusoidal. Sea

$$(88) \quad x(n) = A_0 \cos(\omega_0 n + \theta_0)$$

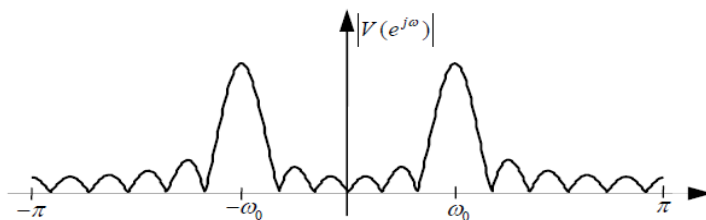


Fig54. DTFT de la señal sinusoidal enventanada.

Vemos en la figura 54 que incluso antes de tomar la DFT (o sea, antes del muestreo en frecuencia) se ha modificado el espectro de la señal, las frecuencias sufren lo que se denomina “leakage” (derrame).

Estudiamos qué sucede si $x(n)$ está formada por dos cosenos:

$$(89) \quad x(n) = A_0 \cos(\omega_0 n + \theta_0) + A_1 \cos(\omega_1 n + \theta_1)$$

Usando en los siguientes ejemplos una ventana rectangular, vamos a analizar en la figura 55 qué sucede si las amplitudes son comparables y las frecuencias están bien separadas.

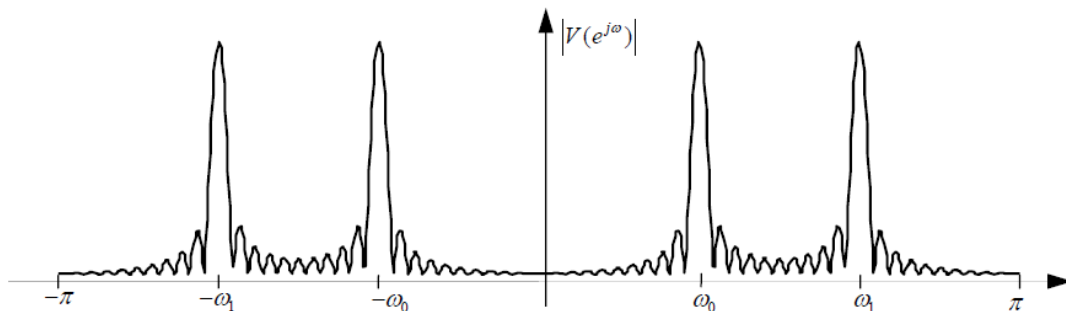


Fig55. DTFT de una señal suma de cosenos cuyas frecuencias w_0 y w_1 están bien separadas.

Aunque siempre se van a solapar los espectros de los cosenos, está claro que en este caso son perfectamente distinguibles. Decimos que se pueden resolver. Eso sí, puede cambiar algo la amplitud de los lóbulos, se producen lo que se llaman fugas.

En las figuras 56 y 57, vemos qué sucede si vamos aproximando ω_0 y ω_1 hasta que están muy cercanos. Vemos que llega un momento en que los cosenos ya no se pueden distinguir. Decimos que la resolución espectral ya no es suficiente.

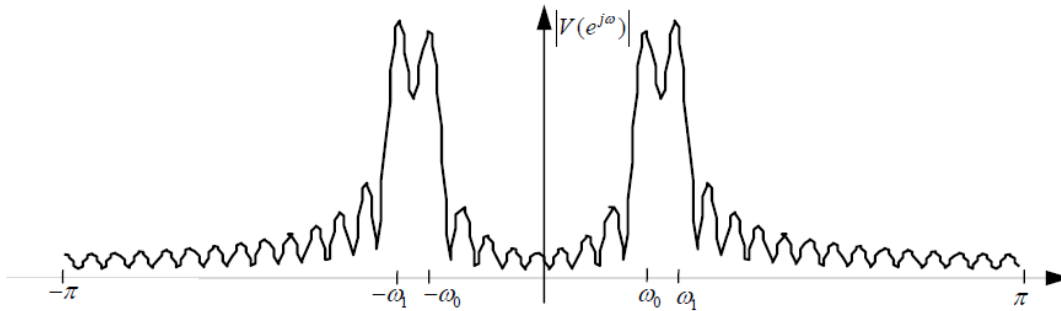


Fig56. DTFT de una señal suma de cosenos cuyas frecuencias ω_0 y ω_1 están muy próximas pero todavía se pueden distinguir.

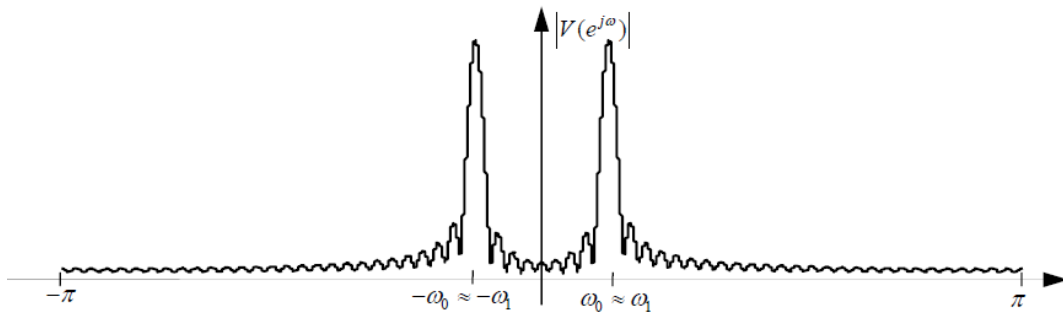


Fig57. DTFT de una señal suma de cosenos cuyas frecuencias ω_0 y ω_1 están muy próximas y ya no se pueden distinguir.

El límite inferior en la separación admisible entre frecuencias, es decir, la máxima resolución viene dada en el caso de la ventana rectangular por

$$(90) \quad \Delta\omega = \frac{2\pi}{M+1}$$

donde M es la longitud de la ventana.

La justificación es simple, los sumandos dejarán de poder resolverse cuando la frecuencia central de cada uno de ellos esté contenida en el lóbulo principal del otro.

Con el fin de comprobar este hecho, hemos usado la siguiente señal suma de cosenos a frecuencias próximas:

$$(91) \quad x = \cos(2 \cdot \pi \cdot 25 / N \cdot n) + \cos(2 \cdot \pi \cdot 26 / N \cdot n)$$

donde n es el tiempo discreto ($N = 0,1,2\dots n$) y N es el número de muestras de la señal x , $N = 1024$.

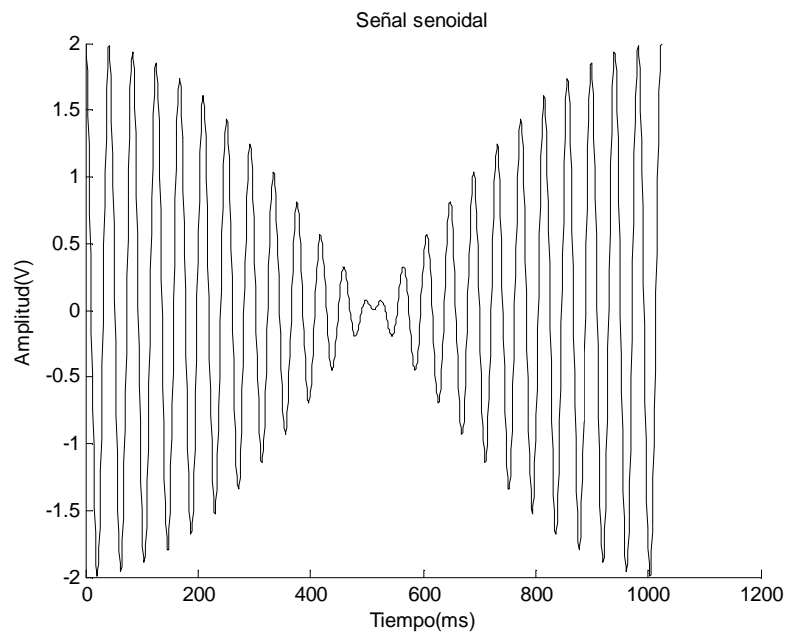


Fig58. Señal x sparse en el dominio de la frecuencia compuesta por la suma de dos cosenos a diferentes frecuencias (25 y 26 Hz). La escasez de esta señal es $k = 2$.

Hemos realizado su DFT de 1024 puntos haciendo uso de la función de Matlab `fft.m`. En la siguiente figura podemos ver las muestras de la DFT, que en este caso son indistinguibles debido a la cercanía en frecuencia de los cosenos que forman la señal.

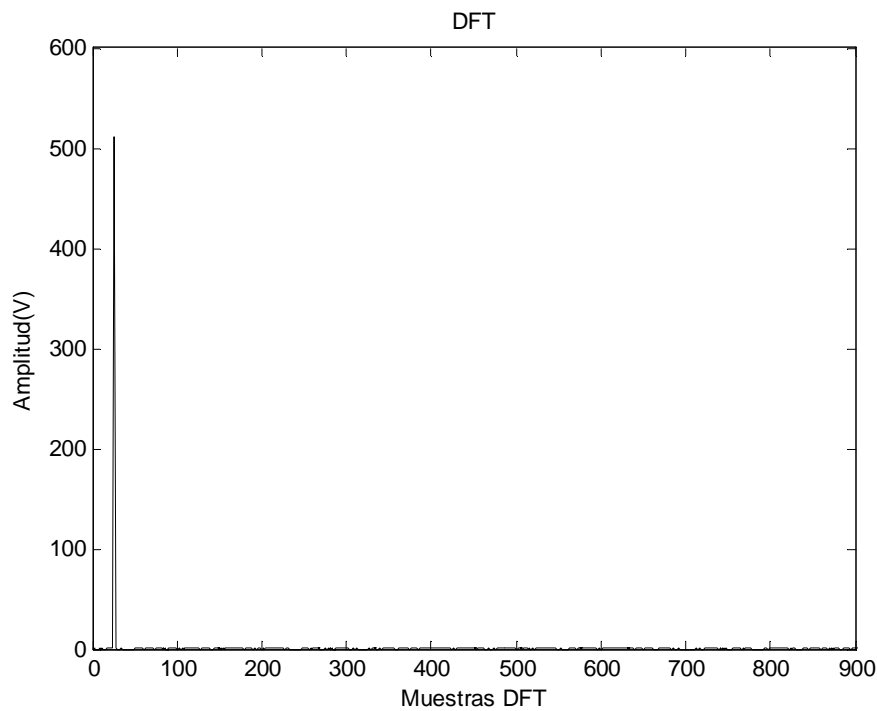


Fig59. DFT de la señal $x = \cos(2 \cdot \pi \cdot 25 / N \cdot n) + \cos(2 \cdot \pi \cdot 26 / N \cdot n)$, donde n es el tiempo discreto y N es el número de muestras de la señal x , $N = 1024$.

Como comprobamos en la figura anterior no se pueden distinguir los cosenos ya que en este caso la separación de frecuencias entre ellos es justo igual a la resolución:

$$(92) \quad \omega_1 - \omega_0 = 2\pi / n + 1 = 2\pi / 1024$$

5.2.2 Análisis frecuencial de señales usando la teoría CS

Emplearemos en este apartado los algoritmos iterativos desarrollados, Matching Pursuit y Orthogonal Matching Pursuit, para la reconstrucción de señales sparse en el dominio de la frecuencia. Como ya vimos anteriormente, el diccionario en el dominio de la frecuencia se genera a partir de la DFT; y será una colección de ondas sinusoidales $\Lambda_{(w,v)}$, donde $w \in [0, 2\pi]$ es la frecuencia angular, y v indica el tipo de fase, si es coseno o seno.

$$(93) \quad \Lambda_{(w,0)} = \cos(\omega t) \quad \Lambda_{(w,1)} = \sin(\omega t)$$

Ahora se definen las frecuencias para las que estarán definidos los cosenos:

$$(94) \quad \omega_k = 2\pi k / n \quad \text{con } k = 0, 1, 2, \dots, n/2$$

y los senos:

$$(95) \quad \omega_k = 2\pi k / n \quad \text{con } k = 0, 1, 2, \dots, (n/2) - 1$$

donde n es el número de columnas que tendrá el diccionario.

Se obtiene una matriz de n columnas, cada una de las cuales representa una función coseno o seno a cierta frecuencia.

Una señal k -sparse en algún dominio Ψ , puede ser definida a partir de una combinación lineal de k átomos provenientes del diccionario Ψ .

$$(96) \quad \mathbf{x} \approx \sum_{i=1}^k \Psi_{ni} s_{ni} = \Psi \mathbf{s}$$

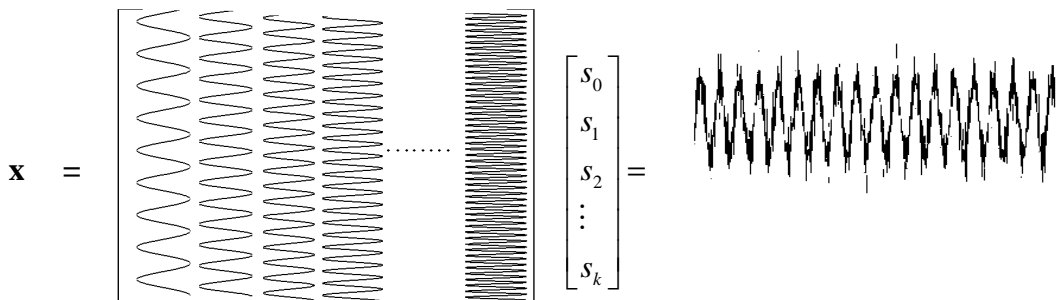


Fig60. Ejemplo de la definición de una señal sparse en el dominio de la frecuencia a partir de los átomos de un diccionario de Fourier, $\mathbf{x} = \Psi \mathbf{s}$.

Para nuestro ejemplo hemos elegido una señal sparse en el dominio de la frecuencia con nivel de escasez igual a dos, es decir, que está formada por la suma de dos cosenos a distintas frecuencias:

$$(97) \quad x = \cos(2 \cdot \pi \cdot 25 / N \cdot n) + \cos(2 \cdot \pi \cdot 26 / N \cdot n)$$

donde N es el tiempo discreto ($N = 0,1,2,\dots,n$) y n es el número de muestras de la señal x.

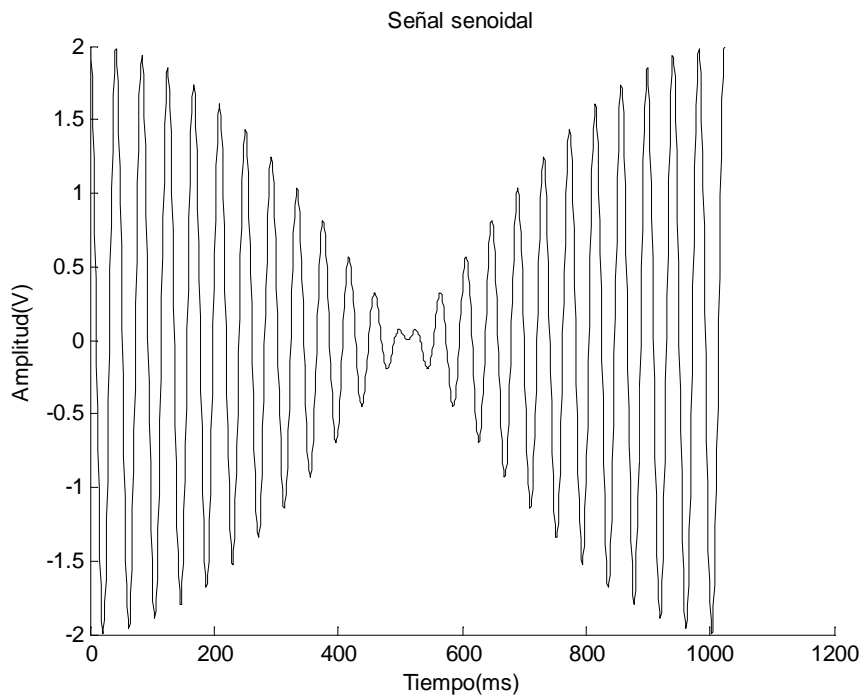


Fig61. Señal x sparse en el dominio de la frecuencia compuesta por la suma de dos cosenos a diferentes frecuencias (25 y 26 Hz). La escasez de esta señal es $k = 2$.

Como ya sabemos tanto MP como OMP son algoritmos iterativos que tratan de encontrar el átomo del diccionario (en este caso el diccionario de Fourier, formado por cosenos y senos) que aporta mayor contribución a la definición de la señal proyectada. En este caso encontrar esos átomos del diccionario, conlleva encontrar las frecuencias que están presentes en nuestra señal suma de cosenos, estas serán:

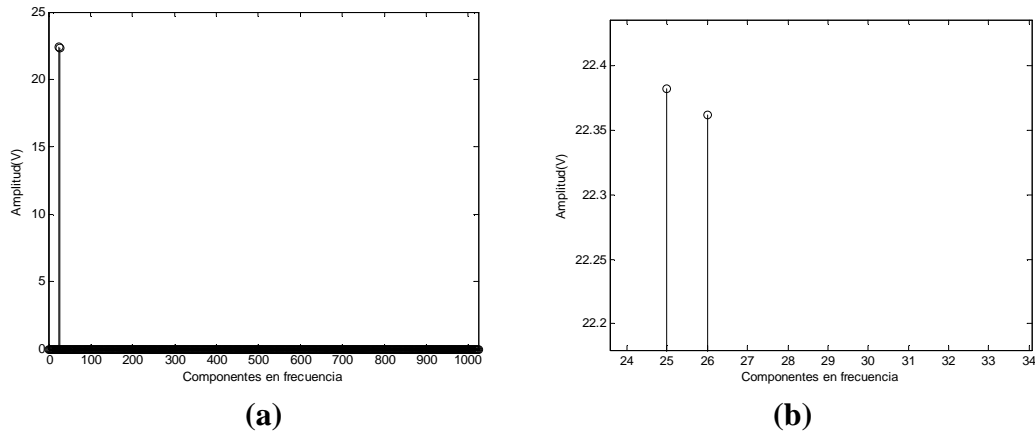


Fig62. (a) Frecuencias de los cosenos presentes en la señal 2-sparse en el dominio de la frecuencia obtenidas de la proyección sobre el diccionario de Fourier en los algoritmos MP y OMP. **(b)** Zoom de las frecuencias de los átomos del diccionario que aportan la mayor contribución a la definición de la señal.

En la figura 62 podemos observar que efectivamente los átomos del diccionario que aportan la mayor definición a la señal son aquellas ondas sinusoidales cuyas frecuencias coinciden con las frecuencias de los cosenos que conforman nuestra señal, es decir, 25 y 26 Hz. Además, comprobamos que aunque esas frecuencias estén bastante próximas, los algoritmos MP y OMP pueden distinguirlas perfectamente. (Se podría usar cualquier valor de frecuencias y el resultado de las simulaciones realizado en este capítulo no se vería alterado de forma significativa).

Tras encontrar el átomo del diccionario que aporta mayor contribución a la definición de la señal proyectada, estos algoritmos encuentran el estimado de la contribución de dicho átomo a la conformación de la señal y remueven dicha contribución que este aporta a la estructura del conjunto de mediciones de la señal, definiendo así un residuo. Se procede iterativamente hasta que el criterio de parada se cumpla.

En primer lugar se va a reconstruir la señal 2-sparse en el dominio de la frecuencia, a partir de los átomos del diccionario seleccionados, mediante el algoritmo Matching Pursuit. Se presentan a continuación la señal original, la señal recuperada por este algoritmo y el error producido en la reconstrucción.

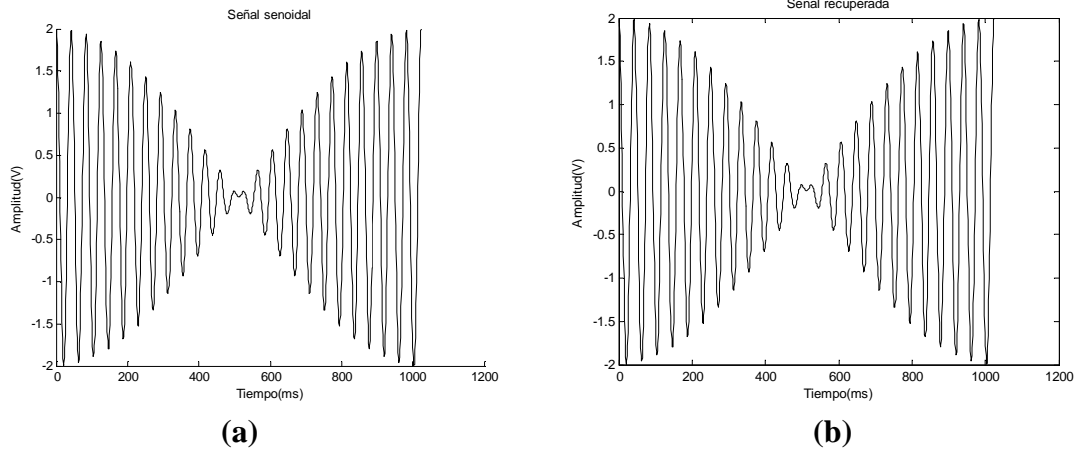


Fig63. Reconstrucción de señal sparse en el dominio de la frecuencia empleando el algoritmo MP. **(a)** Señal original. **(b)** Señal reconstruida.

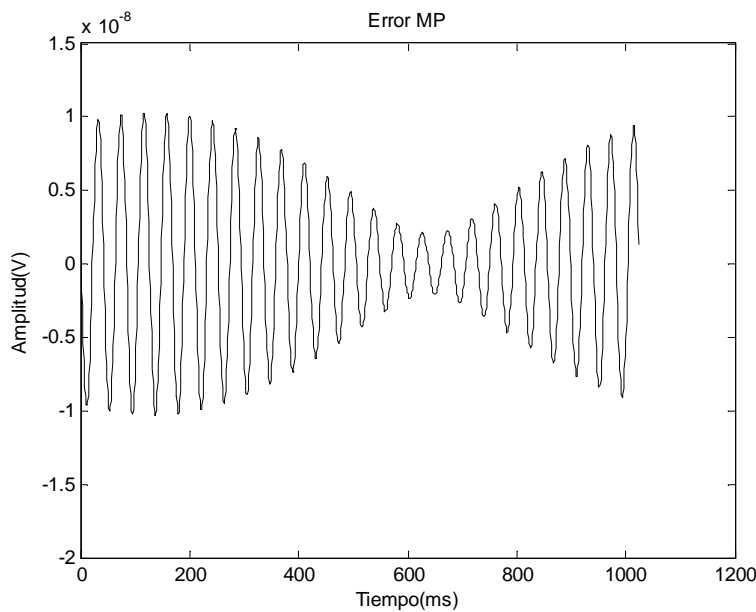


Fig64. Error en la reconstrucción de la señal sparse en el dominio de la frecuencia empleando el algoritmo MP.

Se observa en la figura que la diferencia entre la señal original y su reconstrucción utilizando MP es del orden de 10^{-8} en base a la amplitud. Si calculamos el error cuadrático medio como en los apartados anteriores (70), se obtiene $E_{mc} = 2,8285 \times 10^{-14}$, comparado con la amplitud original de la señal este error es prácticamente nulo ya que está diez órdenes de magnitud por debajo.

La señal fue reconstruida con un conjunto de tan sólo 50 muestras de las 1024 que conforman la señal original, lo cual implica una gran reducción en los datos necesarios para la reconstrucción casi perfecta de la señal.

Ahora se reconstruye la señal mostrada en la Figura 61 utilizando el algoritmo Orthogonal Matching Pursuit, la matriz de medición será la misma que se utilizó en la

reconstrucción con el algoritmo MP, con la finalidad de hacer la reconstrucción bajo exactamente las mismas condiciones de proyección que en el caso anterior.

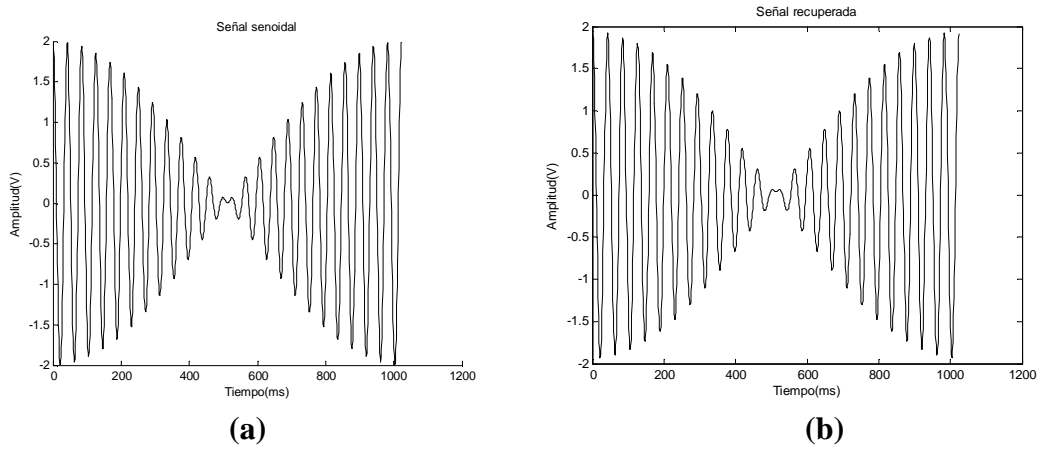


Fig65. Reconstrucción de señal sparse en el dominio de la frecuencia empleando el algoritmo OMP. **(a)** Señal original. **(b)** Señal reconstruida.

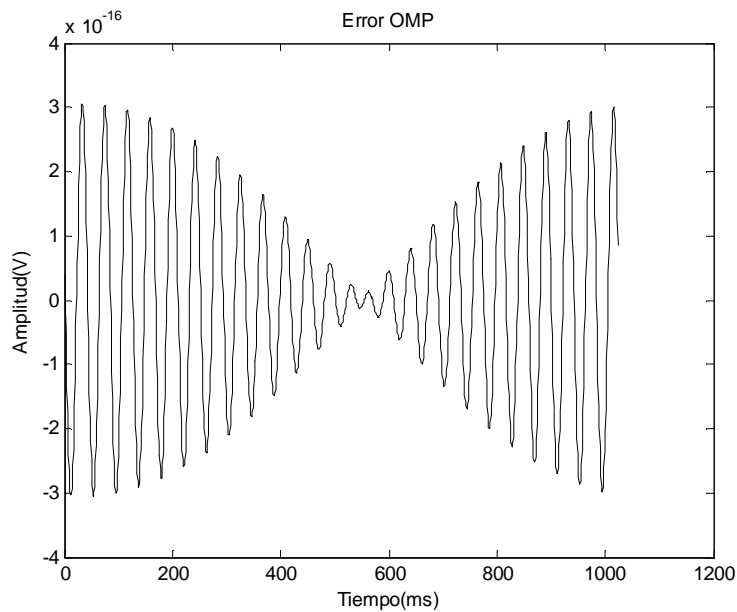


Fig66. Error en la reconstrucción de la señal sparse en el dominio de la frecuencia empleando el algoritmo OMP

La diferencia entre la señal reconstruida y la señal original es del orden de 10^{-16} en base a la amplitud, si calculamos el error cuadrático medio empleando la ecuación (70), se obtiene que $E_{mc} = 5,2111 \times 10^{-29}$.

En la Tabla 7 se muestra el error cuadrático medio que se obtuvo para cada uno de los métodos de reconstrucción al reconstruir mil veces la señal. Comparando los resultados obtenidos utilizando OMP con los obtenidos empleando el algoritmo MP, se puede apreciar fácilmente que el error medio cuadrático generado por el OMP es siete órdenes de magnitud menor que el generado usando MP.

Método de reconstrucción	Error cuadrático medio
MP	$1,23 \times 10^{-11}$
OMP	$2,46 \times 10^{-18}$

Tabla7. Comparación del error obtenido en la reconstrucción de una señal sparse en el dominio de la frecuencia usando ambos algoritmos.

Comprobamos a continuación el funcionamiento de ambos algoritmos si la señal sparse en el dominio de la frecuencia tiene un nivel de escasez mayor, es decir, si la señal está compuesta por la suma de un mayor número de cosenos. La señal empleada en estas simulaciones será:

(98)

$x = \cos(2 \cdot \pi \cdot 25 / N \cdot n) + \cos(2 \cdot \pi \cdot 26 / N \cdot n) + \cos(2 \cdot \pi \cdot 31 / N \cdot n) + \cos(2 \cdot \pi \cdot 32 / N \cdot n)$
 por lo que el nivel de escasez será $k = 4$.

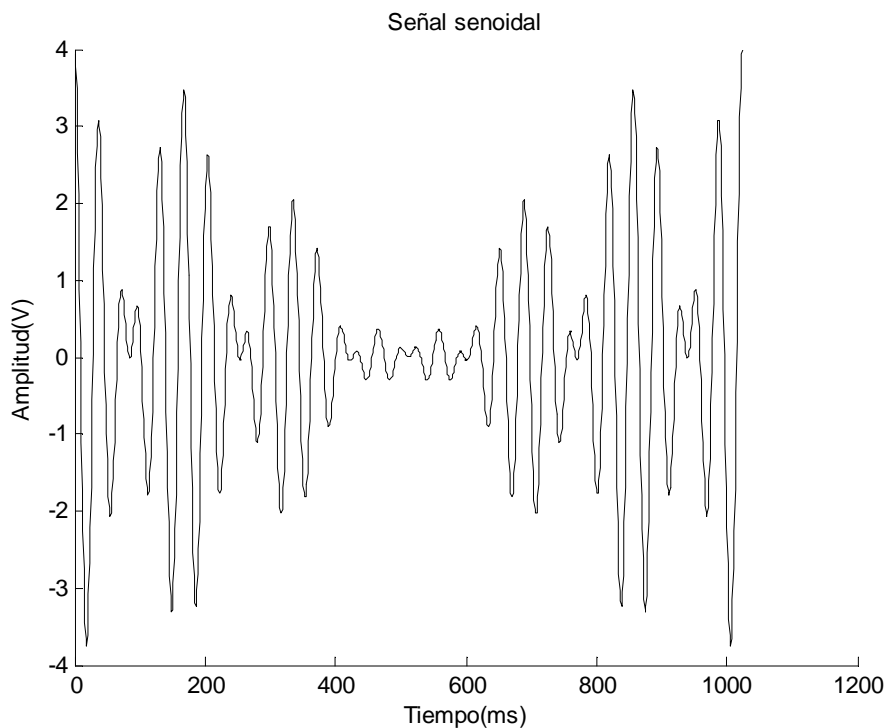


Fig67. Señal x sparse en el dominio de la frecuencia compuesta por la suma de dos cosenos a diferentes frecuencias (25, 26, 31 y 32 Hz). La escasez de esta señal es $k = 4$.

En primer lugar hemos realizado su DFT de 1024 puntos haciendo uso de la función de Matlab `fft.m`, obteniendo:

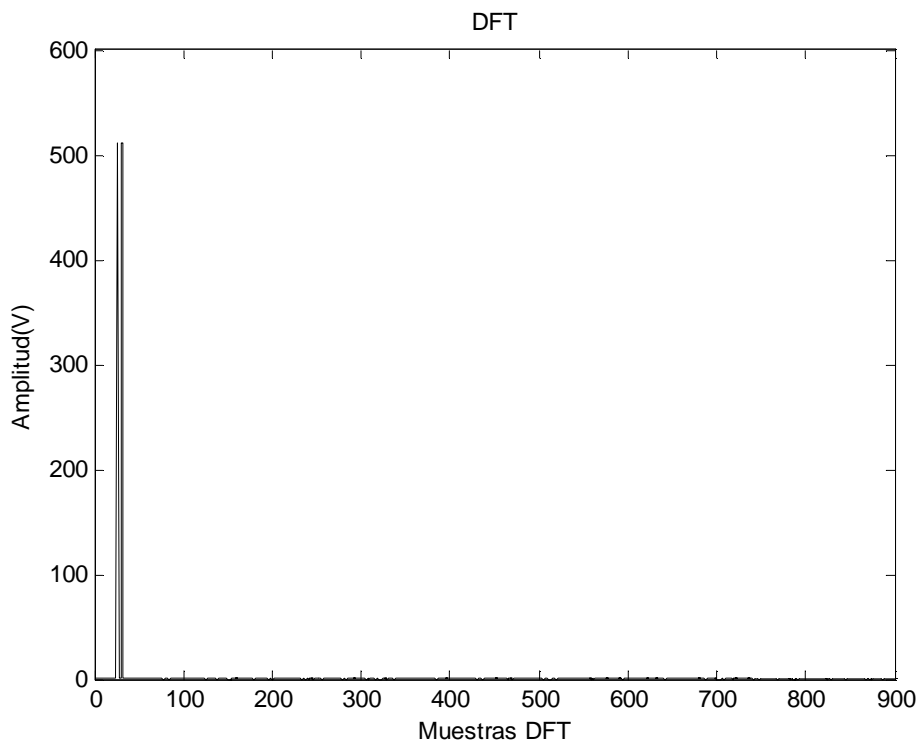


Fig68. DFT de la señal:

$x = \cos(2 \cdot \pi \cdot 25 / N \cdot n) + \cos(2 \cdot \pi \cdot 26 / N \cdot n) \cos(2 \cdot \pi \cdot 31 / N \cdot n) + \cos(2 \cdot \pi \cdot 32 / N \cdot n)$,
donde n es el tiempo discreto y N es el número de muestras de la señal x , $N = 1024$.

Como comprobamos en la figura anterior no se pueden distinguir los cosenos ya que en este caso la separación de frecuencias entre ellos es justo igual a la resolución, como teníamos en el caso 2-sparse.

Vemos en la siguiente figura que tanto MP como OMP encuentran los átomos del diccionario (en este caso el diccionario de Fourier, formado por cosenos y senos) que aportan mayor contribución a la definición de la señal proyectada. En este caso encontrar esos átomos del diccionario, conlleva encontrar las frecuencias que están presentes en nuestra señal suma de cosenos, estas serán:

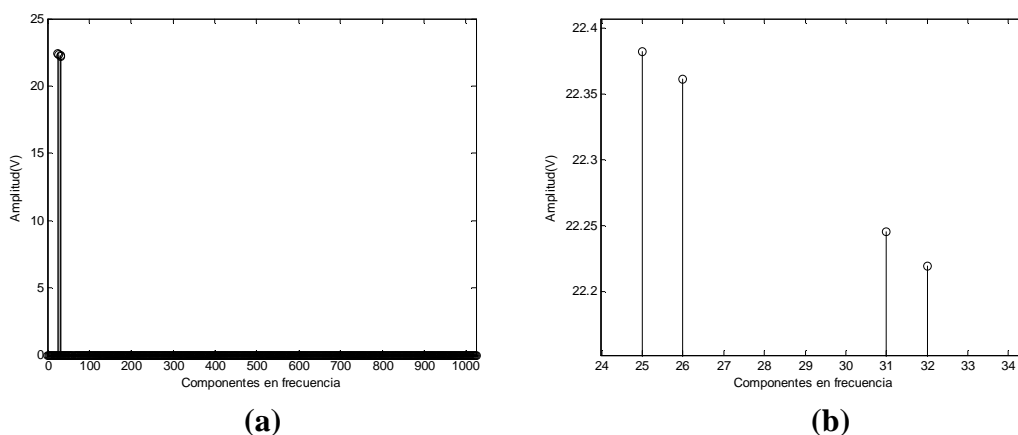


Fig69. (a) Frecuencias de los cosenos presentes en la señal 2-sparse en el dominio de la frecuencia obtenidas de la proyección sobre el diccionario de Fourier en los algoritmos MP y OMP. (b) Zoom de las frecuencias de los átomos del diccionario que aportan la mayor contribución a la definición de la señal.

En primer lugar se va a reconstruir la señal 4-sparse en el dominio de la frecuencia, a partir de los átomos del diccionario seleccionados, mediante el algoritmo Matching Pursuit. Se presentan a continuación la señal original, la señal recuperada por este algoritmo y el error producido en la reconstrucción.

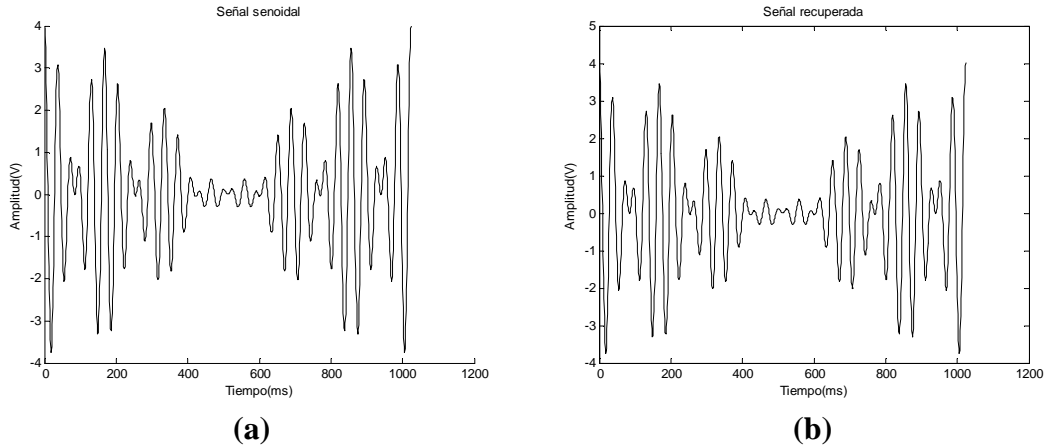


Fig70. Reconstrucción de señal sparse en el dominio de la frecuencia empleando el algoritmo MP. **(a)** Señal original. **(b)** Señal reconstruida.

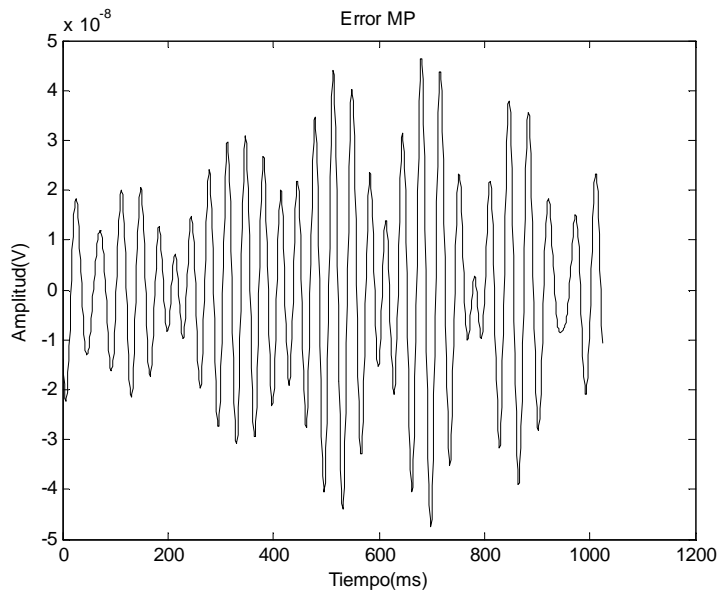


Fig71. Error en la reconstrucción de la señal sparse en el dominio de la frecuencia empleando el algoritmo MP.

Se observa en la figura que la diferencia entre la señal original y su reconstrucción utilizando MP sigue siendo del orden de 10^{-8} en base a la amplitud incluso aumentando el número de cosenos que conforman la señal. Si calculamos el error cuadrático medio como en los apartados anteriores (70), se obtiene $E_{mc} = 3.7573 \times 10^{-13}$, comparado con la amplitud original de la señal este error es prácticamente nulo ya que está diez órdenes de magnitud por debajo.

La señal fue reconstruida con un conjunto de tan sólo 50 muestras de las 1024 que conforman la señal original, lo cual implica una gran reducción en los datos necesarios para la reconstrucción casi perfecta de la señal.

Ahora se reconstruye la señal mostrada en la Figura 67 utilizando el algoritmo Orthogonal Matching Pursuit, la matriz de medición será la misma que se utilizó en la reconstrucción con el algoritmo MP, con la finalidad de hacer la reconstrucción bajo exactamente las mismas condiciones de proyección que en el caso anterior.

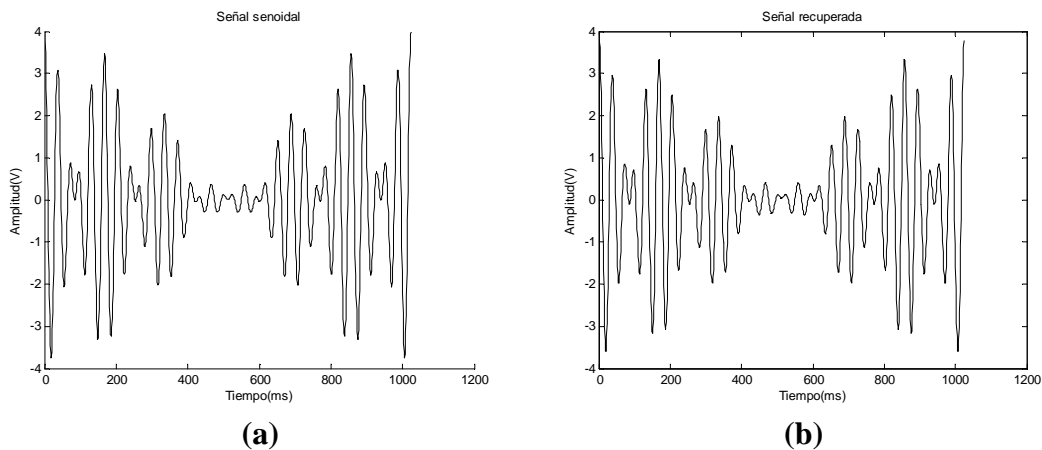


Fig72. Reconstrucción de señal sparse en el dominio de la frecuencia empleando el algoritmo OMP. **(a)** Señal original. **(b)** Señal reconstruida.

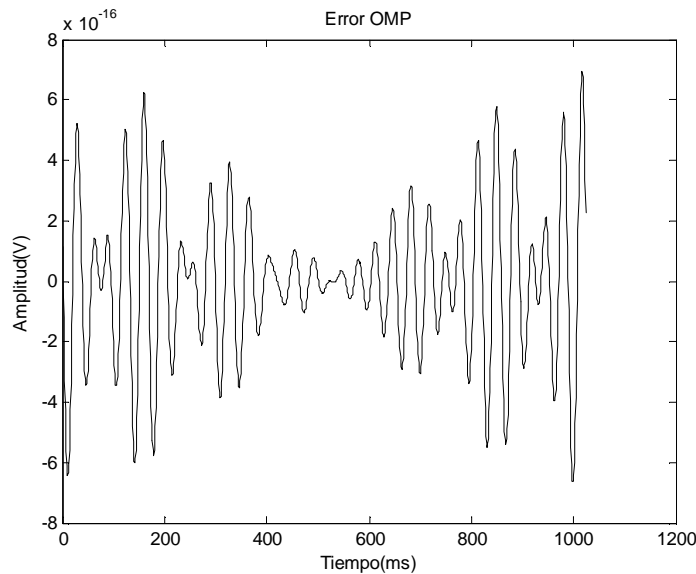


Fig73. Error en la reconstrucción de la señal sparse en el dominio de la frecuencia empleando el algoritmo OMP.

La diferencia entre la señal reconstruida y la señal original es del orden de 10^{-16} en base a la amplitud, si calculamos el error cuadrático medio empleando la ecuación (70), se obtiene que $E_{mc} = 1.2912 \times 10^{-28}$.

Se comprueba con esta última simulación que a pesar de que la señal sparse en el dominio de la frecuencia tenga un nivel de escasez mayor (es decir, esté formada por un número mayor de cosenos), se recuperan perfectamente las frecuencias de sus senoides

y el error cometido en la reconstrucción de la señal original es prácticamente el mismo. Comparamos este error en la siguiente tabla.

Método de reconstrucción	Error de reconstrucción señal 2-sparse	Error de reconstrucción señal 4-sparse
MP	$2,8285 \times 10^{-14}$	3.7573×10^{-13}
OMP	$5,2111 \times 10^{-29}$	1.2912×10^{-28}

Tabal 8. Comparación del error obtenido en la reconstrucción de señales sparse en el dominio de la frecuencia empleando los algoritmos MP y OMP.

Para terminar con las reconstrucciones de señales sparse en el dominio de la frecuencia, hemos realizado reconstrucciones de señales con ruido gaussiano blanco aditivo.

En la figura 74, se muestra el error que genera el algoritmo Matching Pursuit al reconstruir una señal que fue contaminada con ruido, la relación señal a ruido es de 20dB.

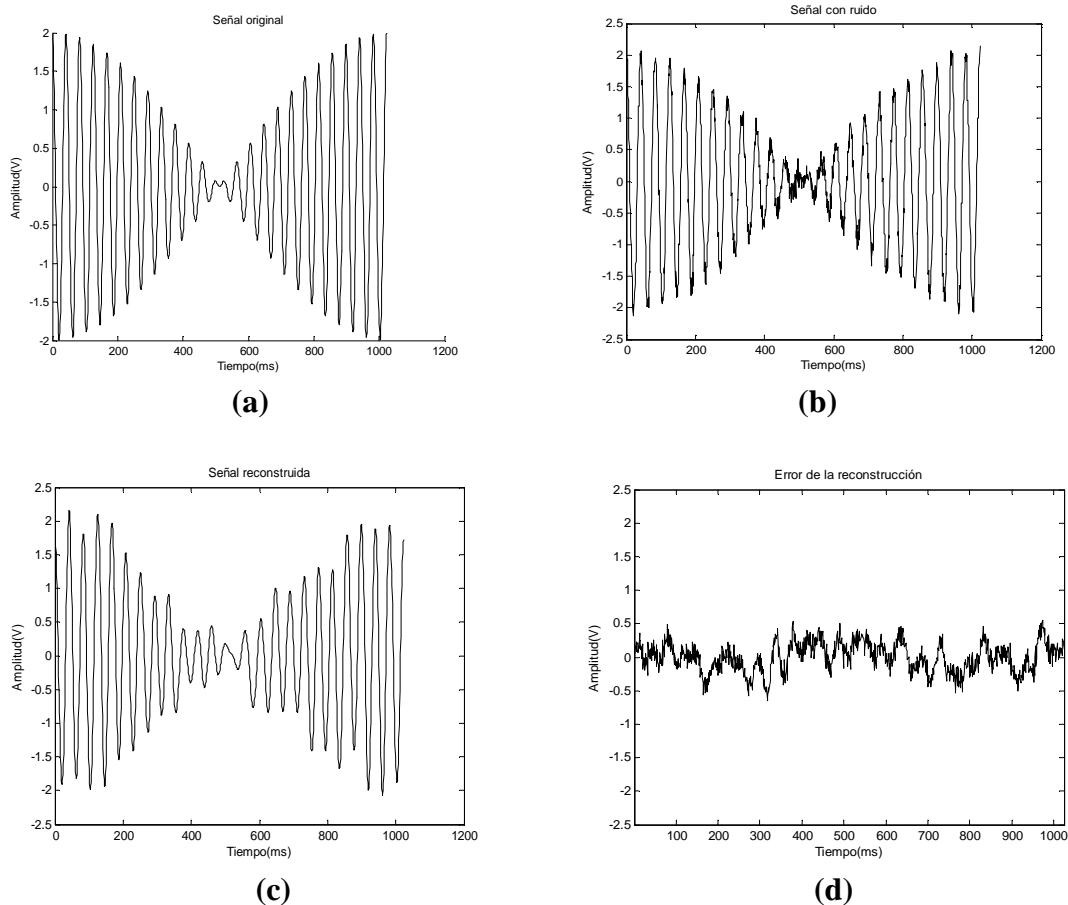


Fig74. Reconstrucción de una señal sparse en el dominio de la frecuencia con ruido blanco gaussiano aditivo con SNR = 20dB utilizando el algoritmo MP. (a) Señal original.(b) Señal contaminada, SNR = 20dB. (c) Señal reconstruida a partir de MP. (d) Error de reconstrucción.

Se esperaba que la diferencia entre la señal original y la reconstruida fuese menor, esto es debido a que como la reconstrucción se hace utilizando los átomos del diccionario, tenemos que en el diccionario no hay átomos que puedan representar el ruido introducido en la señal.

Con el fin de analizar el efecto del ruido en la reconstrucción en cuanto al error cometido, se recuperó la señal anterior con un ruido blanco gaussiano aditivo de diferentes valores de potencia.

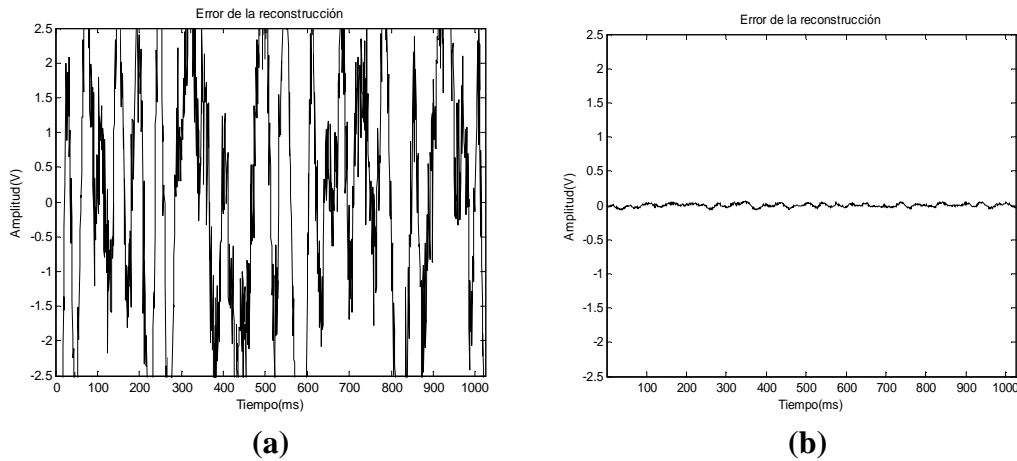


Fig75. Error de reconstrucción de una señal sparse en el dominio de la frecuencia con ruido blanco gaussiano aditivo de diferentes niveles de potencia, siendo el valor de la SNR(dB): **(a)** 5dB **(b)** 40dB.

Se pudo observar que a medida que la potencia de ruido disminuye (es decir, aumenta la relación señal a ruido) también lo hace el error de reconstrucción, sin embargo esto es un resultado muy esperado que no introduce ninguna novedad. En la figura 76 se muestra la dependencia del error de la reconstrucción a medida que la relación señal a ruido aumenta.

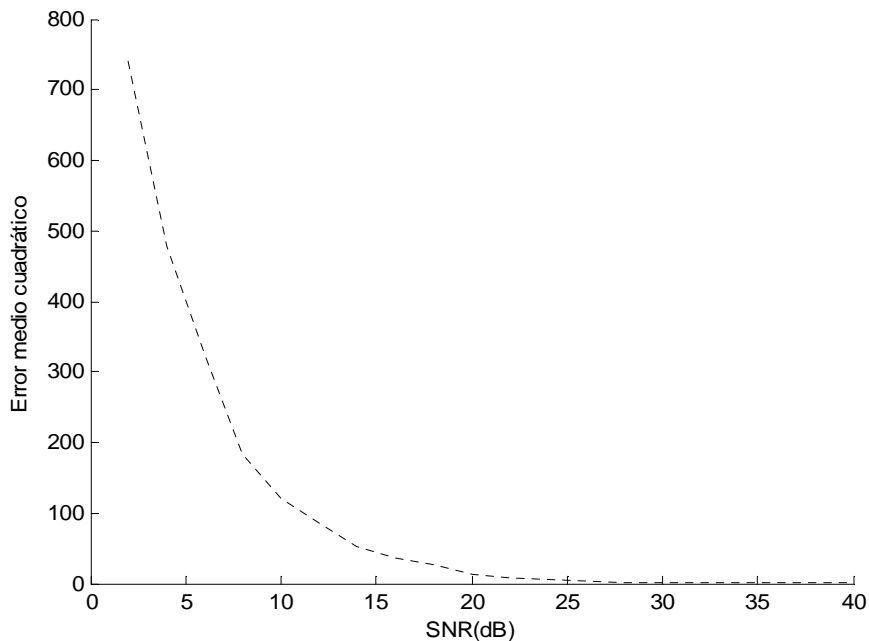


Fig76. Relación existente entre el error de reconstrucción y la SNR para el algoritmo MP en la reconstrucción de señales sparse en el dominio de la frecuencia.

En la figura 77, se muestra el error que genera el algoritmo Orthogonal Matching Pursuit al reconstruir una señal que fue contaminada con ruido, la relación señal a ruido es de 20dB.

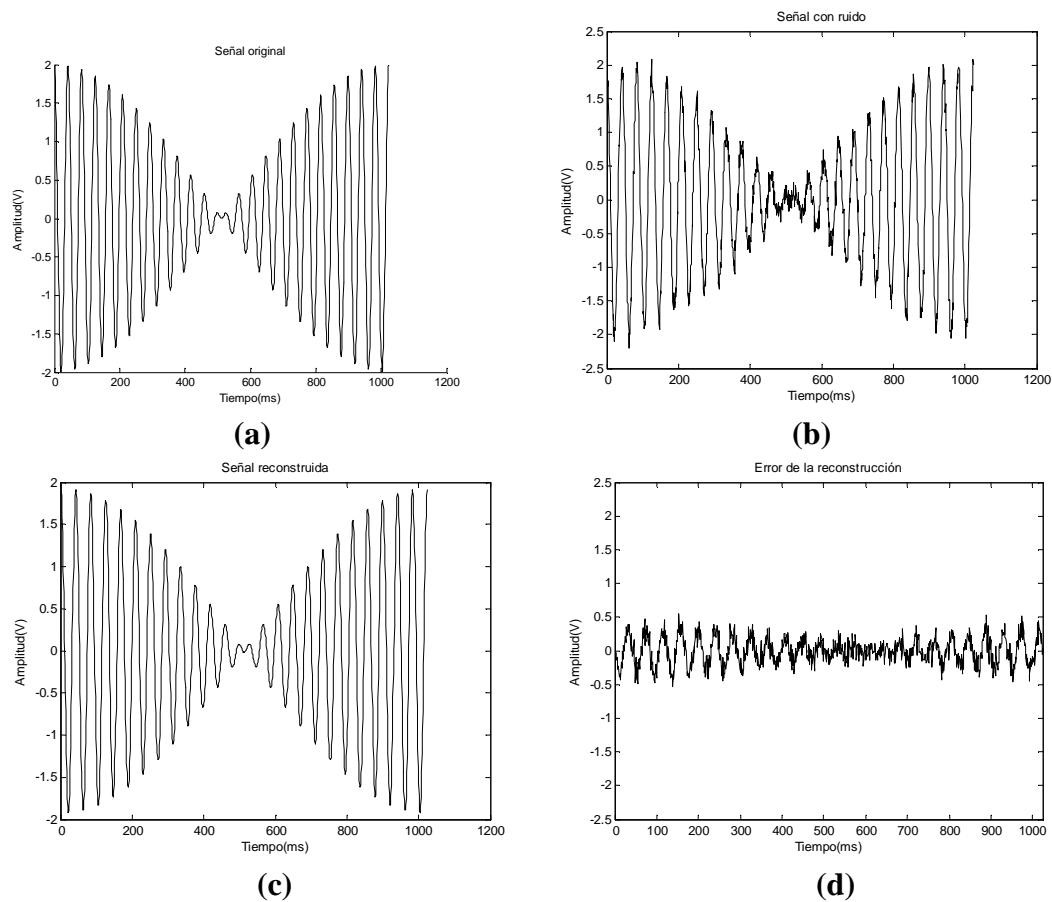


Fig77. Reconstrucción de una señal sparse en el dominio de la frecuencia con ruido blanco gaussiano aditivo con $SNR = 20dB$ utilizando el algoritmo OMP. **(a)** Señal original. **(b)** Señal contaminada, $SNR = 20dB$. **(c)** Señal reconstruida a partir de OMP. **(d)** Error de reconstrucción.

Al igual que en el apartado anterior, al realizar la reconstrucción de la señal original contaminada con un ruido de $SNR = 20dB$, se esperaba que la diferencia entre la señal original y la reconstruida fuese menor, es decir, que la señal reconstruida fuese más limpia.

Con el fin de analizar el efecto del ruido en la reconstrucción en cuanto al error cometido, se recuperó la señal anterior con un ruido blanco gaussiano aditivo de diferentes valores de potencia.

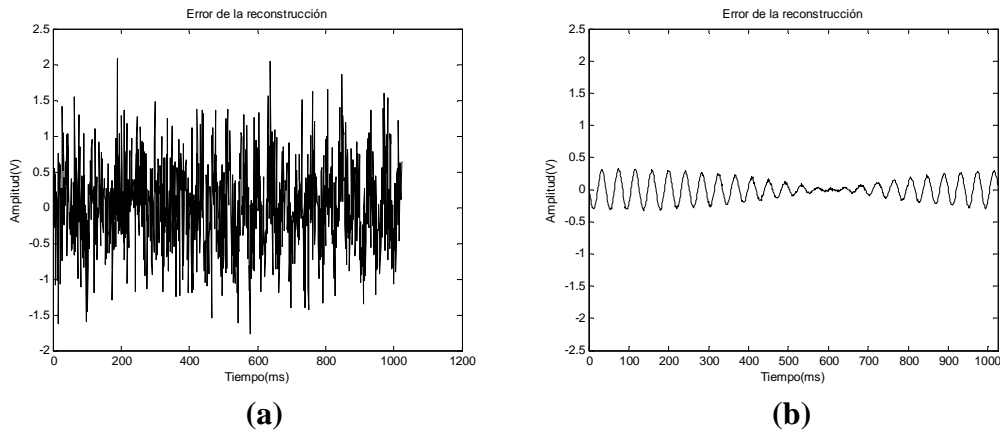


Fig78. Error de reconstrucción de una señal sparse en el dominio de la frecuencia con ruido blanco gaussiano aditivo de diferentes niveles de potencia, siendo el valor de la SNR(dB): **(a)** 5dB **(b)** 40dB.

Se puede observar que a medida que la potencia de ruido disminuye (es decir, aumenta la relación señal a ruido) también lo hace el error de reconstrucción, sin embargo esto es un resultado muy esperado que no introduce ninguna novedad. En la figura 79 se muestra la dependencia del error de la reconstrucción a medida que la relación señal a ruido aumenta.

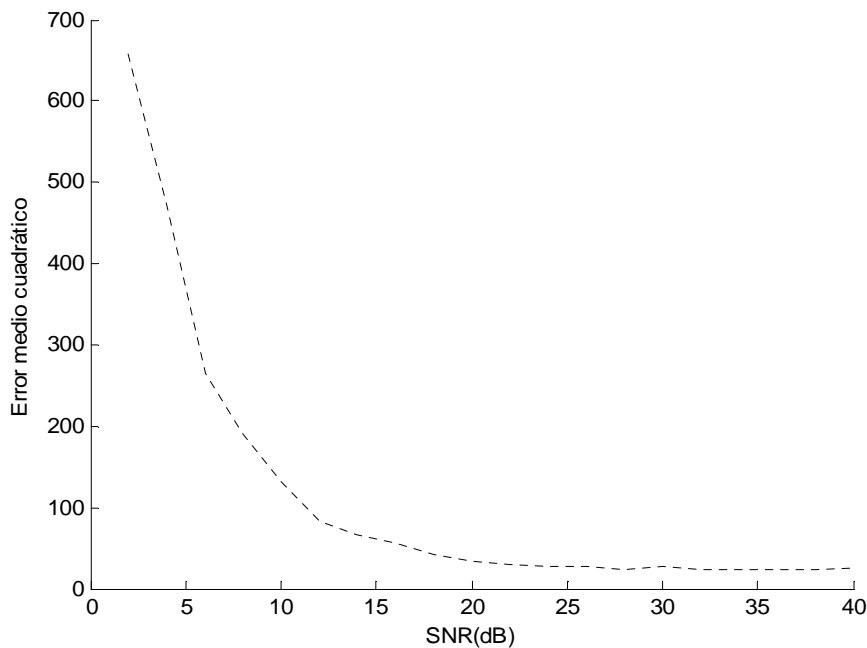


Fig79. Relación existente entre el error de reconstrucción y la SNR para el algoritmo OMP en la reconstrucción de señales sparse en el dominio de la frecuencia.

Al igual que en la reconstrucción de señales sparse en el dominio del tiempo, se esperaba que la diferencia entre la señal original y la reconstruida fuese menor, esto es debido a que como la reconstrucción se hace utilizando los átomos del diccionario,

tenemos que en el diccionario no hay átomos que puedan representar el ruido introducido en la señal.

Como podemos observar en las figuras anteriores, y al igual que ocurría en la reconstrucción de señales sparse en el dominio del tiempo, el error cuadrático medio disminuye, para ambos algoritmos, al disminuir la potencia del ruido (es decir, al aumentar la relación señal a ruido).

5.3 Otras aplicaciones

En este capítulo se trata de aplicar la teoría del Compressive Sensing a la reconstrucción de imágenes. Si bien en este proyecto no se han desarrollado los algoritmos necesarios para llevar a cabo esta tarea (esto se presentará como línea futura), sí que se presentan los algoritmos y el soporte hardware para una nueva teoría denominada Compressive Imaging (CI) y desarrollada por Wakin y Baraniuk, entre otros, [20]. Su proposición está basada en una nueva cámara digital de video e imagen que directamente adquiere proyecciones aleatorias de la señal sin recolectar primero los píxeles. La arquitectura de esta cámara emplea una matriz de micro espejos digitales para realizar cálculos ópticos de las proyecciones lineales de una imagen en pseudo-patrones binarios. Sus características incluyen la capacidad de obtener una imagen con un número de mediciones menor al número de píxeles que forman la imagen, esto puede reducir de forma significativa el cálculo necesario para la adquisición/codificación de video.

Debido a la gran cantidad de datos adquiridos en una imagen/video digital convencional a menudo se necesita una compresión inmediata con el fin de almacenar o transmitir los datos. Esta compresión explota un conocimiento a priori de los datos, tales como el hecho de que una imagen de N píxeles se puede aproximar como una combinación lineal escasa de $k \ll N$ coeficientes wavelet. Estos coeficientes wavelet se pueden calcular de forma eficiente a partir de los valores de los N píxeles y después almacenarlos o transmitirlos fácilmente junto con sus posiciones.

Este proceso tiene dos posibles desventajas importantes. En primer lugar, la adquisición de grandes cantidades de datos de imagen o video puede ser costoso. En segundo lugar, la compresión de los datos puede ser computacionalmente exigente, sobre todo en el caso del video. La teoría que estamos estudiando del Muestreo Compresivo, ofrece la esperanza de poder adquirir directamente una representación comprimida de la señal sin tener que muestrearla primero.

CS se basa en que las señales, imágenes y otros datos a menudo poseen algún tipo de estructura que permite una representación y un procesamiento inteligente. Se explota el hecho de que muchas señales tienen una representación escasa (sparse) en términos de alguna base. Por ejemplo, las imágenes suaves son sparse en el dominio de Fourier, las imágenes suaves a trozos son sparse en el dominio wavelet, y las imágenes estándar JPEG y JPEG2000 directamente explotan su condición de escasez.

El procedimiento estándar es el siguiente: (i) tomar la señal completa de n muestras; (ii) calcular el conjunto de coeficientes $\theta_n = \langle \psi_n, x \rangle$; (iii) localizar los k coeficientes mayores y descartar el resto; (iv) codificar los valores y las posiciones de estos

coeficientes. En los casos donde la longitud de la señal, n , es muy grande y el número de coeficientes sparse, k , es muy pequeño, este procedimiento resulta ser ineficiente.

Esto plantea la siguiente pregunta: para una señal dada, ¿es posible estimar directamente el conjunto de coeficientes mayores que no se descarta por el codificador?. La reciente teoría del Compressive Sensing presentada por Candés, Romberg, y Tao [4] y Donoho [21] demuestra que una señal que es k -sparse en una base (denominada base de escasez), puede ser recuperada por ck proyecciones lineales sobre una segunda base (denominada base de mediciones) que es incoherente con la primera, donde c es una pequeña constante y k es el nivel de escasez. Si bien el proceso de medición es lineal, el proceso de reconstrucción es absolutamente no lineal.

En CS no medimos ni codificamos los k coeficientes mayores sino que se emplea un conjunto de proyecciones m mucho menor que la longitud n de la señal, calculadas como $y(m) = \langle x, \phi_m \rangle$, siendo Φ la matriz de mediciones. En notación matricial tenemos $y = \Phi x$, siendo y el vector de proyecciones de longitud m y x la señal original de longitud n .

En [20] se desarrolla un nuevo sistema en el cual se apoya lo que podemos llamar Compressive Imaging (CI). El sistema incorpora una matriz de espejos micro controlados por bases pseudo aleatorias y uno o varios sensores de fotodiodo. Este hardware calcula una medida de la incoherencia, como establece la teoría del CS, y después se aplican los algoritmos de reconstrucción estudiados para recuperar la imagen.

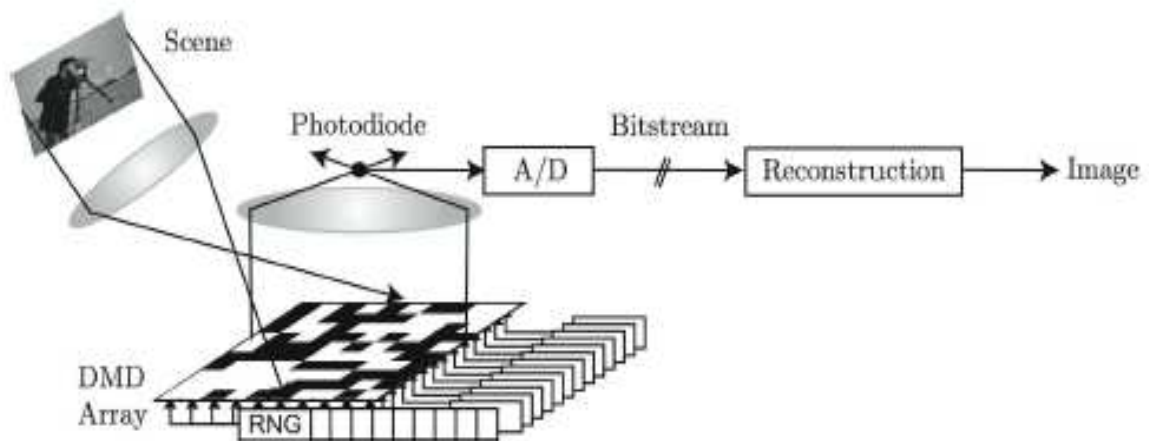


Fig80. Diagrama de bloques de Compressive Imaging (CI). El campo luminoso incidente (que corresponde a la imagen deseada x) se refleja en un dispositivo digital de micro espejos que forman una matriz (Digital Micromirror Device ,DMD). La orientación de estos espejos está modulada por el patrón pseudo aleatorio ϕ_m suministrado por el generador de números aleatorios (Random Number Generator, RNG). Cada espejo produce un voltaje en el fotodiodo que se corresponde con la proyección $y(m)$.

El equipo para la realización del concepto de CI es una cámara de un solo píxel. Combina un gran espejo micro controlado, que muestra una secuencia temporal de m bases pseudo aleatorias ϕ_m de la imagen, con un único sensor óptico para calcular las proyecciones y de la imagen.

El dispositivo digital de micro espejos (DMD) que se emplea es de Texas Instrument (TI). El DMD consiste en una matriz de 1024 x 768 micro espejos accionados electrostáticamente, donde cada espejo de la matriz se suspende sobre una célula de SRAM individual. Cada espejo gira alrededor de una bisagra y se puede colocar en uno de los dos estados (+12° o -12° desde la horizontal), por lo que la luz que cae sobre el DMD puede ser reflejada en dos direcciones dependiendo de la orientación de los espejos.

Con la ayuda de una lente biconvexa, la imagen deseada se forma en el plano del DMD; esta imagen actúa como objeto para la segunda lente biconvexa que centra la imagen en el fotodiodo. La luz es recogida de una de las dos direcciones en las que se refleja (por ejemplo, la luz reflejada por los espejos en el estado de 12°). La luz de una determinada configuración de los espejos del DMD se concentra en el fotodiodo para producir una tensión absoluta que produce un coeficiente $y(m)$ para esa configuración. La salida del fotodiodo se amplifica a través de un circuito op-amp y luego se digitaliza por un convertidor analógico-digital de 12 bits. Estas medidas del fotodiodo se pueden interpretar como el producto interno de la imagen deseada x con el vector de la base de mediciones ϕ_m . En particular, si $\rho(m)$ denota las posiciones de los espejos para el patrón de la medición m-ésima, el voltaje leído por el fotodiodo se puede escribir como:

$$(99) \quad v(m) \propto \langle x, \phi_m \rangle + DCoffset$$

diendo $\phi_m = 1_{\{\rho(m)=+12^\circ\}}$ y 1 es el indicador de función.

Para una imagen dada x , tomamos m medidas $\{y(1), y(2), \dots, y(m)\}$, correspondientes a las configuraciones de los espejos $\{\rho(1), \rho(2), \dots, \rho(m)\}$. Como los patrones $\rho(m)$ son programables, podemos seleccionarlos para ser incoherentes con la matriz de mediciones. Como se mencionó anteriormente, los patrones de medidas aleatorios o pseudo aleatorios cumplen una propiedad de incoherencia universal con cualquier base fijada, por eso se eligen patrones pseudo aleatorios de $\pm 12^\circ$ en los espejos.

Este sistema adquiere directamente un reducido conjunto de m proyecciones incoherentes de la imagen de n pixeles sin adquirir primero los valores de esos n pixeles.

En este artículo [20], encontramos un experimento para la reconstrucción de imágenes. Se presenta una copia impresa de la letra R delante de la cámara, figura 81. Como esta imagen de prueba es constante a trozos (con bordes afilados), puede ser escasamente representada en el dominio wavelet. En las figuras 82 (b) y 82 (c) se muestra esta representación con $k = 205$ y 409, respectivamente. Usando un número de mediciones

$m = 819$ y 166 (aproximadamente 4 veces la k usada en (b) y (c)), se reconstruye la imagen usando el algoritmo OMP.

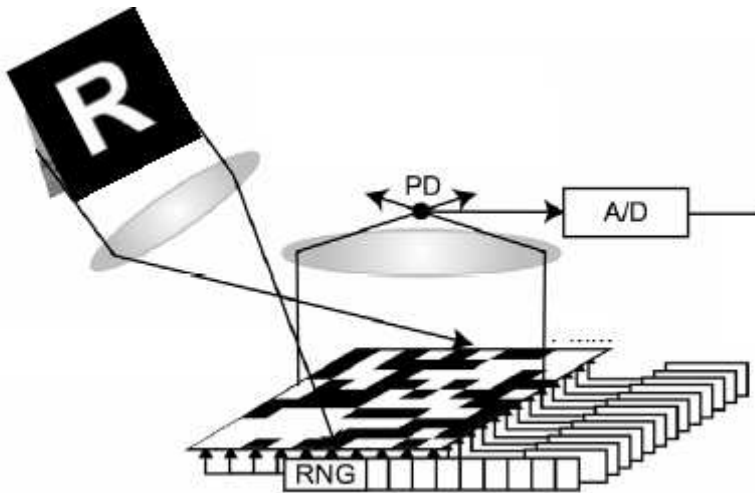


Fig.81. Experimento para la reconstrucción de imágenes; se presenta una copia impresa de la letra R delante de la cámara.

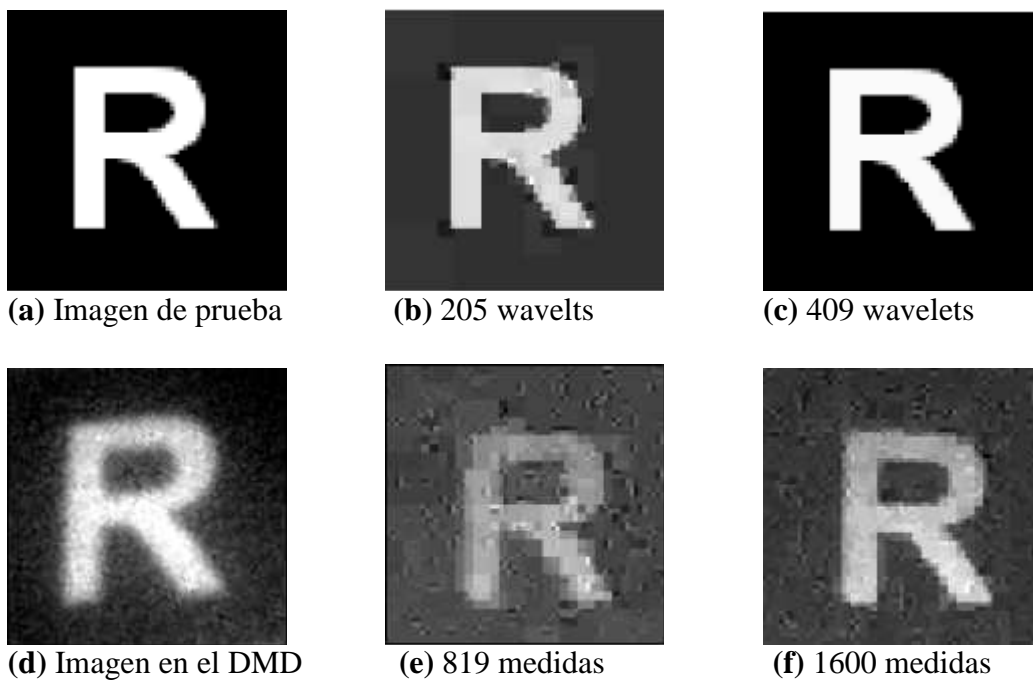


Fig82. CI de una imagen de 64×64 ($n = 4096$ píxeles). (a) Imagen de prueba de alta resolución. (b) Aproximación de la imagen por sus 205 coeficientes wavelets y (c) por sus 409 coeficientes wavelets. (d) Imagen de una cámara convencional de 320×240 adquirida en el plano del DMD. Reconstrucción CS con (e) 819 mediciones y (f) 1600 mediciones.

En este artículo [20], se presenta un prototipo de sistema de imagen que emplea con éxito los principios del Compressive Sensing. La cámara tiene muchas características atractivas, como la simplicidad, universalidad, robustez y escalabilidad, que le permitirá

tener efecto sobre una variedad de aplicaciones diferentes. Otra característica interesante y potencialmente útil de este sistema es que se descarga el proceso de recolección de datos centrándose en la reconstrucción. No sólo se disminuye la complejidad y el consumo de electricidad del dispositivo, sino que también se permitirán nuevos esquemas de medidas.

5.4 Conclusiones

La primera aplicación estudiada en este apartado se refiere a la estimación de canales de comunicaciones sparse. Encontramos canales sparse en muchas aplicaciones para las comunicaciones. Explotando esta escasez, el estimado de un canal se puede obtener mediante el algoritmo Matching Pursuit. Se ha demostrado que esta estimación obtenida es inherentemente sparse, a diferencia de la estimación de mínimos cuadrados (LS) del canal, donde cada uno de los valores de los pulsos serán generalmente distintos de cero.

En el algoritmo MP como la optimización de cada iteración se realiza sobre todos los átomos del diccionario, es posible deseleccionar un átomo previamente seleccionado, disminuyendo la velocidad de convergencia. Este problema de reelección se resuelve con el algoritmo Orthogonal Matching Pursuit, ya que en cada iteración se toma la columna que genera la máxima proyección para formar el nuevo diccionario. Se ha demostrado que eliminando la reelección de columnas, el algoritmo OMP puede obtener un estimado del canal más preciso.

Como vimos en la Tabla 6, el error cuadrático medio obtenido en la reconstrucción del canal sparse empleando el algoritmo OMP ($9,7812 \times 10^{-27}$) es más de diez veces menor que el error obtenido con el MP ($2,9871 \times 10^{-15}$).

Como podemos comprobar, los algoritmos MP y OMP se pueden emplear en la reconstrucción de un canal sparse a pesar de que en este caso no se cumpla $m \ll n$, si no que el número de mediciones m será mayor o igual a la longitud de la señal n .

Por otro lado, se han estudiado estos algoritmos en la recuperación de señales sparse en el dominio de la frecuencia, como aplicación se ha implementado la detección de sinusoides. Hemos comprobado que ambos algoritmos iterativos, MP y OMP, son capaces de distinguir perfectamente las frecuencias de los cosenos que conforman la señal de prueba, y a partir de estas frecuencias recuperar la señal original con un error mínimo y del mismo orden que el producido en la reconstrucción de señales sparse en el dominio del tiempo.

Como vemos en la Tabla 8, estos algoritmos presentan errores prácticamente del mismo orden en la reconstrucción de señales sparse en el dominio de la frecuencia con distintos niveles de escasez, en este caso, se reconstruyeron señales con escasez 2 y 4 (es decir, señales formadas por la suma de dos y de cuatro cosenos).

Por último se ha presentado, pero no se ha desarrollado en este proyecto, la aplicación de la teoría del Compressive Sensing a la reconstrucción de imágenes, estudiando una serie de algoritmos y un hardware específico presentado por Wakin y Baraniuk [20]. Este dispositivo consiste en una cámara de un solo pixel de forma que, el campo luminoso incidente (que corresponde a la imagen deseada x) se refleja en un dispositivo

digital de micro espejos que forman una matriz (Digital Micromirror Device, DMD). La orientación de estos espejos está modulada por el patrón pseudo aleatorio ϕ_m suministrado por el generador de números aleatorios (Random Number Generator, RNG). Cada espejo produce un voltaje en el fotodiodo que se corresponde con la proyección $y(m)$. Este conjunto de proyecciones se empleará en la reconstrucción de la señal a partir de los algoritmos iterativos estudiados, MP y OMP. Algunas características de este dispositivo, como la simplicidad, universalidad, robustez y escalabilidad, le permitirá tener efecto sobre una gran variedad de aplicaciones diferentes.