



## **Trabajo Fin de Grado**

### **Técnicas de Machine Learning aplicadas a la eliminación de ruido en señales de ondas gravitacionales**

Alumno: Pablo Higuera de la Iglesia

Tutor 1: José Antonio Font Roda

Tutor 2: Alejandro Torres Forné

Curso 2019/2020

# Índice

## 1. Introducción

## 2. Fundamento teórico

2.1 Método ROF de variación total

2.2 *Sparse* coding con diccionarios

2.3 Aprendizaje de diccionarios

## 3. Sistemática del experimento

3.1. Generación de nuevos diccionarios a partir de un catálogo de señales de ondas gravitacionales.

3.2. Búsqueda de parámetros óptimos

3.3. Resultados y análisis

3.3.1. Método simple (M1)

3.3.2. Método doble (minimización separada) (M2)

3.3.3. Método doble (minimización simultánea) (M3)

## 4. Generación de los diccionarios

## 5. Reconstrucción de las señales

5.1. Reconstrucción por ROF

5.2. Reconstrucción por diccionarios

## 6. Búsqueda de los parámetros (validación)

6.1. Parámetros de los diccionarios

## 7. Resultados en las señales test

7.1. Señales test y distancias

7.2. Aplicación de M1

7.3. Aplicación de M2

7.4. Aplicación de M3

7.5. Comparación gráfica de los métodos

## 8. Más allá de SNR 8

## 9. Conclusiones

## 10. Bibliografía

## Resumen

Este trabajo tiene por objetivo estudiar el problema de la eliminación del ruido en las señales de ondas gravitacionales observadas por detectores interferométricos (Advanced LIGO y Virgo). Las ondas gravitatorias necesitan de detectores muy sensibles para poder ser observadas y la eliminación del ruido en su procesamiento es imprescindible para su estudio. Para ello, en este trabajo analizamos métodos de variación total con norma  $L_1$  empleados conjuntamente con una técnica de *machine learning* basada en diccionarios. Los métodos son aplicados a un catálogo numérico de señales gravitatorias de colapso estelar inyectadas en ruido Gaussiano, para lo cual utilizamos tres procedimientos diferentes. Comparando los resultados de los tres procedimientos, el estudio demuestra cual de ellos es el más adecuado para llevar a cabo las reconstrucciones de manera más precisa. En particular, el algoritmo que ofrece los mejores resultados es capaz de eliminar el ruido de manera satisfactoria para eventos de colapso estelar con una relación señal ruido del orden de 10. El trabajo también discute las dificultades de los algoritmos para reconstruir señales en situaciones más exigentes, con valores inferiores de la relación señal ruido.

## Abstract

The aim of this work is to study the problem of noise removal in gravitational wave signals observed by interferometric detectors (Advanced LIGO and Virgo). Gravitational wave signals need very sensitive detectors to be observed and the elimination of noise in their processing is essential for their study. To do this, in this work we analyze total-variation methods based on the  $L_1$  norm used together with a machine learning technique based on dictionaries. The methods are applied to a numerical catalog of gravitational wave signals from stellar core collapse injected into Gaussian noise, for which we use three different procedures. Comparing the results of the three procedures, this study shows which one of them is the most suitable to carry out the reconstructions in the most accurate way. In particular, the algorithm that offers the best results can satisfactorily remove noise for stellar collapse events with a signal-to-noise ratio of the order of 10. This work also discusses the difficulties of the algorithms to reconstruct signals in more demanding situations, with lower values of the signal-to-noise ratio.

## 1. Introducción

A principios del siglo XX, Albert Einstein presentaba al mundo su teoría de la Relatividad. La Relatividad General explica de una forma totalmente novedosa cómo se comporta el espacio-tiempo en presencia de cuerpos masivos, proporcionando una interpretación geométrica de la gravedad. La materia curva el espacio-tiempo, lo que genera una variación de la métrica. Las ondas gravitacionales son la manifestación del movimiento de cuerpos masivos en el tejido del espacio-tiempo, representando las perturbaciones de un espacio-tiempo plano *background* que se propagan a la velocidad de la luz. Estas ondas, no obstante, son producidas por fuentes astrofísicas lejanas, por lo que la señal que recibimos en la tierra es extraordinariamente débil, lo que hace que estudiar la física de ondas gravitacionales suponga todo un desafío, aun con la tecnología moderna de la que se dispone.

La primera evidencia indirecta de la existencia de las ondas gravitacionales llegó con el descubrimiento del púlsar binario PSR 1913+16 (Hulse and Taylor [1]), que se trata de un sistema binario de dos estrellas de neutrones. La detección directa de ondas gravitacionales no fue posible hasta que no se inventaron sistemas de medida apropiados para ello. A pesar de que desde la década de los 60 se empezaron a idear máquinas que pudieran medirlas, pasando por nuevas generaciones de detectores resonantes en la década de los 90, no fue hasta el 14 de Septiembre de 2015 que se

pudo detectar por primera vez de forma directa una onda gravitacional, empleando interferometría con láseres de gran precisión [2]. Este evento, llamado GW150914, fue detectado por las colaboraciones científicas LIGO y Virgo, y se anunció en 2016. La detección capturó las últimas órbitas del baile de dos agujeros negros alrededor de su centro de masas hasta fundirse en uno solo. El proceso generó hasta 3 masas solares de energía que se propagaron por el espacio en forma de ondas gravitacionales.

Las ondas gravitacionales con las que he trabajado en el estudio presentado en esta memoria proceden de simulaciones de relatividad numérica, llevadas a cabo con el código CoCoNut de [3]. Este código simula el evento final de la evolución de estrellas masivas: el colapso de su núcleo. Cuando la capa de silicio caliente empuja al núcleo de hierro por encima de su masa efectiva de Chandrasekhar, se inicia el colapso por una combinación de captura electrónica y fotodesintegración de núcleos pesados, llevando a una pérdida de presión en el centro de la estrella. Si la estrella es suficientemente masiva, entre 10 y 100 masas solares, este colapso se prolonga hasta que su núcleo interno alcanza densidades superiores a la densidad nuclear de saturación, donde la ecuación de estado se endurece y se produce un rebote casi instantáneo del núcleo interno, dando lugar así al nacimiento de una supernova hidrodinámica, con una onda de choque que viaja hacia fuera de la estrella en radio y masa, perdiendo rápidamente energía cinética. El remanente de la explosión es una estrella de neutrones o, en casos extremos, un agujero negro.

La detección de las ondas gravitacionales conlleva varios temas tecnológicamente complejos. No solo en el diseño, construcción y mantenimiento de los sistemas de medida (interferómetros), sino también en lo referido al almacenamiento y procesamiento de los datos que éstos generan. En este trabajo vamos a afrontar uno de los problemas asociados a la detección de radiación gravitatoria: el ruido con el que llegan a los interferómetros. Más concretamente, vamos a estudiar dos algoritmos destinados a reducir el ruido en estas señales. Es conveniente, por tanto, hablar antes de qué es el ruido y cómo se ha trabajado previamente en este problema en campos diferentes. El procesamiento de señales es el campo que se encarga del tratamiento y análisis de señales como sonido o imágenes. El ruido es la parte de la señal no deseada y en ocasiones de origen desconocido que dificulta su tratamiento. Este término no deseado puede surgir por complicaciones en el proceso de medida de la señal, en su tratamiento, almacenamiento, etc. La extraordinaria sensibilidad requerida para detectar fuentes astrofísicas de ondas gravitatorias produce que los detectores se vean afectados por numerosas fuentes de ruido. La sensibilidad de los detectores tiene una fuerte dependencia con la frecuencia de forma que la sensibilidad se ve limitada fundamentalmente por 3 fuentes de ruido. A bajas frecuencias ( $<20$  Hz) la limitación viene dada por el ruido sísmico mientras que a altas frecuencias ( $>2$  kHz) por el ruido producido por fluctuaciones cuánticas del láser. La sensibilidad de los detectores alcanza su máximo a frecuencias medias ( $\sim 200$  Hz), donde el movimiento Browniano de las suspensiones y espejos es la fuente limitante. Además, hay muchas más fuentes de ruido transitorias de corta duración, llamadas glitches, cuyo origen es instrumental o ambiental. Todas estas características hacen que el ruido de los detectores sea no blanco (la respuesta no es plana en frecuencia) y no gaussiano (hay fuentes transitorias que no responden a una distribución gaussiana). En el caso de las ondas gravitatorias de objetos lejanos, las señales llegan a nuestros interferómetros se encuentran en el límite de la detección, esto es, con una relación señal-ruido (SNRs, por sus siglas en inglés) muy pequeñas. Por tanto, la eliminación del ruido es una prioridad para analizar las señales y poder entender en última instancia los sucesos físicos que las originan. En este trabajo se ha utilizado ruido gaussiano (no contiene transitorios) cuyo densidad espectral de potencia coincide con la curva de sensibilidad de detectores interferométricos LIGO. Este es el paso inicial en el desarrollo de nuevos algoritmos, una vez se ha probado su eficacia con este ruido simplificado, se estudian usando ruido real de los detectores y, si funcionan adecuadamente, finalmente se integran dentro del software de detección de colaboración LIGO/Virgo.

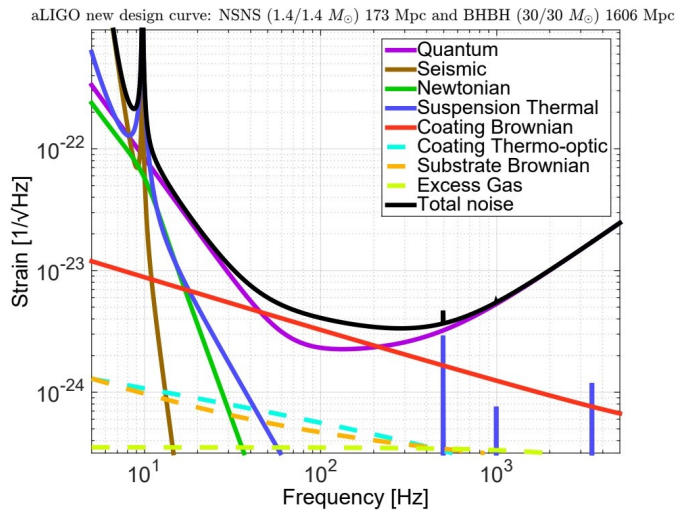


Fig 1. Estimación de la curva de sensibilidad del diseño de Advanced LIGO. Las diferentes fuentes de ruido se muestran en la leyenda.

El campo del procesamiento de imágenes tiene desde hace tiempo muchas herramientas para tratar este problema. La herramienta principal que hemos utilizado en este trabajo es un algoritmo llamado “**dictionary learning**” [4][5]. Este algoritmo de aprendizaje (*machine-learning*) ya ha sido empleado en el contexto del procesamiento de imágenes, aportando muy buenos resultados. Un ejemplo práctico que ilustra muy bien la utilidad de los algoritmos basados en diccionarios se encuentra en [6], donde se genera un diccionario a partir de una imagen para posteriormente eliminar el ruido añadido a esta. Se extraen fragmentos de la imagen para generar un diccionario, y después se reconstruye en base a la imagen con ruido, como se ve en el siguiente ejemplo:

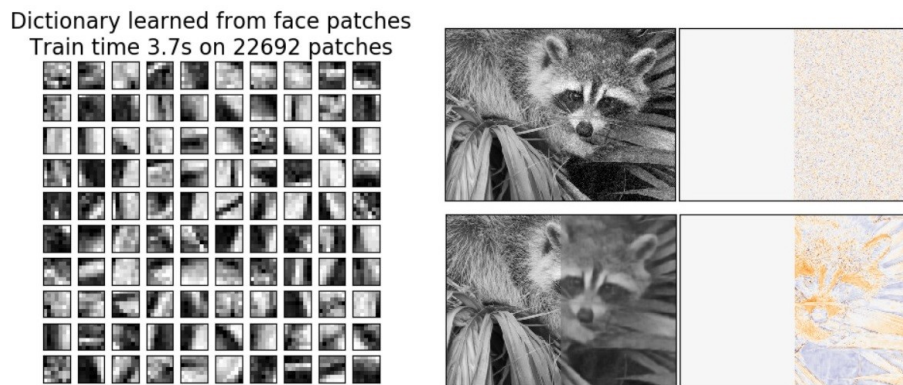


Fig 2. Izquierda: fragmentos del diccionario. Derecha: Imágenes de un mapache con distorsión en la parte de la derecha y su posterior reconstrucción [6]

Además de emplear un algoritmo basado en diccionarios, en este trabajo hemos ido un poco más allá y lo hemos combinado con un algoritmo de eliminación de ruido basado en la norma  $L_1$ , el algoritmo de Rudin-Osher-Fatemi, **ROF** [7][8][9] de ahora en adelante. Este método elimina ruido sin utilizar ninguna información a priori sobre la señal o los datos.

Por tanto, en este trabajo hemos empleado, por un lado, diccionarios de forma aislada, reproduciendo los resultados de [5], así como nuevos diccionarios generados de forma propia. Emplear el algoritmo ROF combinado con los diccionarios permite dos posibilidades para hallar los parámetros óptimos de cada método: minimizando el error de la reconstrucción aplicando primero ROF y después los diccionarios, cada uno de forma independiente; o minimizando el error aplicando ambos métodos conjuntamente. De esta forma, los parámetros principales de cada método están ligados entre sí.

Esta memoria está estructurada de la siguiente forma. Explicaremos en el siguiente apartado, brevemente, las matemáticas que subyacen a las técnicas informáticas empleadas; después comentaremos la sistemática del experimento. Tan importante como la generación de los diccionarios es utilizar los parámetros adecuados para reproducir de forma fiel las señales originales. Ahondaremos en este apartado con suficiente profundidad. Finalmente, procederemos a reconstruir unas señales test que nos servirán para poner a prueba la fiabilidad del método seguido, y enunciaremos las conclusiones que surgen a partir de nuestro trabajo.

## 2. Fundamento teórico

En este apartado comentaremos brevemente las matemáticas sobre las que descansan todas las operaciones computacionales que llevaremos a cabo en nuestro trabajo.

### 2. 1. Método ROF de variación total

Los métodos que hemos utilizado en este trabajo para afrontar el problema de la eliminación del ruido son métodos variacionales basados en la norma  $L_1$ . Estos métodos se utilizan para resolver el problema de *denoising* como un problema variacional (ver, por ejemplo, [5] y referencias allí citadas). Matemáticamente, la situación que queremos resolver es  $y = u + n$ , donde  $y$  es la señal medida,  $u$  la señal original y  $n$  el ruido. Comentaremos primero un método en particular de variación total, el modelo Rudin-Osher-Fatemi (ROF). Posteriormente, explicaremos un método de Aprendizaje de Diccionarios. La solución para la señal  $u$  se obtiene a partir de la siguiente ecuación

$$u_\lambda = \underset{u}{\operatorname{argmin}} \left\{ \mathcal{R}(u) + \frac{\lambda}{2} \mathcal{F}(u) \right\} \quad (1)$$

donde el término  $\mathcal{R}(u)$  es el término de regularización, una ligadura sobre los datos; mientras que  $\mathcal{F}(u)$  es el término de fidelidad, que mide la similitud de la solución con los datos originales. El parámetro  $\lambda$  es un multiplicador de Lagrange que recibe el nombre de parámetro de regularización, y controla el peso relativo de los términos de la ecuación.

A pesar de que ambos métodos, ROF y Diccionarios, resuelven el mismo problema, estos lo hacen de formas muy diferentes. Rudin, Osher y Fatemi propusieron en [9] el uso de la norma TV (*total-variation*) como término de regularización  $\mathcal{R}(u) = \int_\Omega |\nabla u|$  ligado a la condición  $\|y-u\|^2$ , donde  $|\cdot|$  y  $\|\cdot\|$  hacen referencia a las normas  $L_1$  y  $L_2$  respectivamente. Este modelo reduce el ruido favoreciendo las soluciones con gradiente *sparse* (que suele traducirse como “ralas”), es decir, con muchos ceros en la solución, menor número de coeficientes diferentes de cero posibles. La ecuación de Euler-Lagrange asociada presenta una singularidad en  $|\nabla u| = 0$ ,

$$\nabla \cdot \frac{\nabla u}{|\nabla u|} + \lambda(y - u) = 0 \quad (2)$$

como se puede ver en la fórmula anterior. Para solucionar este problema, podemos cambiar la norma TV incluyendo un parámetro pequeño y positivo,  $\beta$ , siguiendo la siguiente ecuación

$$\text{TV}_\beta(u) := \int \sqrt{|\nabla u|^2 + \beta^2} \quad (3)$$

el valor de este parámetro  $\beta$  se especifica más adelante junto con el resto de parámetros del código que ejecuta este método.

## 2. 2. *Sparse coding* con diccionarios

Un diccionario es un conjunto de señales, como, por ejemplo, las ondas gravitacionales, de longitud  $n$ , a las que llamaremos *átomos*. La ecuación que representa la reconstrucción de la señal es

$$\mathbf{u} = \mathbf{D} \cdot \mathbf{a} \quad (4)$$

donde  $\mathbf{D} = [\mathbf{d}_1, \dots, \mathbf{d}_p]$  es el diccionario,  $\mathbf{u}$  la señal a recuperar y  $\mathbf{a}$  un vector  $\mathbf{a} \in \mathbb{R}^p$  que contiene los coeficientes de la representación. Para poder llevar a cabo la reconstrucción de la señal, nuestro código se encarga de encontrar este vector de coeficientes, que en principio ha de ser *sparse*, es decir, tener muchos ceros y, por tanto, pocos coeficientes numéricos con valor distinto de cero. De esta manera, con tan solo unos pocos coeficientes, el diccionario puede representar la señal original. Este proceso se denomina *sparse coding* [10][11][12] y es ampliamente utilizado en el contexto del procesamiento de imágenes. En la siguiente figura aportamos un concepto más visual de esta explicación.

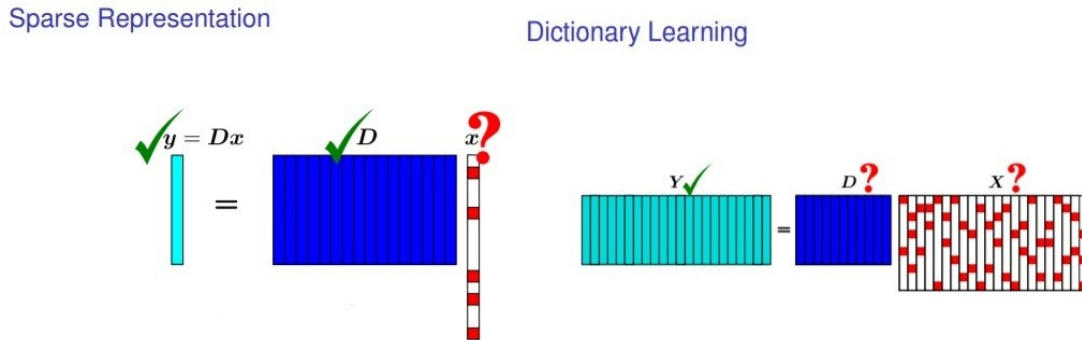


Fig 3. Representación visual de  $\alpha$ . Izquierda: recuperación de una señal como una onda a partir de un diccionario  $D$  y un vector de coeficientes unidimensional,  $X$ . Derecha: recuperación de de una señal como una imagen a partir de un diccionario y un vector de coeficientes bidimensional [12].

Por tanto, en los métodos basados en el uso de diccionarios, la eliminación del ruido se produce asumiendo que la señal original  $u$  se puede representar con tan solo unos pocos átomos del diccionario,  $D$ . El término de fidelidad de la ecuación (1) es por tanto

$$\mathcal{F}(\alpha) = ||y - D\alpha||^2 \quad (5)$$

y como comentábamos antes, el problema se reduce a encontrar el vector de coeficientes para reconstruir las señales.

## 2. 3. Aprendizaje de diccionarios

En el apartado anterior se asume que el diccionario  $D$  es una matriz fija y, por tanto, el problema reside en encontrar un vector de coeficientes apropiado. Tradicionalmente se han utilizado diccionarios fijos basados en tipos de onda. No obstante, los resultados han visto una mejora significativa en el contexto del entrenamiento de estos diccionarios [4][5], en lugar de dejarlos fijos. Este proceso consiste en dividir el diccionario en trozos, o *patches*, de forma que generalmente el número de *patches* de entrenamiento,  $m$ , es mucho mayor que su longitud,  $n$ :  $m \gg n$ . El procedimiento ha de tener en cuenta el diccionario  $D$  también como variable del problema y, para ello, se puede reformular el problema como

$$\alpha_\lambda, D_\lambda = \operatorname{argmin}_{\alpha, D} \left\{ \frac{1}{n} \sum_{i=1}^m \|D\alpha_i - x_i\|_2^2 + \lambda |\alpha_i| \right\} \quad (6)$$

donde  $x_i$  es cada *patch* de entrenamiento. La resolución de este problema de minimización ha sido llevada a cabo con el paquete del lenguaje de programación Python de aprendizaje automático *scikit-learn* [13], y el problema en concreto se explicita en el apartado de Generación y entrenamiento de los diccionarios. El parámetro de regularización  $\lambda$  presente en la ecuación (6) es el parámetro propio del método, el cual más adelante será referenciado como  $\alpha$ , por ser el parámetro de la función de Python que ejecuta la reconstrucción. Aunque estamos empleando el mismo símbolo, este parámetro no tiene nada que ver con el vector de coeficientes  $\alpha$ , que hemos mencionado antes, el cual es necesario encontrar para llevar a cabo las reconstrucciones.

## 3. Sistemática del experimento

Como hemos comentado en la introducción, este trabajo consiste en aplicar técnicas matemáticas a través de algoritmos informáticos para lograr resultados óptimos de un problema físico real. Nuestro experimento es puramente teórico en el sentido en el que nuestro montaje experimental es simplemente el ordenador con el que hemos trabajado para realizar todas las operaciones matemáticas. De este modo, podemos dividir el experimento en 3 fases bien diferenciadas:

### 3. 1. Generación de nuevos diccionarios a partir del catálogo de señales de ondas gravitacionales.

A continuación, pasamos a explicar brevemente la morfología de las señales de nuestro catálogo de ondas gravitatorias asociadas a procesos de colapso estelar y explosión supernova, calculadas numéricamente por Dimmelmeier y colaboradores en [3]. Estas señales han sido empleadas para las tres partes del experimento: generación y entrenamiento, validación y resultados. En la figura 3 se pueden ver diferentes formas de onda representativas del suceso físico de colapso del núcleo de una estrella masiva (con una masa superior a ocho masas solares). Estas señales exhiben un primer pico positivo seguido de un pronunciado pico negativo. Este último se corresponde al instante del rebote hidrodinámico del material que está colapsando, debido al endurecimiento de la ecuación de estado, y a la formación de la onda de choque. Esta parte de la señal se conoce con el nombre de *burst*. Posteriormente, las señales se caracterizan por la presencia de oscilaciones amortiguadas, asociadas a las vibraciones mecánicas de la proto-estrella de neutrones recién formada. Esta fase de la señal se denomina *ring-down*. Las 8 señales de la figura 4 pertenecen al mismo tipo de onda gravitacional, y tienen características cualitativas comunes. No obstante, se pueden dividir en 3 subtipos:



1. Para un núcleo con rotación lenta, la aparición de convección al poco tiempo de producirse el rebote, añade una componente de baja frecuencia a la señal típica del *ring-down* (ver figura 4, señal s11A1O07).
2. En el caso de que la rotación del núcleo colapsante sea moderadamente rápida, de forma que se siga originando un rebote hidrodinámico, la convección se suprime debido a la creciente influencia de gradientes de momento angular y, por tanto, no produce ninguna impronta en el *ring-down* post-rebote de la señal (ver figura 4, señales s11A3O13, s15A2O05, s20A2O09, s40A1O07).
3. Finalmente, si la rotación es suficientemente rápida, el núcleo rebota a densidades subnucleares o ligeramente supernucleares debido a los efectos incrementados de las fuerzas centrífugas. Esto se traduce en señales gravitacionales con más amplitud del pico de rebote y, en general, con frecuencias más bajas (ver figura 4, señal s40A3O13).

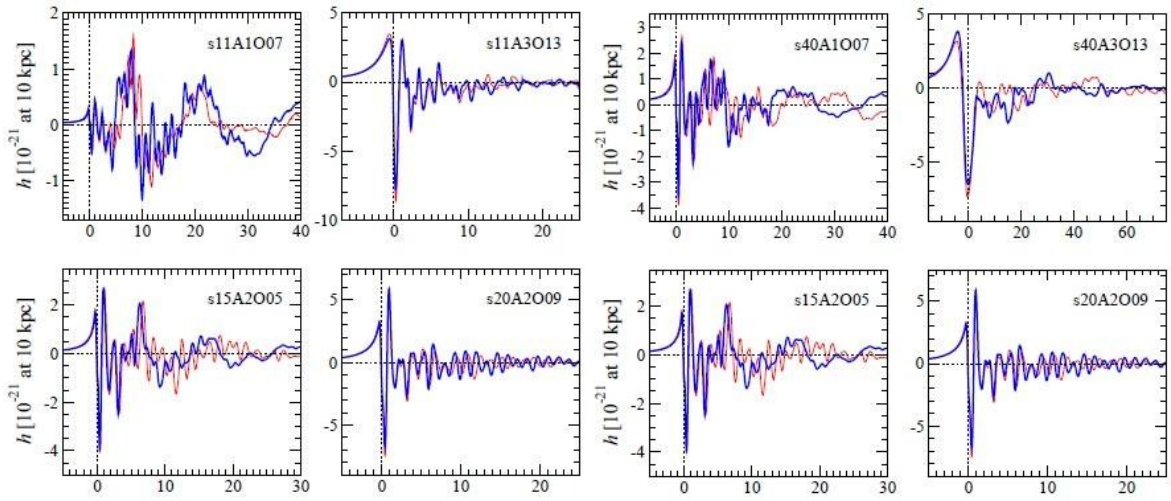


Fig 4. Algunas señales gravitacionales del catálogo de Dimmelmeier et al [3] que hemos utilizado en este trabajo. La diferencia en el color de las señales es la ecuación de estado utilizada para generarlas. Las señales están mostradas de forma que el pico principal (negativo) coincida con el tiempo en 0 s. Se representa la amplitud (o *strain*) de la onda gravitacional en función del tiempo,  $h$ , para un evento de colapso estelar situado a 10 kpc.

El trabajo empieza por tomar una base de datos con 128 señales reales de ondas gravitatorias [3] y entrenar un diccionario. Para este apartado nos centraremos solamente en el bloque de las señales de entrenamiento, un 80% del total, en este caso 100. Estas señales se encargarán de dotar al diccionario de la información necesaria para poder reconstruir otras señales similares lo mejor posible. Para generar el diccionario las señales han de ser previamente recortadas, normalizadas y reordenadas aleatoriamente, ya que el catálogo está ordenado por morfologías. Una vez generado, el entrenamiento consiste en resolver un problema de optimización con el diccionario preexistente.

Como hemos mencionado anteriormente, en este trabajo hemos utilizado ruido gaussiano simulado cuyo espectro en frecuencia se corresponde con la curva de sensibilidad de LIGO. Las señales están muestreadas a 16384 Hz (la frecuencia de muestreo de los detectores LIGO) y se han rellenado con ceros para que todas tengan la misma duración.

### 3. 2. Búsqueda de parámetros óptimos.

Una vez tenemos los diccionarios listos para su uso, debemos hallar los parámetros adecuados para reconstruir las señales de forma óptima. Estos parámetros son el número y la longitud de átomos de los diccionarios. Para estudiar esta dependencia, hemos entrenado diferentes diccionarios variando

estos parámetros. Para poder seleccionar un diccionario válido para reconstruir señales, empleamos cada uno de los que hemos generado con las 20 señales de validación (aproximadamente el 15% del catálogo). Al aplicar cada uno de los diccionarios obtendremos una reconstrucción y un error estadístico, en nuestro caso el MSE (*mean squared error*), que mide la diferencia entre la señal original y la reconstruida. Seleccionaremos el diccionario que minimice mejor el MSE para las señales de validación.

### 3.3. Resultados y análisis

#### 3.3.1. Método simple (M1)

Consiste en emplear únicamente diccionarios. Tras generar varios diccionarios con parámetros distintos, buscamos el parámetro propio del diccionario, que llamaremos  $\alpha$ , que mejores resultados da utilizando las señales de validación. Una vez determinado el  $\alpha$  óptimo, procedemos a obtener resultados en las señales test para comprobar la eficacia de este método.

#### 3.3.2. Método doble (minimización separada) (M2)

Consiste en combinar ROF con diccionarios pero no de forma simultánea. Los diccionarios generados son los mismos que ya hemos generado en el método simple: son reutilizables. Lo que debemos modificar es el parámetro propio de los diccionarios,  $\alpha$ , ya que el valor óptimo no va a ser el mismo. Esto se debe a que ahora, en lugar de enviar las señales con ruido a la reconstrucción con diccionarios, las enviaremos al algoritmo ROF, el cual nos devolverá las señales con un *denoising* parcial. Esto implica hallar el parámetro  $\lambda$  óptimo para cada señal de validación que minimice el MSE, un valor completamente ajeno al parámetro  $\alpha$  de los diccionarios. Inmediatamente después, las señales salientes de la aplicación de ROF son pasadas a los diccionarios para ejecutar un segundo *denoising*, esta vez minimizando el MSE mediante el parámetro  $\alpha$ . Las funciones de minimización trabajan con los métodos de forma independiente, y por tanto, cada parámetro dependerá únicamente del MSE.

#### 3.3.3. Método doble (minimización simultánea) (M3)

A diferencia del apartado anterior, en el que hemos explicado por qué cada parámetro se estudia de forma independiente, en esta tercera opción hemos incluido la optimización de las señales de validación en una función de minimización que busca valores a la vez tanto de  $\alpha$  como de  $\lambda$  para minimizar el MSE. La secuencia de procesamiento de las señales es la misma: primero pasamos las señales a ROF y después a los diccionarios. No obstante, ahora la minimización es multivariable, y por tanto el código se encargará de buscar una combinación de parámetros que no tiene por qué ser la misma que en el apartado anterior. La función de minimización ahora tiene en cuenta ambos parámetros al mismo tiempo, y por tanto, cada parámetro dependerá del otro, así como del MSE. Cabe mencionar que el coste computacional de minimizar el MSE con un optimizador multivariable es significativamente mayor, como hemos comprobado en la validación de las señales (más de 5 horas en un ordenador de 16 GB de RAM con un procesador Intel Core i7).

### 3.4. Test de los parámetros escogidos.

El objetivo de este trabajo es ver hasta dónde somos capaces de recuperar señales con ruido con las técnicas de *denoising* empleadas. En nuestro caso, el problema físico reside en recuperar la forma original de ondas con SNR muy pequeños. El apartado de la búsqueda de parámetros lo hemos llevado a cabo con SNR 20. Primero, por tanto, realizaremos la prueba de recuperar las señales test (las últimas 8 del catálogo) con el SNR establecido anteriormente. A partir de ahí, probaremos a ver

cómo afecta la reducción del SNR a valores menores a la reconstrucción de las señales. En esencia, estaremos comprobando la validez de las técnicas desarrolladas cuando intentemos recuperar señales que provengan de fuentes cada vez más y más lejanas. Para ello, emplearemos los 3 métodos explicados en el apartado anterior, obteniendo los parámetros óptimos para cada uno de ellos por separado, en cada una de las señales test.

## 4. Generación y entrenamiento de los diccionarios

El proceso de generación de los diccionarios tiene dos fases bien diferenciadas: por un lado, la extracción de fragmentos de las señales de entrenamiento (80% del total) para crear una matriz que llamaremos *diccionario*; y por otro lado, el *entrenamiento* del diccionario, *i.e.*, resolver el problema de minimización.

$$(U^*, V^*) = \arg \min_{U, V} \frac{1}{2} \|X - UV\|_2^2 + \alpha \|U\|_1 \quad (7)$$

$$\text{subject to } \|V_k\|_2 = 1 \text{ for all } 0 \leq k < n_{\text{atoms}}$$

Esta ecuación es la misma que (6), pero con una notación diferente (la que emplea la función MiniBatch de Python) [14]. En este caso, V es el diccionario y U es el vector de coeficientes que debemos hallar. La  $\alpha$  que acompaña al término de la norma  $L_1$  es el parámetro de regularización o el parámetro del método propio que buscaremos utilizando las 20 señales de validación.

Empezamos generando el diccionario inicial. Para ello hemos escogido 100 señales de las 128 disponibles. Estas 100 señales son aleatorias dentro del total, y por tanto, agrupan las diferentes morfologías del catálogo. Una vez tenemos las 100 señales seleccionadas, procedemos a recortarlas alrededor del pico de la onda. A lo largo de todo el experimento hemos probado a variar este parámetro entre diferentes valores: 512, 1024 y 2048. El último valor fue desechado rápidamente debido a que agrupaba demasiados ceros a una vez se cubría la parte significativa de la señal. Los valores 1024 y 512 son validados más adelante, como veremos posteriormente en el apartado de resultados. Llegados a este punto, tenemos una matriz 100x512 .

El siguiente paso es extraer trozos de la señal o *patches* de forma aleatoria, de una longitud concreta, que será un parámetro del método a determinar. Para ello generamos dos números aleatorios: uno que seleccione una señal y otro que seleccione un índice a partir del cuál recortar el *patch*. Obtenemos entonces una matriz con, por ejemplo 300 filas o átomos y 128 muestras (longitud de los átomos). Este es nuestro diccionario inicial. Antes de proceder al entrenamiento, normalizamos todos los *patches* seleccionados para acelerar el proceso. Una vez hecho esto, entrenamos el diccionario. El entrenamiento como tal pasa por resolver el problema de minimización y resolvemos este problema con la función *MiniBatchDictionaryLearning* [14] del paquete de científico de Python *scikit-learn* [13]. Esta función tiene por cabecera

```
MiniBatchDictionaryLearning(n_components=number_atoms, alpha = 1.2/np.sqrt(number_atoms),
n_iter=50000, batch_size = 4, fit_algorithm = 'lars')
```

y los parámetros que hemos utilizado para el entrenamiento son:

- $n\_components = N$ , con N el número de átomos
- $\alpha = 1.2/\sqrt{N}$ , el 1.2 es un resultado experimental [15] que favorece resultados *sparse* (muchos ceros en la señal) en los experimentos, mientras que la raíz cuadrada es un término de

normalización. Este  $\alpha$  no es el parámetro propio del método, es simplemente un parámetro numérico de la función predefinido.

- `n_iter` = el número de iteraciones lo hemos fijado a 50000 para asegurar que los átomos del diccionario sean suaves.

- `batch_size` = divide el problema en trozos más pequeños para acelerar el proceso

- `fit_algorithm` = el método matemático que se utiliza para resolver el problema, en este caso, *least angle regression*.

Los parámetros del entrenamiento han sido buscados de forma heurística para poder obtener átomos lo más suaves posible. En la siguiente imagen mostramos 25 átomos del diccionario de 300 átomos y 128 muestras.

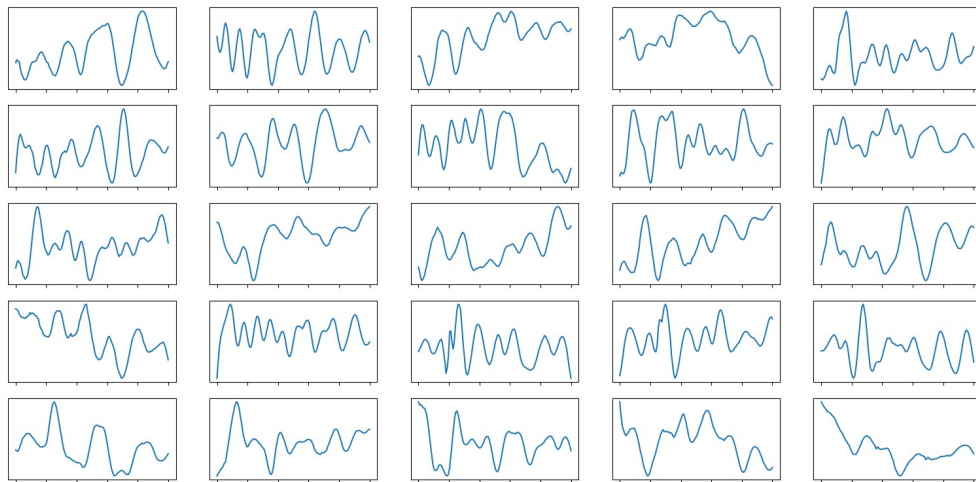


Fig 5. Selección aleatoria de átomos representativos del catálogo de señales gravitacionales de colapso estelar [3]

## 5. Reconstrucción de las señales

Cuando ya tenemos nuestros diccionarios preparados, o el algoritmo ROF compilado y listo para ser usado, podemos reconstruir las señales originales a partir de las medidas. Para ello hemos de coger una señal y escalarla al SNR adecuado, sumarle el ruido y finalmente, normalizarla. La duración de todas las señales es de 1 s. La reconstrucción de cada método sigue el siguiente esquema:

### 5. 1. Reconstrucción por ROF

Compilamos el algoritmo en Fortran para poder ser importado a Python con *f2py*. La velocidad del código es mucho mayor de este modo y la eficiencia a la hora de minimizar el error en la validación aumenta. Luego enviamos la señal inyectada en ruido normalizada a la función importada en Python con los siguientes parámetros:

ROF(Señal,  $h$ ,  $\beta$ ,  $\lambda$ , tol)

- $h$  = step size del algoritmo
- $\beta$  = parámetro numérico de valor pequeño; en el experimento su valor es de 0.01
- $\lambda$  = parámetro de regularización del método
- tol = tolerancia del método; fija el error numérico a partir del cual se detiene la iteración.

Esta función nos devuelve una señal sin ruido, pero todavía susceptible de ser mejorada, que después nosotros pasaremos a los diccionarios para hacer un segundo *denoising*.

## 5. 2. Reconstrucción por diccionarios

Para recuperar la señal original debemos extraer *patches* de la señal con ruido, de forma que estos cubran toda la señal. Después, los enviamos a la función `SparseCoder` [11] para que los transforme en *patches* de la señal original. Finalmente reconstruimos la señal a partir de los *patches* limpios. Esta función requiere de los parámetros

```
SparseCoder(dictionary=D, transform_algorithm='lasso_lars', transform_alpha = alpha)
```

donde la  $\alpha$  es el parámetro a minimizar del método.

## 6. Búsqueda de los parámetros óptimos (validación)

La validación o búsqueda de los parámetros de los métodos es la parte del experimento que nos permite seleccionar los parámetros ideales de un diccionario para poder reconstruir señales de la mejor forma posible. En este apartado vamos a explicar el proceso seguido para hallar el valor de estos parámetros para reconstruir nuestras señales y qué estimador hemos empleado para obtenerlos: el MSE (*mean squared error*) y el SSIM (*structural similarity index*). Estos estimadores estadísticos se encargan de medir la similitud entre la señal original y la reconstruida. En particular, el MSE es el error estadístico que recoge la diferencia total entre ambas señales término a término y por tanto, cuanto más cercano a cero, más similitud. Por otro lado, el SSIM tiene en cuenta factores estructurales y de esta manera un SSIM de 1 sería máxima similitud, mientras que 0 sería mínima. [5]

Es importante tener en cuenta que tanto en esta sección como en la siguiente de resultados, todos los análisis se han llevado a cabo para las muestras de las señales 3700-4600 que en general cubren toda la parte significativa de la mayoría de señales del catálogo. Esto implica que a la hora de realizar la minimización del error, únicamente se ha tenido en cuenta la parte de la señal entre los índices 3700 y 4600 de las 16384 muestras totales, obviando el resto de la señal. Pese a que algunas de ellas se ven ligeramente recortadas, elegir estos índices para evitar llevar a cabo el análisis con demasiados ceros tanto a la izquierda como a la derecha de la parte significativa de la señal es algo imprescindible. Esto implica que todos los valores de los parámetros óptimos, así como los resultados posteriores, están basados en cálculos en relación al MSE para la parte significativa de la señal exclusivamente, en los índices anteriores. En el caso de haber utilizado toda la, habríamos obtenido “mejores” valores de los dos estimadores (con MSEs en un orden de magnitud de  $10^{-5}$  y SSIMs superiores a 0.95, ya que más del 90% de la información de la señal coincide con las reconstrucciones, al ser todo ceros). No obstante, la parte significativa de la señal, que es la que más nos interesa reconstruir con efectividad, estaría perdiendo peso en la minimización al ser menos del 10% de la señal, y por tanto acabaría reconstruyéndose peor.

### 6. 1. Parámetros de los diccionarios

A la hora de decidir qué diccionario es el mejor para reconstruir señales, debemos tener en cuenta varios factores. Por un lado, cada diccionario posee unos parámetros característicos que lo definen: el número de átomos y la longitud de estos. Además, hemos añadido al estudio un parámetro extra: la longitud de las señales recortadas antes de extraer los *patches* con los que generamos posteriormente el diccionario. Para decidir qué valores son los óptimos hemos considerado el MSE como estimador principal para seleccionar un diccionario. Todo el trabajo ha sido llevado a cabo

obteniendo resultados en base a la minimización del MSE. Como veremos a continuación, hemos calculado también los resultados en función del SSIM para tener más información, pero por lo general el resultado óptimo para las reconstrucciones se ha obtenido minimizando el MSE y no maximizando el SSIM.

En cuanto a los parámetros propios de los diccionarios, hemos considerado los valores 512 y 1024 para el número de muestras a los que recortamos las señales; 64, 128 y 256 para la longitud de los átomos; y 300 y 500 para el número de átomos por diccionario. En lugar de utilizar una función de minimización, como hemos hecho en la optimización de los parámetros de ambos métodos, en este caso hemos creado 12 diccionarios en total, uno para cada combinación posible de estos parámetros, y hemos analizado los resultados en torno al MSE que ofrecen las señales reconstruidas con cada diccionario. El número de muestras fue fijado directamente en 512 cuando todos los diccionarios operaban mejor que los de 1024. En cuanto a la longitud de los átomos, 64 es una longitud demasiado corta y no recoge suficiente información de las señales para poder hacer buenas reconstrucciones. Por tanto, hemos centrado nuestra búsqueda en los diccionarios con longitudes 128 o 256; y 300 o 500 átomos. Los dos diccionarios que mejor reconstruyen las señales de validación (señales 100-120 del catalogo, [3]) son los de 300 átomos y 128 muestras de longitud (300,128) y el de 500 átomos y 256 muestras de longitud, (500,256). A continuación, en la figura 6 adjuntamos las gráficas del MSE y SSIM para cada uno de los dos diccionarios.

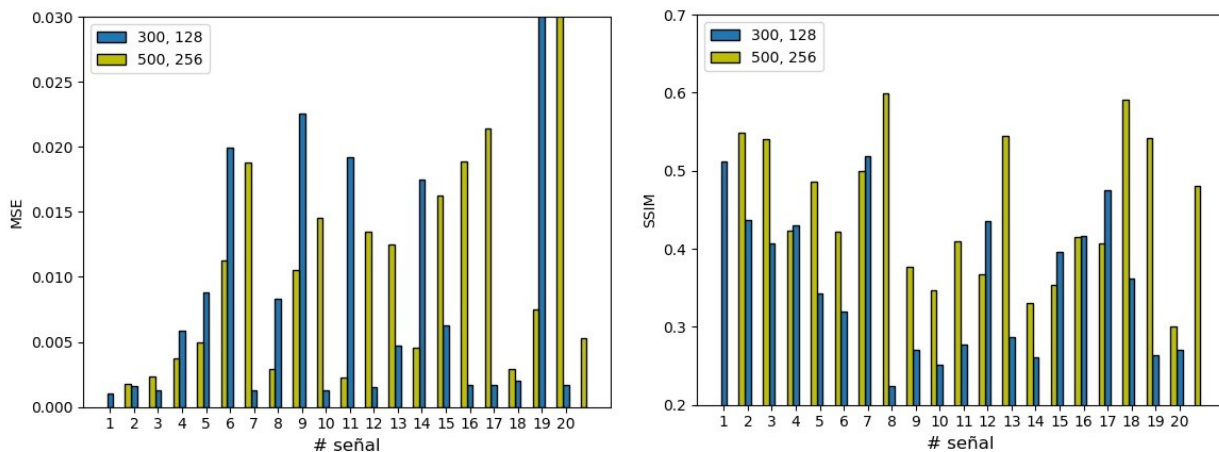


Fig 6. Valores del MSE (Izquierda) y SSIM (derecha) para los 2 diccionarios.

Numéricamente hemos calculado el MSE medio que ofrece cada diccionario a través de las 20 señales de validación:

$$\begin{aligned} \text{MSE}(300, 128) &= 0.0082 & \text{SSIM}(300, 128) &= 0.36 \\ \text{MSE}(500, 256) &= 0.0124 & \text{SSIM}(500, 256) &= 0.45 \end{aligned}$$

Estos valores relativamente altos del MSE y bajos del SSIM se deben a la utilización de una  $\alpha$  fija de 0.5 (lejos del valor óptimo en la mayoría de señales). Aquí se muestra que el diccionario (300,128) minimiza mejor el MSE pero arroja valores peores para el SSIM medio. Hemos seleccionado el diccionario (300,128) por dos motivos. El primero es que en la sección de resultados, hemos utilizado el SSIM y MSE para las reconstrucciones y utilizar la minimización del MSE en lugar de la maximización del SSIM ofrecía resultados más satisfactorios. El segundo motivo es el coste computacional: utilizar el diccionario (300,128) resulta significativamente más eficiente en cuanto al tiempo de computación. Al tener el otro diccionario 500 átomos del doble de longitud, los tiempos de computación podían llegar a ser del doble o el triple que para la versión (300,128). Por estas razones hemos seleccionado el diccionario (300,128) para llevar a cabo el estudio de los resultados en las señales test.



## 7. Test y resultados

Una vez hemos calculado los parámetros óptimos para los diccionarios, podemos proceder a ejecutar los métodos de minimización con dichos parámetros en las señales test para comprobar su efectividad. Como comentábamos en el apartado 3.3., nuestro objetivo es ver cómo afecta una disminución progresiva del SNR en las señales a nuestros resultados. Los resultados se han llevado a cabo reconstruyendo 8 señales del catálogo tipo test, que no han sido incluidas en el entrenamiento de los diccionarios, ni tampoco en la validación de las señales para la búsqueda de los parámetros óptimos. Para obtener estos resultados nos hemos servido del MSE, que es el estimador estadístico en el que hemos basado la búsqueda de los parámetros óptimos de los diccionarios.

### 7. 1. Señales test y distancias

En nuestro trabajo, las señales test que hemos empleado para llevar a cabo las reconstrucciones y el análisis de los resultados están representadas en la figura 7. Se puede ver claramente la morfología de las formas de onda con las que tratamos (ver apartado 3.1., figura 4), las señales 2 y 4 pertenece al subtipo 3, con un pico principal de amplitud significativamente mayor y menores frecuencias; mientras que la señal 6 es claramente un ejemplo del subtipo 1 y el resto son ejemplos del subtipo 2.

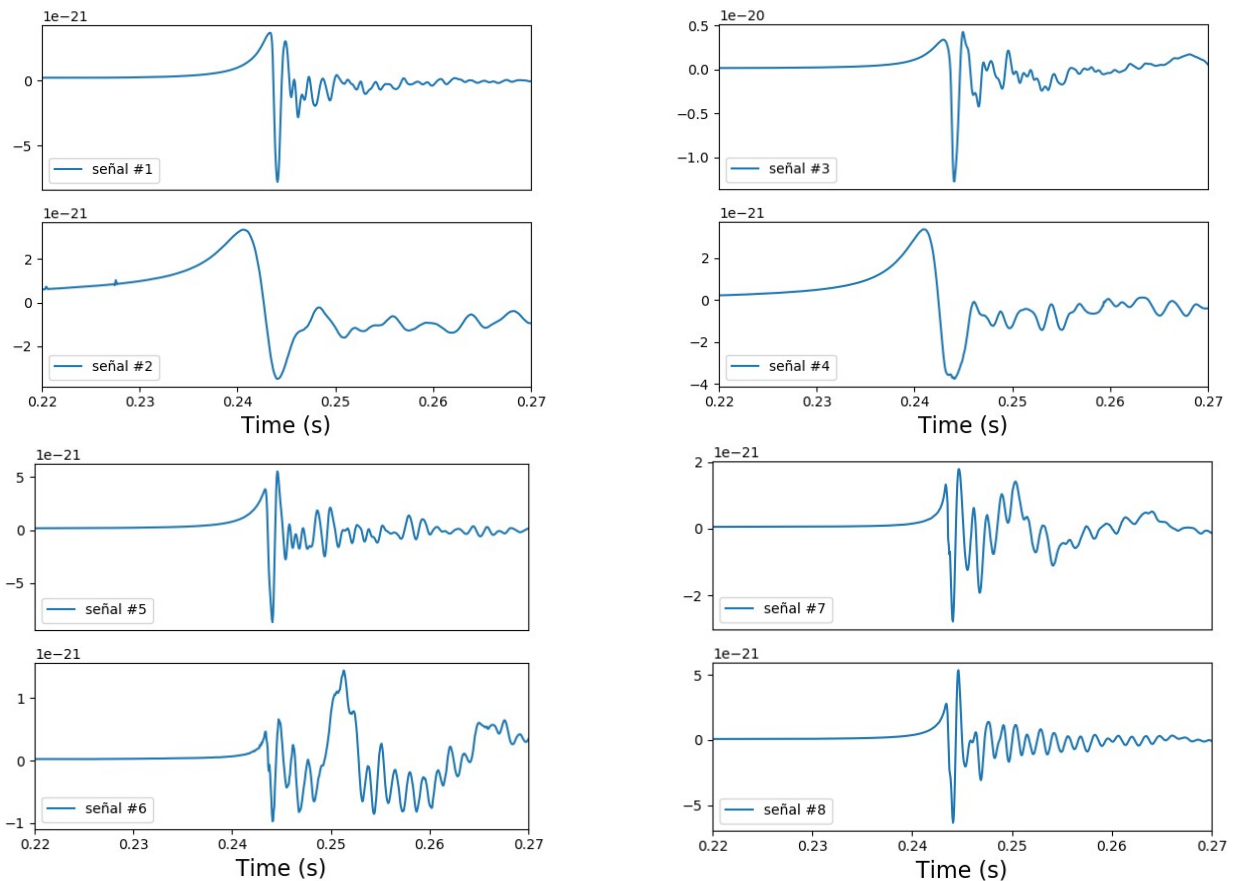


Fig 7. Señales tipo test (120-128) del catálogo de ondas gravitacionales de colapso estelar [3]. Se representa la amplitud de la onda gravitacional (sin ruido) en un intervalo de 0.05 s de tiempo.

Con tal de ilustrar de forma más física lo que implica reconstruir estas señales test a diferentes SNR, en la Tabla 1 se adjuntan las distancias a las que se encuentra cada una de estas señales en función del SNR al que han sido escaladas. Estas distancias han sido calculadas a partir del factor de escalado de cada SNR. Para obtenerlas, hemos tenido en cuenta que las señales del catálogo de

Dimmelmeier et al [3] están situadas a 10 kpc. Con la función de Python empleada para escalar las señales al SNR correspondiente obtenemos un factor de escalado, que es el que se emplea para calcular las distancias finales en base a la inicial de 10 kpc.

Señal	Distancia SNR 20 (kpc)	Distancia SNR 10 (kpc)	Distancia SNR 8 (kpc)
#1	39.37	78.74	98.43
#2	20.39	40.79	50.99
#3	61.12	122.23	153.79
#4	24.82	49.64	62.05
#5	46.57	93.13	116.41
#6	10.46	20.93	26.16
#7	17.49	34.98	43.72
#8	35.96	71.92	89.90

Tabla 1. Distancias de las ocho señales test junto con los correspondientes valores del SNR.

## 7. 2. Resultados aplicando M1

Procedemos a ejecutar el método M1 con el parámetro óptimo encontrado anteriormente, y estudiamos el efecto que tiene disminuir el SNR en los resultados.

Señal	SNR 20		SNR 10		SNR 8	
	$\alpha_{\text{opt}}$	$\text{MSE} \cdot 10^{-3}$	$\alpha_{\text{opt}}$	$\text{MSE} \cdot 10^{-3}$	$\alpha_{\text{opt}}$	$\text{MSE} \cdot 10^{-3}$
#1	0.296	0.6	0.208	2.5	0.203	4.1
#2	0.158	1.6	0.160	4.2	0.159	5.8
#3	0.200	1.0	0.186	2.8	0.184	4.3
#4	0.356	2.0	0.268	4.6	0.235	5.9
#5	0.159	1.3	0.268	3.9	0.243	5.9
#6	0.130	4.6	0.143	11.4	0.147	14.7
#7	0.160	2.1	0.246	4.5	0.240	6.4
#8	0.167	1.5	0.309	2.5	0.239	4.3

Tabla 2. Resultados de M1 en función del MSE para valores descendientes del SNR

Como podemos ver en la Tabla 2, el valor óptimo de  $\alpha$  está en el intervalo (0.1, 0.4) en todos los casos. Las diferencias entre los parámetros para SNR 20 y 10 son algo más grandes, pero el cambio entre SNR 10 y 8 es mínimo. Por otro lado, el MSE se sitúa en valores relativamente altos: el peor caso (#6) lo encontramos en casi 0.005 para SNR 20 con un respectivo valor de 0.015 para SNR 8. El mejor caso lo vemos, en este método, para la señal #1, que proporciona un valor del MSE de tan solo 0.0006.

## 7. 3. Resultados aplicando M2

Análogamente al apartado anterior, aplicamos el método doble con los parámetro óptimos obtenidos a partir de M2 y presentamos en la Tabla 3 los siguientes resultados, para los tres valores de SNR propuestos anteriormente:



	SNR 20		SNR 10			SNR 8			
Señal	$\alpha_{\text{opt}}$	$\lambda_{\text{opt}}$	$\text{MSE} \cdot 10^{-3}$	$\alpha_{\text{opt}}$	$\lambda_{\text{opt}}$	$\text{MSE} \cdot 10^{-3}$	$\alpha_{\text{opt}}$	$\lambda_{\text{opt}}$	$\text{MSE} \cdot 10^{-3}$
#1	0.535	5.516	0.53	0.212	5.962	2.39	0.350	5.911	3.61
#2	0.865	3.000	0.43	0.967	2.354	1.69	0.981	3.000	2.68
#3	0.704	5.635	0.52	0.342	5.912	2.11	0.215	5.972	4.72
#4	1.108	3.005	0.56	1.245	3.318	1.89	0.507	3.316	3.36
#5	0.332	5.702	1.11	0.272	6.030	3.42	0.201	5.927	6.02
#6	0.280	3.298	2.78	0.252	3.429	7.81	0.268	3.505	9.75
#7	0.230	7.204	2.48	0.326	7.244	4.79	0.319	7.239	11.41
#8	0.233	6.062	1.37	0.353	5.984	2.71	0.200	6.071	7.21

Tabla 3. Resultados de M2 en función del MSE y para valores descendientes del SNR

Para el procedimiento M2, al aplicar ROF y tener un parámetro previo a  $\alpha$  que modifica las señales, concretamente el parámetro propio de ROF  $\lambda$ , los valores de  $\alpha_{\text{opt}}$  se sitúan en un rango diferente al de M1. Vemos que algunos llegan a superar la unidad. Los valores del MSE son significativamente mejores que para M1. De hecho, el método M1 arroja un valor medio del MSE de 0.0018, mientras que M2 proporciona 0.0012. Por otro lado, el valor del parámetro óptimo  $\lambda$ , fijado entre 0.01 y 10 es siempre mayor que la unidad. De nuevo podemos ver que para los casos de SNR 10 y 8 el valor óptimo del parámetro de cada método varía ligeramente entre uno y otro SNR.

#### 7.4. Resultados aplicando M3

A pesar de que a efectos prácticos aplicar M3 y M2 es repetir el mismo procedimiento, ya que el código para reconstruir la señal es idéntico (aplicar primero ROF y luego diccionarios), la minimización del MSE para las señales se lleva a cabo de distinta forma y, consecuentemente, los parámetros óptimos son distintos. Por este motivo, hemos decidido reconstruir las señales test con los nuevos parámetros, con tal de ver si hay diferencias significativas y determinar si M3 es nuestro mejor método en general. Los resultados obtenidos por el método M3 (ver Tabla 4) son los siguientes:

	SNR 20		SNR 10			SNR 8			
Señal	$\alpha_{\text{opt}}$	$\lambda_{\text{opt}}$	$\text{MSE} \cdot 10^{-3}$	$\alpha_{\text{opt}}$	$\lambda_{\text{opt}}$	$\text{MSE} \cdot 10^{-3}$	$\alpha_{\text{opt}}$	$\lambda_{\text{opt}}$	$\text{MSE} \cdot 10^{-3}$
#1	2.124	3.025	0.41	1.403	4.612	2.35	0.921	10.000	4.09
#2	0.873	2.986	0.46	1.031	0.951	1.85	1.127	0.941	2.28
#3	1.977	3.003	0.26	2.116	1.360	1.07	1.753	2.982	1.96
#4	1.114	2.976	0.60	1.163	0.906	1.59	1.570	0.900	2.13
#5	1.384	1.494	0.86	1.466	9.936	3.14	1.944	3.105	5.71
#6	1.031	2.967	1.21	1.363	0.836	4.03	1.198	0.875	5.85
#7	1.096	3.030	2.00	1.420	1.764	14.32	1.379	4.141	23.52
#8	1.373	3.034	0.71	1.703	7.264	1.20	1.457	7.696	2.21

Tabla 4. Resultados de M3 en función del MSE y para valores descendientes del SNR

Con los resultados de la Tabla 4 podemos asegurar que realizar una minimización simultánea de los parámetros óptimos de cada método sin duda supone una mejoría. El valor medio del MSE se ha reducido de 0.0012 a 0.0008, y en este caso, hemos llegado a reconstruir la señal #3 con un MSE de tan solo 0.00026. En el siguiente apartado realizamos una comparación de los métodos para los diferentes SNRs.

## 7. 5. Comparación gráfica de los tres métodos

Para poder ver la mejoría que supone la variación del método en cada SNR, ilustramos a continuación los resultados con reconstrucciones visuales para el peor y el mejor de nuestros resultados. Tomaremos los correspondientes escenarios para M1, separados por SNR, y adjuntaremos las correspondientes mejoras que proporciona cambiar a los métodos M2 y M3.

### 7. 5. 1. Mejor escenario a SNR 20: señal #3 (61.12 kpc)

El método M3 arroja un MSE de 0.00026 para la señal #3, lo que implica la mejor reconstrucción para SNR 20 de todos nuestros resultados. A pesar de ser la distancia al origen de la fuente más grande de las señales test (61.12 kpc), la reconstrucción puede haber sido la mejor debido a que las señales utilizadas en el entrenamiento de los diccionarios eran morfológicamente mucho más parecidas a esta señal que al resto. La aplicación de M3 a esta señal ha dado el mejor resultado de nuestro trabajo: visualmente se puede apreciar en la figura 8, ya que la reconstrucción es prácticamente perfecta en toda la señal salvo un pequeño margen de error inmediatamente antes del descenso del pico principal alrededor de 0.242 s y en las pequeñas oscilaciones posteriores antes del tiempo 0.25 s.

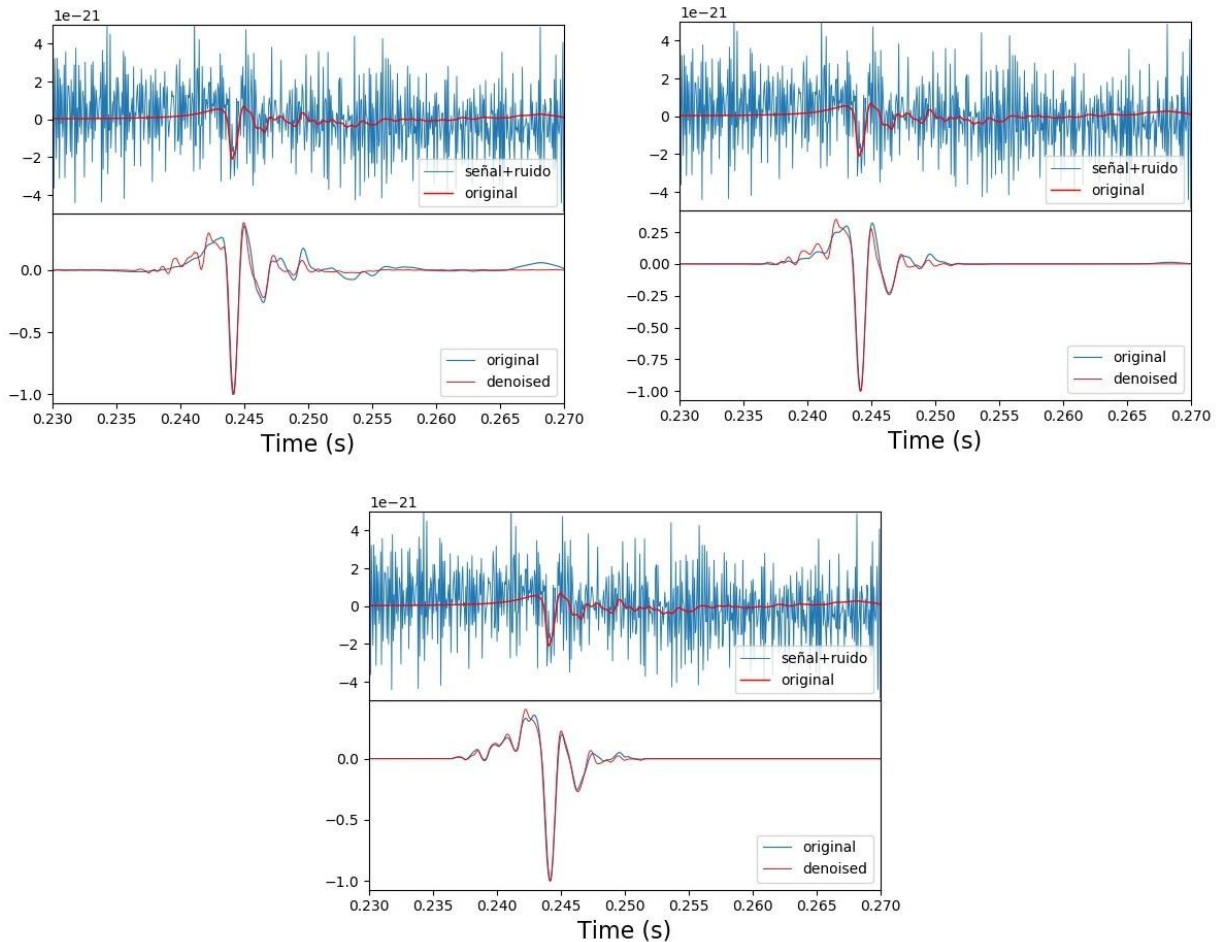


Fig 8. Reconstrucciones de la señal #3 con SNR 20 para los métodos M1 (arriba, izquierda), M2 (arriba, derecha) y M3 (abajo).

### 7. 5. 2. Peor escenario a SNR 20: señal #6 (10.46 kpc)

El método M1 proporciona un MSE para la señal #6 de 0.0046, que implica la peor reconstrucción para SNR 20. Es notable que aunque esta señal es la más cercana de las estudiadas, encontrándose a 10.46 kpc, el resultado del *denoising* es el peor. Esto sucede porque la morfología particular de esta señal aparece con muy poca frecuencia en el catálogo y más particularmente, en el bloque de señales de entrenamiento. Por este motivo, aplicar M1, que implica utilizar únicamente diccionarios, ofrece una reconstrucción significativamente peor que M2 y M3, en la que hay una clara mejora al pasar la señal previamente por el algoritmo ROF (ver figura 9). Estos resultados se corresponden con los MSEs de las tablas anteriores, donde pasamos de 0.0046 a 0.00278 y 0.00121. Visualmente se puede comprobar que la reconstrucción por M3 reconstruye perfectamente la señal hasta después del pico principal situado alrededor de 0.25 s, mientras que las pequeñas ondulaciones posteriores se recuperan visiblemente peor.

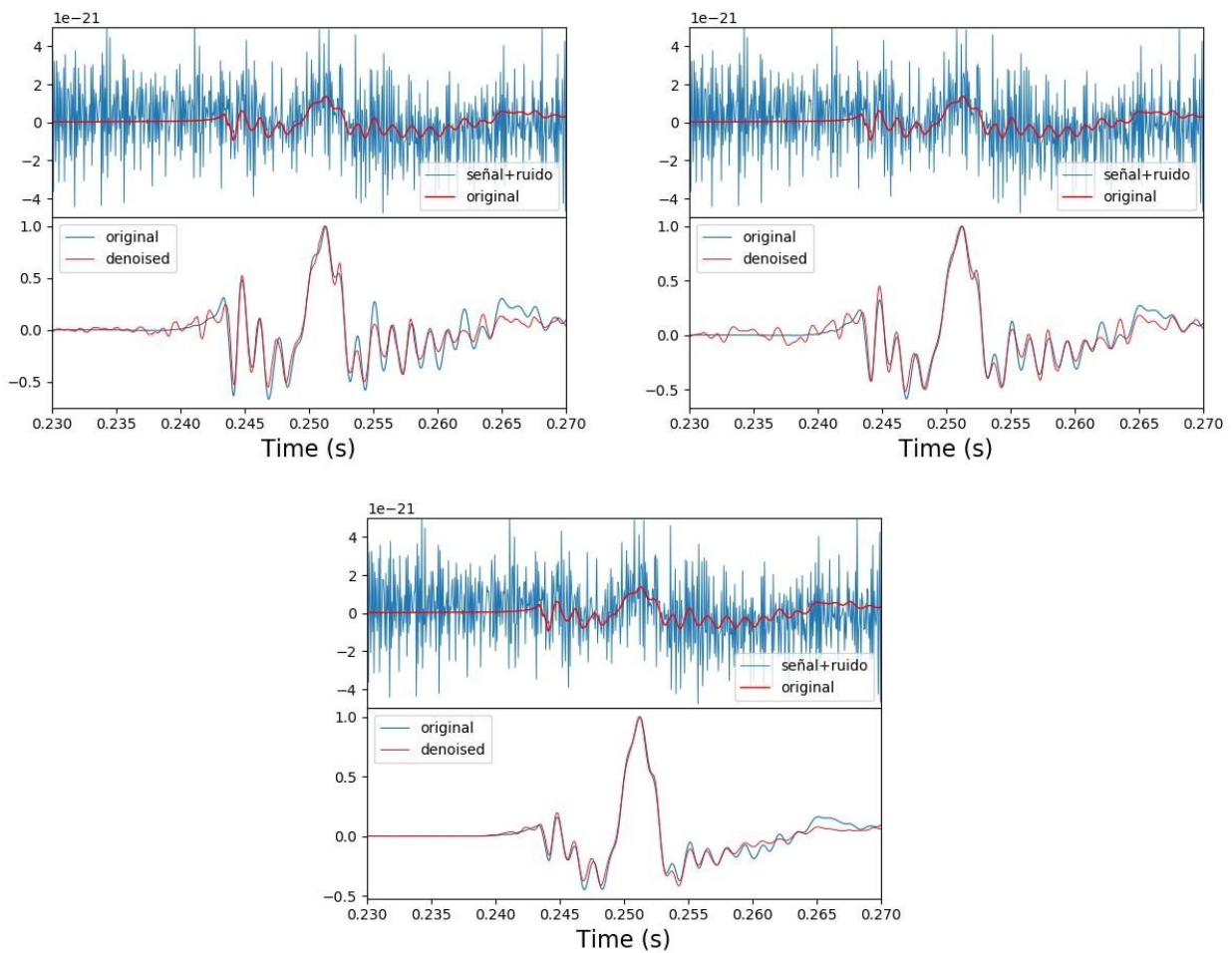


Fig 9. Reconstrucciones de la señal #6 con SNR 20 para los métodos M1 (arriba, izquierda), M2 (arriba, derecha) y M3 (abajo).

### 7. 5. 3. Mejor escenario a SNR 10: señal #8 (71.92 kpc)

A pesar de que la mejor reconstrucción se da, de nuevo, para la señal #3, hemos decidido adjuntar la representación de la señal #8, ya que es la segunda más fiel a la señal original y la diferencia es de apenas 0.00013 entre ambos MSE; de este modo, podemos ver la reconstrucción de otra señal test. Esta señal tiene una morfología ampliamente representada en el catálogo: un pico inicial negativo (asociado al instante del rebote de las capas más internas de la estrella) seguido de uno positivo de amplitud muy pequeña, ambos seguidos de las oscilaciones amortiguadas de la proto-estrella de

neutrones resultante. Por estar bien representada en el bloque de señales de entrenamiento, los tres métodos son capaces de reconstruirla bien, aunque M3 ofrece el MSE menor (ver figura 10). Tanto para M1 como para M2, el SNR 10 induce a que se generen demasiadas oscilaciones, cosa que se aprecia en el tramo alrededor del tiempo 0.24 s, antes del pico principal. M3 consigue mejorar los resultados en este tramo, aunque a pesar de ello, se sigue viendo que después de las oscilaciones amortiguadas, la reconstrucción no es del todo limpia, ya que en la parte de amplitud cero se ven pequeñas oscilaciones generadas por el ruido.

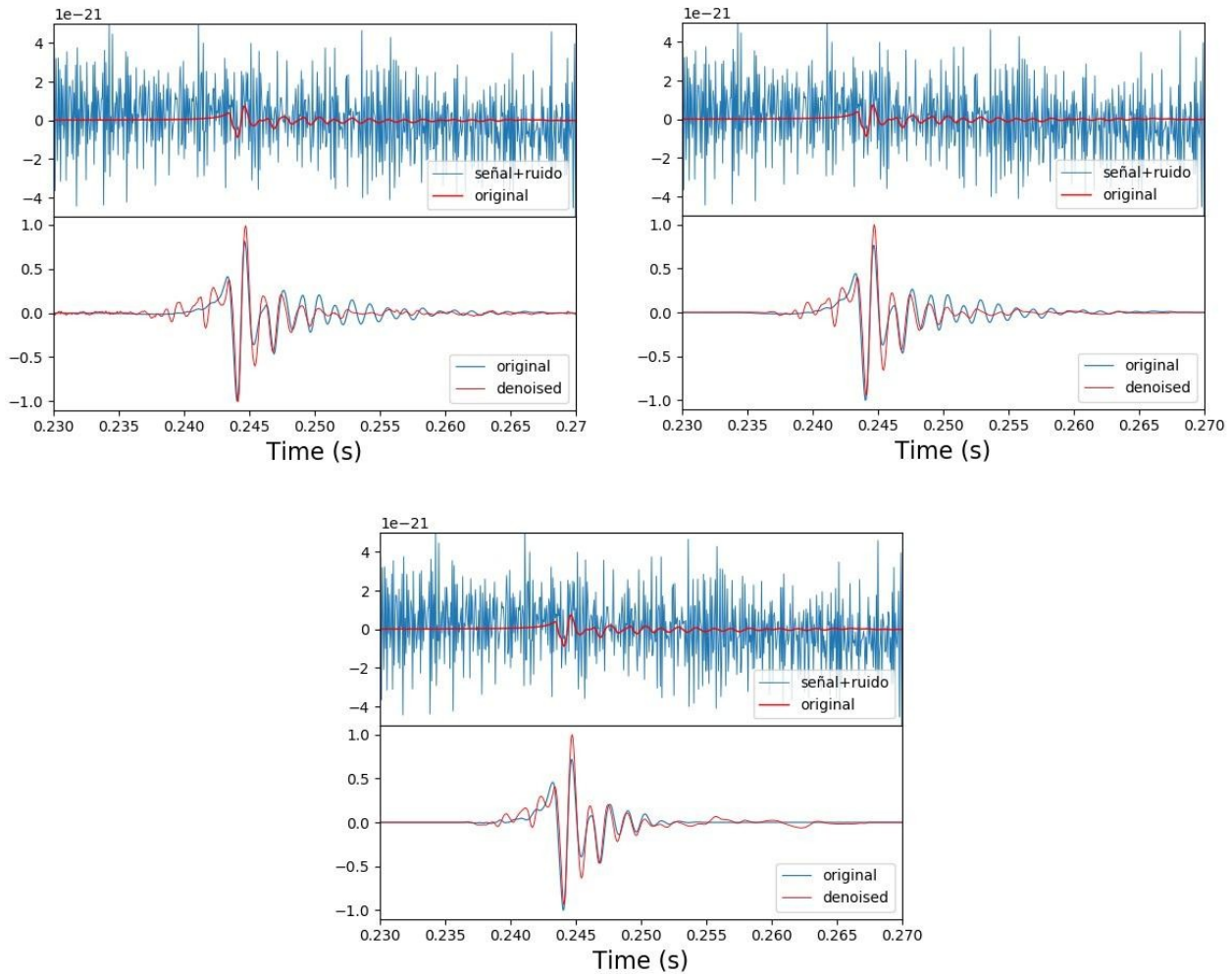


Fig 10. Reconstrucciones de la señal #8 con SNR 10 para los métodos M1 (arriba, izquierda), M2 (arriba, derecha) y M3 (abajo).

#### 7. 5. 4. Peor escenario a SNR 10: señal #7

Por otro lado, el método M3 es el que también produce la peor reconstrucción en general, correspondiente a la señal #7, con un MSE de 0.0143, el máximo valor obtenido. Aunque la señal #6 también se reconstruye mal a este SNR, especialmente por M1 que arroja un MSE de 0.0114, los motivos por los que puede suceder están comentados ya en el apartado 7.2.3.2., por lo que hemos escogido la señal #7 para incluir en este apartado, de tal forma que podemos ver la reconstrucción de otra de las señales test. El motivo por el que la reconstrucción no es buena es el mismo que para la señal #6: la morfología no se encuentra ampliamente representada en el bloque de las señales de entrenamiento. Además del pico principal, las oscilaciones amortiguadas son relativamente acusadas en comparación a otras señales y, adicionalmente, alrededor de  $t=0.25$  s hay dos picos que generan un valle, antes de volver a caer y finalizar la morfología en unas oscilaciones amortiguadas algo irregulares (ver figura 11).

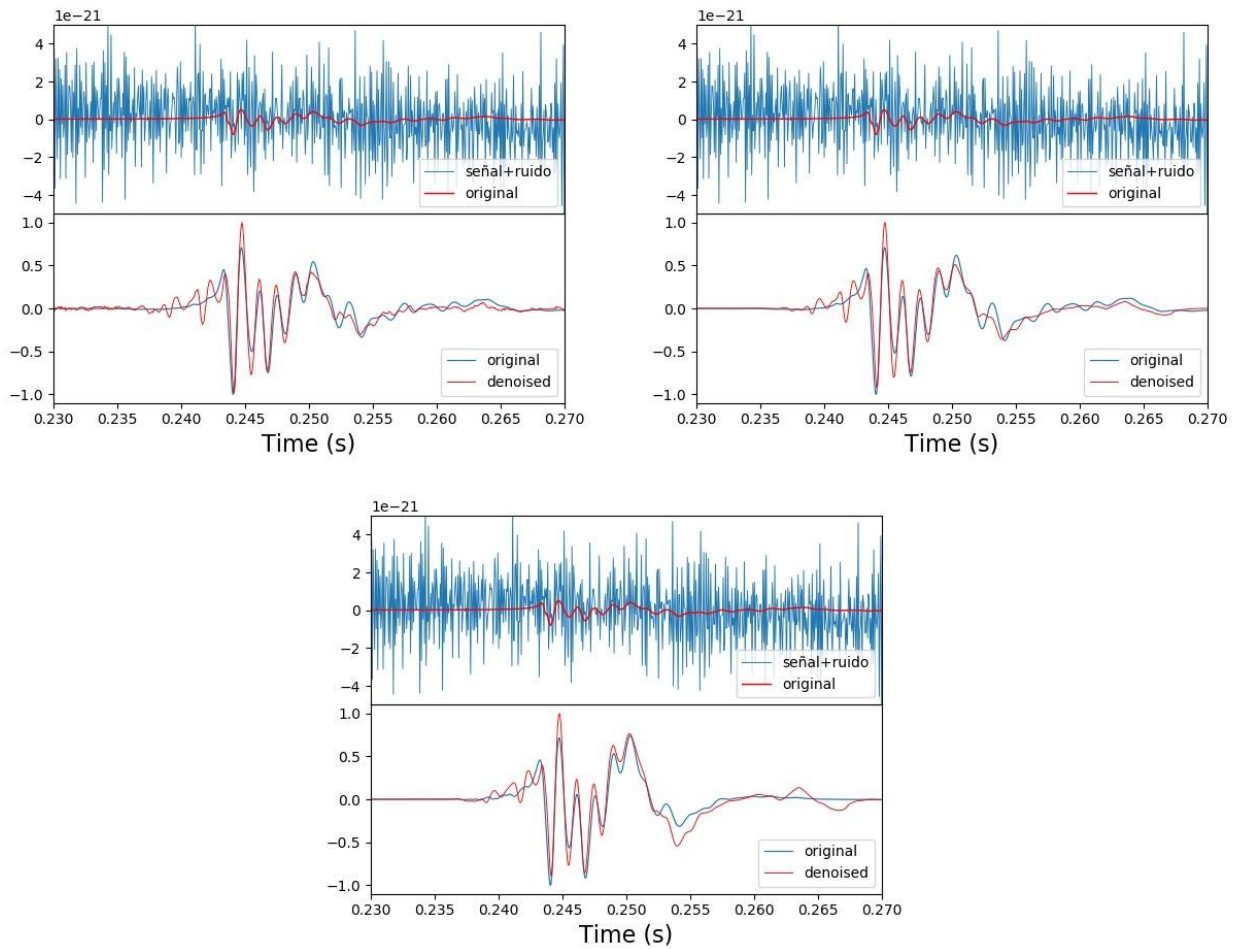


Fig 11. Reconstrucciones de la señal #7 a SNR 10 para los métodos M1 (arriba, izquierda), M2 (arriba, derecha) y M3 (abajo).

### 7. 5. 5. Mejor y peor escenario a SNR 8

Visualmente, en las gráficas de la figura 12 se puede apreciar el efecto en la reconstrucción con un SNR tan pequeño. No obstante, al no haber gran diferencia entre SNR 10 y 8, hemos adjuntado la reconstrucción de señales previamente mostradas, esta vez con el SNR más bajo estudiado, con tal de apreciar diferencias en esta variación. Los motivos por los que las reconstrucciones no son buenas están comentados en los apartados anteriores.

Como mejor reconstrucción, la señal #3 con M3 ofrece un MSE de 0.00196. Este resultado es mejorable. Ya en las reconstrucciones por M1 y M2 se observan estructuras numéricas generadas por el ruido considerables, sobre todo antes del pico principal. M3 ofrece la mejor reconstrucción, pero salvando el pico principal que se sigue reconstruyendo bien y la parte de ceros a izquierda y derecha de la onda, el *denoising* del resto de la morfología es significativamente peor que como hemos podido ver en SNR 20.



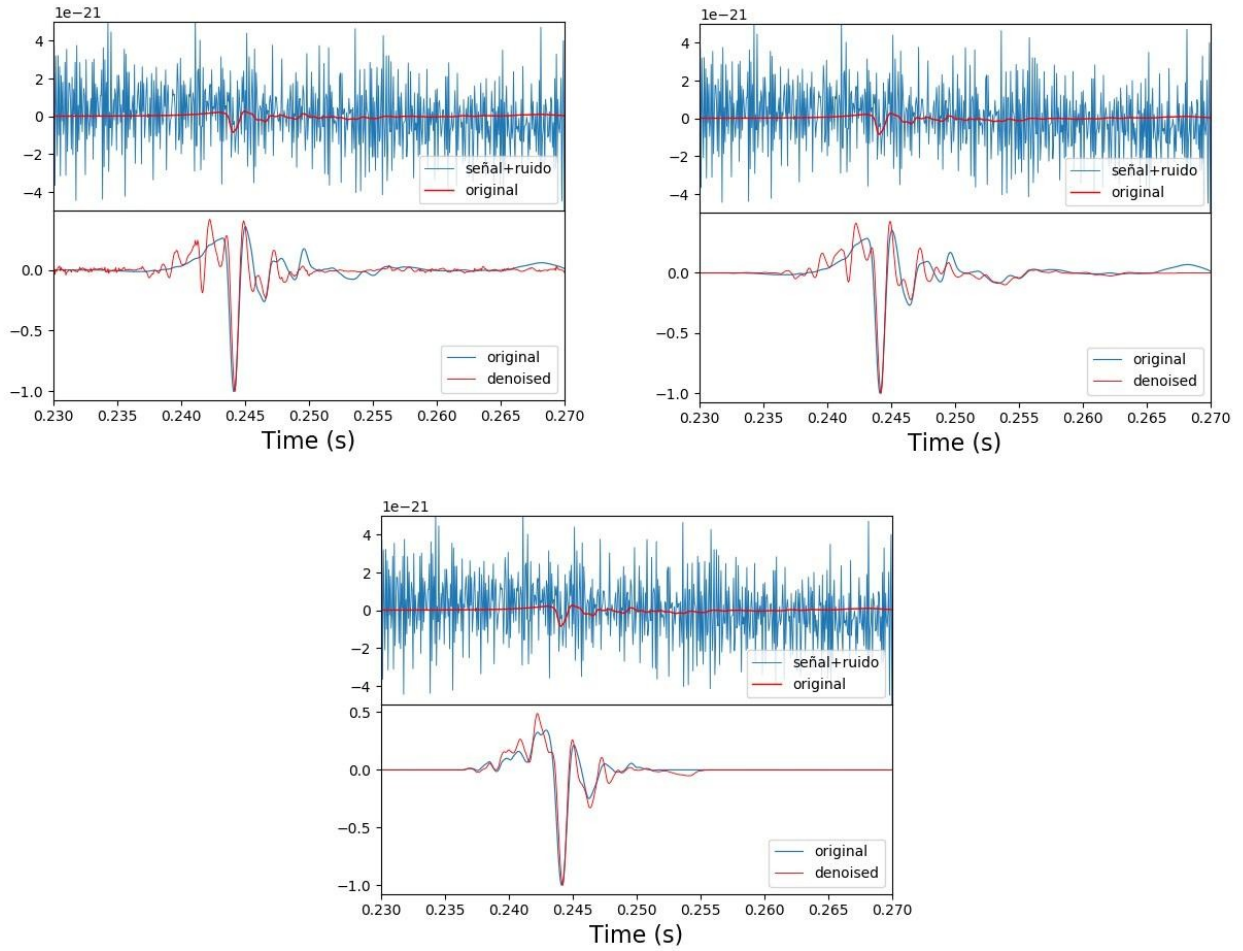


Fig 12. Reconstrucciones de la señal #3 con SNR 8 para los métodos M1 (arriba, izquierda), M2 (arriba, derecha) y M3 (abajo).

Por otro lado, la señal #7 se reconstruye de forma deficiente para los 3 métodos (ver figura 13). M1 genera estructuras numéricas a causa del ruido y además se pueden apreciar en las zonas donde la señal debería ser plana, como se puede ver en el tramo 0.23 s - 0.235 s. M3 consigue solucionar este problema muy bien, sin embargo, el tramo de oscilaciones situado aproximadamente entre 0.246 s y 0.26 s se reconstruye notablemente mal, por este motivo este método ha obtenido un valor del MSE tan elevado, el más alto de todos nuestros resultados: 0.0235.

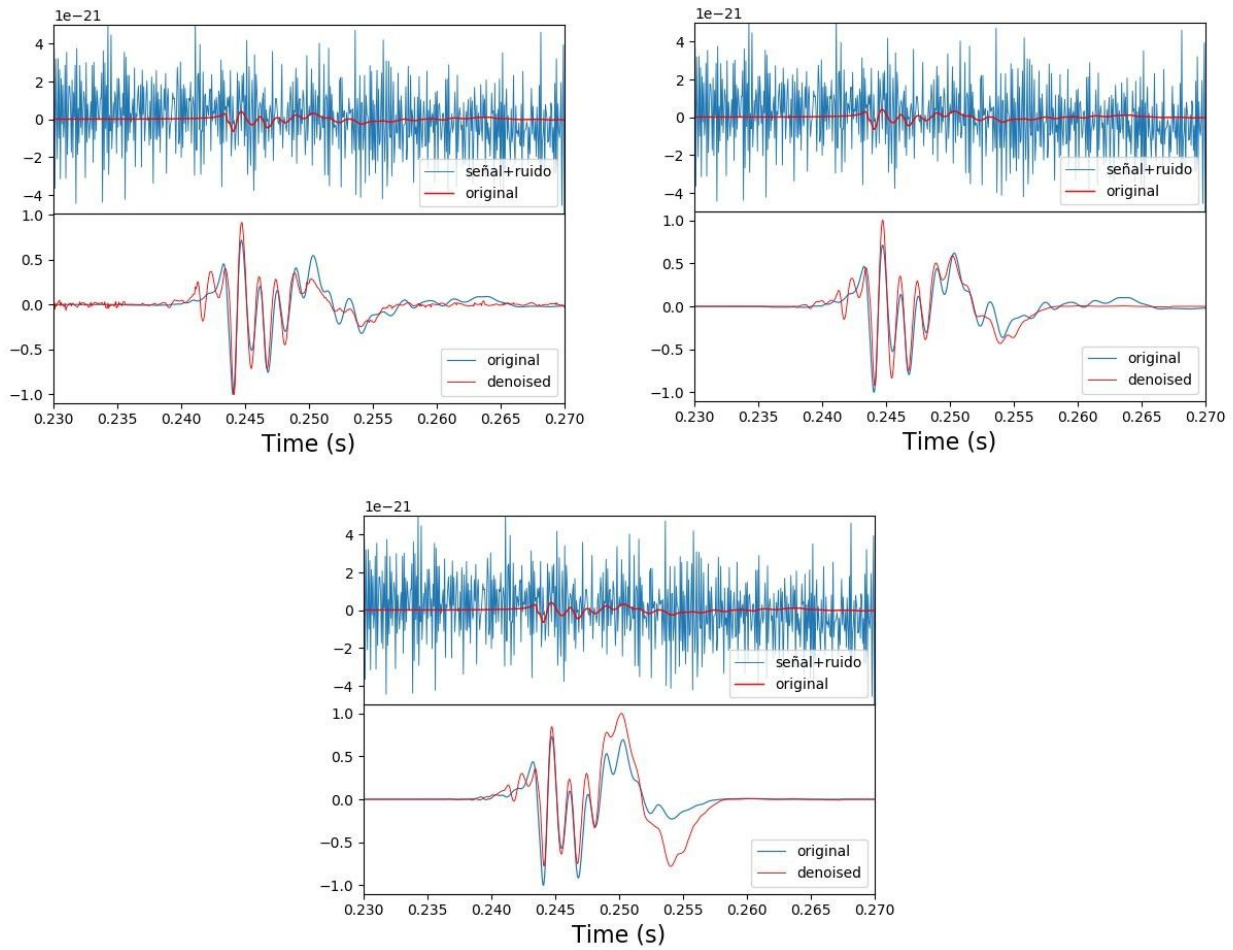


Fig 13. Reconstrucciones de la señal #7 con SNR 8 para los métodos M1 (arriba, izquierda), M2 (arriba, derecha) y M3 (abajo).

## 8. Más allá de SNR 8

Con el objetivo de averiguar hasta dónde podríamos llevar las reconstrucciones de señales con los métodos estudiados, nos proponemos escalar las señales que mejor hemos reconstruido a valores del SNR menores a 8. Probaremos 3 reconstrucciones para las señales #1 y #3 (0.00041 y 0.00026 de MSE respectivamente a SNR 20) bajando a valores del SNR de 7, 6 y 5. Los resultados que hemos obtenido son los siguientes.

Señal	SNR 7				SNR 6				SNR 5			
	d (kpc)	$\alpha_{opt}$	$\lambda_{opt}$	MSE	d (kpc)	$\alpha_{opt}$	$\lambda_{opt}$	MSE	d (kpc)	$\alpha_{opt}$	$\lambda_{opt}$	MSE
#1	112.49	0.944	10.000	0.0056	131.24	0.922	10.000	0.0082	157.49	0.766	10.000	0.0136
#3	174.61	1.468	4.201	0.0041	203.72	0.848	10.000	0.0077	244.46	0.723	10.000	0.0145

Tabla 5. Resultados en función del MSE y para SNR7,6,5

Las imágenes de las reconstrucciones son las adjuntadas a continuación en las figuras 14 y 15.

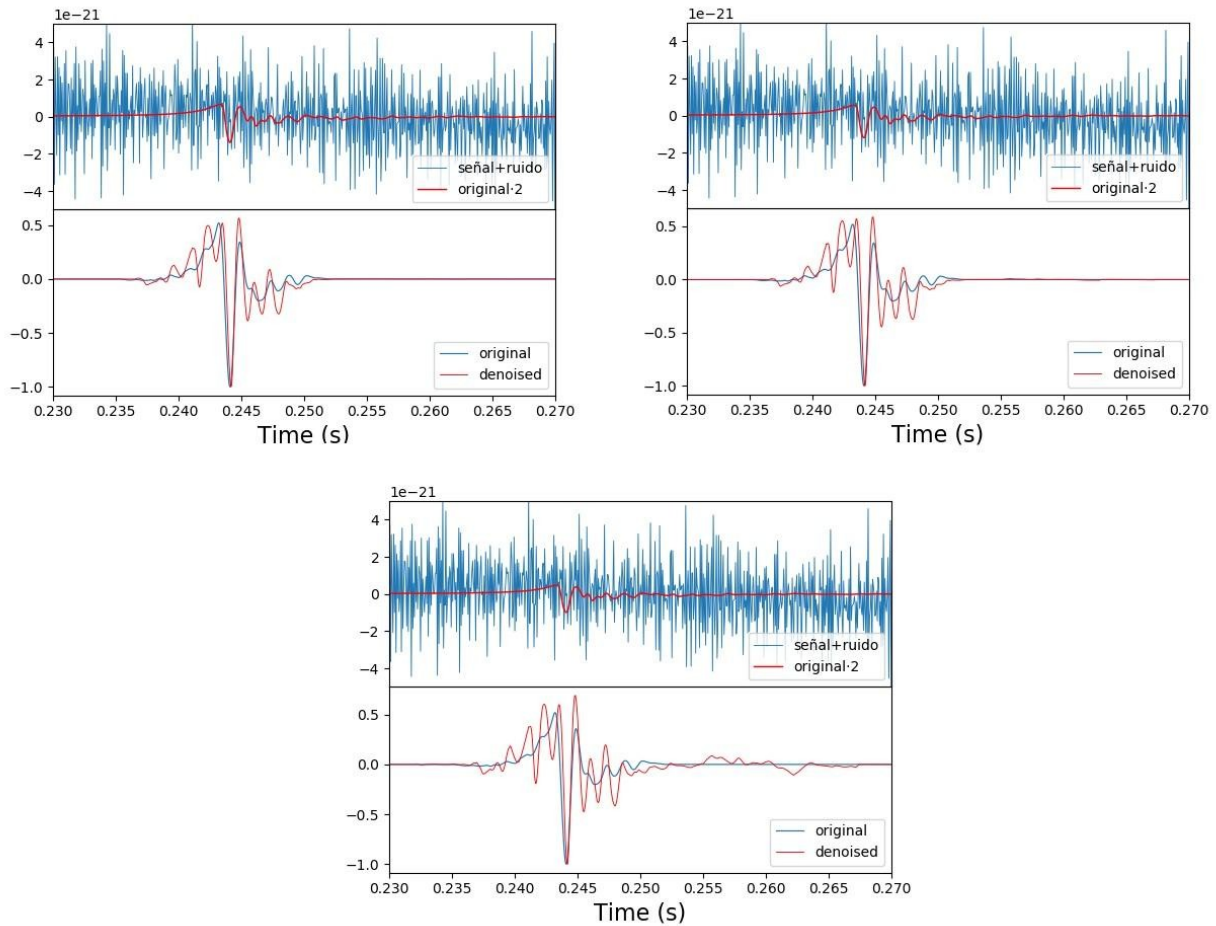


Fig 14. Reconstrucciones de la señal #1 a SNR 7 (arriba, izquierda), 6 (arriba, derecha) y 5 (abajo)

Como podemos observar, la reconstrucción para SNR 7 ya es mala: el MSE es de un orden mayor que para SNR 20 (0.0056). Únicamente se reconstruye bien el pico principal y los ceros de la señal a izquierda y derecha. Las oscilaciones antes y después del pico principal son demasiado exageradas, y reduciendo el SNR, evidentemente la reconstrucción empeora, hasta el punto de superar el valor 0.01 de MSE. Además, con SNR 5, el ruido es tan grande en comparación a la señal que ni siquiera se puede reconstruir correctamente la parte de ceros a la derecha de la señal, donde se aprecian pequeñas oscilaciones. Podemos decir que para la señal #1, bajar a valores del SNR menores que 8 supone que las reconstrucciones no son lo suficientemente fieles a las señales originales y por lo tanto no podríamos recuperarlas de forma satisfactoria.



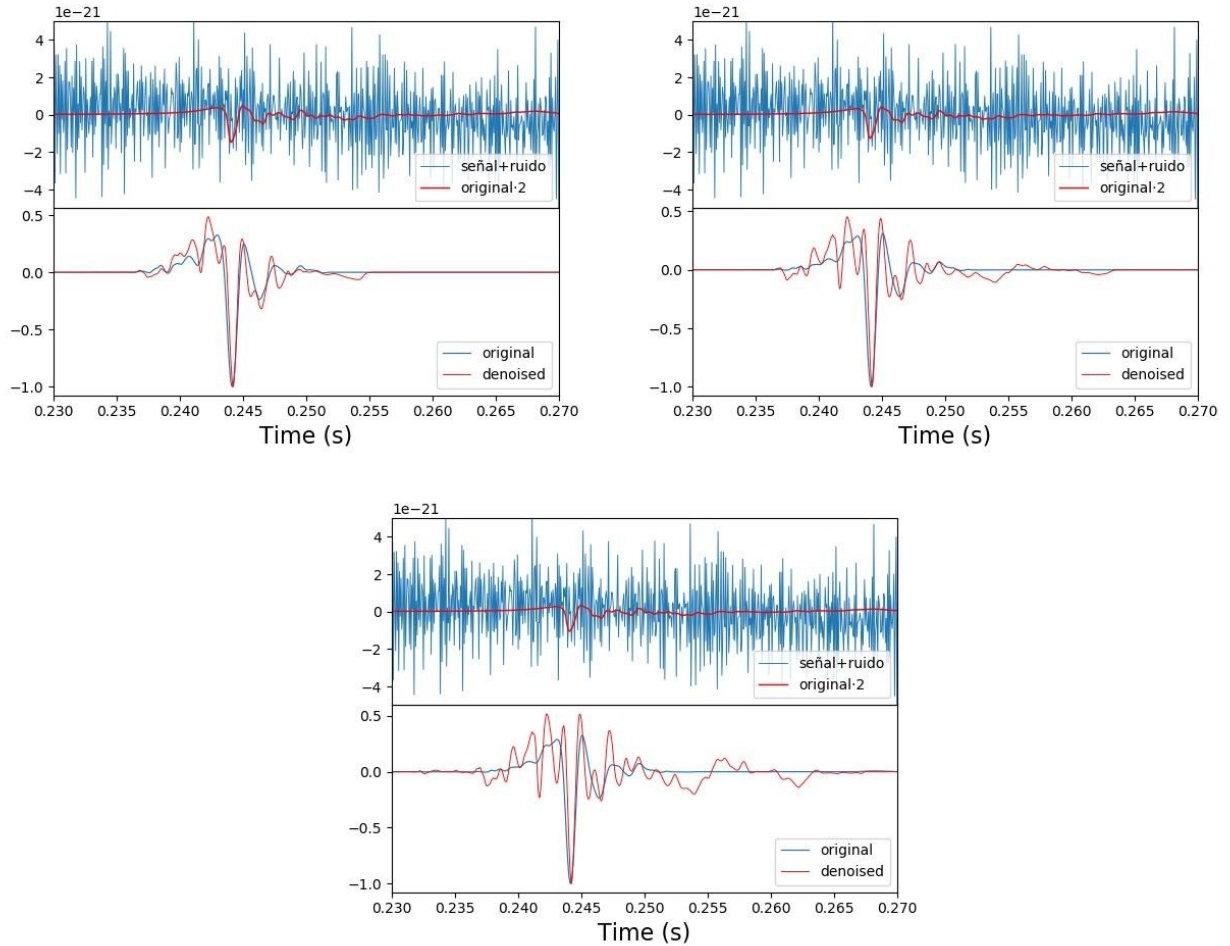


Fig 15. Reconstrucciones de la señal #2 a SNR 7 (arriba, izquierda), 6 (arriba, derecha) y 5 (abajo)

Las reconstrucciones para la señal #3 no son muy diferentes a los anteriores casos discutidos. En el caso de SNR 7 de nuevo vemos un incremento de hasta un orden superior en el MSE (0.0041) en comparación a SNR (0.00026). No obstante, la reconstrucción no es del todo mala: el pico principal se recupera bien, y aunque las oscilaciones previas no se recuperan muy bien, las posteriores, así como los ceros de la señal, se reconstruyen de forma bastante satisfactoria. Cuando bajamos a SNR 6 y 5 los resultados son ya demasiado malos: el MSE para la mayor distancia (0.0145) es incluso mayor que la señal #1 (0.0136). El pico principal es lo único que se reconstruye bien, ni los ceros de la señal son recuperados adecuadamente, ni tampoco las oscilaciones anteriores o posteriores, que son mucho más exageradas que las de la señal original. Por tanto, la reconstrucción a 174 kpc se podría llegar a tomar como válida, pero definitivamente ni esta señal ni la anterior podrían ser reconstruidas a SNRs tan bajos como 5 o 6.

## 9. Conclusiones

Recapitulamos todo lo que hemos hecho en este trabajo. Después de una introducción acerca del campo de estudio en el que se ubica el presente trabajo, el catálogo de ondas gravitacionales que hemos utilizado y una breve explicación de los fundamentos matemáticos de los algoritmos utilizados, hemos explicado la sistemática de nuestro experimento: qué hemos hecho y cómo funciona. Hemos descrito cada uno de los tres métodos empleados por separado y sus características. Finalmente, hemos seleccionado los parámetros adecuados de los métodos para emplear un diccionario óptimo para nuestro estudio, llevando a cabo las reconstrucciones de las señales y explicando los resultados obtenidos. A continuación, vamos a repasar y puntualizar los resultados más importantes e interesantes de este trabajo y comentar posibles extensiones futuras.

En cuanto a los diccionarios, hemos comprobado en el proceso de validación de los métodos que el intercambio de un diccionario (500,256) por uno (300,128) es razonable, ya que el MSE medio obtenido es menor para el diccionario pequeño. A pesar de que el SSIM medio es mucho mayor para el diccionario grande, hemos basado todos nuestros resultados en el estimador MSE, por lo que la selección de los parámetros 300 átomos y longitud 128 diccionario es clara. En el caso de tener acceso a mayores recursos computacionales de los que he dispuesto en este estudio, se podría utilizar el diccionario (500,256) y realizar el trabajo completo como un problema de maximización del SSIM en lugar de un problema de minimización del MSE. Incluso se podría aumentar el número de átomos y jugar con nuevas combinaciones de longitudes 128 y 256, como (1000, 128) o (1000, 256), con tal de ver si hay una mejora importante utilizando diccionarios todavía más grandes.

Una vez terminado el proceso de validación, seleccionamos nuestro diccionario en base a los parámetros óptimos escogidos y empezamos a reconstruir las señales del bloque de test, que son las últimas 8 señales del catálogo. En este proceso debemos minimizar el MSE para obtener los parámetros óptimos para cada SNR diferente. Los resultados obtenidos por cada método han sido adjuntados en sus correspondientes tablas. De forma general, nuestros resultados indican que añadir el algoritmo ROF para ejecutar un *denoising* previo de las señales mejora las reconstrucciones (método M2) en lugar de aplicar únicamente diccionarios (método M1). Además, si la minimización se lleva a cabo de forma simultánea para ambos métodos, ROF y diccionarios, las reconstrucciones son todavía mejores (método M3).

En cuanto a la dependencia de la calidad de los resultados con la disminución progresiva del SNR, hemos comprobado que los resultados empeoran, como era lógico esperar. Sin embargo, la mayoría de señales se pueden reconstruir de forma relativamente aceptable. Exceptuando la señal número 7 (MSE de 0.23), el resto de señales reconstruidas por el procedimiento M3 a SNR 8 tienen valores del MSE por debajo de 0.006. Con objeto de averiguar hasta qué punto nuestro código es capaz de recuperar señales a mayores distancias, hemos escogido las 2 señales con mejores reconstrucciones a SNR grandes y las hemos escalado a SNR menores que los propuestos inicialmente, concretamente 7, 6 y 5. Los resultados de las nuevas reconstrucciones demuestran que bajar de SNR 8 implica reconstrucciones deficientes; la única que podríamos dar por válida sería la correspondiente a la señal #3 a SNR 7 y ya presenta un error elevado. Por tanto, nuestros algoritmos no son suficientes para llevar a cabo un buen *denoising* de señales muy lejanas, es decir, con SNRs muy pequeños. Podemos concluir, pues, que nuestro procedimiento es eficaz para recuperar señales en torno a SNR 20 y valores más bajos, algunas señales incluso se reconstruyen bien a SNR 10. No obstante, la reconstrucción de señales con valores del SNR menores a 8 es pobre a la hora de obtener la forma de onda, pero puede ser interesante para detectar la señal puesto que el ruido se reduce considerablemente. Por tanto, se podrían reconstruir incluso algunas de las señales observadas por los interferómetros LIGO y Virgo, que pese a ser correspondientes a fusiones de binarias de objetos compactos (estrellas de neutrones y agujeros negros) y no a explosiones de supernovas (que son intrínsecamente más débiles), presentan valores del SNR superiores a 10.

Terminamos esta memoria sugiriendo algunas posibles direcciones en las que continuar el estudio iniciado en este trabajo con objeto de obtener todavía mejores resultados. Por ejemplo, se podría subclasificar el catálogo según las señales que tengan morfologías más o menos similares e investigar si generar diccionarios por separado en base a cada morfología reduce el error para las señales test. En este procedimiento se separaría el catálogo reuniendo un conjunto de ondas cuya morfología fuera muy semejante (ver 7.1) con tal de ver si el entrenamiento de los diccionarios, al ser llevado a cabo con señales de una única morfología, mejora los resultados cuando los aplicamos en la eliminación del ruido. También, como hemos comentado anteriormente, se podría llevar a cabo el trabajo de forma alternativa maximizando el SSIM, o incluso optimizando simultáneamente los dos estimadores. Habría que decidir qué valores compensan más del SSIM o del MSE, pero se

podría obtener un conjunto de parámetros óptimos que quizá mejorara los resultados obtenidos en el presente trabajo.

## 9. Conclusions

We now review everything we have done in this work. After introducing the field of study to which this work belongs, the gravitational wave catalog we have used, and a brief explanation of the mathematical framework of the algorithms we have employed, we have explained our experiment, : what we have done and the way it works. Each of our three methods and their main characteristics have been described separately. We have selected the appropriate parameters in order to make use of an optimal dictionary for our study. Hereafter, we summarize the most important and interesting results of this work and we comment possible future lines of research.

Regarding dictionaries, we have checked throughout the validation process of the methods that exchanging a large dictionary (500,256) by a small one (300,128) is quite reasonable, since the mean value of the MSE obtained is lower for the small one. We note that the mean SSIM estimator is higher for the large dictionary. However, all of our results have been based in the MSE statistical estimator only, which justifies using the small dictionary with 300 atoms of length 128. If we could have access to more powerful computational resources than those used in this work, a larger dictionary could have been used, such as the (500,256) dictionary, which could have allowed to carry out the whole study maximizing the SSIM instead of minimizing the MSE. We could have even increased the number of atoms and try new combinations of lengths and atoms, like (1000,128) o (1000,256) in order to find out by how much our results improve.

Once the validation procedure is finished, we select our dictionary based on the optimal parameters obtained and we begin reconstructing our test signals, namely the last eight of the core-collapse supernova catalog of Dimmelman et al [3]. In this process we must minimize the MSE estimator in order to obtain the optimal parameters for each one of the SNRs considered. The corresponding results have been reported in a number of tables. Broadly speaking, our results show that using the ROF algorithm to perform an early denoising on the test signals improves the reconstructions (M2 method) compared to the outcome of simply performing one single denoising with the dictionary algorithm (M1 method). Moreover, if the minimization of the MSE is carried out simultaneously for both algorithms, ROF and dictionaries, our results show even better reconstructions (M3 method).

Regarding the quality of our results depending on the progressive decreasing of the SNR, we have shown that the lower the SNR the worse the results, which is what we expected. However, most signals can still be reconstructed relatively well. Except signal #7 (MSE of 0.23), the rest of the reconstructed signals using the M3 method with SNR 8 yield values of the MSE lower than 0.006. With the goal of studying how well our algorithms perform when applied to further distance signals (lower SNR) we have selected the two signals with the best reconstructed waveforms with SNR 20 (our highest value) and we have scaled them to even lower SNR values than initially considered, namely 7, 6 and 5. The results of the new reconstructed signals show that lowering the SNR below 8 implies poor reconstructions, with signal #3 with SNR 7 being the only acceptable result, despite having already a high MSE value. Therefore, our algorithms are not good enough to perform a proper denoising with faraway signals, i.e., low values of the SNR. We can thus conclude that the algorithms developed and tested in this work are efficient when it comes to denoising signals with SNR around 20 and a bit lower, with some signals even being well reconstructed with SNR 10. However, signals with lower SNR than 8 are poorly reconstructed regarding the waveform, though the algorithms could be used in order to detect the signal, since noise is considerably reduced. In addition, we note that our algorithms could reconstruct some of the signals observed by LIGO and Virgo since, despite they correspond to mergers of binary compact objects (neutron stars and black

holes) and not to core-collapse supernova explosions (which are intrinsically weaker), they present SNR values higher than 10.

We conclude this report by suggesting some possible lines of work to continue this study in order to improve even further our results. For instance, the waveform catalog could be divided based on the signals that share a similar morphology which would allow to investigate if generating dictionaries separately for each morphology reduces the error for the test signals. In this procedure the catalog would be split into different morphologies, gathering 2 or 3 subtypes of waveforms (as explained in section 7.1) so that we could study if the training of dictionaries with separate catalogs improves our results when denoising the test signals. In addition, as mentioned before, this work could be carried out as a SSIM maximizing problem, or even as a simultaneous optimization of the two estimators. In such case, we would have to decide which values are actually worth obtaining for both SSIM and MSE. Ideally we could come up with a set of optimal parameters that might improve the results obtained in the present work.

## 10. Referencias

- [1] Bambi C. (2018) Gravitational Waves. In: Introduction to General Relativity. Undergraduate Lecture Notes in Physics. Springer, Singapore. [https://doi.org/10.1007/978-981-13-1090-4\\_12](https://doi.org/10.1007/978-981-13-1090-4_12)
- [2] J. Antonio Font, apuntes de Relatividad y cosmología, tema 4, curso 2019-2020.
- [3] H. Dimmelmeier, C. D. Ott, A. Marek, and H.-T. Janka, Phys. Rev. D 78, 064056 (2008).
- [4] Torres-Forné Alejandro, Cuoco Elena, Font José A., Marquina Antonio. (2020). Application of dictionary learning to denoise LIGO's blip noise transients.
- [5] Torres-Forné Alejandro, Marquina Antonio, Font Jose A., Ibáñez José. (2016). Denoising of gravitational wave signals via dictionary learning algorithms. Physical Review D. 94. 10.1103/PhysRevD.94.124040.
- [6] Image denoising using dictionary learning, [https://scikit-learn.org/stable/auto\\_examples/decomposition/plot\\_image\\_denoising.html#sphx-glr-auto-examples-decomposition-plot-image-denoising-py](https://scikit-learn.org/stable/auto_examples/decomposition/plot_image_denoising.html#sphx-glr-auto-examples-decomposition-plot-image-denoising-py).
- [7] Torres-Forné Alejandro, Cuoco Elena, Marquina Antonio, Font Jose A., Ibáñez José. (2018). Total-variation methods for gravitational-wave denoising: Performance tests on Advanced LIGO data. Physical Review D. 98. 10.1103/PhysRevD.98.084013.
- [8] Torres-Forné Alejandro, Marquina Antonio, Font Jose A., Ibáñez José. (2014). Total-variation-based methods for gravitational wave denoising. Physical Review D. 90. 084029. 10.1103/PhysRevD.90.084029.
- [9] L. I. Rudin, S. Osher, and E. Fatemi, Physica (Amsterdam), 60D 259 (1992).
- [10] Sparse coding with a precomputed dictionary, [https://scikit-learn.org/stable/auto\\_examples/decomposition/plot\\_sparse\\_coding.html#sphx-glr-auto-examples-decomposition-plot-sparse-coding-py](https://scikit-learn.org/stable/auto_examples/decomposition/plot_sparse_coding.html#sphx-glr-auto-examples-decomposition-plot-sparse-coding-py)
- [11] SparseCoder function documentation <http://scikit-learn.org/stable/modules/generated/sklearn.decomposition.SparseCoder.html>
- [12] Wei Dai Boris Mailhé, Wenwu Wang, Dictionary Learning for Sparse Representations Algorithms and Applications, May 2013, [http://www.ee.imperial.ac.uk/wei.dai/PaperDiscussion/Material2013/ICASSP2013\\_tutorial\\_overall.pdf](http://www.ee.imperial.ac.uk/wei.dai/PaperDiscussion/Material2013/ICASSP2013_tutorial_overall.pdf)
- [13] Librería o paquete de Python de aprendizaje automático *scikit-learn*, <https://scikit-learn.org/stable/>
- [14] MiniBatchDictionaryLearning function documentation <https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.MinibatchDictionaryLearning.html>

[15] Mairal Julien, Bach Francis, Ponce J, Sapiro Guillermo. (2009). Online dictionary learning for sparse coding. Proceedings of the 26th International Conference On Machine Learning, ICML 2009. 382. 87. 10.1145/1553374.1553463.