

INFLUENCIA EN LA DETECCIÓN DE PATRONES  
DE LA SOLUCIÓN DEL SISTEMA NO LINEAL EN  
UNA TRANSFORMADA SHAPELET DISCRETA II

INFLUENCE ON PATTERN DETECTION FOR THE  
SOLUTION OF THE NONLINEAR SYSTEM ON A  
DISCRETE SHAPELET TRANSFORM II

DAMIAN VALDÉS-SANTIAGO<sup>1</sup>      ÁNGELA M. LEÓN-MECÍAS<sup>2</sup>  
MARTA L. BAGUER DÍAZ-ROMAÑACH<sup>3</sup>  
MANUEL GONZÁLEZ-HIDALGO<sup>4</sup>      ANTONI JAUME-I-CAPÓ<sup>5</sup>

*Received: 10/Mar/2023; Accepted: 14/Nov/2023*

---

*Revista de Matemática: Teoría y Aplicaciones* is licensed under a Creative Commons  
Reconocimiento-NoComercial-Compartirigual 4.0 International License.  
Creado a partir de la obra en <http://www.revistas.ucr.ac.cr/index.php/matematica>



---

<sup>1</sup> Universidad de La Habana, Facultad de Matemática y Computación, La Habana, Cuba. E-Mail: [dvs89cs@matcom.uh.cu](mailto:dvs89cs@matcom.uh.cu)

<sup>2</sup> Universidad de La Habana, Facultad de Matemática y Computación, La Habana, Cuba. E-Mail: [angela@matcom.uh.cu](mailto:angela@matcom.uh.cu)

<sup>3</sup> Universidad de La Habana, Facultad de Matemática y Computación, La Habana, Cuba. E-Mail: [mbaguer@matcom.uh.cu](mailto:mbaguer@matcom.uh.cu)

<sup>4</sup> Universidad de las Islas Baleares, Departamento de Ciencias Matemáticas e Informática, Palma, España. E-Mail: [manuel.gonzalez@uib.es](mailto:manuel.gonzalez@uib.es)

<sup>5</sup> Universidad de las Islas Baleares, Departamento de Ciencias Matemáticas e Informática, Palma, España. E-Mail: [antoni.jaume@uib.es](mailto:antoni.jaume@uib.es)

### Resumen

El uso de *wavelets* adaptadas para el reconocimiento de patrones es muy atractivo por la multiescalaridad de la transformada *wavelet*. Sin embargo, el buen desempeño de estos algoritmos en la detección de patrones depende fuertemente de la construcción de los filtros adaptados al patrón de interés. La Transformada *Shapelet* Discreta II [9] (DST-II) es un algoritmo inspirado en la transformada *wavelet*, que permite el diseño de filtros a la medida para la detección de patrones en señales unidimensionales. La construcción de estos filtros requiere la solución de un sistema de ecuaciones no lineales, que según [9] se puede efectuar mediante cualquier método iterativo. Esta investigación presenta un novedoso y exhaustivo estudio numérico que demuestra el impacto de la elección del método numérico adecuado para la solución del sistema no lineal en la DST-II. La eficacia de los filtros estimados repercute en el desempeño de esta transformada en la detección de patrones. Los mejores resultados se obtienen al combinar el método de Newton con preiteración mediante el algoritmo de continuación. La convergencia alcanzada para el 55,37% de los patrones sugiere que la DST-II podría ser adecuada para patrones con formas específicas, de utilidad en aplicaciones sobre señales biomédicas.

**Palabras clave:** diseño de filtros wavelet; wavelet adaptada; transformada shapelet discreta; transformada wavelet discreta.

### Abstract

The use of adapted wavelets for pattern recognition is very attractive because of the multiscalarity of the wavelet transform. However, the good performance of these algorithms in pattern detection strongly depends on the construction of the filters adapted to the pattern of interest. The Discrete Shapelet Transform II [9] (DST-II) is an algorithm inspired by the wavelet transform, which allows the design of tailored filters for pattern detection in one-dimensional signals. The construction of these filters requires the solution of a system of nonlinear equations, which according to [9] can be performed by any iterative method. This research presents a novel and comprehensive numerical study that demonstrates the impact of the choice of the appropriate numerical method for the solution of the nonlinear system in DST-II. The efficiency of the estimated filters has an impact on the performance of this transform in pattern detection. The best results are obtained by combining Newton's method with preiteration using the continuation algorithm. The convergence achieved for 55,37% of the patterns suggests that DST-II could be suitable for patterns with specific shapes, useful in biomedical signal applications.

**Keywords:** wavelet filter design; matched wavelet; discrete shapelet transform; discrete wavelet transform.

**Mathematics Subject Classification:** Primario: 65T; Secundario: 65T60, 42C40, 65F99, 65H10

## 1 Introducción

La literatura científica recoge varios métodos para construir *wavelets* adaptadas a patrones específicos. Aldroubi et al. [1] obtuvieron nuevas *wavelets* aplicando convoluciones a otras ya definidas, minimizando la diferencia con el patrón. Mesa [17] utilizó un esquema *lifting* [18] para construir *wavelets* biortogonales adaptadas con varios momentos nulos [16]. Misiti et al. [18] utilizaron la transformada *wavelet* continua para construir *wavelets* adaptadas en el sentido de los mínimos cuadrados. Jawali et al. [12] usaron el aprendizaje automático para diseñar *wavelets* ortonormales de soporte compacto con determinados momentos nulos.

Guido y colaboradores [11, 9, 10] construyeron *wavelets* adaptadas con la Transformada *Shapelet* Discreta (DST), que determina el soporte en el tiempo no solo de las frecuencias, sino también de la forma de un patrón. La DST-I [11] incluyó una restricción basada en dimensión fractal y resolvió el sistema de ecuaciones no lineales (SENL) formado por esta restricción y otras de energía unitaria, momentos nulos y ortogonalidad. Su costo computacional fue alto. Por eso, la DST-II [9] sustituyó la restricción de fractalidad por dos restricciones de correlación con el patrón. En la tercera versión, DST-III [10], se propuso la obtención de *wavelets* casi simétricas cambiando la restricción de momentos nulos por la de simetría [16].

Según [9, p. 10 y p. 11] y [10, p. 3], este SENL puede resolverse “mediante cualquier procedimiento numérico iterativo” para obtener los coeficientes del filtro de paso alto de la *shapelet*. Las investigaciones citadas muestran dos ejemplos numéricos de obtención de *shapelets* para patrones artificiales y experimentos de detección de siete patrones del sistema visual de una mosca. En estos experimentos no se especifica cuál algoritmo numérico fue utilizado para encontrar la solución del SENL, ni cuál fue la aproximación inicial usada para la iteración; aspecto determinante en la convergencia [4]. Tampoco se muestran indicadores como el número de evaluaciones del sistema, la norma del residual, la similitud patrón-*shapelet* y el tiempo de ejecución. Luego de la revisión de la literatura científica sobre la DST-I, II y III, no se encontraron referencias a experimentos numéricos sobre el aspecto señalado anteriormente.

En un estudio exploratorio previo [24], se implementó la DST-II [9] y se evaluaron tres algoritmos para resolver el SENL subyacente, mostrándose una gran dependencia entre la elección de dichos algoritmos y la precisión en la detección del patrón. De ahí, la novedad de desarrollar un estudio más profundo sobre esta dependencia, dado que no se reporta en la literatura revisada una investigación similar. Por ello, en esta investigación se presenta un novedoso y amplio estudio numérico que demuestra la influencia de la elección del método iterativo de solución del sistema para construir la *shapelet*, así como de la elección de la aproximación inicial. Se compararon 12 algoritmos para resolver el sistema desde del vector nulo. Luego, se establecieron combinaciones de métodos para mejorar la solución, com-

probando la efectividad de la pre-iteración. Se evaluó la capacidad de los filtros para detectar 103 patrones en 17 señales.

El trabajo está estructurado como sigue: en la sección 2 se introducen conceptos y elementos básicos de la DST-II, y se presenta el SENL objeto de estudio. La sección 3 se dedica a los aspectos teóricos de algoritmos de solución de un SENL utilizados en los experimentos. En la sección 4 se presentan los cuatro experimentos realizados. Una amplia discusión de los resultados se expone en la sección 5. Finalmente, se presentan conclusiones.

## 2 Transformada Shapelet Discreta de segunda generación (DST-II)

Con un enfoque cercano a la Transformada *Wavelet* Discreta (TWD), Rodrigo Capobianco Guido y colaboradores [11, 9, 10] presentan tres versiones de la llamada Transformada *Shapelet* Discreta (DST, en inglés); una transformada creada no solo para determinar el soporte en el tiempo de las frecuencias de una señal, sino también para detectar un patrón en esta. La DST está construida para detectar dicho patrón independientemente de su amplitud, y presenta la misma complejidad temporal de cualquier TWD al descomponer una señal para un mismo tamaño de soporte. La DST es capaz de ajustarse a un patrón predefinido, pero no a un conjunto de patrones como lo haría un algoritmo de aprendizaje automático.

La DST-I [11] utiliza el enfoque original de Daubechies para construir *wavelets*, pero desde el dominio temporal, imponiendo restricciones a los coeficientes del filtro de paso alto como la ortogonalidad, energía unitaria y un cierto número de momentos nulos. Para lograr que la *wavelet* se ajuste al patrón deseado, se agrega una restricción basada en dimensión fractal del patrón. Luego, se resuelve el sistema de ecuaciones no lineales (SENL) formado por estas restricciones. A pesar del éxito alcanzado por la *wavelet* adaptada siguiendo este algoritmo en la detección de patrones del complejo sistema visual de una mosca, su costo computacional fue relativamente alto [11].

La DST-II [9] surge como una modificación a la DST-I, que disminuye su costo computacional, y favorece la solución del SENL y permite una interpretación simplificada de la señal transformada [9, p. 9]. Esta sustituye la restricción de fractalidad para detectar el patrón por dos restricciones de detección basadas en la correlación entre el patrón y el filtro *wavelet*, que fueron definidas de manera que los productos escalares involucrados en el cálculo de la DST-II, según el algoritmo de Mallat, capturen el patrón buscado. La inclusión de estas dos restricciones permite que el SENL sea cuadrado.

En la tercera versión, DST-III [10], se propone la obtención de *wavelets* casi simétricas utilizando un enfoque muy similar a la DST-II, donde se cambia la restricción de momentos nulos por la usada por Daubechies para construir sus Symlets [5].

En esta investigación se utilizará la DST-II [9]. Esta transformada tiene las siguientes restricciones para los coeficientes del filtro de paso alto  $q$ : i) el tamaño de soporte del filtro es  $N \geq 6$  y necesariamente par para tener reconstrucción perfecta [23]; y ii) el patrón  $m[\cdot]$  debe tener un tamaño impar igual a  $N + 1$ .

El SENL  $\mathbf{F}(q_0, q_1, \dots, q_{N-1}) = \mathbf{0}$  de  $N$  ecuaciones y  $N$  incógnitas es el siguiente:

- Energía unitaria:

$$\left( \sum_{k=0}^{N-1} q_k^2 \right) - 1 = 0;$$

- los  $\frac{N}{2} - 2$  momentos nulos para la *wavelet*:

$$\sum_{k=0}^{N-1} q_k \cdot k^b = 0,$$

donde  $b = 0, 1, \dots, \frac{N}{2} - 3$ ;

- las  $\frac{N}{2} - 1$  condiciones de ortogonalidad:

$$\sum_{k=0}^{N-1} q_k \cdot q_{k+2l} = \delta_{0,l},$$

donde  $\delta$  es la función delta de Dirac y  $l \in \mathbb{Z}$ ; y

- dos condiciones para detectar el patrón:

$$\sum_{k=0}^{N-1} q_k \cdot m_k = 0, \quad \sum_{k=0}^{N-1} q_k \cdot m_{k+1} = 0. \quad (2.1)$$

Para obtener el banco de filtros de reconstrucción perfecta (PRFB, en inglés) [23] de una DST-II se computan:

- los  $p[\cdot]$  y  $q[\cdot]$ , con  $p_k = (-1)^k \cdot q_{N-k-1}$  ( $0 \leq k \leq N-1$ ), que forman los filtros de respuesta finita al impulso (FIR, en inglés) con un tamaño de soporte par,  $N$ ; y
- los  $\bar{p}[\cdot]$  y  $\bar{q}[\cdot]$ , donde  $\bar{p}_k = p_{N-k-1}$  y  $\bar{q}_k = (-1)^{k+1} \cdot p_k$ , que caracterizan el banco de filtros de síntesis.

En investigaciones futuras se utilizará la DST-II para descomponer una señal, modificar sus coeficientes y reconstruirla. Mientras más modificaciones se realicen a los coeficientes, más se parecerá la señal reconstruida a la forma de las funciones escala y *wavelet* de la DST-II [8]. Es por ello, que se computan las funciones *shapelet* mayor

$$\Gamma(x) = \sum_k p_k \cdot \Gamma(2N - k)$$

y *shapelet* menor

$$\Theta(x) = \sum_k q_k \cdot \Gamma(2N - k),$$

similares a las funciones de escala y *wavelet* de la TWD [23].

### 3 Métodos numéricos para hallar el filtro de paso alto de una DST-II

Para obtener los coeficientes del filtro de paso alto  $q$  de la *shapelet* debe resolverse el SENL descrito. A continuación se presentan elementos teóricos y consideraciones específicas de un conjunto de métodos utilizados para su solución numérica.

#### 3.1 Método de Newton

El método de Newton es un proceso de iteración de punto fijo basada en la matriz jacobiana. En lugar de calcular la matriz jacobiana inversa, se resuelve un sistema de ecuaciones lineales (consultar [4]). Puede que la matriz del sistema sea singular, por lo que se utilizó la pseudoinversa del jacobiano<sup>1</sup>.

La matriz jacobiana de un sistema lineal de orden  $n$  requiere que las  $n^2$  derivadas parciales sean determinadas y evaluadas. La evaluación exacta de las derivadas parciales es numéricamente inconveniente, por lo que se emplean sistemas simbólicos del paquete SymPy de Python. Resolver un sistema lineal requiere  $O(n^2)$  operaciones.

Se programó una implementación propia del método de Newton en Python, que usó la aproximación simbólica del jacobiano y la pseudoinversa (denominada `newton`).

#### 3.2 Algoritmo de Levenberg-Marquardt modificado

La opción `lm` del método `scipy.optimize.root` es un *wrapper* de la rutina LM-DIF1 de MINPACK-1 [19]. Esta minimiza la suma de los cuadrados de  $m$  funciones no lineales en  $n$  variables mediante una modificación del algoritmo de Levenberg-Marquardt. El jacobiano se aproxima por diferencias hacia adelante.

<sup>1</sup>La pseudoinversa de Moore–Penrose  $A^+$  invierte  $A$  cuando esta es invertible y tiene el mismo rango  $r$ . Si  $A = U\Sigma V^T$  es la descomposición de valores singulares de  $A$ , entonces  $A^+ = U\Sigma^+V^T$ , donde  $U$  y  $V$  son matrices ortogonales,  $\Sigma$  es una matriz diagonal que consiste en los llamados valores singulares de  $A$  y luego  $\Sigma^+$  es la matriz diagonal que consiste en los recíprocos de los valores singulares de  $A$  [22].

El número de operaciones aritméticas que necesita LMDIF1 es  $\mathcal{O}(n^3)$  para procesar cada evaluación de función y  $\mathcal{O}(mn^2)$  para procesar cada aproximación al jacobiano. LMDIF1 requiere  $mn + 2m + 6n$  posiciones de almacenamiento de doble precisión y  $n$  posiciones de almacenamiento de enteros.

### 3.3 Métodos de Broyden

Los métodos de Broyden [21] tienen como base un algoritmo cuasi-Newton. Estos se diferencian en la forma de estimar la matriz jacobiana [4].

La matriz  $B_k$  es una hessiana construida a partir de iteraciones anteriores. Los métodos del paquete SciPy usan la actualización de Broyden-Fletcher-Goldfarb-Shanno (BFGS).

La matriz inicial  $B_0$  se aproxima con la matriz jacobiana (método “malo” de Broyden, `broyden2`). La variante que actualiza directamente la inversa de la matriz jacobiana con la fórmula de Sherman-Morrison se llama método “bueno” de Broyden, `broyden1`. Existen tres variantes más: i) se realiza una aproximación escalar del jacobiano (`linear mixing`), ii) aproximación del jacobiano mediante matriz diagonal que se estima en cada iteración (`exciting mixing`), y iii) aproximación diagonal del jacobiano (`diagbroyden`).

Estos métodos tienen convergencia superlineal, requieren  $n$  evaluaciones de función por iteración y  $\mathcal{O}(n^2)$  operaciones.

### 3.4 Método de Powell

El método de Powell [20, 21], se utiliza para actualizar la solución actual, en cada iteración, si el paso del algoritmo de Newton está dentro de la región de confianza. Si no es así, el algoritmo busca el mínimo de la función objetivo a lo largo de la dirección de descenso más pronunciada, conocida como punto de Cauchy. Si el punto de Cauchy está fuera de la región de confianza, se trunca hasta el límite de esta y se toma como nueva solución. Si el punto de Cauchy está dentro de la región de confianza, la nueva solución se toma en la intersección entre el límite de la región de confianza y la línea que une el punto de Cauchy y el paso del método de Newton [20, 21].

Este método está implementado en MINPACK-1 [19] como HYBRD. Su número de operaciones es  $\mathcal{O}(m^2)$  por cada llamada de función y requiere  $(3m^2 + 17m)/2$  para la memoria. Las demostraciones pueden consultarse en [20].

### 3.5 Método de Anderson

Sean los vectores de entrada y salida, respectivamente, de las  $M$  iteraciones que se van a considerar:

$$\bar{x}^{(k)} := x^{(k)} + \sum_{j=1}^M \vartheta_j^{(k)} (x^{(k-j)} - x^{(k)}),$$

donde  $0 \leq M \leq k - 1$ .

Los coeficientes  $\vartheta_j^{(k)}$  se buscan de forma que dicha combinación lineal minimice la norma del vector residual general. Para ello, se resuelve un sistema de ecuaciones lineales cuya solución son los coeficientes  $\vartheta_j^{(k)}$  [2].

Una vez encontrada la combinación lineal óptima, se fija el vector de entrada para la siguiente iteración:  $x^{(k+1)} = \bar{x}^{(k)} + \beta^{(k)}\bar{R}^{(k)}$ , donde  $\beta^{(k)}$  es el parámetro de mezcla.

El método de Anderson necesita el almacenamiento de los vectores  $(2M + 2)N$ -componentes  $x^{(k)}$  y  $R^{(k)}$  de las iteraciones actuales y de las  $M$  anteriores, y, además, la solución de un sistema lineal de orden  $M$ .

En los experimentos se utilizó la implementación de este método brindada por la opción `anderson` de `scipy.optimize.root`.

### 3.6 Método de Newton–Krylov

Se computa la inversa del jacobiano con el método iterativo de Krylov [13], donde los vectores producto del jacobiano son aproximados mediante diferenciación numérica. El módulo `scipy.sparse.linalg` brinda una selección de solucionadores de Krylov que usa LGMRES.

### 3.7 Método del residuo espectral sin derivadas

La Cruz et al. [14] introducen el algoritmo DF-SANE (*Derivative-free SANE*), una técnica de búsqueda lineal no monótona que utiliza las mismas direcciones de búsqueda y longitudes de paso iniciales que el algoritmo SANE, pero sin usar derivadas direccionales. En [14], se analiza la convergencia global del método y se muestran experimentos numéricos donde DF-SANE es frecuentemente mejor que el método de Newton–Krylov en problemas de gran escala.

### 3.8 Método de continuación

Un método de continuación [4] intenta determinar  $x(\lambda = 1)$  resolviendo la sucesión de problemas correspondientes a  $\lambda_0 = 0 < \lambda_1 < \lambda_2 < \dots < \lambda_n = 1$ . La aproximación inicial a la solución de  $\lambda_i \mathbf{F}(x) + (1 - \lambda_i)(\mathbf{F}(x) - \mathbf{F}(x_0)) = 0$ , sería la solución,  $x(\lambda = \lambda_i)$ , del problema  $\lambda_{i-1} \mathbf{F}(x) + (1 - \lambda_{i-1})(\mathbf{F}(x) - \mathbf{F}(x_0)) = 0$ .

Para resolverlo, se construye una función de homotopía  $H(x, \lambda)$  con un parámetro de continuación  $\lambda$  en  $F$ . En general, una ecuación de homotopía lineal puede escribirse como una combinación lineal del sistema inicial y del sistema objetivo; es decir,  $H(x, \lambda) = (1 - \lambda)p(x) + \lambda \mathbf{F}(x) = 0$ , con  $\lambda \in [0, 1]$ ,  $p(x)$  es el sistema inicial y  $\mathbf{F}(x)$  el objetivo.

Hay varias formas de definir el sistema inicial, una de ellas es la homotopía de Newton:  $H(x, \lambda) = \mathbf{F}(x) - (1 - \lambda)\mathbf{F}(x^{(0)}) = 0$ , con  $x^{(0)} \in \mathbb{R}^n$  es el punto inicial de  $x$ .

La idea del método de homotopía es encontrar  $x^*$  deformando el sistema inicial –cuya solución es  $x_0$ – al sistema objetivo. Así, para encontrar la solución  $x^*$  del



sistema objetivo, primero se halla una solución inicial  $x^{(0)}$  del sistema estableciendo  $\lambda_0 = 0$ . Entonces,  $\lambda_0$  se traslada a  $\lambda_1$  tanto como  $\Delta\lambda$  se acerque a 1. Luego, surge una nueva homotopía  $H(x, \lambda_1)$  con una solución  $x_1$ . Para encontrar  $x_1$ , se realizan dos pasos: predicción y corrección.

El método de continuación puede no converger si el jacobiano es singular, es por ello que numéricamente se utiliza la pseudoinversa del jacobiano.

En los experimentos se utilizó una implementación propia de este método (denominada *continuation*).

## 4 Diseño de la experimentación

Para mostrar el impacto del método numérico para resolver el SENL en la construcción de la *shapelet*, se realizó una amplia experimentación que transitó por las siguientes etapas:

- a) Selección y preparación de los datos: Se seleccionaron 103 patrones diferentes; patrón en [9], patrón y *spike* en [10], *spike* medio estándar, *wavelet* de Battle-Lemarié con soporte 12, *wavelet* de Beylkin con soporte 18 y *wavelet* de Vaidyanathan con soporte 24 [9], Daubechies 2-38, Symlets 2-18 y Coiflet 2-12 tomadas de PyWavelets [15], y 28 complejos K extraídos de electroencefalogramas en DREAMS [6].

Los complejos K se decimaron por un factor de 12, conservando su forma con un filtro antialiasing de orden 8 (`scipy.signal.decimate`).

Para los patrones de tamaño impar, se repitió la primera muestra al final para alcanzar un número par de muestras del patrón. Así, la dimensión de los patrones osciló entre 8 y 102 muestras. Las amplitudes de los patrones fueron normalizadas para disminuir errores numéricos y mejorar el condicionamiento de las matrices.

- b) Obtención del banco de filtros de la DST-II: El SENL para obtener  $q$  se codificó simbólicamente en SymPy y se resolvió con 12 métodos numéricos (ver sección 3). Se usaron las rutinas de MINPACK-1 [19, 21] implementadas en el paquete `scipy.optimize.root` de Python, así como implementaciones propias.
- c) Se realizaron experimentos con doble precisión:
  - *Experimento I*: Comparar las *shapelets* obtenidas por los métodos numéricos con las de [9, 10] para dos patrones artificiales, comenzando la iteración desde el vector nulo. Se utilizó como criterio de convergencia  $\|F(q^*)\| \leq 10^{-6} \cdot \sqrt{N}$ , donde  $q^*$  es la *shapelet* estimada [3].
  - *Experimento II*: Extender el experimento anterior para todos los patrones.

- *Experimento III*: Examinar el efecto de la elección de la aproximación inicial en la convergencia de los métodos. Para ello, se ejecutaron todos los pares posibles de aproximación inicial (obtenidas en el segundo experimento) y los 12 métodos numéricos considerados. Se seleccionaron las cinco mejores combinaciones utilizando un criterio de calidad, para elegir la más adecuada.
- *Experimento IV*: La combinación seleccionada se usó para evaluar su efectividad al detectar los patrones frente a otros filtros *wavelet*<sup>2</sup>. Para lograrlo, se insertaron los patrones en la posición 11 de 17 señales con 64 muestras de `pywt.data.demo_signal`.

La detección se basó en la medida de similitud normalizada

$$\mathbb{S} = e^{-(\text{DST-II}(f[\cdot]))^\alpha}, (0 < \alpha \leq 1);$$

que enfatiza la presencia de ceros en la  $\text{DST-II}(f[\cdot])$ , siendo  $f[\cdot]$  la señal analizada. En [9] se detectó el patrón en la posición donde la medida  $\mathbb{S}$  tomó valor 1. En los experimentos no se alcanzó este valor. Después de un análisis de dicha medida, se decidió tomar como criterio de detección del patrón el índice  $k$  del coeficiente *wavelet* de la *shapelet* adaptada con mayor valor de la medida  $\mathbb{S}(\alpha = 0, 15)$ . De esta forma, se estima que el patrón  $m[\cdot]$  comienza en  $f_{(k \cdot 2^{j-1})}$ . Para descomponer la señal se utilizó el algoritmo en [9].

En estos experimentos se evaluaron indicadores numéricos como el número de evaluaciones del sistema (Evals.), la norma del residual ( $\|\mathbf{F}(q^*)\|_2$ ), el tiempo de ejecución total (Tpo. Total), el tiempo de ejecución del algoritmo para resolver el sistema (Tpo. SENL), la similitud *shapelet*–patrón (Sim.) según el valor de la evaluación de la primera condición de detección en (2.1), y la regularidad de Sobolev de la *shapelet* mayor (Reg.). Además, se calcularon sus respuestas de frecuencia y de fase, y el diagrama de ceros de los filtros.

Numéricamente, la medida de regularidad se calculó mediante la fórmula [23]:

$$s_c = -\frac{\log \rho}{\log 4}, \quad \rho = \max_{\lambda \neq 1, \frac{1}{2}, \frac{1}{4}, \dots, (\frac{1}{2})^{2^{c-1}}} \{|\lambda(T)|\},$$

siendo  $c$  el número de ceros que tiene la transformada  $z$  de  $p[\cdot]$  en  $-1$ . La matriz de transición se define como  $T = (\downarrow 2)2HH^T$ , donde la matriz de Toeplitz  $H_{ij} = p[i-j]$  en la que  $p[k]$  es el filtro de descomposición de paso bajo, cuyos coeficientes están normalizados

$$\sum_k p[k] = 1.$$

A mayor  $s_c$ , mayor suavidad de la función de escala [23].

<sup>2</sup>Haar, Daubechies con soporte 4, 6, 8, 10, 20, 30 y 38, y Symlet con soporte 8 y 16, Coiflets con soporte 6, 12 y 16.

Se utilizó la estadística descriptiva sobre los indicadores numéricos. Las pruebas  $t$  para muestras relacionadas [7] comprobaron si las diferencias de medias de la pre-iteración y la post-iteración fueron estadísticamente significativas y su tamaño del efecto. Se realizaron análisis de varianza (ANOVA) de una vía [7] para comparar las medias de los indicadores numéricos, según los métodos usados para resolver el SENL. De existir diferencias, se ejecutó la prueba *post-hoc* HSD de Tukey para averiguar qué métodos difieren. Estos análisis estadísticos, la matriz de confusión y las curvas ROC de la detección se obtuvieron usando el lenguaje R.

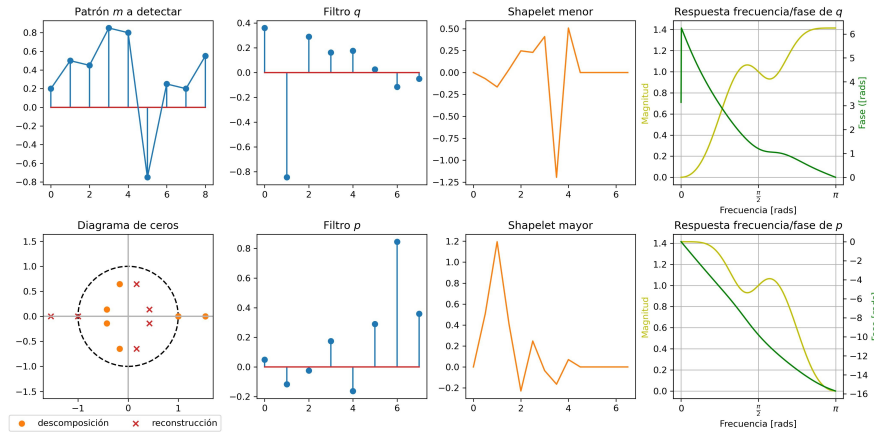
## 5 Resultados y discusión

### 5.1 Shapelets de patrones artificiales

#### 5.1.1 Patrón en [9]

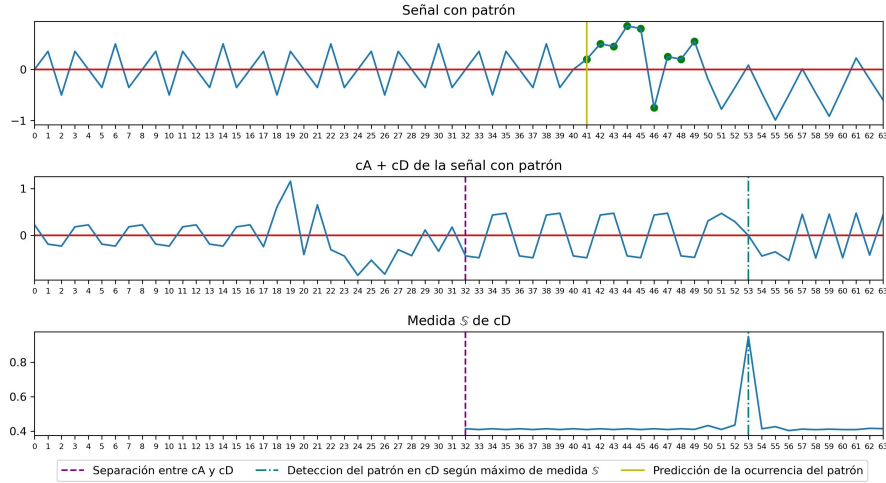
Se diseñó una DST-II con tamaño de soporte  $N = 8$  para detectar el patrón de coordenadas:  $(0, 0, 20)$ ;  $(1, 0, 50)$ ;  $(2, 0, 45)$ ;  $(3, 0, 85)$ ;  $(4, 0, 80)$ ;  $(5, -0, 75)$ ;  $(6, 0, 25)$ ;  $(7, 0, 20)$  y  $(8, 0, 55)$ .

El método `lm` convergió con mayor precisión y obtuvo la mayor similitud con el patrón. El método `hybr` fue el más rápido en converger. El método `newton` realizó menos evaluaciones del sistema. Excepto el método `hybr`, todos alcanzaron la misma regularidad de la *shapelet*. Para el método `hybr` se obtuvo una *shapelet* de fase no lineal con una respuesta de frecuencia muy oscilante respecto al punto de corte  $\frac{\pi}{2}$ . Para el resto de los métodos la fase fue casi lineal y se alcanzó una mejor diferenciación en frecuencia, aunque todavía oscilante. En la Figura 1 se muestra el resultado de menor norma del residual obtenido con el método `lm`.



**Figura 1:** Resultado de menor valor de  $\|\mathbf{F}(q^*)\|_2$  del método `lm` para el patrón en [9]:  
 Evals. = 68;  $\|\mathbf{F}(q^*)\|_2 = 8,51 \cdot 10^{-17}$ ; Tpo. Total = 0,31; Tpo. SENL = 0,01;  
 Sim. =  $-6,24 \cdot 10^{-17}$ ; Reg. = 0,71.

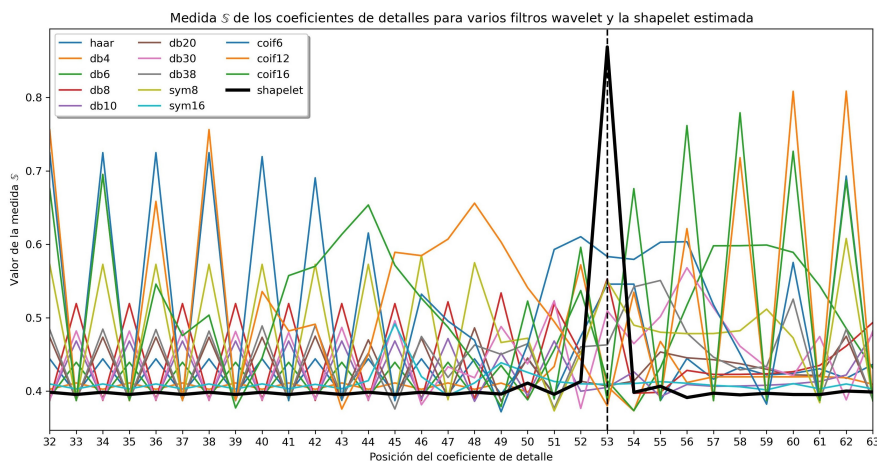
Se utilizó la *shapelet* obtenida por el método `lm` para detectar el patrón en la posición 41 dentro de la señal de 64 muestras propuesta en [9]. La Figura 2 presenta dicha señal con el patrón destacado en verde (arriba), la DST-II de la señal (centro) y la medida  $\mathbb{S}$  de los coeficientes de detalle, y debajo se marca el coeficiente seleccionado por el algoritmo de detección (línea discontinua azul claro). Arriba, la línea amarilla señala la predicción del patrón. Una línea discontinua morada separa los coeficientes de aproximación y detalle.



**Figura 2:** Localización del patrón de [9] insertado en una señal usando la *shapelet* obtenida por el método `lm`.

Se confirmó la presencia del patrón dentro de la señal con el máximo valor de la medida  $\mathbb{S} = 0,95$ , en lugar del valor  $\mathbb{S} = 1$  obtenido en [9], en el coeficiente *wavelet* número 53. Este punto muestral de la señal transformada corresponde al índice  $k = 21$  de la subbanda analizada; ya que, para una DST-II de primer nivel de una señal de 64 muestras de longitud, la parte correspondiente a los coeficientes de detalle comienza en el índice 32, que corresponde al índice 0 de dicha subbanda. Por tanto, el patrón comienza en  $f_{(k \cdot 2^{j-1})} = f_{(21 \cdot 2^{1-1})} = f_{41}$ , que coincide con el resultado obtenido en [9].

Posteriormente, se comparó esta detección con la de otros filtros *wavelet*, computando los errores de predicción y el máximo de la medida  $\mathbb{S}$ . La Figura 3 muestra la medida  $\mathbb{S}$  de los coeficientes de detalle de la DST-II aplicada a la señal para la *shapelet* (en negro) y los filtros *wavelet*. Se resalta en línea discontinua la muestra 53, que es la posición donde debe detectarse el patrón. Se observó que solo tres filtros (Haar, Daubchies 8 y la *shapelet*) detectaron el patrón, pero esta última alcanzó un mayor valor de la medida  $\mathbb{S} = 0,95$ .



**Figura 3:** Medida  $\mathbb{S}$  para la *shapelet* (en negro) y otros 13 filtros *wavelet* aplicados a la señal con el patrón de [9].

### 5.1.2 Patrón en [10]

Se diseñó una DST-II con tamaño de soporte  $N = 12$  para que coincida con el patrón de coordenadas: (0, 0, 01); (1, 0, 01); (2, 0, 01); (3, 0, 02); (4, 0, 05); (5, 0, 23); (6, 0, 62); (7, 0, 90); (8, 0, 98); (9, 0, 88); (10, 0, 02); (11, 0, 01) y (12, 0, 01).

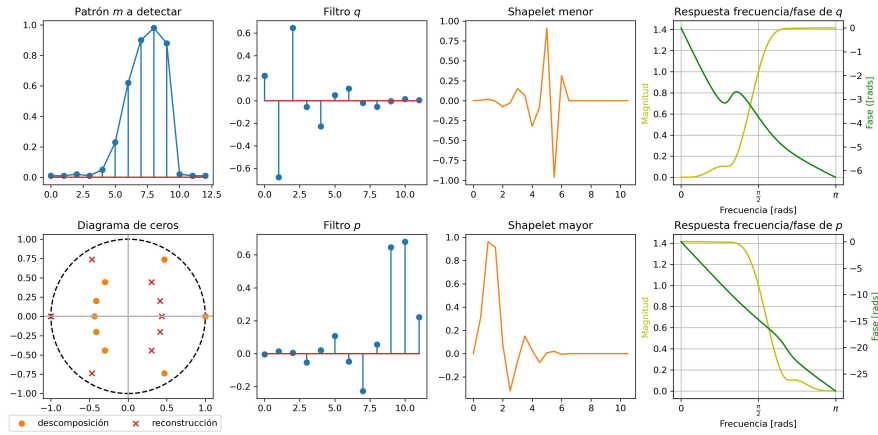
Nuevamente, el método *lm* convergió con mayor precisión. El método *hybr* fue el más rápido en converger. El método *continuation* realizó menos evaluaciones del sistema. La mayor similitud con el patrón se alcanzó con el método *hybr*. La mayor regularidad se obtuvo para *newton hybr*. Las *shapelets* estimadas presentaron fase casi lineal y una adecuada diferenciación de frecuencias respecto al punto de corte  $\frac{\pi}{2}$ , excepto para el método *continuation* cuya respuesta de frecuencia fue oscilante. En la Figura 4 se muestra el resultado de menor norma del residual para el método *lm*.

Los resultados de detección del patrón en la posición 41 de la señal de 64 muestras en [10], se muestra en la Figura 5. Nuevamente, se confirmó la presencia del patrón dentro de la señal con el máximo valor de la medida  $\mathbb{S} = 0,95$ .

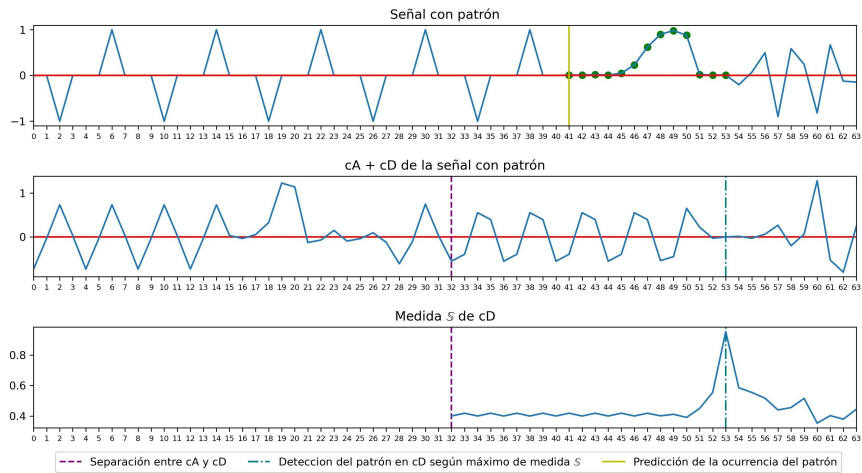
La Figura 6 compara la detección usando esta *shapelet* y otros filtros *wavelet*. Se observó que solo dos filtros (Daubechies 8 y la *shapelet*) detectaron con precisión el patrón, esta última reportó un mayor valor de  $\mathbb{S} = 0,95$ .

En resumen, el método *lm* convergió con mayor precisión y la *shapelet* obtenida fue muy similar al patrón, su regularidad fue baja, presentó una adecuada diferenciación de frecuencias y fase casi lineal. Respecto a otros filtros *wavelet*, la *shapelet* detectó con éxito el patrón con valor de la medida  $\mathbb{S} \approx 1$ . Surge entonces la

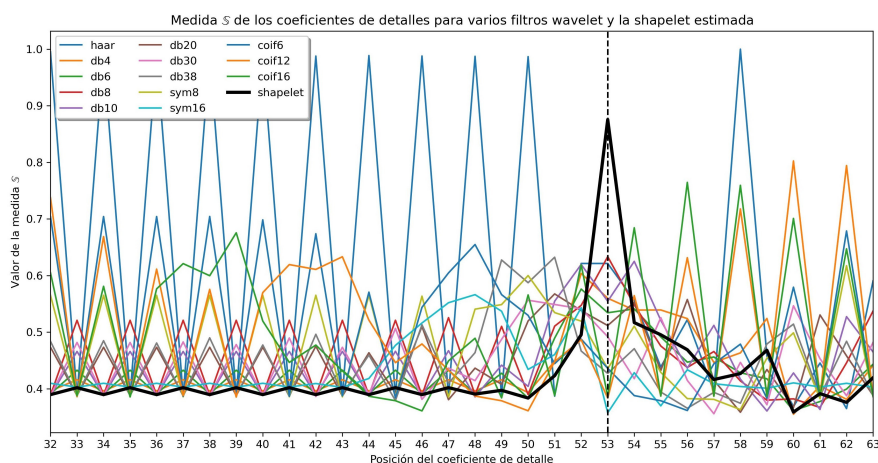
pregunta: ¿el comportamiento observado en este experimento podrá generalizarse para una variedad mayor de patrones? Para ello se hizo el siguiente experimento.



**Figura 4:** Resultado de menor valor de  $\|\mathbf{F}(q^*)\|_2$  del método 1m para el patrón en [10]: Evals. = 135;  $\|\mathbf{F}(q^*)\|_2 = 7,12 \cdot 10^{-15}$ ; Tpo. Total = 0,65; Tpo. SENL = 0,005; Sim. =  $2,11 \cdot 10^{-17}$ ; Reg. = 1,96.



**Figura 5:** Localización del patrón de [10] insertado en una señal usando el método 1m con aproximación inicial nula.



**Figura 6:** Medida  $S$  para la *shapelet* (en negro) y otros filtros *wavelet* aplicados a la señal con el patrón de [10].

## 5.2 Solución del SENL a partir de la aproximación nula

Se estimaron las *shapelets* para 103 patrones, utilizando 12 métodos de solución del SENL partiendo del vector nulo. Así, se realizaron 1 236 ejecuciones; de ellas, 185 corridas (14,97 %) no terminaron por mal condicionamiento de las matrices involucradas en los cálculos, errores de *overflow* o la evaluación del inverso del jacobiano en el vector nulo.

La convergencia se alcanzó en 102 de las 1 051 corridas que terminaron (9,70 %). Los algoritmos utilizados para resolver el SENL convergieron para un total de 45 patrones (43,69 %), no siendo así para los patrones *spike*, *spike* medio estándar, Vaidyanathan 24, Daubechies 6, 7, 13-38; Coiflets 4-17; Symlets 7, 12-20, y en tres de los complejos K.

El método *continuation* fue el que más veces convergió (39 patrones, 37,86 %), seguido de *newton* (32; 31,07 %), *hybr* (15; 14,56 %) y *lm* (15; 14,56 %). En el caso del método *broyden1* solo convergió una vez. El resto de los métodos no convergió para ningún patrón.

En los casos donde hubo convergencia, esta se logró para patrones con tamaño entre 8 y 24 muestras. Pudo observarse que, para patrones con más de 22 muestras, decreció la cantidad de veces que se logró la convergencia, y a partir de las 26 muestras no se alcanzó la convergencia.

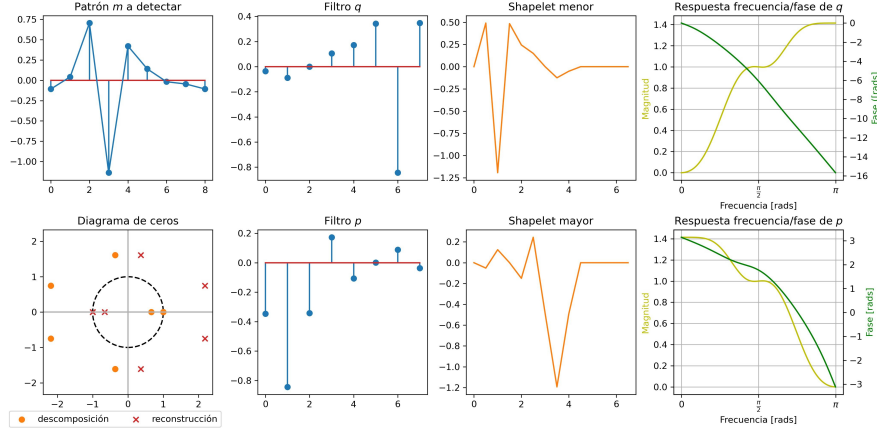
A continuación se analizaron los indicadores numéricos para las corridas donde hubo convergencia. La mediana de la cantidad de evaluaciones del SENL fue 135, desde un mínimo de 26 hasta un máximo de 986 evaluaciones. En promedio, estimar una *shapelet* demoró 4,11 segundos ( $DE = 4,13$ ); de ellos, se emplearon como

promedio 0,71 segundos ( $DE = 1,01$ ) para resolver el sistema. Se obtuvo una regularidad promedio de 1,52 ( $DE = 0,73$ ), alcanzándose una regularidad máxima de 2,83. En todos los casos, la condición de detección del SENL (2.1) se evaluó con valores menores que  $10^{-7}$ .

Se reportó un efecto significativo del algoritmo numérico usado sobre el número de evaluaciones del sistema,  $F(4) = 23,23$ ,  $p < 0,001$ ,  $\eta^2 = 0,49$ . Las pruebas *post-hoc* revelaron diferencias significativas entre el método **hybr** y **continuation** ( $t = -3,61$ ,  $p < 0,01$ ), **lm** y **continuation** ( $t = 5,16$ ,  $p < 0,001$ ), **newton** y **continuation** ( $t = -5,15$ ,  $p < 0,001$ ), **lm** y **hybr** ( $t = 7,29$ ,  $p < 0,001$ ) y **newton** y **lm** ( $t = -8,94$ ,  $p < 0,001$ ).

También se halló un efecto significativo del algoritmo numérico usado sobre el tiempo de ejecución,  $F(4) = 6,18$ ,  $p < 0,001$ ,  $\eta^2 = 0,20$ . Las pruebas *post-hoc* mostraron diferencias significativas entre el método **hybr** y **continuation** ( $t = -3,86$ ,  $p < 0,01$ ), **lm** y **continuation** ( $t = -3,69$ ,  $p < 0,01$ ), **newton** y **hybr** ( $t = 3,08$ ,  $p < 0,05$ ), y **newton** y **lm** ( $t = 2,92$ ,  $p < 0,05$ ).

En la Tabla 1 se reportan los indicadores numéricos de la solución con menor  $\|\mathbf{F}(q^*)\|_2$ , para los métodos que convergieron. La Figura 7 muestra el resultado de menor norma del residual usando el método **continuation**. En esta figura se observa la alta similitud entre el patrón y la *shapelet* menor, así como una adecuada respuesta de frecuencia, aunque oscilante, y una fase casi lineal, por lo que las *wavelets* obtenidas fueron casi simétricas. La simetría y la ortogonalidad no son posibles simultáneamente, excepto para la *wavelet* de Haar [16].



**Figura 7:** Resultado de menor norma del residual usando el método **continuation** desde el vector nulo. Patrón = Symlet 4; Dim. = 8; Evals. = 348;  $\|\mathbf{F}(q^*)\|_2 = 9,25 \cdot 10^{-13}$ ; Tpo. SENL = 0,37; Sim. =  $6,94 \cdot 10^{-18}$ ; Reg. = 0,70.

En suma, al comparar los algoritmos de solución del SENL desde el vector nulo, se alcanzó la convergencia para un número reducido de corridas (9,70%), aunque



se estimaron *shapelets* para el 43,69 % de los patrones. El método *continuation* fue el que más veces convergió. Se logró la convergencia para patrones con tamaño de hasta 24 muestras. En promedio, los métodos se ejecutaron rápidamente. Se reportó un efecto significativo del algoritmo sobre el número de evaluaciones del sistema y el tiempo total de ejecución. Es lógico entonces preguntarse: ¿mejorarán los parámetros numéricos y de detección si se realiza una pre-iteración con alguno de los métodos mencionados?

### 5.3 Solución del SENL con pre-iteración

Se realizaron 13 596 ejecuciones; de ellas, 4 119 corridas (30,29 %) no terminaron debido a errores numéricos. La convergencia se alcanzó en 1 129 de las 9 477 corridas (11,91 %). Los algoritmos convergieron para 57 patrones (55,34 %). Sin embargo, no fue así para los patrones *spike* medio estándar, Daubechies 7, 13-38; Coiflets 4-17; Symlets 7, 12-20, y en solo uno de los complejos K.

Posteriormente, se analizaron los indicadores numéricos para las combinaciones donde hubo convergencia. La mediana de la cantidad de evaluaciones del SENL fue 62, desde un mínimo de 3 hasta un máximo de 18 589 evaluaciones. En promedio, estimar una *shapelet* demoró 27,94 segundos ( $DE = 414,77$ ); de ellos, se emplearon como promedio 4,23 segundos ( $DE = 90,01$ ) para resolver el sistema. Se obtuvo una regularidad promedio de 1,57 ( $DE = 0,66$ ), alcanzándose un máximo de 3,58. En todos los casos, la condición de detección del SENL se evaluó con valores menores que  $10^{-7}$ .

Hubo un efecto significativo del algoritmo numérico usado sobre el número de evaluaciones del SENL,  $F(4) = 23,23$ ,  $p < 0,001$ ,  $\eta^2 = 0,49$ . Las pruebas post-hoc revelaron diferencias significativas entre *hybr* y *continuation* ( $t = -3,61$ ,  $p < 0,01$ ), *lm* y *continuation* ( $t = 5,16$ ,  $p < 0,001$ ), *newton* y *continuation* ( $t = -5,15$ ,  $p < 0,001$ ), *lm* y *hybr* ( $t = 7,29$ ,  $p < 0,001$ ) y *newton* y *lm* ( $t = -8,94$ ,  $p < 0,001$ ).

También hubo un efecto significativo del algoritmo numérico sobre el tiempo de ejecución,  $F(4) = 6,18$ ,  $p < 0,001$ ,  $\eta^2 = 0,20$ . Las pruebas post-hoc revelaron diferencias significativas entre *hybr* y *continuation* ( $t = -3,86$ ,  $p < 0,01$ ), *lm* y *continuation* ( $t = -3,69$ ,  $p < 0,01$ ), *newton* y *hybr* ( $t = 3,08$ ,  $p < 0,05$ ), y *newton* y *lm* ( $t = 2,92$ ,  $p < 0,05$ ). No se hallaron diferencias respecto a la regularidad.

Dada la cantidad de combinaciones de métodos ( $12 \times 11 = 132$ ), se decidió establecer un *ranking* para elegir las cinco primeras combinaciones con mejor desempeño. Para ello, se contó la cantidad de veces que convergió cada combinación y se consideraron las combinaciones con: i) menor valor promedio de  $\|\mathbf{F}(q^*)\|_2$ , ii) menor promedio de evaluaciones del SENL, iii) menor tiempo promedio para solucionarlo, iv) mayor similitud promedio entre la *shapelet* y el patrón (i.e., valores de la primera condición de detección en (2.1) más cercanos a 0), así como v) mayor

regularidad promedio. En la Tabla 2 se reportan los indicadores numéricos de estas combinaciones ordenados según el criterio explicado.

La combinación `continuation->newton` fue la que más veces convergió y disminuyó muy significativamente el número de evaluaciones de funciones de la post-iteración ( $M = 59,68$ ;  $EE = 6,08$ ) respecto a la pre-iteración ( $M = 296,52$ ;  $EE = 17,23$ ),  $t(61) = 12,84$ ;  $p < 0,001$ ;  $r = 0,85$ .

La combinación `continuation->lm` mostró una disminución muy significativa del tiempo total promedio de la post-iteración ( $M = 1,36$ ,  $EE = 0,16$ ) respecto a la pre-iteración ( $M = 0,10$ ;  $EE = 0,04$ ),  $t(61) = 7,32$ ;  $p < 0,001$ ;  $r = 0,68$ ; además de un aumento significativo del tiempo promedio para solucionar el SENL de la post-iteración ( $M = 1,05$ ;  $EE = 0,12$ ) respecto a la pre-iteración ( $M = 0,78$ ;  $EE = 0,17$ ),  $t(61) = -2,73$ ;  $p < 0,01$ ;  $r = 0,33$ .

La combinación `lm->newton` reportó una disminución muy significativa del número de evaluaciones de la post-iteración ( $M = 54,56$ ;  $EE = 5,96$ ) respecto a la pre-iteración ( $M = 1351,76$ ;  $EE = 216,29$ ),  $t(61) = 5,96$ ;  $p < 0,001$ ;  $r = 0,61$ . También, se halló una disminución significativa del tiempo total promedio de la post-iteración ( $M = 2,57$ ;  $EE = 0,78$ ) respecto a la pre-iteración ( $M = 7,44$ ;  $EE = 0,92$ ),  $t(61) = 3,82$ ;  $p < 0,001$ ;  $r = 0,44$ ; y un aumento muy significativo de la regularidad de la *shapelet* de la post-iteración ( $M = 0,94$ ;  $EE = 0,15$ ) respecto a la pre-iteración ( $M = -0,0003$ ;  $EE = 0,0008$ ),  $t(61) = -6,22$ ;  $p < 0,001$ ;  $r = 0,62$ .

La combinación `continuation->hybr` mostró una disminución muy significativa del número de evaluaciones de la post-iteración ( $M = 122,79$ ;  $EE = 21,76$ ) respecto a la pre-iteración ( $M = 292,68$ ;  $EE = 17,41$ ),  $t(61) = 7,07$ ;  $p < 0,001$ ;  $r = 0,67$ . Asimismo, una disminución muy significativa del tiempo total promedio de la post-iteración ( $M = 0,03$ ;  $EE = 0,006$ ) respecto a la pre-iteración ( $M = 1,35$ ;  $EE = 0,16$ ),  $t(61) = 8,08$ ;  $p < 0,001$ ;  $r = 0,72$ .

Por último, para la combinación `continuation->hybr` se reportó una disminución muy significativa del número de evaluaciones de la post-iteración ( $M = 245,35$ ;  $EE = 22,70$ ) respecto a la pre-iteración ( $M = 1351,76$ ;  $EE = 216,29$ ),  $t(61) = 4,96$ ;  $p < 0,001$ ;  $r = 0,54$ . También, se evidenció una disminución significativa del tiempo total promedio de la post-iteración ( $M = 2,32$ ;  $EE = 0,89$ ) respecto a la pre-iteración ( $M = 7,44$ ;  $EE = 0,92$ ),  $t(61) = 3,84$ ;  $p < 0,001$ ;  $r = 0,44$ ; y una disminución muy significativa de la similitud de la *shapelet* al patrón, en la post-iteración ( $M = -9,00 \cdot 10^{-5}$ ;  $EE = 7,38 \cdot 10^{-5}$ ) respecto a la pre-iteración ( $M = 0,0005$ ;  $EE = 8,11 \cdot 10^{-5}$ ),  $t(61) = 6,00$ ;  $p < 0,001$ ;  $r = 0,61$ . Se obtuvo un aumento muy significativo de la regularidad de la *shapelet* de la post-iteración ( $M = 0,80$ ;  $EE = 0,17$ ) respecto a la pre-iteración ( $M = -0,0003$ ;  $EE = 0,0008$ ),  $t(61) = -4,72$ ;  $p < 0,001$ ;  $r = 0,52$ ; aunque el valor promedio de la regularidad para la post-iteración fue relativamente bajo.

La Figura 8 muestra la *shapelet* de menor norma del residual para `continuation->newton`. Las pocas evaluaciones del sistema, los valores muy ba-

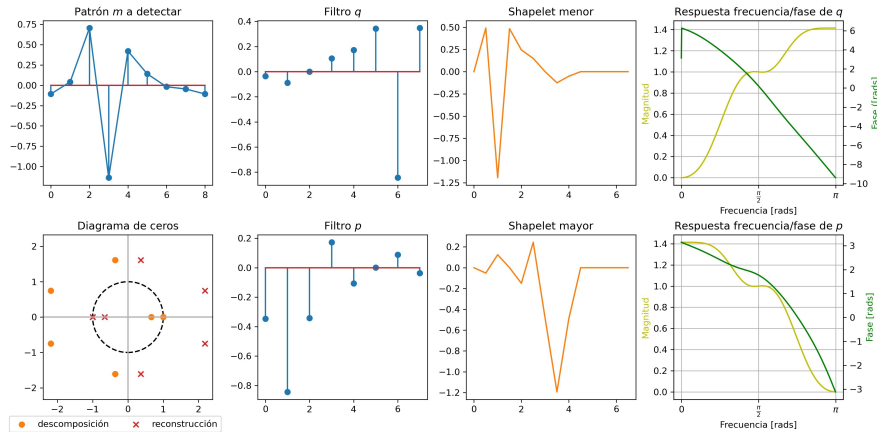
Patrón	Método	Dim.	Evals.	$\ F(q^*)\ _2$	Tpo. Total	Tpo. SENL	Sim.	Reg.
Symlet 4	continuation	8	348	$9,25 \cdot 10^{-13}$	1,548	0,373	$-6,94 \cdot 10^{-18}$	0,70
Guido2018	newton	8	54	$2,57 \cdot 10^{-16}$	0,488	0,147	$4,86 \cdot 10^{-17}$	0,71
Complejo K #41	hybr	16	100	$6,79 \cdot 10^{-13}$	1,196	0,005	$3,12 \cdot 10^{-19}$	1,81
Guido2018	lm	8	68	$8,51 \cdot 10^{-17}$	0,315	0,011	$-6,24 \cdot 10^{-17}$	0,71
Daubechies 4	broyden1	8	136	$1,32 \cdot 10^{-6}$	0,212	0,015	$-1,97 \cdot 10^{-7}$	2,01

**Tabla 1:** Indicadores numéricos de los cinco mejores resultados de convergencia con aproximación inicial nula.

Combinación	Frec. convergencia	Prom. $\ F(q^*)\ _2$	Prom. Evals.	Tpo. SENL	Prom. Prom.	Prom. Sim.	Prom. Reg.
continuation->newton	39	$5,26 \cdot 10^{-7}$	34,72	1,55	$7,31 \cdot 10^{-10}$	1,44	
continuation->lm	38	$9,51 \cdot 10^{-8}$	72,13	0,05	$1,82 \cdot 10^{-10}$	1,50	
lm->newton	38	$1,38 \cdot 10^{-7}$	24,60	2,78	$2,36 \cdot 10^{-11}$	1,56	
continuation->hybr	36	$6,41 \cdot 10^{-9}$	31,30	0,01	$2,75 \cdot 10^{-17}$	1,54	
lm->continuation	36	$3,92 \cdot 10^{-7}$	103,89	3,33	$3,39 \cdot 10^{-11}$	1,61	

**Tabla 2:** Descripción de indicadores numéricos de las cinco mejores combinaciones ordenadas según criterio de *ranking*.

jos del residual y la alta similitud con el patrón indican que la pre-iteración fue efectiva.



**Figura 8:** *Shapelet* de menor valor para  $\|\mathbf{F}(q^*)\|_2$  la combinación `continuation->newton`. Patrón = Symlet 4; Dim. = 8; Evals. = 3;  $\|\mathbf{F}(q^*)\|_2 = 7,02 \cdot 10^{-17}$ ; Tpo. SENL = 0,06; Sim. =  $6,94 \cdot 10^{-18}$ ; Reg. = 0,70.

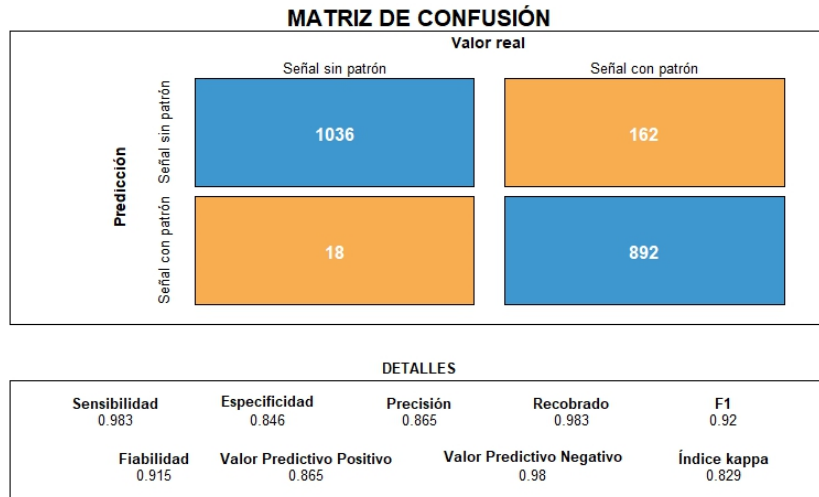
En resumen, la convergencia se alcanzó para el 11,91 % de las corridas realizadas y se pudieron estimar *shapelets* para el 55,34 % de los patrones. De las cinco combinaciones que más convergieron se eligió `continuation->newton`, pues fue la que más convergió, y además reportó una disminución muy significativa del número de evaluaciones de funciones de la post-iteración respecto a la pre-iteración, lo que es un indicador de la velocidad de convergencia. La combinación escogida realizó pocas evaluaciones del sistema, obtuvo valores muy bajos del residual y una alta similitud con el patrón. Esto indica que la post-iteración fue efectiva. En ocasiones, las respuestas de frecuencia son oscilantes alrededor del punto de corte, aunque las respuestas de fase son casi lineales.

## 5.4 Análisis de forma y efectividad de la detección

En el último experimento se insertaron los patrones en 17 señales de 64 muestras, siempre en la posición 11. Se realizó la detección con la *shapelet* obtenida mediante la combinación `continuation->newton`, y se comparó con otros filtros *wavelet*. Además, se realizó la detección en las mismas señales, sin insertar el patrón.

En la Figura 9 se muestra que, al usar las *shapelets* estimadas para cada patrón, se alcanzó una muy alta sensibilidad, 0,98 (i.e., la mayor parte de los patrones insertados en la posición 11 fueron detectados correctamente) y una alta especificidad, 0,85 (i.e., si el patrón no estuvo en la señal, este no es detectado por el algoritmo). Igualmente, se obtuvo un valor predictivo positivo alto, 0,86 (i.e., sabiendo que el algoritmo detectó el patrón, cuál es la probabilidad de que este

ocurra realmente en la señal) y un muy alto valor predictivo negativo, 0,98 (i.e., sabiendo que el algoritmo no detectó el patrón, cuál es la probabilidad de que este realmente no esté).



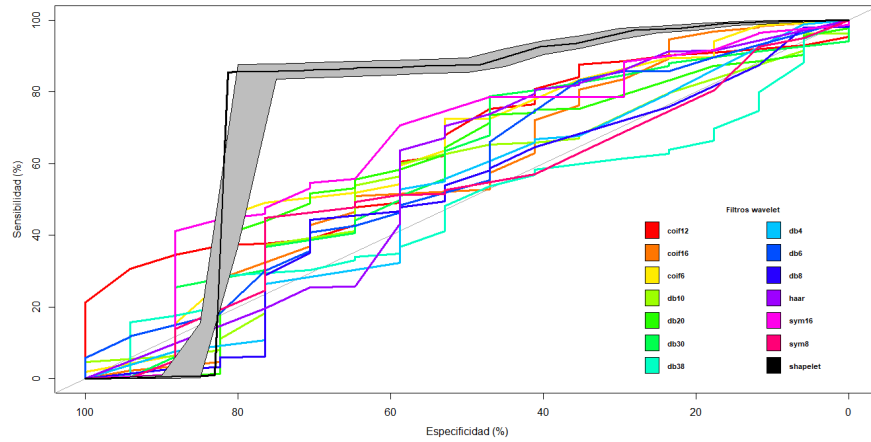
**Figura 9:** Matriz de confusión sobre la efectividad de la detección de los patrones usando las *shapelets* correspondientes.

El algoritmo de detección al usar las *shapelets* alcanzó un área bajo la curva ROC = 75,83% ( $IC_{95\%} = [73,5\% - 78,2\%]$ ) y fue superior al resto de los filtros *wavelet* (Figura 10). En dicha figura se destaca en color negro la curva de las *shapelets* con su intervalo de confianza al 95%.

## 6 Conclusiones

Los resultados obtenidos demostraron el impacto de la elección del método numérico iterativo de solución del sistema de ecuaciones no lineales, así como de la aproximación inicial, en la construcción de la *shapelet*. La gran influencia que tiene el método numérico en la eficacia de los filtros estimados que conforman la *shapelet* repercute en el desempeño de esta transformada en la detección de patrones.

El método de Newton con pre-iteración mediante el algoritmo de continuación fue la combinación que convergió más frecuentemente (para el 37,86% de los patrones) y reportó una disminución significativa del número de evaluaciones de funciones en la post-iteración (de una mediana de 296,52 a 59,68 evaluaciones). Además, los filtros estimados con esta combinación alcanzaron valores muy bajos del residual (del orden de  $10^{-7}$ ), alta similitud con el patrón (del orden de  $10^{-10}$ ) y una regularidad promedio de la *shapelet* de 1,44.



**Figura 10:** Curva ROC de la detección de los patrones usando las *shapelets* y 13 filtros *wavelet*.

Al utilizar la DST-II para la detección de patrones en diferentes señales usando los filtros estimados por esta vía, se alcanzaron valores de sensibilidad = 0,98, especificidad = 0,85, valor predictivo positivo = 0,86, valor predictivo negativo = 0,98 y  $AUC = 75,83\%$ , considerados como altos. Estos filtros mostraron su superioridad en la detección de patrones respecto a otros 13 filtros *wavelets*. Por tanto, se recomienda el método de Newton con pre-iteración mediante el algoritmo de continuación para estimar los filtros en una DST-II.

Adicionalmente, se replicaron parcialmente los resultados originales [9, 10], pues en estas publicaciones no se explicitó el método numérico usado para hallar el filtro, ni se reportaron indicadores sobre la convergencia, el tiempo de ejecución, la respuesta de frecuencia y fase, entre otros.

Otro resultado importante de este trabajo está relacionado con el hecho de que numéricamente se pueden estimar filtros *shapelets* para cualquier tipo de patrón, usando un criterio de convergencia sobre el residual. Sin embargo, esto ocurrió solo para el 55,37% de los patrones considerados. Esto sugiere que la DST-II podría ser adecuada para patrones con formas específicas (por ejemplo, simétricos, como se menciona en [10]), lo cual sería de utilidad en métodos de detección de patrones específicos en algunas aplicaciones del procesamiento y análisis de señales e imágenes biomédicas. En próximas investigaciones se utilizará esta técnica para detectar patrones tipo complejo K en electroencefalogramas y anomalías tipo masa en imágenes de mamografía digital.

## Agradecimientos

Agradecemos a Proyecto “Métodos numéricos para problemas en múltiples escalas”, Programa Nacional de Ciencias Básicas, PN223LH010-003, Ministerio de Ciencia, Tecnología y Medio Ambiente (CITMA), Cuba; Programa Erasmus+ “Student Mobility for Traineeships” (2021), Universidad de las Islas Baleares y la Universidad de La Habana; Proyecto I+D+i PID2019-104829RAI00 - “EXPLainable Artificial INtelligence systems for health and well-beING (EXPLAINING)”, MCIN/AEI/10.13039/501100011033; y Proyecto I+D+i PID2020-113870GB-I00 - “Desarrollo de herramientas de Soft Computing para la Ayuda al Diagnóstico Clínico y a la Gestión de Emergencias (HESOCODICE)”, MCIN/AEI/10.013039/501100011033.

## Referencias

- [1] A. Aldroubi, P. Abry y M. Unser, *Construction of biorthogonal wavelets starting from any two multiresolutions*. IEEE Transactions on Signal Processing **46**(1998), no. 4, 1130-1133. DOI: [10.1109/78.668563](https://doi.org/10.1109/78.668563)
- [2] D. G. Anderson, *Iterative Procedures for Nonlinear Integral Equations*. Journal of the ACM **12**(1965), no. 4, 547-560. DOI: [10.1145/321296.321305](https://doi.org/10.1145/321296.321305)
- [3] E. G. Birgin, J. L. Gardenghi, D. S. Marcondes y J. M. Martínez, *Accelerated derivative-free spectral residual method for nonlinear systems of equations*. arXiv. 2021. DOI: [10.48550/arXiv.2104.13447](https://doi.org/10.48550/arXiv.2104.13447)
- [4] R. L. Burden, J. D. Faires y B. A. M., *Soluciones numéricas de sistemas de ecuaciones no lineales*. 10a ed. Cengage Learning, Mexico, D.F., 2017.
- [5] I. Daubechies, *Ten Lectures on Wavelets*. Society for Industrial and Applied Mathematics, Philadelphia, Pennsylvania, 1992.
- [6] S. Devuyst, *The DREAMS Databases and Assessment Algorithm*. Zenodo, ene. de 2005. DOI: [10.5281/zenodo.2650142](https://doi.org/10.5281/zenodo.2650142)
- [7] A. Field, J. Miles y Z. Field, *Discovering Statistics Using R*. SAGE Publications Limited, 2012.
- [8] R. C. Guido, *A note on a practical relationship between filter coefficients and scaling and wavelet functions of Discrete Wavelet Transforms*. Applied Mathematics Letters **24**(2011), no. 7, 1257-1259. DOI: [10.1016/j.aml.2011.02.018](https://doi.org/10.1016/j.aml.2011.02.018)
- [9] R. C. Guido, *Fusing time, frequency and shape-related information: Introduction to the Discrete Shapelet Transform's second generation (DST-II)*. Information Fusion **41**(2018), 9-15. DOI: [10.1016/j.inffus.2017.07.004](https://doi.org/10.1016/j.inffus.2017.07.004)

- [10] R. C. Guido, *Nearly symmetric orthogonal wavelets for time-frequency-shape joint analysis: Introducing the discrete shapelet transform's third generation (DST-III) for nonlinear signal analysis*. Communications in Nonlinear Science and Numerical Simulation **97**(2021), 1-12. DOI: [10.1016/j.cnsns.2020.105685](https://doi.org/10.1016/j.cnsns.2020.105685)
- [11] R. C. Guido et al., *Introduction to the Discrete Shapelet Transform and a new paradigm: Joint time-frequency-shape analysis*. 2008 IEEE International Symposium on Circuits and Systems (ISCAS). Institute of Electrical y Electronics Engineers (IEEE), 2008, 2893-2896. DOI: [10.1109/ISCAS.2008.4542062](https://doi.org/10.1109/ISCAS.2008.4542062)
- [12] D. Jawali, A. Kumar y C. S. Seelamantula, *A Learning Approach for Wavelet Design*. ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). Institute of Electrical y Electronics Engineers (IEEE), 2019, 5018-5022. DOI: [10.1109/ICASSP.2019.8682751](https://doi.org/10.1109/ICASSP.2019.8682751)
- [13] D. A. Knoll y D. E. Keyes, *Jacobian-free Newton–Krylov methods: a survey of approaches and applications*. Journal of Computational Physics **193**(2004), no. 2, 357-397. DOI: [10.1016/j.jcp.2003.08.010](https://doi.org/10.1016/j.jcp.2003.08.010)
- [14] W. La Cruz, J. M. Martínez y M. Raydan, *Spectral Residual Method without Gradient Information for Solving Large-Scale Nonlinear Systems of Equations*. Mathematics of Computation **75**(2006), no. 255, 1429-1448. URL: <http://www.jstor.org/stable/4100282>.
- [15] G. R. Lee et al., *PyWavelets: A Python package for wavelet analysis*. Journal of Open Source Software **4**(2019), no. 36, 1237. DOI: [10.21105/joss.01237](https://doi.org/10.21105/joss.01237)
- [16] S. Mallat, *A Wavelet Tour of Signal Processing: The Sparse Way*. 3rd. Academic Press, Burlington, Massachusetts, 2009. DOI: [10.1016/B978-0-12-374370-1.X0001-8](https://doi.org/10.1016/B978-0-12-374370-1.X0001-8)
- [17] H. Mesa, *Adapted Wavelets for Pattern Detection*. A. Sanfeliu y M. L. Cortés (Eds.). Progress in Pattern Recognition, Image Analysis and Applications. Iberoamerican Congress on Pattern Recognition. Springer, Berlin, Heidelberg, 2005, 933-944. DOI: [10.1007/11578079\\_96](https://doi.org/10.1007/11578079_96)
- [18] M. Misiti, Y. Misiti, G. Oppenheim y J.-M. Poggi, *Wavelets and their Applications*. ISTE Ltd, London, 2007. DOI: [10.1002/9780470612491](https://doi.org/10.1002/9780470612491)
- [19] J. J. Moré, B. S. Garbow y K. E. Hillstrom, *User guide for MINPACK-1*. Argonne National Laboratory (1980). DOI: [10.2172/6997568](https://doi.org/10.2172/6997568)
- [20] M. J. D. Powell, *A Hybrid Method for Nonlinear Equations*. Numerical Methods for Nonlinear Algebraic Equations. P. Rabinowitz (Ed.). Gordon and Breach, London, 1970, 87-114.
- [21] W. H. Press, S. A. Teukolsky, W. T. Vetterling y B. P. Flannery, *Numerical Recipes: The Art of Scientific Computing*. 3rd ed. Cambridge University Press, Cambridge, 2007.
- [22] G. Strang, *Linear Algebra and Learning from Data*. Wellesley-Cambridge Press, Wellesley, Massachusetts, 2019.



- [23] G. Strang y N. Truong, *Wavelets and filter banks*. 2nd ed. Wellesley-Cambridge Press, Wellesley, Massachusetts, 1997.
- [24] D. Valdés Santiago et al., *Exploratory study on estimation, detection and robustness of a Discrete Shapelet Transform II*. Revista Investigación Operacional **44**(2023), no. 2, 184-205. URL: <https://n9.cl/61jql>.