

DETECCIÓN AUTOMÁTICA DE COMPLEJOS K CON UNA VARIANTE DE TRANSFORMADA SHAPELET DISCRETA II

Damian Valdés-Santiago[†], Ángela León-Mecías y Marta L. Baguer Díaz-Romañach
Universidad de La Habana, Cuba.

RESUMEN

La señal de la electroencefalografía (EEG) contiene información importante sobre la actividad eléctrica del cerebro, que puede revelar muchas patologías. Esta información se transmite en determinadas formas de onda y eventos, uno de los cuales es el complejo K. La detección de estos patrones es esencial en estudios del sueño para diagnosticar trastornos neurofisiológicos y cognitivos. Los métodos de detección existentes dependen en gran medida de una inspección manual de la forma de onda del EEG tediosa, lenta y propensa a errores. Por ello, la identificación automática de los complejos K es de gran interés. En este artículo se construyen wavelets adaptadas para detectar complejos K usando una variante de la Transformada *Shapelet* Discreta II (DST-II). Esta transformada está diseñada para detectar patrones (como estos complejos K), localizándolos en tiempo y frecuencia. Esto se logra resolviendo numéricamente un sistema de ecuaciones no lineales que estima un filtro wavelet adaptado al patrón. En esta investigación se presenta una estrategia numérica de solución del sistema no lineal de la DST-II. Se evaluó la influencia de dicha estrategia en la detección de los complejos K en diferentes señales y la precisión de la detección frente a otros filtros wavelet clásicos.

PALABRAS CLAVE: Complejos K, EEG, Diseño de filtros wavelet, Wavelet adaptada, Transformada Shapelet Discreta.

MSC: 65T60, 42C40, 65F99, 65H10

ABSTRACT

The electroencephalography (EEG) signal contains important information about the electrical activity of the brain, which can reveal many pathologies. This information is transmitted in certain waveforms and events, one of which is the K-complex. Detection of these patterns is essential in sleep studies to diagnose neurophysiological and cognitive disorders. Existing detection methods rely heavily on tedious, time-consuming and error-prone manual inspection of the EEG waveform. Therefore, automatic identification of K-complexes is of great interest. In this paper, adapted wavelets are constructed to detect K-complexes using a variant of the Discrete *Shapelet* Transform II (DST-II). This transform is designed to detect patterns (such as these K-complexes) by localizing them in time and frequency. This is achieved by numerically solving a system of nonlinear equations that estimates a wavelet filter adapted to the pattern. In this research, a numerical strategy for solving the nonlinear system of DST-II is presented. The influence of such a strategy on the detection of K-complexes in different signals and the detection accuracy versus other classical wavelet filters were evaluated.

[†]Autor para correspondencia: mailto:dvs89cs@gmail.com

KEYWORDS: K-complex, EEG, Wavelet filter design, Adapted wavelet, Discrete Shapelet Transform.

1. INTRODUCCIÓN

El análisis visual del electroencefalograma (EEG) nocturno es una tarea que requiere mucho tiempo y no es consistente, pues los resultados subjetivos conducen a una tasa de concordancia muy baja entre diferentes calificadores del sueño [8]. Esto exige técnicas avanzadas de procesamiento de señales de EEG para realizar una detección fiable y automática de los microeventos del sueño.

En particular, el complejo K es una “onda aguda negativa bien delineada, seguida inmediatamente por un componente positivo que sobresale del EEG de fondo, con una duración total $\geq 0,5$ s, generalmente de máxima amplitud cuando se registra con derivaciones frontales” [3].

La detección fiable de los complejos K es esencial para la determinación de los estadios del sueño, ya que constituyen (junto con los husos) uno de los principales marcadores de la transición del estadio de sueño 1 o REM al estadio de sueño 2 (o 3 o 4). Lo ideal es que se observen claramente delimitados, pero en la práctica suelen ser difíciles de distinguir de las ondas agudas delta y vértice [8]. Su identificación visual requiere mucho tiempo y es muy subjetiva [16]. Por ello, la identificación automática de los complejos K es de gran interés.

En la literatura se presentan varios métodos para detectar complejos K en señales de EEG. Devuyt et al. presentan un método automático de detección de los complejos K basado en la extracción de características y el uso de umbrales difusos [6]. La propuesta se validó sobre la base de dos evaluadores que detectaron visualmente los complejos K en 5 extractos de 30 minutos de un EEG, pertenecientes a diferentes etapas del sueño. El algoritmo obtuvo una tasa de verdaderos positivos del 61,72% y una proporciones de falsos positivos del 19,62%.

Erdamar et al. [8] proponen un algoritmo que examina la morfología del complejo K y determina las características de amplitud y duración de su forma de onda. El algoritmo se basa en transformada wavelet y el operador de energía *teager*. Un análisis ROC mostró hasta un 91% de acierto en la detección de los complejos K.

En un estudio piloto desarrollado por Krohne et al. [15] un algoritmo semiautomático de detección de complejos K que utiliza la transformación wavelet para identificar pseudocomplejos K y varios umbrales de características para rechazar falsos positivos. El algoritmo obtuvo una tasa media de verdaderos positivos del 74% y un valor predictivo positivo del 65%.

Patti et al. [23] construyen una wavelet específica para la estructura de los complejos K para detectarlos. Los resultados mostraron una tasa de verdaderos positivos del 84% con un valor predictivo positivo del 62%.

Recientemente, desde un enfoque de aprendizaje automático, Dumitrescu et al. [7] proponen un nuevo método para la detección automática de complejos K combinando el método de recursión y reasignación de la clase Cohen y las redes neuronales profundas, obteniendo una estrategia recursiva orientada a aumentar el porcentaje de clasificación hasta un 98,3% y reducir el tiempo de cómputo (hasta 5 segundos) necesario para detectar complejos K.

Khasawneh et al. [13] desarrollaron un sistema de detección de complejos K de alta precisión basado en múltiples redes neuronales convolucionales con extracción de características e imágenes de forma

de onda EEG. Los modelos entrenados VGG19 e Inceptionv alcanzaron hasta un 99,8% de precisión y un error de 0,2%, en diferentes escenarios experimentales.

A pesar del éxito de estos métodos en la detección de complejos K, en particular, los que utilizan modelos de aprendizaje automático, es interesante considerar métodos que modelen los complejos K mediante filtros wavelet. Este enfoque permite estimar los filtros con solamente un ejemplo del patrón, lo que ahorraría la complejidad y el tiempo de entrenamiento de modelos basados en redes neuronales. Además, los filtros wavelet pueden cumplir propiedades muy satisfactorias para el procesamiento de señales como la multiescalaridad, el soporte compacto, la cantidad de momentos nulos, la ortogonalidad, la suavidad, así como adecuadas respuestas de frecuencia y fase [22, 18]. Es por ello, que en esta investigación se aplicará este enfoque para solucionar el problema de la detección de complejos K.

La literatura científica recoge varios métodos para construir wavelets adaptadas a patrones específicos, que pueden ser usados para detectar complejos K. Aldroubi et al. [1] obtuvieron nuevas wavelets aplicando convoluciones a otras ya definidas, minimizando la diferencia con el patrón. Mesa [19] utilizó un esquema *lifting* [20] para construir wavelets biortogonales adaptadas con varios momentos nulos [18]. Misiti et al. [20] 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 [11] construye wavelets adaptadas con la Transformada *Shapelet* Discreta II (DST-II), que determina el soporte en el tiempo no solo de las frecuencias, sino también de la forma de un patrón. La DST-II resuelve un sistema de ecuaciones no lineales (SENL) formado por restricciones de energía unitaria, momentos nulos, ortogonalidad y de correlación con el patrón. Según [11, p. 10 y p. 11], 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 experimentos de detección de siete patrones del sistema visual de una mosca, en los que 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 [26].

Por tanto, esta investigación propone un método novedoso para detectar patrones como los complejos K usando filtros wavelet adaptados mediante la DST-II. Para ello, se realiza un estudio numérico que demuestra la influencia de la elección del método iterativo de solución del sistema subyacente en la DST-II para construir dichos filtros. Se establecieron combinaciones de métodos numéricos para resolver dicho sistema y se evaluó la capacidad de los filtros para detectar los complejos K en 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 experimentos realizados para construir la wavelet adaptada al complejo K, así como su detección en diferentes señales. La discusión de los resultados se expone en la sección 5.. Al cierre, se presentan conclusiones.

2. TRANSFORMADA SHAPELET DISCRETA DE SEGUNDA GENERACIÓN (DST-II)

La DST-II [11] 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 [28]; 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: $\left(\sum_{k=0}^{N-1} q_k^2 \right) - 1 = 0$;
- $\frac{N}{2} - 2$ momentos nulos para la wavelet: $\sum_{k=0}^{N-1} q_k \cdot k^b = 0$, donde $b = 0, 1, \dots, \frac{N}{2} - 3$;
- $\frac{N}{2} - 1$ condiciones de ortogonalidad: $\sum_{k=0}^{N-1} q_k \cdot q_{k+2l} = \delta_{0,l}$, donde δ 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) [28] de una DST-II se computan:

- $p[\cdot]$ 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
- $\bar{p}[\cdot]$ 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 [10]. 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 [28].

3. MÉTODOS NUMÉRICOS PARA HALLAR EL FILTRO DE PASO ALTO DE UNA DST-II

Para obtener los coeficientes del filtro de paso alto q , adaptado para detectar el complejo K en una DST-II, 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 [26]). Puede que la matriz del sistema sea singular, por lo que se utilizó la pseudoinversa del jacobiano¹.

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 $\mathcal{O}(n^2)$ 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 `LMDF1` de `MINPACK-1` [21]. 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.

El número de operaciones aritméticas que necesita `LMDF1` es $\mathcal{O}(n^3)$ para procesar cada evaluación de función y $\mathcal{O}(mn^2)$ para procesar cada aproximación al jacobiano. `LMDF1` 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 [25] tienen como base un algoritmo cuasi-Newton. Estos se diferencian en la forma de estimar la matriz jacobiana [26].

La matriz B_k es una hessiana construida a partir 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.

¹La 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, Σ es una matriz diagonal que consiste en los llamados valores singulares de A y luego Σ^+ es la matriz diagonal que consiste en los recíprocos de los valores singulares de A [27].

3.4. Método de Powell

El método de Powell [24, 25], en cada iteración, si el paso del algoritmo de Newton está dentro de la región de confianza, este se utiliza para actualizar la solución actual. 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 [24, 25].

Este método está implementado en MINPACK-1 [21] 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 [24].

3.5. Método de Anderson

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

$$\bar{x}^{(k)} := x^{(k)} + \sum_{j=1}^M \vartheta_j^{(k)} \left(x^{(k-j)} - x^{(k)} \right), \text{ donde } 0 \leq M \leq k - 1.$$

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 [14], 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. [17] 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 [17], 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 [26] 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 λ 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, λ_0 se traslada a λ_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, 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 28 complejos K extraídos de electroencefalogramas en la base de datos DREAMS [5]. La base de datos DREAMS© K-Complex contiene 30 minutos de datos de diez sujetos [5]. Cinco de estos sujetos fueron puntuados de forma independiente por dos expertos en sueño, por lo que en este estudio se utilizó el canal EEG CZ-A1 de estos sujetos. Los datos se extrajeron de grabaciones de PSG de toda la noche con una frecuencia de muestreo de 200 Hz.

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 10 y 24 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 combinando 12 métodos numéricos (ver sección 3.). Se usaron las rutinas de MINPACK-1 [21, 25] implementadas en el paquete `scipy.optimize.root` de Python, así como implementaciones propias.

- c) *Experimento I*: Comparar los filtros wavelets obtenidos por los métodos numéricos para los patrones de complejos K. Primeramente, se obtuvieron soluciones del SENL con 12 métodos numéricos, partiendo del vector nulo. Luego, estas soluciones se usaron como aproximación inicial de 11 métodos numéricos diferentes a los que las generaron. Finalmente, se seleccionaron las ocho mejores combinaciones utilizando un criterio de calidad, para elegir la más adecuada.
- d) *Experimento II*: La combinación seleccionada se usó para evaluar su efectividad al detectar los complejos K frente a otros filtros wavelet². 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 DST-II($f[\cdot]$), siendo $f[\cdot]$ la señal analizada. Se tomó como criterio de detección del complejo K el índice k del coeficiente wavelet de la DST-II 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 [11].

En estos experimentos se utilizó como criterio de convergencia $\|F(q^*)\| \leq 10^{-6} \cdot \sqrt{N}$, donde q^* es el filtro wavelet estimado [4]. Además, 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 [28]: $s_c = -\frac{\log \rho}{\log 4}$, $\rho = \max_{\lambda \neq 1, \frac{1}{2}, \frac{1}{4}, \dots, (\frac{1}{2})^{2c-1}} \{|\lambda(T)|\}$, 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 $H_{ij} = p[i-j]$ donde $p[k]$ 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 [28].

Se utilizó la estadística descriptiva sobre los indicadores numéricos. Las pruebas t para muestras relacionadas [9] 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 [9] 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.

²Haar, 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.

5. RESULTADOS Y DISCUSIÓN

5.1. Experimento I: Obtención de filtros wavelet adaptados a complejos K

Se realizaron 3 696 ejecuciones, de ellas, 529 corridas (14,31%) no terminaron debido a errores numéricos. La convergencia se alcanzó en 651 de las 3 147 corridas (20,69%). Al menos una de las combinaciones de métodos probadas convergió para todos los patrones complejos K.

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 46,93 segundos ($DE = 545,60$), de ellos, se emplearon como promedio 7,15 segundos ($DE = 118,49$) para resolver el sistema. Se obtuvo una regularidad promedio de 1,68 ($DE = 0,67$), alcanzándose un máximo de 3,58. En todos los casos, la condición de detección del SENL se evaluó con valores menores que 10^{-7} .

Dada la cantidad de combinaciones de métodos ($12 \times 11 = 132$), se decidió establecer un *ranking* para elegir las ocho 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 1 se reportan los indicadores numéricos de estas combinaciones ordenados según el criterio explicado.

Tabla 1: Descripción de indicadores numéricos de las ocho mejores combinaciones ordenadas según criterio de *ranking*.

Combinación	Frec. convergencia	Prom. $\ \mathbf{F}(q^*)\ _2$	Prom. Evals.	Tpo. SENL Prom.	Prom. Sim.	Prom. Reg.
df-sane -> continuation	25	$2,04 \cdot 10^{-7}$	194,96	1,95	$1,63 \cdot 10^{-11}$	1,60
df-sane -> newton	23	$1,22 \cdot 10^{-7}$	43,09	4,64	$2,69 \cdot 10^{-11}$	1,52
lm -> newton	23	$1,46 \cdot 10^{-7}$	23	4,08	$2,54 \cdot 10^{-11}$	1,70
lm -> continuation	23	$5,95 \cdot 10^{-7}$	112,43	4,97	$5,30 \cdot 10^{-11}$	1,72
continuation -> newton	23	$6,76 \cdot 10^{-7}$	43,35	2,24	$1,14 \cdot 10^{-9}$	1,45
continuation -> hybr	22	$1,03 \cdot 10^{-8}$	38,32	0,02	$1,99 \cdot 10^{-17}$	1,55
hybr -> continuation	22	$3,24 \cdot 10^{-7}$	126,45	139,33	$5,14 \cdot 10^{-11}$	1,83
hybr -> newton	22	$3,45 \cdot 10^{-7}$	28,91	5,73	$5,68 \cdot 10^{-11}$	1,78

La combinación **df-sane -> continuation** fue la que más veces convergió (25) y disminuyó muy significativamente el número de evaluaciones de funciones de la post-iteración ($M = 138,00$, $EE = 21,11$) respecto a la pre-iteración ($M = 1000,00$, $EE = 0,00$), $t(24) = 38,14$, $p < 0,001$, $r = 0,99$. Se reportó una disminución muy significativa del residual de la post-iteración ($M = 0,00$, $EE = 0,00$) respecto a la pre-iteración ($M = 1,00$, $EE = 0,00$), $t(24) = 1118,6$, $p < 0,001$, $r = 1,00$. Se halló un aumento significativo del tiempo de ejecución para resolver el SENL de la post-iteración ($M = 1,64$, $EE = 0,44$) respecto a la pre-iteración ($M = 0,12$, $EE = 0,01$), $t(24) = -4,13$, $p < 0,001$, $r = 0,64$. La post-iteración aumentó la similitud *shapelet*-patrón de manera muy significativa ($M = 0,00$, $EE = 0,00$ versus $M = -0,03$, $EE = 0,01$), $t(24) = -4,93$, $p < 0,001$, $r = 0,71$. Esto también ocurrió con la regularidad de la *shapelet* obtenida ($M = 1,64$, $EE = 0,11$ versus $M = -0,50$, $EE = 0,00$), $t(24) = -19,80$, $p < 0,001$, $r = 0,97$.

La combinación **df-sane -> newton** convergió 23 veces y disminuyó muy significativamente el número de evaluaciones de funciones de la post-iteración ($M = 23,00$, $EE = 7,84$) respecto a la pre-iteración

($M = 1000$, $EE = 0,00$), $t(22) = 122,06$, $p < 0,001$, $r = 0,99$. Se reportó una disminución muy significativa del residual de la post-iteración ($M = 0,00$, $EE = 0,00$) respecto a la pre-iteración ($M = 1,00$, $EE = 0,00$), $t(22) = 1026,5$, $p < 0,001$, $r = 1,00$. La post-iteración aumentó la similitud *shapelet*-patrón de manera muy significativa ($M = 0,00$, $EE = 0,00$ versus $M = -0,01$, $EE = 0,01$), $t(22) = -3,95$, $p < 0,001$, $r = 0,64$. Esto también ocurrió con la regularidad de la *shapelet* obtenida ($M = 1,62$, $EE = 0,13$ versus $M = -0,50$, $EE = 0,00$), $t(22) = -15,27$, $p < 0,001$, $r = 0,96$.

La combinación `lm -> newton` convergió 23 veces y disminuyó muy significativamente el número de evaluaciones de funciones de la post-iteración ($M = 7,00$, $EE = 6,85$) respecto a la pre-iteración ($M = 3403$, $EE = 361,81$), $t(22) = 6,40$, $p < 0,001$, $r = 0,81$. Se halló un aumento significativo del tiempo de ejecución para resolver el SENL de la post-iteración ($M = 2,54$, $EE = 1,84$) respecto a la pre-iteración ($M = 0,24$, $EE = 0,03$), $t(22) = -2,10$, $p < 0,05$, $r = 0,41$. La post-iteración aumentó la regularidad de la *shapelet* obtenida ($M = 1,63$, $EE = 0,11$ versus $M = 0,00$, $EE = 0,17$), $t(22) = -5,29$, $p < 0,001$, $r = 0,75$.

La combinación `lm -> continuation` convergió 23 veces y disminuyó muy significativamente el número de evaluaciones de funciones de la post-iteración ($M = 102,00$, $EE = 13,53$) respecto a la pre-iteración ($M = 3403,00$, $EE = 368,24$), $t(22) = 6,03$, $p < 0,001$, $r = 0,79$. La post-iteración aumentó la regularidad de la *shapelet* obtenida ($M = 1,64$, $EE = 0,11$ versus $M = 0,00$, $EE = 0,17$), $t(22) = -5,51$, $p < 0,001$, $r = 0,76$.

La combinación `continuation -> newton` convergió 23 veces y disminuyó muy significativamente el número de evaluaciones de funciones de la post-iteración ($M = 11,00$, $EE = 9,60$) respecto a la pre-iteración ($M = 154,00$, $EE = 3,80$), $t(22) = 10,81$, $p < 0,001$, $r = 0,92$. Se halló un aumento significativo del tiempo de ejecución para resolver el SENL de la post-iteración ($M = 1,80$, $EE = 0,60$) respecto a la pre-iteración ($M = 0,41$, $EE = 0,12$), $t(22) = -2,62$, $p < 0,05$, $r = 0,49$. La combinación `continuation -> hybr` convergió 22 veces y disminuyó muy significativamente el número de evaluaciones de funciones de la post-iteración ($M = 23,50$, $EE = 9,76$) respecto a la pre-iteración ($M = 155,00$, $EE = 14,28$), $t(21) = 8,14$, $p < 0,001$, $r = 0,87$. Se halló una disminución muy significativa del tiempo de ejecución para resolver el SENL de la post-iteración ($M = 0,00$, $EE = 0,13$) respecto a la pre-iteración ($M = 0,40$, $EE = 0,13$), $t(22) = 4,58$, $p < 0,05$, $r = 0,71$.

La combinación `hybr -> continuation` convergió 22 veces y aumento significativamente el número de evaluaciones de funciones de la post-iteración ($M = 132,00$, $EE = 15,25$) respecto a la pre-iteración ($M = 57,00$, $EE = 5,15$), $t(21) = -3,39$, $p < 0,001$, $r = 0,59$. Se reportó una disminución muy significativa del residual de la post-iteración ($M = 1,38 \cdot 10^{-10}$, $EE = 2,24 \cdot 10^{-7}$) respecto a la pre-iteración ($M = 9,99 \cdot 10^{-1}$, $EE = 1,04 \cdot 10^{-1}$), $t(21) = 6,06$, $p < 0,001$, $r = 0,80$. La post-iteración aumentó la regularidad de la *shapelet* obtenida ($M = 1,81$, $EE = 0,19$ versus $M = 0,00$, $EE = 0,19$), $t(21) = -5,37$, $p < 0,001$, $r = 0,76$.

Por último, para la combinación `hybr -> newton` se reportó una disminución muy significativa del número de evaluaciones de la post-iteración ($M = 16,50$, $EE = 7,10$) respecto a la pre-iteración ($M = 57,00$, $EE = 8,13$), $t(21) = 3,48$, $p < 0,001$, $r = 0,60$. Se reportó una disminución muy significativa del residual de la post-iteración ($M = 1,51 \cdot 10^{-11}$, $EE = 2,38 \cdot 10^{-7}$) respecto a la pre-iteración ($M = 9,99 \cdot 10^{-1}$, $EE = 1,07 \cdot 10^{-1}$), $t(21) = 5,52$, $p < 0,001$, $r = 0,77$. También, se evidenció un aumento significativo del tiempo total promedio de la post-iteración ($M = 1,78$,

$EE = 2, 01$) respecto a la pre-iteración ($M = 0, 00$, $EE = 0, 00$), $t(61) = -2, 85$, $p < 0, 001$, $r = 0, 53$; y un aumento muy significativo de la regularidad de la *shapelet* de la post-iteración ($M = 1, 81$, $EE = 0, 12$) respecto a la pre-iteración ($M = 0, 00$, $EE = 0, 19$), $t(21) = -5, 18$, $p < 0, 001$, $r = 0, 75$. La Figura 1 muestra las *shapelets* de menor norma del residual para cada una de las combinaciones mencionadas. Puede notarse la suavidad de las wavelets obtenidas, así como la adecuada diferenciación en frecuencias que logran. Las respuestas de fase en general son lineales, aunque no ocurre así con las *shapelets* estimadas para las últimas cuatro combinaciones de métodos.

En resumen, la convergencia se alcanzó para el 20,69% de las corridas realizadas y se pudieron estimar *shapelets* para todos los patrones del complejo K. De las combinaciones que más convergieron se eligió `df-sane -> continuation` 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. Además, alcanzó una disminución muy significativa del residual de la post-iteración, aumentó la similitud *shapelet*-patrón y la regularidad de manera muy significativa. Todo esto hace que esta sea la combinación recomendada para hallar el filtro wavelet adaptado a complejos K, que será evaluado a continuación respecto a la detección de dichos complejos en señales.

5.2. Experimento II: Análisis de forma y efectividad de la detección

En este 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 `df-sane -> continuation`, y se comparó con otros filtros wavelet. Además, se realizó la detección en las mismas señales, sin insertar el patrón.

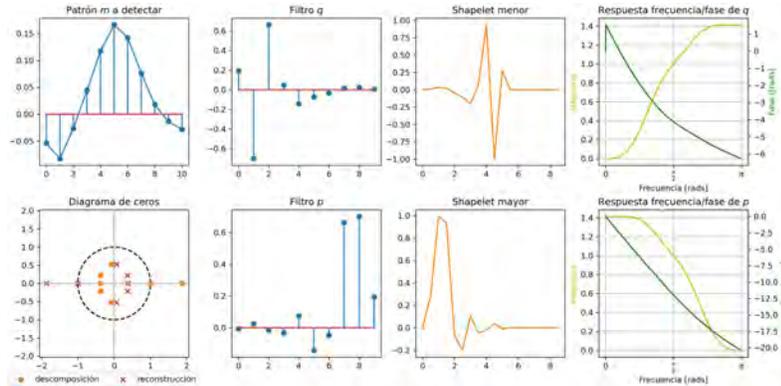
En la Figura 2 se muestra que, al usar las *shapelets* estimadas para cada patrón, se alcanzó una muy alta sensibilidad, 0,99 (i.e., la mayor parte de los patrones insertados en la posición 11 fueron detectados correctamente) y una alta especificidad, 0,87 (i.e., si el patrón no estuvo en la señal, este no es detectado por el algoritmo). Igualmente, se obtuvo un valor predictivo positivo alto, 0,88 (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, 0,99 (i.e., sabiendo que el algoritmo no detectó el patrón, cuál es la probabilidad de que este realmente no esté).

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 3). En dicha figura se destaca en color negro la curva de las *shapelets* con su intervalo de confianza al 95%.

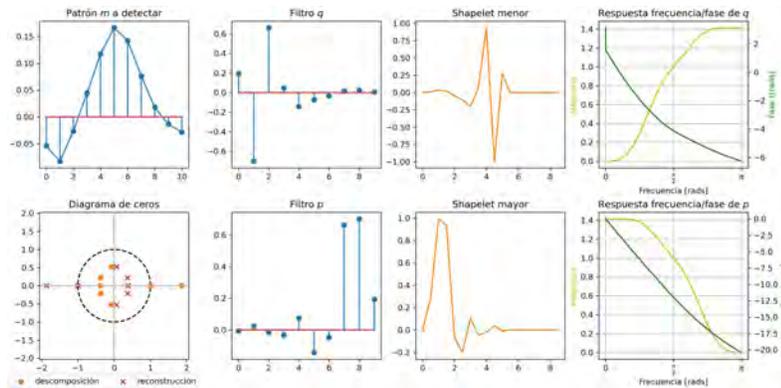
6. CONCLUSIONES

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* adaptada a patrones del complejo K.

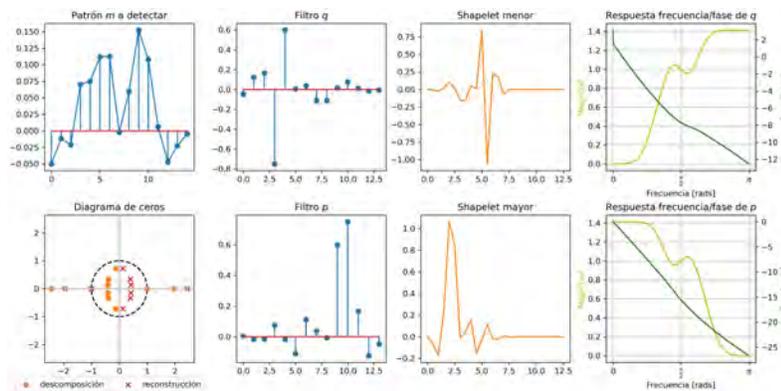
El método de continuación con pre-iteración mediante el algoritmo DF-SANE fue la combinación que convergió más frecuentemente y reportó una disminución significativa del número de evaluaciones de funciones en la post-iteración (de una mediana de 1000 a 138 evaluaciones), una disminución



(a) *Shapelet* de menor valor para $\|\mathbf{F}(q^*)\|_2$ la combinación *df-sane* -> *continuation*. Patrón = Complejo K #1, Dim. = 10, Evals. = 124, $\|\mathbf{F}(q^*)\|_2 = 1,88 \cdot 10^{-16}$, Tpo. SENL = 0,99, Sim. = $6,67 \cdot 10^{-18}$, Reg. = 1,41.

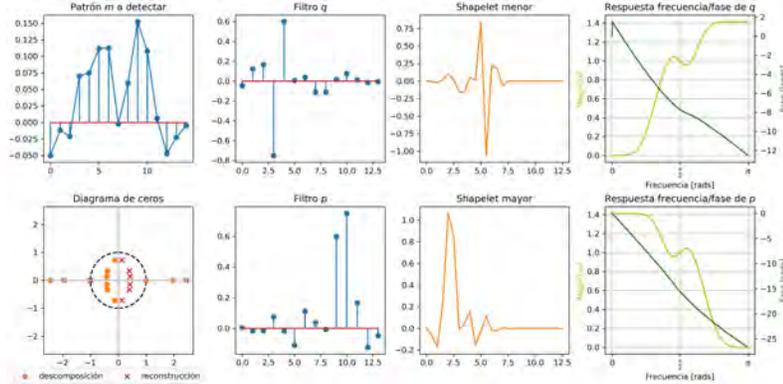


(b) *Shapelet* de menor valor para $\|\mathbf{F}(q^*)\|_2$ la combinación *df-sane* -> *newton*. Patrón = Complejo K #1, Dim. = 10, Evals. = 10, $\|\mathbf{F}(q^*)\|_2 = 5,11 \cdot 10^{-14}$, Tpo. SENL = 0,71, Sim. = $1,25 \cdot 10^{-17}$, Reg. = 1,41.

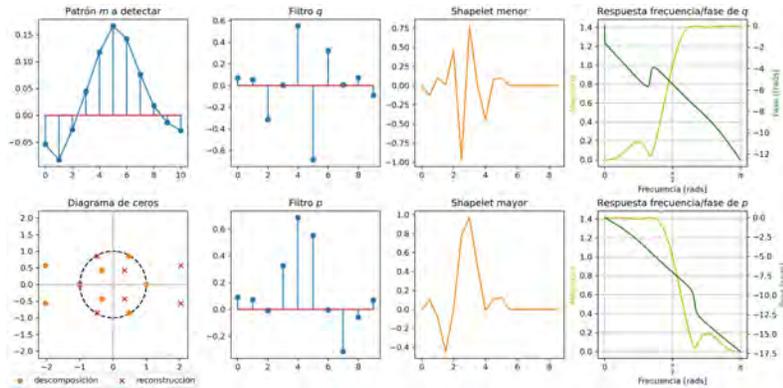


(c) *Shapelet* de menor valor para $\|\mathbf{F}(q^*)\|_2$ la combinación *lm* -> *newton*. Patrón = Complejo K #4, Dim. = 14, Evals. = 25, $\|\mathbf{F}(q^*)\|_2 = 7,29 \cdot 10^{-14}$, Tpo. SENL = 44,11, Sim. = $1,30 \cdot 10^{-17}$, Reg. = 1,56.

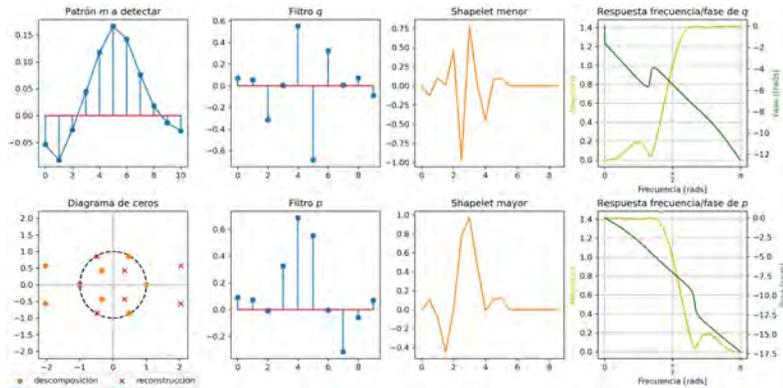
Figura 1: Resultados de menor valor del residual $\|\mathbf{F}(q^*)\|_2$ para las combinaciones analizadas.



(d) *Shapelet* de menor valor para $\|\mathbf{F}(q^*)\|_2$ la combinación `lm -> continuation`. Patrón = Complejo K #4, Dim. = 14, Evals. = 152, $\|\mathbf{F}(q^*)\|_2 = 3,45 \cdot 10^{-13}$, Tpo. SENL = 0,35, Sim. = $1,08 \cdot 10^{-18}$, Reg. = 1,56.

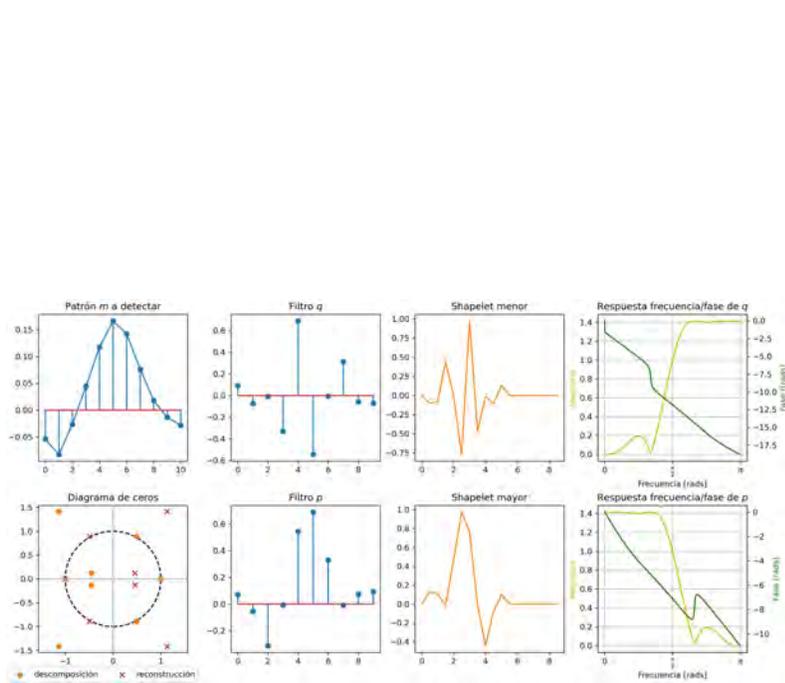


(e) *Shapelet* de menor valor para $\|\mathbf{F}(q^*)\|_2$ la combinación `continuation -> newton`. Patrón = Complejo K #1, Dim. = 10, Evals. = 3, $\|\mathbf{F}(q^*)\|_2 = 9,76 \cdot 10^{-16}$, Tpo. SENL = 0,71, Sim. = $6,50 \cdot 10^{-18}$, Reg. = 1,38.

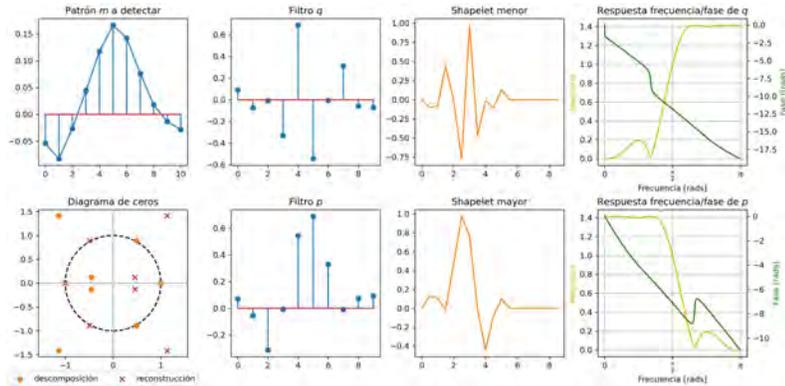


(f) *Shapelet* de menor valor para $\|\mathbf{F}(q^*)\|_2$ la combinación `continuation -> hybr.` Patrón = Complejo K #1, Dim. = 10, Evals. = 15, $\|\mathbf{F}(q^*)\|_2 = 2,51 \cdot 10^{-16}$, Tpo. SENL = 0,0009, Sim. = $2,60 \cdot 10^{-17}$, Reg. = 1,38.

Figura 1: Resultados de menor valor del residual $\|\mathbf{F}(q^*)\|_2$ para las combinaciones analizadas.



(g) *Shapelet* de menor valor para $\|\mathbf{F}(q^*)\|_2$ la combinación **hybr** -> **continuation**. Patrón = Complejo K #4, Dim. = 14, Evals. = 62, $\|\mathbf{F}(q^*)\|_2 = 2,09 \cdot 10^{-15}$, Tpo. SENL = 0,71, Sim. = $2,69 \cdot 10^{-17}$, Reg. = 1,57.



(h) *Shapelet* de menor valor para $\|\mathbf{F}(q^*)\|_2$ la combinación **hybr** -> **newton**. Patrón = Complejo K #1, Dim. = 10, Evals. = 3, $\|\mathbf{F}(q^*)\|_2 = 1,79 \cdot 10^{-15}$, Tpo. SENL = 0,75, Sim. = $1,7 \cdot 10^{-18}$, Reg. = 1,56.

Figura 1: Resultados de menor valor del residual $\|\mathbf{F}(q^*)\|_2$ para las combinaciones analizadas.

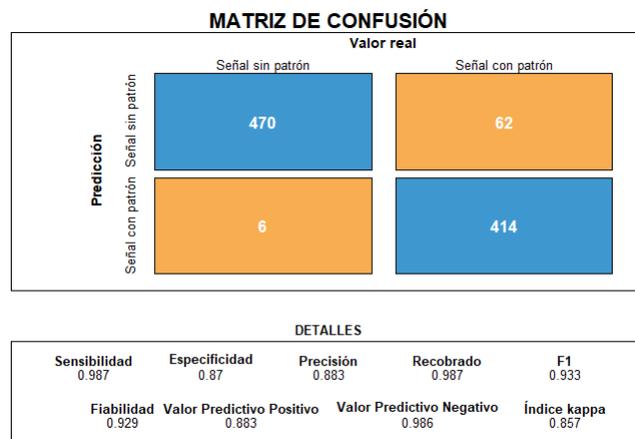


Figura 2: Matriz de confusión sobre la efectividad de la detección de los patrones usando las *shapelets* correspondientes.

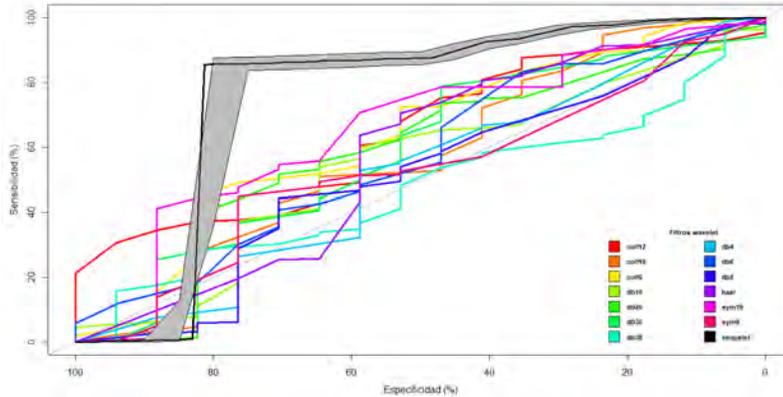


Figura 3: Curva ROC de la detección de los patrones usando las *shapelets* y 13 filtros wavelet.

muy significativa del residual de la post-iteración (de una mediana de 1,00 a 0,00), un aumento muy significativo de la similitud *shapelet*-patrón y de la regularidad de la *shapelet* obtenida (de una mediana de $-0,50$ a $1,64$). Además, los filtros estimados con esta combinación alcanzaron valores promedio muy bajos del residual (del orden de 10^{-7}), alta similitud con el patrón (del orden de 10^{-11}) y una regularidad promedio de la *shapelet* de 1,60.

Al utilizar la DST-II para la detección de complejos K en diferentes señales usando los filtros estimados por esta vía, se alcanzaron valores de sensibilidad = 0,99, especificidad = 0,85, valor predictivo positivo = 0,87, valor predictivo negativo = 0,99 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 continuación con pre-iteración mediante el algoritmo de DF-SANE para estimar los filtros en una DST-II.

Para continuar la investigación se recomienda realizar la detección en señales más largas de EEG y comprobar si esta puede funcionar en tiempo real para la detección de los complejos K en estudios de sueño.

AGRADECIMIENTOS

Los autores agradecen el apoyo del Proyecto “Métodos numéricos para problemas en múltiples escalas”, perteneciente al Programa Nacional de Ciencias Básicas, PN223LH010-003, Ministerio de Ciencia, Tecnología y Medio Ambiente (CITMA), Cuba. Igualmente, agradecemos la asesoría dada por el Lic. Jose Antelo, especialista de estudios del sueño mediante EEG, del Centro de Neurociencias de Cuba (CNEURO).

RECEIVED: DECEMBER, 2022.

REVISED: JUNE, 2023.

REFERENCIAS

- [1] Aldroubi, A., Abry, P., and Unser, M. (1998): Construction of biorthogonal wavelets starting from any two multiresolutions **IEEE Transactions on Signal Processing**, 46(4):1130–1133.
- [2] Anderson, D. G. (1965): Iterative Procedures for Nonlinear Integral Equations **Journal of the ACM**, 12(4):547–560.
- [3] Berry, R. B., Brooks, R., Gamaldo, C. E., Harding, S. M., Marcus, C. L., and Vaughn, B. V. (2018): The AASM Manual for the Scoring of Sleep and Associated Events Version 2.5 Technical report, American Academy of Sleep Medicine, Darien, IL.
- [4] Birgin, E. G., Gardenghi, J. L., Marcondes, D. S., and Martínez, J. M. (2021): Accelerated derivative-free spectral residual method for nonlinear systems of equations **arxiv**.
- [5] Devuyst, S. (2005): The DREAMS© Databases and Assessment Algorithm.
- [6] Devuyst, S., Dutoit, T., Stenuit, P., and Kerkhofs, M. (2010): Automatic K-complexes detection in sleep EEG recordings using likelihood thresholds In **2010 Annual International Conference of the IEEE Engineering in Medicine and Biology**, pages 4658–4661. IEEE.
- [7] Dumitrescu, C., Costea, I. M., Cormos, A. C., and Semenescu, A. (2021): Automatic detection of k-complexes using the cohen class recursiveness and reallocation method and deep neural networks with eeg signals **Sensors**, 21(21).
- [8] Erdamar, A., Duman, F., and Yetkin, S. (2012): A wavelet and teager energy operator based method for automatic detection of K-Complex in sleep EEG **Expert Systems with Applications**, 39(1):1284–1290.
- [9] Field, A., Miles, J., and Field, Z. (2012): **Discovering statistics using R** Sage Publications Ltd.
- [10] Guido, R. C. (2011): A note on a practical relationship between filter coefficients and scaling and wavelet functions of discrete wavelet transforms **Applied Mathematics Letters**, 24(7):1257–1259.
- [11] Guido, R. C. (2018): Fusing time, frequency and shape-related information: Introduction to the Discrete Shapelet Transform’s second generation (DST-II) **Information Fusion**, 41:9–15.
- [12] Jawali, D., Kumar, A., and Seelamantula, C. S. (2019): A Learning Approach for Wavelet Design In **ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)**, pages 5018–5022. IEEE.
- [13] Khasawneh, N., Fraiwan, M., and Fraiwan, L. (2022): Detection of K-complexes in EEG signals using deep transfer learning and YOLOv3 **Cluster Computing**, 22(297):1–14.
- [14] Knoll, D. A. and Keyes, D. E. (2004): Jacobian-free Newton-Krylov methods: A survey of approaches and applications **Journal of Computational Physics**, 193:357–397.

- [15] Krohne, L. K., Hansen, R. B., Christensen, J. A. E., Sorensen, H. B. D., and Jennum, P. (2014): Detection of K-complexes based on the wavelet transform In **2014 36th Annual International Conference of the IEEE Engineering in Medicine and Biology Society**, pages 5450–5453. IEEE.
- [16] Kryger, M., Roth, T., and Dement, W. C., editors (2017): **Principles and Practice of Sleep Medicine** Elsevier, 6th edition.
- [17] La Cruz, W., Martínez, J. M., and Raydan, M. (2006): Spectral residual method without gradient information for solving large-scale nonlinear systems of equations **Mathematics of Computation**, 75:1429–1449.
- [18] Mallat, S. (2009): **A Wavelet Tour of Signal Processing: The Sparse Way** Academic Press, Burlington, MA, 3rd edition.
- [19] Mesa, H. (2005): Adapted wavelets for pattern detection In Sanfeliu, A. and Cortés, M. L., editors, **Progress in Pattern Recognition, Image Analysis and Applications**, pages 933–944, Berlin, Heidelberg. Springer.
- [20] Misiti, M., Yves, M., Oppenheim, G., and Poggi, J.-M. (2007): **Wavelets and their Applications** ISTE Ltd, London.
- [21] Moré, J. J., Garbow, B. S., and Hillstom, K. E. (1980): User guide for minpack-1 Technical report, Argonne National Laboratory, Argonne, Illinois.
- [22] Oppenheim, A. V. and Schafer, R. W. (2011): **Tratamiento de señales en tiempo discreto** Pearson Educación, S.A., Ribera del Loira, Madrid, 3era edition.
- [23] Patti, C. R., Abdullah, H., Shoji, Y., Hayley, A., Schilling, C., Schredl, M., and Cvetkovic, D. (2016): K-complex detection based on pattern matched wavelets In **IECBES 2016 - IEEE-EMBS Conference on Biomedical Engineering and Sciences**, pages 470–474.
- [24] Powell, M. J. D. (1970): A Hybrid Method for Nonlinear Equations In Rabinowitz, P., editor, **Numerical Methods for Nonlinear Algebraic Equations**, pages 87–114. Gordon and Breach.
- [25] Press, W. H., Teukolsky, S. A., Vetterling, W. T., and Flannery, B. P. (2007): **Numerical Recipes: The Art of Scientific Computing** Cambridge University Press, Cambridge, 3rd edition.
- [26] R. L. Burden, D. J. Faires, and A. M. Burden (2017): Soluciones numéricas de sistemas de ecuaciones no lineales In **Análisis numérico**, chapter 10, pages 475–504. Cengage Learning, 10a. edition.
- [27] Strang, G. (2019): **Linear Algebra and Learning from Data** Wesley-Cambridge Press, Wellesley MA.
- [28] Strang, G. and Nguyen, T. (1997): **Wavelets and filter banks** Wellesley-Cambridge, Wellesley, Massachusetts, revised 2nd edition.