

UNIVERSIDAD AUTÓNOMA DE MADRID

ESCUELA POLITÉCNICA SUPERIOR



**Grado en Ingeniería de Tecnologías y Servicios de
Telecomunicación**

TRABAJO FIN DE GRADO

**EXTRACCIÓN DE CARACTERÍSTICAS PARA SEÑALES
TEMPORALES PROCEDENTES DE SENSORES
INDUSTRIALES**

**Autor: Iglesias Arias, Álvaro
Tutor: Ramos Castro, Daniel**

JUNIO 2017

Extracción de Características para Señales Temporales Procedentes de Sensores Industriales

AUTOR: Álvaro Iglesias Arias
TUTOR: Daniel Ramos Castro

Grupo Audias-ATVS
Dpto. Tecnología Electrónica y de las Comunicaciones
Escuela Politécnica Superior
Universidad Autónoma de Madrid
Junio de 2017

Resumen

En este Trabajo de Fin de Grado se ha realizado un acercamiento al proceso de detección de eventos en señales procedentes de sensores industriales. El análisis de piezas en el ámbito industrial es de vital importancia como forma de asegurar unos estándares de calidad mínimos. Es por ello que la detección de cualquier irregularidad en estas señales es de vital importancia, ahorrando posibles problemas en el futuro.

Por ello, en este trabajo se ha propuesto, mediante el uso de técnicas de procesado de señal y reconocimiento de patrones, la implementación de un detector automático de eventos, enfocado a su uso sobre las señales proporcionadas por una empresa externa, para la posterior clasificación de estos.

Como se irá explicando a lo largo de esta memoria para abordar este problema se ha hecho uso de dos posibles esquemas; un detector por correlación y un detector basado en características espectrales ambos implementados mediante un sistema del tipo *Fron-tend-Back-end*.

Si bien el proceso de clasificación posterior a la detección es de también de vital importancia en un problema como el planteado, en este trabajo nos centraremos en el proceso de selección y extracción de las características correctas para una adecuada separación entre eventos, así como en la detección de estos, que corresponde al *Front-end* de nuestro clasificador.

Resaltar que este trabajo ha sido realizado desde cero en Matlab con el objetivo de crear una base adecuada y funcional desde la que se puede continuar y perfeccionar, ya sea aumentando la base de datos disponible o mediante el uso de detectores y clasificadores más complejos.

Palabras clave

Detector, señales industriales, sensor, correlación, *front-end*, *back-end*

Abstract

In this Bachelor Thesis, an approach to signal processing and automatic event recognition has been performed in an industrial and mechanical environment. The industrial parts analysis is very important as a form of detection and supervision, that is, to ensure a minimum quality standard. Therefore, the detection of any irregularity in these signals is of vital importance, saving from possible future damages.

In this work, it has been proposed the implementation of an automatic event detector focused on signals provided by an external enterprise. All this has been possible by using pattern recognition techniques.

All along this paper will be explained there has been used two different schemes, a correlation detector or a more complex system based on a Frontend-Backend scheme.

Throughout this article we explain that we have used two different schemes, a correlation detector or a more complex spectral detector both systems based on a Front-end-Back-end scheme.

In this essay, we will focus on the process of selection and extraction of the correct characteristics for an adequate separation between events, as well as in the Detection of these, which corresponds to the Front-end of our classifier.

Finally, it is also important to note that this work has been implemented in Matlab from the ground with the objective of create a worthy and functional baseline from which to continue and improve this work goal.

Keywords

Detector, industrial signals, sensor, correlation, front-end, back-end

Agradecimientos

Llegamos al, que puede que sea, el punto más difícil de escribir de esta memoria. Mucha gente, muchos momentos y muchos años por medio llegamos al final de una etapa marcada por el curro, las prisas, las risas, los exámenes, las cervezas, y las noches en vela. Sigue quedando mucho por recorrer, pero la primera parte del camino acaba en este papel. Por ello se hace difícil tener presente a toda la gente que merece aparecer aquí nombrada y que me ha acompañado este tiempo, aun así, les intentare hacer justicia todo lo posible.

Así que empecemos, antes de nada, por agradecer todo el apoyo dado por mi familia, que me ha aguantado en los buenos momentos y los malos. A Belén, mi hermana, que cada día muerde un poco menos y a la que pocas cosas hay en este mundo que le importen menos que este trabajo.

También, claro está, a mi tutor Daniel que me ha guiado durante todos estos meses y que ha conseguido que este trabajo no descarrilase. Mencionar a Adrián que ha sido casi mi segundo tutor y que siempre ha estado ahí para resolver mis infinitas dudas y aguantar mis gilipolleces, gracias, de verdad.

Acordarme también de los grandes amigos que he hecho aquí y que me acompañaran toda la vida. A Lamas y a Belt que estuvieron en las buenas y me sostuvieron en las malas, y con los que, en los últimos años, he pasado más tiempo que con mi propia familia, sin vosotros habría sido todo mucho más difícil. A Alex, con quien se puede contar para cualquier cosa y que, con su forma de ser, consiguió hacerme superar la vergüenza ajena. Hay mucha más gente a la que tengo que dar las gracias: a Javi, Julio, Rober, Fernández, Basar, Pabli, ... pero de momento dejémoslo aquí, ellos saben quiénes son. Antes de terminar no me quiero olvidar de Mercedes, que me ha acompañado durante todo el TFG y cuyo trabajo he acabado conociendo casi tan bien como el mío.

Mención especial a los desemados a los que, durante este tiempo, he visto dos veces al mes con suerte, en el caso de Fer una vez al año, y aun así siempre que estoy con ellos da la sensación de que nos vemos todos los días. Pocas cosas hay como las noches de verano en el río con vosotros.

Y gracias a ti, que vas a leer esta memoria. Espero que sea leve.

ÍNDICE DE CONTENIDOS

1 INTRODUCCIÓN.....	1
1.1 MOTIVACIÓN	1
1.2 OBJETIVOS.....	1
1.3 ORGANIZACIÓN DE LA MEMORIA	2
2 ESTADO DEL ARTE	3
2.1 PROCESADO DE SEÑAL.....	3
2.1.1 <i>Filtrado paso bajo</i>	3
2.1.2 <i>Extracción de características</i>	4
2.2 RECONOCIMIENTO DE PATRONES, TIPOS DE DETECTORES.....	6
2.2.1 <i>Correlación cruzada</i>	6
2.2.2 <i>Distancia Euclídea</i>	6
2.2.3 <i>Distancia de Mahalanobis</i>	7
2.2.4 <i>Backend</i>	8
2.3 MEDIDAS DE RENDIMIENTO EN DETECTORES Y CLASIFICADORES	8
2.3.1 <i>Falso rechazo y falsa aceptación</i>	8
2.3.1 <i>Curvas DET</i>	10
2.3.2 <i>Otras medidas de rendimiento: Precisión y recall</i>	10
3 DISEÑO	11
3.1 BASE DE DATOS, SEÑALES DISPONIBLES	11
3.1.1 <i>Características de las señales</i>	11
3.1.2 <i>Tipos de eventos</i>	12
3.1.3 <i>Protocolo Experimental</i>	14
3.2 ARQUITECTURA DEL SISTEMA	14
3.2.1 <i>Detector por Correlación</i>	16
3.2.2 <i>Detector Espectral</i>	16
4 DESARROLLO.....	17
4.1 EXTRACCIÓN DE CARACTERÍSTICAS	17
4.1.1 <i>Pre-procesado y características de las Señales</i>	17
4.1.2 <i>Características detector espectral, análisis de varianzas</i>	19
4.2 DETECTOR POR CORRELACIÓN.....	21
4.3 DETECTOR ESPECTRAL	23
4.3.1 <i>Enventanado</i>	23
4.3.2 <i>Etiquetado</i>	24
4.3.3 <i>Extracción de características</i>	24
4.3.4 <i>Entrenamiento del detector</i>	24
4.4 FUSIÓN DE DETECTORES	25
5 INTEGRACIÓN, PRUEBAS Y RESULTADOS	27
5.1 DETECTOR POR CORRELACIÓN.....	27
5.1.1 <i>Resultados sobre base de desarrollo</i>	27
5.1.2 <i>Resultado sobre base de generalización</i>	28
5.1 DETECTOR ESPECTRAL	29
5.1.1 <i>Resultados sobre base de desarrollo</i>	29
5.1.2 <i>Resultado sobre base de generalización</i>	33
5.1.3 <i>Comparación con detector por correlación</i>	35
5.2 FUSIÓN DE DETECTORES	35
5.2.1 <i>Resultados sobre base de desarrollo</i>	36
5.2.2 <i>Resultados sobre base de generalización</i>	36
6 CONCLUSIONES Y TRABAJO FUTURO.....	37
6.1 CONCLUSIONES.....	37

6.2 TRABAJO FUTURO	37
REFERENCIAS	39
ANEXO A	- 1 -
<i>Etiquetas Ground Truth</i>	- 1 -
<i>Puntuaciones fusión de detectores</i>	- 3 -

ÍNDICE DE FIGURAS

FIGURA 1: ESQUEMA CÁLCULO DE COEFICIENTES DELTA.....	5
FIGURA 2: COMPARACIÓN ENTRE DISTANCIA EUCLÍDEA Y DE MAHALANOBIS MEDIANTE LA REPRESENTACIÓN DE ZONAS EQUIDISTANTES A LA MEDIA DE UN CONJUNTO DE PUNTOS SITUADOS EN UN ESPACIO BIDIMENSIONAL.....	7
FIGURA 3: HISTOGRAMA DE PUNTUACIONES OBTENIDAS PARA CLASE TARGET Y PARA NON TARGET	9
FIGURA 4: CURVA DET OBTENIDA SOBRE PUNTUACIONES DE DISTRIBUCIÓN GAUSSIANA	9
FIGURA 5: SEÑAL DE LA BASE DE DATOS DE DESARROLLO Y ZONA ACTIVA DE ESTA CON SUS EVENTOS ETIQUETADOS	13
FIGURA 6: ESQUEMA BASE CLASIFICADOR FRONTEND-BACKEND.....	15
FIGURA 7: ESQUEMA BÁSICO FRONTEND.	15
FIGURA 8: ESQUEMA BÁSICO FUNCIONAMIENTO DE UN DETECTOR POR CORRELACIÓN.....	16
FIGURA 9: ZONA ACTIVA DE LA SEÑAL Y DERIVADA DE ESTA.	18
FIGURA 10: RESULTADOS ANÁLISIS DE VARIANZAS PARA LOS PRIMEROS 10 COEFICIENTES DE LA FFT	19
FIGURA 11: REPRESENTACIÓN DE LOS DIFERENTES EVENTOS SEGÚN ALGUNAS DE SUS CARACTERÍSTICAS EXTRAÍDAS	20
FIGURA 12: PROCESO DE CREACIÓN DE PLANTILLAS	22
FIGURA 13: SEÑAL Y PUNTUACIÓN OBTENIDA MEDIANTE LA CORRELACIÓN DE LA SEÑAL CON UNA DE LAS PLANTILLAS	22
FIGURA 14: EJEMPLO DE REGIONES DEFINIDAS PARA CADA EVENTO MEDIANTE DISTANCIA EUCLÍDEA Y DISTANCIA DE MAHALANOBIS EN UN ESPACIO BIDIMENSIONAL	25
FIGURA 15: REPRESENTACIÓN DE LOS DIFERENTES EVENTOS SEGÚN LA CORRELACIÓN OBTENIDA ENTRE CADA VENTANA Y LAS DIFERENTES PLANTILLAS.....	26
FIGURA 16: CURVAS DET PARA CADA EVENTO OBTENIDAS SOBRE LA BASE DE DATOS DE DESARROLLO	27
FIGURA 17: SCORES OBTENIDOS POR EL DETECTOR DE CADA EVENTO SOBRE UNA DE LAS SEÑALES DE LA BASE DE DATOS DE DESARROLLO	28
FIGURA 18: CURVAS DET PARA CADA EVENTO OBTENIDAS SOBRE LA BASE DE DATOS DE GENERALIZACIÓN.....	29
FIGURA 19: SCORES OBTENIDOS POR EL DETECTOR POR CORRELACIÓN DE CADA EVENTO SOBRE UNA DE LAS SEÑALES DE LA BASE DE DATOS DE GENERALIZACIÓN	30

FIGURA 20: CURVAS DET PARA CADA EVENTO (BASE DE DATOS DE DESARROLLO) CON DETECTOR ESPECTRAL Y EUCLÍDEA SIN COEFICIENTES DELTA (IZQUIERDA) Y CON COEFICIENTES DELTA (DERECHA).....	31
FIGURA 21: CURVAS DET PARA CADA EVENTO (BASE DE DATOS DE DESARROLLO) CON DETECTOR ESPECTRAL Y MAHALANOBIS SIN COEFICIENTES DELTA (IZQUIERDA) Y CON COEFICIENTES DELTA (DERECHA).....	31
FIGURA 22: SCORES OBTENIDOS PARA SEÑAL DE BASE DE DATOS DE DESARROLLO POR LOS DETECTORES ESPECTRALES DE CADA EVENTO TANTO CON DISTANCIA EUCLÍDEA (IZQUIERDA) COMO DE MAHALANOBIS (DERECHA).....	32
FIGURA 23: CLASIFICACIÓN REALIZADA POR EL BACKEND SOBRE UNA DE LAS SEÑALES DE LA BASE DE DESARROLLO PARA EL DETECTOR ESPECTRAL CON DISTANCIA EUCLÍDEA SIN COEFICIENTES DELTA. FSCORE = 0.87, ACCURACY = 0.97, PRECISION= 0.83.....	33
FIGURA 24: CLASIFICACIÓN REALIZADA POR EL BACKEND SOBRE UNA DE LAS SEÑALES DE LA BASE DE DESARROLLO PARA EL DETECTOR ESPECTRAL CON DISTANCIA MAHALANOBIS Y COEFICIENTES DELTA. FSCORE = 0.91, ACCURACY = 0.98, PRECISION = 0.94.....	33
FIGURA 25: DET OBTENIDAS SOBRE BASE DE DATOS DE GENERALIZACIÓN. A LA IZQUIERDA POR EL DETECTOR ESPECTRAL CON DISTANCIA EUCLÍDEA, A LA DERECHA CON MAHALANOBIS.....	34
FIGURA 26: CLASIFICACIÓN REALIZADA POR EL BACKEND SOBRE SEÑAL DE LA BASE DE DATOS DE GENERALIZACIÓN. A LA IZQUIERDA CON DETECTOR ESPECTRAL DE DISTANCIA EUCLÍDEA, A LA DERECHA CON MAHALANOBIS.....	34
FIGURA 27: IZQUIERDA PUNTUACIONES DETECTOR ESPECTRAL CON DISTANCIA EUCLÍDEA SOBRE SEÑAL DE LA BASE DE DATOS DE GENERALIZACIÓN. A LA DERECHA PUNTUACIONES CON DISTANCIA MAHALANOBIS.....	35
FIGURA 28: IZQUIERDA DET FUSIÓN EUCLÍDEA, DERECHA DET FUSIÓN MAHALANOBIS CALCULADAS SOBRE BASE DE DATOS DE GENERALIZACIÓN.....	36
FIGURA 29: IZQUIERDA DET FUSIÓN EUCLÍDEA, DERECHA DET FUSIÓN MAHALANOBIS CALCULADAS SOBRE BASE DE DATOS DE DESARROLLO.....	36
FIGURA 30: ETIQUETAS GROUND TRUTH SEÑAL DE LA BASE DE DATOS DE GENERALIZACIÓN.....	- 1 -
FIGURA 31: ETIQUETAS GROUND TRUTH SEÑAL DE LA BASE DE DATOS DE GENERALIZACIÓN.....	- 1 -
FIGURA 32: ETIQUETAS GROUND TRUTH SEÑAL DE LA BASE DE DATOS DE DESARROLLO.....	- 2 -
FIGURA 33: SCORES OBTENIDOS POR DETECTOR FUSIÓN MEDIANTE DISTANCIA EUCLÍDEA (IZQUIERDA) Y DISTANCIA DE MAHALANOBIS (DERECHA) SOBRE SEÑAL DE LA BASE DE DATOS DE DESARROLLO.....	- 3 -
FIGURA 34: SCORES OBTENIDOS POR EL DETECTOR FUSIÓN MEDIANTE DISTANCIA EUCLÍDEA (IZQUIERDA) Y DISTANCIA DE MAHALANOBIS (DERECHA) SOBRE SEÑAL DE LA BASE DE DATOS DE GENERALIZACIÓN.....	- 4 -

1 Introducción

En este apartado se busca introducir al lector al escenario de aplicación planteado, así como a las distintas soluciones que se han propuesto y sus diferentes limitaciones. Es por ello que, antes de nada, se explicara el problema que se busca atajar con en este trabajo.

1.1 Motivación

El análisis y detección de posibles errores en materiales o piezas industriales, es una parte de vital importancia en el proceso de control de calidad de cualquier empresa y que requiere generalmente de la actuación de personal especializado que analice y examine éstas con el objetivo de encontrar cualquier tipo de deficiencia. Es por ello que en este TFG se ha buscado, mediante el uso de técnicas de procesado de señal y reconocimiento de patrones, aportar una solución mediante la detección automática de estos defectos o eventos.

Para ello, disponemos de una serie de señales obtenidas tras el paso de las piezas por distintos sensores que realizan diversos tipos de mediciones sobre ellas. Estas piezas pueden sufrir distintos defectos que se reflejan en la señal obtenida y que se dividen en distintos tipos de imperfección dependiendo de las características de esta. Es por ello que en este trabajo no solo buscaremos la detección de posibles irregularidades, sino que además intentaremos distinguir el tipo de evento concreto, cualidad muy deseable a la hora de resolver este problema puesto que no todos los posibles eventos tienen la misma relevancia ni afectan de la misma forma a la pieza.

Como base de datos en este trabajo se dispone de una serie de señales obtenidas sobre piezas con un gran número de defectos, destinadas a la realización de pruebas de detección. Aun así, el número de señales que componen esta base es relativamente pequeño, por lo que se han implementado métodos robustos a esto, y se ha tenido en cuenta de forma especial la generalización del sistema a datos no vistos.

1.2 Objetivos

Este trabajo se ha centrado en la implementación de un sistema de detección automática de eventos sobre señales procedentes de sensores industriales, que poseen unas características muy marcadas. Podríamos concretar los objetivos como:

1. Elección y extracción de características de las señales que faciliten y hagan robusta la detección de los diferentes eventos.
2. Implementación y ajuste de diversos detectores con la finalidad de discriminar de la mejor forma entre los diferentes eventos.
3. Análisis de los diferentes tipos de detectores implementados, así como de sus ventajas e inconvenientes. Observar su rendimiento sobre un sistema de salida básico.

4. Observar las limitaciones de los distintos modelos usados y analizar cuáles son las posibles mejoras a implementar en el futuro.
5. Gestión y uso de las señales de la base de datos que se dispone, que es limitada, de forma que se evite el *overfitting* en el detector y que este se comporte adecuadamente sobre otras señales.

1.3 Organización de la memoria

La memoria consta de los siguientes capítulos:

- **Capítulo 1: Introducción.** La presentación del problema, los objetivos a abordar y la organización de la memoria se encuentran en este apartado.
- **Capítulo 2: Estado del arte.** En este apartado se explica brevemente la teoría detrás de diversas técnicas utilizadas tanto el procesado de señal como en el ámbito del reconocimiento de patrones y que han servido de base en el desarrollo de este trabajo.
- **Capítulo 3: Diseño.** Se presentan los sistemas utilizados en este trabajo desde un punto de vista de estructura y función. También se introducen las bases de datos utilizada y sus particularidades.
- **Capítulo 4: Desarrollo.** Se profundizará en el funcionamiento de los diferentes sistemas, así como en los diferentes problemas surgidos durante su desarrollo.
- **Capítulo 5: Integración, pruebas y resultados.** Los distintos resultados obtenidos por estos sistemas se encuentran en este apartado, así como algunas de las conclusiones fruto del análisis tanto de cada uno de los sistemas como de la comparación entre ellos.
- **Capítulo 6: Conclusiones y trabajo futuro.** En este apartado se hace un análisis general de los diferentes objetivos alcanzados mediante este proyecto, así como de los que se pueden lograr en un futuro a partir de él.

2 Estado del arte

En este capítulo se presentan distintas técnicas utilizadas tanto en el ámbito del procesado de señal como del aprendizaje máquina y que serán especialmente útiles a la hora de entender y profundizar en las técnicas empleadas en este trabajo.

2.1 Procesado de señal

Este trabajo, como se ha comentado, si bien tiene como objetivo la detección de eventos en diferentes piezas, tiene un gran componente de tratamiento y procesado de señal. Por ello se explicarán diversas técnicas relacionadas y que se han usado en distintos momentos.

2.1.1 Filtrado paso bajo

Las señales usadas en este trabajo, como la mayoría de señales presentes en otros ámbitos, presentan un ruido característico que puede dificultar las labores de detección. Es por ello que se ha propuesto el uso de filtros paso bajo básicos como forma de reducir en todo lo posible el efecto de este. De entre toda la variedad de filtros posibles a usar destacaremos los siguientes:

- Filtro Butterworth: Presenta la respuesta en frecuencia más plana de todos los filtros posibles, además de un retardo de grupo casi constante. A cambio requiere de un mayor orden que el resto de filtros para obtener las mismas especificaciones que estos.
- Filtro Chebyshev I o directo: Presenta rizado en la banda de paso pudiendo producir efectos no deseados sobre la señal.
- Filtro Chebyshev II o inverso: Presenta una respuesta en la banda de paso plana y un retardo de grupo casi constante, evitando de esta forma la introducción de distorsiones de fase.
- Filtros elípticos: Presentan anchos de banda de transición muy estrechos, resultando muy útiles a la hora de implementar filtros muy restrictivos. Si bien sufren de un retardo de grupo poco constante, sobre todo para ordenes grandes del filtro. Además, presenta rizado tanto en la banda de paso como en la de corte.

Tras observar las diferentes características de cada filtro se decidió que un filtro Chebyshev de tipo II o inverso era el adecuado para utilizarse en nuestro problema. Cuya función de transferencia queda definida por:

$$|H(j\bar{\omega})|^2 = \frac{\varepsilon^2 C_n^2(1/\bar{\omega})}{1 + \varepsilon^2 C_n^2(1/\bar{\omega})}$$

Donde ε es el factor de rizado definido como:

$$\varepsilon = \left(10^{\frac{\alpha_{min}}{10}} - 1\right)^{-1/2}$$

Tal que α_{min} es la atenuación mínima de la banda del corte en dB y $C_n(\bar{\omega})$ el polinomio de Chebyshev normalizado de orden n . (1)

2.1.2 Extracción de características

A continuación, se comentará brevemente las diferentes características que tradicionalmente se han utilizado en procesamiento de habla para caracterizar las señales y que, con variaciones, se han usado en este trabajo.

Características espectrales, MFCC

Las señales de voz se caracterizan fundamentalmente por dos características fundamentales: la componente armónica o pitch, que indica el género del hablante, y los máximos de la envolvente o formantes, que indican el tipo de sonido articulado. Ambas características se distinguen con facilidad en el dominio frecuencial, si bien a la hora de procesar estas señales ambas partes son difíciles de separar. Como forma de realizar esta separación se usa el dominio pseudo temporal conocido como dominio cepstral, donde ambas características aparecen separadas. De entre todos los conjuntos de características utilizados en procesamiento de habla no podemos dejar de comentar los MFCC o *Mel-Frequency Cepstrum Coefficients* que, junto a los LPCC, son los más usados en este ámbito. (2)

Estos coeficientes destacan por concentrar en un vector de pocas dimensiones la información presente en la señal haciendo uso tanto de la percepción auditiva del ser humano, mediante el uso de la escala de Mel, como del dominio Cepstral. Para ello se aplica sobre el espectro de la señal un banco de filtros triangulares cuyas frecuencias centrales coinciden con las de las bandas críticas de Bark y se obtiene la energía en cada uno, lo que nos dará un único coeficiente por banda (o filtro). Matemáticamente esto se define como:

$$Y(i) = \sum_{k \in B_i} \log |S(k)| \cdot H_i(2\pi k/N)$$

Donde B_i es la banda crítica número i , k los distintos coeficientes de la FFT de la señal y N el número de puntos totales de ésta.

Tras la obtención de la energía de cada filtro, contenida cada una en un único coeficiente situado en la frecuencia central de su banda se rellena el resto de este con ceros:

$$\tilde{Y} = \begin{cases} Y(i), & k = k_i \\ 0, & \text{resto} \end{cases}$$

Finalmente se realiza la transformada inversa de Fourier pasando así al dominio Cepstral:

$$c(n) = \frac{1}{N} \cdot \sum_{k=0}^{N-1} \tilde{Y}(k) \cdot e^{jk(2\pi/N)n}$$

Coeficientes delta

Los coeficientes delta y delta-delta, también conocidos como coeficientes velocidad y aceleración, son ampliamente utilizados en reconocimiento de habla junto a otras características como los MFCC ya mencionados o los LPC. Se pueden definir de forma aproximada como:

$$\Delta C_m[n] \approx \mu \sum_{k=-L}^{k=L} k \cdot c_m[n+k]$$

Donde μ es una constante de normalización y L el tamaño del vecindario, en número de tramas, sobre el que se calculan los coeficientes. (2)

En la Figura 1 se puede observar la aproximación realizada en este trabajo al cálculo de estos sobre un conjunto de ventanas de características.

La importancia de estos coeficientes reside en su utilidad a la hora de aportar información extra sobre las características extraídas de la señal, de gran interés en análisis de voz, como puede ser la velocidad de variación de estas, así como información del propio entorno de la ventana.

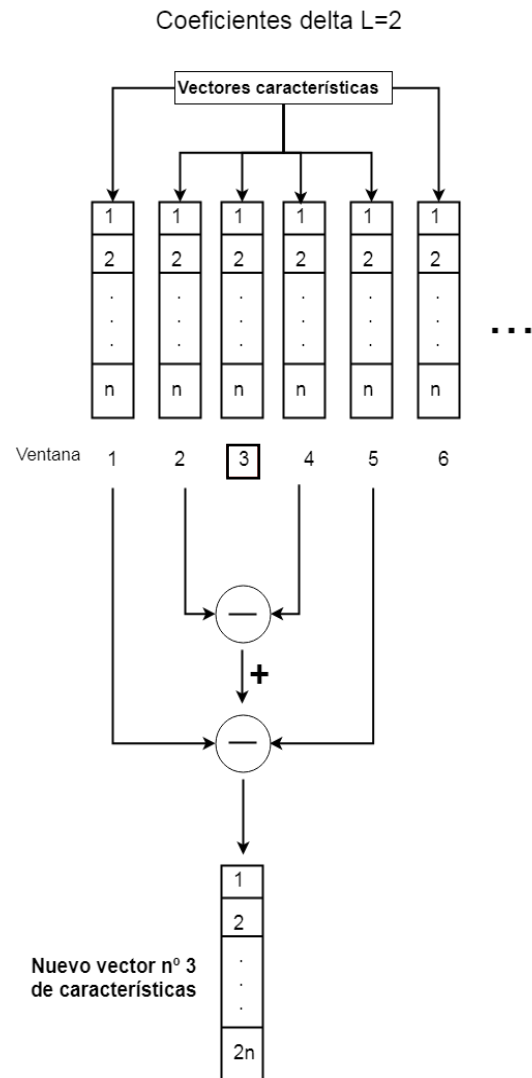


Figura 1: Esquema cálculo de coeficientes delta

2.2 Reconocimiento de patrones, tipos de detectores

Tras tratar el procesado de señal y la extracción de características típicas de voz se hará una breve descripción de distintas técnicas de reconocimiento de patrones para así aportar cierta perspectiva que será útil cuando posteriormente se expliquen los diversos detectores implementados en este trabajo.

2.2.1 Correlación cruzada

La correlación cruzada es una medida que indica la similitud entre dos señales dependiendo del desplazamiento que se dé entre ambas. Se usa tradicionalmente como forma de alinear señales, así como en detección de formas. (3) Matemáticamente la correlación cruzada entre dos señales $f(t)$ y $g(t)$ se define como:

$$\text{corr}(f(t), g(t)) \stackrel{\text{def}}{=} \int_{-\infty}^{\infty} f^*(\tau)g(t + \tau)d\tau$$

Donde f^* es el conjugado complejo de f y τ el desplazamiento de la señal $g(t)$. Es interesante comentar la relación entre la convolución, ampliamente utilizada en nuestro ámbito con la correlación, siendo esta una correlación en la que una de las dos señales es invertida:

$$\text{conv}(f(t), g(t)) \stackrel{\text{def}}{=} \int_{-\infty}^{\infty} f(\theta)g(t - \theta)d\theta$$

2.2.2 Distancia Euclídea

La distancia euclídea se define como la distancia al unir dos puntos dentro de un espacio euclidiano mediante una línea recta. En reconocimiento de patrones se utiliza como medida de similitud tanto en métodos de agrupamiento como de clasificación o detección. Matemáticamente queda definida para un espacio n -dimensional (3) como:

$$d(p, q) = \sqrt{\sum_{i=1}^n (q_i - p_i)^2}$$

Donde p y q son dos puntos en el espacio euclidiano n -dimensional. Como se puede observar la fórmula es equivalente a la medida del módulo del vector que une ambos puntos.

El uso de la distancia euclídea en algoritmos de reconocimiento de patrones presenta una ventaja respecto de otras que veremos más adelante, permite definir una distribución de

puntos en el espacio mediante un único representante, un centroide obtenido a partir de la posición media de ellos. Este servirá como referencia sobre la que hacer las medidas entre nuestro conjunto de puntos y otros puntos de este espacio. En la Figura 2 se puede ver como para un espacio de dos dimensiones y un único centroide se distribuyen las zonas que se encuentran a la misma distancia del centro.

2.2.3 Distancia de Mahalanobis

Si bien la distancia euclídea puede dar una idea de cómo se distribuyen los datos en el espacio, a la hora de medir esta solo tiene en cuenta la posición media del conjunto de puntos. En cambio, la distancia de Mahalanobis tiene en cuenta tanto lo posición media de los puntos como las direcciones de mayor varianza de estos dando lugar a un espacio de distancias que se adapta mejor a la distribución de estos puntos (ver Figura 2).

Matemáticamente la distancia de Mahalanobis de un punto \vec{x} respecto a una distribución de puntos $\vec{\mu}$ con una matriz de covarianzas S queda definida como: (3)

$$D_M(\vec{x}) = \sqrt{(\vec{x} - \vec{\mu})^T S^{-1} (\vec{x} - \vec{\mu})}$$

En caso de que la matriz de covarianzas sea la identidad, es decir, todas las variables tengan varianza unidad, la distancia de Mahalanobis pasa a ser equivalente a la euclídea.

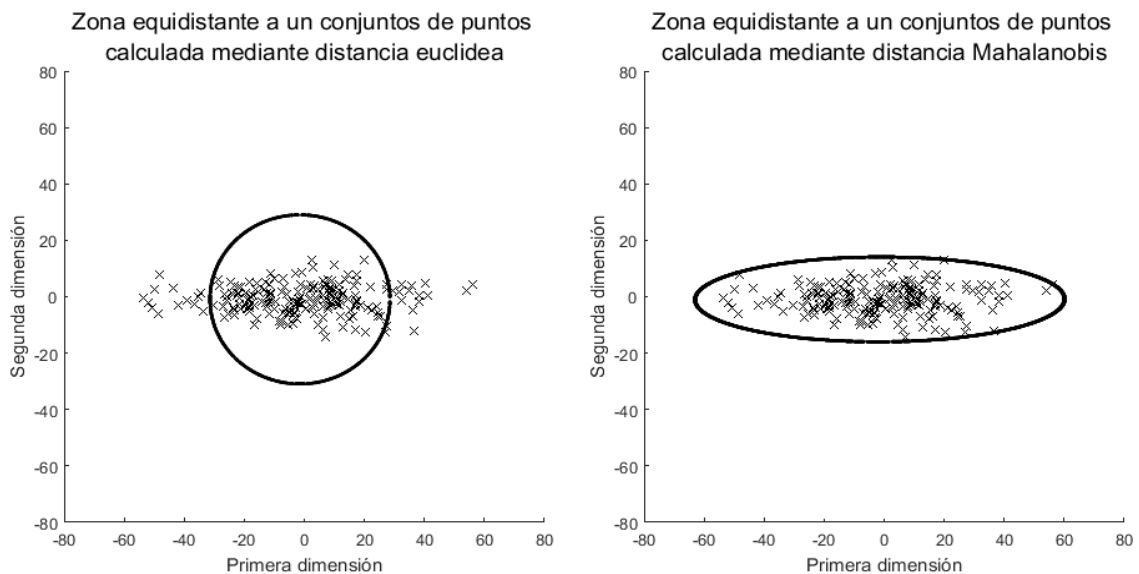


Figura 2: Comparación entre distancia euclídea y de Mahalanobis mediante la representación de zonas equidistantes a la media de un conjunto de puntos situados en un espacio bidimensional

2.2.4 Backend

Modelo ampliamente utilizado en reconocimiento de idioma como parte final del sistema y que mapea las distintas puntuaciones, obtenidas por los detectores previos, a una etiqueta clase-evento. Esto otorga una mayor robustez en detección además de permitir la integración de distintos detectores en un mismo espacio de dimensiones, realizando una fusión entre los diversos *scores* obtenidos por cada uno. Gracias a esto permite atajar problemas multiclase mediante la unión de distintos detectores, aumentando de esta forma el rendimiento del sistema. Hay diversos tipos de *backend*, de entre los que destacan los basados en gaussianas, utilizados como parte final en los sistemas implementados en este trabajo. Un ejemplo de ellos es el basado en la gaussiana multivariada:

$$\mathcal{N}(\mu, \Sigma) = \frac{1}{(2\pi)^{n/2} |\Sigma|^{1/2}} \exp\left(-\frac{1}{2} (x - \mu)^T \Sigma^{-1} (x - \mu)\right)$$

Donde n es el número de dimensiones Σ la matriz de covarianzas y μ la media. Una descripción más completa de sus distintos tipos y características se puede encontrar en (4).

2.3 Medidas de rendimiento en detectores y clasificadores

Una parte muy importante a la hora de implementar un detector o clasificador robusto y fiable es disponer de métodos y herramientas que nos permitan comprobar su rendimiento de forma objetiva. Es por ello que a continuación se comentarán diversas técnicas que se usan con este objetivo.

2.3.1 Falso rechazo y falsa aceptación

En sistemas que funcionan asignando puntuaciones según la clase se puede medir su rendimiento según la relación de estas con los *target* y *non target* del sistema. Otra medida interesante sobre estas puntuaciones es la que nos dice hasta que punto son separables ambas clases con los *scores* obtenidos y en qué medida estos se solapan. Como forma de visualizar esta separación entre ambas clases según sus puntuaciones, se emplean histogramas como el de la Figura 3. En ella también se observa cómo se solapan las puntuaciones obtenidas para cada clase por lo que una separación perfecta entre ambas nunca es posible y se ha de asumir cierto grado de error a la hora de distinguir entre ambas. Por ello, al separar en clases se pueden dar dos tipos de errores:

- Falso rechazo o falso negativo: Cuando se considera como *non target* un elemento que era *target*.
- Falsa aceptación o falso positivo: Cuando se considera *target* un elemento que realmente es *non target*.

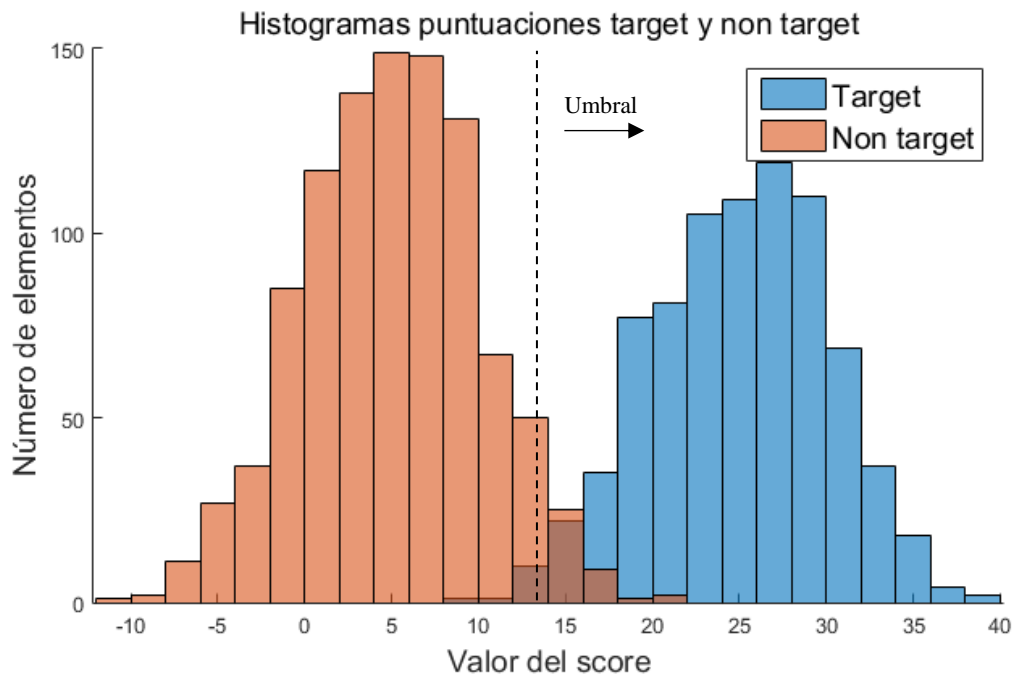


Figura 3: Histograma de puntuaciones obtenidas para clase target y para non target

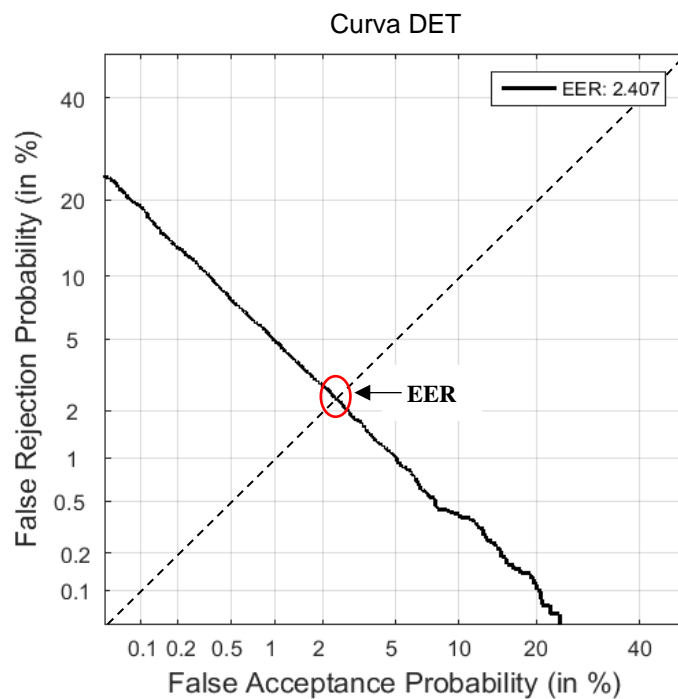


Figura 4: Curva DET obtenida sobre puntuaciones de distribución gaussiana

El umbral de decisión que separa el histograma de forma que ambos tipos de error se produzcan en la misma cantidad es el llamado *equal error rate (EER)*. Es importante resaltar que, dependiendo del tipo de aplicación del sistema, puede ser más importante minimizar uno de los dos tipos de error que mantener una igualdad entre ellos. Por ejemplo, en aplicaciones de biometría y seguridad es más importante evitar los accesos de personal no autorizado (minimizar las falsas aceptaciones) que no dejar fuera a alguien que debería poder entrar (falso rechazo). En cualquier caso, un sistema debería guardar cierto compromiso entre ambos en orden de ser lo más fiable posible.

2.3.1 Curvas DET

Una manera de representar toda esta información de manera sencilla es mediante el uso de curvas DET (*Detection error trade-off*). Estas gráficas representan como varía tanto la falsa aceptación como el falso rechazo según el umbral que se decida para clasificar entre *target* y *non target*. Además, tiene la propiedad de que para puntuaciones cuya distribución es gaussiana la DET resultante es recta. El punto en el que corta la DET con la diagonal es el EER del sistema como se puede ver en la Figura 4.

2.3.2 Otras medidas de rendimiento: Precision y recall

Otras medidas de rendimiento ampliamente utilizadas en reconocimiento de patrones y basadas también en los conceptos de falsos negativos y falsos positivos son la *precision*, *recall* y *f-score*. El *recall* da información sobre el número de *targets* obtenidos correctamente respecto del total de *targets*:

$$R = \frac{T_p}{T_p + F_n}$$

En cambio, la precisión mide el número de *targets* clasificado correctamente respecto del total de elementos que hemos considerado como *target*:

$$P = \frac{T_p}{T_p + F_p}$$

Donde T_p es el número de elementos clasificados correctamente, y F_p el número de falsos positivos y F_n es el número de falsos negativos. Por último, el *F-score* da la relación entre las dos medidas anteriores:

$$F = 2 \frac{P \times R}{P + R}$$

3 Diseño

En este capítulo se explicarán brevemente los distintos esquemas de detección empleados, así como la base de datos utilizada tanto en entrenamiento como en test, centrándonos en las singularidades y propiedades de las señales que la componen y que son parte fundamental a la hora de entender y profundizar en este trabajo.

3.1 Base de datos, señales disponibles

Para la realización de este trabajo se ha dispuesto de una serie de señales procedentes del análisis de piezas industriales mediante sensores que miden ciertas propiedades físicas de estas. Estas señales han sido proporcionadas por una empresa externa, lo que complica el acceso a nuevas señales sobre las que probar los detectores, así como a ampliar la base de datos.

3.1.1 Características de las señales

Base de datos de desarrollo

La primera base de datos empleada en este trabajo está compuesta por una serie de señales especiales que contienen una gran cantidad de eventos, extraños de ver en una señal normal. La utilidad de estas señales es, por tanto, la de testear algoritmos de detección, como el implementado, de forma que se pueda cuantificar fácilmente el número de aciertos sobre una gran cantidad de eventos haciendo los resultados obtenidos más robustos.

Aunque este tipo de señales nos son extremadamente útiles a la hora tanto de entrenar como de probar nuestro detector, también hay una serie de inconvenientes que no podemos dejar de comentar.

- El número de señales que componen esta base de datos es reducido, en torno a 30 señales distintas obtenidas. Esto puede dar problemas tanto a la hora de entrenar como a la hora de realizar detección sobre señales que estén fuera de la base de datos.
- Todas las señales de este tipo siguen una estructura idéntica, puesto que han sido obtenidas a partir de diversos escaneos de una misma pieza. Si bien no son todas iguales, difieren tanto en amplitud como en muestras, todos los eventos están siempre ubicados de la misma forma y son parecidos entre ellos. Esto hace que se haya de tener especial cuidado para evitar crear un detector con problemas de *overfitting* y que sea generalizable a señales distintas, que es el objetivo último de este proyecto.

Base de datos de generalización

Esta segunda base de datos, más reducida, contiene señales muy distintas de la anterior con eventos similares, pero de distintas magnitudes y formas, puesto que se han obtenido de piezas distintas a las de la base de desarrollo. Por ello esta base será muy útil de cara a probar la robustez de nuestros sistemas. En el anexo A se pueden observar algunas de estas señales, así como los eventos que contienen.

Propiedades

Las señales de ambas bases de datos, por ser resultado del análisis de piezas industriales y debido a las características intrínsecas de estas, en las que no entraremos, tienen una serie de rasgos muy marcados y que hay que tener en cuenta a la hora de tratarlas.

- El primero es que las señales están divididas en tres partes claramente diferenciadas como se ve en la Figura 5. Las zonas de los extremos, claramente marcadas por subidas y bajadas muy pronunciadas indican su comienzo y final, y la zona intermedia, de mucha menor amplitud contiene los distintos eventos a detectar. Es por ello que a la hora de analizar estas señales solo nos interesará lo que sucede en esta zona, conocida como la *zona activa* de la pieza (ver Figura 5).
- Otra característica inherente a estas señales es la de que según avanzan en número de muestras esta tiende a subir en amplitud. Esto que denotaremos como *deriva* sucede en todas las señales procedentes de estos sensores.

3.1.2 Tipos de eventos

En lo que llevamos de memoria hemos nombrado en reiteradas ocasiones el concepto de evento, entendiendo por este las diferentes variaciones que puede sufrir las señales y que deseamos detectar. Ahora definiremos sus diferentes clases, así como sus características más relevantes.

- **Desviaciones:** Variaciones muy fuertes y rápidas en la medición del sensor. Importantes y visibles a simple vista. En la Figura 5 aparecen representadas en rojo.
- **Gaps o huecos:** Pequeñas variaciones. Desviaciones de menor orden y por tanto importancia. Se pueden confundir con las desviaciones e incluso el ruido. En verde en la Figura 5
- **Transiciones:** Zonas de paso entre distintas zonas de la pieza. De gran importancia en detección tanto por ser un propio evento como por delimitar diferentes zonas de la pieza. En azul en la Figura 5.
- **Ruido:** El propio sensor que capta la señal introduce un ruido muy característico y que se desea detectar en ausencia de otros eventos.

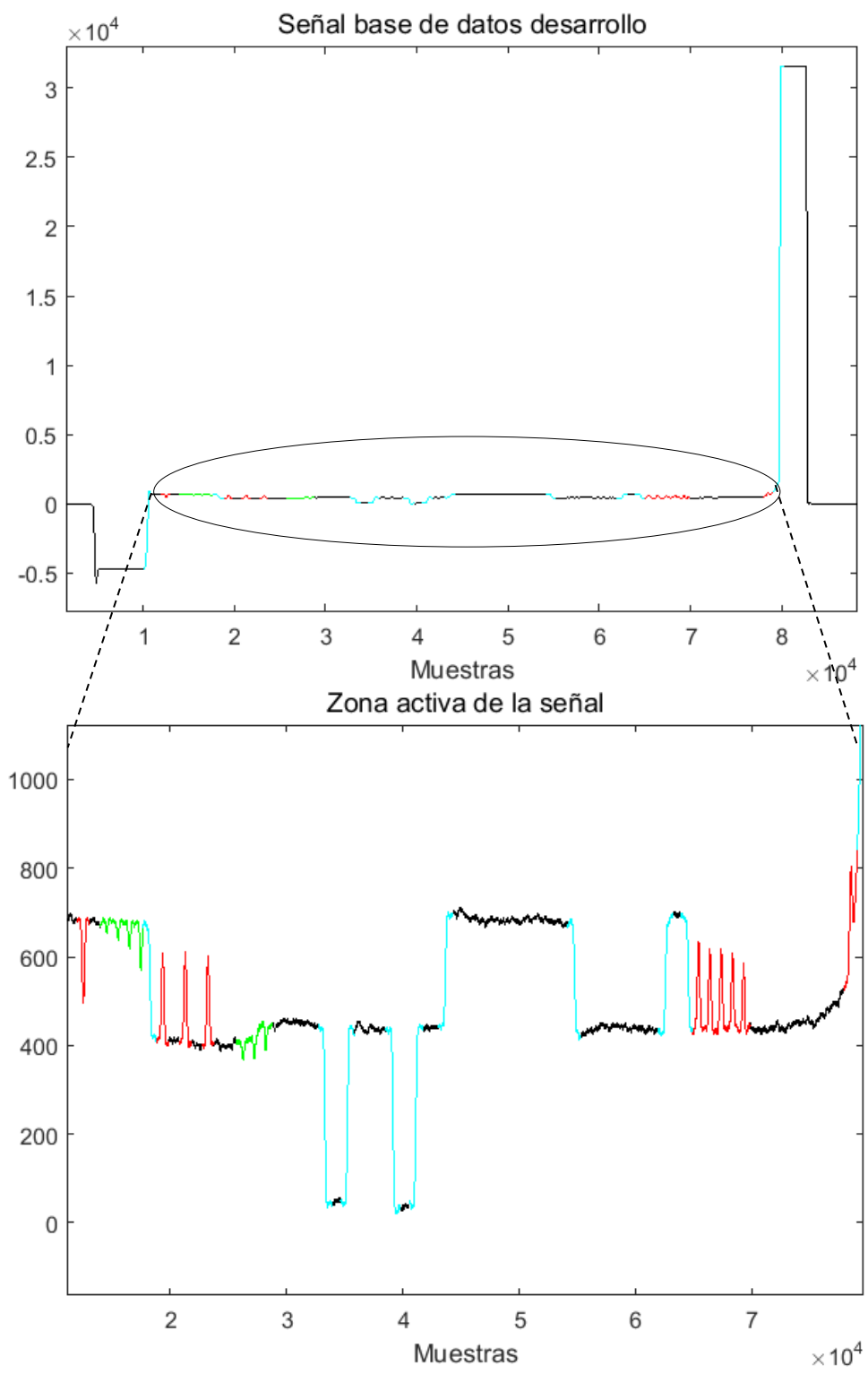


Figura 5: Señal de la base de datos de desarrollo y zona activa de esta con sus eventos etiquetados

3.1.3 Protocolo Experimental

Tras comentar brevemente de qué está compuesta nuestra base de datos, así como las características de esta, vamos a explicar cómo se ha dividido a la hora de realizar tanto el entrenamiento como el *test*. Debido a los problemas comentados anteriormente, como es lo reducido de la base de datos de desarrollo, se decidió hacer uso de un sistema de **validación cruzada** o *'leave one out'* con objetivo de reforzar los resultados obtenidos sobre esta base y hacerlos más robustos.

Este método, destinado a la evaluación de resultados obtenidos por el detector, consiste en ir rotando las señales de la base usadas, tanto en entrenamiento como en test. Para ello se usan todas las señales de la base de datos menos una para entrenar el detector y se hacen las pruebas de rendimiento pertinentes sobre la restante. Tras esto se almacenan los resultados obtenidos y se cambian los conjuntos de entrenamiento y test; se selecciona otra señal distinta para test y el resto de la base se usa para entrenamiento. Este proceso se repite hasta que se obtienen resultados sobre todas las señales.

Las pruebas realizadas sobre la base de datos de generalización, en cambio, se realizaron sobre la base al completo, sin validación cruzada, tras entrenar con la de desarrollo.

3.2 Arquitectura del Sistema

En este trabajo se han implementado fundamentalmente dos esquemas distintos de detectores de eventos. Uno basado en correlación, es decir, detección por forma de señal, y uno más complejo basado en las características espectrales de los eventos. Todos ellos forman parte de un sistema que sigue un esquema *frontend-backend*. representado en la Figura 6. y en la que cada parte realiza las siguientes funciones:

- La primera parte, llamada *frontend*, consta de varios detectores (uno por evento) y se encarga tanto de la extracción de características en las señales como del cálculo de las puntuaciones de cada detector sobre la señal (ver Figura 7). Esta ha sido la parte sobre la que se ha trabajado en este proyecto, pero aun así se explicará brevemente el funcionamiento del *backend* para que el lector pueda entender el sistema completo.
- La segunda parte, el *backend* toma la decisión final sobre la clase a la que pertenece cada muestra. Para ello trabaja directamente sobre los *scores* obtenidos por los detectores, buscando la relación entre estos y los diferentes eventos, de forma que si uno de los detectores falla para un tipo de evento determinado pero el resto de detectores se comporta de forma coherente, el clasificador es capaz de discernir el tipo de evento correctamente. Este sistema ha de ser entrenado, como el *frontend*, antes de usarse en clasificación.

En la Figura 7 se pueden observar las diferentes etapas que sigue el *frontend* y que se explicarán brevemente.

- Antes de nada, se realiza el pre-procesado necesario sobre la señal (inventariado, etiquetado, diezmo, etc.) según las características tanto de nuestro detector como de las señales sobre la que se trabaja. Es de vital importancia en esta primera etapa una correcta selección de parámetros para conseguir un rendimiento adecuado del detector.
- Tras esto se extraen las características que usará nuestro detector para discriminar entre los diferentes eventos. Es importante en este paso una selección correcta de características, por lo que un estudio previo a la selección es necesario para un correcto funcionamiento del detector. Las características seleccionadas deben cumplir dos compromisos fundamentales: permitir una distinción entre eventos lo mayor posible, y ser lo suficientemente robustas como para poder ser utilizadas con distintos conjuntos de entrenamiento, evitando el *overfitting* todo lo posible.
- Finalmente, el detector de cada evento recoge estas características y siguiendo un criterio determinado (distancias, varianzas, etc.) obtiene unas puntuaciones. La puntuación obtenida será mayor cuanto más parecidas sean las características analizadas con las características introducidas al sistema durante el entrenamiento. Estas puntuaciones son enviadas al *backend* para su análisis y clasificación final.

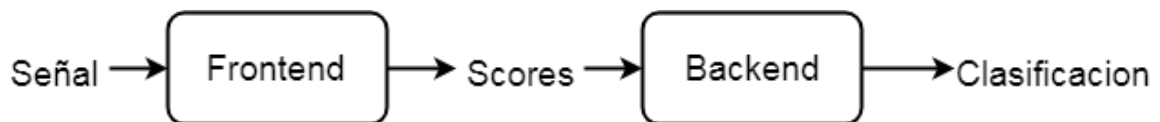


Figura 6: Esquema base clasificador *frontend-backend*.

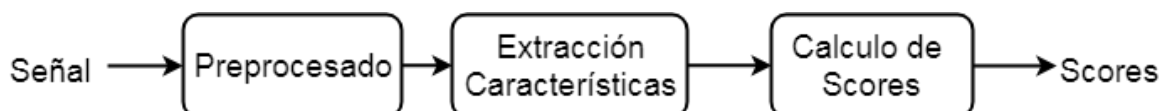


Figura 7: Esquema básico *frontend*.

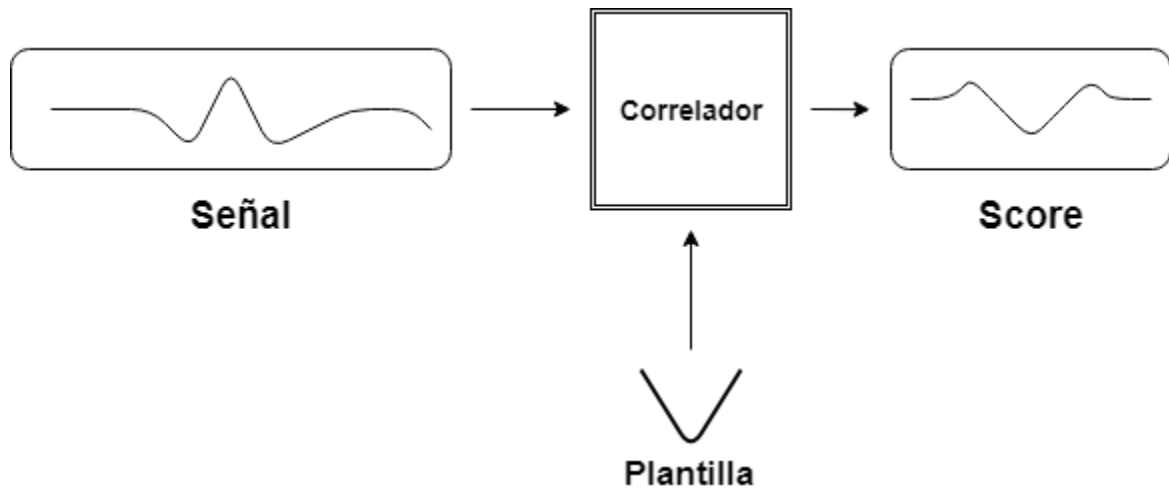


Figura 8: Esquema básico funcionamiento de un detector por correlación

3.2.1 Detector por Correlación

Frontend basado en la detección de eventos según su forma. Para ello se crean varios modelos similares a las partes de la señal que se desean detectar y tras ello se aplica la correlación cruzada entre la señal, sobre la que se desea hacer las detecciones, y cada uno de estos modelos (plantillas). Esto nos da unos valores de correlación, que usaremos como puntuaciones de nuestros detectores (un detector por modelo), y que serán mayores cuanto más parecida sea la forma de nuestra plantilla y la parte de la señal sobre la que se aplica. Este tipo de sistemas se suelen usar en presencia de ruido gaussiano aditivo como forma de descomponer en señal y ruido (5), así como para detección en señales digitales.

3.2.2 Detector Espectral

El segundo *frontend* implementado en este trabajo ha sido el detector espectral, considerablemente más complejo que el anterior y que explota las diferencias de los eventos a nivel frecuencial para discernir entre ellos. Para ello tras pre-procesar la señal y inventanarla se extraen estas características y agrupan según los diferentes eventos. Tras esto se sigue un criterio de distancias, ya sea euclídeas o de Mahalanobis, para comparar estas agrupaciones con datos desconocidos que se deseen clasificar.

4 Desarrollo

En este apartado se profundizará en el proceso de creación e implementación de los diferentes detectores, así como en su funcionamiento y en las diferentes técnicas utilizadas. Para ello siguiendo el esquema mostrado en la Figura 7, empezaremos explicando el procedimiento utilizado para una correcta extracción de características, así como para la selección de las más relevantes.

4.1 Extracción de características

Como se ha explicado anteriormente la selección de unas características lo suficientemente discriminantes entre los diferentes eventos es probablemente el paso más importante a la hora de obtener un detector que funcione correctamente, puesto que de nada sirve la utilización de algoritmos extremadamente complejos si estos no disponen de la información necesaria para una correcta distinción de clases. Es por ello que antes de pasar a hablar del funcionamiento en profundidad de los diversos detectores implementados vamos a analizar las señales y eventos tratados en el apartado 3.1 con más profundidad, observando sus características concretas.

4.1.1 Pre-procesado y características de las Señales

Como se ha explicado en el apartado 3.1.2 y se puede ver en la Figura 5 nuestras señales disponen de varios tipos de eventos con características muy diferentes entre ellos que nos permitirán, idealmente, una discriminación clara.

Para facilitar una correcta extracción de características y hacer esta más robusta se hace un cierto pre-procesado a la señal previo a la extracción y que es común a todos los detectores implementados en este trabajo.

- Se aplica sobre todas las señales un filtrado paso bajo Chebyshev Tipo II para eliminar parte del ruido.
- Tras esto se hace la derivada de la señal. Esto tiene varios motivos: la zona activa de las señales tiene zonas de diferente amplitud. La derivada, al eliminar la componente continua de la señal, nos permite tratar todos los eventos de la misma forma, sin preocuparnos de la zona en la que se encuentren. Además, esto nos permite contrarrestar el efecto de deriva comentado anteriormente. La señal resultante se puede observar en la Figura 9.

Sobre la señal resultante de este pre-procesado se realiza la extracción de características. De todas ellas podemos destacar para cada evento:

- Las desviaciones destacan por su estrechez y gran amplitud, y pueden ser tanto positivas como negativas. En la derivada aparecen representadas como un doble pico.

- Las transiciones son grandes subidas o bajadas, más lentas (anchas) que las desviaciones. En la derivada aparecen representadas como picos tanto hacia arriba como hacia abajo
- Los gaps, o huecos aparecen como picos en la señal original y como dobles picos en la derivada. Se diferencian fundamentalmente de las desviaciones en que siempre son hacia abajo y de mucha menor amplitud. Estas características hacen que sea un evento difícil de detectar y clasificar.
- Por último, el ruido, que destaca por su baja amplitud y mayor presencia en altas frecuencias que el resto de eventos.

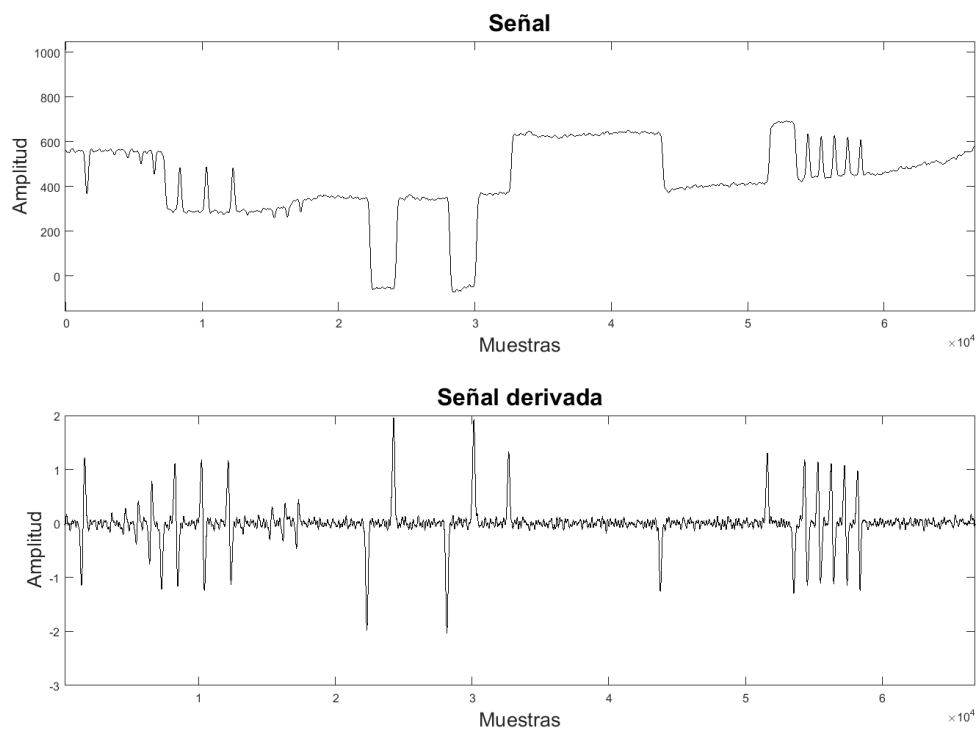


Figura 9: Zona activa de la señal y derivada de esta.

Del análisis de características hecho en el apartado anterior sobre los eventos podemos destacar una serie de características que se podrían usar para su detección:

- Forma: Algunos de los eventos como las desviaciones tienen una forma muy distinta a otros como las transiciones (forma de dobles picos y picos en la señal derivada).
- Temporales: Energía, potencia, variaciones, ...
- Estadísticas: Media, varianza, ...
- Espectrales: Módulo, fase, MFCC, ...

Tras listar las posibles características a utilizar en un problema como el planteado se decidió hacer dos implementaciones distintas:

- Un detector por correlación que explota las diferencias de forma entre eventos a través de la función de correlación cruzada.
- Un detector a partir de las componentes espectrales de la señal, puesto que estas contienen información tanto de la orientación de la señal como de su amplitud y velocidad de variación, todas ellas características relevantes a la hora de distinguir entre los diferentes eventos.

4.1.2 Características detector espectral, análisis de varianzas

Tras la elección de este segundo conjunto de características, y debido a que el número de estas era demasiado grande, dando lugar a problemas computacionales y de *overfitting*, se hizo una discriminación de características con el objetivo de mantener tan solo aquellas que permitiesen diferenciar entre los eventos con mayor facilidad.

Para ello se hizo un análisis de las diferentes características por varianza. Este análisis consiste en la comparación entre la variación de las diferentes características entre eventos distintos (varianza interclase) con como varían estas mismas características en eventos del mismo tipo (varianza intraclase). De esta forma se obtiene una medida objetiva sobre el valor de las diferentes características a la hora de distinguir entre clases. Algunos de los resultados obtenidos mediante este análisis se pueden ver en la Figura 10. Tras esto se seleccionaron las características con mejor relación inter/intra dando lugar a un total de 15 características espectrales: 10 coeficientes del módulo y 5 de la fase. Algunos ejemplos de cómo los diferentes eventos quedan distribuidos según estas características se encuentran en la Figura 11.

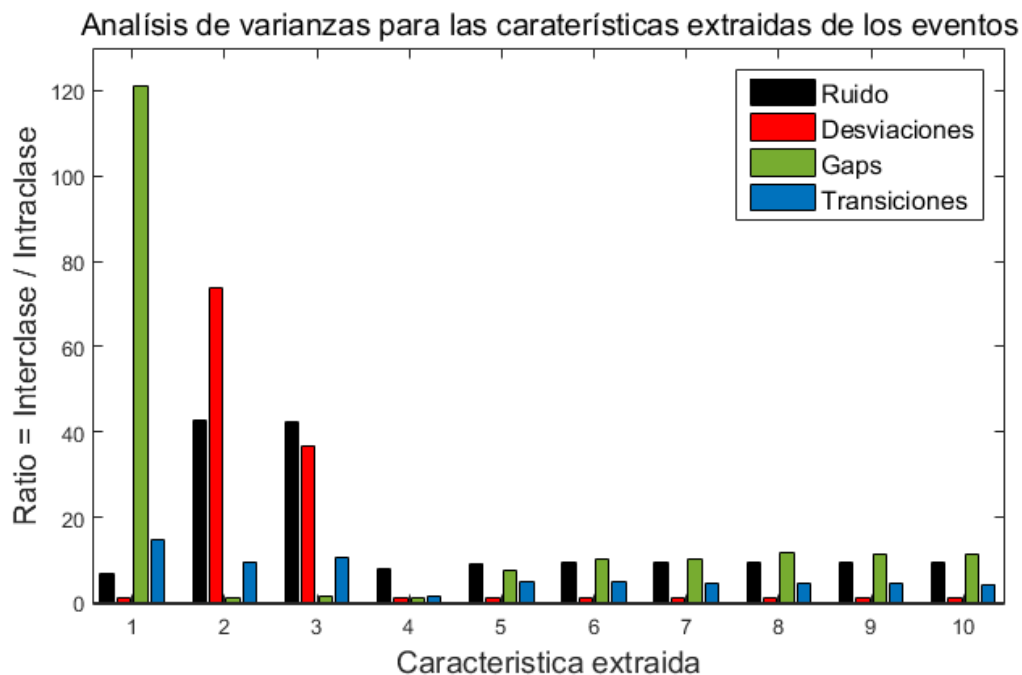


Figura 10: Resultados análisis de varianzas para los primeros 10 coeficientes de la FFT

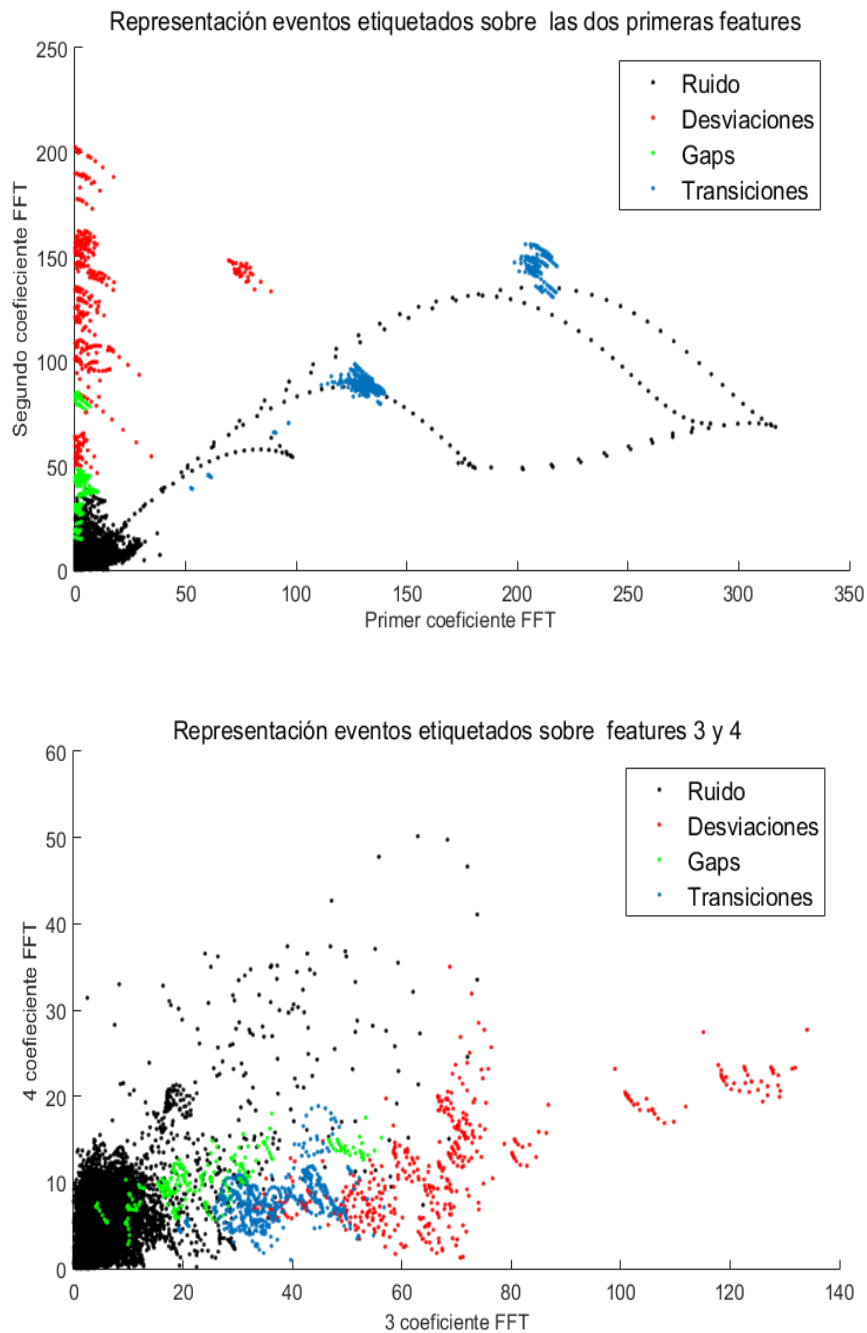


Figura 11: Representación de los diferentes eventos según algunas de sus características extraídas

Coefficientes delta

Como manera de reforzar las características obtenidas y debido a las propiedades inherentes a los diferentes eventos se realizó sobre estas una variación de los coeficientes delta, técnica ampliamente utilizada en voz, que a cambio de duplicar el número de características aumenta enormemente la robustez de nuestros detectores. Estos coeficientes son especialmente útiles a la hora de detectar eventos sobre las señales inventanadas, puesto que la información del transitorio de los eventos estará presente en

las ventanas adyacentes. Destaca la utilidad de este método a la hora de diferenciar entre dos de los eventos: las desviaciones y los gaps, eventos que, como ya se comentó antes, pueden llegar a ser realmente parecidos y cuya distinción no es nada trivial.

4.2 Detector por correlación

El primer detector implementado en este trabajo fue un detector por correlación que hace uso de las diferencias de forma entre los distintos eventos y de la función de correlación cruzada para así diferenciar entre los distintos tipos de eventos presentes en la señal. Como muchos detectores este requiere de una fase previa de entrenamiento antes de poder ser utilizado, por lo que se han de dividir las señales de la base en conjuntos de entrenamiento y test. Para ello se han seguido los siguientes pasos:

1. Selección de una serie de señales de entrenamiento con sus distintos eventos etiquetados.
2. Aplicación del pre-procesado indicado en el apartado 4.1.1 sobre estas señales. Se guardan y agrupan los distintos eventos presentes en estas para su uso posterior.
3. Estos eventos guardados, aun siendo del mismo tipo, son parecidos, pero no idénticos. Para hacer estos eventos comparables entre si son procesados, realizando un alineamiento por máxima correlación y un recorte posterior, para hacerlos de la misma longitud, de forma que tras esto todos los eventos son perfectamente equivalentes.
4. Tras esto se hace una media de todos los eventos del mismo tipo obteniendo una plantilla de cada uno. Plantilla que cada detector usará como modelo a la hora de hacer la correlación con la señal de test. Estas plantillas son normalizadas entre cero y uno. En la Figura 12 podemos observar el proceso de creación de una de estas plantillas.

Tras este entrenamiento consistente en la creación de las distintas plantillas, se procede a la detección de la siguiente forma:

Se realiza la correlación cruzada de la señal sobre la que se desea analizar con cada una de las plantillas creadas con las señales de entrenamiento. Cada una de estas correlaciones nos dará unos valores que, teóricamente, serán más altos cuanto mayor sea el parecido entre el evento presente en la señal y la plantilla sobre la que se compara. Tras realizar este proceso con cada una de las plantillas obtendremos tantas puntuaciones (correlaciones) distintas como cantidad de plantillas hayamos creado. Estas serán enviadas al *backend* para su posterior clasificación. En la Figura 13 está representada la puntuación obtenida sobre una señal tras aplicar una plantilla con forma de pico negativo sobre ella, dando valores mayores en las zonas más parecidas.

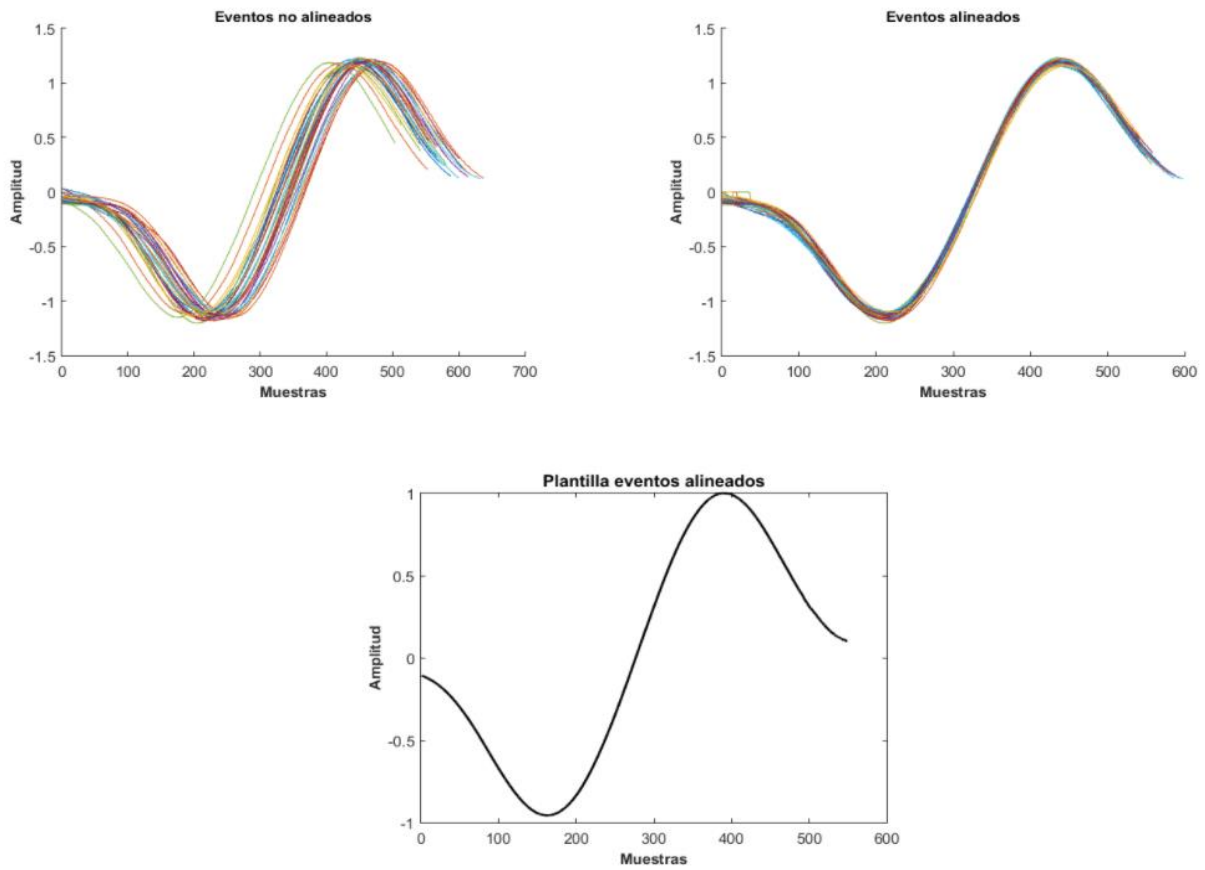


Figura 12: Proceso de creación de plantillas

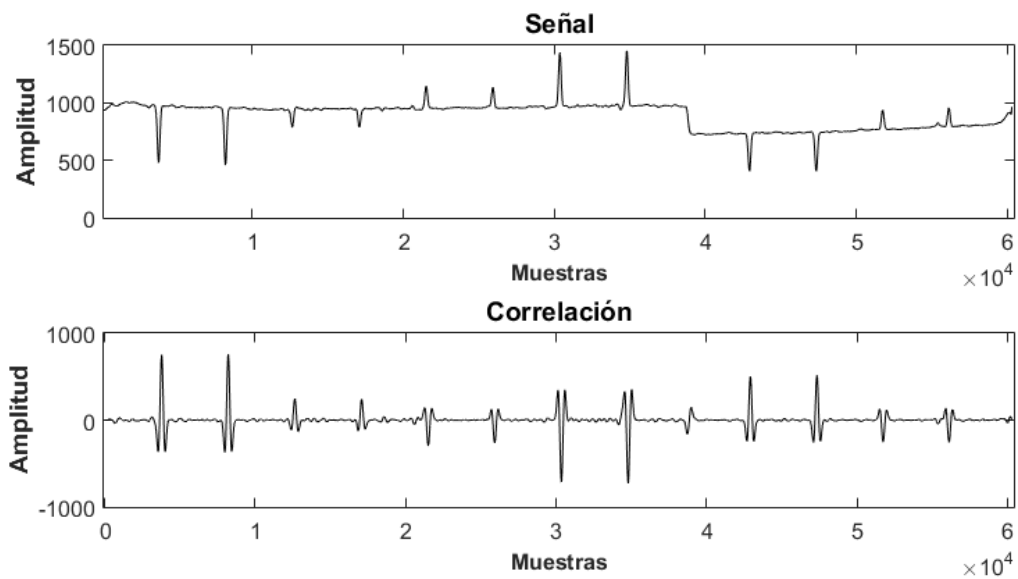


Figura 13: Señal y puntuación obtenida mediante la correlación de la señal con una de las plantillas

Como se puede observar en a Figura 13 este tipo de detector da lugar a unos resultados muy precisos a la hora de distinguir entre zonas sin evento y con evento, pero tiene una serie de limitaciones que no se pueden ignorar:

- Diferencias de forma: Algunos eventos pueden tener varias formas distintas (por ejemplo, las desviaciones pueden ser tanto hacia arriba como hacia abajo), lo que complica su detección y requieren de la creación de varias plantillas para un único evento.
- Influencia de la amplitud de los eventos: Es muy sensible a las diferencias de amplitud de los eventos. Si bien los resultados obtenidos mediante la correlación son ponderados según su forma, una gran diferencia de amplitud entre dos eventos distintos puede dar lugar a detecciones incorrectas.
- La forma de dos de los eventos que se desean detectar es la misma: Tanto algunas desviaciones como los gaps son caracterizados como picos hacia abajo o dobles picos en su derivada. Por ello este sistema no permite discernir entre ambos eventos.

4.3 Detector espectral

El segundo tipo de detector implementado en este trabajo hace uso de las características espectrales explicadas anteriormente y sigue el esquema *frontend-backend* visto en el punto 3.2.2. Además, este detector, a diferencia del correlador, no solo distingue entre desviaciones, gaps y transiciones, sino que también tiene su propio detector de ruido modelado según sus características. Como el anterior, requiere de una fase de entrenamiento previa a su uso, tanto en el *frontend* como en el *backend*, y hace uso del pre-procesado comentado en el apartado 4.1.1, véase filtrado paso bajo como forma de mitigar el ruido y derivada de la señal para eliminar la continua. Tras esto se realizan varios pasos.

4.3.1 Enventanado

Tras el pre-procesado se procede a enventanar la señal y etiquetar cada ventana. Esta tarea, que parece trivial, es de vital importancia a la hora de conseguir un entrenamiento del detector eficiente y funcional. Se han de elegir ventanas lo suficientemente grandes como para que puedan contener totalmente los eventos (de forma que las características extraídas de esta ventana sean representativas de estos) pero no tan grandes como para que se puedan dar varios eventos en la misma ventana o se introduzca un ruido excesivo. Tras diversas pruebas y debido a diferentes propiedades de las piezas de las que han sido extraídas nuestras señales se llegó a unos valores de tamaño de ventana de 600 muestras con un desplazamiento entre ventana de tan solo 15 muestras. Además, se usó como ventana una de tipo rectangular, puesto que suaviza menos las componentes frecuenciales de la señal.

4.3.2 Etiquetado

El cómo etiquetar cada ventana es también una tarea en la que las diferentes decisiones de diseño son de vital importancia, puesto que ¿cuál es la forma óptima de etiquetar una ventana de manera que se refuerce el entrenamiento todo lo posible?, ¿etiquetando todas las ventanas que contengan alguna muestra de un evento como pertenecientes a este?, ¿solo las ventanas que contengan un % del evento?, ¿las que contengan el evento completo? Tras diversas pruebas y desarrollos se llegó a la conclusión de que lo ideal era solo etiquetar una ventana como perteneciente a un evento en caso de que este se encuentre completamente contenido por la ventana. De esta forma las características del evento no se ven distorsionadas por el proceso de inventanado y etiquetado. Todas las ventanas que incluyen eventos de forma parcial son descartadas a la hora de entrenar.

4.3.3 Extracción de características

Como se comentó en el apartado 4.1 se hace uso de las características frecuenciales de la señal para así distinguir entre los eventos. Para ello se hace la FFT de cada ventana y se guardan tanto el módulo como la fase resultante. Tras esto se seleccionan los coeficientes más relevantes, es decir, los que mejor puntuación obtuvieron en el análisis de varianzas interclase/intraclase. Estos son: 10 coeficientes procedentes del módulo y 5 procedentes de la fase. Con estas 15 características resultantes se calculan los coeficientes delta sobre las ventanas adyacentes como forma de obtener información sobre el entorno del evento. Esto nos duplica el número de características dejándonos con un total de 30 que representaran cada ventana.

4.3.4 Entrenamiento del detector

Una vez se tienen las características que representan los diferentes eventos se pasa a la utilización de distintas técnicas que agrupan estas por evento y nos permiten comparar nuevas características con las ya agrupadas en entrenamiento. En este detector se probaron dos métodos distintos de hacer este proceso:

- **Distancia Euclídea:** A la hora de entrenar se agrupan todos los vectores de características procedentes de los mismos eventos y se hace la media de ellos, dando lugar a un único centroide por evento, sobre el que se comparara cualquier vector desconocido que se desee clasificar. Para ello, en test, se hará una medida de distancia Euclídea de este vector n-dimensional (siendo n el número de características de nuestro vector, 30 en nuestro caso) a cada uno de los centroides representantes de cada evento, obteniendo de esta manera tantas distancias como centroides (eventos) se hayan entrenado. De esta forma tendremos un detector por centroide, cada uno con su puntuación, es decir, distancia del vector de test al centroide.
- **Distancia de Mahalanobis:** Como en el caso anterior para entrenar se agrupan los vectores según el evento al que pertenecen. Pero a diferencia de la distancia euclídea en este caso no se pueden reducir estos vectores a un único centroide, puesto que, como ya se explicó en el apartado 2.2.3, la distancia de Mahalanobis no depende solo del módulo de la distancia sino también de la matriz de covarianzas del conjunto de puntos obtenidos a partir de los datos de entrenamiento. Por ello

se guardan completos en una matriz todos los vectores procedentes del mismo evento obteniendo de esta forma una matriz por cada detector de evento. Esto nos dará en test una puntuación por cada evento basada en la distancia del vector de test a cada matriz.

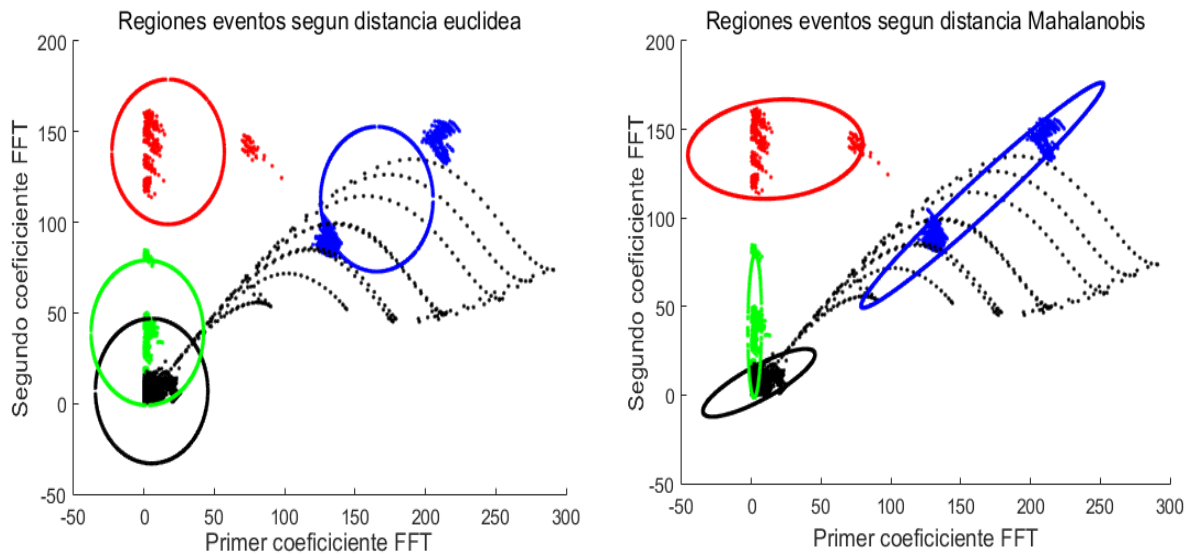


Figura 14: Ejemplo de regiones definidas para cada evento mediante distancia euclídea y distancia de Mahalanobis en un espacio bidimensional

En la Figura 14 se puede observar claramente como la distancia de Mahalanobis, al tener en cuenta las direcciones de máxima varianza de los datos, crea regiones mucho más cercanas a la distribución real de los eventos que la distancia euclídea. En esta gráfica los círculos representan zonas que se encuentran todas a la misma distancia del centro, definido por los diferentes eventos en el espacio de dos dimensiones conformado por sus dos primeras características.

Tras la obtención de las diferentes distancias (puntuaciones) en el *frontend* los resultados son mandados al *backend*. La implementación dada para este trabajo ha sido la de un *backend* gaussiano. Con estas puntuaciones calculara una serie de gaussianas que modelan la distribución de los distintos scores según el evento, de esta forma ante la llegada de una serie de nuevos scores que se desean clasificar obtendrá unas probabilidades determinadas de que estos representen un evento.

4.4 Fusión de detectores

El último detector implementado en este trabajo es el resultante de la combinación de los dos detectores anteriores a nivel de *frontend*. Es decir, mediante la combinación de toda la información que se puede extraer de las características anteriores, tanto de forma, como de espectro, y el posterior uso de estas en un esquema de *frontend-backend*. Para ello simplemente se varía el espacio de características que se utilizaba en el detector espectral,

ya comentado, añadiendo cuatro nuevas: el resultado de aplicar la correlación en cada ventana con las plantillas creadas para el detector por correlación. Esto añade más información que podrá ser usada por nuestro detector tanto en su forma Euclídea como de Mahalanobis para separar con mayor facilidad entre los diferentes eventos, como se puede observar en la Figura 15.

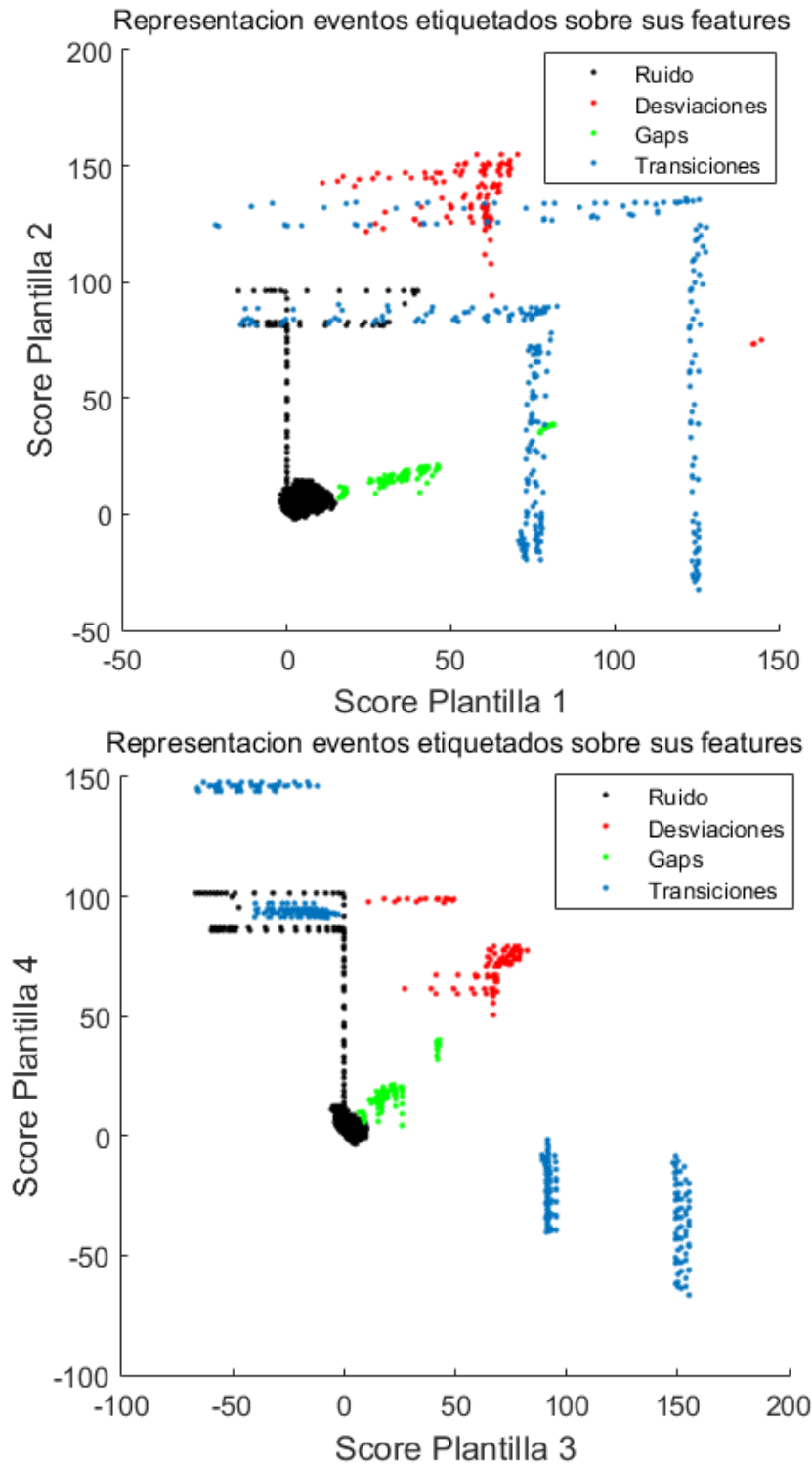


Figura 15: Representación de los diferentes eventos según la correlación obtenida entre cada ventana y las diferentes plantillas

5 Integración, pruebas y resultados

A continuación, se procede a presentar los distintos resultados obtenidos para los detectores ya comentados. Se prestará especial atención a los resultados obtenidos a nivel de detección, pero también se mostrarán, en los casos que proceda, resultados obtenidos en clasificación tras el paso de las puntuaciones por el *backend* gaussiano. Para ello haremos uso de las diversas técnicas comentadas en el apartado 2.3, como el EER o las curvas DET. Los *scores target* y *non target* con los que se han dibujado todas las curvas DET de este apartado son los máximos locales de las puntuaciones obtenida por cada detector, ya que nos permiten evaluar con mayor precisión el rendimiento de estos. Además, todas las pruebas realizadas han seguido los procedimientos explicados en el apartado 3.1.3.

5.1 Detector por correlación

Los primeros resultados a mostrar serán los obtenidos mediante el detector por correlación que, como se ha comentado anteriormente, está compuesto por 5 detectores diferentes, uno por cada evento con sus 5 plantillas correspondientes. Se analizarán primero los resultados obtenidos sobre las señales que conforman nuestra base de datos de desarrollo y en un segundo apartado sobre la base de generalización.

5.1.1 Resultados sobre base de desarrollo

Los resultados obtenidos, mediante validación cruzada, por este detector son extremadamente buenos sobre las señales de esta base, discerniendo perfectamente en que puntos se encuentran los eventos y dando lugar a unas buenas curvas DET y a unos EER muy bajos, en algunos casos prácticamente nulos (ver Figura 16). Aun así, hay un problema que se ha de comentar y que se aprecia en la Figura 17, los *scores* obtenidos tanto para las desviaciones negativas como para los *gaps* son exactamente iguales puesto

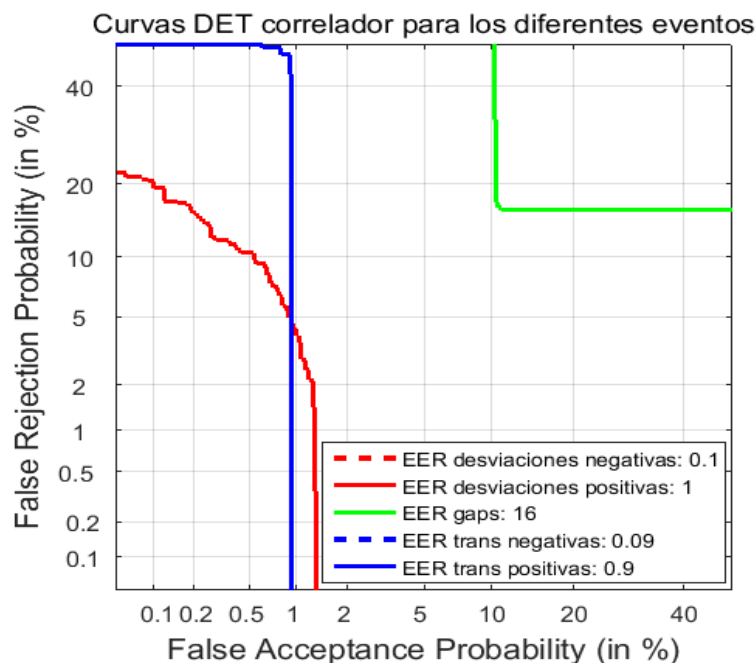


Figura 16: Curvas DET para cada evento obtenidas sobre la base de datos de desarrollo

que tienen la misma forma. Esto hace imposible que el detector distinga entre ambos eventos. Además, existen falsos positivos de gran valor que provocan la aparición de rectas en las curvas DET.

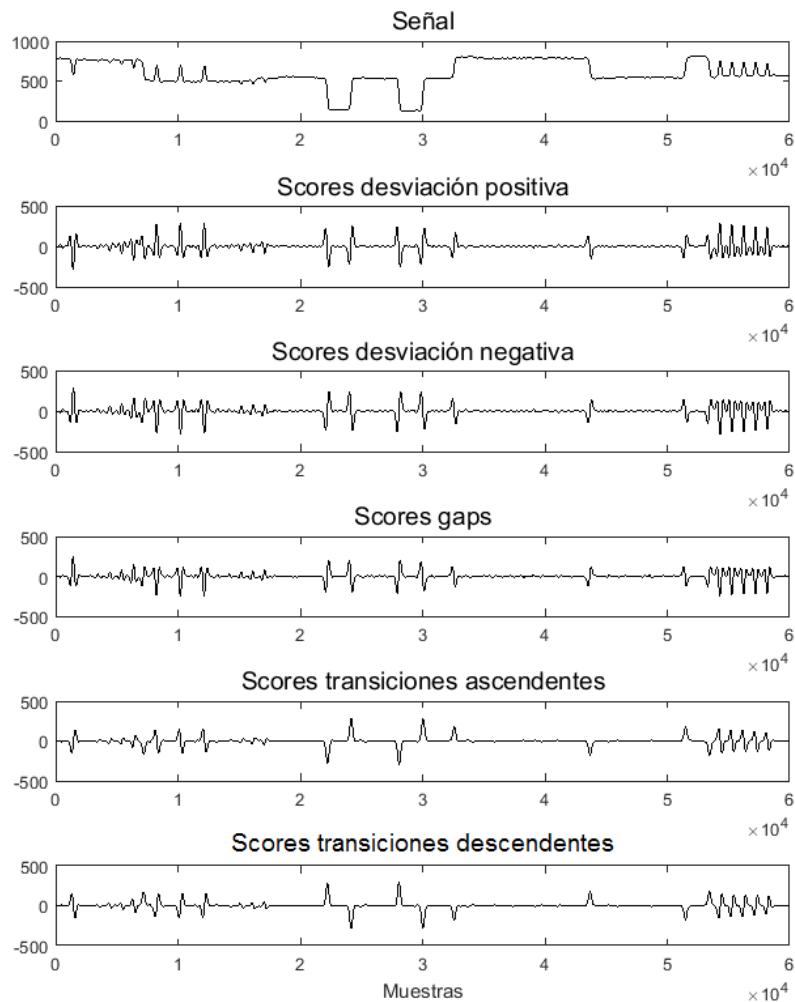


Figura 17: Scores obtenidos por el detector de cada evento sobre una de las señales de la base de datos de desarrollo

5.1.2 Resultado sobre base de generalización

En este apartado se presentarán los resultados obtenidos tras entrenar las plantillas con las señales de la base de datos de desarrollo y aplicar detección sobre las de la base de datos de generalización. De este segundo conjunto de señales cabe destacar la ausencia de eventos tipo *gap*, lo que evitará el problema comentado en el apartado anterior, pero aun así el detector presenta sobre estas señales otro tipo de problemas que no podemos ignorar.

Como se ve en la Figura 18 el EER y las curvas DET en general empeoran sobre este conjunto de señales. Esto es debido fundamentalmente a las diferencias de amplitud tan grandes

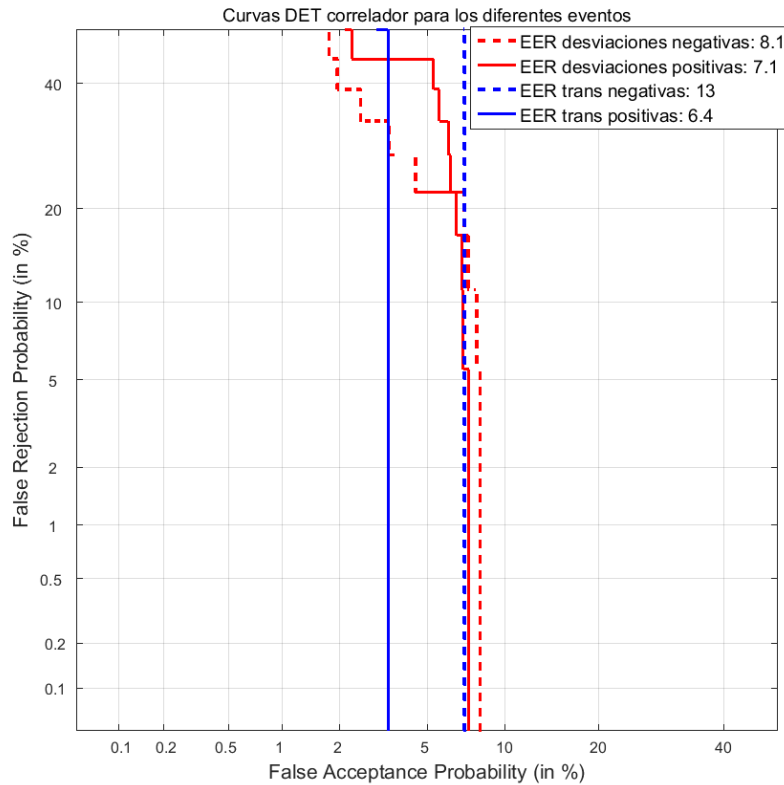


Figura 18: Curvas DET para cada evento obtenidas sobre la base de datos de generalización

que hay entre los diferentes tipos de eventos. Como se aprecia en el ejemplo de la Figura 19 las desviaciones tanto positivas como negativas tienen una amplitud mucho mayor que el resto de eventos, lo que nos lleva al segundo problema asociado a este tipo de detector, la gran influencia que tienen las amplitudes de los eventos sobre los *scores* obtenidos. Por ejemplo, en esa misma figura se aprecia como los *scores* de transiciones dan máximos incorrectos en las desviaciones de la señal debido a esa diferencia de amplitud y esto queda representado en la curva DET de la Figura 18 como rectas, ya que tenemos falsos positivos de mayor valor que los positivos reales.

5.1 Detector espectral

Como se ha comentado en apartados anteriores en este trabajo se han realizado dos versiones diferentes del detector, una basada en distancia euclídea y otra basada en distancia de Mahalanobis. Por ello se presentarán los resultados obtenidos con ambas, así como la diferencia de rendimiento resultantes de la inclusión, o no, de los coeficientes delta en las características.

5.1.1 Resultados sobre base de desarrollo

Las primeras pruebas se realizaron sobre la base de desarrollo, comentada en el apartado 3.1. En la Figura 20 y la Figura 21 se pueden ver los resultados obtenidos mediante la evaluación del sistema siguiendo un procedimiento de validación cruzada. Se observa como ambas implementaciones funcionan muy bien sobre esta base dando lugar a curvas muy bajas. Cabe destacar la gran mejora de rendimiento aportada por los coeficientes delta en

todos los eventos y como Mahalanobis se adapta mejor a estos datos que la euclídea. Sin embargo, habrá que comprobar si esta adaptación sobre la base de datos no es excesiva, dando problemas de *overfitting*.

En la Figura 23 y la Figura 24 se observan las clasificaciones obtenidas por el sistema sobre una de las señales tras el paso de las puntuaciones representadas en la Figura 22 por el *backend* gaussiano. Se observa como las etiquetas obtenidas a la salida del sistema son prácticamente idénticas a las etiquetas de *Ground Truth* (ver Anexo A), obteniendo en ambos casos unas medidas de rendimiento del clasificador muy altas.

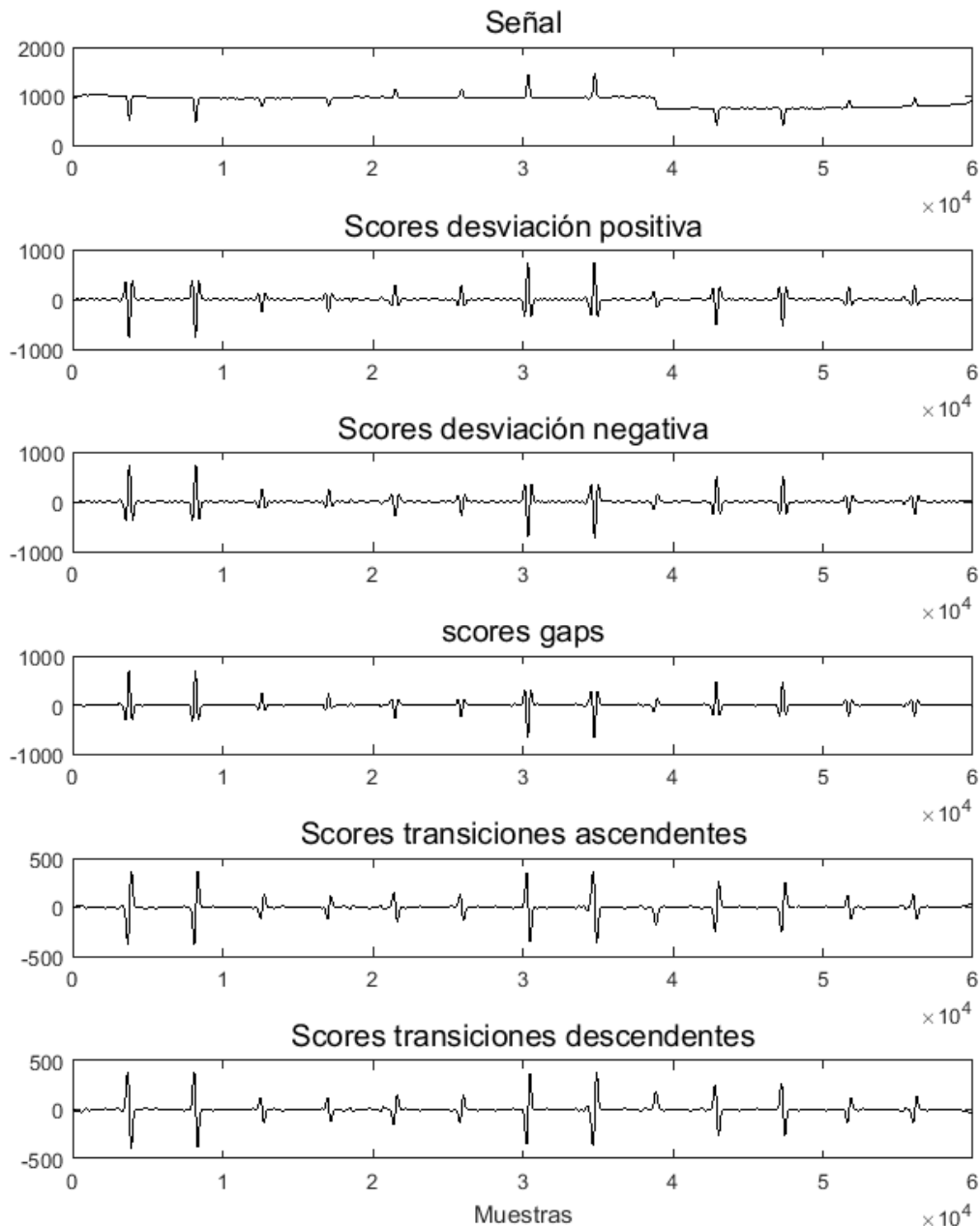


Figura 19: Scores obtenidos por el detector por correlación de cada evento sobre una de las señales de la base de datos de generalización

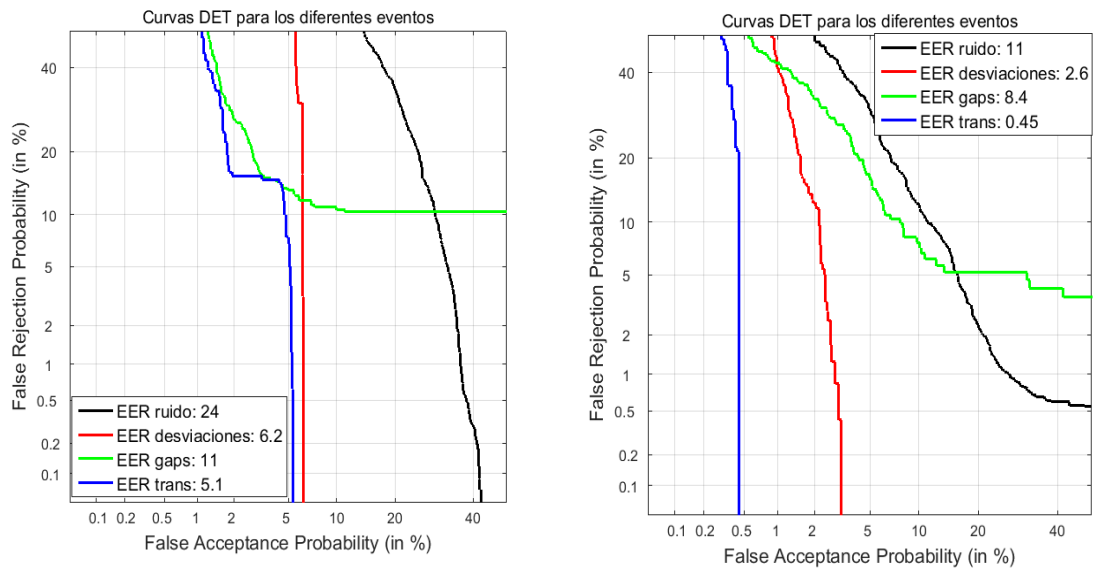


Figura 20: Curvas DET para cada evento (base de datos de desarrollo) con detector espectral y euclídea sin coeficientes delta (izquierda) y con coeficientes delta (derecha)

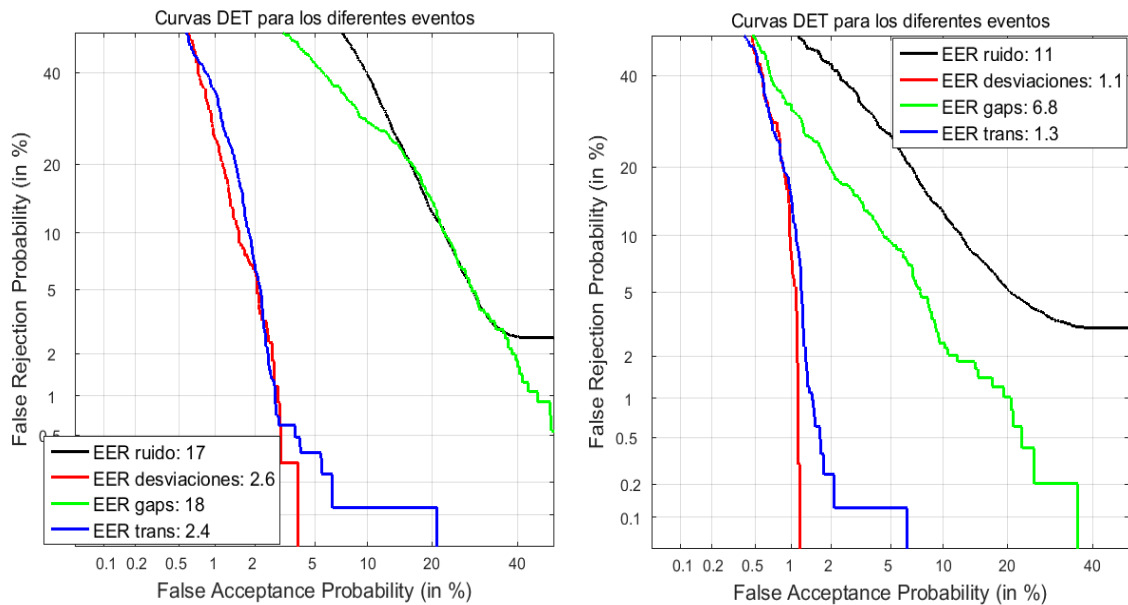


Figura 21: Curvas DET para cada evento (base de datos de desarrollo) con detector espectral y Mahalanobis sin coeficientes delta (izquierda) y con coeficientes delta (derecha)

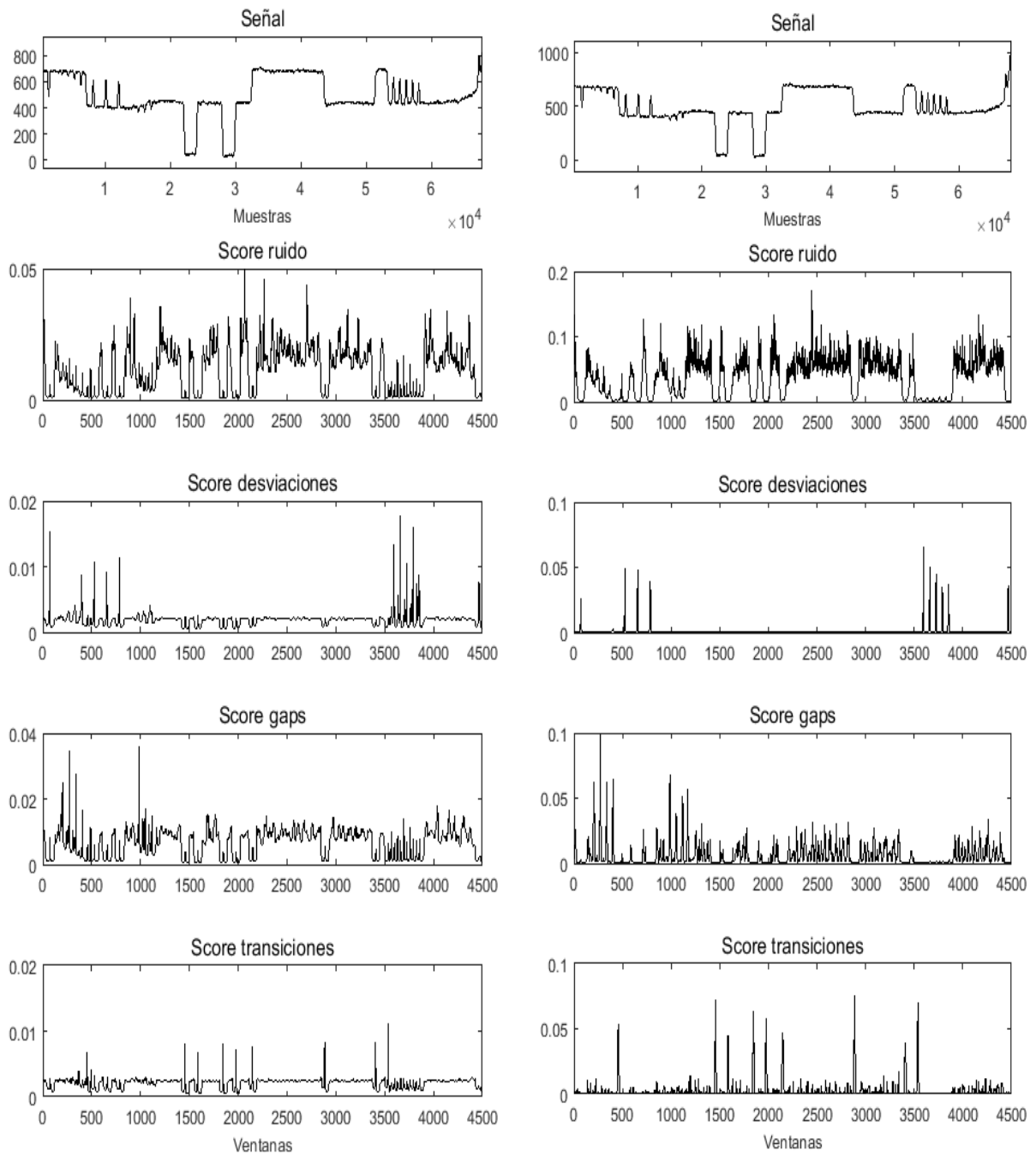


Figura 22: Scores obtenidos para señal de base de datos de desarrollo por los detectores espectrales de cada evento tanto con distancia euclídea (izquierda) como de Mahalanobis (derecha)

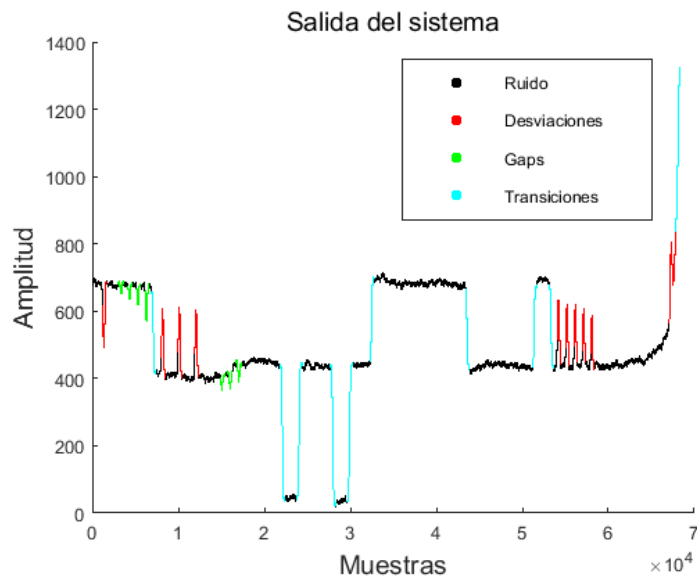


Figura 23: Clasificación realizada por el *backend* sobre una de las señales de la base de desarrollo para el detector espectral con distancia euclídea sin coeficientes delta. $Fscore = 0.87$, $Accuracy = 0.97$, $Precision = 0.83$

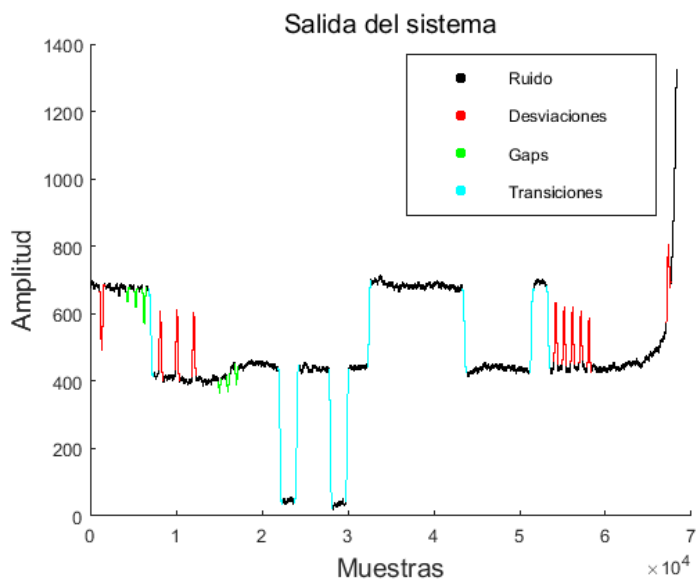


Figura 24: Clasificación realizada por el *backend* sobre una de las señales de la base de desarrollo para el detector espectral con distancia Mahalanobis y coeficientes delta. $Fscore = 0.91$, $Accuracy = 0.98$, $Precision = 0.94$

5.1.2 Resultado sobre base de generalización

Tras realizar las diferentes pruebas sobre la base de datos de desarrollo y como forma de obtener unos resultados más robustos se han replicado estas mismas pruebas sobre la segunda base de datos que contiene señales que difieren en gran medida de las anteriores. Estos resultados se han obtenido entrenando el sistema con la base de datos de desarrollo al completo y evaluando este sobre toda la base de datos de generalización.

En la Figura 25 se puede observar los resultados, en forma de curvas DET, obtenidos para esta base de datos. Si bien los resultados son peores que los obtenidos en el apartado anterior siguen dando lugar a EER bajos con eventos fácilmente distinguibles. En la Figura 27 están representados los *scores* obtenidos por cada detector sobre una de las señales de esta base y en la Figura 26 la salida del *backend* para otra de las señales. En todos los casos los resultados son buenos, con tasas de error bajas.

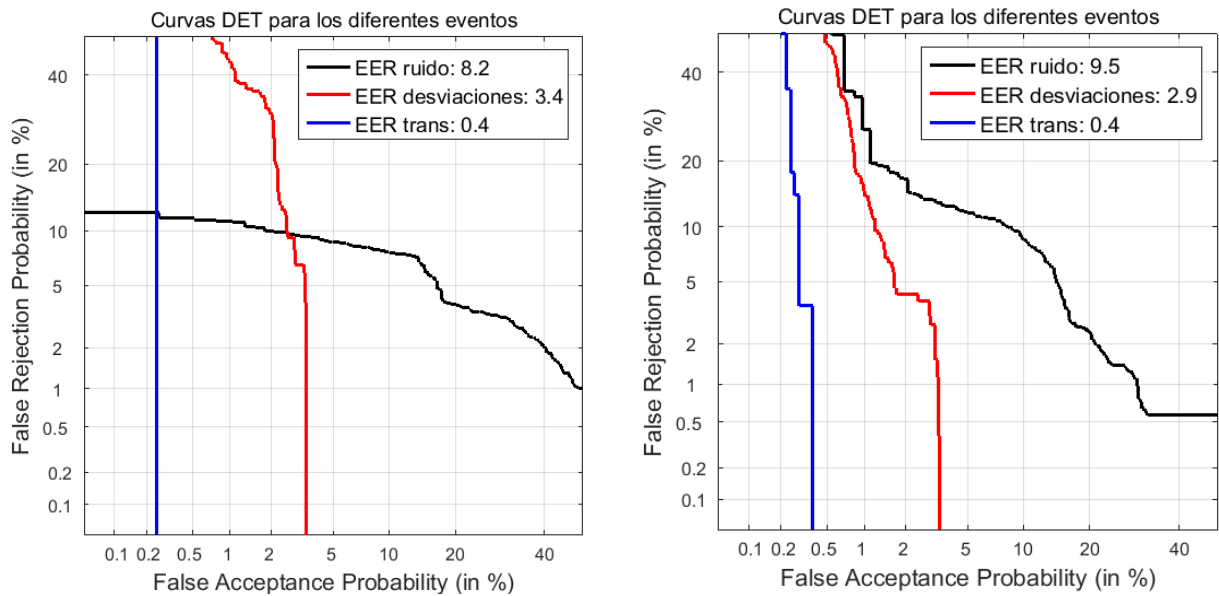


Figura 25: DET obtenidas sobre base de datos de generalización. A la izquierda por el detector espectral con distancia euclídea, a la derecha con Mahalanobis

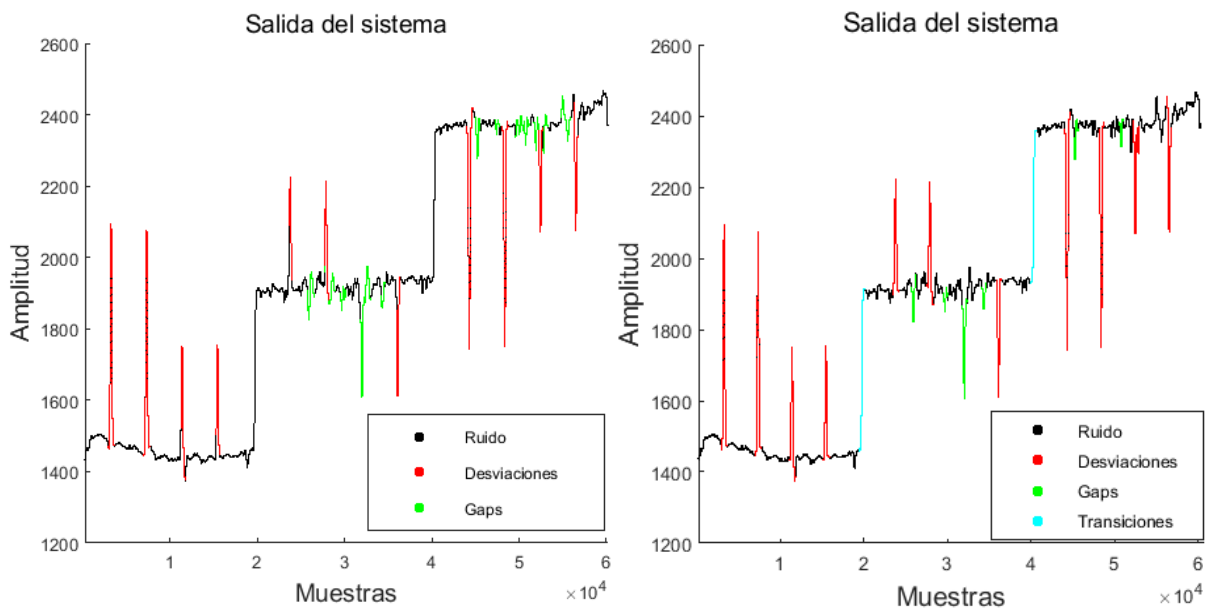


Figura 26: Clasificación realizada por el backend sobre señal de la base de datos de generalización. A la izquierda con detector espectral de distancia euclídea, a la derecha con Mahalanobis

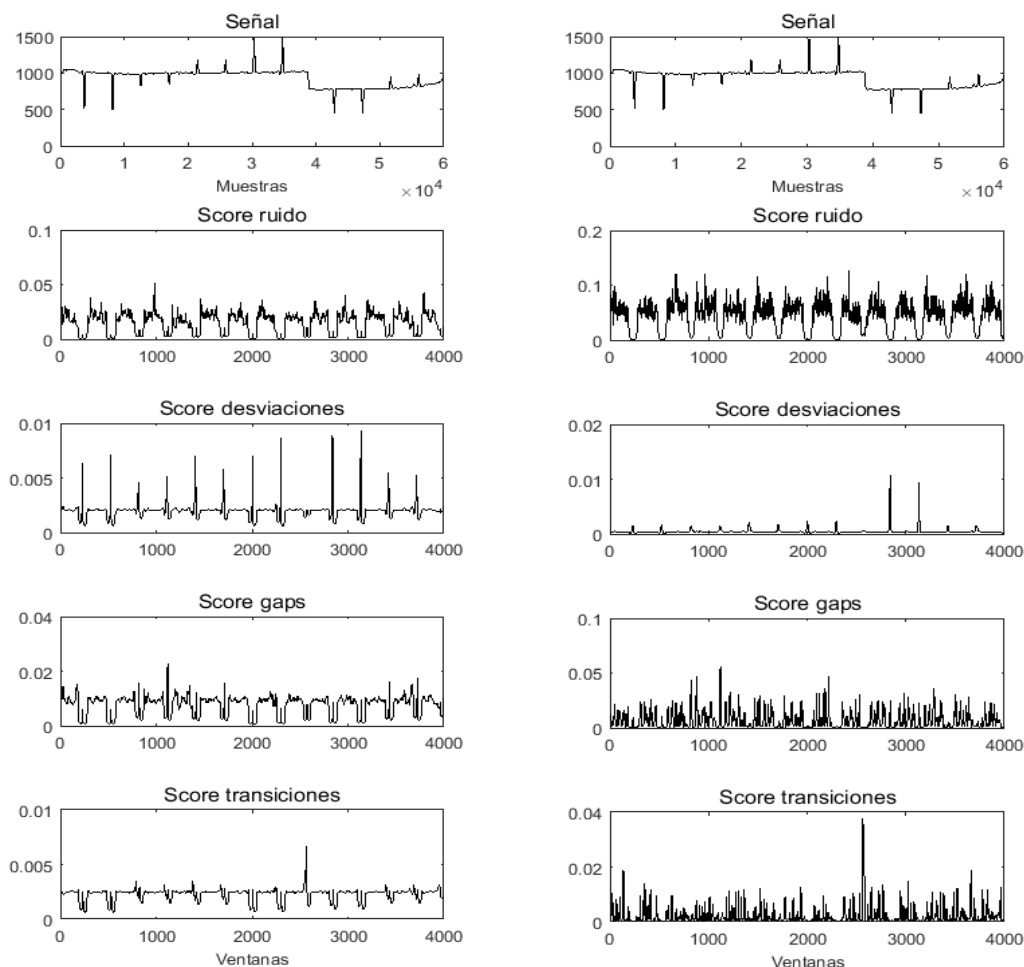


Figura 27: Izquierda puntuaciones detector espectral con distancia euclídea sobre señal de la base de datos de generalización. A la derecha puntuaciones con distancia Mahalanobis

5.1.3 Comparación con detector por correlación

Con los resultados obtenidos podemos afirmar que el detector espectral, en comparación con el de correlación, es mucho más robusto y generalizable; como se puede observar en los resultados de ambos casos sobre la base de generalización. Además, cabe destacar que carece de dos de los grandes problemas asociados al correlador: la gran variabilidad del detector según la amplitud de los eventos y la imposibilidad de distinguir entre desviaciones negativas y *gaps*. Otra de sus ventajas es la presencia de un detector de ruido propio al contrario que en el correlador.

5.2 Fusión de detectores

Por último, se presentarán los resultados obtenidos con el detector por fusión tras someterse a la misma serie de pruebas que el espectral, usando el mismo protocolo de pruebas y bases de datos.

5.2.1 Resultados sobre base de desarrollo

Como se puede observar en la Figura 29 las curvas DET resultantes son tremendamente dispares para las dos configuraciones del detector, funcionando extremadamente mal con distancia euclídea y obteniendo mejores resultados que el espectral con Mahalanobis. Esta disparidad también se aprecia en las puntuaciones representadas en la Figura 33, presente en el Anexo A.

5.2.2 Resultados sobre base de generalización

Sobre la base de datos de generalización los resultados obtenidos con distancia euclídea mejoran respecto de la base de desarrollo mientras que la implementación con Mahalanobis da buenos resultados en ambos casos. Un ejemplo de los *scores* obtenidos sobre una de las señales de esta base se puede encontrar en el Anexo A, Figura 34.

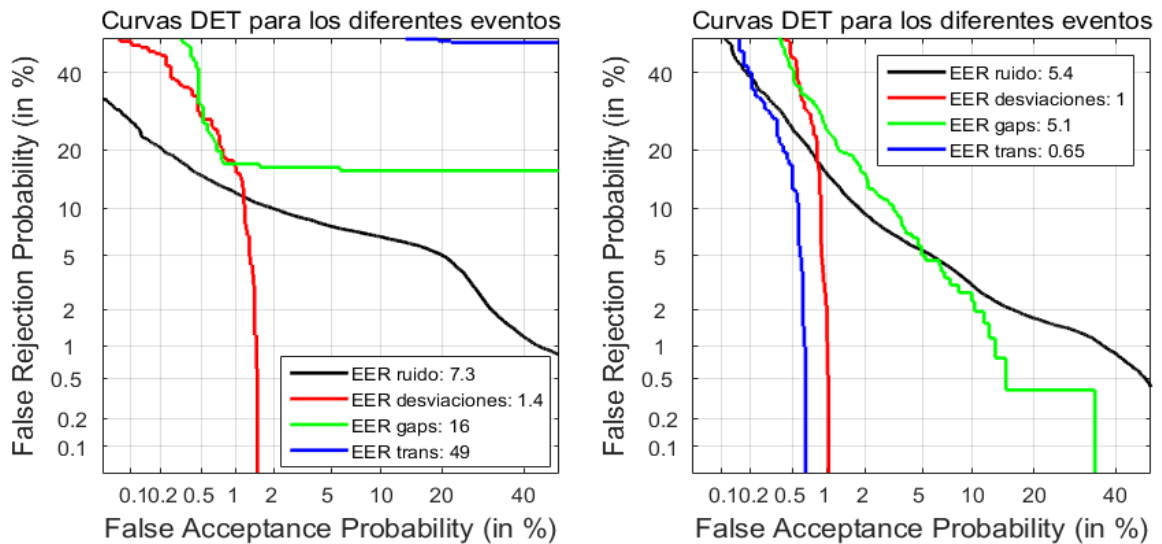


Figura 29: Izquierda DET Fusión euclídea, derecha DET Fusión Mahalanobis calculadas sobre base de datos de desarrollo

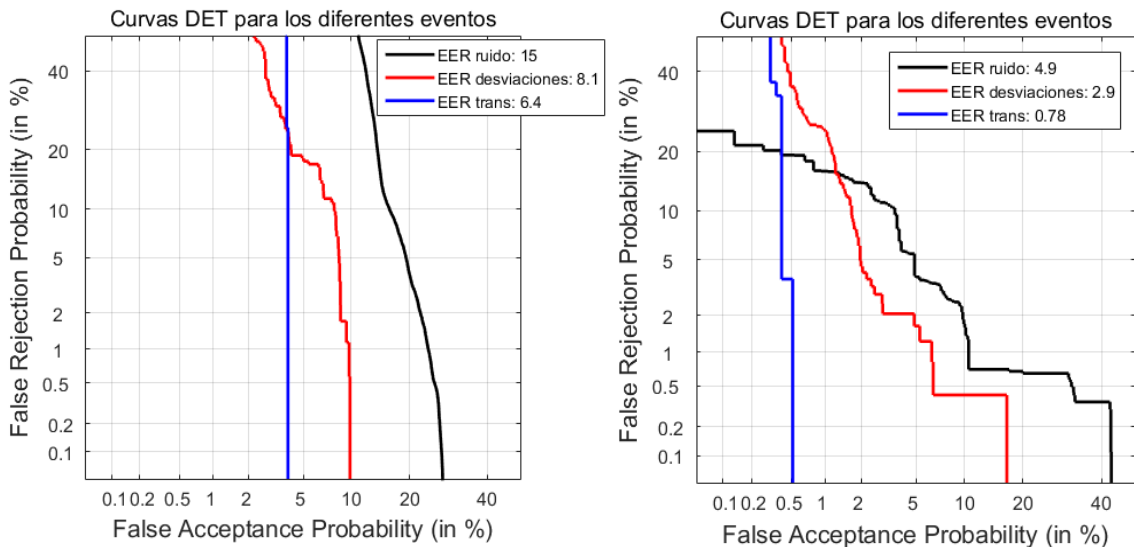


Figura 28: Izquierda DET Fusión euclídea, derecha DET Fusión Mahalanobis calculadas sobre base de datos de generalización

6 Conclusiones y trabajo futuro

6.1 Conclusiones

En este trabajo se han realizado una serie de detectores con el objetivo de analizar y hallar diferentes irregularidades en señales resultantes de la medición de piezas industriales mediante sensores. Para lograr este objetivo se han propuesto 3 sistemas diferentes y, tras la realización de diversas pruebas sobre ellos y analizar los resultados obtenidos, se ha llegado a las siguientes conclusiones:

- La distinción entre la presencia de eventos o no eventos sin discernir el tipo de estos es correcta dando buenos resultados en todos los detectores.
- La distinción entre eventos concretos es más compleja y los resultados obtenidos en esta tarea varían según el detector. Siendo mejores en el detector espectral que, en el correlador, debido a las grandes limitaciones de este último en la detección de eventos como los *gaps*.
- Las bases de datos obtenidas, si bien son limitadas parecen suficiente como para realizar unas medidas de rendimiento moderadamente robustas de los sistemas.
- Las componentes frecuenciales y los coeficientes delta permiten caracterizar correctamente cada evento dando lugar a una distinción entre ellos clara.
- El uso de un *backend* sobre los detectores aporta una gran robustez al sistema y permite mayor flexibilidad de este a la hora de tratar con señales que difieren mucho de las de entrenamiento.
- El detector fusión da resultados prometedores, pero se comporta en ocasiones de forma impredecible.
- Pese a los buenos resultados obtenidos es necesario obtener tasas de error mucho menor de cara a la implantación de estos sistemas en procesos industriales reales.

6.2 Trabajo futuro

Como puntos a desarrollar o analizar en el futuro podemos destacar:

- Aplicación de sistemas de detección y clasificación más complejos, en orden de bajar más el error.
- Aumentar el tamaño de la base de datos como forma de hacer más robusto el sistema.
- Hacer pruebas de entrenamiento mixto con distintas bases.

Referencias

- [1]. **Schaumann, Rolf and Van Valkenburg, Mac E.** *Design og analog filters*. s.l. : Oxford University Press, 2010. ISBN-13: 978-0195373943.
- [2]. **Huang, Xuedong, Acero, Alex and Hon, Hsiao-Wuen.** *Spoken language processing: a guide to theory, algorithm, and system*. s.l. : Financial Times Prentice Hall, 2001. ISBN-13: 978-0130226167.
- [3]. **Theororidis, Sergios and Koutroumbas, Konstantinos.** *Pattern Recognition*. s.l. : Academic Press, 2008. ISBN: 9781597492720.
- [4]. **Labrador Serrano, Beltrán.** *Clasificación de eventos en señales temporales procedentes de señales industriales*. s.l. : Universidad Autónoma de Madrid, 2017.
- [5]. **Proakis, John G.** *Digital Communications*. 4ª. s.l. : Mc Graw Hill. pp. 234-238.

Anexo A

Etiquetas *Ground Truth*

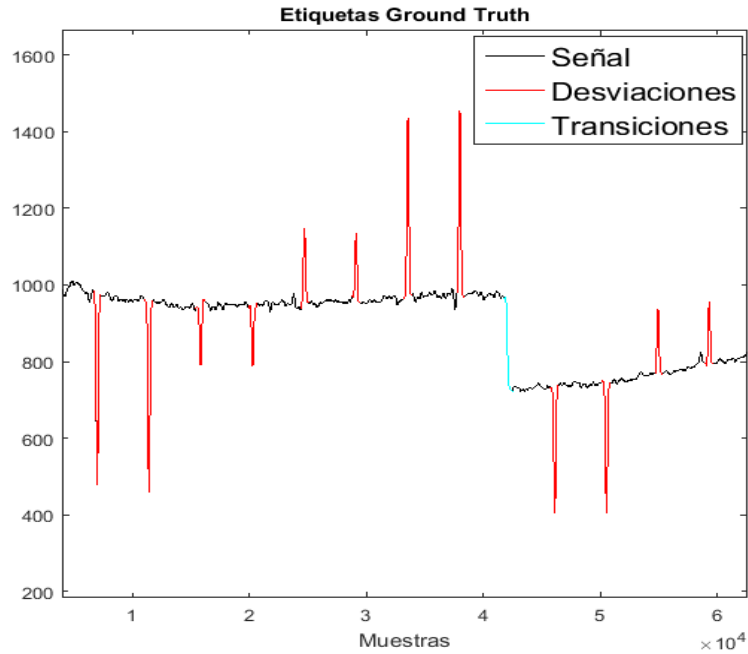


Figura 30: Etiquetas *Ground Truth* señal de la base de datos de generalización

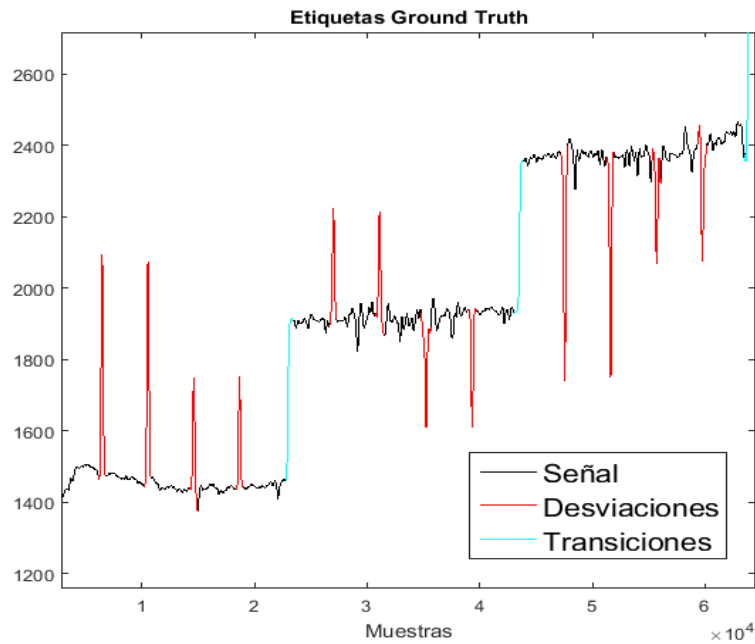


Figura 31: Etiquetas *Ground Truth* señal de la base de datos de generalización

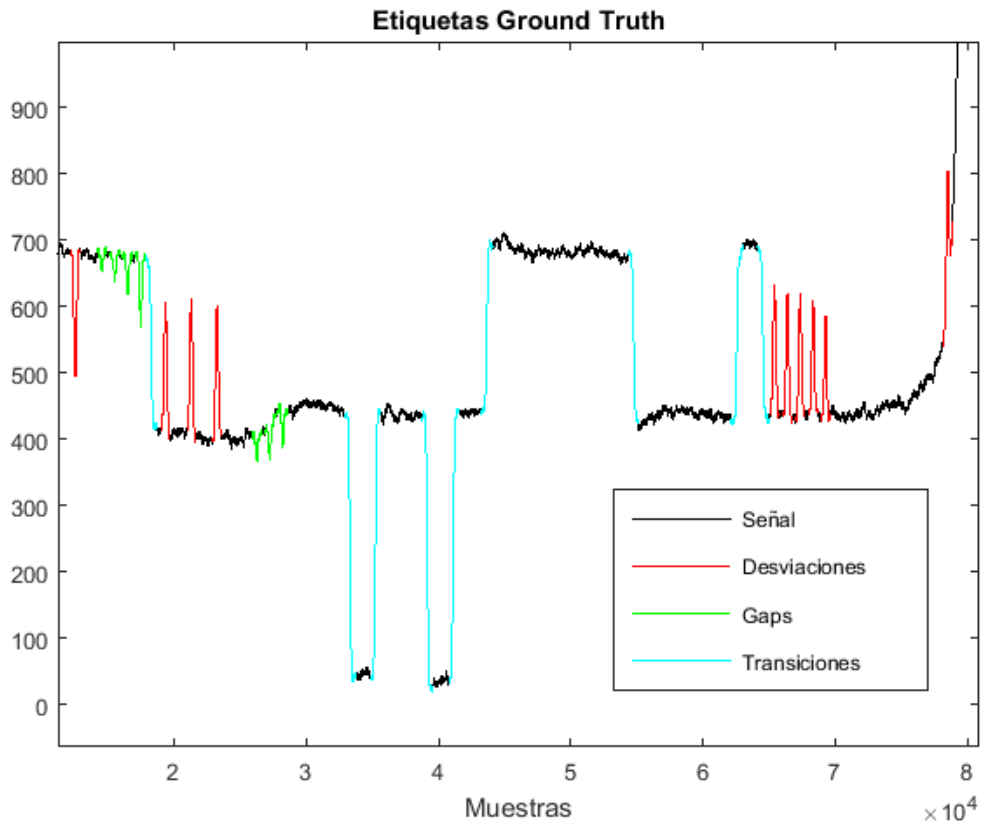


Figura 32: Etiquetas *Ground Truth* señal de la base de datos de desarrollo

Puntuaciones fusión de detectores

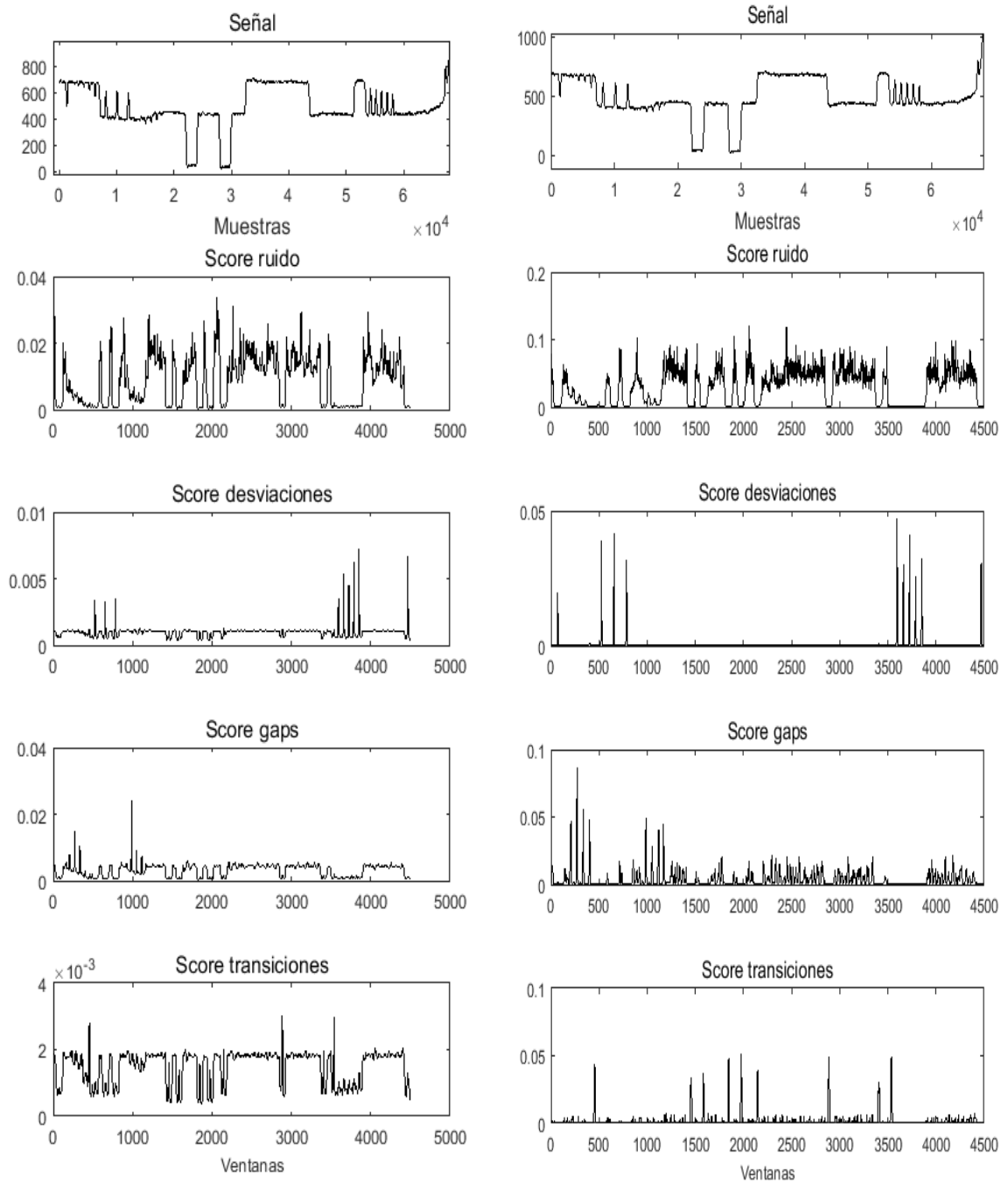


Figura 33: Scores obtenidos por detector fusión mediante distancia euclídea (izquierda) y distancia de Mahalanobis (derecha) sobre señal de la base de datos de desarrollo

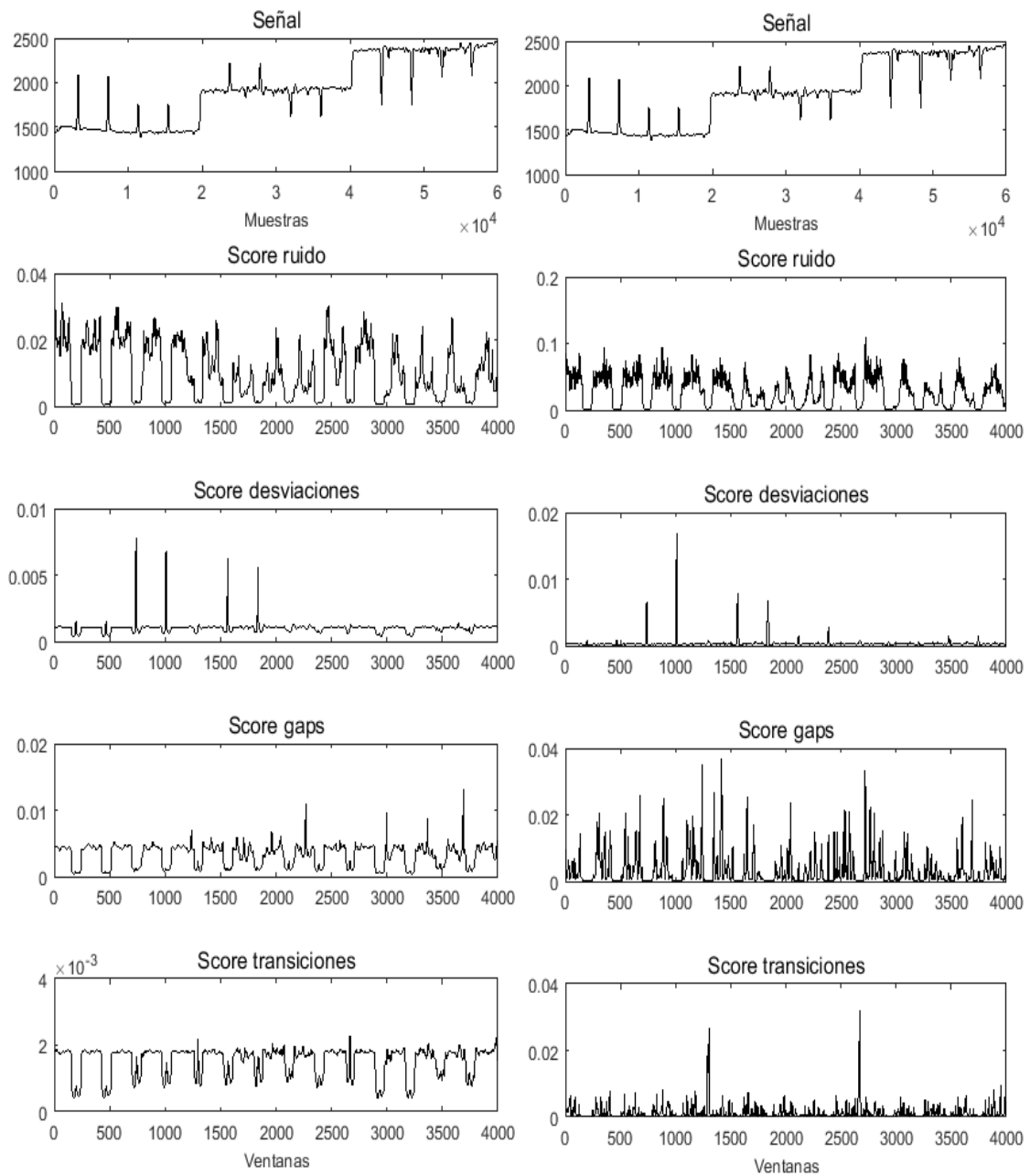


Figura 34: Scores obtenidos por el detector fusión mediante distancia euclídea (izquierda) y distancia de Mahalanobis (derecha) sobre señal de la base de datos de generalización