

Angel Carlos Herrero Lastra

Uso de Dispositivos de Bajo Coste en Entornos Industriales

Director/es

Martínez Domínguez, Francisco José
Sangüesa Escorihuela, Julio Alberto

<http://zaguan.unizar.es/collection/Tesis>



Universidad de Zaragoza
Servicio de Publicaciones

ISSN 2254-7606



Universidad
Zaragoza

Tesis Doctoral

USO DE DISPOSITIVOS DE BAJO COSTE EN ENTORNOS INDUSTRIALES

Autor

Angel Carlos Herrero Lastra

Director/es

Martínez Domínguez, Francisco José
Sangüesa Escorihuela, Julio Alberto

UNIVERSIDAD DE ZARAGOZA
Escuela de Doctorado

Programa de Doctorado en Ingeniería de Sistemas e Informática

2023

UNIVERSIDAD DE ZARAGOZA



DEPARTAMENTO DE INFORMÁTICA E INGENIERÍA
DE SISTEMAS

Uso de Dispositivos de Bajo Coste en Entornos Industriales

Tesis presentada para la obtención
del título de Doctor en Ingeniería de Sistemas e Informática

Ángel Carlos Herrero Lastra

Directores:

Dr. Francisco J. Martínez Domínguez

Dr. Julio A. Sangüesa Escorihuela

Teruel, septiembre 2023

A mi mujer, a mis hijos, a mis padres y a mi hermana, por todo.

Agradecimientos

Mi agradecimiento al Dr. Julio Sangüesa y al Dr. Francisco J. Martínez por su apoyo y supervisión en la realización de esta Tesis. Una carrera de fondo, que sin ninguna duda he conseguido gracias a su gestión, sabiduría y buen hacer. Han hecho que me sienta parte del equipo en todo momento, integrándome en el grupo Intelligent Networks and Information Technologies (iNiT) desde el primer día. Su orientación continua y valiosos consejos han contribuido a la calidad de este trabajo, enriqueciéndome en lo académico y en lo personal.

Al Dr. Carlos T. Calafate, por sus contribuciones llenas de luz en momentos en los que hacía falta una visión distinta.

A Luisa Marcos, directora ejecutiva de Marcos Conde, por dar todas las facilidades para operar en una industria de tal magnitud como es la suya.

Finalmente, me gustaría hacer un agradecimiento especial a mi familia. A Patricia, mi mujer, por su incommensurable amor, comprensión y sacrificio. A mis hijos Ángel y Marco, por las horas y dedicación robadas durante estos años. A mis padres y hermana, por la educación recibida, comprensión y amor. Sin todos ellos no hubiese sido posible haber llegado hasta aquí.

Ángel Carlos Herrero Lastra
Teruel, septiembre 2023

Resumen

La transición hacia la Industria 4.0 requiere una gran inversión en dispositivos y mecanismos que permitan la interconectividad entre las personas, máquinas y procesos. En esta Tesis se presentan varias aportaciones relacionadas directamente con la Industria 4.0 y el Internet de las Cosas (IoT). En primer lugar, se presenta un sistema de bajo coste, basado en Raspberry Pi, para la estimación en tiempo real del Overall Equipment Effectiveness (OEE), un índice que permite a los responsables de las empresas detectar qué aspectos deben tener en cuenta para mejorar la eficacia del proceso productivo.

La configuración del sistema permite la medición de las variables de eficiencia productiva de las máquinas termoselladoras en la industria, con un coste sensiblemente menor que otras soluciones propietarias. El sistema recoge y procesa algunas de las variables generadas; en concreto, las operaciones de sellado realizadas y la disponibilidad de las selladoras.

Por otra parte, dado que los dispositivos de bajo coste presentan una mayor predisposición a verse afectados por las interferencias electromagnéticas (EMIs), se proponen dos mecanismos de filtrado de EMIs, que permitan contemplar únicamente las señales válidas y así poder estimar el OEE de forma precisa. El primer mecanismo es el Data Base Filter (DBF), que se ha diseñado para recoger correctamente las señales de sellado. Este mecanismo opera sobre la base de datos, filtrando las señales erróneas que hayan sido insertadas. El segundo mecanismo, el Smart Coded Filter (SCF), se emplea para el filtrado de señales erróneas en las mediciones de disponibilidad de las máquinas.

Hemos validado el sistema propuesto y los mecanismos de filtrado diseñados, en un entorno real, es decir, en una línea de producción de una fábrica de elaborados lácteos, y los resultados obtenidos demuestran el correcto funcionamiento de nuestro sistema. En concreto, tras la eliminación de señales erróneas producidas por las interferencias electromagnéticas, mediante los mecanismos DBF y SCF, se presenta un error máximo del 0.93 %, un error promedio del 0.28 % y una raíz del error cuadrático medio (RMSE) de 4.26 en las señales de sellado, así como un error del 0 % en las señales de disponibilidad, lo que sin duda, permite realizar una estimación precisa del OEE.

Por último, se presenta el algoritmo Mode Binary Search (MoBiSea), diseñado para automatizar algunas operaciones necesarias para el correcto funcionamiento del mecanismo de filtrado DBF. Más concretamente, MoBiSea agrupa las señales de sellado y determina, sin la necesidad de intervención por parte de los operarios,

los instantes de inicio y fin de la producción de cada tipo de producto, así como el número de productos elaborados durante el proceso de fabricación.

Hemos comparado el funcionamiento de MoBiSea con tres mecanismos de agrupamiento, como el k-means, el x-means y el k-rms, obteniendo unos resultados significativamente mejores. En todos los casos analizados, el máximo error cometido por MoBiSea es de un 2.69 %, lo que mejora los resultados obtenidos por el resto de mecanismos.

En definitiva, el sistema de bajo coste presentado en esta Tesis, resulta efectivo, permite una estimación precisa del OEE, y por tanto, que los responsables puedan tomar decisiones adecuadas para la mejora en la producción. Además, permite reducir considerablemente los costes de implantación, comparado con los sistemas propietarios que presentan similares funcionalidades.

Abstract

The transition to Industry 4.0 requires a large investment in devices and mechanisms that enable interconnectivity between people, machines and processes. This Thesis presents several contributions directly related to Industry 4.0 and the Internet of Things (IoT). First, a low-cost system based on Raspberry Pi is presented for the real-time estimation of the Overall Equipment Effectiveness (OEE), a metric that allows company managers to detect which aspects they should consider to improve the effectiveness of the production process.

The system configuration allows the measurement of the production efficiency variables of heat-sealing machines in the industry, at a significantly lower cost than other proprietary solutions. The system collects and processes some of the variables generated; in particular, the sealing operations performed and the availability of the heat-sealing machines.

On the other hand, since low-cost devices are prone to being affected by electromagnetic interferences (EMIs), two EMI filtering mechanisms are proposed, which allow considering only valid signals and thus enabling an accurate OEE estimation. The first mechanism is the Data Base Filter (DBF), which has been designed to correctly collect the sealing signals. This mechanism operates on the basis of the database, filtering out the erroneous signals that have been inserted. The second mechanism, the Smart Coded Filter (SCF), is used for filtering out erroneous signals in machine availability measurements.

We have validated the proposed system and the designed filtering mechanisms in a real environment, i.e. in a production line of a dairy factory, and the results obtained demonstrate the correct operation of our system. Specifically, after the elimination of erroneous signals produced by electromagnetic interferences, by means of the DBF and SCF mechanisms, a maximum error of 0.93 %, an average error of 0.28 % and a root mean square error (RMSE) of 4.26 are presented in the sealing signals, as well as an error of 0 % in the availability signals, which undoubtedly allows an accurate estimation of the OEE.

Finally, the Mode Binary Search (MoBiSea) algorithm is presented, which was specifically designed to automate some operations required for the correct operation of the DBF filtering mechanism. More specifically, MoBiSea groups the sealing signals and determines, without the need for operator intervention, the start and end times of the production of each type of product, as well as the number of finished products during the manufacturing process.

We have compared the performance of MoBiSea with three clustering mecha-

nisms, i.e., k-means, x-means and k-rms, obtaining significantly better results. Out of all the cases we analyzed, the maximum error committed by Mode Binary Search is limited to 2.69 %, thereby outperforming all others.

Overall, the low-cost system presented in this Thesis is effective, allows accurate estimation of OEE, and therefore, allows business managers to make appropriate decisions for production improvement. In addition, it significantly reduces implementation costs compared to proprietary systems that have similar functionalities.

Índice general

1. Motivación, Objetivos, y Organización de la Tesis	1
1.1. Motivación	1
1.2. Objetivos de la Tesis	2
1.3. Organización de la Tesis	3
2. Industria 4.0: introducción, aspectos relevantes y consideraciones previas	4
2.1. Introducción	4
2.2. La evolución de la industria: Industria 4.0	7
2.3. Overall Equipment Effectiveness (OEE)	9
2.4. Dispositivos low-cost	10
2.4.1. Mercado de IoT	12
2.4.2. Raspberry Pi. Modelos, funcionalidad y características. . .	14
2.4.3. Uso de Raspberry Pi en la Industria 4.0	15
2.5. Interferencias Electromagnéticas (EMIs)	18
2.5.1. Compatibilidad electromagnética	18
2.5.2. Fuentes de EMIs y tipología	19
2.5.3. Eliminación de EMIs en dispositivos low-cost	24
2.6. Agrupamiento de datos y búsqueda binaria	26
2.6.1. k-means, x-means y k-rms	26
2.6.2. Uso de los algoritmos k-means, x-means y k-rms	28
2.6.3. Búsqueda Binaria	30
2.6.4. Uso del algoritmo Búsqueda Binaria	31
2.7. Conclusiones	32
3. Arquitectura low-cost para la medición del OEE en la Industria 4.0	34
3.1. Introducción	34
3.2. Arquitectura del sistema	35
3.2.1. Elementos del sistema	35
3.2.2. Diagrama de flujo del sistema	37
3.3. Tipos de señales	39
3.3.1. Señal de sellado	39
3.3.2. Señal de disponibilidad	40

3.3.3. Conexiones necesarias para la recogida de las señales de sellado y disponibilidad	42
3.4. Medición del OEE	47
3.5. Implantación del sistema	49
3.6. Conclusiones	54
4. Mecanismos de filtrado de EMIs: Data Base Filter y Smart Coded Filter	55
4.1. Introducción	55
4.2. Mecanismo DBF: Filtrado de EMIs en señales de sellado	56
4.2.1. Funcionamiento del mecanismo de filtrado DBF	56
4.2.2. Resultados del mecanismo de filtrado DBF	65
4.3. Mecanismo SCF: Filtrado de EMIs en señales de disponibilidad	88
4.3.1. Funcionamiento del mecanismo de filtrado SCF	88
4.3.2. Resultados del mecanismo de filtrado SCF	92
4.4. Funcionamiento conjunto de los mecanismos DBF y SCF	99
4.5. Conclusiones	102
5. Detección automática de productos: MoBiSea	104
5.1. Introducción	104
5.2. MoBiSea: Mode Binary Search	105
5.2.1. Ejemplo de funcionamiento de MoBiSea	108
5.2.2. Validación de MoBiSea	115
5.3. Comparativa con otros algoritmos de agrupamiento	119
5.4. Integración de MoBiSea en el mecanismo de filtrado DBF	127
5.5. Conclusiones	128
6. Conclusiones, Publicaciones y Trabajo futuro	130
6.1. Conclusiones	130
6.2. Publicaciones relacionadas con la Tesis	131
6.2.1. Revistas	131
6.2.2. Congresos internacionales	133
6.2.3. Congresos nacionales	135
6.3. Trabajo futuro	136

Lista de Figuras

2.1. Pilares estratégicos en la Industria 4.0 [Pod18].	5
2.2. Parámetros del OEE.	10
2.3. Porcentaje de empresas que utilizaron dispositivos IoT en España en 2021 [Ins22].	13
2.4. Previsión de ingresos por segmento en España [Sta22].	13
2.5. Previsión de ingresos en IoT por países para el año 2027 [Sta22].	14
2.6. Raspberry Pi 4B.	16
2.7. Elementos básicos de un problema de EMC [Pau06].	18
2.8. Formas de propagación de las EMIs [FA12].	20
2.9. EMIs conducidas modo diferencial [Sch18].	21
2.10. EMIs conducidas modo común [Sch18].	21
2.11. Acoplamiento capacitivo [FA12].	22
2.12. Acoplamiento inductivo [Sch18].	22
2.13. Acoplamiento por radiación [Sch18].	23
2.14. Búsqueda Binaria [Knu99].	30
3.1. Arquitectura del sistema.	35
3.2. Selladora industrial [ULM22].	36
3.3. Relé Omron MY4IN [Omr22].	36
3.4. Raspberry Pi 3 [Ras22].	37
3.5. Diagrama de flujo del sistema propuesto.	38
3.6. Señales de sellado y EMIs en un turno de ejemplo.	40
3.7. Señales de disponibilidad y EMIs en un turno de ejemplo.	41
3.8. Armario de componentes electromecánicos.	42
3.9. Esquema de conexiones para la obtención de señal de sellado.	43
3.10. Vista en planta de la selladora y señales recogidas por la máquina.	44
3.11. Esquema de conexiones para la obtención de señal de disponibilidad.	45
3.12. Cableado de los relés con la Raspberry Pi y la selladora industrial.	46
3.13. Ejemplo de cálculo de los diferentes parámetros del OEE.	50
3.14. Gráfica de Eficiencia Global de la Máquina (OEE).	50
3.15. Línea de sellado termoplástico: (a) perfil de la selladora, (b) operarios cargando la selladora con cuñas de queso y (c) detalle de la colocación del plástico en la selladora.	52

3.16. Detalles del sistema instalado: (a) relés Omron para recoger las señales de disponibilidad y sellado, (b) Raspberry Pi funcionando, junto con la cámara de grabación del comportamiento de las señales y (c) Raspberry Pi encapsulada en el cuadro eléctrico y maletín de monitorización de la lectura de las señales.	53
4.1. Ejemplo de la determinación de la moda en un turno.	58
4.2. Error cometido durante el proceso de filtrado al variar el offset, para todos los productos y turnos analizados.	63
4.3. RMSE: (a) sin offset (RAW) y (b) offset elegido (13.04%).	64
4.4. Determinación de la moda para el producto A en el turno 1.	68
4.5. Determinación de la moda: (a) producto A, (b) producto B y (c) producto C en el turno 2.	69
4.6. Determinación de la moda: (a) producto A, (b) producto B y (c) producto C en el turno 3.	70
4.7. Determinación de la moda: (a) producto A y (b) producto B en el turno 4.	71
4.8. Determinación de la moda: (a) producto A, (b) producto B y (c) producto C en el turno 5.	72
4.9. Determinación de la moda: (a) producto A, (b) producto B y (c) producto C en el turno 6.	74
4.10. Señales de sellado: (a) RAW y (b) filtradas por el DBF en el turno 1.	75
4.11. Señales de sellado: (a) RAW y (b) filtradas por el DBF en el turno 2.	76
4.12. Señales de sellado: (a) RAW y (b) filtradas por el DBF en el turno 3.	77
4.13. Señales de sellado: (a) RAW y (b) filtradas por el DBF en el turno 4.	78
4.14. Señales de sellado: (a) RAW y (b) filtradas por el DBF en el turno 5.	79
4.15. Señales de sellado: (a) RAW y (b) filtradas por el DBF en el turno 6.	80
4.16. Señales de sellado RAW en el turno 1.	81
4.17. Señales de sellado RAW en el turno 2, agrupadas por producto.	82
4.18. Señales de sellado RAW en el turno 3, agrupadas por producto.	82
4.19. Señales de sellado RAW en el turno 4, agrupadas por producto.	83
4.20. Señales de sellado RAW en el turno 5, agrupadas por producto.	83
4.21. Señales de sellado RAW en el turno 6, agrupadas por producto.	84
4.22. Señales de sellado en DBF en el turno 1: cuartos de queso (Producto A).	85
4.23. Señales de sellado en DBF en el turno 2: (a) cuartos de queso (Producto A), (b) quesos enteros (Producto B), y cuñas de queso (Producto C).	85
4.24. Señales de sellado en DBF en el turno 3: (a) quesos enteros (Producto B), (b) cuñas de queso (Producto C), y cuartos de queso (Producto A).	86
4.25. Señales de sellado en DBF en el turno 4: (a) cuartos de queso (Producto A) y (b) quesos enteros (Producto B).	86
4.26. Señales de sellado en DBF en el turno 5: (a) cuartos de queso (Producto A), (b) quesos enteros (Producto B), y cuñas de queso (Producto C).	87

LISTA DE FIGURAS

4.27. Señales de sellado en DBF en el turno 6: (a) cuñas de queso (Producto C), (b) quesos enteros (Producto B), y cuartos de queso (Producto A).	87
4.28. Señales de sellado erróneas recogidas durante todos los turnos. . .	88
4.29. Diagrama de flujo del funcionamiento del mecanismo SCF.	89
4.30. Disponibilidad en turno 1: (a) Señales RAW y (b) Señales filtradas por SCF.	94
4.31. Disponibilidad en turno 2: (a) Señales RAW y (b) Señales filtradas por SCF.	95
4.32. Disponibilidad en turno 3: (a) Señales RAW y (b) Señales filtradas por SCF.	95
4.33. Disponibilidad en turno 4: (a) Señales RAW y (b) Señales filtradas por SCF.	96
4.34. Disponibilidad en turno 5: (a) Señales RAW y (b) Señales filtradas por SCF.	96
4.35. Disponibilidad en turno 6: (a) Señales RAW y (b) Señales filtradas por SCF.	97
4.36. Disponibilidad con datos RAW y con filtrado SCF en todos los turnos analizados.	98
4.37. Comparación de sellados y disponibilidad en el turno 1 durante dos horas de producción: (a) Señales RAW y (b) Señales filtradas por DBF y SCF.	100
4.38. Comparación de sellados y disponibilidad en el turno 1 durante dos horas de producción: (a) Señales RAW y (b) Señales filtradas por DBF y SCF.	101
5.1. Funcionamiento de MoBiSea.	109
5.2. Modas de los tiempos de sellado agrupados en el turno 5 (isize = 50).	111
5.3. Detalle de modas de los tiempos de sellado (señales agrupadas 701-800) en el turno 5, con isize = 25.	111
5.4. Detalle de modas de los tiempos de sellado (señales agrupadas 751-800) en el turno 5, con isize = 12.	112
5.5. Detalle de modas de los tiempos de sellado (señales agrupadas 763-787) en el turno 5, con isize = 6.	113
5.6. Detalle de modas de los tiempos de sellado (señales agrupadas 776-787) en el turno 5, con isize = 3.	113
5.7. Detalle de modas de los tiempos de sellado (señales agrupadas 776-781) en el turno 5, con isize = 2.	114
5.8. Detalle de modas de los tiempos de sellado (señales 778 y 779) en el turno 5, con isize = 1.	115
5.9. Ejemplo de todas las señales recopiladas durante el turno 5.	117
5.10. Detalle del análisis del cambio de producto.	118
5.11. Productos estimados por x-means, k-means, k-rms y MoBiSea en el turno 1.	124
5.12. Productos estimados por x-means, k-means, k-rms y MoBiSea en el turno 2.	124

LISTA DE FIGURAS

5.13. Productos estimados por x-means, k-means, k-rms y MoBiSea en el turno 3.	125
5.14. Productos estimados por x-means, k-means, k-rms y MoBiSea en el turno 4.	126
5.15. Productos estimados por x-means, k-means, k-rms y MoBiSea en el turno 5.	126
5.16. Productos estimados por x-means, k-means, k-rms y MoBiSea en el turno 6.	127
5.17. Esquema de integración de MoBiSea en DBF.	128

Lista de Tablas

2.1. Evolución de la industria [PAMBEU20]	8
2.2. Diferentes modelos de la Raspberry Pi	15
2.3. Comparación de las Raspberry Pi 4B y Zero W y el Arduino Uno [Jol21]	16
3.1. Nomenclatura de las señales de guardas, emergencias, marcha y paro de la selladora industrial	44
3.2. Ejemplo de datos de sellado	48
3.3. Ejemplo de datos de disponibilidad	48
3.4. Datos para el cálculo del OEE	49
4.1. Ejemplo de valores recogidos que permiten determinar el offset necesario para tener un error del 0 % para el producto C en el turno 5	60
4.2. Cálculo del offset para valores de error 0 %, <0.5 %, <1 % y <5 % .	61
4.3. Resultados de DBF con diferentes valores de offset	62
4.4. Señales recogidas por el mecanismo DBF y errores cometidos con un offset del 13.04 %	66
4.5. Modas calculadas para cada turno y producto	67
4.6. Resultados de señales de disponibilidad obtenidos en los seis turnos analizados	93
5.1. Extracto de identificación de señales, timestamp y tiempos de sellado en el turno 5	110
5.2. Grupos analizados con isize = 25	111
5.3. Grupos analizados con isize = 12	112
5.4. Grupos analizados con isize = 6	113
5.5. Grupos analizados con isize = 3	114
5.6. Grupos analizados con isize = 2	114
5.7. Grupos analizados con isize = 1	114
5.8. Clústeres determinados por MoBiSea al variar el tamaño del intervalo (25, 50, 100, 300 y 500) y el threshold (0.10, 0.05 y 0.01). Los valores incorrectos aparecen en rojo	115
5.9. Tiempos de ejecución de MoBiSea al variar el tamaño del intervalo	116
5.10. Comparativa entre los valores reales (Clusters / Center) y los resultados obtenidos por x-means, k-means, k-rms y MobiSea.	120

LISTA DE TABLAS

- 5.11. Comparativa entre los valores reales (Number of signals (error)) y los resultados obtenidos por x-means, k-means, k-rms y MobiSea. . 121
- 5.12. Comparativa entre los valores reales (Start / Finish) y los resultados obtenidos por x-means, k-means, k-rms y MobiSea. 122

Lista de Algoritmos

1.	Búsqueda Binaria	31
2.	Data Base Filter (DBF)	57
3.	Filter Logic Signals	57
4.	Filter Candidate Signals	58
5.	GPIO Sender	91
6.	GPIO Listener	91
7.	Inactive Machine Checker	92
8.	Mode Binary Search (MoBiSea)	106

Capítulo 1

Motivación, Objetivos, y Organización de la Tesis

1.1. Motivación

La aparición de nuevas tecnologías como la Inteligencia Artificial (IA), el aprendizaje automático y la conectividad a Internet de gran cantidad de dispositivos, ha suscitado un profundo cambio en la sociedad y en su modelo económico. Estos cambios han repercutido de forma favorable en la industria, en la cual se ha modificado la interacción entre el ser humano y las máquinas, derivando en una cuarta revolución industrial.

La automatización en las tareas, la reducción de costes, la gestión sistemática, el cumplimiento de estándares y la eficiencia de recursos, son factores importantes en la industria, y los avances que la Industria 4.0 está introduciendo va a permitir mejorarlos. Sin embargo, a la hora de implantar nuevos procesos y funcionalidades relacionadas con la Industria Conectada, surgen problemas, especialmente cuando se consideran las pequeñas y medianas empresas, que suelen presentar limitaciones de capital, de habilidades, de conocimientos y tecnológicas [NY18]. En estos casos, la adopción de medidas de bajo coste (low-cost), más asequibles, va a facilitar y acelerar la transición hacia la Industria 4.0.

La Tesis Doctoral que se presenta, propone la utilización de sistemas capaces de realizar las mismas operaciones que otros sistemas propietarios de reconocido prestigio, pero realizando una menor inversión económica. Para ello, se pretende implantar en los procesos industriales, dispositivos low-cost tales como la Raspberry Pi, que permiten por sí solos, o combinados con otros periféricos, la monitorización y el control de variables, el análisis de datos y, en definitiva, conectar el estado de las máquinas, para el análisis en tiempo real del proceso de fabricación. Más concretamente, se pretende realizar la medición del Overall Equipment Effectiveness (OEE), un índice que permite a los responsables de las empresas detectar qué aspectos deben analizar en detalle para poder mejorar la eficacia del proceso productivo y así incrementar la rentabilidad.

Sin embargo, estos dispositivos electrónicos de bajo coste, presentan algunos

inconvenientes relacionados con el entorno electromagnético en el que suelen estar ubicados, normalmente cuadros eléctricos encapsulados en los que conviven con otros automatismos, fuentes de alimentación, y otros elementos generadores de Interferencias Electromagnéticas (EMIs). Las EMIs suelen aparecer cuando los dispositivos electrónicos trabajan próximos unos de otros y a frecuencias similares, teniendo esto un impacto claro en el rendimiento y confiabilidad de los equipos electrónicos [YSY⁺10] [TJP⁺13]. La compatibilidad electromagnética constituye un grave problema, ya que este fenómeno físico puede afectar al correcto funcionamiento de este tipo de instalaciones y sistemas, causando errores y fallos que pueden conducir a consecuencias fatales. Como veremos, la Raspberry Pi es un sistema que contiene elementos electrónicos sensibles a las interferencias electromagnéticas [MKV⁺17], y por ello, se presentan dos mecanismos de filtrado de EMIs diseñados para recoger de forma precisa las señales de sellado y de disponibilidad de las máquinas termoselladoras.

Finalmente, se propone un algoritmo de automatización capaz de determinar de manera no supervisada, es decir, sin la necesidad de intervención por parte de los operarios, los instantes de inicio y fin de la producción de cada producto elaborado. Dicho algoritmo, por tanto, determina los instantes en los que se producen los cambios de productos elaborados, con el objetivo de automatizar completamente uno de los mecanismos de filtrado de EMIs que se han diseñado.

1.2. Objetivos de la Tesis

El objetivo principal de esta Tesis es presentar un sistema de bajo coste basado en la Raspberry Pi, diseñado para la monitorización de las variables que intervienen en el cálculo del OEE, de forma que permita la mejora de los parámetros de producción, como son la disponibilidad, el rendimiento y la calidad. Que el sistema propuesto esté basado en dispositivos de bajo coste, pretende que sea fácilmente escalable, y que cubra las necesidades de los diferentes entornos industriales, incluidas las pequeñas y medianas empresas.

Otro objetivo importante es evitar el efecto negativo de las EMIs en el cálculo del OEE, filtrando las señales erróneas debidas a las interferencias electromagnéticas. Para ello, se han diseñado e implementado dos mecanismos de filtrado, el Data Base Filter (DBF) y el Smart Coded Filter (SCF).

La corrección de estas interferencias electromagnéticas se suele realizar mediante elementos hardware que eliminan las señales anómalas generadas, utilizando elementos físicos y circuitos de filtrado. Sin embargo, los mecanismos que se presentan en esta Tesis se basan en el filtrado software, permitiendo el filtrado óptimo de las EMIs sin importar su procedencia, eliminando con ello el problema, sin que sea necesaria la instalación de diferentes tipos de hardware en función del tipo de interferencia. El sistema diseñado puede ser instalado en entornos de todo tipo, mostrando gran flexibilidad con respecto al lugar que ocupe su instalación y a los niveles de ruido electromagnético a los que esté expuesto.

El tercer objetivo es identificar los tipos de producto que se elaboran en cada turno, así como los tiempos de inicio y fin del procesado de cada uno. Todo ello se realiza a través de la automatización del proceso, mediante la aplicación de

un algoritmo de agrupamiento diseñado, al que se ha denominado Mode Binary Search (MoBiSea).

Como cuarto y último objetivo, se pretende validar el correcto funcionamiento del sistema propuesto desplegándolo en una fábrica del sector alimentario, más concretamente, en una industria láctea en la que se ha incorporado el sistema a las líneas dedicadas al sellado termoplástico de los envases de producto.

1.3. Organización de la Tesis

La Tesis está estructurada de la siguiente manera: en el Capítulo 2 se hace una introducción a la Industria 4.0 y al factor OEE, así como a los dispositivos low-cost que intervienen en el Internet of Things (IoT). Se aporta una visión general de la situación del mercado y futuro del IoT, se presenta la Raspberry Pi con sus modelos y características, se analizan los trabajos relacionados con los dispositivos low-cost en la Industria 4.0 y, además, se realiza una revisión conceptual de las interferencias electromagnéticas y la compatibilidad electromagnética, exponiendo a su vez algunos trabajos relacionados con la eliminación de estos defectos en sistemas electrónicos de similares características al sistema propuesto en esta Tesis. Para finalizar este capítulo, se incluyen diferentes aspectos relacionados con el agrupamiento de datos (clusterización), presentando en detalle los algoritmos k-means, x-means, k-rms y la búsqueda binaria, así como algunas de las propuestas más relevantes, basadas en estos algoritmos.

En el Capítulo 3 se describe y presenta la arquitectura del sistema propuesto, incluyendo los elementos que intervienen en el sistema desarrollado, el diagrama de flujo de su funcionamiento, la naturaleza de las señales, las conexiones entre la máquina y el sistema, así como la procedencia de las señales que se transmiten al dispositivo low-cost. Por último, se incluye cómo se realiza el proceso de medición del OEE y se explica el proceso de despliegue del sistema.

Los mecanismos de filtrado de interferencias electromagnéticas Data Base Filter (DBF) y Smart Coded Filter (SCF) se presentan en el Capítulo 4. Más en detalle, se describen los algoritmos diseñados para cada uno de estos mecanismos, así como los resultados obtenidos, en términos de precisión, a la hora de estimar el número de productos elaborados y la disponibilidad de las máquinas termoselladoras.

En el Capítulo 5 se presenta el algoritmo MoBiSea, diseñado para detectar automáticamente el producto que se está elaborando en cada momento, así como el tiempo que se dedica al procesamiento de ese producto en ese turno, es decir, el inicio y final de la producción del mismo. En concreto, se describe su funcionamiento, se comparan los resultados obtenidos con otros algoritmos de agrupamiento y se explica cómo quedaría integrado MoBiSea en el mecanismo DBF.

Finalmente, el Capítulo 6, presenta un resumen de los principales resultados y contribuciones de esta Tesis, junto con algunas observaciones finales. También se incluye una lista de las publicaciones relacionadas con la Tesis, y se comentan brevemente algunos aspectos que pueden derivarse del trabajo aquí presentado, y que son considerados de interés para ser tratados en el futuro.

Capítulo 2

Industria 4.0: introducción, aspectos relevantes y consideraciones previas

La Industria 4.0 pretende mejorar la eficiencia y la eficacia de las industrias. Para ello, precisa de la intervención de diferentes factores, elementos y técnicas. En este capítulo, se hace una introducción a la Industria 4.0, así como al factor OEE y otros aspectos relevantes, como la incorporación de los dispositivos low-cost como elementos del IoT. Se analiza el mercado del IoT y el pronóstico en la implantación de estos sistemas, se presenta el dispositivo Raspberry Pi y se analizan algunos trabajos que utilizan dispositivos low-cost en la industria. Además, se presenta un resumen conceptual de las interferencias electromagnéticas y sus formas de propagación, realizando un repaso de algunos trabajos relacionados con la eliminación de EMIs en sistemas de similares características a los de nuestra propuesta. Para finalizar, se analizan los algoritmos k-means, x-means, k-rms y búsqueda binaria, y se presentan algunas propuestas relevantes basadas en éstos, con el objetivo de conocer el estado de la cuestión.

2.1. Introducción

El desarrollo actual de la industria apunta hacia una estructura más flexible y modular, capaz de adaptarse a las necesidades cambiantes del mercado. Para satisfacer estas necesidades, han aparecido algunos nuevos paradigmas relacionados con el entorno industrial, como es el caso de lo que conocemos como la Industria 4.0. Además, debido al desarrollo de la Internet Industrial de las Cosas (IIoT), ésta ha sufrido un crecimiento exponencial en los últimos años [GNZ⁺18].

El concepto de Industria 4.0 es relativamente reciente y se refiere a la cuarta revolución industrial, que consiste en la introducción de las tecnologías digitales en la industria. Éstas permiten que los dispositivos y sistemas electrónicos colaboren entre ellos, permitiendo modificar los productos, los procesos y los modelos de ne-



Figura 2.1: Pilares estratégicos en la Industria 4.0 [Pod18].

gocio [Gob15]. La Figura 2.1 muestra los elementos fundamentales que intervienen en el concepto de Industria 4.0. Así, se distinguen:

- Los Robots, que pasan de ser máquinas de automatización a transformarse en robots colaborativos inteligentes. Son capaces de obtener información de su entorno y de otras máquinas implicadas en el proceso, para tomar decisiones en función de la información recopilada.
- El Big Data, que permite analizar grandes cantidades de datos que son utilizados para dotar de un mayor rendimiento y mejorar la eficiencia de los sistemas.
- La Simulación, que permite anticipar y perfeccionar el correcto funcionamiento de los procesos.
- Realidad Aumentada, que permite al usuario visualizar procesos y procedimientos paso a paso de forma guiada, así como recibir instrucciones de expertos, de forma visual y en tiempo real, mediante sistemas de telegestión.
- El Cloud Computing, que permite tener disponibles servicios de computación a través de Internet, en el momento en el que éstos sean necesarios, sin que las empresas tengan que adquirir o gestionar la infraestructura necesaria.
- La ciberseguridad, elemento clave para la protección de sistemas y datos, ante los ataques y fallos de seguridad.

CAPÍTULO 2. INDUSTRIA 4.0: INTRODUCCIÓN, ASPECTOS RELEVANTES Y CONSIDERACIONES PREVIAS

- Sistema Integración, que permite una experiencia integral en la gestión, automatizando completamente las cadenas de valor.
- El Internet de las Cosas, capaz de realizar la conexión entre los objetos físicos y los dispositivos electrónicos para el intercambio de datos entre ellos.

Gracias a la conexión entre las operaciones físicas y los sistemas informáticos, los procesos de fabricación se transforman, personalizan y se orientan al consumidor final. Esto permite la fabricación en serie con un menor número de unidades por producto y disminuir los tiempos y el número de operaciones a realizar, facilitando los cambios de producción de un producto concreto a otro, de diferentes características, en tiempos mínimos. De esta forma, se puede pasar a una fabricación con altos niveles de customización.

Para lograr un control total del proceso, los productos son marcados con códigos que incluyen su trazabilidad, lo que facilita la detección y corrección de fallos en el modelo productivo. Con ello, se eliminan los defectos allí donde se generan, garantizando la calidad, el conocimiento de todos los pasos realizados para la consecución final del producto y la información necesaria para intervenir en tiempo real si fuese preciso, así como proveer al consumidor de toda la información del producto terminado.

Otro de los factores a destacar de la Industria 4.0, es la mejora alcanzada en la operatividad de las plantas industriales a través de la conversión del mantenimiento preventivo y correctivo en mantenimiento predictivo. Este logro es posible a partir de la adquisición de datos tomados de multitud de sensores, midiendo, procesando y recogiendo las variables que generan éstos, a través de técnicas basadas en *Big Data* y *Machine Learning*. La combinación de la Inteligencia Artificial y el tratamiento de los datos obtenidos permite conocer en todo momento la situación y estado de las máquinas empleadas en los diferentes procesos industriales auditados, para con ello, evitar fallos, programar paradas y reducir de manera considerable las pérdidas a nivel global.

Por otra parte, los elementos de control, monitorización, sensores, así como la amplia tipología de dispositivos y equipos utilizados en la Industria Conectada se caracterizan, en su mayor parte, por ser sistemas exclusivos de los fabricantes, por el alto coste que conlleva su implantación, así como por la baja estandarización de protocolos de comunicación e interoperabilidad. Debido a estos inconvenientes, esta Tesis presta especial atención al uso de sistemas de bajo coste, que tienen funcionalidades similares a otros propietarios de reconocido prestigio, pero que presentan muchas otras ventajas, ya que su bajo coste facilita su despliegue en los entornos de producción. Este paradigma, todavía incipiente en el ámbito industrial, pretende dotar a los procesos industriales de dispositivos con las propiedades y funcionalidades demandadas por y para la digitalización de la industria, pero a un coste mucho menor.

En la tecnología de bajo coste se incluyen microcontroladores, sensores y otros dispositivos hardware, como Raspberry Pi o Arduino, que permiten por sí solos o combinados con otros periféricos complementarios, la monitorización y el control de variables, el análisis de datos y actuar como servidores; en definitiva, conectar el estado de las máquinas para su análisis en tiempo real o diferido.

Mediante la introducción del IoT, en las áreas en las que se utilizaban las Tecnologías de la Información y Comunicación (TIC) de forma tradicional, se ha visto mejorar su alcance y consolidar su uso, al abaratar los dispositivos, mejorar la conectividad y almacenar los datos de forma abierta. Sin embargo, los sistemas de control y teledadida han sido tradicionalmente desarrollados en entornos cerrados y requieren una inversión económica elevada. En la actualidad, éstos pueden ser sustituidos por soluciones de bajo coste, con arquitecturas, protocolos de comunicación, y sistemas de representación de datos abiertos [COM⁺15].

En este capítulo, se presenta una visión general de los conceptos más importantes relacionados con las propuestas de la presente Tesis Doctoral. Más en detalle, en la Sección 2.2 se hace un repaso de cómo ha ido evolucionando la industria, hasta llegar a lo que conocemos como Industria 4.0. En la Sección 2.3 se explican los parámetros principales que se tienen en cuenta para el cálculo del OEE. La Sección 2.4 presenta el estado del mercado de IoT, la Raspberry Pi, así como el uso de dispositivos low-cost más conocidos y su uso en la industria. La Sección 2.5 introduce el concepto de interferencias electromagnéticas, sus fuentes generadoras, la tipología de transmisión, y se presenta un análisis sobre la eliminación de EMIs en dispositivos low-cost. La Sección 2.6 presenta los algoritmos k-means, x-means y k-rms para el agrupamiento de datos, así como la búsqueda binaria. Por último, en la Sección 2.7 se presentan las conclusiones de este capítulo.

2.2. La evolución de la industria: Industria 4.0

Desde el inicio de la industrialización, los saltos tecnológicos han propiciado cambios de paradigma denominados “revoluciones industriales”. Tal y como se observa en la Tabla 2.1, la primera revolución industrial se produjo en el campo de la mecanización. Con el uso de la energía eléctrica se produjo la segunda revolución industrial, y de la digitalización generalizada, automatización de procesos e informatización, la tercera revolución industrial.

El concepto de Industria 4.0 aparece por primera vez en Alemania [DBAF18] y es el nombre que se ha adoptado para denominar a una cuarta etapa de la evolución técnico-económica de la humanidad. Recibe este nombre como reminiscencia a las actualizaciones de software.

Sobre la base de una digitalización avanzada dentro de las fábricas, la interacción entre Internet y las tecnologías orientadas al futuro en el campo de los objetos inteligentes (máquinas y productos), es decir IoT, se ha producido un nuevo cambio de paradigma fundamental en la producción industrial.

La producción contiene sistemas de fabricación modulares y eficientes, y caracteriza escenarios en los que los productos controlan su propio proceso de fabricación. Por ello, la fabricación de productos individuales en un tamaño de lote de uno es posible, mientras se mantienen las condiciones económicas de la producción en masa [LFK⁺14].

Estos aspectos plantean retos y generan oportunidades en la industria, la cual está adaptando sus productos, modelos de negocio y los procesos, de cara a conseguir una adecuada digitalización de la sociedad y de la industria. El cliente tiene acceso inmediato a la oferta de empresas industriales de todo el mundo, debido a

CAPÍTULO 2. INDUSTRIA 4.0: INTRODUCCIÓN, ASPECTOS RELEVANTES Y CONSIDERACIONES PREVIAS

Tabla 2.1: Evolución de la industria [PAMBEU20]

Versión	Nombre	Innovación	Periodo y Características
1.0	1ª revolución industrial	Talleres mecánicos	Entre los siglos XVIII y XIX, apoyada en la introducción de equipos de producción mecánicos impulsados por agua y vapor
2.0	2ª revolución industrial	Líneas de producción transportadoras (1870)	En el siglo XIX, basada en la producción en masa, gracias al concepto de división de tareas y uso de energía eléctrica
3.0	3ª revolución industrial	Controladores lógicos programables (1969)	Final del siglo XX, debido al uso de la electrónica y la informática en los procesos, permite automatizar algunas tareas repetitivas en las líneas de producción
4.0	4ª revolución industrial	Basada en el uso de sistemas ciberfísicos y el Internet de las Cosas (actualidad)	Comenzó a principios del siglo XXI. Gracias a los avances tecnológicos y de Internet, la “cuarta revolución industrial” trae consigo las “fábricas inteligentes”. Fundamentada en el uso de sistemas de información, recolección de datos y el Internet de las Cosas

la hiperconectividad existente. El entorno es muy competitivo, pero a la vez con multitud de oportunidades para las empresas.

Los cambios que conlleva la Industria 4.0 permitirán un nuevo modelo de innovación colaborativa, de medios productivos conectados y flexibles, de clientes digitales y cadenas de suministro integradas.

En España, la estrategia Industria Conectada 4.0 responde a un triple objetivo:

1. Incrementar el valor añadido y el empleo cualificado en el sector industrial.
2. Favorecer el modelo de futuro para la industria española, con el fin de potenciar los sectores industriales de futuro de la economía española y aumentar su potencial de crecimiento, desarrollando a su vez la oferta local de soluciones digitales.
3. Desarrollar palancas competitivas diferenciales para favorecer la industria española e impulsar sus exportaciones.

Para alcanzar estos objetivos y garantizar una transformación digital eficaz, la Secretaría General de Industria y de la Pequeña y Mediana Empresa (SGIPYME) ha diseñado diferentes programas de apoyo a industrias [Min21]. Estos programas son los siguientes:

- HADA: Herramienta de Autodiagnóstico Digital Avanzada, que permite a las empresas conocer su madurez digital, a través de una aplicación online [Min22].
- ACTIVA Industria 4.0: consiste en un asesoramiento especializado, el cual permite a las empresas obtener un diagnóstico de situación y la elaboración de un plan de transformación hacia la implantación de los sistemas digitales necesarios.

2.3. OVERALL EQUIPMENT EFFECTIVENESS (OEE)

- **ACTIVA Financiación:** con este plan se pretende el apoyo a proyectos que promuevan la transformación digital en las industrias, así como su mejora en la sostenibilidad, resultado de las mejoras digitales.
- **ACTIVA StartUps:** es un programa que apoya las colaboraciones entre empresas emergentes y empresas consolidadas con potencial de innovación. Pretende impulsar la innovación y el crecimiento de empresas mediante entornos colaborativos y el impulso de la digitalización.
- **ACTIVA Ciberseguridad:** el objetivo de este programa es que las empresas determinen el nivel de seguridad actual y protejan sus sistemas, así como la información corporativa.
- **ACTIVA Crecimiento:** incluye el asesoramiento con consultores especializados para actuar en áreas como innovación, recursos humanos, operaciones, digitalización, marketing y comercialización y finanzas.

En definitiva, la Industria 4.0 genera innumerables ventajas al sector industrial y es un concepto que está en plena expansión. Va a significar un reto para las empresas, que deben adaptarse a los cambios con nuevas herramientas, apuestas firmes y constante apoyo al proceso de innovación.

2.3. Overall Equipment Effectiveness (OEE)

La herramienta de medición OEE se desarrolló a partir del concepto Total Productive Maintenance (TPM) propuesto por Nakajima [Nak88]. El objetivo del TPM era eliminar, en la medida de lo posible, todos los fallos provocados por los equipos, generando una mejora directa en la tasa de producción, la reducción de los costes e inventario, así como el aumento de la productividad laboral.

El OEE se define como una medida del rendimiento total de los equipos, es decir, el grado en que el equipamiento está haciendo lo que se supone que debe hacer [Wil06]. Esta métrica permite clasificar las principales razones por las que se produce un rendimiento deficiente y, por lo tanto, proporciona la base para establecer el análisis del origen de los defectos y prioridades de mejora.

La Figura 2.2 muestra los principales parámetros que se tienen en cuenta para el cálculo del OEE, y que son los siguientes:

- **Availability:** para medirla, se tienen en cuenta los siguientes tiempos: (i) Net Available Time (NAT), que se obtiene restando el Scheduled Downtime (SD) al Total Available Time (TAT), y (ii) Operating Time (OT), que es el tiempo que resulta de descontar del NAT, el tiempo perdido debido a Downtime Losses (DL).
- **Performance:** para su medida se tiene en cuenta: (i) el OT y (ii) el Net Operating Time (NOT), que se obtiene descontando el tiempo perdido debido a las Speed Losses (SL) al OT.
- **Quality:** para determinar este componente, se tendrá en cuenta: (i) el NOT, que es el tiempo del que realmente se dispone para producir, y el (ii) Fully

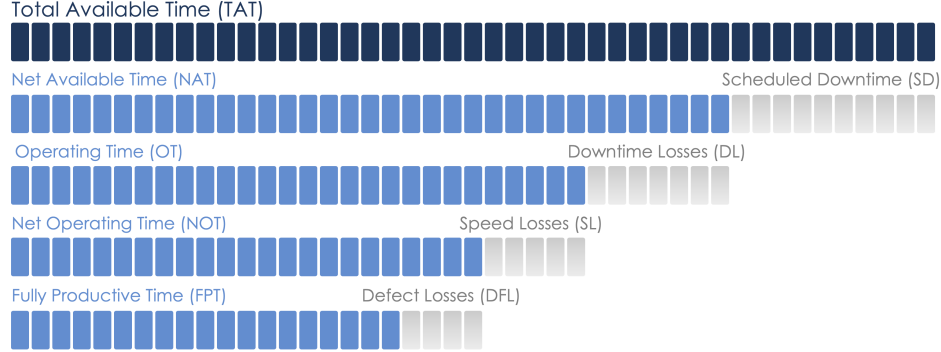


Figura 2.2: Parámetros del OEE.

Productive Time (FPT), que se obtiene restándole el tiempo perdido debido a Defect Losses (DFL) al NOT.

La Ecuación 2.1 muestra la fórmula de cálculo del OEE.

$$OEE = Availability \cdot Performance \cdot Quality \quad (2.1)$$

donde: $Availability = \frac{OT}{NAT}$, $Performance = \frac{NOT}{OT}$, y $Quality = \frac{FPT}{NOT}$.

Los resultados del OEE permiten la monitorización y comparación de las unidades de producción en la industria. Más en detalle, el OEE proporciona datos significativos que permiten aumentar la optimización de los recursos y reducir el tiempo de trabajo, por lo que se puede utilizar para aumentar el rendimiento del sistema y la mejora continua del trabajo. Además, puede revelar casos de alta demanda de trabajo y baja producción, lo que posibilitará adoptar una mejor actuación en la industria [FSAD19].

2.4. Dispositivos low-cost

A medida que el término IoT se ha hecho más común, el aumento en la cantidad de placas de desarrollo disponibles en el mercado se hace más notable. Estas placas varían en tamaño desde el tamaño de una moneda de un euro hasta el de una tarjeta de crédito. Aparte de los principales fabricantes de hardware como son Intel o IBM entre otros, muchas de estas placas son producidas por empresas financiadas por crowdfunding. La rentabilidad, la eficiencia energética, el bajo coste y la sencilla implementación, hacen de estos dispositivos candidatos más que adecuados para aplicaciones IoT [RIAI16].

Las aplicaciones de IoT se extienden en ámbitos que incluyen:

- Redes de actuadores y sensores inalámbricos
- Sistemas de monitorización de salud
- Hogares inteligentes

- Vehículos autónomos conectados
- Redes inteligentes automatizadas
- Industria 4.0

Durante los últimos años, se han diseñado diferentes tipos de dispositivos, teniendo en cuenta los requisitos de IoT. Estos incluyen Microcontroladores, System-on-Chips (SoC) y Computadoras de placa única (SBC). Estos dispositivos tienen diferentes cualidades, tales como:

- Múltiples pines para ser utilizados como GPIO (General Purpose Input/Output), I2C (Inter-Integrated Circuit), UART (Universal Asynchronous Receiver-Transmitter), SPI (Serial Port Interface), etc.
- ADC (Analog-to-Digital Converter) integrado
- Múltiples protocolos de comunicación
- Capacidades de red como BLE (Bluetooth Low Energy), Wi-Fi y Ethernet.
- Baja potencia de procesamiento
- Fácil implementación y bajo mantenimiento
- Compatibilidad con varios sistemas operativos
- Amplia compatibilidad con multitud de dispositivos
- Diseños de código abierto
- Administración remota
- Bajo coste

El coste de los dispositivos IoT ha registrado una caída significativa y, junto con las placas de desarrollo, han contribuido enormemente a la mejora de la tecnología IoT. Dicho coste viene relacionado por sus características computacionales integradas, de tal forma que cuanto más altas sean estas capacidades, tales como la memoria y funciones de procesamiento, mayor será su coste.

Los dispositivos IoT son generalmente de pequeño tamaño y bajo coste, lo que hace que la tecnología que los integra sea pequeña, resultando de gran importancia minimizar el área de silicio de estos dispositivos. Además, la cantidad de espacio requerido para la memoria, no sólo debe mantenerse al mínimo, sino que debe minimizarse cualquier coste adicional de procesamiento de las obleas de silicio a máscaras o pasos de procesamiento adicionales, puesto que cuanto más área de la oblea de silicio se requiera, mayor será el coste del dispositivo.

Los factores de forma del hardware de IoT (en dispositivos de IoT de gama baja, media y alta) se han reducido a niveles milimétricos con características interesantes para satisfacer las necesidades computacionales de las aplicaciones de IoT.

La ley de Moore, establece que aproximadamente cada dos años se duplica el número de transistores del microprocesador, es decir, cada dos años la densidad

de transistores en un chip será duplicada, por lo que los dispositivos aumentan sus prestaciones. Lo óptimo debería ser encontrar un equilibrio entre el tamaño y el rendimiento.

Además, las condiciones ambientales en las que se utilizan estos dispositivos juegan un papel importante en la apariencia física del diseño del dispositivo. Dado que la mayoría de los dispositivos de IoT se encuentran al aire libre o en entornos con condiciones adversas, es crucial equipar estos dispositivos con el nivel adecuado de protección a golpes, vibraciones, suciedad o salpicaduras [OGPS18].

2.4.1. Mercado de IoT

En esta subsección, se exponen algunos datos de interés relacionados con el sector y su previsión de crecimiento, con el objetivo de analizar y conocer las situación del mercado de IoT. Los aspectos positivos de la implantación del IoT son notables y además pueden incrementar los beneficios de cualquier organización, muy especialmente en la industria, claramente enfocada a la transformación digital.

En los próximos años, se espera un crecimiento exponencial del IoT. Se advierte la creación de un ecosistema en el que, tanto las aplicaciones como los servicios serán impulsados por los datos que hayan sido recopilados por los dispositivos, actuando como sensores y también capaces de interactuar con su entorno y con otros dispositivos, permitiendo así la conexión de millones de dispositivos.

El mercado de Internet de las Cosas es una de las tendencias tecnológicas más prometedoras de nuestro tiempo y sus posibilidades aumentan cada día. La tecnología IoT se encarga de las cuatro perspectivas de las tecnologías de la información: hardware, software, servicios y conectividad [Sta22].

Las aplicaciones van desde soluciones de hogar inteligente, hasta las soluciones de seguimiento logístico y la gestión energética. Abarcan sectores económicos como la sanidad, la educación, la agricultura, la energía, el transporte o la fabricación, entre otras. En España, el sector TIC es el que mayor uso da al IoT, seguido del sector de información y comunicaciones, y el de energía y agua. Sin embargo, existen otros sectores con menor utilización del IoT, como son el de construcción, actividades inmobiliarias y actividades administrativas y servicios auxiliares [Obs22]. En general, son las grandes empresas las que tienden a utilizar el IoT; el empleo de esta tecnología casi duplica a las pequeñas. Uno de los motivos por el que la gran mayoría de empresas que utilizan el IoT son grandes, podría ser el coste de los sistemas propietarios y, es por ello, que nuestra propuesta se basa en la tecnología low-cost, con el fin de acercar a las pequeñas y medianas empresas la Industria 4.0. La Figura 2.3 muestra el porcentaje de empresas, agrupadas por sector, que utilizaron dispositivos IoT en España durante el año 2021.

El crecimiento de los ingresos por IoT, en España, será tan notable, que se prevé un crecimiento, para los sectores de tecnologías para el hogar y tecnologías financieras de 96.58 millones de euros en 2023, llegando a ser de 140.19 millones en el año 2027 (ver Figura 2.4). El crecimiento en ingresos en IoT a nivel mundial para el 2027, lo encabezará China con 6846 millones de euros, seguido de Estados Unidos con 6288 millones de euros y México con 4055 millones. La Figura 2.5 muestra la comparativa de ingresos en IoT por países, previstos para el año 2027, destacando los cinco países que más ingresos obtendrán en ese año.

2.4. DISPOSITIVOS LOW-COST

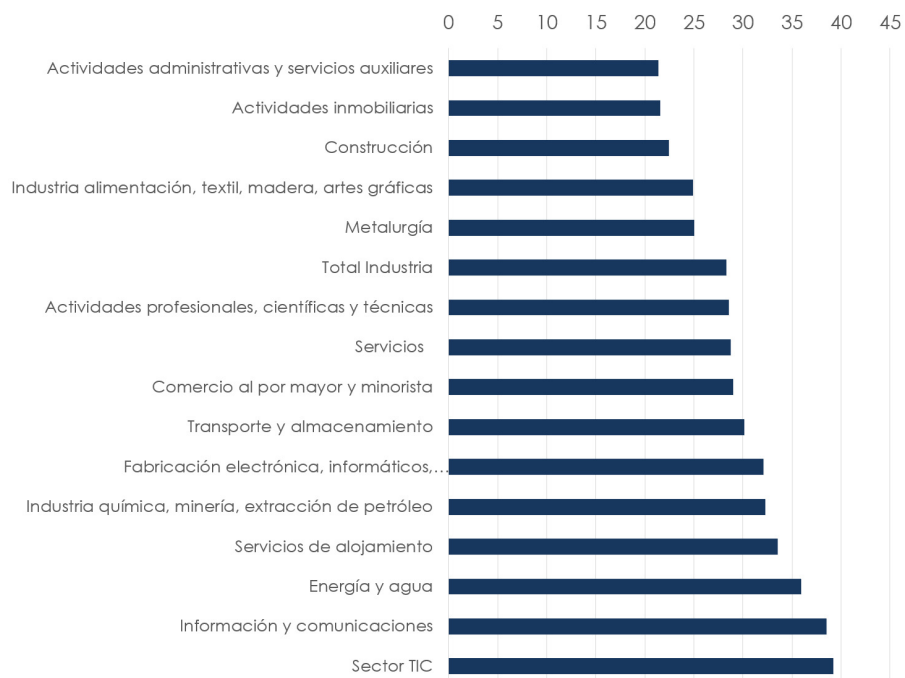


Figura 2.3: Porcentaje de empresas que utilizaron dispositivos IoT en España en 2021 [Ins22].

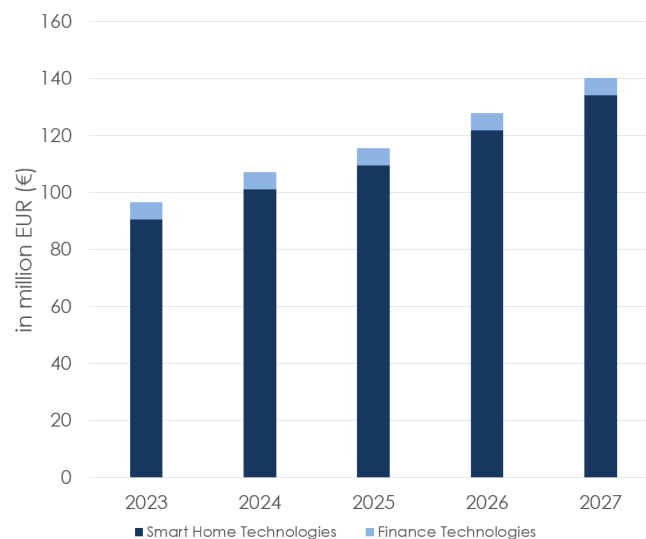


Figura 2.4: Previsión de ingresos por segmento en España [Sta22].

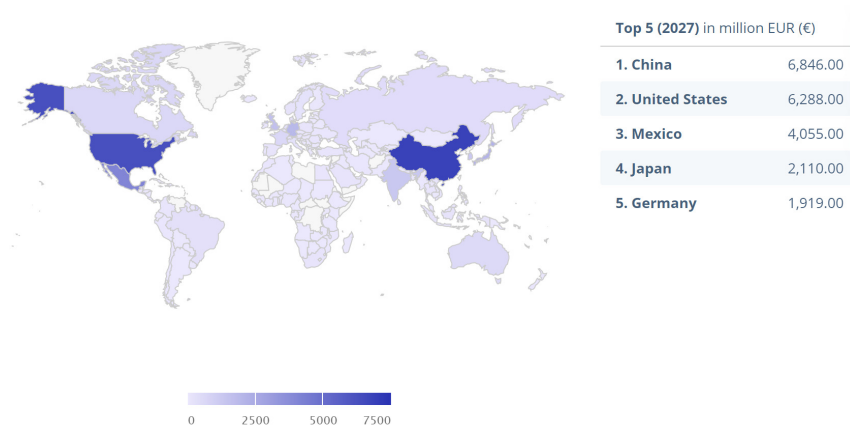


Figura 2.5: Previsión de ingresos en IoT por países para el año 2027 [Sta22].

2.4.2. Raspberry Pi. Modelos, funcionalidad y características.

A continuación, se explican los aspectos más destacados de la Raspberry Pi, sus modelos, algunas de sus funcionalidades y sus características principales.

La Raspberry Pi es un miniordenador potente y eficiente, con una dimensión aproximada al tamaño de una tarjeta de crédito y se considera un dispositivo low-cost de gama alta. Fue creada por la fundación Raspberry Pi (con sede en el Reino Unido), y su objetivo principal era su uso en la educación. Desde su lanzamiento, son innumerables las comunidades de código abierto que han contribuido con sistemas operativos de código libre y otras aplicaciones. Además es de gran aplicación al mundo académico e investigador, puesto que posee características que permiten su aplicación y desarrollo en infinidad de proyectos.

Raspberry Pi permanece en constante evolución, tanto en términos de software como de hardware, convirtiéndose así en una computadora que permite realizar múltiples tareas y resolver procesos en un tiempo determinado [GSS21].

La Tabla 2.2 compara las versiones de la Raspberry Pi en lo relativo a las versiones, tamaño, CPU, memoria, número de conectores USB, coste en el momento de aparición y año de lanzamiento.

Las bases de los modelos de Raspberry Pi son muy similares, cada uno con un sistema en un chip (SoC), que consta de una CPU integrada, una unidad de procesamiento de gráficos (GPU), una memoria RAM integrada y una entrada de alimentación de 5 V CC. Todos los modelos poseen un puerto para conectar una cámara dedicada, así como una serie de pines de entrada/salida de uso general (GPIO), que pueden usarse para la comunicación con una amplia gama de componentes electrónicos, relés de potencia, servomotores y una amplia gama de sensores (ver Figura 2.6).

Además, la mayoría de los modelos cuentan con conexión Ethernet y conectividad inalámbrica (Wi-Fi y Bluetooth) que, en combinación con los puertos GPIO, le aportan a la Raspberry Pi una gran versatilidad. De hecho, la Raspberry Pi tie-

Tabla 2.2: Diferentes modelos de la Raspberry Pi

Nombre	Tamaño (mm.)	CPU	RAM	Conectores USB	Coste en su lanzamiento (\$)	Año de lanzamiento
Pi 1	85.6 x 54	700MHz ARM1176JZF-S	256MB	1	20	2012
Pi 2	85 x 56	900MHz QUAD ARM Cortex-A7	1GB	4	35	2015
Pi 3B	85 x 56	1.2 GHz QUAD ARM Cortex-A53	1GB	4	35	2016
Pi 3B+	85 x 56	1.4 GHz QUAD ARM Cortex-A53	1GB	4	35	2018
Pi 4B	85 x 56	1.5 GHz QUAD ARM Cortex-A72	1, 2 o 4GB	2 (2.0) 2 (3.0)	40	2019
Zero	65 x 30	1 GHz ARM1176JZF-S	512 MB	1 Micro	5	2015
Zero W	65 x 30	1 GHz ARM1176JZF-S	512 MB	1 Micro	10	2017

ne todas las funcionalidades de una computadora estándar. Puede conectarse un ratón, teclado y pantalla, sin requerir configuración adicional, y tener el control sobre un entorno de escritorio Linux fácil de usar, así como otros sistemas operativos más populares, incluidos Windows 10 IoT y Android.

La Raspberry Pi puede utilizarse como una unidad sin necesidad de teclado y pantalla conectados y puede controlarse de forma remota para ejecutar scripts de forma autónoma, utilizando una amplia gama de lenguajes de programación [Jol21].

Por otra parte, la Raspberry Pi es un dispositivo multitarea, distinto a otros microcontroladores, como por ejemplo el Arduino, que únicamente se puede programar para ejecutar un solo programa escrito por el usuario. La Tabla 2.3 muestra las diferencias entre la Raspberry Pi y Arduino Uno, como por ejemplo, el sistema operativo, Ethernet de serie, conectividad Wi-Fi, la diferencia en cantidad de memoria, la capacidad de almacenamiento, el conector HDMI, o incluso el voltaje de estas placas.

2.4.3. Uso de Raspberry Pi en la Industria 4.0

Los dispositivos low-cost pueden aplicarse en entornos industriales propios de la Industria 4.0, proporcionando una mayor flexibilidad en la comunicación entre las máquinas y haciendo más efectiva la organización de la producción industrial, sin que esto conlleve costes muy elevados.

A la hora de abordar estos retos, y teniendo en cuenta las características de los sistemas propietarios, fundamentalmente en términos de coste y exclusividad, en la literatura podemos encontrar trabajos relativos al uso de sistemas basados en dispositivos de bajo coste, para ofrecer nuevas propuestas relacionadas con la Industria 4.0.

En cuanto a los dispositivos low-cost utilizados para la industria, una de estas propuestas es la presentada por Seguna et al. [STSS19], en la que los autores

CAPÍTULO 2. INDUSTRIA 4.0: INTRODUCCIÓN, ASPECTOS RELEVANTES Y CONSIDERACIONES PREVIAS

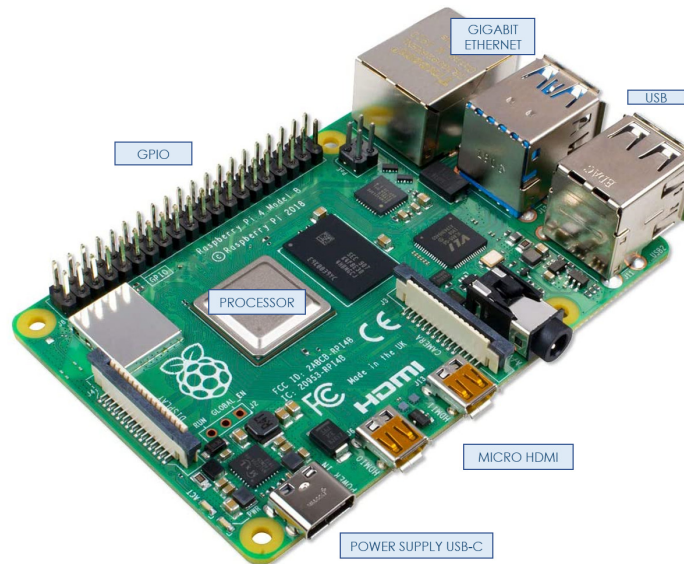


Figura 2.6: Raspberry Pi 4B.

Tabla 2.3: Comparación de las Raspberry Pi 4B y Zero W y el Arduino Uno [Jol21]

Spec	Raspberry Pi 4B / Zero W	Arduino Uno
Type	Single-board Computer	Microcontroller
Operating system	Multiple operating systems possible	None
Dimensions	85.6 mm × 56.6 mm / 65 × 30 mm	68.6 mm × 53.4 mm
Weight	46 g / 9 g	25 g
Price	40 (2 GB) – €85 (8 GB) / €12	€22
Multitasking	Yes	No
Setting up required	Yes	No
Processor	64-bit / 32-bit	8-bit
Memory	up to 8 GB / 512 MB	32 Kb
Clock speed	4× 1.5 GHz / 1 GHz	16 MHz
Ethernet	Gigabit / Adapter needed	No
Wi-Fi	Yes	No
Bluetooth	Yes	No
USB	2× USB 2 & 3 / micro-USB	No
Camera port	Yes	No
Audio port	Yes / No	No
HDMI	2× / 1× micro-HDMI	No
Input voltage	5 V	7–12 V
GPIO Ports	40 pins: 5 V, 3.3 V, Ground Digital I/O	20 pins: 5 V, 3.3 V, Ground Digital, analogue
Shut down required	Yes	No
Storage	MicroSD card (up to 1 TB)	flash (32 kb)
Desktop interface	Yes	No, C/C++ required
Power consumption (idle)	3,000 mW / 750 mW	<250 mW

muestran un sistema de monitorización y control de bajo coste, y en tiempo real, para una mini cámara de clima industrial y otros sistemas de climatización. Más en detalle, el sistema propuesto es capaz de controlar y monitorizar variables como la temperatura, mediante la utilización de una Raspberry Pi y una tarjeta controladora STM32F [STM19]. El sistema es implantado en un entorno óptimo para los dispositivos low-cost, por lo que no existe ruido electromagnético. A diferencia de nuestro trabajo, los autores no contemplan las dificultades que podrían presentarse en un entorno hostil, como por ejemplo, en el interior de un cuadro eléctrico con automatismos, bobinados y fuentes de alimentación.

Por otra parte, Caiza et al. [CSPA18] presentan una solución que permite desplegar controladores de circuito cerrado y ser utilizados en procesos industriales. En concreto, utilizan un sistema propietario, el módulo MPS® PA de FESTO, que ofrece un conjunto de sensores industriales y una Raspberry Pi. En general, el trabajo evidencia la efectividad de la Raspberry Pi en conjunto con diferentes dispositivos propietarios, aunque las pruebas se basan en simulación y no en entornos industriales reales. Poner en práctica este tipo de propuestas en entornos reales, como hemos hecho nosotros, podría cambiar los resultados obtenidos, fundamentalmente debido a la sensibilidad que presentan este tipo de dispositivos de bajo coste frente a las EMIs.

Othman et al. [OZAD17] proponen un sistema de monitorización remota en tiempo real, de una instalación fotovoltaica, mediante la utilización de una Raspberry Pi. El sistema es capaz de mostrar parámetros como el voltaje y la temperatura. Los autores indican que su propuesta dispone de cierta flexibilidad para su adaptación a una instalación de mayor complejidad, aunque los ensayos se han llevado a cabo mediante simulación, y no en una instalación real, en la que las inclemencias meteorológicas jugarían un factor importante en la fiabilidad del sistema.

Molano et al. [RCM⁺18] describen una propuesta de arquitectura de IoT aplicada a la industria. En concreto, presentan un metamodelo de integración (Internet de las Cosas, Redes Sociales y la Nube) para la generación de aplicaciones enfocadas a la Industria 4.0. Los autores, además desarrollan un prototipo de monitorización de fabricación implementado con Raspberry, un servidor de almacenamiento en la nube y un dispositivo móvil para controlar de forma remota un proceso de producción. Sin embargo, no realizan una implementación en un entorno real, por lo que tampoco abordan los posibles problemas que pudieran presentarse.

Utilizando una Raspberry Pi, Ramalingam et al. [RBK19] proponen un sistema de monitorización de la contaminación basado en IoT. La propuesta realiza mediciones de temperatura, humedad y el nivel de gases liberados durante el proceso industrial. Para ello, utilizan un mecanismo de control y monitorización de datos por separado. A través de una aplicación móvil del servidor nativo BLYNK [Bly20], muestra el nivel de contaminación ambiental en tiempo real.

Todas las propuestas comentadas muestran la posibilidad de utilizar la Raspberry Pi en la Industria 4.0, aunque a diferencia de nuestra propuesta, en la mayoría de los casos, los sistemas han sido validados en entornos muy básicos o basados en simulación. En esta Tesis pretendemos demostrar la validez de nuestro sistema aplicado a la industria real.



Figura 2.7: Elementos básicos de un problema de EMC [Pau06].

2.5. Interferencias Electromagnéticas (EMIs)

En esta sección se realiza un repaso por los diferentes conceptos que están asociados a las interferencias y la compatibilidad electromagnética, la tipología de las EMIs, su origen y formas de transmisión. Además, se presentan algunas recomendaciones para su eliminación.

2.5.1. Compatibilidad electromagnética

Las EMIs, provocan defectos y alteraciones en el funcionamiento de una gran cantidad de dispositivos electrónicos, como placas, sensores, microcontroladores y microcomputadores. De hecho, los problemas de Compatibilidad Electromagnética (EMC) están adquiriendo un mayor protagonismo en la industria [QZL⁺08], y se presentan en mayor medida cuando usamos dispositivos de bajo coste, que no suelen incorporar complejos sistemas que permitan evitarlos.

La EMC se refiere a la habilidad de un sistema de no causar interferencias electromagnéticas a otros equipos, pero al mismo tiempo ha de ser insensible a las emisiones que pueden causar otros sistemas [FA12].

La EMC tiene en cuenta la generación, transmisión y recepción de energía electromagnética. Estos tres aspectos del problema forman el marco básico de cualquier diseño que tenga en cuenta la EMC (ver Figura 2.7). La fuente es el emisor y una ruta de transferencia o acoplamiento transfiere la energía a un receptor, donde se procesa, dando como resultado un comportamiento deseado o no. La interferencia ocurre si la energía recibida hace que el receptor se comporte de una manera no deseada. La transferencia de energía electromagnética se produce con frecuencia a través de modos de acoplamiento involuntarios. Sin embargo, la transferencia involuntaria de energía causa interferencia solo si la energía recibida es de suficiente magnitud y/o contenido espectral en la entrada del receptor, para hacer que el receptor se comporte de una manera no deseada.

La transmisión o recepción involuntaria de energía electromagnética no es necesariamente perjudicial; es el comportamiento no deseado del receptor el que marca si constituye una interferencia. Por tanto, el procesamiento de la energía recibida por el receptor es una parte importante de la cuestión de si se producirá interferencia o no. Es difícil determinar, a priori, si una señal que incide en un receptor causará interferencia en ese receptor, por lo que no es sencillo identificar de manera única los tres aspectos del problema que se muestran en la Figura 2.7. También es importante comprender que una fuente o un receptor pueden clasificarse como intencionada o no intencionada. El hecho de que la fuente o el receptor sea inten-

2.5. INTERFERENCIAS ELECTROMAGNÉTICAS (EMIS)

cionada o no, depende de la ruta de acoplamiento, así como del tipo de fuente o receptor. Existen tres formas de prevenir la interferencia:

- Suprimir la emisión en su fuente.
- Que la ruta de acoplamiento sea lo más ineficaz posible.
- Hacer que el receptor sea menos susceptible a la emisión.

Es necesario tener en cuenta estas tres alternativas. La primera medida contra las interferencias es suprimir la emisión en la fuente, tanto como sea posible. Sin lugar a dudas, esta es la mejor solución, eliminando la emisión, se elimina el problema. No obstante, no en todas las ocasiones es posible la identificación de la fuente perturbadora y, por ello, no siempre será posible eliminarla.

Para el segundo caso, existen métodos de “fuerza bruta” para reducir la eficiencia de la ruta de acoplamiento. Por ejemplo, colocar el receptor en un recinto metálico, es decir proporcionarle un escudo, que servirá para reducir la eficiencia de la ruta de acoplamiento. Sin embargo, los dispositivos blindados son más costosos que reducir el tiempo de subida o bajada del emisor y, la mayoría de las veces, su rendimiento real en una instalación es menor de lo esperado.

Por su parte, reducir la susceptibilidad del receptor es a menudo difícil de implementar, sobre todo preservando la función deseada del producto. Un ejemplo de implementación de susceptibilidad reducida al ruido de un receptor, sería el uso de códigos de corrección de errores en un receptor digital. Aunque en el receptor incida energía electromagnética no deseada, los códigos de corrección de errores pueden permitir que el receptor funcione correctamente en presencia de una señal potencialmente problemática [Pau06].

Algunas de las causas principales por las que la compatibilidad electromagnética cobra cada vez más importancia son las siguientes:

- Aumento de los equipos electrónicos, tanto en la industria como en el hogar.
- Equipos más grandes y complejos.
- Aumento de sistemas de telecomunicación.
- Disminución del margen de ruido de los sistemas digitales.
- Aumento de la frecuencia de trabajo de los equipos.

2.5.2. Fuentes de EMIs y tipología

Las fuentes de EMIs son de dos tipos, las que pueden considerarse como naturales y las debidas a la intervención humana.

En lo relativo a las fuentes de EMIs de origen natural, se pueden destacar los relámpagos, cuyas descargas pueden llegar hasta los 10kV, y los efectos solares que afectan a la ionosfera. En cuanto a las generadas por el hombre destacan: las descargas electrostáticas, los sistemas eléctricos y electrónicos, los elementos de telecomunicaciones y los pulsos electromagnéticos [FA12].

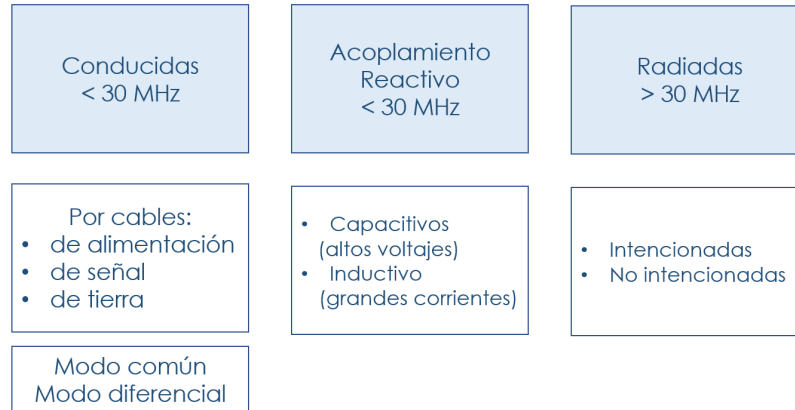


Figura 2.8: Formas de propagación de las EMI [FA12].

Atendiendo al medio de propagación que utilicen las interferencias electro-magnéticas para perjudicar a la calidad de las señal o al funcionamiento de un equipo (ver Figura 2.8), las EMI pueden clasificarse como:

1. EMI conducidas: Son las que se propagan a través de cables de alimentación, de señal o de tierra. La frecuencia de estas señales no supera los 30 MHz.
2. EMI propagadas por acoplamiento reactivo (caso particular de la propagación radiada), que a su vez presentan dos categorías:
 - Por acoplamiento capacitivo: son las que se generan por efecto de campo eléctrico. Su origen principal son los lugares donde se producen grandes variaciones de tensión respecto al tiempo.
 - Por acoplamiento inductivo: son las producidas por el efecto de los campos magnéticos, principalmente originados por los circuitos de potencia, los cuales conducen corrientes elevadas ($> 10 \text{ A}$).
3. EMI radiadas: se producen gracias a la generación de ondas electromagnéticas. Son consideradas radiadas y no acopladas cuando la distancia entre fuente y víctima es superior a la mitad de la longitud de onda de la interferencia.

A continuación, comentamos con un poco más de detalle cada una de ellas.

Las dos formas en las que pueden aparecer las EMI conducidas son en modo diferencial y en modo común. La Figura 2.9 representa la propagación en modo diferencial, la cual se realiza en conductores activos del sistema. Así, la corriente de modo diferencial se propaga por uno de los conductores, pasando a través del equipo provocando, o no, fallos en su funcionamiento para después regresar por otro conductor.

La corriente de modo común, se propaga por los conductores activos y también el conductor de tierra del sistema, es decir, por todos los conductores en el mismo

2.5. INTERFERENCIAS ELECTROMAGNÉTICAS (EMIS)

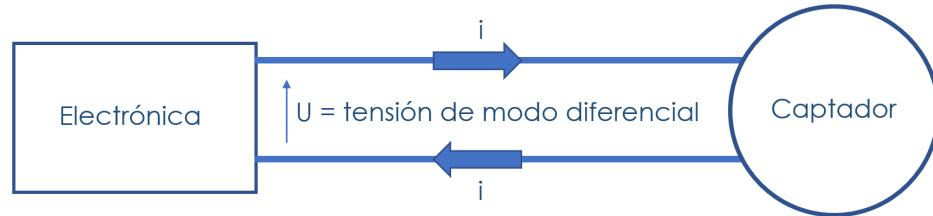


Figura 2.9: EMIs conducidas modo diferencial [Sch18].

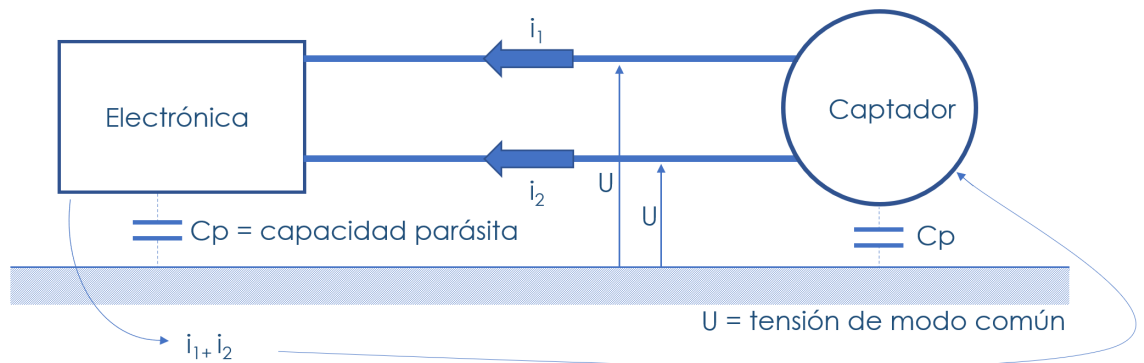


Figura 2.10: EMIs conducidas modo común [Sch18].

sentido, para regresar a través de las capacidades parásitas, tal y como se observa en la Figura 2.10.

En lo relativo al primer tipo de EMIs de acoplamiento reactivo, es decir, las EMIs de acoplamiento capacitivo, éstas son las generadas por una diferencia de potencial variable entre dos circuitos, uno eléctrico (componente, cable, etc.) y otro cercano (masa, conductor). Esta diferencia de potencial genera una corriente eléctrica circulante de uno hacia a otro, formando una capacidad parásita (condensador) como se muestra en la Figura 2.11.

En cuanto al segundo tipo, la Figura 2.12 muestra los elementos necesarios para que se produzca un acoplamiento inductivo. Como se observa, se precisa de un hilo conductor que transporte una corriente creadora de campo magnético y un bucle víctima en el que se generará una fuerza electromotriz capaz de perturbar. La corriente que circule debe ser alta. Este tipo de corrientes suelen ser generada por circuitos de potencia mayores de 10 A.

Ambos acoplamientos reactivos, es decir, el capacitivo y el inductivo, también reciben el nombre de acoplamientos de campo cercano, mientras que el acoplamiento por radiación electromagnética se denomina acoplamiento de campo lejano. La diferencia entre estos dos casos, es la distancia y la frecuencia, es decir, cuando la

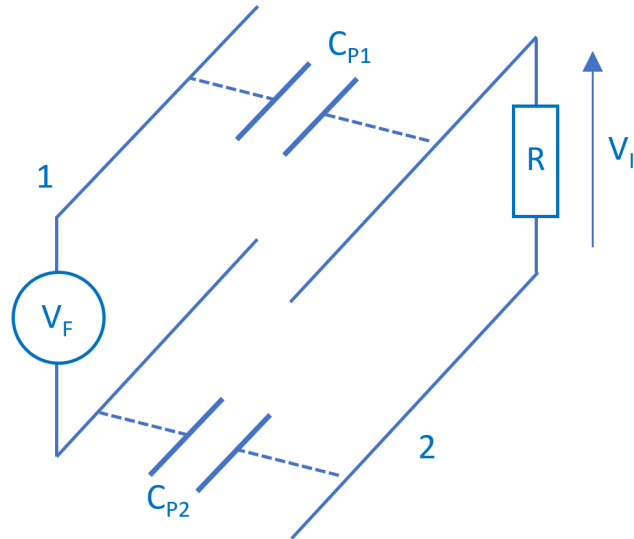


Figura 2.11: Acoplamiento capacitivo [FA12].

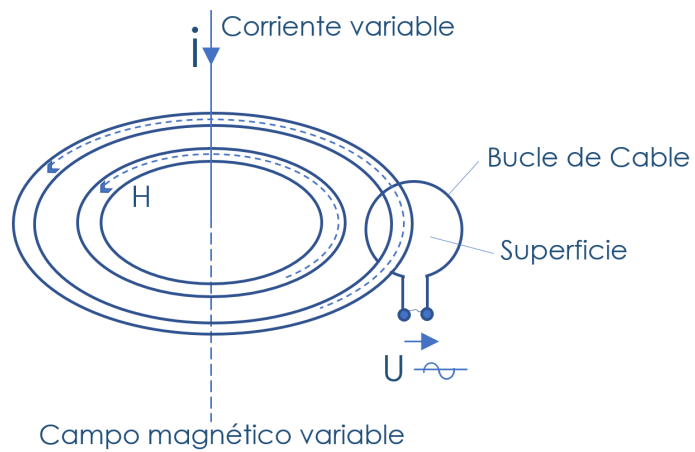


Figura 2.12: Acoplamiento inductivo [Sch18].

2.5. INTERFERENCIAS ELECTROMAGNÉTICAS (EMIS)

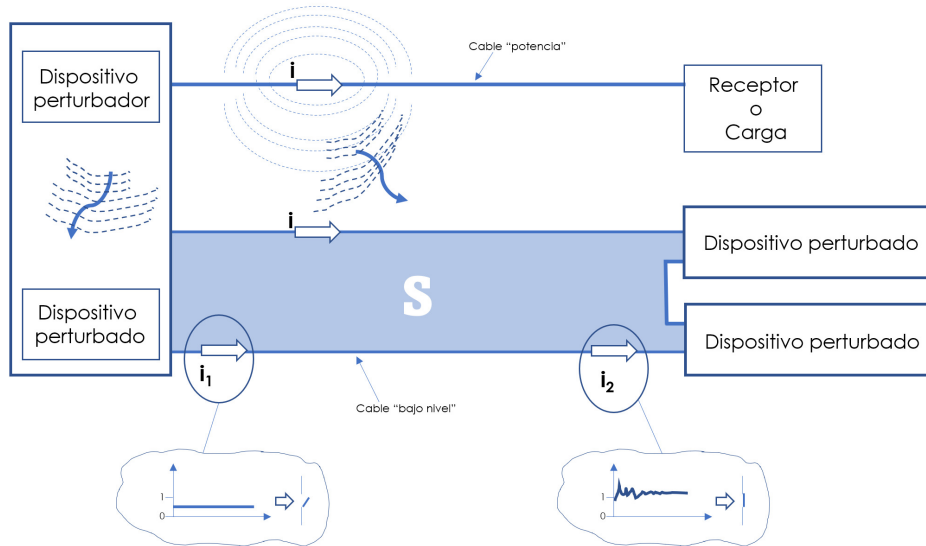


Figura 2.13: Acoplamiento por radiación [Sch18].

víctima está a una distancia igual o superior a $\lambda/2\pi$, siendo λ la longitud de onda de la perturbación, se considera campo lejano [FA12].

Finalmente, los acoplamientos por radiación reciben también el nombre de efecto pelicular (ver Figura 2.13). El valor de la capacidad parásita formada por las dos partes del circuito es proporcional a la superficie que forman los dos circuitos e inversamente proporcional a la distancia entre ellos [Sch18].

En la actualidad, los dispositivos y sistemas electrónicos deben cumplir unas normas que marcan los límites de las EMIs que pueden generar, así como las que pueden recibir sin perjudicar el funcionamiento. El organismo encargado de marcar estas normas en Europa es el Comité Europeo de Normalización Electrotécnica (CENELEC) [FA12]. Dichas normas cubren los cuatro ámbitos problemáticos en lo referido a la EMC:

- Susceptibilidad radiada
- Emisiones radiadas
- Susceptibilidad conducida
- Emisiones conducidas

A pesar de esto, existen equipos que se pueden ver afectados por las interferencias electromagnéticas y muy especialmente los dispositivos electrónicos low-cost. Aunque todos los modelos de Raspberry Pi han sido sometidos a pruebas de EMC y están marcados con el sello de conformidad CE, no significa que todos los sistemas que incluyan la Raspberry Pi cumplan automáticamente con los estándares

relativos a la EMC. Las pruebas a las que se somete la Raspberry Pi para obtener el marcado CE son solamente en concepto de producto terminado, por lo que funcionará sin perturbaciones en un entorno idóneo, es decir, fuera de todo tipo de fuentes de EMI. Si bien es cierto que la Raspberry Pi no emite EMIs de forma significativa, su electrónica es muy susceptible a interferencias de su entorno [Ele21].

Nuestro sistema se ha instalado en los cuadros de mando y maniobra de las propias máquinas. Esto quiere decir, que convive con otros elementos como contactores, conductores eléctricos con diferentes tensiones, relés, interruptores automáticos, fuentes de alimentación y autómatas. Además, el sistema está próximo a los motores de la propia máquina y dentro del campo de influencia de la iluminación fluorescente de las salas. Esto significa, que las interferencias electromagnéticas a las que está expuesto nuestro sistema van, desde emisiones conducidas, producidas por la propia fuente de alimentación de la Raspberry Pi, cableado o las propias pistas de la PCB, hasta radiadas por acoplamiento reactivo (inductivo y capacitivo), es decir, interferencias de campo cercano, y radiadas por campo lejano, todas ellas producidas por campos eléctricos o magnéticos que son transferidos al sistema víctima a través del espacio. A diferencia de nuestra propuesta, como veremos a continuación, lo más común es mitigar las EMIs conducidas y radiadas mediante filtros físicos.

2.5.3. Eliminación de EMIs en dispositivos low-cost

Como se ha comentado anteriormente, los dispositivos low-cost presentan una mayor sensibilidad a las EMIs que otros dispositivos. Este aspecto, ha sido estudiado por diferentes autores, con el fin de conocer los problemas y buscar soluciones para mitigar las EMIs.

Con el objetivo de determinar el estado de la plataforma de desarrollo Raspberry Pi con respecto a la compatibilidad electromagnética, Mach et al. [MKV⁺17], analizan y comparan la compatibilidad electromagnética de campo cercano y campo lejano de la Raspberry Pi. Los resultados obtenidos indican que la Raspberry Pi contiene una gran cantidad de componentes que generan interferencias electromagnéticas muy superiores a las esperadas y además presentan una gran susceptibilidad a éstas.

Mynster y Jensen [MJ16] describen ciertos desafíos y soluciones que están relacionadas con la EMC y las EMIs en los dispositivos IoT. En concreto, prestan especial atención a los sistemas integrados y a las perturbaciones radiadas electromagnéticas, puesto que existen diferencias notables entre los entornos de prueba estandarizados por las normativas y los entornos en los que los dispositivos operan realmente.

Li et al. [LWHZ16] presentan cómo el aumento de la densidad de los componentes y la disminución del voltaje de funcionamiento han provocado que los microcontroladores sean más sensibles a las EMIs. Las corrientes bajas y los voltajes acoplados al microcontrolador, a través de sus pines, son el motivo por el que se pueden producir fallos en el funcionamiento e incluso daños al dispositivo. Los autores también tienen en cuenta los efectos del envejecimiento de los dispositivos y la degradación que sufren los parámetros físicos de los semiconductores,

pudiendo debilitar la protección de dispositivos contra las EMIs.

La aparición de nuevos compuestos poliméricos, que han sustituido a los materiales metálicos convencionales en el apantallamiento y blindaje de EMIs, es abordada por Sankaran et al. [SDAP18]. Estos nuevos polímeros presentan un valor añadido, ya que evitan la corrosión, son más ligeros, y poseen propiedades dieléctricas, térmicas, mecánicas y magnéticas superiores, lo que reduce el ruido electromagnético.

En el ámbito del análisis de las Printed Circuit Boards (PCBs) (la Raspberry está construida sobre una PCB), Xiao et al. [XQL⁺19], proponen un método de simulación field-TL-circuit para analizar el impacto de los parámetros de cableado en el acoplamiento electromagnético de las PCB dentro de los equipos electrónicos. En concreto, los autores han desarrollado un modelo de equipo electrónico con una PCB, capaz de analizar el problema de las EMIs en las PCB con tres rutas de cableado diferentes. Los resultados obtenidos indican que el circuito sensible debe mantenerse alejado del área de cableado que interfiera en mayor medida, y corroboran que la instalación de cableado adecuado puede reducir las interferencias electromagnéticas radiadas.

Chen et al. [CZC19] también estudian los efectos de las EMIs en las PCB. Hacen alusión a la estructura de blindaje en estas placas, prestando atención, tanto al campo cercano y campo lejano, comparando diferentes blindajes y cómo pueden afectar a la efectividad de la PCB. Los autores demuestran el gran efecto que tienen las cajas de blindaje en la reducción de las EMIs. Más en detalle, comprueban que se logra un blindaje efectivo cuando existe un espacio de 30 mm en la caja de blindaje; sin embargo, si se aumenta esa distancia, el efecto de las EMIs cada vez es más perjudicial. Además, los autores analizan la influencia del tamaño de la PCB, resultando más afectada por las EMIs cuanto mayor es su tamaño, puesto que el cableado impreso es más largo y la capacidad de eliminar las interferencias disminuye. No obstante, si el tamaño fuese demasiado pequeño, la PCB se vería afectada fácilmente por el cableado adyacente, además de no tener una buena disipación de calor. Además, recomiendan que la PCB se conecte a tierra, siempre utilizando un punto, y evitando que el cableado sea largo, para no generar un efecto antena.

En resumen, la mayoría de los trabajos que tratan sobre las interferencias electromagnéticas en este tipo de dispositivos, realizan mediciones y cuantifican sus efectos, pero no llevan a cabo una eliminación de las mismas. Por otra parte, los autores que abordan su eliminación, en su mayoría analizan las interferencias conducidas a través de la línea eléctrica, es decir, se limitan a una sola forma de transmisión de EMIs y utilizan soluciones hardware. Nuestra propuesta, sin embargo, ofrece una solución software para la eliminación de EMIs en los dispositivos low-cost. No realizamos una distinción del origen, la fuente o la forma de transmisión de las interferencias (radiadas o conducidas), eliminando las señales erróneas que afectan al dispositivo. Esta peculiaridad hace que nuestro sistema aporte más flexibilidad para adaptarse a diferentes entornos y aplicaciones.

2.6. Agrupamiento de datos y búsqueda binaria

En la actualidad, existen diferentes algoritmos que tienen como fin el agrupamiento de datos (clustering). El clustering es un proceso para agrupar datos en varios grupos, de modo que los datos en un clúster tengan un nivel máximo de similitud, y por el contrario, que los datos entre clústeres tengan una similitud mínima [Kau16].

Por otra parte, existen algoritmos que permiten la localización exacta de un elemento en una lista ordenada de datos, como es el caso del algoritmo de búsqueda binaria. Este método se utiliza dividiendo repetidamente a la mitad, la lista que podría contener el dato buscado, hasta encontrarlo.

Como veremos más adelante en el Capítulo 5, en esta Tesis se ha diseñado un algoritmo que permite automatizar algunas de las operaciones que se llevan a cabo en el mecanismo de filtrado DBF. En concreto, dicho algoritmo agrupa las señales de sellado y determina, sin la necesidad de intervención por parte de los operarios, el inicio y fin de la producción de cada tipo de producto, así como el número de tipos de productos elaborados durante el proceso. Para poder entender mejor su funcionamiento, en esta sección se muestran en detalle algunos algoritmos clásicos, diseñados para el agrupamiento de datos, y el de la búsqueda binaria. En concreto, se exponen sus fórmulas y funcionamiento, así como algunos de los casos de uso para los que se han empleado.

2.6.1. k-means, x-means y k-rms

El k-means y x-means son dos de los algoritmos de agrupamiento más utilizados en el aprendizaje no supervisado. El k-means agrupa objetos en k grupos (siendo el valor de k previamente fijado por el usuario), minimizando la suma de distancias entre cada objeto y el centroide de su grupo o clúster. Mientras, el x-means es una variación del k-means en la que no es necesario indicar inicialmente el número de clústeres que se pretende obtener, puesto que es el propio mecanismo el que establece el número de grupos, en base a los datos analizados. Por otro lado, el k-rms es una variante del algoritmo k-means, más actual y mejorada, pues requiere un menor número de iteraciones y proporciona una mayor precisión.

Los algoritmos, k-means y x-means han sido muy utilizados para la clusterización y su uso se ha hecho más notable con la aparición del Big Data o el Machine Learning. Por su parte, el k-rms es más novedoso y presenta ciertas ventajas sobre el k-means, aunque a diferencia del x-means, también es necesario proporcionarle el número de clústeres en los que se desea dividir el conjunto de datos.

k-means [DHS73] es un algoritmo de aprendizaje no supervisado que resuelve el problema de agrupamiento. Consideramos (X_1, X_2, \dots, X_n) como el conjunto de n puntos que queremos agrupar en k grupos, los cuales denominaremos S_j , que contienen N_j puntos de datos. El algoritmo k-means encuentra una partición que minimiza el error cuadrático medio entre el centroide de cada grupo (μ_j) y todos los puntos del mismo ($X_n \in S_j$). La función de agrupamiento viene dada por la siguiente ecuación [Bis95]:

$$J = \sum_{j=1}^k \sum_{n \in S_j} \|X_n - \mu_j\|^2 \quad (2.2)$$

donde μ_j es la media aritmética de los datos del conjunto S_j , y viene dada por:

$$\mu_j = \frac{1}{N_j} \sum_{n \in S_j} X_n \quad (2.3)$$

El k-means es sencillo de implementar, además de tener una buena escalabilidad en lo referente a los datos. La desventaja clara es que, para utilizar este método de clustering, inicialmente debemos indicar de forma manual el número de grupos en los que queremos dividir el conjunto de datos.

El mecanismo x-means [Ish00] solventa este inconveniente, ya que no necesita que el usuario defina el número de conglomerados (clústeres), sino que realiza un preprocesamiento de los datos, que le permite estimar el número de clústeres adecuado en función de los datos que se desean agrupar. De hecho, se basa en el k-means, pero permite comprobar, de forma automática, diferentes agrupaciones dentro de un rango variable.

Más en detalle, el algoritmo x-means comienza con un número de clústeres (k) igual al límite inferior del rango dado (k_{min}), y va agregando clústeres hasta alcanzar el límite superior (k_{max}). Durante ese proceso, calcula el Criterio de Información Bayesiano (BIC) para cada una de esas soluciones, y finalmente selecciona la que obtiene la mejor puntuación.

Por su parte, el k-rms [GD20] ha sido diseñado de tal forma que reduce el número de iteraciones y aumenta la precisión con respecto al k-means. A diferencia de éste, utiliza la raíz de la media cuadrática (Root Mean Square, RMS) para actualizar los centroides de cada clúster, en lugar de utilizar la distancia media de los puntos, tal y como lo hace el k-means. Esto permite reducir el número de iteraciones de forma significativa. Además, el k-rms es más resistente a los valores atípicos, lo que sin duda permite que obtenga mejores resultados a la hora de agrupar los datos.

El funcionamiento del k-rms parte de un conjunto de datos representados por pares (X_i, Y_i) , generando de manera aleatoria M centroides (X_j, Y_j) , para formar M clústeres. Una vez hecho esto, calcula la distancia euclidiana promedio de cada dato a cada centroide mediante la siguiente fórmula:

$$\delta_{ij} = \sqrt{\frac{(X_i - X_j)^2 + (Y_i - Y_j)^2}{n}} \quad (2.4)$$

donde $1 \leq i \leq n, 1 \leq j \leq M$.

Posteriormente, se calcula la RMS de los datos asignados a cada centroide, de forma separada para las coordenadas X e Y (ver ecuaciones 2.5 y 2.6), obteniendo un nuevo valor del centroide (X_{rms}, Y_{rms}) .

$$X_{rms} = \sqrt{\frac{X_1^2 + X_2^2 + \dots + X_n^2}{n}} \quad (2.5)$$

$$Y_{rms} = \sqrt{\frac{Y_1^2 + Y_2^2 + \dots Y_n^2}{n}} \quad (2.6)$$

Finalmente, se miden las distancias inicial y final de cada centroide, y en el caso de que dicha distancia sea menor a un umbral fijado, se considera que ha convergido y no se realizan más iteraciones para ese centroide en particular. Para el resto de centroides que no cumplan dicha condición, se comienza de nuevo el algoritmo.

2.6.2. Uso de los algoritmos k-means, x-means y k-rms

A continuación presentamos, algunos trabajos que analizan el funcionamiento de estos tres métodos de agrupamiento (i.e., los algoritmos k-means, x-means y k-rms), o que presentan propuestas basadas en los mismos.

En relación al k-means, existen trabajos que lo aplican en diferentes áreas, lo comparan con otros mecanismos, e incluso lo mejoran. Rahamathunnisa et al. [RNASK20] proponen un sistema, basado en k-means y Multi SVM (Support Vector Machine) para la detección de enfermedades en productos vegetales y orgánicos. Para ello, utilizan el procesamiento de imágenes, atendiendo al color, forma, tamaño y textura, y dichas características son tratadas con el algoritmo k-means, agrupando los vegetales infectados. Los autores comparan el agrupamiento de k-means y el algoritmo SVM con el procesamiento de imágenes digitales en MATLAB, alcanzando su propuesta una mayor precisión, y empleando menor tiempo. En nuestro caso, no realizamos un procesamiento de imágenes para identificar los productos, sino que determinamos el número de clústeres en función de las variaciones de las modas estadísticas en los valores de sellado.

Siswanto et al. [SPAI15], utilizando el operador de Sobel y el k-means, realizan una segmentación automática de imágenes de productos alimenticios, para así poder identificarlos. El operador Sobel les permite determinar la región de interés, y luego emplean el mecanismo de agrupamiento k-means para separar el objeto y el fondo, consiguiendo una identificación clara de qué producto se trata. Los autores indican que los resultados de identificación son más precisos mediante la combinación de Sobel y k-means, que solamente usando el k-means. En nuestra propuesta, el objetivo también es identificar los diferentes productos que pasan por la línea de fabricación, pero no utilizamos un sistema de reconocimiento de imágenes, sino que utilizamos los tiempos de sellado para identificar los productos elaborados en cada momento.

Hüseyinli et al. [HYA18] utilizan el algoritmo k-means para categorizar productos de comercio electrónico, automatizando la extracción de las especificaciones técnicas, a través de la minería de textos. Para ello, en primer lugar preprocesan el conjunto de datos y realizan la extracción de atributos de un conjunto. Los datos procesados son el precio del producto, la disponibilidad, la imagen del producto, así como sus especificaciones. Posteriormente, realizan la operación de agrupamiento mediante el k-means. Los resultados obtenidos muestran una efectividad del 98 %. En nuestra propuesta, sin embargo, no es necesario hacer un preprocesado de los datos, y la efectividad de los resultados obtenidos, como se verá en la Sección 5.3, es del 100 %.

Hochdörffer et. al [HLL17], teniendo en cuenta los requisitos de producción en términos de variedad de productos y tareas de planificación relacionadas, proponen un método de agrupamiento denominado Product Variety Management (PVM), capaz de manejar datos binarios relacionados con el proceso de producción (es decir, datos que describen si una tecnología de producción es o no aplicable a una variante de producto), y datos métricos relacionados con la capacidad de producción (datos que describen la capacidad para aplicar una tecnología de producción en un paso del proceso). Aplicando este método logran que los productos se dividan en grupos, en los que las variantes de los productos tengan requisitos similares. Los resultados obtenidos demuestran que PVM permite reducir la complejidad de la planificación en el diseño de redes de producción. Además, los autores indican que los métodos de agrupamiento tradicionales (como el k-means), no tienen capacidad para manejar datos de características mixtas. Nuestra propuesta trabaja con datos de las mismas características, no siendo necesaria la utilización de un algoritmo que aplique características mixtas para determinar los tipos de productos elaborados, así como el instante de inicio o final de la producción de los mismos.

En lo relativo al uso del x-means, Noorbehhahani et. al [NM18] abordan las limitaciones que presentan los enfoques tradicionales para la clasificación del tráfico de red. Más en detalle, proponen métodos de clasificación semisupervisados, puesto que consideran que los métodos modernos de clasificación de tráfico, que utilizan aprendizaje automático, requieren una gran cantidad de datos etiquetados para extraer un modelo de clasificación preciso, lo que consideran costoso y con un gran requerimiento de tiempo. Su nuevo método semisupervisado está basado en el x-means, junto con una nueva técnica de propagación de etiquetas. La precisión del método propuesto demuestra su eficacia para aprender a clasificar el tráfico de red utilizando datos etiquetados limitados.

Por su parte, Imamura et. al [IKH10] proponen una técnica de extracción automática de objetos en movimiento en vídeos, utilizando el método de agrupamiento x-means. Su propuesta permite extraer los puntos característicos de cada fotograma del vídeo en función de la afinidad de los movimientos. Los resultados experimentales revelan que la propuesta consigue la extracción precisa de los distintos objetos. Sin embargo, en nuestro caso, el x-means devuelve valores erróneos en términos de número de clústeres para los diferentes turnos estudiados, tal y como mostramos en la Sección 5.3.

Finalmente, cabe destacar que la reciente aparición del k-rms ha impedido que todavía haya sido utilizado de forma mayoritaria. No obstante, para evaluar su rendimiento, Avishek y Dipankar [GD20] realizaron experimentos utilizando 12 conjuntos de datos de diferente naturaleza, obtenidos del archivo web de la Universidad de California. Los resultados obtenidos demuestran que k-rms obtiene unos mejores resultados en comparación con el algoritmo k-means tradicional, presentando una mayor precisión y requiriendo un menor número de iteraciones. Cabe destacar que, a la hora de identificar los diferentes productos que han sido elaborados en la industria quesera donde hemos validado nuestra propuesta, el k-rms también obtiene unos mejores resultados no solo comparados con el k-means, sino también con el x-means.

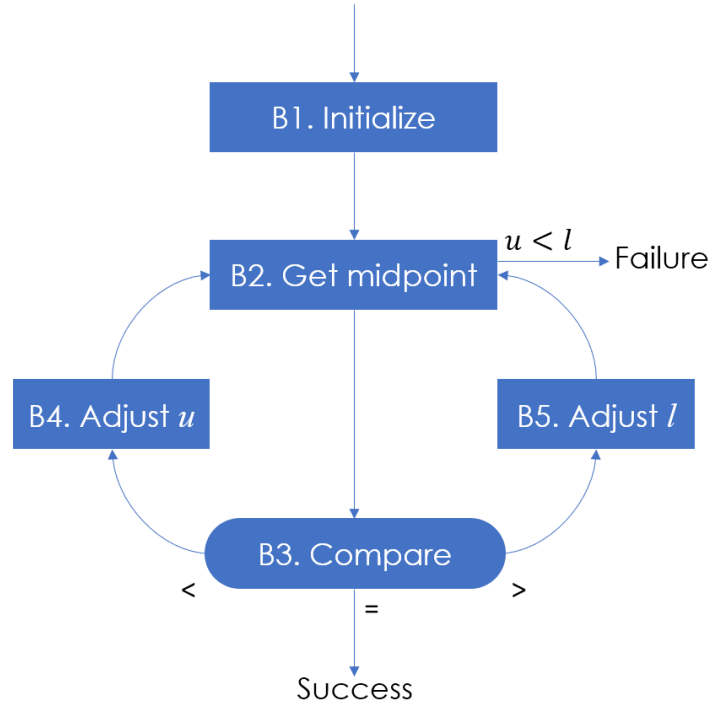


Figura 2.14: Búsqueda Binaria [Knu99].

2.6.3. Búsqueda Binaria

El algoritmo de búsqueda binaria, también conocido como búsqueda de medio intervalo o búsqueda logarítmica, es un algoritmo iterativo de búsqueda eficiente, que funciona con una lista ordenada, y que divide el espacio de búsqueda en dos mitades en cada iteración. En concreto, compara el valor requerido con el elemento que se sitúa en el medio de la secuencia y, si el elemento no se encuentra en esa posición, se descarta la mitad en la que no se encuentra el elemento, y se realiza la búsqueda del valor requerido en la otra mitad. Si el elemento no se encuentra, el proceso se repite hasta que se encuentra la coincidencia, o la lista se agota (ver Figura 2.14). En nuestra propuesta, utilizando un método similar a la búsqueda binaria, podremos detectar cuándo se producen cambios de producto durante el proceso de fabricación.

La búsqueda binaria funciona tal y como se detalla en el Algoritmo 1. Este algoritmo tiene como ventajas que se puede aplicar, tanto a datos en listas lineales como a árboles binarios de búsqueda, además de ser el método más eficiente para encontrar elementos en una lista ordenada. Sin embargo, el algoritmo funciona solamente con listas ordenadas, por lo que suele requerir de un preprocesamiento de datos.

Algoritmo 1 Búsqueda Binaria

```
/*  $R_1 R_2 \dots R_N$  tabla de registros cuyas claves están en orden creciente  
 $K_1 < K_2 < \dots < K_N$  */  
// el algoritmo busca un argumento determinado  $K$   
//  $l$  es el puntero inferior del conjunto de datos tomado  
//  $u$  es el puntero superior del conjunto de datos tomado  
//  $i$  es el punto medio del conjunto de datos tomado
```

[Step 1] [Inicializar] Se establece $l \leftarrow 1, u \leftarrow N$

[Step 2] [Obtener punto medio] En este punto sabemos que si K está en la tabla, satisface $K_l \leq K \leq K_u$. Si $u < l$ el algoritmo termina sin éxito, de lo contrario se establecerá $i \leftarrow \lceil (l + u)/2 \rceil$

[Step 3] [Comparar] Si $K < K_i$, ir a Step 4; si $K > K_i$, ir a Step 5; y si $K = K_i$, el algoritmo termina satisfactoriamente

[Step 4] [Ajustar u] Establecer $u \leftarrow i - 1$ y volver a Step 2

[Step 5] [Ajustar l] Establecer $l \leftarrow i + 1$ y volver a Step 2

2.6.4. Uso del algoritmo Búsqueda Binaria

La búsqueda binaria es uno de los mecanismos de búsqueda más utilizados para la aplicación a un conjunto de datos. A continuación, presentamos algunos trabajos que han hecho propuestas basadas en la búsqueda binaria.

Puesto que en la búsqueda binaria la comparación se realiza dividiendo los n elementos por la mitad en cada iteración, Vuyyuru [Vuy19] presenta un nuevo algoritmo que reduce el número promedio de comparaciones necesarias para buscar un elemento en función del tamaño del mismo, mostrando que, al considerar sólo aquellos elementos cuyo tamaño es igual al de la clave, se reduce el número de pasos necesarios para su búsqueda. Este algoritmo necesita conocer el tamaño del elemento, así como la clave buscada, para reducir el tiempo medio necesario que reduzca el tamaño de comparaciones. En nuestra propuesta, lo que buscamos precisamente es el valor de cambio entre los datos que se van produciendo de manera continuada en el turno de producción, sin que sea necesario conocer previamente el tamaño y el valor que buscamos.

Jacob et. al [JAD17] realizan una combinación de la búsqueda binaria y de la búsqueda lineal, creando un nuevo algoritmo con una forma eficaz de buscar un elemento clave determinado en una matriz sin clasificar, en un tiempo limitado. Los autores tratan los problemas de cada algoritmo por separado ya que, en la búsqueda lineal, cada elemento se consulta y se compara con el elemento clave de forma secuencial y, en la búsqueda binaria, los datos deben ordenarse de alguna forma, por lo que se requiere mucho tiempo para ello. En nuestro caso, a diferencia de esta propuesta, no es necesario ordenar los datos, puesto que tratamos las señales en el mismo orden en el que se van generando durante el proceso de

producción.

Finalmente, Bai et. al [BYZX17] proponen un algoritmo de búsqueda binaria mejorado, que está basado en la combinación de búsqueda binaria dinámica y el algoritmo de búsqueda binaria hacia atrás. Los resultados experimentales muestran que el algoritmo mejorado reduce significativamente los tiempos de búsqueda y la cantidad de datos transmitidos, en comparación con la búsqueda binaria tradicional, mejorando así la eficiencia. Más concretamente, los resultados muestran una reducción en la búsqueda entre un 75 % y un 85 %, en términos de tráfico.

El algoritmo MoBiSea, que presentamos en esta Tesis, permite determinar el punto exacto en el que se produce el cambio de producto en el proceso productivo, agrupa los productos elaborados en clústeres de manera eficiente, y no necesita un ordenamiento de los datos para funcionar correctamente, puesto que los datos son tratados en el orden en el que se producen. Además, comparamos MoBiSea con los mecanismos de agrupamiento k-means, x-means y k-rms, en lo referido a la detección de los clústeres, en la precisión en la detección del inicio y fin de la elaboración de cada producto, así como en el número de señales en cada clúster.

2.7. Conclusiones

La Industria 4.0 representa una corriente hacia la tecnificación de la industria, así como signos de una evolución propicia hacia una mejora de la eficiencia en los procesos, su flexibilidad y comunicación e integración de toda la cadena de valor de productos y procesos.

Sin embargo, los elementos de control y monitorización, los sensores, así como la amplia tipología de dispositivos y equipos utilizados en la Industria 4.0 se caracterizan por ser sistemas exclusivos de los fabricantes, y por el alto coste que conlleva su despliegue. Debido a estos inconvenientes, esta Tesis presenta un sistema basado en dispositivos de bajo coste que permite obtener los parámetros necesarios para la estimación del Overall Equipment Effectiveness (OEE), un factor que permite tomar decisiones estratégicas en aras del aumento de la productividad en la industria y la mejora continua del trabajo.

Como hemos visto, las propuestas relacionadas con el uso de dispositivos de bajo coste muestran la posibilidad de utilizar la Raspberry Pi en la Industria 4.0, aunque a diferencia de nuestra propuesta, en la mayoría de los casos, los sistemas han sido validados en entornos muy básicos o basados en simulación. En esta Tesis vamos a validar nuestro sistema en una industria real.

Por tanto, el uso de dispositivos low-cost en la industria, aunque en la actualidad es todavía incipiente, es posible. Sin embargo, uno de los problemas que puede presentar el uso de este tipo de dispositivos de bajo coste es que suelen ser muy sensibles a las interferencias electromagnéticas, pudiendo afectar en gran medida a la precisión de los valores medidos. Para solventar estos problemas, esta Tesis también presenta diferentes mecanismos de filtrado de EMIs en un dispositivo low-cost, como es la Raspberry Pi.

En lo relativo a las interferencias electromagnéticas, hemos visto que la mayoría de los trabajos que las tratan, realizan mediciones y cuantifican sus efectos, pero no llevan a cabo una eliminación de las mismas. Por otra parte, existen trabajos

que abordan su eliminación, pero en su mayoría utilizan soluciones hardware y únicamente analizan las interferencias conducidas a través de la línea eléctrica, es decir, se limitan a una sola forma de transmisión de EMIs. Nuestra propuesta, sin embargo, ofrece una solución software para la eliminación de EMIs en los dispositivos low-cost. No realizamos una distinción del origen, la fuente o la forma de transmisión de las interferencias (radiadas o conducidas), pero eliminamos de manera efectiva las señales erróneas que afectan al dispositivo. Esta peculiaridad hace que nuestro sistema aporte más flexibilidad para adaptarse a diferentes entornos y aplicaciones.

Capítulo 3

Arquitectura low-cost para la medición del OEE en la Industria 4.0

En esta Tesis se presentan varias aportaciones relacionadas directamente con la Industria 4.0 y el Internet de las Cosas (IoT). La primera de las aportaciones, es un sistema de bajo coste, basado en Raspberry Pi, que permite la estimación en tiempo real del Overall Equipment Effectiveness (OEE), un índice que permite a los responsables de las empresas detectar qué aspectos deben tener en cuenta para mejorar la eficacia del proceso productivo.

La idea fundamental es poder medir de forma precisa las variables relativas a la eficiencia productiva de las máquinas termoselladoras en la industria (i.e., el OEE), con un coste sensiblemente menor que otras soluciones propietarias. En concreto, el sistema de bajo coste propuesto, recoge y procesa las operaciones de sellado realizadas y la disponibilidad de las selladoras.

En este capítulo se presenta en detalle una arquitectura de bajo coste basada en el uso de Raspberry Pi, capaz de calcular el índice de la efectividad global de la maquinaria industrial (OEE).

3.1. Introducción

El uso del sistema presentado durante este capítulo permitirá abaratar los costes y mejorar la capacidad de reacción de la empresa, pues posibilitará detectar los aspectos que se deben mejorar para aumentar globalmente la productividad.

A continuación, se comentan brevemente las diferentes secciones del presente capítulo. La Sección 3.2 presenta en detalle la arquitectura del sistema propuesto, los elementos de los que se compone, y un diagrama de flujo del mismo. En la Sección 3.3 se muestran los tipos de señales que el sistema va a recoger (i.e., las señales de sellado y las de disponibilidad de la máquina). Además, se entra en detalle en las conexiones necesarias para la recogida de dichas señales. En la

3.2. ARQUITECTURA DEL SISTEMA

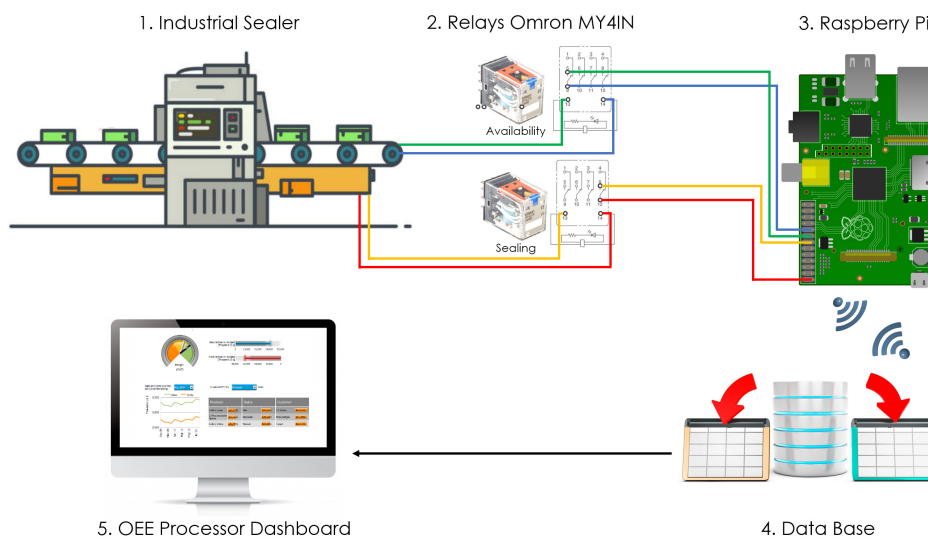


Figura 3.1: Arquitectura del sistema.

Sección 3.4 se comentan en detalle los datos obtenidos por nuestro sistema, y se presenta un ejemplo de cómo se calcula el OEE. La Sección 3.5 muestra cómo se ha implantado el sistema en la fábrica Marcos Conde, una industria quesera de reconocido prestigio. Por último, la Sección 3.6 presenta las conclusiones de este capítulo.

3.2. Arquitectura del sistema

La configuración del sistema que se presenta en esta Tesis es capaz de realizar mediciones de variables que intervienen en el cálculo del OEE a través de un dispositivo low-cost. Para ello, es necesaria la utilización de diferentes elementos que se presentan a continuación.

3.2.1. Elementos del sistema

El sistema está compuesto por cinco elementos claramente diferenciados (ver Figura 3.1). Estos elementos son:

- Selladora Industrial (ver Figura 3.2): elemento de la cadena de producción que realiza el envasado. Una vez que se ha preparado y colocado el material plástico en la materia prima, esta máquina es la encargada de realizar las operaciones de sellado de los envases. Este elemento es el emisor de las señales que recogerá nuestro dispositivo de bajo coste.
- Relés Omron MY4IN (ver Figura 3.3): reciben los pulsos eléctricos de la máquina relativos a las señales de sellado y disponibilidad. Son los dispositivos electromagnéticos que se estimulan a partir de la corriente eléctrica



Figura 3.2: Selladora industrial [ULM22].

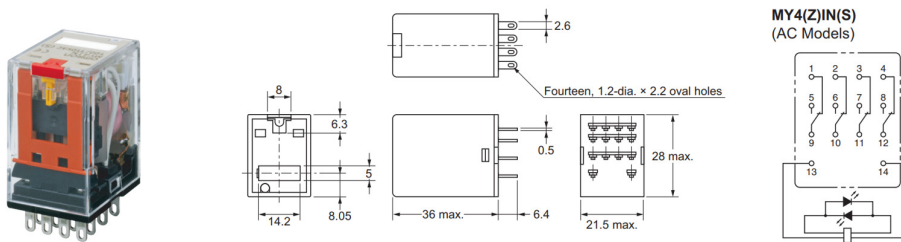


Figura 3.3: Relé Omron MY4IN [Omr22].

recibida por la selladora industrial. Éstos abrirán o cerrarán el circuito, produciendo la correspondiente señal que se enviará a la Raspberry Pi. En el sistema propuesto, existen dos unidades encargadas de recibir las señales de disponibilidad y sellado independientemente, enviando cada una de estas señales a la Raspberry Pi.

- Raspberry Pi (ver Figura 3.4): microcomputadora low-cost programada para recibir las señales, gestionar la base de datos y aplicar los mecanismos de filtrado de EMIs propuestos en esta Tesis, i.e., DBF y SCF (que se presentan en las Secciones 4.2 y 4.3), así como el mecanismo MoBiSea (presentado en la Sección 5.2). Es el dispositivo encargado de procesar las señales, para su posterior envío a la base de datos a través de una conexión inalámbrica. Para ello, se ha implementado una aplicación específica en Java, utilizando la librería Raspberry Pi 4 Java [Ora22], que permite la recogida de señales de los puertos *General Purpose Input/Output* (GPIO) de la Raspberry Pi. Además, realiza la función de servidor, permitiendo el acceso remoto desde los dispositivos autorizados, para la visualización y monitorización de las líneas de producción. La utilización de Raspberry Pi se debe, no solo a su bajo coste, sino por la necesidad de realizar las inserciones de las señales

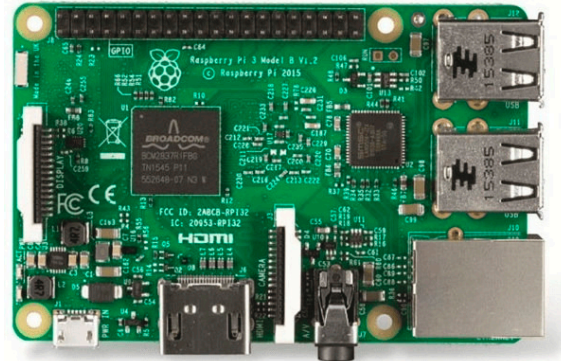


Figura 3.4: Raspberry Pi 3 [Ras22].

recogidas en la base de datos, y por la capacidad para monitorizar estos datos, a través del servidor programado en el propio dispositivo.

- Base de Datos: se encarga de almacenar todas las señales recogidas y procesadas por las Raspberry Pi (una por cada línea de producción). De esta forma, se podrán almacenar los datos de las medidas recogidas de las diferentes líneas en las que se implante el sistema.
- OEE Dashboard: permite la monitorización de los datos procesados, así como visualizar las distintas métricas relacionadas con la efectividad global (es decir, los ratios de disponibilidad, de rendimiento y de calidad).

El sistema propuesto, por tanto, recoge y procesa las variables aportadas por la selladora industrial. En concreto, se registra la operación de sellado, es decir, cuándo baja el pistón sellador, y la disponibilidad, es decir, cuándo la selladora está o no disponible para trabajar, o está trabajando.

3.2.2. Diagrama de flujo del sistema

Para explicar el funcionamiento del sistema, se ha elaborado un diagrama de flujo (ver Figura 3.5), que muestra el proceso que ocurre cuando la Raspberry Pi recibe las diferentes señales de sellado o disponibilidad. El dispositivo está programado para poder realizar el procesamiento de las señales procedentes de la máquina y su posterior inserción en la base de datos.

En concreto, el sistema permanece a la espera de la llegada de las señales a través de los GPIO de la Raspberry Pi, las procesa y comprueba su correcta inserción en la base de datos. Las señales se separan, en función del origen, como GPIO Availability y GPIO Sealings, y seguirán el mismo proceso en ambos casos, pero con diferentes rutas. Una vez recibida, la señal se envía a la Base de Datos correspondiente, Sealings o Availability, y se comprueba si se ha insertado correctamente. Si no se ha realizado su inserción de manera correcta, normalmente debido a la falta de conectividad, la señal pasa al Pending File Storage y el sistema retornaría a la espera de una nueva señal. En el caso de una inserción correcta en

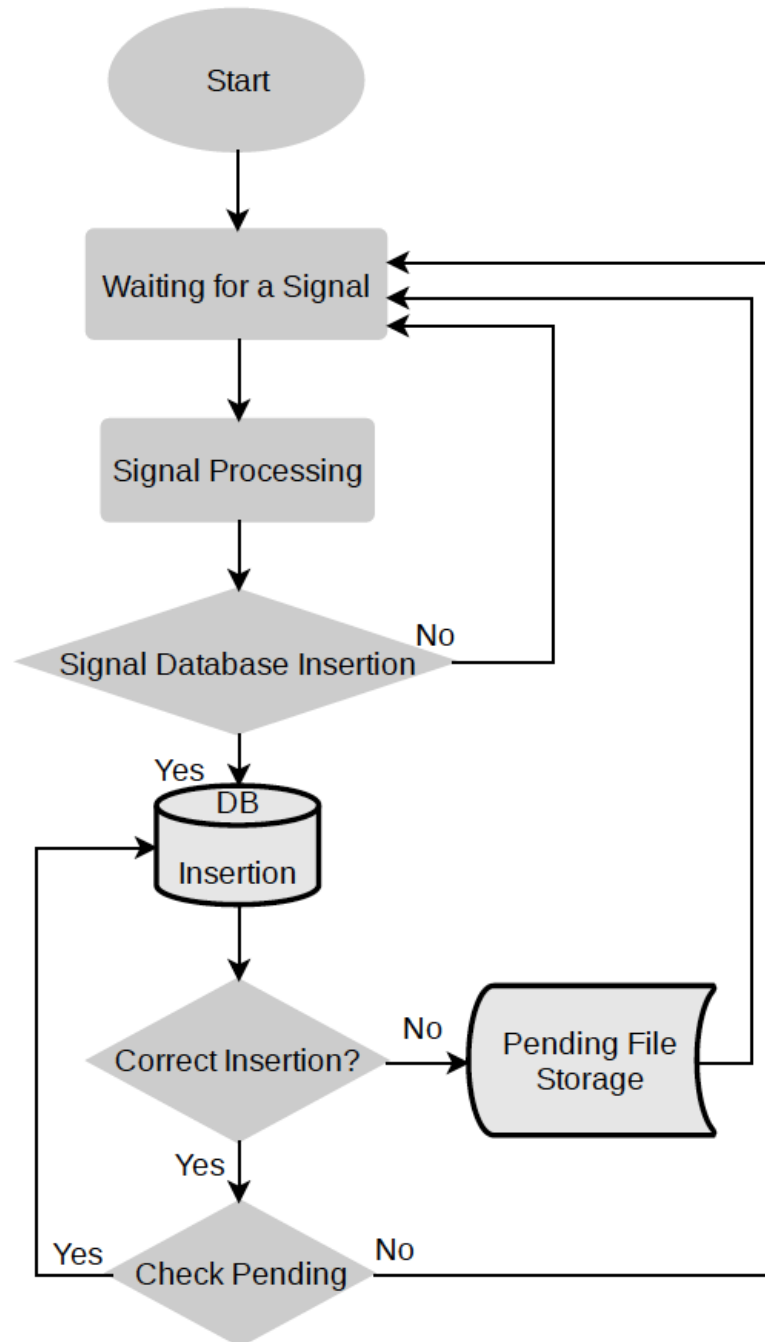


Figura 3.5: Diagrama de flujo del sistema propuesto.

la base de datos, el sistema comprobará si existen señales pendientes de insertar, almacenadas en el Pending File Storage. Si no existen señales pendientes, el sistema queda a la espera de nuevas señales, pero si hubiese señales pendientes las insertaría en la Base de Datos.

3.3. Tipos de señales

Las señales que recogemos de la selladora son de dos tipos, sellado y disponibilidad. La señal de sellado está compuesta por un ON y un OFF de forma conjunta. Más concretamente, el sellado se produce con una señal de 1 y 0 (correspondiente a una bajada y subida del pistón de sellado) y, de esta forma, el sistema registrará un sellado cuando aparezcan las dos señales de forma consecutiva.

Por su parte, la señal de disponibilidad indica si la máquina está preparada para funcionar y por tanto, dicha señal puede ser de dos tipos: ON (1), cuando existe disponibilidad, y OFF (0), cuando la selladora no está disponible.

Como se ha indicado anteriormente, el procedimiento a seguir por el sistema es permanecer a la espera de la llegada de señales, procesarlas, separarlas en función de su origen y comprobar su correcta inserción en la base de datos.

A continuación, se presentan en detalle los dos tipos de señales que va a recoger el sistema.

3.3.1. Señal de sellado

La señal de sellado medida por el sistema propuesto, proporciona los datos necesarios para determinar el rendimiento de la máquina industrial. Gracias a esta señal, podemos obtener el número de sellados realizados y el instante en los que éstos se producen, y así poder estimar el cálculo del factor OEE.

Las señales de sellado presentan un patrón claro, en función de los tiempos que emplea la máquina para realizar un sellado. El tiempo que permanece la máquina en estado 1 corresponde al tiempo durante el cual el pistón está sellando el termoplástico, por lo que este tiempo es el que se considera que tarda la máquina en realizar una operación de sellado. Por su parte, el tiempo que la máquina permanece en estado 0 (pistón levantado) es el que se considera como tiempo entre sellados.

Es importante destacar que, dichos patrones pueden variar en función de la selladora industrial utilizada, y sobre todo, del producto a sellar, ya que el tiempo requerido para sellar una cuña de queso no es el mismo que para sellar un queso entero o un cuarto de queso.

La Figura 3.6 muestra un ejemplo de la información recogida por el sistema, relativa al proceso de sellado durante un turno. Como se observa, pueden diferenciarse los productos que se han envasado y las paradas realizadas. Las líneas de color verde representan los tiempos de sellado de varios productos, a los cuales denominamos Sealing Time (Ts). Más concretamente, se observa el sellado de tres productos diferentes: cuartos de queso (Product A), quesos enteros (Product B) y cuñas de queso (Product C).

CAPÍTULO 3. ARQUITECTURA LOW-COST PARA LA MEDICIÓN DEL OEE EN LA INDUSTRIA 4.0

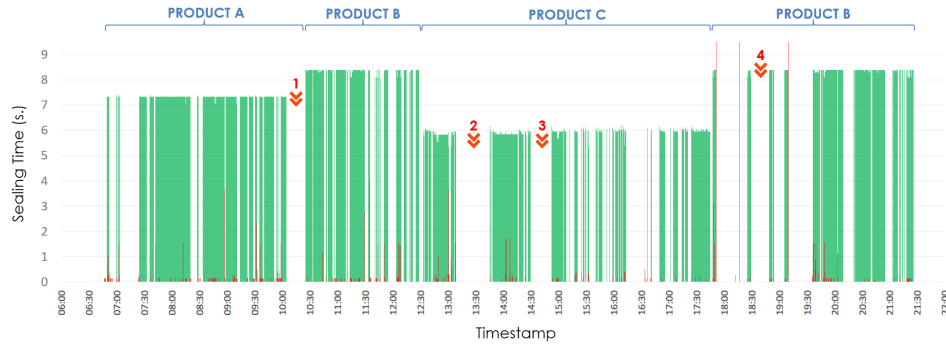


Figura 3.6: Señales de sellado y EMIs en un turno de ejemplo.

El sellado de los quesos enteros (Product B), se ha realizado en dos momentos diferentes del turno y es el proceso que mayor Sealing Time requiere, 8.38 segundos. Los cuartos de queso (Product A) emplean menos tiempo en ser sellados que los quesos (7.32 segundos), y las cuñas (Product C) requieren aún un tiempo menor, concretamente 5.83 segundos, en promedio. Por otra parte, se pueden diferenciar claramente las paradas realizadas durante el turno. A modo de ejemplo, se observan las paradas programadas de descanso y relevo de personal a las 10:00 y las 13:00 horas (1 y 2, respectivamente), así como una parada por mantenimiento a las 14:30 horas (3) y un cambio de rollo de plástico a las 18:30 horas (4). El resto de paradas, a excepción de la parada de descanso de personal a las 19:00, no son paradas programadas.

Debido a que la señal de sellado presenta patrones de comportamiento en los tiempos de sellado, siendo homogéneos y repitiéndose éstos de manera constante, la alteración anómala de estos patrones en los turnos de fabricación indicará la presencia de ruido electromagnético.

De hecho, la Figura 3.6 también muestra los efectos de las EMIs en color rojo; estas señales representan los valores que no responden al patrón de sellado del producto y que manifiestan claramente el ruido electromagnético recogido por el dispositivo. Por lo tanto, se identifican los productos y las paradas, pero también se observa las señales producidas por las EMIs. Concretamente, en el ejemplo presentado, se han contabilizado 3916 sellados, mientras que el número de sellados reales es de 2540, lo que supone un 35.14 % de señales erróneas debido a las EMIs. De hecho, como se ha visto, se puede determinar qué productos han sido elaborados durante el turno, sin necesidad de eliminar el ruido, pero no se puede realizar una estimación correcta del OEE en términos de productividad, puesto que no se puede obtener exactamente el número de sellados que realmente se han producido, sin eliminar previamente las señales anómalas debidas a las EMIs.

3.3.2. Señal de disponibilidad

La disponibilidad es otro de los parámetros que conforman el factor OEE y que mide nuestro sistema teniendo en cuenta el estado de la máquina (ON/OFF). A

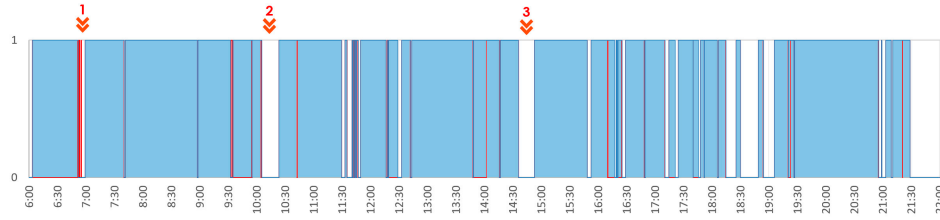


Figura 3.7: Señales de disponibilidad y EMIs en un turno de ejemplo.

través del análisis de esta señal se determinan los tiempos en los que la máquina ha estado disponible y cuándo no. De esta forma también podemos determinar cuándo la máquina ha estado lista para realizar tareas, pero no ha ejecutado ninguna operación, así como los tiempos de parada y compararlos con las paradas de producción planificadas.

A diferencia del proceso de sellado, las señales que indican la disponibilidad de la máquina son unitarias, es decir, no son un conjunto de dos valores (0 y 1). Además, son asíncronas y no presentan un patrón como en el caso de las señales de sellado.

La Figura 3.7 muestra en azul, un ejemplo de los valores obtenidos relativos a la disponibilidad de la máquina, durante un turno completo. Se pueden observar puntos característicos que indican diferentes incidencias ocurridas durante el proceso. Sirva como ejemplo la parada a las 6:50 (1), que representa un reajuste en el plástico utilizado en la operación de sellado de los blíster. Por otra parte, la parada que existe entre las 10:00 y las 10:30 (2), así como la existente entre las 14:30 y las 15:00 (3), son paradas programadas; la primera de descanso del personal de turno y la segunda de relevo de personal. El resto de paradas, a excepción de la realizada a las 18:30, por cambio de rollo de plástico, y la de las 19:00, por descanso de personal, no son paradas programadas.

La detección de señales erróneas en disponibilidad es más compleja. Lo que nos muestra la presencia de interferencias electromagnéticas es la aparición de señales consecutivas del mismo tipo, es decir, si la máquina está parada (0), no podrá volver a enviar una señal de parada (0), y lo mismo ocurriría en el caso de que la máquina presentase un estado (1). Por otra parte, las interferencias electromagnéticas en disponibilidad, podrán ser detectadas cuando se contrastan las señales de sellado con el estado de la máquina, es decir, si la máquina está sellando no tendría que manifestar una señal de (0) en disponibilidad. La Figura 3.7 muestra con líneas rojas las interferencias electromagnéticas que se han producido en el turno representado.

En definitiva, se puede identificar cuándo la máquina ha estado disponible y cuándo no, así como las alteraciones producidas por EMIs en la señal de disponibilidad.

Por otra parte, si tenemos en cuenta las señales mostradas en la Figura 3.7, incluidas las debidas al ruido, observamos una disponibilidad del 69.12 % (11 horas y 3 minutos), mientras que si eliminamos las señales producidas por las EMIs, la disponibilidad es del 81.50 % (13 horas y 2 minutos). De hecho, las señales totales



Figura 3.8: Armario de componentes electromecánicos.

registradas por el dispositivo son 354, resultando válidas únicamente 82, es decir, se registran un 76.84 % de señales erróneas. Estos datos demuestran la necesidad de eliminar las EMIs, para así poder calcular de forma precisa el OEE.

3.3.3. Conexiones necesarias para la recogida de las señales de sellado y disponibilidad

Las señales de sellado y disponibilidad se recogen en el armario de componentes electromecánicos de la selladora industrial (ver Figura 3.8). A continuación, se comentan las características y singularidades de estas señales.

- Captación de la señal de sellado:

La señal de sellado se corresponde con la señal que realiza la bajada y subida del pistón sellador. En la configuración de la máquina, esta señal la recibe una válvula electroneumática que pone en funcionamiento el cilindro neumático para que realice la función de sellado. Por ello, la señal que necesitamos captar en nuestro dispositivo es la misma que ordena al cilindro neumático el sellado, es decir, la obtenemos de la señal que opera en la válvula electroneumática de ese cilindro. La Figura 3.9 muestra cómo se obtiene esa señal y se dirige al relé Omron que la envía a la Raspberry Pi.

- Captación de la señal de disponibilidad:

3.3. TIPOS DE SEÑALES

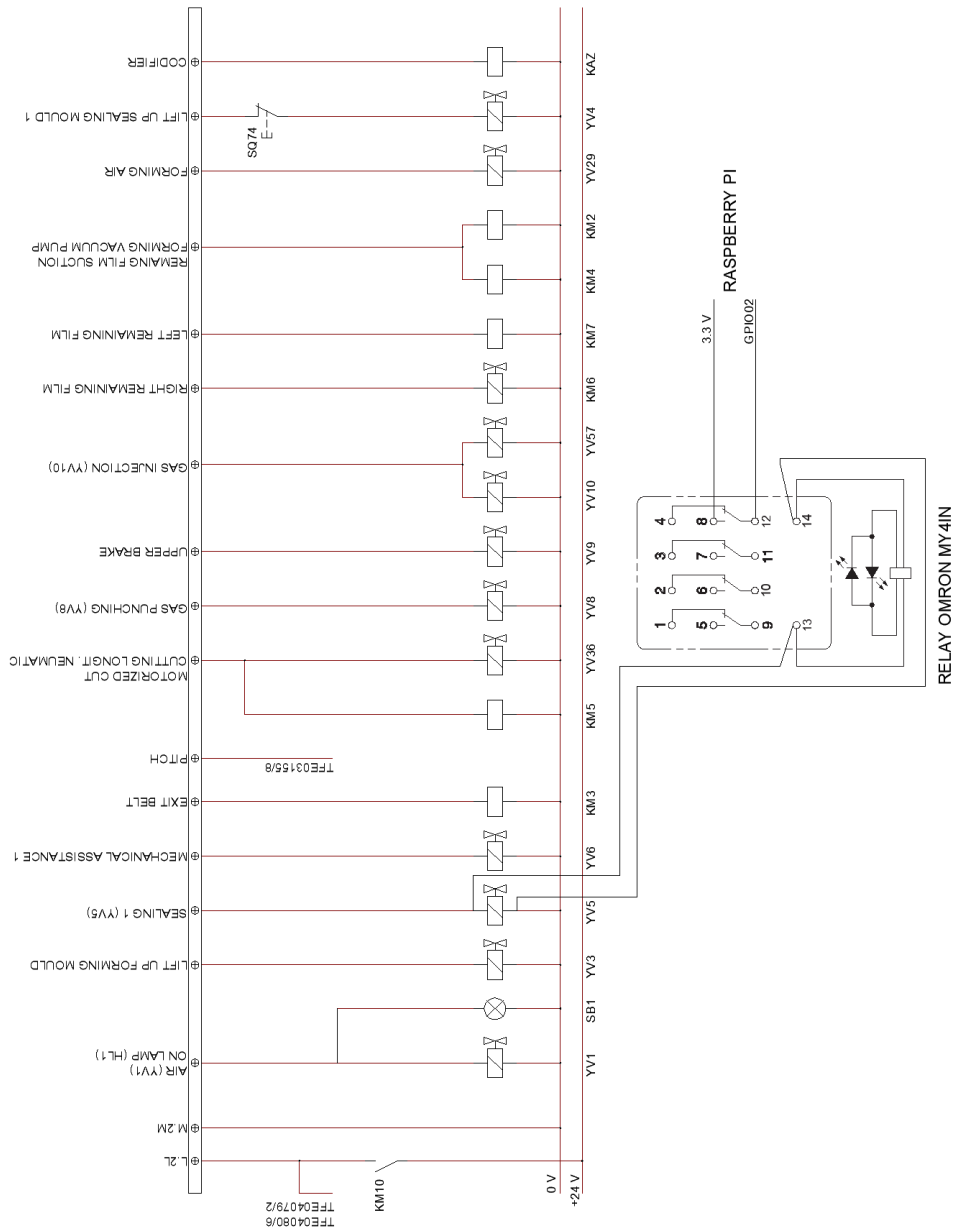


Figura 3.9: Esquema de conexiones para la obtención de señal de sellado.

CAPÍTULO 3. ARQUITECTURA LOW-COST PARA LA MEDICIÓN DEL OEE EN LA INDUSTRIA 4.0

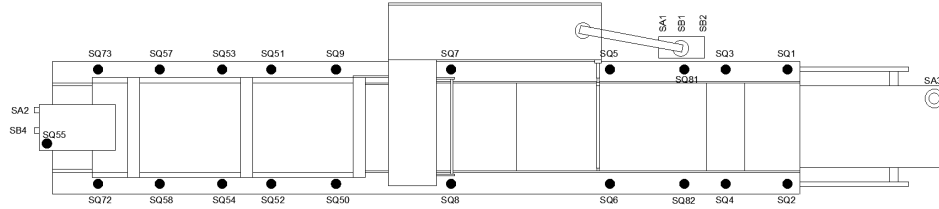


Figura 3.10: Vista en planta de la selladora y señales recogidas por la máquina.

Tabla 3.1: Nomenclatura de las señales de guardas, emergencias, marcha y paro de la selladora industrial

Nombre	Uso	Nombre	Uso
SA2	Seta emergencia	SQ7	Guarda sellado
SB4	Paro	SQ8	
SQ55	Guarda cinta salida	SQ5	Guarda zona de carga
SQ73	Guardas de corte	SQ6	
SQ57		SA1	Seta emergencia
SQ53		SB1	Marcha
SQ51		SB2	Paro
SQ9		SQ81	Guardas de formado
SQ72		SQ3	
SQ58		SQ1	
SQ54		SQ82	
SQ52		SQ4	
SQ50		SQ2	

La Figura 3.10 y la Tabla 3.1 muestran el diseño de la selladora, así como las diferentes señales que la máquina es capaz de recoger. Estas señales se corresponden con los diferentes mecanismos que permiten operar a la máquina y hacer un seguimiento del proceso de forma pormenorizada. En concreto, se tienen señales de guarda de formado plástico, de la zona de carga, del sellado termoplástico, del corte y la salida de producto, así como de las setas de parada de emergencia y de los pulsadores de marcha y paro.

Todas estas señales también están representadas en la Figura 3.11, pues intervienen en la obtención de la señal de disponibilidad. Para garantizar que la señal de disponibilidad es obtenida únicamente cuando la selladora está disponible y no sólo conectada, la señal se recoge de un relé de seguridad. Este relé de seguridad se encarga de velar que cada operación se está realizando de manera adecuada y que todos los elementos de la máquina se encuentran en situación favorable y de seguridad para operar en el termosellado, termoformado y corte del plástico que se va a utilizar para las operaciones.

La Figura 3.11 muestra cómo el relé de seguridad verifica que la máquina está lista para poder funcionar de manera correcta, comprobando las posi-

CAPÍTULO 3. ARQUITECTURA LOW-COST PARA LA MEDICIÓN DEL OEE EN LA INDUSTRIA 4.0

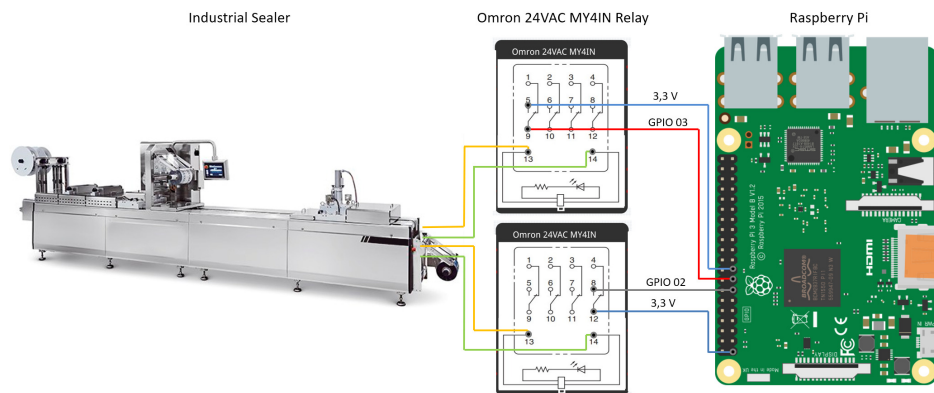


Figura 3.12: Cableado de los relés con la Raspberry Pi y la selladora industrial.

ciones de los finales de carrera de los elementos de seguridad y protección dispuestos a lo largo de la máquina, para evitar atrapamientos y otros accidentes laborales, así como el armado correcto de las setas de seguridad. Esta información es recibida por el relé a través de dos líneas, una de contactos normalmente cerrados y otra de contactos normalmente abiertos. Ambas condiciones deben de cumplirse para garantizar la salida del relé de seguridad, la cual indica que todo es correcto y que la máquina está disponible. En ese momento, el relé de seguridad emite la señal de disponibilidad de la máquina selladora, y se toma la señal que se dirige al relé Omron de disponibilidad, para posteriormente remitírsela a la Raspberry Pi. Concretamente del contacto 13 y 14 del relé de seguridad, dirigimos la señal a los bornes 13 y 14 del relé Omron, obteniendo la señal de disponibilidad de la máquina selladora.

La Figura 3.12 muestra el esquema de conexiones de los relés, tanto con la selladora, como con la Raspberry Pi. Como se puede observar, los relés poseen dos partes diferenciadas de actuación: (i) por un lado, las conexiones 13 y 14 que corresponden a los contactos de activación del relé y que reciben la señal procedente de la máquina (24 V), y (ii) los contactos con números del 1 al 12 (con una menor tensión, 3.3 V, que es la tensión de trabajo de los pines GPIO de la Raspberry Pi), y que actúan en función de la señal recibida por los bornes anteriores (13 y 14).

Cabe destacar las diferentes configuraciones de los contactos del relé Omron, siendo las conexiones 1, 2, 3 y 4 dedicadas a una configuración con contactos normalmente cerrados, es decir, se abren en la conexión, y los bornes 5, 6, 7 y 8 para una configuración de contactos normalmente abiertos, es decir, los contactos se cierran cuando reciben la tensión de servicio. Los bornes 9, 10, 11 y 12 se utilizan como bornes comunes y sirven, tanto para la conexión de contactos normalmente abiertos como contactos normalmente cerrados.

La señal de sellado, llega al relé de sellado, a través de sus conexiones 13 y 14, actuando en el contacto 8 y enviando a la Raspberry Pi la señal a través de su GPIO 02. En el caso de la disponibilidad, la señal enviada por la selladora

industrial, la reciben las conexiones 13 y 14 del relé dedicado a disponibilidad, actuando en el contacto 9 para enviar la señal a la Raspberry Pi mediante el GPIO 03. Los contactos 5 y 12, de los relés de sellado y disponibilidad, se utilizan para alimentar la señal del relé, es decir, suministrar tensión a los contactos libres de potencial.

En definitiva, todas las señales recibidas por estos dos relés Omron son las comentadas anteriormente, es decir, el relé destinado a sellado recibe la señal cuando se envía la señal de sellado a la electroválvula del pistón neumático de sellado, y la disponibilidad recibe las señales del relé de seguridad de la selladora.

3.4. Medición del OEE

A continuación, comentamos en detalle los datos que se obtienen al utilizar nuestro sistema. En concreto, existen diferentes datos obtenidos a través de las mediciones, que van a ser tratados por el sistema, con el fin de monitorizar y almacenar los parámetros que intervienen en el cálculo del OEE. Estos datos son los siguientes:

1. Código de Línea de Producción: este código, formado por cuatro dígitos, permite identificar la línea de producción a la que pertenece la máquina de sellado de la cual se obtienen los datos. De esta manera, se identifica de forma unívoca los datos de eficiencia obtenidos para cada línea.
2. *Timestamp*: sirve para determinar el momento exacto en el que se ha producido el evento registrado. Más en detalle, se refiere a la cantidad de milisegundos transcurridos desde las 00:00:00 UTC del 1 de enero de 1970, hasta el momento en el que se produce dicho evento. En nuestro sistema, utilizamos el *Timestamp* para poder determinar cuándo se ha producido cada sellado en cada línea de producción, y cuándo está disponible cada una de las selladoras.
3. Tipo: este dato se utiliza únicamente para medir la disponibilidad de la máquina, ya que para las señales de sellado no es necesario. En concreto, se identifica el estado de la máquina con un 1 o un 0, en función de si la máquina está disponible o no, de forma que al combinar dicha información con el *Timestamp*, se podrá calcular fácilmente el ratio de disponibilidad en cada momento.

Las Tablas 3.2 y 3.3 incluyen algunos registros, a modo de ejemplo, de los datos recogidos en la planta, relativos tanto al sellado como a la disponibilidad. Más en detalle, la Tabla 3.2 muestra el momento exacto en el que se han producido las operaciones de sellado para seis líneas de producción (4001-4006). La primera columna indica el código de línea al que pertenece el sellado y en la segunda columna aparece el *Timestamp* de dicha operación.

Por su lado, la Tabla 3.3 muestra los datos necesarios para determinar la disponibilidad del sistema. La primera columna indica el código de línea, la segunda columna el *Timestamp* de cuándo ha variado el estado, y en la tercera columna

CAPÍTULO 3. ARQUITECTURA LOW-COST PARA LA MEDICIÓN DEL OEE EN LA INDUSTRIA 4.0

Tabla 3.2: Ejemplo de datos de sellado

Línea	Timestamp
4004	1548996653297
4005	1548996659752
4001	1548996666061
4002	1548996672370
4004	1548996678681
4006	1548996684992
4005	1548996691301
4001	1548996697611
4003	1548996703922

Tabla 3.3: Ejemplo de datos de disponibilidad

Línea	Timestamp	Tipo
4004	1548995124051	1
4004	1548995129606	0
4004	1548995588100	1
4006	1548995600172	1
4004	1548995812593	0
4005	1548995948159	1
4006	1548996004220	0
4005	1548996033515	1
4003	1548996139873	1

aparece el dato de disponibilidad (0 indica que la máquina no está disponible y 1 que la máquina puede trabajar).

Con estos datos seremos capaces de determinar los tiempos efectivos de trabajo de la máquina, el número de operaciones realizadas en los tiempos programados, los tiempos perdidos, las pérdidas de velocidad y otros parámetros que son necesarios para el cálculo del índice OEE.

El sistema propuesto, tras el procesamiento de las señales, es capaz de calcular los ratios de Rendimiento y de Disponibilidad, puesto que puede procesar a partir del Tiempo de Producción Planificado, el tiempo perdido debido a la inactividad y a las paradas menores. Además, nuestro sistema permite introducir los datos relativos a la producción efectiva, es decir, el número de productos que realmente han sido sellados de forma correcta, con lo que es sencillo estimar el tiempo perdido debido a los productos defectuosos (es decir, lo que denominamos tiempo perdido por defectos).

A modo de resumen, la Tabla 3.4 incluye los datos recogidos por nuestro sistema durante 10 horas de trabajo programadas. En concreto, se muestran las horas relativas a las diversas pérdidas (es decir, por paros programados, por inactividad, así como las pérdidas de velocidad debidas a paradas menores y por defectos). Estos valores permiten calcular los diferentes parámetros que afectan a la disponibilidad,

Tabla 3.4: Datos para el cálculo del OEE

Concepto	Horas
Total Available Time (TAT)	10
Scheduled Downtime (SD)	1.5
Downtime Losses (DL)	1
Speed Losses (SL)	0.5
Defect Losses (DFL)	1.3
Net Available Time (NAT)	8.5
Operating Time (OT)	7.5
Net Operating Time (NOT)	7
Fully Productive Time (FPT)	5.7

al rendimiento y a la calidad, que a su vez, permiten obtener el valor final del OEE.

La Figura 3.13 muestra el desglose de cálculos que son necesarios para obtener cada uno de los términos que aparecen en la Fórmula 3.1, suponiendo los valores de la Tabla 3.4.

$$OEE = Disponibilidad \cdot Rendimiento \cdot Calidad \quad (3.1)$$

$$OEE = 88.24 \% \cdot 93.33 \% \cdot 81.43 \% = 67.06 \% \quad (3.2)$$

Tras hacer los cálculos para hallar cada uno de los términos de la fórmula del OEE, se realizan las operaciones pertinentes para obtener el resultado final (ver Ecuación 3.2).

Finalmente, la Figura 3.14 muestra de forma visual los valores del OEE y las pérdidas producidas.

A la vista de los resultados, los gestores de la empresa verían que, aunque el rendimiento es relativamente bueno, para mejorar su productividad sería necesario mejorar los aspectos relativos a la calidad (reduciendo el número de paquetes defectuosos) y de disponibilidad (reduciendo, en la medida de lo posible, los periodos de inactividad).

3.5. Implantación del sistema

Para poner en marcha nuestro sistema, a la vez que se valida nuestra propuesta, hemos realizado la implantación de los dispositivos en una fábrica de quesos en la provincia de Zamora. Esta fábrica tiene una antigüedad de 70 años y se ha consolidado como una de las más importantes de su sector, vendiendo sus productos a nivel internacional. Los productos son de máxima calidad y dan respuesta a una demanda de mercado muy competitiva. La fábrica dispone de recursos humanos con una gran cualificación, así como medios tecnológicos avanzados y modernos. Los productos que elaboran son quesos de oveja y de mezcla de cabra y vaca, y son elaborados con materia prima de la propia provincia. La empresa actualmente está en expansión y ha abierto una nueva nave de producción y almacenaje de 5.000 m² [Mar22].

CAPÍTULO 3. ARQUITECTURA LOW-COST PARA LA MEDICIÓN DEL OEE EN LA INDUSTRIA 4.0



Figura 3.13: Ejemplo de cálculo de los diferentes parámetros del OEE.

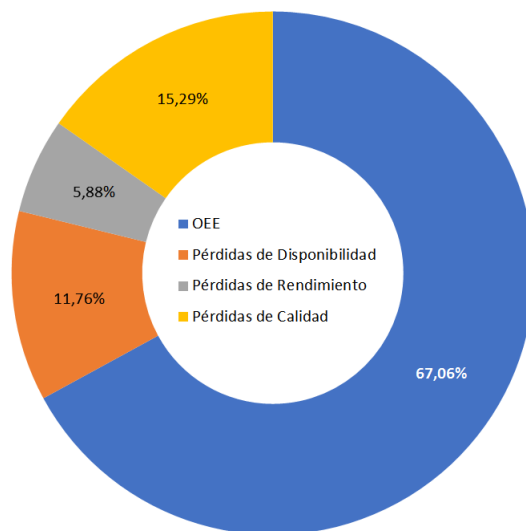


Figura 3.14: Gráfica de Eficiencia Global de la Máquina (OEE).

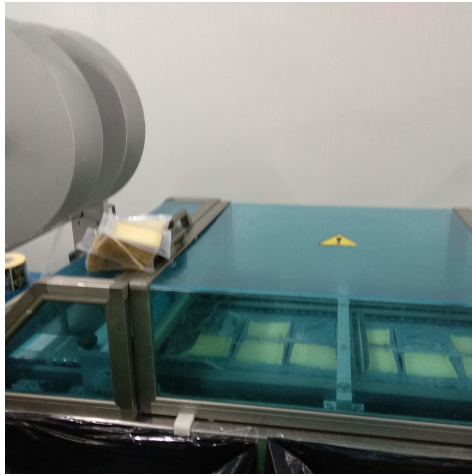
La fábrica cuenta con varias líneas de producción y diferentes productos. Como hemos indicado anteriormente, los productos que elaboran tienen diferentes formatos, siendo principalmente quesos enteros, cuartos y cuñas de queso.

A continuación se muestran detalles de las líneas donde hemos implantado nuestra propuesta y desde las cuales hemos recogido los datos que se han analizado durante la realización de esta Tesis. Cabe destacar que en todo momento se han seguido las medidas de seguridad alimentaria, protocolos de higiene, así como las normas de prevención de riesgos laborales establecidas en la fábrica.

En la Figura 3.15(a) se observa una de las selladoras con las cuñas dispuestas para ser selladas con el plástico que aparece en la parte superior de la imagen. Esta línea la manipulan dos operarios que preparan las cuñas de forma manual y activan las máquinas de sellado a demanda, controlando en todo momento el correcto funcionamiento del proceso. En la Figura 3.15(b) los operarios colocan las cuñas en las bandejas para que posteriormente sean selladas. El detalle de la colocación del plástico de la selladora se muestra en la Figura 3.15(c).

El sistema propuesto se ha instalado en los cuadros de actuación de cada una de las diferentes líneas, de tal forma que los dispositivos que forman parte del sistema conviviesen con el conjunto de elementos, relés, automatismos, autómatas y fuentes de alimentación propias de un cuadro de estas características.

Finalmente, la Figura 3.16 muestra diferentes aspectos de la instalación de nuestro sistema. Más concretamente, la Figura 3.16(a) incluye el detalle de los dos relés que reciben las señales de sellado y disponibilidad, así como la Raspberry que recibe estas señales. En la Figura 3.16(b) puede observarse la cámara que ha grabado el comportamiento de los relés, en función de las señales aportadas por la selladora, con la que se ha comprobado las señales recibidas por nuestro sistema. Además, muestra algunos detalles del entorno con el que convive el sistema propuesto y en el que pueden observarse autómatas, contactores, distintos cableados, etc. Por su parte, la Figura 3.16(c) muestra la visión general del cuadro de otra de las líneas de sellado, remarcado en rojo nuestro dispositivo Raspberry Pi, junto con el dispositivo de monitorización en tiempo real de los parámetros y variables que se están midiendo en ese momento.



(a)



(b)



(c)

Figura 3.15: Línea de sellado termoplástico: (a) perfil de la selladora, (b) operarios cargando la selladora con cuñas de queso y (c) detalle de la colocación del plástico en la selladora.



(a)



(b)



(c)

Figura 3.16: Detalles del sistema instalado: (a) relés Omron para recoger las señales de disponibilidad y sellado, (b) Raspberry Pi funcionando, junto con la cámara de grabación del comportamiento de las señales y (c) Raspberry Pi encapsulada en el cuadro eléctrico y maletín de monitorización de la lectura de las señales.

3.6. Conclusiones

En este capítulo se ha presentado la arquitectura del sistema para la medición de las variables que son necesarias para el cálculo del OEE. El sistema está formado por cuatro elementos clave: (i) la máquina selladora de la que se recogen las señales de sellado y disponibilidad, (ii) los dos relés Omron que sirven para canalizar estas señales al tercer elemento del sistema, (iii) la Raspberry Pi y, por último, (iv) el entorno de monitorización (dashboard).

El sistema propuesto recoge y procesa las variables aportadas por la selladora industrial. En concreto, se registra la operación de sellado, es decir, cuándo baja el pistón sellador, y la disponibilidad, es decir, cuándo la selladora está o no disponible para trabajar, o está trabajando.

Las señales recogidas por el sistema, es decir, las señales de sellado y disponibilidad, se han presentado en detalle, así como las singularidades que presentan. Las de sellado presentan patrones de comportamiento con tiempos de sellado (Sealing Time) concretos para cada producto y, las de disponibilidad, que obedecen a estados de On (1) y Off (0). En ambos tipos de señales, se ha observado el efecto de las EMIs y cómo afectan a la fiabilidad y eficiencia del cálculo del OEE. Además, se han presentado los esquemas de conexión para la recogida de señales de las máquinas, cómo se obtienen cada una de las señales, cómo se derivan a los relés y posteriormente a la Raspberry Pi.

El uso del sistema presentado permitirá abaratar los costes y mejorar la capacidad de reacción de la empresa, pues posibilitará detectar los aspectos que se deben mejorar para aumentar globalmente la productividad.

Finalmente, para poner en marcha el sistema, a la vez que se valida nuestra propuesta, hemos realizado la implantación de los dispositivos en una fábrica de quesos en la provincia de Zamora, en concreto, en la fábrica Marcos Conde, una industria quesera de reconocido prestigio, que se ha consolidado como una de las más importantes en su sector.

Capítulo 4

Mecanismos de filtrado de EMIs: Data Base Filter y Smart Coded Filter

Como hemos mostrado en el capítulo anterior, para medir las variables utilizadas y así estimar de forma precisa el factor OEE con dispositivos low-cost, resulta imprescindible el filtrado de EMIs. En concreto, se precisa la eliminación del ruido electromagnético en las señales de sellado y disponibilidad, para así obtener valores fiables de los diferentes parámetros que conforman el OEE, y poder realizar una correcta estimación del rendimiento total de los equipos.

En este capítulo se presentan los dos mecanismos que se han diseñado para el filtrado de EMIs, el Data Base Filter (DBF), aplicado a las señales de sellado, y el Smart Coded Filter (SCF), aplicado a las señales de disponibilidad. Además, se valida el correcto funcionamiento de los mismos, se presenta cómo funcionan los dos mecanismos en conjunto, y se comentan en detalle los resultados obtenidos.

4.1. Introducción

Los dos mecanismos de filtrado de EMIs, que se presentan como aportación fundamental de la presente Tesis Doctoral, son el DBF, encargado de filtrar las señales de sellado de los productos elaborados, y el SCF, que permite captar correctamente las señales relativas a la disponibilidad de las máquinas termoselladoras.

A continuación, se comentan brevemente las diferentes secciones del presente capítulo. La Sección 4.2 muestra las características del mecanismo DBF, la estructura del algoritmo, su funcionamiento, el análisis del offset y los resultados obtenidos una vez aplicado el DBF a todos los turnos analizados. En la Sección 4.3 se muestra el funcionamiento del mecanismo SCF. Más concretamente, el algoritmo que se utiliza para realizar el filtrado de EMIs en las señales de disponibilidad y los resultados del mecanismo tras aplicarlo en los seis turnos analizados. En la Sección 4.4 se ofrece una visión en conjunto de la propuesta de ambos mecanismos,

es decir, cómo funciona el sistema aplicando los dos mecanismos (DBF y SCF), de manera conjunta. Por último, la Sección 4.5 incluye las conclusiones del trabajo presentado en este capítulo.

4.2. Mecanismo DBF: Filtrado de EMIs en señales de sellado

El Data Base Filter (DBF) es el mecanismo que hemos diseñado para mitigar la presencia de señales erróneas, debidas a las EMIs, en las señales de sellado. Este mecanismo realiza un filtrado de las señales procedentes del ruido electromagnético, permitiendo determinar las señales válidas, y así mejorar la precisión de nuestro sistema al estimar el OEE. Para ello, se ha tenido en cuenta el comportamiento y naturaleza de la señal de sellado.

Las señales de sellado siguen un patrón de repetición, en lo que a tiempos de sellado se refiere, y es en este aspecto en el que se centra el mecanismo DBF. El mecanismo analiza los datos recogidos, determinando el valor que más se repite, es decir, la moda, y posteriormente aplica un offset por encima y por debajo de ese valor. Los datos que se encuentran dentro del intervalo, serán los filtrados como señales válidas y los que estén fuera serán considerados como señales erróneas.

4.2.1. Funcionamiento del mecanismo de filtrado DBF

Las señales, tras ser almacenadas en la base de datos, son procesadas por el mecanismo de filtrado (ver Algoritmo 2). Dicho mecanismo, descarta las señales erróneas, atendiendo al cumplimiento del orden lógico de los valores que conforman una señal de sellado, no pudiendo considerarse una señal válida si se repiten valores lógicos seguidos, es decir, un 0 seguido de otro 0, o dos 1's consecutivos (*filterLogicSignals*). Este proceso aparece detallado en el Algoritmo 3).

Posteriormente, se filtran también las señales que tienen una duración menor de un segundo (*filterCandidateSignals*), tal y como refleja el Algoritmo 4, dado que la puesta en marcha del sistema nos ha demostrado que ninguna señal de sellado termoplástico dura menos de un segundo.

Una vez que el DBF ha realizados estos dos pasos, obtiene las franjas horarias de inicio y fin de cada producto, que han sido previamente almacenadas en la base de datos, de forma manual, por los operarios (*getProductTimesDB*).

Después, el DBF genera un array denominado *signalsByProduct*, que contiene las señales de cada producto, pudiendo así identificar el valor del tiempo de sellado que más se repite en el proceso de sellado (*mode*).

Finalmente, se establecen los límites inferior y superior aplicando el offset fijado, y una vez que se determinan los límites, se realiza el filtrado con este criterio y se obtienen las señales consideradas válidas de cada producto (*dbfSignals*), que serán las señales que tienen un tiempo de sellado dentro del intervalo fijado. El valor del offset se ha establecido tras un análisis exhaustivo de las operaciones de sellado y sus tiempos de duración (ver subsección 4.2.1.1), de forma que se minimice el error al filtrar las señales erróneas.

4.2. MECANISMO DBF: FILTRADO DE EMIS EN SEÑALES DE SELLADO

Algoritmo 2 Data Base Filter (DBF)

```
/* shiftSignals: array which includes all the signals collected by the thermosealing machine */
/* logicSignals: array which includes the signals filtered by a logic order (0-1) */
/* prefilteredSignals: array includes all the signals except those lower than one second */
/* productTimes: array which includes product start and end times */
/* signalsByProduct: array which includes signals for each product */
/* mode: most repeated sealing time */
/* offset: time added or subtracted to the mode to determine the valid sealings */
/* maxLimit: upper offset limit */
/* minLimit: lower offset limit */
/* dbfSignals: array which includes all the valid sealing signals considered by aDBF */
```

```
logicSignals[] = filterLogicSignals(shiftSignals[]);

prefilteredSignals[] = filterCandidateSignals(logicSignals[]);

productTimes[] = getProductTimesDB(shift);

foreach product in productTimes[] do
    signalsByProduct[] = getProductSignals(prefilteredSignals[],
    product.TimeStart, product.TimeEnd);
    mode = getMode(signalsByProduct[]);

    foreach signal in signalsByProduct[] do
        maxLimit = mode + offset;
        minLimit = mode - offset;
        if (signal < maxLimit) && (signal > minLimit) then
            dbfSignals.add(signal);
        end
    end
end
return dbfSignals;
```

Algoritmo 3 Filter Logic Signals

```
/* filteredLogicSignals: array which includes the signals filtered by a logic order (0-1) */

signal[] filterLogicSignals(allSignals[])
for (i=1; i<allSignals.size(); i++) do
    currentState = allSignals[i].state;
    previousState = allSignals[i-1].state;
    if (currentState != previousState) then
        filteredLogicSignals.add(currentState);
    end
end
return filteredLogicSignals;
```

CAPÍTULO 4. MECANISMOS DE FILTRADO DE EMIS: DATA BASE FILTER Y SMART CODED FILTER

Algoritmo 4 Filter Candidate Signals

```
/* candidateSignals: array which includes all the signals except those lower than one second */  
  
signal[] filterCandidateSignals(filteredLogicSignals[])  
integer duration;  
for (i=1; i<filteredLogicSignals.size(); i++) do  
    duration = filteredLogicSignals[i-1].timestamp -  
    filteredLogicSignals[i].timestamp;  
    if (duration > 1) then  
        | candidateSignals.add(logicSignals[i]);  
    end  
end  
return candidateSignals;
```

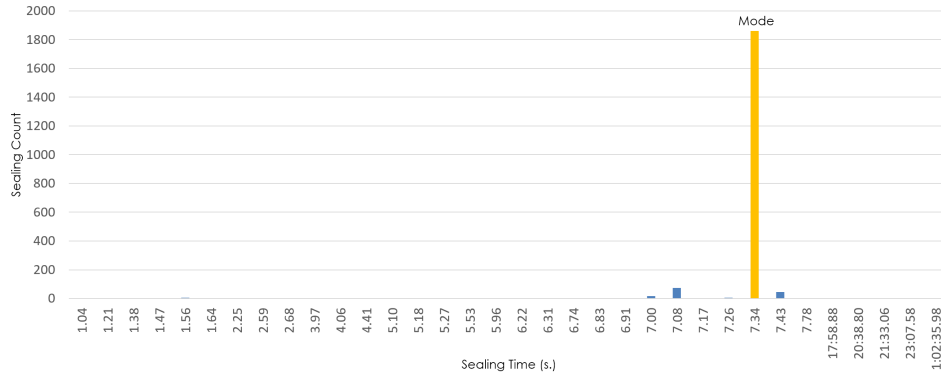


Figura 4.1: Ejemplo de la determinación de la moda en un turno.

A modo de ejemplo, la Figura 4.1 muestra el cálculo de la moda del producto A en el turno 1. En ella se observa el valor de la moda del producto que se ha elaborado en el turno (7.34 segundos), una vez eliminadas las señales inferiores a 1 segundo. Además, a partir de los valores que se muestran, pueden observarse las señales que podrían estar incluidas dentro del intervalo considerado (suponiendo un offset del 13.04%), es decir, las señales con tiempos de sellado comprendidos entre 6.38 y 8.30 segundos. Los valores fuera de dicho intervalo, se considerarán ruido electromagnético, por lo que no se tendrán en cuenta como señales válidas.

4.2.1.1. Análisis del Offset

El valor del offset aplicado sobre el valor de la moda tiene como objetivo obtener las señales válidas, descartando las señales debidas a las EMIs. Dicho valor se ha establecido tras el análisis exhaustivo de las operaciones de sellado, sus tiempos de duración y los errores que se producen en cada uno de los turnos y productos. En concreto, hemos analizado los resultados del proceso de sellado de tres productos

4.2. MECANISMO DBF: FILTRADO DE EMIS EN SEÑALES DE SELLADO

en seis turnos diferentes. Los productos son: cuartos de quesos (Product A), quesos (Product B) y cuñas de queso (Product C). De no realizar un filtrado de EMIs, la incidencia de éstas no permitiría un cálculo preciso del rendimiento, y por ende del OEE, puesto que el número de señales erróneas debido a las EMIs es elevado.

La idea fundamental es quedarnos, de todo el conjunto de señales recogidas, con aquéllas que se corresponden con la operación real. Para ello, debemos fijar un valor de offset que genere el mínimo error en todos los turnos y productos. El análisis se ha realizado teniendo en cuenta las desviaciones existentes en la duración de los sellados, buscando en cada uno de los productos los valores de offset adecuados para que el error resultase 0%; más concretamente, para poder obtener todas las señales válidas, es decir, las que se corresponden con la operación real y desechar las producidas por las EMIs.

A modo de ejemplo, la Tabla 4.1 muestra el análisis de offset para el producto C en el turno 5. Para determinar el offset necesario que recoja todas las señales reales, se calcula la moda de los tiempos de sellado, siendo ésta 6.048 segundos, ya que es el valor más repetido (325 veces), tal y como se observa en la segunda columna. A partir de ese valor, se calcula el offset necesario para recoger señales por encima y por debajo de la moda, calculándose el error que se produce con cada porcentaje de offset propuesto. En este caso, el offset para que el error sea 0 % se establece en un 11.43 %, puesto que aplicando este valor se recogen las 441 reales.

Con el objetivo de encontrar un offset común, se ha realizado este mismo análisis para cada turno y producto, es decir, se ha calculado el porcentaje de offset que permite tener un 0 % de error en cada caso. Además, se han analizado otros porcentajes de error para calcular el offset que mejor se adecúe a nuestro mecanismo DBF. La Tabla 4.2 muestra el análisis para alcanzar los siguientes errores: <5 %, <1 %, <0.5 % y 0 %.

En la columna 2 de la Tabla 4.2, se observa que el porcentaje máximo de offset, cuando buscamos un 0 % de error para cada turno y producto, es del 61.43 %, que corresponde al turno 3, producto C; el resto de turnos y productos necesitarían un porcentaje inferior para obtener un 0 % de error.

Por su parte, si se quisiera obtener un porcentaje de error menor del 0.5 %, entonces el offset necesario sería el 45.88 %, el valor correspondiente en este caso al turno 3, producto A.

Para alcanzar un error menor del 1 % en todos los turnos y productos, entonces se necesitaría un offset del 13.04 %, que es el valor que se corresponde con el turno 2, producto C.

Por último, para obtener un error menor del 5 % en todos los turnos y productos, el offset que debería aplicarse sería el 7.25 %, que corresponde al valor máximo que se presenta en la columna 7, turno 2, producto C.

Una vez obtenidos los porcentajes máximos de offset, en función del error máximo que estemos en condiciones de asumir para el cálculo del OEE, se analizan en detalle los valores de error obtenidos, de acuerdo a los porcentajes de offset anteriormente comentados, es decir, 7.25 %, 13.04 %, 45.88 % y 61.43 %. La Tabla 4.3 muestra este análisis para cada turno y cada producto. Además, muestra el valor de error máximo en cada caso, el valor medio, la desviación típica y la Raíz del Error Cuadrático Medio (RMSE).

CAPÍTULO 4. MECANISMOS DE FILTRADO DE EMIS: DATA BASE FILTER Y SMART CODED FILTER

Tabla 4.1: Ejemplo de valores recogidos que permiten determinar el offset necesario para tener un error del 0 % para el producto C en el turno 5

Sealing Time	Frequency	Offset	Error
1.469	2	75.71 %	0.91 %
1.814	1	70.00 %	0.45 %
2.074	1	65.71 %	0.23 %
5.357	1	11.43 %	0.00 %
5.530	4	8.57 %	0.23 %
5.616	1	7.14 %	1.13 %
5.702	2	5.71 %	1.36 %
5.789	13	4.29 %	1.81 %
5.875	6	2.86 %	4.76 %
5.962	50	1.43 %	7.48 %
6.048	325	0.00 %	26.30 %
6.134	33	1.43 %	7.48 %
6.221	6	2.86 %	4.76 %
19.786	1	227.14 %	1.13 %
51.926	1	758.57 %	1.36 %
2:28.781	1	2360.00 %	1.59 %
19:03.677	1	18810.00 %	1.81 %

Se observa que aplicando el 7.25 % de offset para todos los turnos y productos, se obtiene un error máximo del 3.76 %, concretamente el error cometido para el turno 2, producto C; además de un error medio del 0.95 %, con una desviación típica de 1.00 y una raíz del error cuadrático medio de 11.81.

Por su parte, al aplicar un offset del 13.04 %, entonces el error máximo se cometería en el turno 4, producto A, teniendo éste un valor del 0.93 %; el error medio es del 0.28 %, la desviación típica es de 0.31 y la raíz del error cuadrático medio es de 4.26. Destaca, para este porcentaje de offset, el número de productos con un error del 0 % que existen en los diferentes turnos y productos, más concretamente, son seis los productos en los que el mecanismo de filtrado actúa de manera precisa, eliminando correctamente todas las señales erróneas, pero sin eliminar ninguna señal de sellado real.

Los resultados de aplicar un offset del 45.88 % pueden verse en las columnas 9 y 10 de la Tabla 4.3. Concretamente, los valores que ofrece el mecanismo DBF al aplicar este porcentaje de offset son un error máximo de 0.26 %, un error medio de 0.12 %, una desviación típica de 0.09 y una raíz del error cuadrático medio de 1.12. En este caso, aunque el valor del error máximo es menor que cuando se aplica un 13.04 % de offset, el número de celdas con un error del 0 % disminuye en comparación con el anterior offset.

Por último, las columnas 11 y 12 de la Tabla 4.3, muestran las señales filtradas y el error cometido al aplicar el mayor de los offset analizados (61.43 %). En este caso, el valor del error máximo es del 0.58 % (mayor que en el caso anterior), el error medio es del 0.18 %, la desviación típica es de 0.16 y la raíz del error

4.2. MECANISMO DBF: FILTRADO DE EMIS EN SEÑALES DE SELLADO

Tabla 4.2: Cálculo del offset para valores de error 0 %, <0.5 %, <1 % y <5 %

Shift	Error = 0 %			Error <0.5 %			Error <1 %			Error <5 %		
	Offset (%)	Error (%)	Error (%)	Offset (%)	Error (%)	Error (%)	Offset (%)	Error (%)	Error (%)	Offset (%)	Error (%)	Error (%)
1 Product A	44.71 %	0.00 %	0.00 %	8.24 %	0.44 %	0.44 %	4.71 %	0.94 %	0.94 %	3.53 %	1.73 %	1.73 %
2 Product A	7.06 %	0.00 %	0.00 %	5.88 %	0.39 %	0.39 %	4.71 %	0.66 %	0.66 %	3.53 %	2.49 %	2.49 %
2 Product B	8.25 %	0.00 %	0.00 %	6.19 %	0.26 %	0.26 %	5.15 %	0.77 %	0.77 %	3.09 %	2.06 %	2.06 %
2 Product C	39.13 %	0.00 %	0.00 %	15.94 %	0.44 %	0.44 %	13.04 %	0.66 %	0.66 %	7.25 %	3.76 %	3.76 %
2 Product B	10.31 %	0.00 %	0.00 %	10.31 %	0.00 %	0.00 %	9.28 %	0.62 %	0.62 %	3.09 %	2.47 %	2.47 %
3 Product B	4.12 %	0.00 %	0.00 %	3.09 %	0.19 %	0.19 %	3.09 %	0.19 %	0.19 %	1.03 %	1.15 %	1.15 %
3 Product C	61.43 %	0.00 %	0.00 %	24.29 %	0.35 %	0.35 %	12.86 %	0.53 %	0.53 %	5.71 %	3.33 %	3.33 %
3 Product A	61.18 %	0.00 %	0.00 %	45.88 %	0.14 %	0.14 %	11.76 %	0.87 %	0.87 %	3.53 %	4.92 %	4.92 %
4 Product A	32.94 %	0.00 %	0.00 %	23.53 %	0.31 %	0.31 %	12.94 %	0.93 %	0.93 %	3.53 %	3.32 %	3.32 %
4 Product B	32.99 %	0.00 %	0.00 %	14.43 %	0.49 %	0.49 %	9.28 %	0.91 %	0.91 %	4.12 %	3.95 %	3.95 %
5 Product B	11.34 %	0.00 %	0.00 %	6.19 %	0.39 %	0.39 %	6.19 %	0.39 %	0.39 %	1.03 %	3.81 %	3.81 %
5 Product A	4.71 %	0.00 %	0.00 %	3.53 %	0.23 %	0.23 %	3.53 %	0.23 %	0.23 %	0.00 %	4.12 %	4.12 %
5 Product C	11.43 %	0.00 %	0.00 %	8.57 %	0.23 %	0.23 %	8.57 %	0.23 %	0.23 %	2.86 %	4.76 %	4.76 %
6 Product C	5.80 %	0.00 %	0.00 %	5.80 %	0.00 %	0.00 %	4.35 %	0.91 %	0.91 %	2.90 %	2.60 %	2.60 %
6 Product B	1.03 %	0.00 %	0.00 %	1.03 %	0.00 %	0.00 %	1.03 %	0.00 %	0.00 %	1.03 %	0.00 %	0.00 %
6 Product A	12.94 %	0.00 %	0.00 %	8.24 %	0.36 %	0.36 %	7.06 %	0.73 %	0.73 %	4.71 %	1.45 %	1.45 %

CAPÍTULO 4. MECANISMOS DE FILTRADO DE EMIS: DATA BASE FILTER Y SMART CODED FILTER

Tabla 4.3: Resultados de DBF con diferentes valores de offset

Shift	Real signals	RAW		DBF 7.25 % offset		DBF 13.04 % offset		DBF 45.88 % offset		DBF 61.43 % offset	
		Received signals	Erroneous signals	Filtered signals	Error	Filtered signals	Error	Filtered signals	Error	Filtered signals	Error
1 Product A	2023	5697	64.49 %	2010	0.64 %	2014	0.45 %	2023	0.00 %	2025	0.10 %
2 Product A	702	1394	45.34 %	762	0.00 %	762	0.00 %	763	0.13 %	764	0.26 %
2 Product B	388	505	23.17 %	387	0.26 %	388	0.00 %	388	0.00 %	389	0.26 %
2 Product C	904	1166	22.47 %	870	3.76 %	898	0.66 %	904	0.00 %	904	0.00 %
2 Product B	486	851	42.89 %	480	1.23 %	487	0.21 %	487	0.21 %	487	0.21 %
3 Product B	520	717	27.48 %	521	0.19 %	521	0.19 %	521	0.19 %	523	0.58 %
3 Product C	571	971	41.19 %	560	1.93 %	568	0.53 %	570	0.18 %	571	0.00 %
3 Product A	691	1247	44.59 %	678	1.88 %	687	0.58 %	690	0.14 %	691	0.00 %
4 Product A	965	1750	44.86 %	950	1.55 %	956	0.93 %	964	0.10 %	967	0.21 %
4 Product B	1214	2487	51.19 %	1198	1.32 %	1206	0.66 %	1216	0.16 %	1216	0.16 %
5 Product B	702	1650	53.82 %	759	0.39 %	762	0.00 %	764	0.26 %	765	0.39 %
5 Product A	437	609	28.24 %	438	0.23 %	438	0.23 %	438	0.23 %	438	0.23 %
5 Product C	441	759	41.90 %	436	1.13 %	441	0.00 %	441	0.00 %	441	0.00 %
6 Product C	883	949	7.06 %	883	0.00 %	884	0.11 %	885	0.23 %	885	0.23 %
6 Product B	30	31	3.23 %	30	0.00 %	30	0.00 %	30	0.00 %	30	0.00 %
6 Product A	1101	1790	38.49 %	1093	0.73 %	1101	0.00 %	1102	0.09 %	1103	0.18 %
Maximum			64.49 %		3.76 %		0.93 %		0.26 %		0.58 %
Average			36.28 %		0.95 %		0.28 %		0.12 %		0.18 %
Std Deviation			16.53		1.00		0.31		0.09		0.16
RMSE			1,068.18		11.81		4.26		1.12		1.68

4.2. MECANISMO DBF: FILTRADO DE EMIS EN SEÑALES DE SELLADO

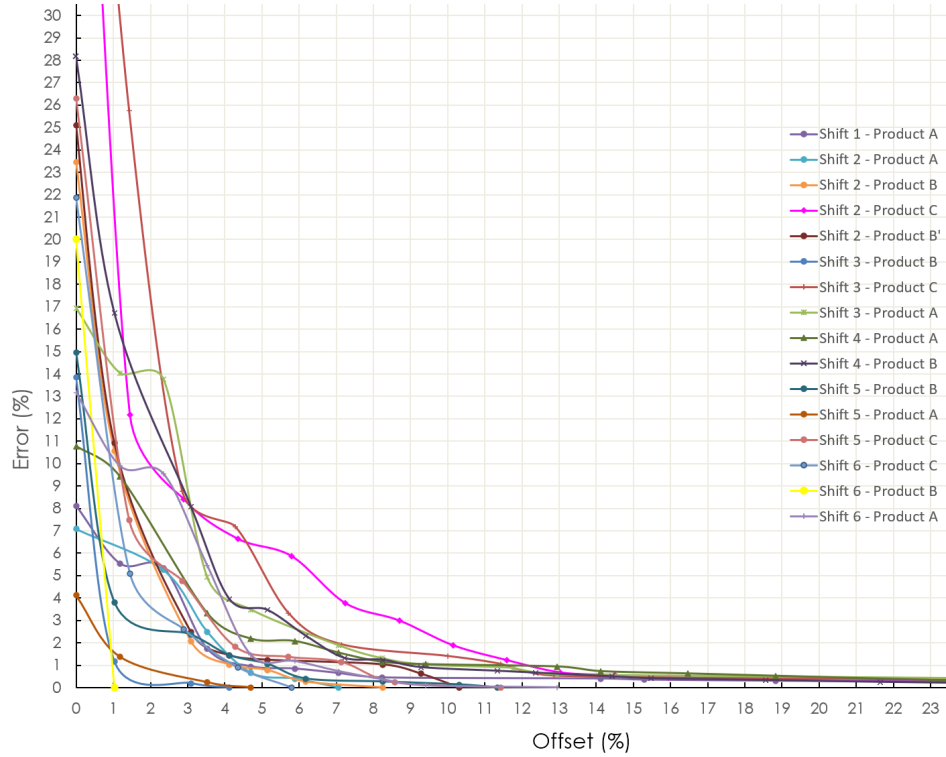


Figura 4.2: Error cometido durante el proceso de filtrado al variar el offset, para todos los productos y turnos analizados.

cuadrático medio es de 1.68, un valor ligeramente mayor que cuando se aplica el offset 45.88 %.

El análisis de offset realizado, permite tomar una decisión para aplicar un offset que pueda utilizarse en todos los turnos y productos, que a su vez sea el más ventajoso en cuanto a error y señales válidas detectadas, teniendo en cuenta un equilibrio entre la magnitud del offset y el error cometido. Por ello, el offset que se ha considerado aplicar es un 13.04 %, asumiendo un error máximo del 0.93 % en nuestro sistema, un error medio de 0.28 % y un RMSE de 4.26. A pesar de que se podría alcanzar un error menor con otros offset mayores, se considera que el beneficio conseguido con el aumento del offset no es significativo con respecto al error cometido, y además, se incrementaría el número de turnos y productos en los que no se obtiene un error del 0 %, por lo que se considera adecuado usar el 13.04 % como offset general.

La elección de un offset con valor 13.04 %, se ve reforzada con la Figura 4.2, la cual muestra, de forma visual, los porcentajes de error cometidos y los porcentajes de offset necesarios para ello. Se puede observar que si utilizamos un offset inferior al 13.04 % el error cometido sería mayor al 1 % en casi todos los turnos y productos, mientras que si usamos un offset mayor, el error cometido apenas se reduce.

CAPÍTULO 4. MECANISMOS DE FILTRADO DE EMIS: DATA BASE FILTER Y SMART CODED FILTER

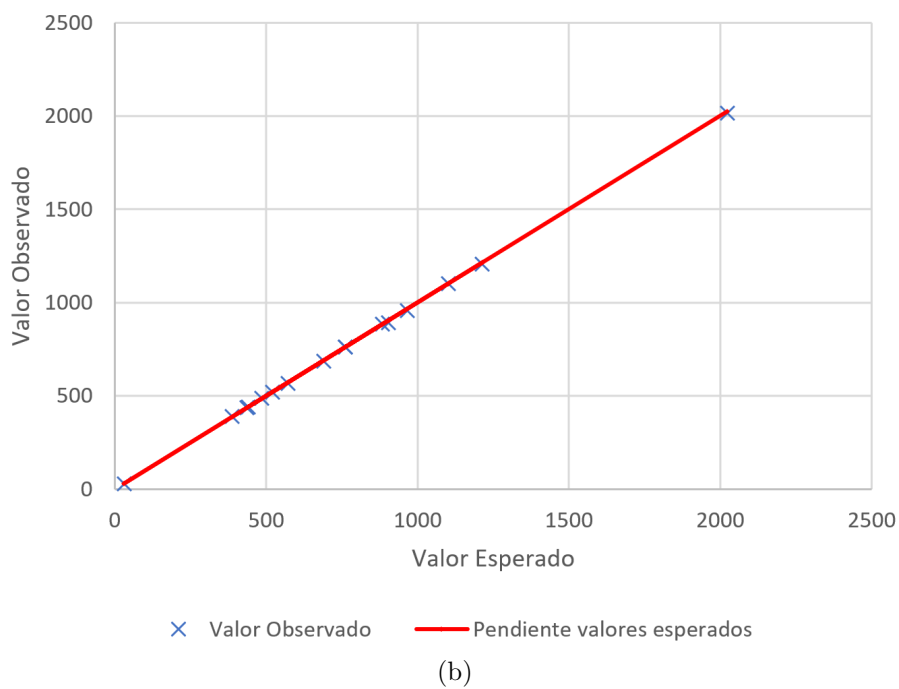
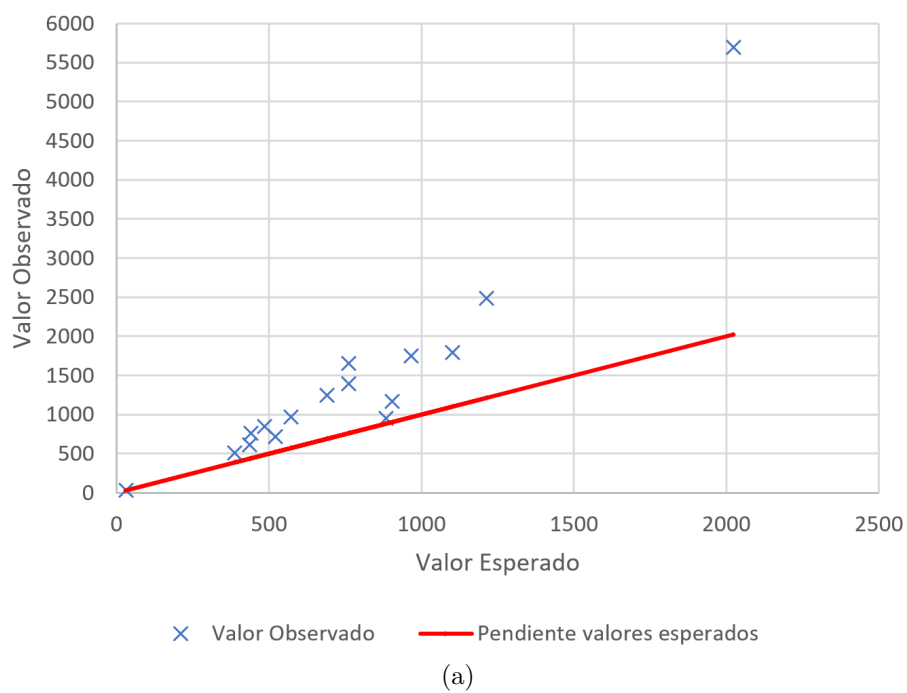


Figura 4.3: RMSE: (a) sin offset (RAW) y (b) offset elegido (13.04%).

4.2. MECANISMO DBF: FILTRADO DE EMIS EN SEÑALES DE SELLADO

La Figura 4.3 representa la RMSE del análisis realizado sin offset (RAW) y con el offset elegido del 13.04 %, para todos los turnos y productos. En concreto, la Figura 4.3(a) representa la distribución de las señales con respecto a las señales sin filtrar. En ella se puede observar la diferencia que existe entre los valores esperados, es decir, los valores reales (línea roja) y los puntos obtenidos en la medición (puntos azules). La Figura 4.3(b) representa la distribución de las señales con respecto al mecanismo de filtrado con un offset del 13.04 %, el elegido para aplicar a todos los turnos y productos. En este caso los valores medidos se aproximan mucho más a los valores esperados, es decir, las mediciones realizadas son más cercanas a los valores reales.

Por su parte, la Tabla 4.4 muestra los datos correspondientes a los resultados de cada turno y producto, una vez aplicado el mecanismo DBF con un offset del 13.04 %. La primera columna muestra el turno y el producto trabajado. La segunda columna muestra los datos de las señales reales, recogidas en fábrica. En las columnas 3 y 4, se muestran las señales totales recibidas en el dispositivo y el error producido, teniendo en cuenta las interferencias electromagnéticas. Por último, las columnas 5 y 6, muestran las señales que han sido obtenidas por el mecanismo DBF y el error cometido. De acuerdo a lo anteriormente comentado, el DBF considera que las señales cuyos tiempos de sellado están alejados de la moda (en un porcentaje mayor al offset), han sido generadas por ruido electromagnético.

Como se puede observar, el mecanismo DBF filtra la totalidad de las señales erróneas debidas a las EMIs, aunque en algunas ocasiones filtramos también señales válidas. No obstante, el error cometido sin usar el DBF es de hasta un 64.49 %, mientras que el error máximo, una vez filtradas las señales, es del 0.93 %, lo que sin duda nos va a permitir estimar el OEE de forma precisa.

4.2.2. Resultados del mecanismo de filtrado DBF

En esta sección, se presentan los resultados del mecanismo de filtrado DBF. En la subsección 4.2.2.1 se realiza un análisis de los datos reales de todos los turnos y productos, de los recibidos con EMIs y las señales filtradas por el mecanismo DBF con el offset fijado. Además, se analizan los cálculos de los valores de la moda para todos los turnos y productos. La subsección 4.2.2.2 muestra los resultados del mecanismo DBF en el proceso de sellado, analizando cada turno y producto en su orden de elaboración. Por último, la subsección 4.2.2.3 muestra el agrupamiento de las señales de sellado, así como las señales erróneas debidas a las interferencias electromagnéticas.

4.2.2.1. Análisis de las modas

La Tabla 4.5 contiene los valores de las modas calculados para cada turno y producto. En ella se observa que los valores de los productos A y B se repiten, es decir, son iguales en todos los turnos, variando solamente el producto C en 9 centésimas, puesto que el valor de la moda oscila entre los 5.96 y los 6.05 segundos.

En las Figuras 4.4-4.9 se analizan todos los turnos estudiados con sus correspondientes gráficas de cada producto que ha sido elaborado en el turno. En ellas se observan las modas calculadas y los valores de las señales que están en torno a

CAPÍTULO 4. MECANISMOS DE FILTRADO DE EMIS: DATA BASE FILTER Y SMART CODED FILTER

Tabla 4.4: Señales recogidas por el mecanismo DBF y errores cometidos con un offset del 13.04 %

Shift	Real signals	RAW		DBF (offset 13.04 %)	
		Received signals	Error	Filtered signals	Error
1 Product A	2023	5697	64.49 %	2014	0.45 %
2 Product A	762	1394	45.34 %	762	0.00 %
2 Product B	388	505	23.17 %	388	0.00 %
2 Product C	904	1166	22.47 %	898	0.66 %
2 Product B	486	851	42.89 %	487	0.21 %
3 Product B	520	717	27.48 %	521	0.19 %
3 Product C	571	971	41.19 %	568	0.53 %
3 Product A	691	1247	44.59 %	687	0.58 %
4 Product A	965	1750	44.86 %	956	0.93 %
4 Product B	1214	2487	51.19 %	1206	0.66 %
5 Product B	437	609	28.24 %	438	0.23 %
5 Product A	762	1650	53.82 %	762	0.00 %
5 Product C	441	759	41.90 %	441	0.00 %
6 Product C	1101	1790	38.49 %	1101	0.00 %
6 Product B	30	31	3.23 %	30	0.00 %
6 Product A	883	949	7.06 %	884	0.11 %

estas modas, para así determinar el filtrado que se realizaría con el offset fijado. Observando en detalle dichas figuras, así como los datos presentados en la Tabla 4.4, podemos ver claramente el funcionamiento del DBF, así como los resultados obtenidos y el error absoluto cometido.

En la Figura 4.4 se observa que se ha elaborado únicamente un producto durante el turno 1. Este producto tiene una moda de 7.34 segundos, correspondiéndose con el producto A, cuartos de queso. Realizando las operaciones con el offset fijado (13.04 %) se agrupan los sellados desde los 6.38 segundos de límite inferior, a los 8.30 segundos en su límite superior, considerando todas ellas válidas. La práctica totalidad de las señales de sellado se encuentran en ese intervalo; solamente un porcentaje muy pequeño de señales correctas no son tenidas en cuenta por el mecanismo DBF. Concretamente, 9 señales de un total de 2023, no son recogidas por nuestro sistema, lo que supone un error absoluto de 0.45 %.

La Figura 4.5(a) muestra el valor de la moda calculada en el producto A en el turno 2. Aplicando el offset fijado, el valor mínimo en tiempos de sellado se establece en 6.38 segundos y el valor máximo 8.30 segundos. De esta forma todos los valores que se sitúen en este intervalo serán considerados como señales válidas. Los valores que se observan inferiores al offset mínimo, tales como 1.04, 1.47, 1.56, 2.33, 3.63 y 5.01 serán considerados como ruido. En el caso de los valores superiores, todos están dentro del offset, ya que el sistema no ha captado ninguna señal superior a 7.43 segundos, siendo la moda de este producto 7.34 segundos. En la Figura 4.5(b) se observan los registros relativos al producto B en el turno 2. La

4.2. MECANISMO DBF: FILTRADO DE EMIS EN SEÑALES DE SELLADO

Tabla 4.5: Modas calculadas para cada turno y producto

Shift	Mode
1 Product A	7.34
2 Product A	7.34
2 Product B	8.38
2 Product C	5.96
2 Product B	8.38
3 Product B	8.38
3 Product C	6.05
3 Product A	7.34
4 Product A	7.34
4 Product B	8.38
5 Product B	8.38
5 Product A	7.34
5 Product C	6.05
6 Product C	5.96
6 Product B	8.38
6 Product A	7.34

moda tiene un valor de 8.38 segundos y aplicando el offset, los valores de sellado válidos serían los comprendidos entre 7.29 y 9.47 segundos, viéndose de manera clara que el límite inferior recoge todos los sellados válidos, puesto que los valores a partir de 3.97 hasta 1.12 son considerados como interferencias electromagnéticas. El producto C durante el turno 2 es mostrado en la Figura 4.5(c), en la que se representa la moda con un valor de 5.96 segundos. Si aplicamos el offset en este producto los valores mínimos se sitúan en 5.18 segundos y el valor máximo es de 6.74 segundos. En el offset superior, todos los sellados se consideran como válidos, no siendo así al aplicar el offset inferior, ya que los valores inferiores a 5.18 segundos se quedan fuera de la consideración de sellados válidos.

En el caso de la Figura 4.6(a) la moda se sitúa en 7.34 segundos, quedando dentro de los sellados válidos los comprendidos entre 6.38 segundos y 8.30 segundos, considerándose como ruido todo lo que esté por encima o por debajo de esos valores límite. Se observa que existen algunas señales que no recoge nuestro dispositivo, pero son mínimas. En el caso de los valores que existen entre 1.47 segundos y 3.97 segundos se toman como ruido, no considerándose válidas. La Figura 4.6(b), representa la moda obtenida que es 8.38 segundos (producto B), y una vez aplicado el offset, los valores descartados son los que se encuentran por debajo de 7.29 segundos y por encima de 9.47 segundos. En este caso, el offset funciona perfectamente, puesto que todos los valores que se encuentran en el intervalo del offset estarían dentro de los considerados válidos. En la Figura 4.6(c) se determina la moda del producto C en el turno 3, quedando estipulado el tiempo de sellado en 6.05 segundos. Una vez aplicado el offset, los valores que se encuentren por debajo de 5.26 segundos y por encima de 6.84 segundos, serán considerados como no válidos. De esta forma, el DBF permite eliminar el ruido producido por las EMIs en las señales de sellado.

La Figura 4.7 presenta los datos relativos al turno 4. En la Figura 4.7(a) se

CAPÍTULO 4. MECANISMOS DE FILTRADO DE EMIS: DATA BASE FILTER Y SMART CODED FILTER

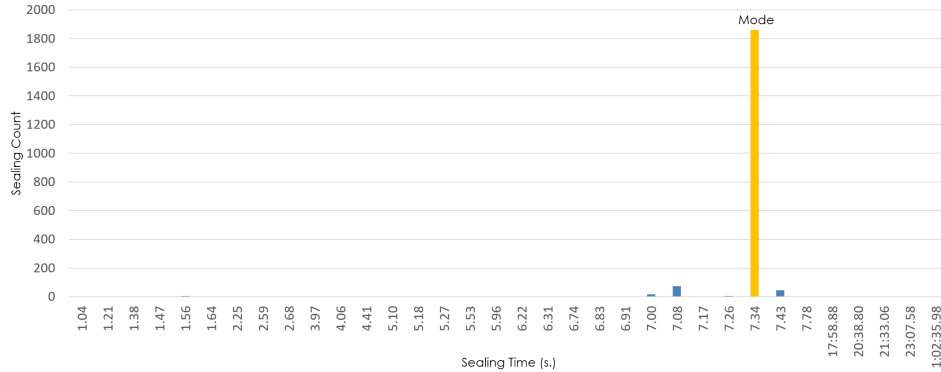


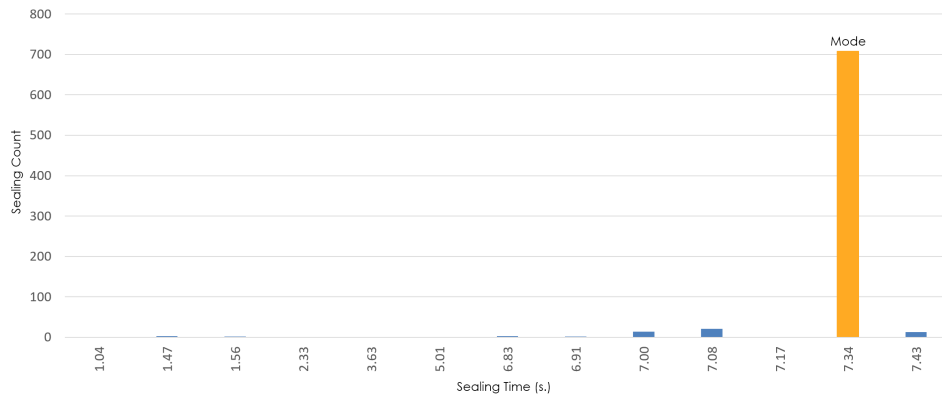
Figura 4.4: Determinación de la moda para el producto A en el turno 1.

observa, para el caso del producto A, una moda de 7.34 segundos y unos valores mínimos y máximos del offset de 6.38 y 8.30 segundos, respectivamente. En este producto se recogen la mayoría de las señales, 956 de las 965 reales, es decir, se produce un error absoluto del 0.93 %. Concretamente, el mecanismo filtra de forma incorrecta 9 señales de sellado que son válidas. La Figura 4.7(b) representa el cálculo de la moda del producto B, siendo el valor de esta 8.38 segundos. Aplicando el offset, las señales que se captan son las comprendidas entre 7.29 y 9.47 segundos.

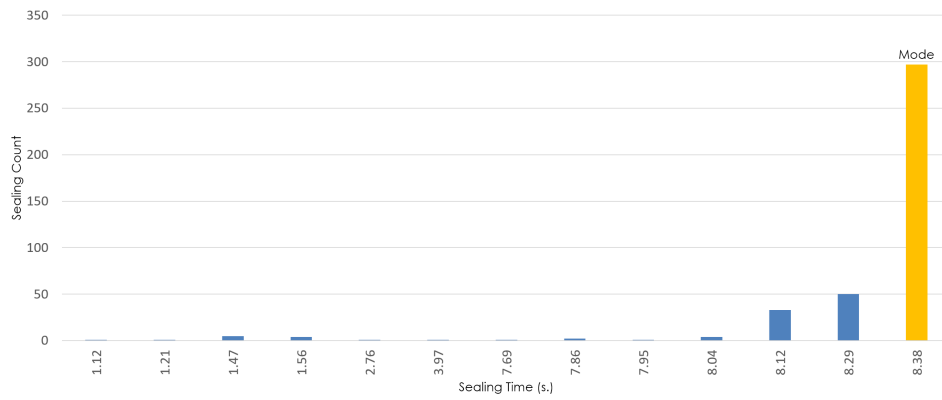
La Figura 4.8(a) muestra el valor de la moda calculada en el producto A en el turno 5. Si aplicamos el offset fijado a estos valores, se marcan los límites de 6.38 segundos como inferior y 8.30 como límite superior, quedando fuera de los valores de sellado válidos, todas aquellas señales que se encuentren fuera del rango. De esta forma, los valores de 1.47 segundos que se aprecian, quedan fuera de las señales válidas, así como las señales que se sitúan por encima de los 8.30 segundos. En la Figura 4.8(b) se muestran las señales obtenidas para el producto B en el turno 5. En este caso, la moda se fija en 8.38 segundos, y por tanto, todos los valores que estén comprendidos entre 7.29 y 9.47 serán considerados válidos. Aquí se demuestra que el sistema funcionaría sin errores, puesto que todas las señales que se aprecian desde 1.12 segundos hasta 4.84 segundos se consideran ruido. Además, todas las señales superiores a 9.47 se descartan, viéndose claramente que no pertenecen a las señales de sellado válidas. La Figura 4.8(c) representa la moda del producto C en el turno 5, la cual se establece con el valor 6.05 segundos. Por tanto, las señales que se pueden observar entre 1.47 segundos y 2.07 segundos, además de las superiores a 6.84 segundos, no serán consideradas como válidas.

Por último, la Figura 4.9(a) representa el producto A en el turno 6. Este producto tiene una moda de 7.34 y aplicando el offset fijado se obtienen 1101 señales válidas, es decir se recogen todas y cada una de las señales válidas, y se omiten las señales producidas por las EMIs. La Figura 4.9(b) representa el producto B en el turno 6. Este producto tiene una moda de 8.38 segundos y es el producto que menos señales ha recogido de todos los analizados, concretamente 31 en total, y al igual que en el anterior producto, el mecanismo de filtrado obtiene de forma

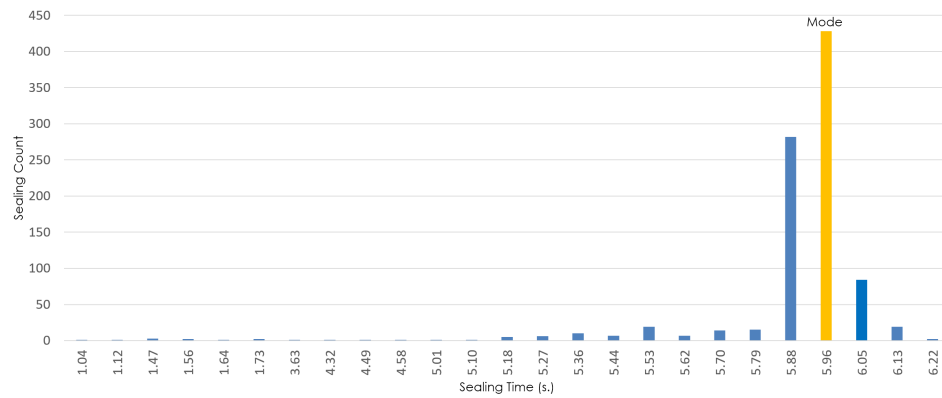
4.2. MECANISMO DBF: FILTRADO DE EMIS EN SEÑALES DE SELLADO



(a)



(b)



(c)

Figura 4.5: Determinación de la moda: (a) producto A, (b) producto B y (c) producto C en el turno 2.

CAPÍTULO 4. MECANISMOS DE FILTRADO DE EMIS: DATA BASE FILTER Y SMART CODED FILTER

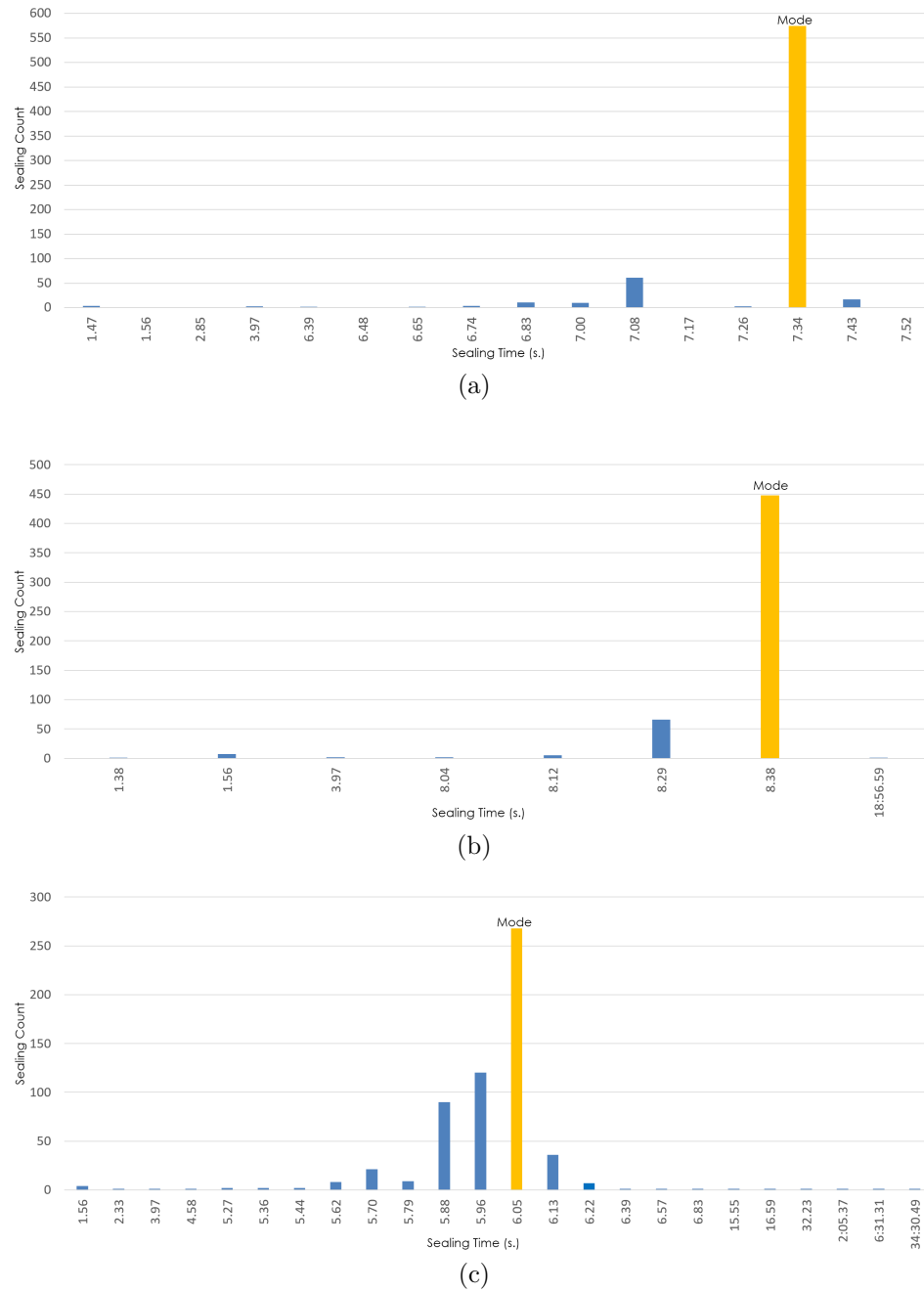
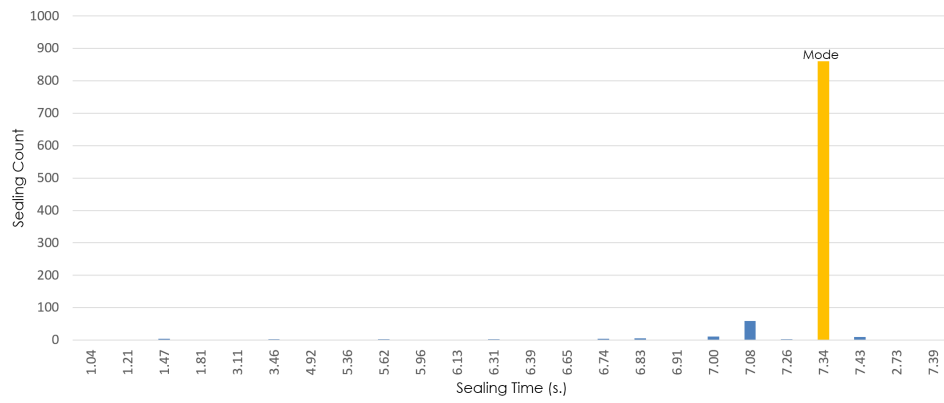
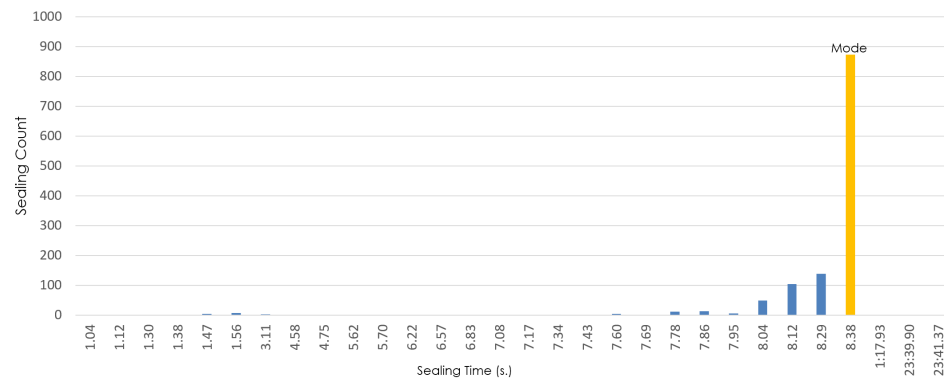


Figura 4.6: Determinación de la moda: (a) producto A, (b) producto B y (c) producto C en el turno 3.

4.2. MECANISMO DBF: FILTRADO DE EMIS EN SEÑALES DE SELLADO



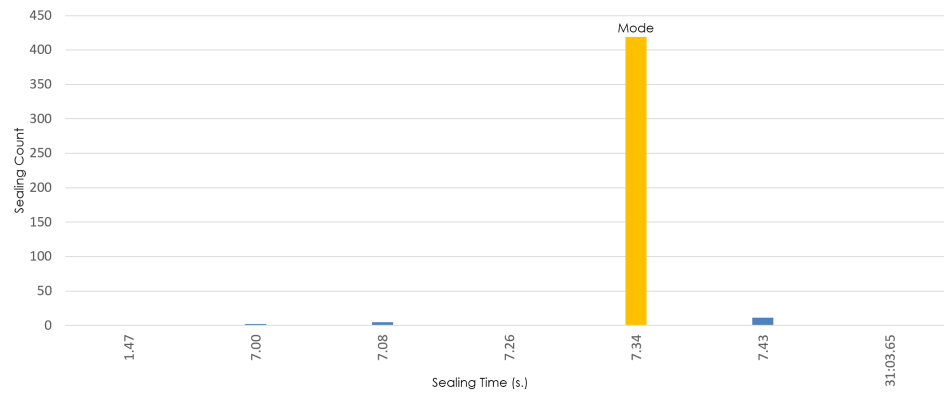
(a)



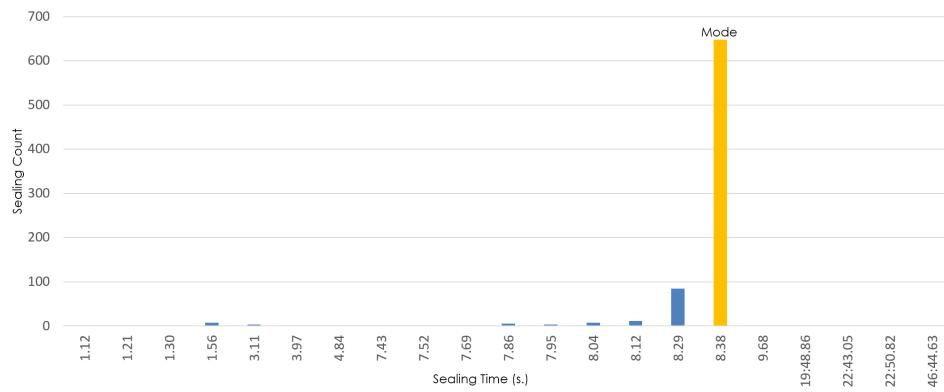
(b)

Figura 4.7: Determinación de la moda: (a) producto A y (b) producto B en el turno 4.

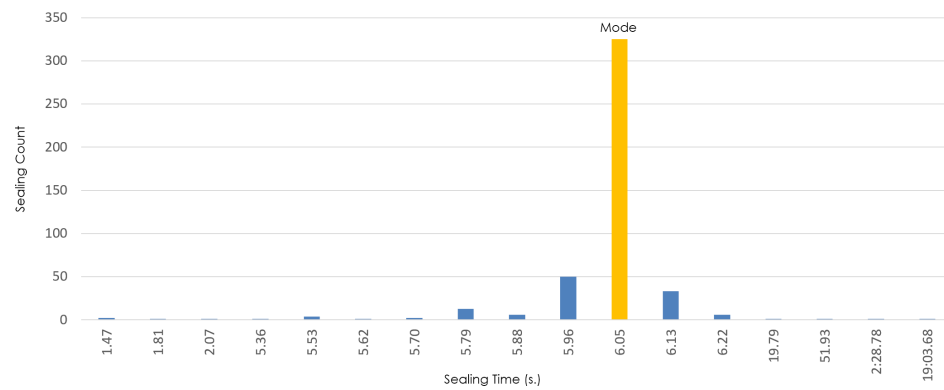
CAPÍTULO 4. MECANISMOS DE FILTRADO DE EMIS: DATA BASE FILTER Y SMART CODED FILTER



(a)



(b)



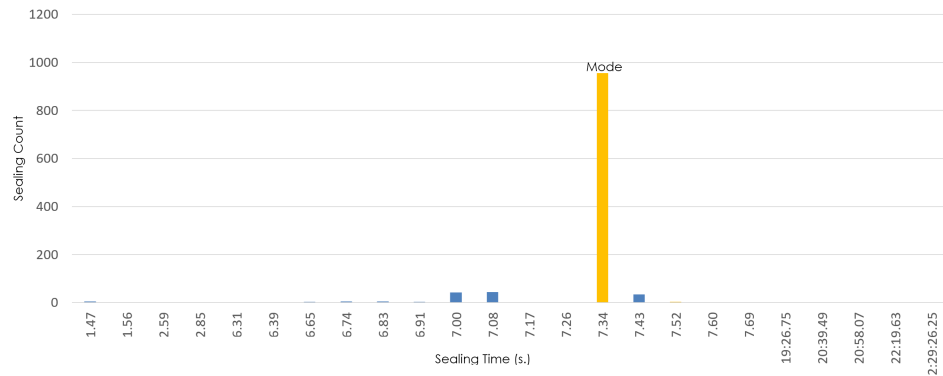
(c)

Figura 4.8: Determinación de la moda: (a) producto A, (b) producto B y (c) producto C en el turno 5.

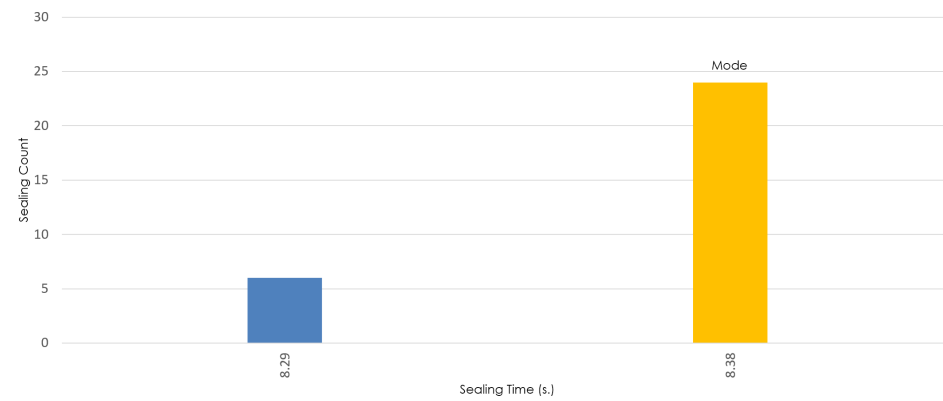
4.2. MECANISMO DBF: FILTRADO DE EMIS EN SEÑALES DE SELLADO

correcta las señales válidas. En la Figura 4.9(c) se representa el análisis de la moda del producto C en el turno 6. Este producto tiene una moda de 5.96 segundos y aplicando el offset fijado del 13.04 %, nuestro sistema considera válidas 884 señales de las 883 reales, teniendo un error asumible del 0.11 % en términos absolutos.

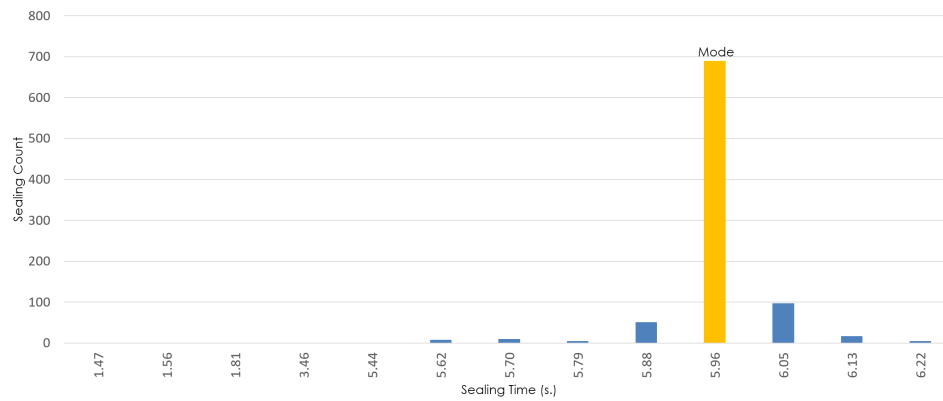
CAPÍTULO 4. MECANISMOS DE FILTRADO DE EMIS: DATA BASE FILTER Y SMART CODED FILTER



(a)



(b)



(c)

Figura 4.9: Determinación de la moda: (a) producto A, (b) producto B y (c) producto C en el turno 6.

4.2. MECANISMO DBF: FILTRADO DE EMIS EN SEÑALES DE SELLADO

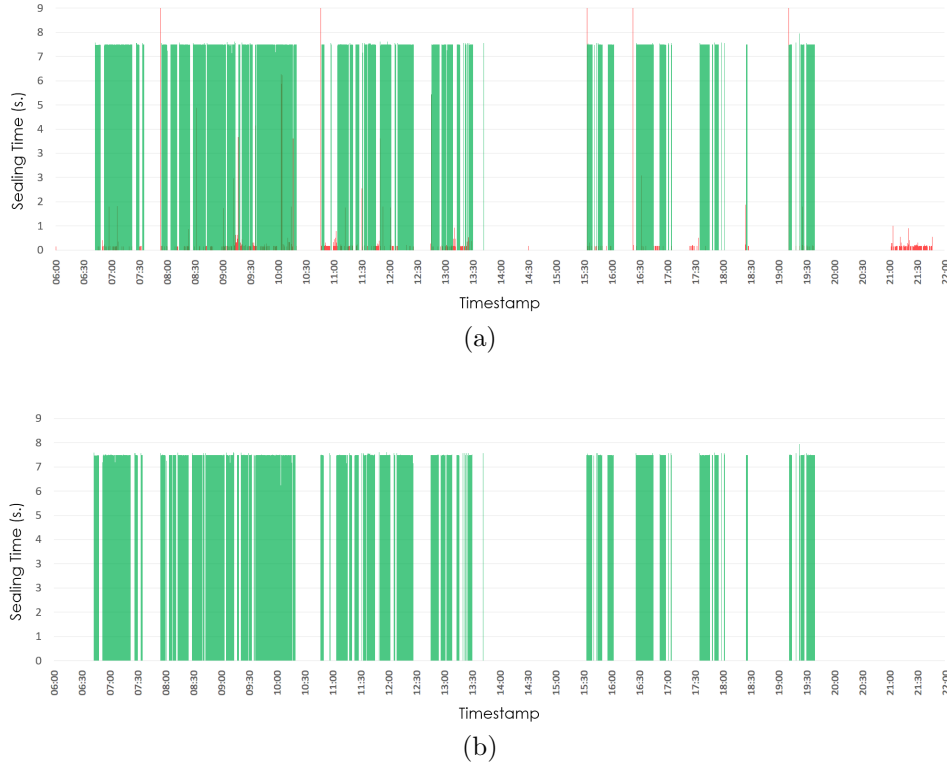


Figura 4.10: Señales de sellado: (a) RAW y (b) filtradas por el DBF en el turno 1.

4.2.2.2. Análisis del filtrado en los tiempos de sellado

Con el objetivo de representar las señales de sellado recogidas en los turnos analizados, así como visualizar las señales producidas por las EMIs, se presentan las Figuras 4.10 - 4.15, las cuales muestran el efecto de filtrado de nuestro mecanismo DBF. Más en detalle, se muestran en verde los Sealing Time (Ts), es decir, la duración de las señales de sellado en el momento exacto en el que se producen y en color rojo las señales producidas por interferencias electromagnéticas.

Para una mejor visualización del efecto de nuestro mecanismo de filtrado, hemos dispuesto las figuras de tal forma que se represente en las Figuras 4.10 - 4.15(a), las señales de sellado junto con el ruido producido en los diferentes momentos de los turnos, mientras que en las Figuras 4.10 - 4.15(b), se observan únicamente las señales validadas por el mecanismo DBF, una vez filtradas las EMIs. Como puede observarse, el DBF elimina las señales de ruido, al contemplar como buenas únicamente las señales que se encuentran dentro del intervalo marcado por la moda y el offset fijado ($\pm 13.04\%$), aunque en algunas ocasiones filtramos también señales válidas, tal y como hemos visto en la Tabla 4.4.

La Figura 4.10(a) muestra un único patrón de sellados, es decir, no existen cambios en los Sealing Time de color verde. Esto significa que solamente se ha

CAPÍTULO 4. MECANISMOS DE FILTRADO DE EMIS: DATA BASE FILTER Y SMART CODED FILTER

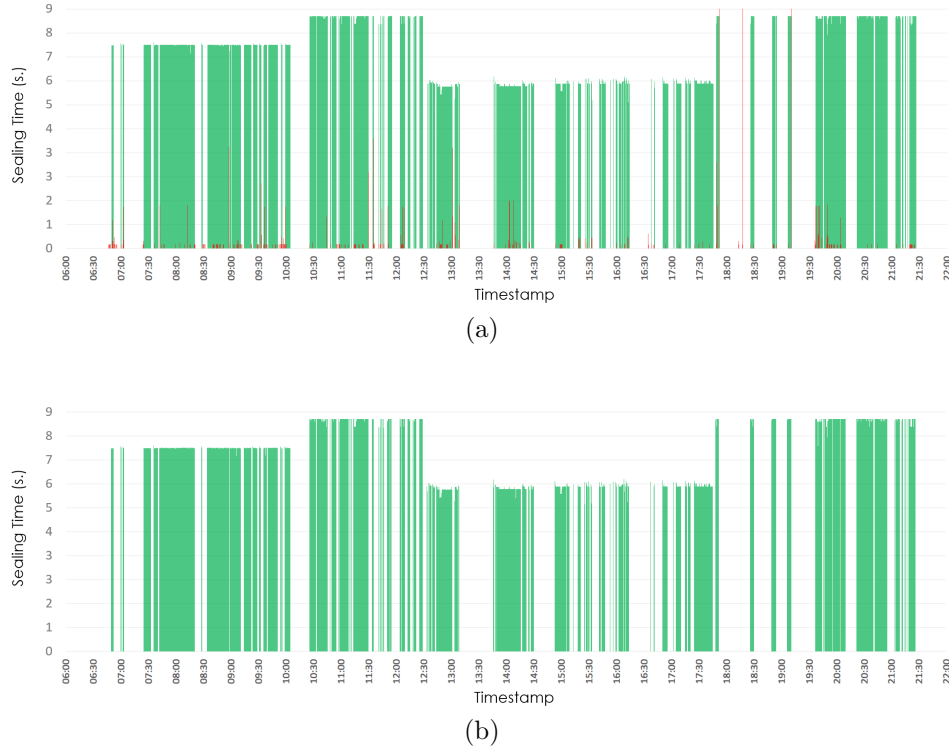


Figura 4.11: Señales de sellado: (a) RAW y (b) filtradas por el DBF en el turno 2.

trabajado con un producto, concretamente durante el periodo comprendido entre las 6:00 y las 22:00 h el producto elaborado ha sido el A, cuartos de queso, con un tiempo de sellado de 7.34 segundos. Además, pueden observarse las señales debidas a interferencias electromagnéticas, señales que una vez aplicado el mecanismo DBF, se eliminan completamente. Así lo muestra la Figura 4.10(b), en la que no existe ninguna señal que no sea correspondiente al patrón de sellado del producto A, basándonos en el filtrado que se realiza al calcular la moda y aplicar el offset fijado. Uno de los aspectos a señalar en este turno, es la existencia de interferencias electromagnéticas en el intervalo de las 21:00 y las 22:00 h, de no aplicar el mecanismo DBF estas señales serían contabilizadas como sellados. El mecanismo DBF elimina un total de 3683 señales, es decir un 64.65 % de las recibidas inicialmente.

En la Figura 4.11(a) se pueden observar los productos que intervienen en el turno 2, los cuales pueden ser identificados por los Sealing Time representados. Así, los productos elaborados durante este turno son el producto A (cuartos de queso), el producto B (quesos enteros), el producto C (cuñas de queso) y finalmente, otra vez, el producto B. Además de observarse los productos, pueden determinarse las paradas por cambios de rollo de plástico, paradas planificadas, cambios de turno y otras paradas que quedan reflejadas igualmente. En la Figura 4.11(b), se observa la actuación del mecanismo de filtrado DBF, quedándose únicamente las señales

4.2. MECANISMO DBF: FILTRADO DE EMIS EN SEÑALES DE SELLADO

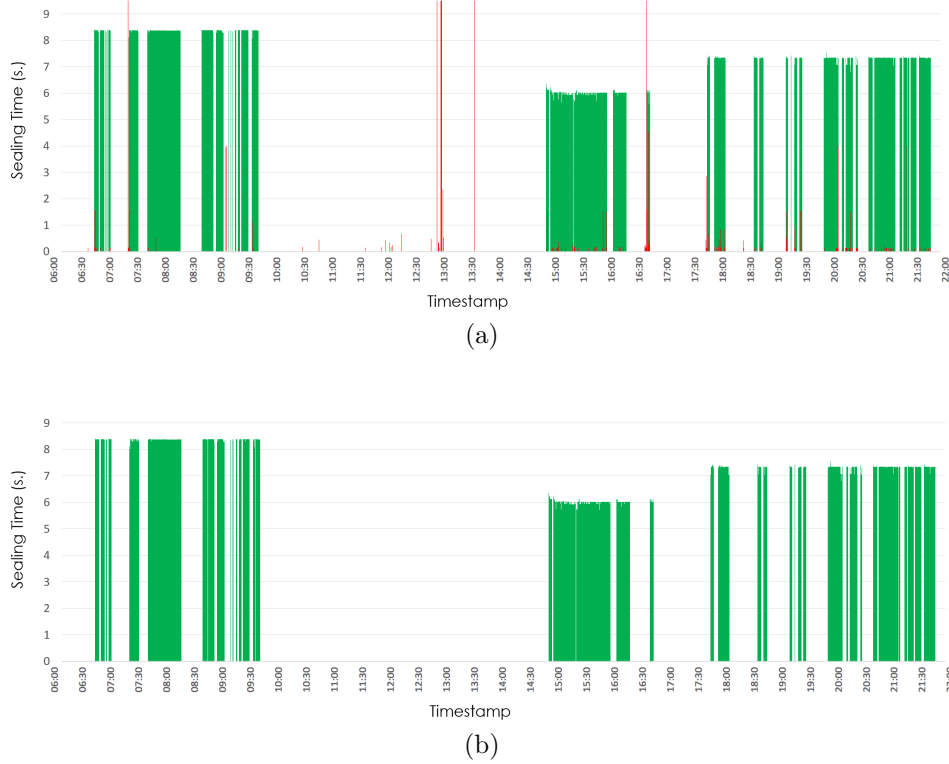


Figura 4.12: Señales de sellado: (a) RAW y (b) filtradas por el DBF en el turno 3.

válidas que se encuentren dentro de los valores del offset. En este caso, se eliminan 1386 señales de las recogidas inicialmente, lo que supone un 35.39 % del total de las señales.

La Figura 4.12(a) muestra los productos que han intervenido en el turno 3, los sellados realizados y el ruido que ha interferido desde las 6:00h hasta las 22:00h. De esta forma pueden determinarse claramente por su Sealing Time, la existencia de tres productos elaborados en el proceso (quesos, cuñas y cuartos, en ese orden). Existen distintos aspectos significativos que conviene señalar, como la existencia de interferencias electromagnéticas de larga duración que se sitúan entre las 12:30h y las 13:30h, posteriormente a las 16:30h se visualizan señales de las mismas características. Todas las señales anómalas son filtradas por nuestro mecanismo DBF, tal y como se observa en la Figura 4.12(b), además de visualizarse de forma clara las paradas, cambios de turno y periodos sin actividad. En este turno se han eliminado 1159 señales de un total de 2935 recogidas por el sistema, esto supone un 39.49 % de señales eliminadas por el mecanismo DBF.

La actividad en el turno 4 se indica en la Figura 4.13(a). En ella se observan los tiempos de sellado válidos y las interferencias electromagnéticas encontradas durante la duración del turno. Puede observarse claramente que se han elaborado

CAPÍTULO 4. MECANISMOS DE FILTRADO DE EMIS: DATA BASE FILTER Y SMART CODED FILTER

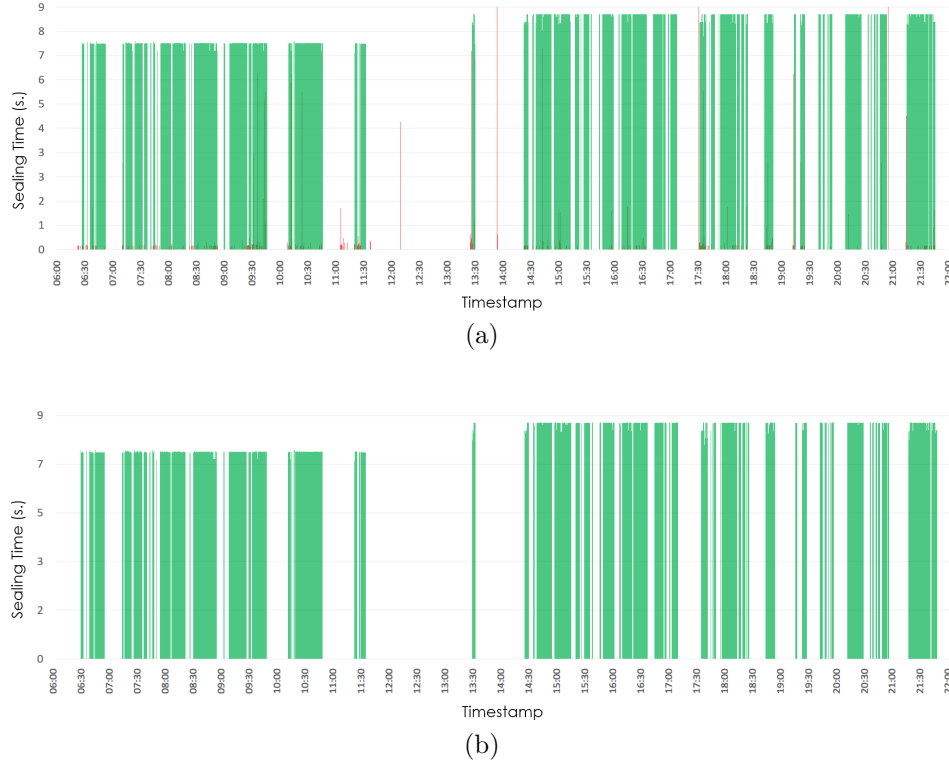


Figura 4.13: Señales de sellado: (a) RAW y (b) filtradas por el DBF en el turno 4.

solamente dos productos, cuartos de queso y quesos enteros, productos A y B, respectivamente. Se aprecia claramente los dos momentos en los que se produce cada uno de ellos, el primero de 6:30 a 11:30h y el segundo, quesos enteros, desde las 13:30 hasta las 22:00h. La figura manifiesta claramente los cambios de personal, paradas programadas, incluso las paradas por avería, como por ejemplo la que se realiza entre las 13:30 y las 14:30h. La Figura 4.13(b) muestra el comportamiento del turno, una vez eliminadas las señales de ruido. En este turno se han recogido inicialmente 4237 señales, eliminando el mecanismo DBF 2075, al no considerarlas válidas, es decir, un total de 48.97 % de las señales.

Por otra parte, en la Figura 4.14(a), se observa la existencia de ruido electro-magnético durante todo el turno, ya que el sistema recoge 1377 señales erróneas en total, lo que supone un 45.6 % de señales que deben eliminarse. Una vez filtradas las señales erróneas, en la Figura 4.14(b), pueden distinguirse el número de productos que han formado parte del turno de producción 5, exactamente tres: quesos, cuartos y cuñas. La distinción entre productos es clara durante el turno; desde las 6:30 hasta las 15:00 se realiza el sellado de quesos enteros, con un Sealing Time de 8.38 segundos; desde las 15:30 hasta las 18:10 se sellan cuartos de queso, con un Sealing Time de 7.34 segundos; y por último, desde las 19:30 hasta las 22:00, se produce el sellado de cuñas de queso, con un Sealing Time de 6.05 segundos. El

4.2. MECANISMO DBF: FILTRADO DE EMIS EN SEÑALES DE SELLADO

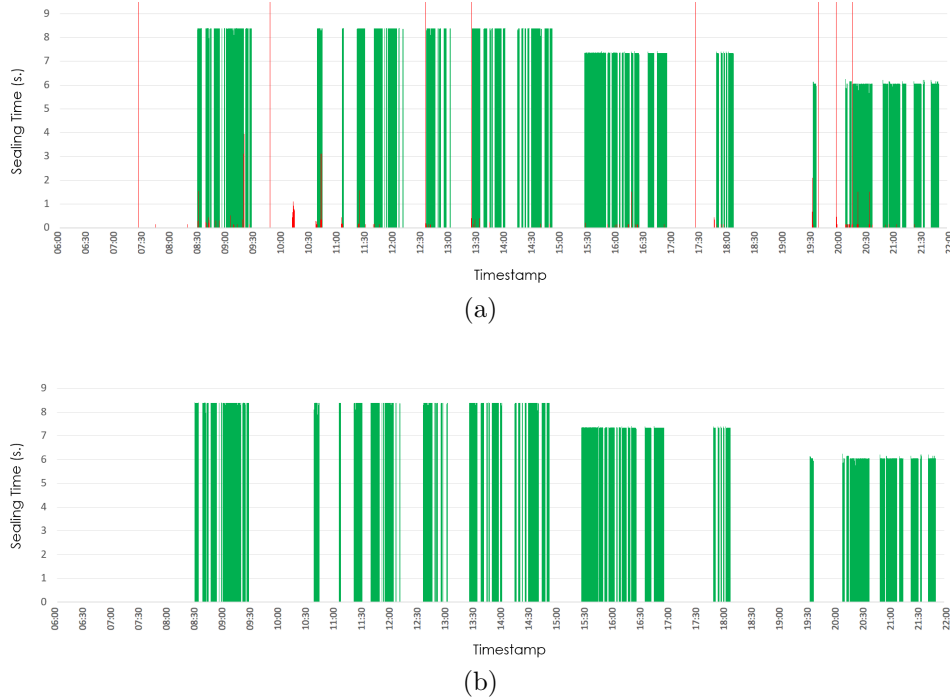


Figura 4.14: Señales de sellado: (a) RAW y (b) filtradas por el DBF en el turno 5.

mecanismo DBF, en este turno, ha eliminado 1377 señales de las 3018 recogidas inicialmente, lo que supone haber eliminado el 45.63% de las señales.

Finalmente, en el turno 6, representado en las Figuras 4.15(a) y 4.15(b), se elaboran los mismos productos que en el turno 5. El primer producto que se observa es el sellado de cuñas, con el que se inicia el turno a las 6:30 hasta las 10:10, y que tiene un Sealing Time de 5.93 segundos. El segundo proceso que se observa es el sellado de quesos, cuyo Sealing Time es de 8.34 segundos. El último producto sellado y que ocupa la mayoría del tiempo del turno (desde las 10:55 hasta el final del turno a las 20:20), es el envasado de cuartos con un Sealing Time de 7.36 segundos. En la Figura 4.15.(a), puede verse claramente el efecto de las EMIs y en la Figura 4.15(b) cómo se han eliminado las señales erróneas. Concretamente, existen un total de 756 señales eliminadas por el DBF, es decir, un 27.29% del total de señales inicialmente recogidas por el sistema.

CAPÍTULO 4. MECANISMOS DE FILTRADO DE EMIS: DATA BASE FILTER Y SMART CODED FILTER

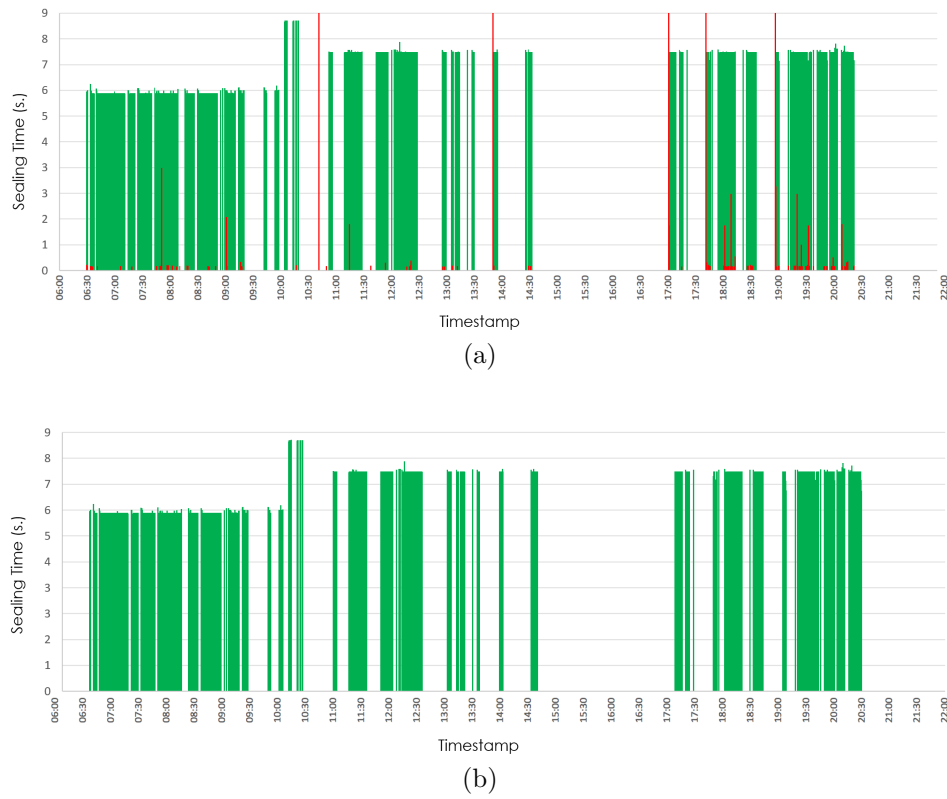


Figura 4.15: Señales de sellado: (a) RAW y (b) filtradas por el DBF en el turno 6.

4.2. MECANISMO DBF: FILTRADO DE EMIS EN SEÑALES DE SELLADO

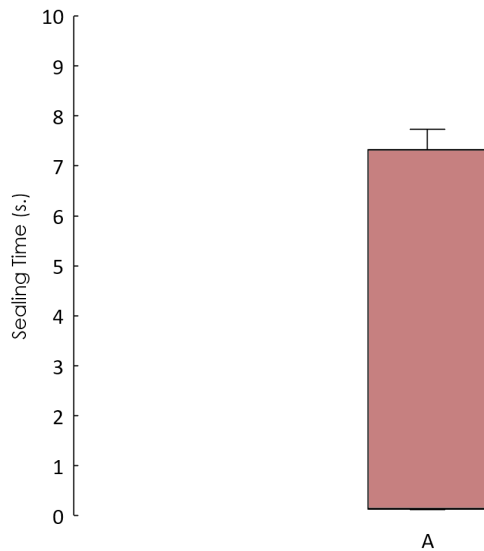


Figura 4.16: Señales de sellado RAW en el turno 1.

4.2.2.3. Distribución de las señales recogidas

A continuación, se presenta la distribución de señales de sellado antes y después de usar nuestro mecanismo DBF. En particular, las Figuras 4.16 - 4.21, muestran las señales de sellado RAW recopiladas durante todos los turnos analizados, mientras que las Figuras 4.22 - 4.27 presentan las señales de sellado que se obtuvieron, para cada uno de los productos, después de aplicar el filtrado DBF.

Como se muestra, las señales de sellado RAW están lejos de estar distribuidas simétricamente. En la Figura 4.16 los datos están muy dispersos y los valores de la señal van desde valores muy bajos (cerca de 0) hasta valores de 8 segundos. Ocurre lo mismo en la Figura 4.17, para los valores de los productos A y B, no siendo así para el producto C, en el que se observa que apenas existen EMIs y los valores se agrupan alrededor de la moda del tiempo de sellado.

La Figura 4.18 muestra el agrupamiento de señales en el turno 3. La dispersión de las señales en los productos B, C y A, oscilan entre los valores cercanos a 0 segundos y los 9. En este caso la mayoría de las señales de sellado se encuentran alrededor de la moda del tiempo de sellado de cada uno de los productos.

En el turno 4 (ver Figura 4.19), las señales del producto A se agrupan en torno a la moda, no siendo así en el producto B, en el que la mayoría de las señales se agrupan en tiempos inferiores a 1 segundo, lo que indica un alto nivel de ruido electromagnético.

La Figura 4.20 representa el agrupamiento de las señales RAW en el turno 5. Aquí las señales están dispersas encontrándose valores muy próximos 0 segundos, hasta los 10 segundos. Los datos muestran diferencias significativas entre las señales de los productos. Más concretamente, en la Figura 4.20 más del 50 % de las señales corresponden a un tiempo de sellado superior a 6 s, mientras que la mayoría de

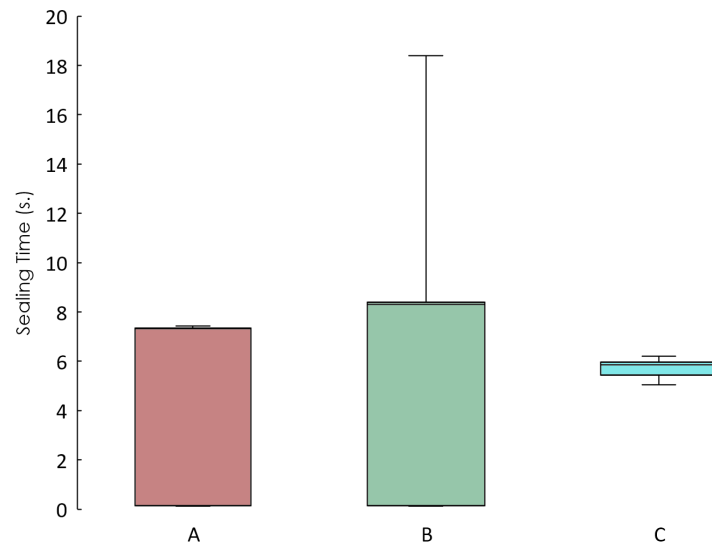


Figura 4.17: Señales de sellado RAW en el turno 2, agrupadas por producto.

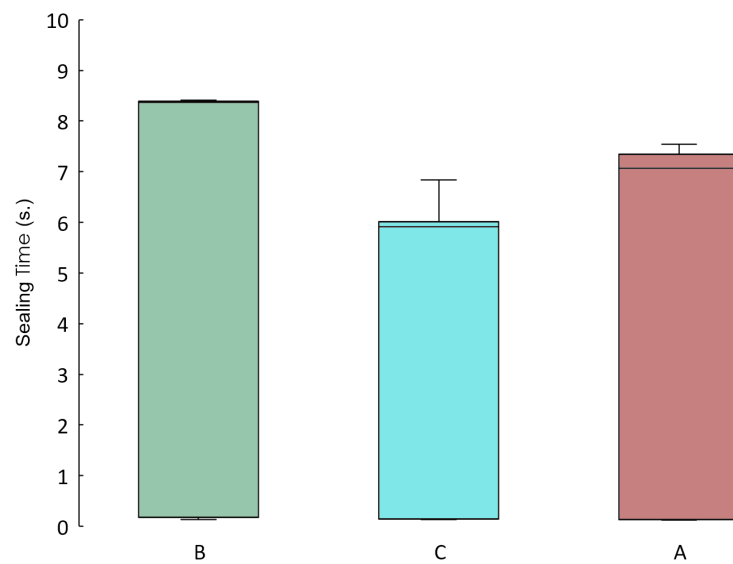


Figura 4.18: Señales de sellado RAW en el turno 3, agrupadas por producto.

4.2. MECANISMO DBF: FILTRADO DE EMIS EN SEÑALES DE SELLADO

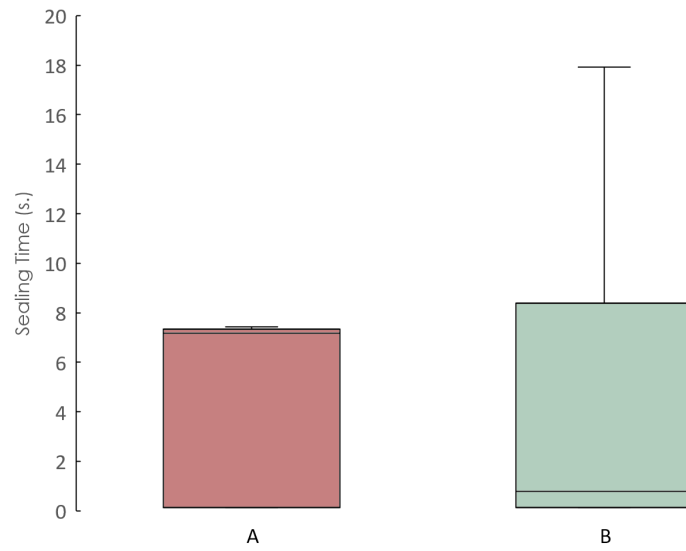


Figura 4.19: Señales de sellado RAW en el turno 4, agrupadas por producto.

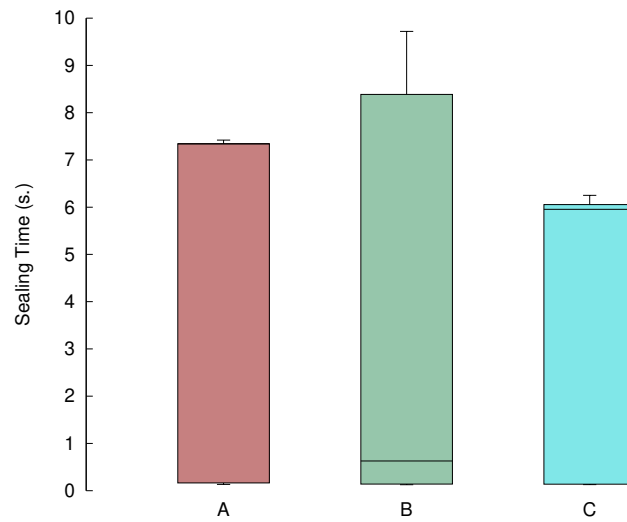


Figura 4.20: Señales de sellado RAW en el turno 5, agrupadas por producto.

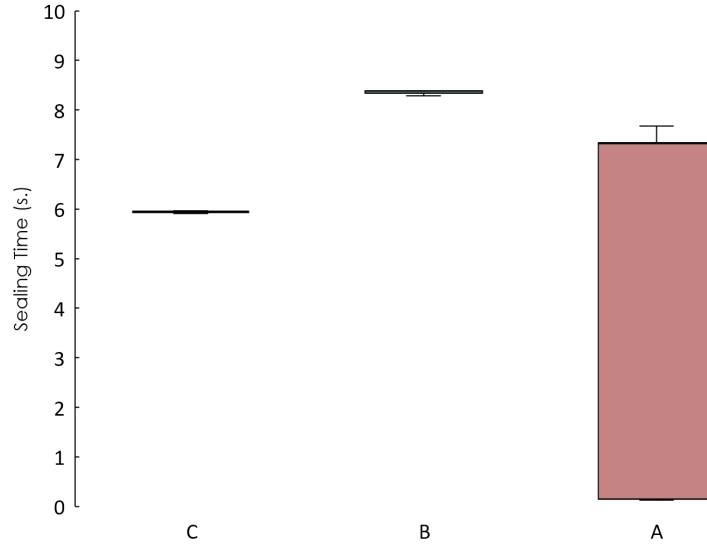


Figura 4.21: Señales de sellado RAW en el turno 6, agrupadas por producto.

las señales de sellado en el producto B se concentran alrededor de 1 s.

En el turno 6 (ver Figura 4.21), la dispersión de las señales varía en función del producto. En este caso las señales para los productos C y B se agrupan en torno a los valores de la moda, lo que indica que existe poco ruido electromagnético. Sin embargo, para el producto A, la dispersión de señales es muchísimo más amplia, teniendo valores cercanos a 0, hasta tiempos de sellado de casi 8 segundos.

Con respecto al efecto de filtrado del DBF, las Figuras 4.22 - 4.27 confirman la efectividad de nuestro mecanismo de filtrado, ya que una vez que el DBF es aplicado (y se eliminan las señales erróneas), los datos se distribuyen más simétricamente en todos los productos. Así lo refleja la Figura 4.22 en el turno 1, en el que las señales se distribuyen mayoritariamente alrededor de la moda. En el turno 2 (ver Figura 4.23) todos los productos, el A, el B y el C, se comportan de forma similar, es decir, se agrupan los valores de los sellados en torno a la moda. Lo mismo ocurre en los turnos 3, 4, 5 y 6 (Figuras 4.24 - 4.27), en los que las señales se distribuyen en torno a los valores de la moda, recogiendo la mayor parte de las señales válidas con el offset aplicado. Además, observamos que los tiempos de sellado para cada producto difieren menos de 5 décimas de segundo en el peor de los casos (ver Figura 4.23(c)).

Finalmente, la Figura 4.28 muestra la distribución de señales de sellado erróneas debido a las EMIs, recopiladas durante todos los turnos, incluidas aquellas cuya duración es menor a un segundo (que, de acuerdo al mecanismo DBF, son prefiltradas por la función `getCandidateSignalsDB`).

Como se muestra, las señales producidas por las EMIs no dependen del turno ni del producto envasado. De hecho, es notable que la gran mayoría de las señales erróneas se concentran alrededor de 0.14 s. Sin embargo, también hay algunos productos cuyas señales de sellado erróneas varían mucho en comparación con el

4.2. MECANISMO DBF: FILTRADO DE EMIS EN SEÑALES DE SELLADO

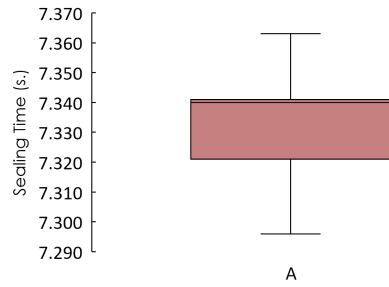


Figura 4.22: Señales de sellado en DBF en el turno 1: cuartos de queso (Producto A).

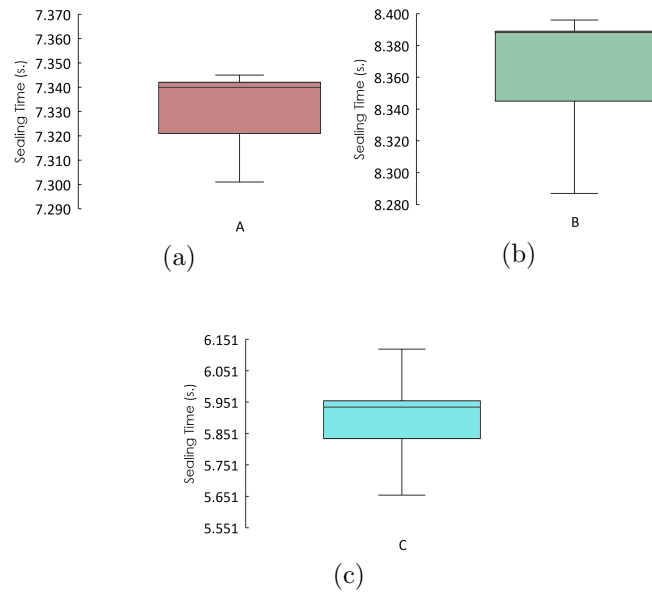


Figura 4.23: Señales de sellado en DBF en el turno 2: (a) cuartos de queso (Producto A), (b) quesos enteros (Producto B), y cuñas de queso (Producto C).

CAPÍTULO 4. MECANISMOS DE FILTRADO DE EMIS: DATA BASE FILTER Y SMART CODED FILTER

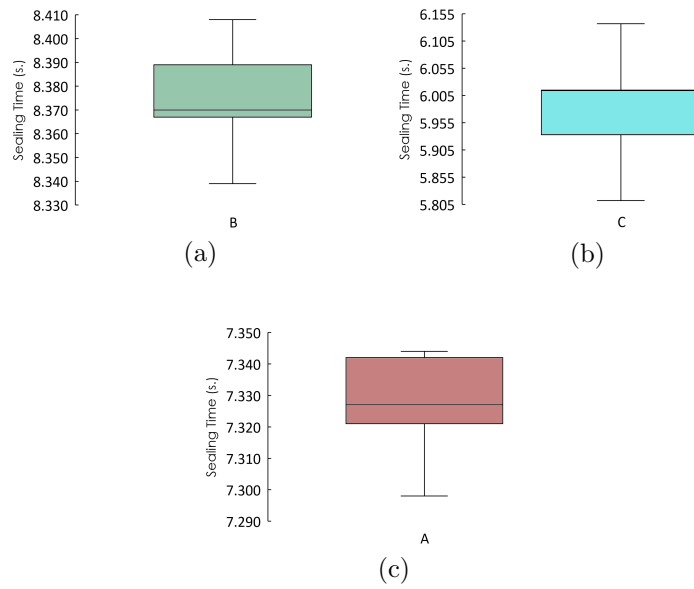


Figura 4.24: Señales de sellado en DBF en el turno 3: (a) quesos enteros (Producto B), (b) cuñas de queso (Producto C), y cuartos de queso (Producto A).

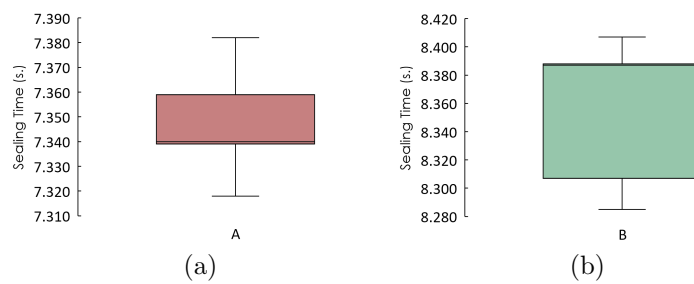


Figura 4.25: Señales de sellado en DBF en el turno 4: (a) cuartos de queso (Producto A) y (b) quesos enteros (Producto B).

4.2. MECANISMO DBF: FILTRADO DE EMIS EN SEÑALES DE SELLADO

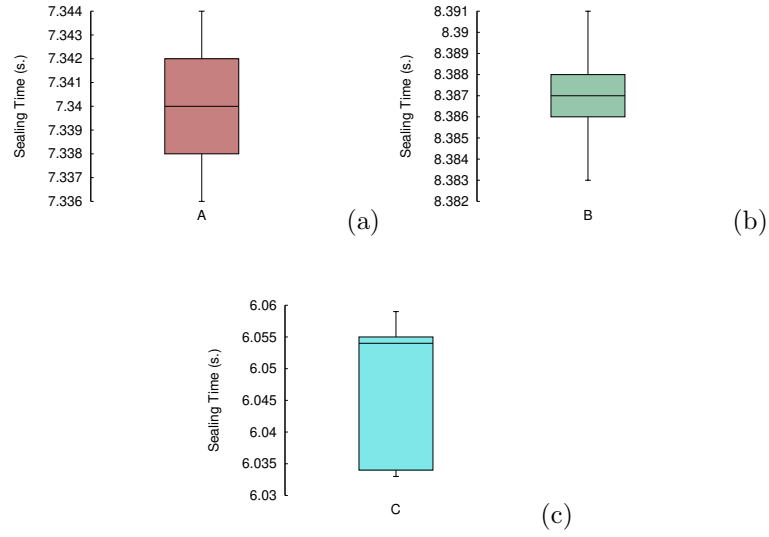


Figura 4.26: Señales de sellado en DBF en el turno 5: (a) cuartos de queso (Producto A), (b) quesos enteros (Producto B), y cuñas de queso (Producto C).

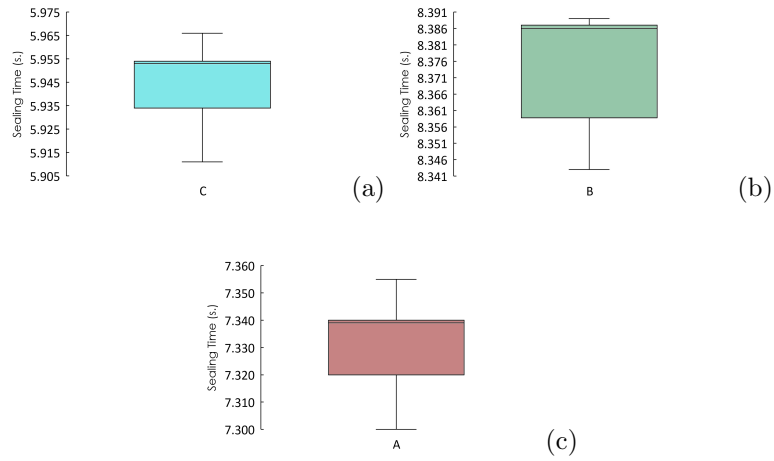


Figura 4.27: Señales de sellado en DBF en el turno 6: (a) cuñas de queso (Producto C), (b) quesos enteros (Producto B), y cuartos de queso (Producto A).

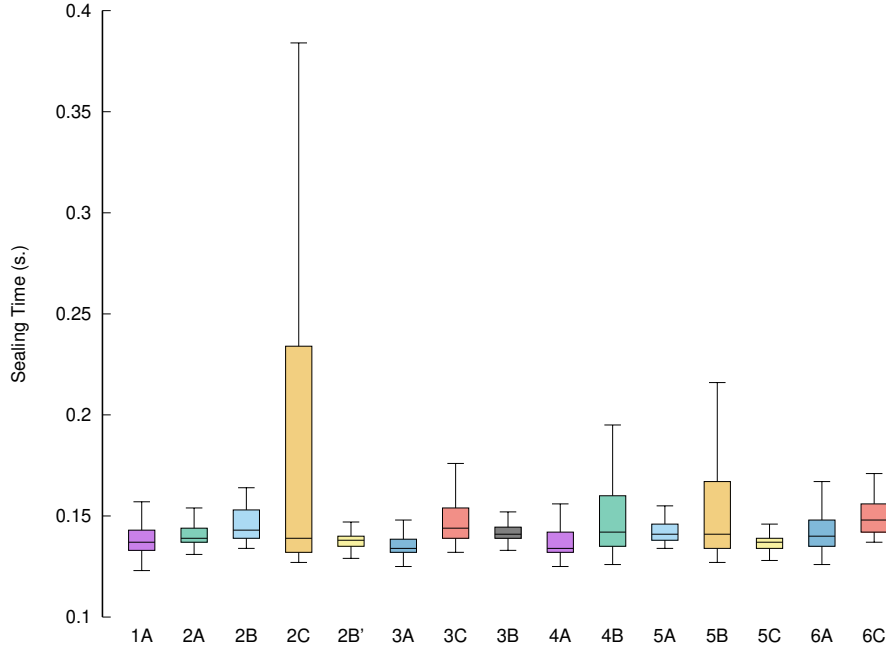


Figura 4.28: Señales de sellado erróneas recogidas durante todos los turnos.

resto, especialmente los presentados en 2C (es decir, señales erróneas recopiladas al sellar el producto C en el turno 2).

4.3. Mecanismo SCF: Filtrado de EMIs en señales de disponibilidad

Para filtrar las señales erróneas de disponibilidad, debemos tener en cuenta su singularidad y diferencias con respecto a las señales de sellado, dado que no es posible identificar patrones. Para ello, hemos diseñado el Smart Coded Filter (SCF), que evita la introducción de señales erróneas en el sistema, permitiendo una mejor precisión en el cálculo de la disponibilidad, y por ende en el cálculo del OEE.

4.3.1. Funcionamiento del mecanismo de filtrado SCF

La Figura 4.29 presenta un diagrama de flujo que resume cómo opera el mecanismo SCF. Más concretamente, muestra los dos procesos que se ejecutan en paralelo en la Raspberry Pi. Por un lado, el proceso emisor envía continuamente (cada segundo), utilizando el GPIO 03, la secuencia (0-1-0-1-0-1) siguiendo una duración predeterminada por bit (50ms, 150ms, 75ms, 125ms, y 100ms), es decir, el sistema envía un “0” durante 50ms, luego envía un “1” durante 150ms, un

4.3. MECANISMO SCF: FILTRADO DE EMIS EN SEÑALES DE DISPONIBILIDAD

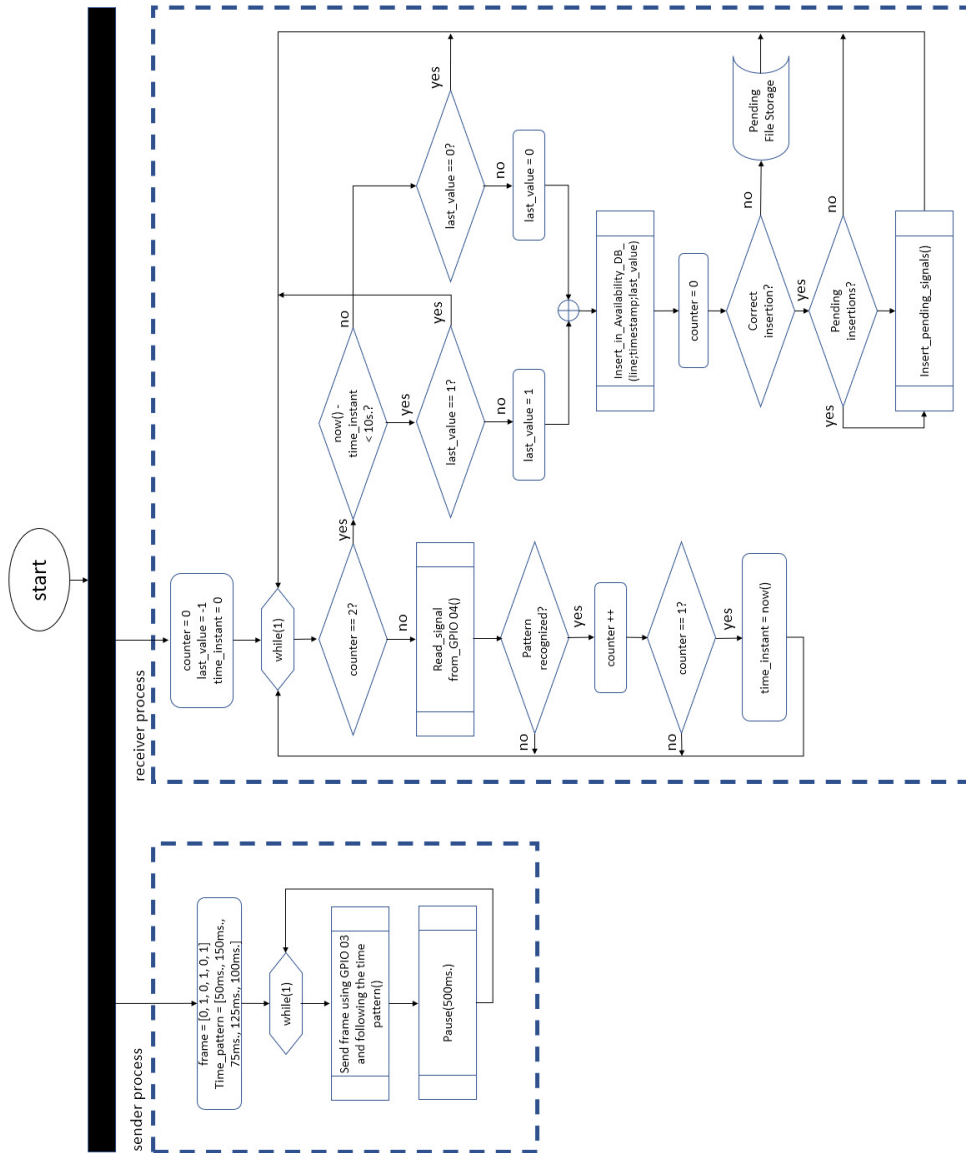


Figura 4.29: Diagrama de flujo del funcionamiento del mecanismo SCF.

CAPÍTULO 4. MECANISMOS DE FILTRADO DE EMIS: DATA BASE FILTER Y SMART CODED FILTER

“0” durante 75ms, etc., y espera 500ms para repetir el proceso nuevamente. Por otro lado, el proceso receptor monitoriza continuamente el puerto GPIO 04 a la espera de reconocer la trama enviada por el proceso emisor, ya que la trama sólo podrá ser recibida correctamente cuando la selladora encienda el circuito, es decir, cuando la máquina esté disponible.

Para aumentar considerablemente la confiabilidad del sistema, sólo considera que una señal de disponibilidad es correcta cuando la trama se recibe correctamente dos veces en los últimos 10 segundos (ver parámetro contador en el diagrama de flujo). De lo contrario, significa que la máquina no está disponible en ese momento. Esta doble verificación asegura que las señales recibidas no sean generadas de forma ficticia por ruido electromagnético.

Una vez identificada la señal, se almacena en la base de datos de disponibilidad. Si los datos no se insertan correctamente, generalmente debido a la falta de conectividad, la señal se almacena en el archivo de señales pendientes y el proceso volverá a esperar una nueva señal. Por el contrario, es decir, en el caso que exista conexión, el sistema verificará si hay señales almacenadas en el fichero de señales pendientes y las insertará en la base de datos.

Los datos almacenados en la base de datos de disponibilidad contienen básicamente la siguiente información: (i) línea, un código de cuatro dígitos que identifica la máquina de donde se obtiene los datos, (ii) timestamp, que determina el instante de tiempo en que se ha recibido la señal y (iii) valor de la señal, es decir, 1 ó 0 cuando la máquina está disponible o no, respectivamente. Vale la pena mencionar que, en el caso del último parámetro, el sistema evita insertar valores iguales de manera consecutiva en la base de datos, ya que una vez que se ha almacenado una señal de disponibilidad (o de no disponibilidad), la próxima señal a almacenar no puede ser la misma.

El propósito de SCF es descartar las señales recibidas que no representan la disponibilidad de la máquina. De acuerdo con esto, las señales erróneas son filtradas inmediatamente por el sistema, siempre que no cumplan con el protocolo SCF. En consecuencia, SCF evita que se inserten valores erróneos relativos a la disponibilidad en la base de datos.

El funcionamiento del SCF viene determinado por los siguientes algoritmos (GPIO Sender, GPIO Listener y el Inactive Machine Checker), que se ejecutan en paralelo en la Raspberry Pi, y cuyo código aparece en los Algoritmos 5, 6 y 7. El GPIO Sender (Algoritmo 5) es el proceso encargado de enviar un mensaje codificado con el objetivo de indicar que la máquina está disponible, enviando un patrón de señales en un tiempo concreto. El GPIO Listener (Algoritmo 6) está encargado de recibir el mensaje, y estimar si la máquina está disponible. Para ello, comprueba que el patrón se ha recibido un mínimo de veces (MINPATTERN). Finalmente, el Inactive Machine Checker (Algoritmo 7) es el encargado de detectar que la selladora no está activa.

A continuación, explicamos con más detalle cada uno de los procesos nombrados anteriormente. Para explicar el Algoritmo 5, es importante destacar que en la librería utilizada para la implementación del código, no es posible enviar un determinado estado (0 o 1), sino que se envían mensajes de cambio de estado (changeState). De esta forma, el GPIO Sender, se encarga de enviar un patrón de

4.3. MECANISMO SCF: FILTRADO DE EMIS EN SEÑALES DE DISPONIBILIDAD

Algoritmo 5 GPIO Sender

```
/* ***** */
/* pause: pause time in milliseconds */
/* pattern: pattern sent with the determined durations */
/* ***** */
int pause = 500;
int pattern[] = [50,150,75,125,100];
while rpiON do
    foreach time in pattern do
        | send(changeState) sleep(time)
    end
    sleep(pause)
end
```

Algoritmo 6 GPIO Listener

```
if counter < MINPATTERN then
    if patternComplete() then
        counter++; // checking that the signal received is the last of
        a right pattern
        if counter >= MINPATTERN AND !active then
            | insertDB(currentMillis(), 1) active = 1
        end
    end
end
end
```

cambios de estado a través de un GPIO, en un patrón de tiempo fijado (50 ms, 150 ms, 75 ms, 125 ms y 100 ms). Es decir, el sistema emite un cambio de estado, espera 50 ms, después envía otro cambio de estado y espera, 150 ms, envía otro cambio de estado, espera 75 ms y así sucesivamente, hasta completar el patrón de tiempos. Una vez completado, espera 500 ms (pause) para repetir el proceso de nuevo.

El proceso GPIO Listener monitoriza las señales recibidas, esperando el reconocimiento del patrón enviado por el proceso GPIO Sender. El GPIO Listener, se ejecuta automáticamente cada vez que se recibe una interrupción por un cambio de estado, y en el caso de que reciba el patrón dos veces (MINPATTERN), establece que la máquina está disponible (active=1) y lo inserta en la base de datos.

Finalmente, el Inactive Machine Checker, que se ejecuta de manera independiente al GPIO Listener, es el encargado de comprobar cada 10 segundos (TIMEOUT) si la máquina está disponible. En este caso, comprueba que el número de mensajes recibidos durante ese periodo son inferiores al valor de MINPATTERN. Si esta condición se cumple, considerará que la máquina está inactiva (active=0) y lo insertará en la base de datos. La doble comprobación, (i.e., MINPATTERN=2) permite al sistema asegurar que las señales recibidas no son debidas a interferencias electromagnéticas.

El GPIO Listener y el Inactive Machine Checker, utilizan variables comunes

CAPÍTULO 4. MECANISMOS DE FILTRADO DE EMIS: DATA BASE FILTER Y SMART CODED FILTER

Algoritmo 7 Inactive Machine Checker

```
while rpiON do
  if counter < MINPATTERN AND active then
    | insertDB(currentMillis(), 0) active = 0;
  end
  counter = 0 sleep(TIMOUT)
end
```

entre ellos y que comentamos en detalle a continuación:

- **MINPATTERN**: indica el número mínimo de veces que debe repetirse el patrón enviado por GPIO sender para considerar que la máquina está activa. Su valor es 2, para asegurarnos que el mensaje recibido no ha sido producido por las EMIs.
- **TIMEOUT**: es el tiempo máximo considerado para que se cumplan las condiciones para establecer si la máquina está activa o no. Su valor es de 10 s.
- **counter**: es la variable que realiza el conteo de patrones y su valor inicial es 0.
- **active**: indica el estado de la máquina y tiene un valor inicial 0.

4.3.2. Resultados del mecanismo de filtrado SCF

La Tabla 4.6 muestra los resultados obtenidos para todos los turnos analizados. La primera columna indica el turno estudiado, la segunda columna muestra el número de señales reales, las columnas tercera y cuarta (RAW) indican las señales de disponibilidad del sistema sin ningún tipo de filtrado y el porcentaje de señales erróneas, respectivamente. En las columnas quinta y sexta (SCF signals), se muestran los valores obtenidos, una vez aplicado el algoritmo de filtrado SCF.

Tal y como se refleja, las señales erróneas recogidas por el sistema sin filtrar el ruido, oscilan entre un 31.98 % y un 96.25 %, mientras que el error, una vez filtradas las señales con el SCF, es del 0 % en todos y cada uno de los turnos, lo que valida el mecanismo de filtrado de EMIs para la obtención de datos relativos a la disponibilidad de las máquinas.

No obstante, es importante tener en cuenta que el error mencionado está relacionado con el número de señales de disponibilidad recibidas, pero no con los tiempos de disponibilidad de la máquina. En las columnas séptima y octava de la tabla se indican los porcentajes de disponibilidad, con respecto a la duración total del turno, teniendo en cuenta las señales RAW y cuando las señales están filtradas por el mecanismo SCF. Además, en la última columna se muestra la diferencia.

Como se observa, el error en la cantidad de señales recibidas no está relacionado directamente con el error cometido en términos de duración de la disponibilidad de la máquina. A modo de ejemplo, se observa un error en la recepción de señales del turno 4 del 68.18 %, mientras que en términos de porcentaje de disponibilidad

4.3. MECANISMO SCF: FILTRADO DE EMIS EN SEÑALES DE DISPONIBILIDAD

Tabla 4.6: Resultados de señales de disponibilidad obtenidos en los seis turnos analizados

Shift	Real Signals	RAW		SCF Signals		Availability		
		Received Signals	Erroneous Signals	Signals	Erroneous Signals	w/o SCF	w/ SCF	Diff
1	52	1.385	96.25 %	52	0.00 %	31.84 %	72.19 %	40.35 %
2	82	354	76.84 %	82	0.00 %	69.12 %	81.50 %	12.38 %
3	117	172	31.98 %	117	0.00 %	66.16 %	77.43 %	11.27 %
4	28	88	68.18 %	28	0.00 %	80.27 %	82.94 %	2.67 %
5	51	79	35.44 %	51	0.00 %	49.72 %	74.06 %	24.34 %
6	29	67	56.72 %	29	0.00 %	59.87 %	80.20 %	20.33 %

el error es de un 2.67 %. Por el contrario, en el turno 5 se observa un error menor en la recepción de señales con respecto al turno 4, concretamente de un 35.44 %, pero la diferencia en la disponibilidad es sustancial (24.34 %).

En la Tabla 4.6 además, se observa que el turno 1 presenta un gran número de señales recibidas en RAW, lo que también aparece reflejado en los datos de disponibilidad sin filtrado SCF. En este turno existen 1385 señales en RAW y un error del 96.25 % con respecto a las señales reales, una cantidad muy grande de señales erróneas en RAW que también repercute, lógicamente, en la estimación de la disponibilidad. En este caso, la diferencia de disponibilidad es de un 40.35 %, lo que no nos permitiría estimar con precisión el OEE.

En conclusión, las señales erróneas producidas por las EMIs van a afectar negativamente a los resultados obtenidos a la hora de estimar la disponibilidad de una máquina, a priori, aumentando o disminuyendo el porcentaje de disponibilidad de forma aleatoria, aunque a la vista de los resultados obtenidos en el despliegue, vemos que la disponibilidad estimada es siempre menor a la real.

En cualquier caso, vemos que es de gran importancia asegurarnos de que todas las señales sean correctas, ya que una sola señal podría afectar de forma significativa a la disponibilidad, por lo que el uso de nuestro mecanismo de filtrado SCF se hace totalmente indispensable en este tipo de entornos, donde tenemos interferencias electromagnéticas.

Las Figuras 4.30-4.35, muestran la diferencia entre las disponibilidades obtenidas sin filtrado y las que incluyen el filtrado mediante SCF en todos los turnos analizados.

La Figura 4.30, muestra claramente las diferencias de disponibilidades captadas por el dispositivo sin el filtrado del mecanismo SCF y con la aplicación del SCF. Una prueba de ello, que sirve como ejemplo, es la generada entre las 6:30 y las 8:00 h. En este caso existe una falta de disponibilidad en la Figura 4.30(a) que no se manifiesta en la realidad y que se observa en la Figura 4.30(b). La diferencia entre la disponibilidad estimada sin usar el SCF y la disponibilidad estimada una vez aplicado el mecanismo SCF es sustancial.

La Figura 4.31 muestra situaciones de similares características al anterior turno. En este caso se representa la disponibilidad del turno 2. Se puede observar que la Figura 4.31(a) no muestra disponibilidad en diferentes periodos del turno, cuando en realidad sí existe esa disponibilidad, y así se refleja una vez aplicado el SCF (ver Figura 4.31(b)). Algunos ejemplos de esta falta de disponibilidad son los periodos

CAPÍTULO 4. MECANISMOS DE FILTRADO DE EMIS: DATA BASE FILTER Y SMART CODED FILTER

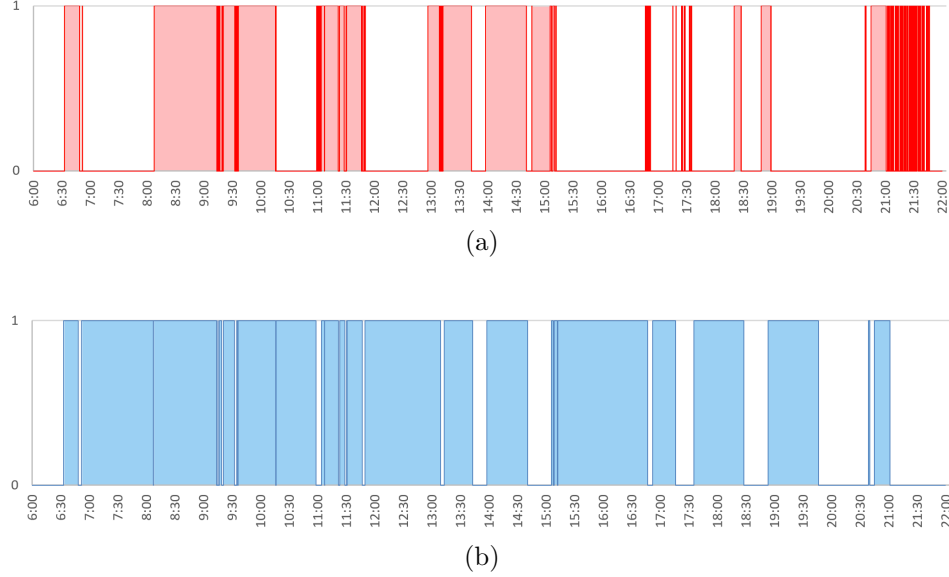


Figura 4.30: Disponibilidad en turno 1: (a) Señales RAW y (b) Señales filtradas por SCF.

entre las 6:00 y las 7:00 h, así como de 9:30 a 10:00 h, y el que se observa entre las 12:00 y 12:30 h, entre otros.

La disponibilidad del turno 3 se representa en la Figura 4.32. Comparando la Figura 4.32(a) con la 4.32(b), se observa la diferencia que existe en términos de disponibilidad entre los datos recibidos (RAW) y los datos una vez se ha realizado el filtrado de las señales con el SCF.

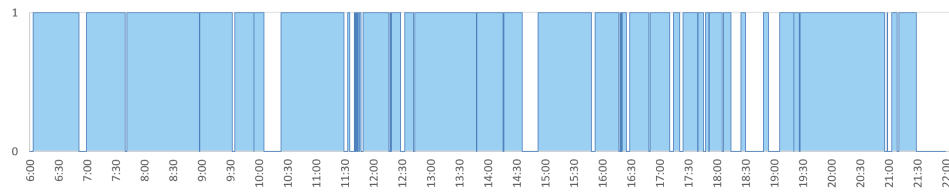
Las señales erróneas generadas por EMIs se observan de manera clara al comparar las Figuras 4.33(a) y 4.33(b). Dos de los errores más visibles aparecen entre las 11:30h y las 12:15h, cuando el dispositivo detecta dos paradas y puestas en marcha de la máquina, mientras que con el filtrado SCF dichas señales erróneas se ignoran. En la Figura 4.33(b) se observan las paradas que se realizan en el turno con diferentes fines. Concretamente un cambio de producto entre las 11:00h y las 11:20h para sellar una serie corta, una parada planificada entre las 11:30h y las 13:00h, una parada debida al relevo de personal entre las 14:00h y las 14:30h, así como una parada por cambio de plástico a las 19:30h, y otro cambio de producción a las 20:00h.

En la Figura 4.34(a) también se observan de manera significativa las señales erróneas producidas por las EMIs en el turno 5. En concreto, se observan errores de disponibilidad entre las 7:00h y las 8:30h, entre las 13:00h y las 14:00h, las 17:00h y las 18:00h, así como entre las 19:30h y las 20:00h. La disponibilidad real de la máquina se muestra en la Figura 4.34(b), donde el SCF permite obtener la disponibilidad real de la selladora.

4.3. MECANISMO SCF: FILTRADO DE EMIS EN SEÑALES DE DISPONIBILIDAD

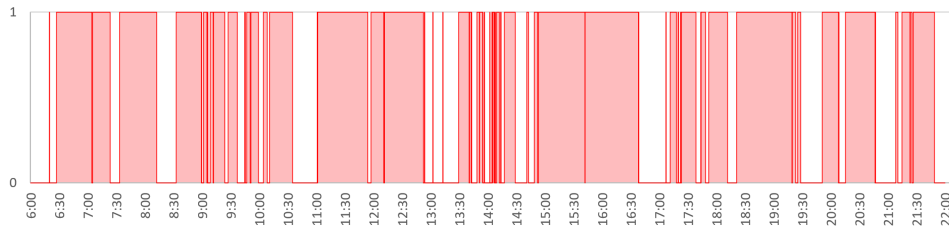


(a)

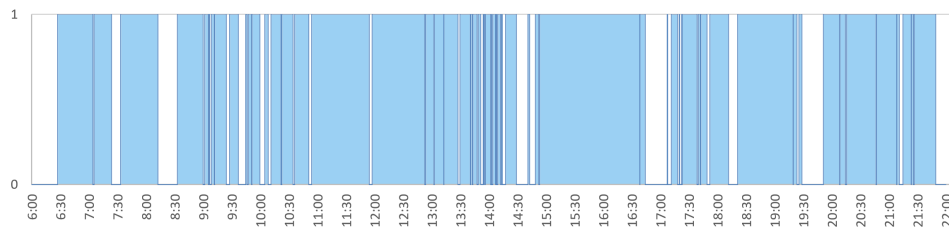


(b)

Figura 4.31: Disponibilidad en turno 2: (a) Señales RAW y (b) Señales filtradas por SCF.



(a)



(b)

Figura 4.32: Disponibilidad en turno 3: (a) Señales RAW y (b) Señales filtradas por SCF.

CAPÍTULO 4. MECANISMOS DE FILTRADO DE EMIS: DATA BASE FILTER Y SMART CODED FILTER

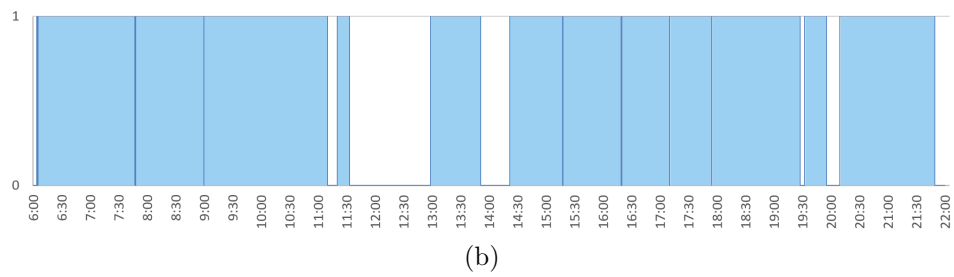
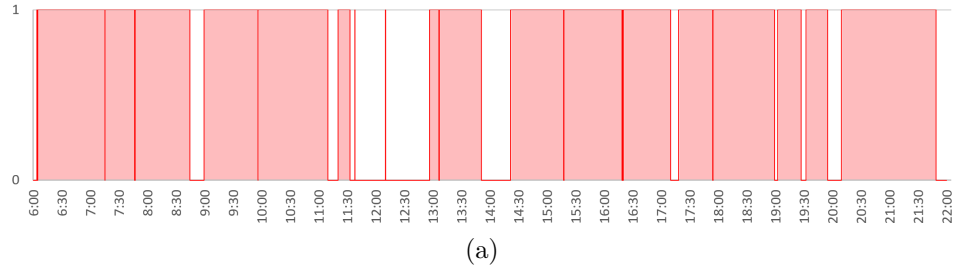


Figura 4.33: Disponibilidad en turno 4: (a) Señales RAW y (b) Señales filtradas por SCF.

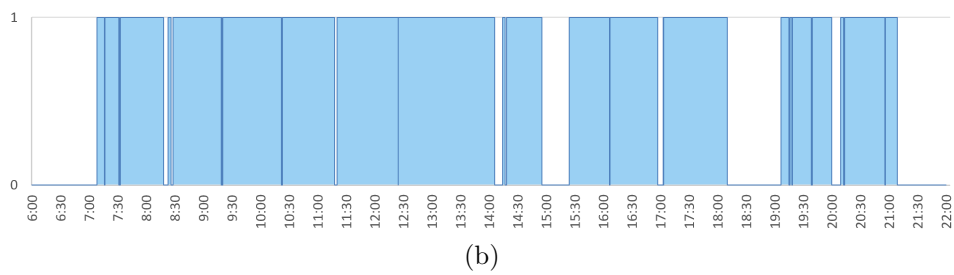
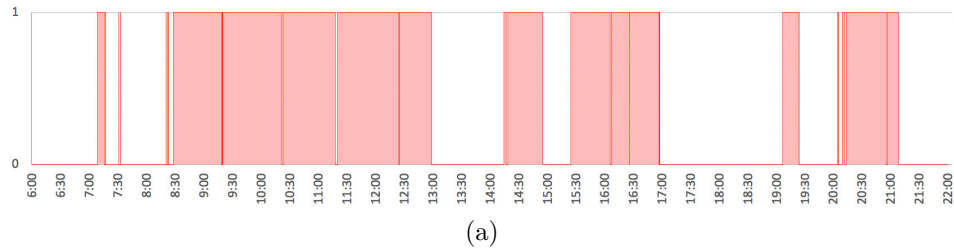


Figura 4.34: Disponibilidad en turno 5: (a) Señales RAW y (b) Señales filtradas por SCF.

4.3. MECANISMO SCF: FILTRADO DE EMIS EN SEÑALES DE DISPONIBILIDAD

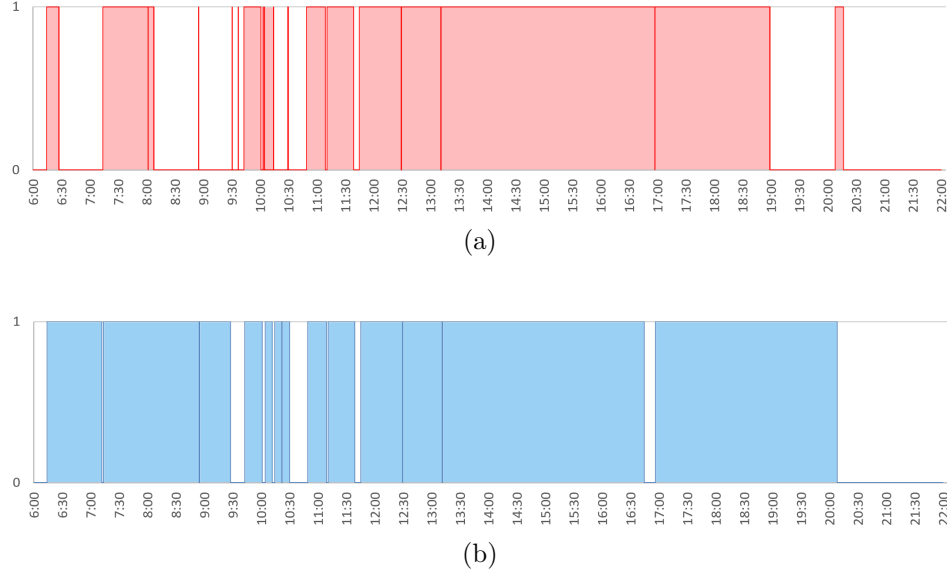


Figura 4.35: Disponibilidad en turno 6: (a) Señales RAW y (b) Señales filtradas por SCF.

En lo relativo al turno 6, la Figura 4.35(a) muestra de forma clara los errores cometidos por el sistema debidos a las interferencias electromagnéticas, lo que provoca que el sistema considere periodos en los que la máquina no estaba disponible, sin que en realidad esto haya ocurrido. Entre las 6:15 y las 6:30, entre las 8:00 y las 9:30, entre las 10:00 y 10:30, así como las 19:00 y las 20:15, existen faltas de disponibilidad o ficticias. Esto queda reflejado de forma evidente, una vez filtradas las señales con nuestro mecanismo de filtrado SCF, tal y como se observa en la Figura 4.35(b). En este turno, puede verse, de manera evidente, que el sistema sin filtrado estima de manera equivocada, grandes periodos de no disponibilidad y otros de disponibilidad, cuando la máquina realmente no estaba disponible.

Finalmente, la Figura 4.36 muestra, a modo de resumen, los valores obtenidos en todos los turnos analizados, relativos a la disponibilidad que existe antes de utilizar el mecanismo de filtrado SCF y después de utilizarlo. Las diferencias observadas, demuestran que obtener las señales correctas de disponibilidad afecta directamente en el cálculo del OEE y la fiabilidad de este factor. Como puede observarse, en el turno 1 existe una disponibilidad inicial de 31.84 %, y posteriormente, aplicando el mecanismo SCF se obtiene un valor del 72.19 %. Esto indica una gran diferencia en los datos ofrecidos por el sistema sin aplicar el SCF y una vez aplicado; concretamente un 40.35 % más de disponibilidad aplicando el SCF.

Para el turno 2 la diferencia entre la disponibilidad con interferencias electromagnéticas y sin ellas, es de un 12.38 %, puesto que inicialmente se recoge una disponibilidad del 69.12 % y aplicado el mecanismo SCF, el 81.50 %. En este caso

CAPÍTULO 4. MECANISMOS DE FILTRADO DE EMIS: DATA BASE FILTER Y SMART CODED FILTER

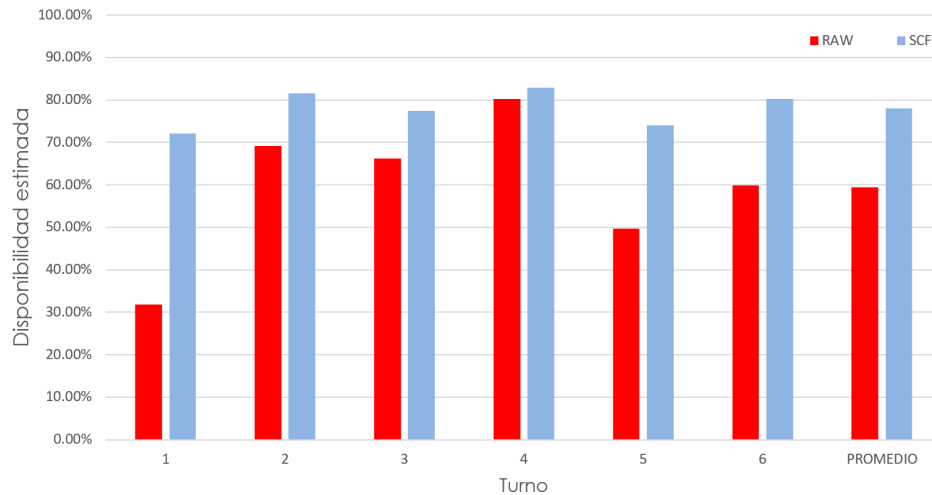


Figura 4.36: Disponibilidad con datos RAW y con filtrado SCF en todos los turnos analizados.

la diferencia existente entre aplicar o no el SCF es del 12.38 %.

En el turno 3 se recoge una disponibilidad del 66.16 % inicialmente, es decir, sin filtrado, y posteriormente con filtrado se recoge una disponibilidad del 77.43 %.

En el caso del turno 4, la diferencia de disponibilidad sin aplicar el mecanismo SCF y una vez que se aplica el SCF es bastante inferior a los anteriores escenarios, pasando de un 80.27 % sin aplicar el SCF, a un 82.94 % una vez aplicado el mecanismo.

Es notable también, la diferencia existente en el turno 5, cuyos datos sin SCF son del 49.72 % de disponibilidad, mientras que aplicando el SCF se obtiene que la disponibilidad real es del 74.06 % del tiempo total del turno, es decir, un 24.34 % menos si no se aplica el SCF.

En el turno 6 existe una diferencia entre disponibilidad sin SCF, cuyo valor es del 59.87 %, y una disponibilidad con SCF, del 80.20 %. Más concretamente, un 20.33 %.

En lo referido a los valores promedio, los datos que se muestran indican una disponibilidad sin SCF del 59.50 % y una disponibilidad una vez aplicado el SCF del 78.05 %, es decir, una diferencia, en promedio, de un 18.55 % de disponibilidad más, una vez aplicado el SCF.

4.4. Funcionamiento conjunto de los mecanismos DBF y SCF

Para ofrecer una visión en conjunto de nuestra propuesta, vamos a mostrar los dos tipos de señales, es decir, disponibilidad y sellados en una misma gráfica, comparando las señales recibidas por el dispositivo, y por otro lado, las señales obtenidas al aplicar los mecanismos de filtrado propuestos. Las Figuras 4.37 y 4.38 muestran diferentes momentos del turno 4.

Más en detalle, la Figura 4.37(a) presenta un extracto del turno 1 desde las 10:30 hasta las 12:30, en el que se muestran todas las señales de sellado y disponibilidad recibidas, incluyendo las producidas por interferencias electromagnéticas. El eje vertical izquierdo, indica el estado de disponibilidad de la máquina, mientras que el eje vertical derecho representa los tiempos de sellado. Como se puede observar, el ruido producido en las señales de sellado se manifiesta con valores fuera del patrón que representa el sellado (1), es decir, sellados con una duración fuera de los límites del offset fijado. Además, se aprecia claramente la no disponibilidad manifestada por el sistema desde las 10:30 hasta las 11:00, demostrando los efectos negativos de las EMIs en las señales de disponibilidad, ya que se altera de forma significativa la estimación de este parámetro. Otro ejemplo de señales de disponibilidad que no corresponden con el funcionamiento real de la máquina, son las producidas a las 11:00h (2), donde se aprecian interferencias electromagnéticas que se manifiestan con valores de disponibilidad de la máquina de muy corta duración, o cambios de estado repentinos, cuando en realidad la máquina no está disponible. Además, se observan señales de sellado de corta duración que son producidas por las EMIs, y que aparecen en momentos en los que la máquina realmente no está disponible (3). En la parte final del ejemplo mostrado, se vuelve a apreciar que la máquina no está disponible (Figura 4.37(a)), cuando en realidad sí lo está (Figura 4.37(b)). De otra forma, los sellados durante ese periodo no deberían ser considerados válidos.

En este ejemplo, se demuestra el efecto negativo de las EMIs a la hora de estimar el OEE. Cabe destacar que, sin aplicar los mecanismos de filtrado propuestos, en lo referido a sellados, el sistema recoge 1125 señales de sellado. Sin embargo, únicamente tenemos 390 sellados cuando aplicamos el mecanismo de filtrado DBF. Por otro lado, el valor en disponibilidad es de un 33.06 % sin aplicar el mecanismo de filtrado SCF, y una vez aplicado, vemos que la disponibilidad real de la máquina es de un 89.94 % del periodo de dos horas analizado, lo que evidencia la eficacia y la importancia del uso de los mecanismos de filtrado de EMIs que en esta Tesis se proponen.

En la Figura 4.38 también se observa cómo el ruido producido en las señales de sellado se manifiesta en diferentes instantes, viéndose claramente una nube de puntos entre las 9:00 y 9:30. Además, también aparecen diferentes cambios en la disponibilidad, todos ellos producidos por las interferencias electromagnéticas. Más concretamente, se observan señales de sellado desde las 7:30 hasta las 8:10 y éstas se producen cuando supuestamente la máquina no está en marcha, evidenciando que el fallo se está produciendo en la disponibilidad, puesto que la máquina no podría estar sellando sin estar disponible. En la Figura 4.38(b) se aprecia claramente

CAPÍTULO 4. MECANISMOS DE FILTRADO DE EMIS: DATA BASE FILTER Y SMART CODED FILTER

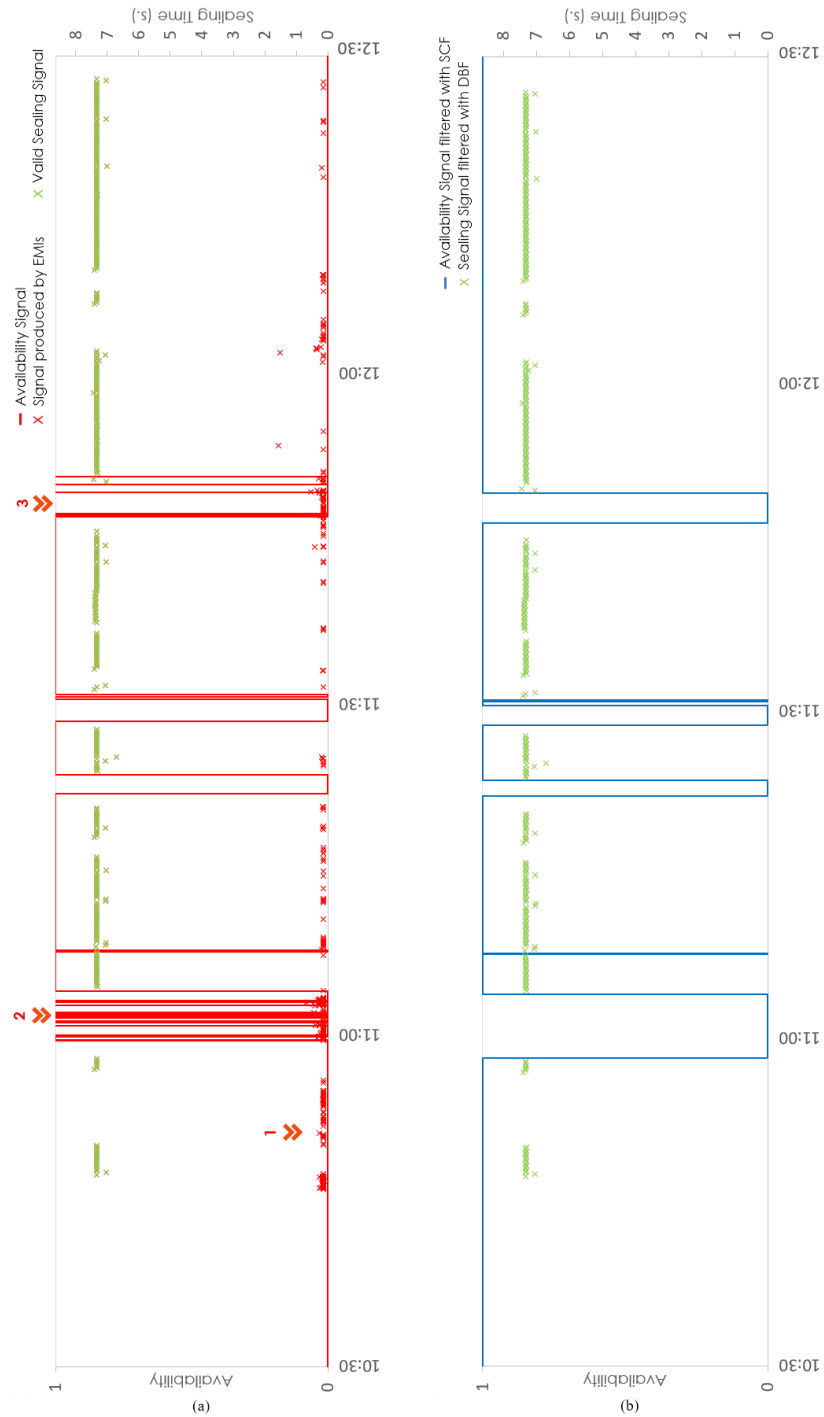


Figura 4.37: Comparación de sellados y disponibilidad en el turno 1 durante dos horas de producción: (a) Señales RAW y (b) Señales filtradas por DBF y SCF.

4.4. FUNCIONAMIENTO CONJUNTO DE LOS MECANISMOS DBF Y SCF

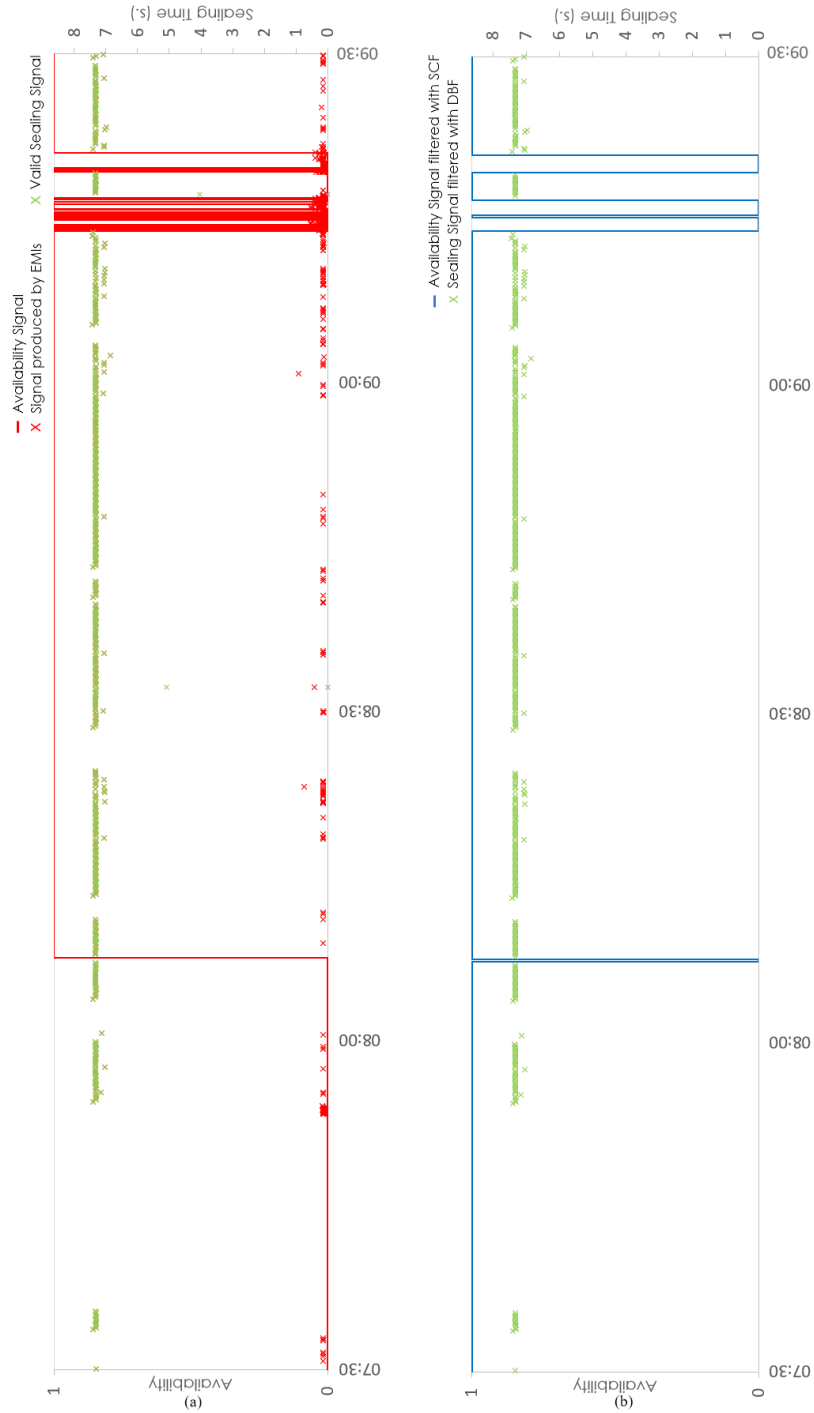


Figura 4.38: Comparación de sellados y disponibilidad en el turno 1 durante dos horas de producción: (a) Señales RAW y (b) Señales filtradas por DBF y SCF.

que la selladora está disponible, pero tras haber pasado el SCF y el DBF todo es coherente, es decir la máquina está disponible y está sellando sin ruido.

4.5. Conclusiones

En este capítulo se han presentado dos mecanismos de filtrado de EMIs, el Data Base Filter (DBF) y el Smart Coded Filter (SCF). El DBF se ha utilizado para mitigar la presencia de las EMIs en las señales de sellado, es decir, las señales que afectan a los sellados de los diferentes productos que se elaboran en la cadena de producción. Mientras, el SCF es el mecanismo de filtrado de EMIs que se ha diseñado para la eliminación de señales erróneas en la disponibilidad de la máquina.

Una vez analizados los resultados de ambos mecanismos de filtrado, se pueden destacar los siguientes aspectos:

- Las señales de sellado y disponibilidad reales, que se pueden comprobar con el vídeo grabado, mientras se realizaban los turnos, permiten verificar la fiabilidad y precisión del sistema, una vez aplicados los mecanismos de filtrado.
- El filtrado de EMIs mediante DBF se considera eficaz, puesto que presenta errores poco significativos y asumibles para un correcto cálculo del OEE. Más concretamente, los datos demuestran que si consideramos un offset de 13.04 %, el máximo error cometido es del 0.93 %. Teniendo en cuenta esto, el error cometido apenas va a afectar al cálculo del OEE.
- La escasa duración de las señales de sellado, la aparición de éstas en momentos de no disponibilidad de la máquina, así como la no pertenencia a los patrones esperados, permiten determinar con gran precisión las señales erróneas.
- El mecanismo de filtrado SCF actúa adecuadamente, filtrando las señales de disponibilidad erróneas debidas a las EMIs que se recogen cuando la máquina está inactiva. A diferencia de las señales de sellado, con este mecanismo se ha filtrado correctamente todas las señales de disponibilidad erróneas, lo que permite estimar, de forma precisa, el cálculo del OEE.

Los resultados obtenidos muestran que las señales erróneas recogidas por el sistema sin filtrar el ruido, oscilan entre un 31.98 % y un 96.25 %, mientras que el error, una vez filtradas las señales con el SCF, es del 0 % en todos y cada uno de los turnos. Además, se ha demostrado la importancia de eliminar las EMIs que afectan a las señales de disponibilidad, ya que una sola señal podría afectar de forma significativa a la disponibilidad. De hecho, los resultados indican diferencias, a la hora de usar o no el SCF, de hasta el 40.35 %, en términos de disponibilidad.

- Los resultados obtenidos demuestran que utilizando el SCF para la monitorización de las señales, somos capaces de obtener los datos exactos de disponibilidad necesarios para el cálculo del OEE, puesto que nuestra propuesta filtra completamente las señales incorrectas producidas por el ruido electromagnético.

- Se pone de manifiesto que no existe correlación entre el número de señales erróneas y los porcentajes de disponibilidad efectiva, es decir, las señales erróneas pueden afectar en mayor o menor medida a la disponibilidad estimada, si bien, queda claro que es muy importante obtener valores precisos que permitan estimar correctamente el OEE.
- El objetivo fundamental del filtrado de señales es obtener las señales sin ruido, puesto que la determinación del OEE no sería la correcta si no eliminamos las señales erróneas. No obstante, el sistema nos ofrece beneficios extra de nuestra propuesta, pues produce una reducción considerable del número de transacciones innecesarias, así como de la cantidad de datos a almacenar en la base de datos, lo que podría sobrecargar el sistema.

Capítulo 5

Detección automática de productos: MoBiSea

Durante la implantación del sistema low-cost que se presenta en esta Tesis, se han encontrado problemas de EMIs en las señales recogidas y, por ello, se han diseñado dos mecanismos de filtrado que permiten eliminar las señales erróneas en sellado (DBF) y disponibilidad (SCF).

En este capítulo presentamos una propuesta centrada en la detección automática de los productos elaborados, ya que para funcionar correctamente, el mecanismo de filtrado DBF necesita conocer los instantes de inicio y fin de la elaboración de cada tipo de producto. Hasta ahora, esos datos eran introducidos de forma manual por los operarios, por lo que se ha diseñado un algoritmo capaz de estimar de forma automática los momentos en los que se ha fabricado cada uno de los diferentes tipos de productos, sin que sea necesaria la intervención de los operarios. A este algoritmo lo hemos denominado Mode Binary Search (MoBiSea).

5.1. Introducción

MoBiSea agrupa los tiempos de sellado, con el objetivo de identificar de manera automática los diferentes productos que se han elaborado, y más concretamente, los instantes de inicio y fin en la elaboración de los mismos. Los productos que deberá seleccionar y agrupar el algoritmo son los propios de la fabricación, es decir, todos los que se elaboran durante el proceso productivo, a diferentes horas, e indistintamente en cada turno. MoBiSea nos proporciona la automatización que necesita nuestro sistema low-cost basado en Raspberry Pi, y está principalmente orientado a la medición del OEE, aunque puede ser aplicado a otros sistemas.

Para comprobar el funcionamiento de MoBiSea y validar la propuesta, se ha comparado con los algoritmos k-means, x-means y k-rms. En concreto, se ha analizado el número de señales agrupadas en cada clúster, el inicio y fin de cada grupo, así como la detección de los productos elaborados en cada turno. Además, hemos comparado los clústeres reales (es decir, los distintos productos que se han elaborado) con los detectados por nuestro algoritmo, así como los clústeres estimados

por el algoritmo x-means, que es una versión del k-means que permite determinar, de manera autónoma, el número óptimo de clústeres.

En este capítulo, se presentan de manera detallada todas las singularidades del algoritmo MoBiSea. La Sección 5.2 recoge la estructura y funcionamiento del algoritmo, así como la validación del mismo en el entorno industrial en el que se ha desplegado el sistema. Más concretamente, se muestran los resultados en los seis turnos analizados, los tiempos de procesado y las diferentes opciones en cuanto al tamaño del intervalo y al *threshold*, que son dos parámetros muy importantes de MoBiSea. En la Sección 5.3, se muestra la comparativa de rendimiento entre el MoBiSea, el x-means, el k-means y el k-rms, con el objetivo de confirmar el correcto funcionamiento del algoritmo diseñado, así como evaluar su precisión. Una vez validado y medido su rendimiento, la Sección 5.4 presenta cómo quedaría integrado MoBiSea dentro del mecanismo DBF, de forma que este último pueda funcionar de manera autónoma, sin que los operarios tengan que introducir, de forma manual, los instantes en los que se cambia el producto a fabricar. Por último, en la Sección 5.5 se presentan las conclusiones más importantes del trabajo realizado durante el presente capítulo.

5.2. MoBiSea: Mode Binary Search

El desarrollo de MoBiSea surge por la necesidad de automatizar el DBF, el mecanismo de filtrado de EMIs en las señales de sellado, que ha sido diseñado para la monitorización del OEE a través de dispositivos low-cost.

El mecanismo DBF permite filtrar las señales de sellado válidas, evitando las señales erróneas debidas a las EMIs. No obstante, es necesario destacar que una parte de éste se hacía de forma manual, ya que los operarios debían introducir en el sistema, los instantes en los que se producía un cambio de producto en el proceso de fabricación. Para solucionar este inconveniente, se propone el MoBiSea, permitiendo que esos datos puedan ser estimados automáticamente, y por tanto, el sistema funcione de manera autónoma.

El Algoritmo 8 presenta la estructura de MoBiSea. Se puede observar que está compuesto por dos funciones (*getSignalProductChange* y *getmodeChanges*) que permiten la localización de los diferentes productos en los turnos del proceso productivo, así como la identificación de los instantes exactos en los que tiene lugar la elaboración de cada producto.

En primer lugar, se inicializan y definen las variables necesarias, es decir, el número de señales que contendrá cada intervalo (*isize*) y el umbral aplicado a los valores de la moda en cada grupo de señales (*threshold*), teniendo en cuenta las señales prefiltradas. Además, se establece la estructura de datos (*im, intervalMode*), que estará compuesta por el inicio del intervalo (*start*), el final (*end*), y la moda de cada intervalo (*mode*).

El algoritmo, en primer lugar, obtiene el conjunto de señales que previamente había pre-filtrado el DBF (es decir, todas las señales de sellado, a excepción de aquellas que no seguían el patrón lógico de sellado y las que tenían una duración menor a 1 segundo (*getprefilteredSignalsDB*)). A continuación, lo almacena en *signals*, y calcula las modas de cada uno de los intervalos guardándolas en el array

Algoritmo 8 Mode Binary Search (MoBiSea)

```

/* isize: interval size                                     */
/* ims: interval modes array                               */
/* im: interval mode, each of the elements within the ims array */
/* mcs: mode changes array                                 */
/* mc: mode change, each of the elements within the mcs array */

integer isize = 50;           // [25, 50, 100, 300, 500]
float threshold = 0.05;       // [0.01, 0.05, 0.1]
structure intervalMode{
integer start;                //interval start signal
integer end;                  //interval end signal
double mode;
}

void main(){
float signals [];
intervalMode ims [];
integer mcs [];

signals [] = getprefilteredSignalsDB();
ims [] = calculateModes (0, signals.size(), isize);
mcs [] = getModeChanges(ims);
foreach (mc in mcs) do
|   im = getSignalProductChange (mc,isize)
|   insertProductTimeDB(im, shift);
end
}

integer getSignalProductChange(integer signalModeChange, integer isize){
    intervalMode imsAux [];

if (isize ≤ 1) then
|   return signalModeChange;
else
|   imsAux = calculateModes (signalModeChange - isize, signalModeChange + isize, isize/2);
|   isize = isize/2;
|   modeChangesAux[] = getModeChanges(imsAux);
end
return getSignalProductChange (modeChangesAux[0], isize);}

integer[] getModeChanges(intervalMode[] ims)
    integer[] modeChanges;
    intervalMode im, aux;
    aux = ims[0]

foreach (im in ims) do
|   if (abs(im-aux)>threshold) then
|   |   modeChanges.add(im)
|   end
|   aux = im;
end
return modeChanges;
}

```

ims. Tras esta operación, se obtienen los cambios de moda que se guardan en el array *mcs*; para ello, utiliza la función *getmodeChanges*, que es la encargada de detectar cuándo la moda de un intervalo, respecto al otro, tiene una diferencia mayor que el umbral establecido (*threshold*), siendo el *mc* cada uno de los cambios de moda que se almacenan en *mcs*. Posteriormente, se recorre el array con los cambios de moda, buscando la señal exacta en la que se produce cada cambio de producto y almacenando dicha información en la base de datos.

La función *getSignalProductChange* es una función recursiva que permite detectar exactamente cuándo se produce el cambio de producto. Para ello, toma el intervalo donde se ha producido el cambio y el anterior, dividiendo los grupos en la mitad de señales que el intervalo previo, y finaliza determinando la señal exacta donde se ha producido dicho cambio.

MoBiSea se asemeja a la búsqueda binaria (ver Sección 2.6.3) ya que, en cada iteración, divide en dos el conjunto de datos tomado anteriormente con el objetivo de encontrar la señal exacta donde se produce el cambio de producto. MobiSea presenta dos partes claramente diferenciadas: una inicial que determina el número de productos distintos que han sido elaborados durante el turno analizado, y otra que determina cuándo se producen exactamente los cambios entre los diferentes productos.

La Figura 5.1 ilustra de manera gráfica el funcionamiento de MobiSea. En ella se pueden observar las dos partes en las que se divide el proceso: la primera, que consiste en determinar el número de productos, aparece representada en la parte superior de la figura, ya que las señales se agrupan en grupos de tamaño *isize* y se calcula su moda estadística. De esta forma, cuando se observa, por ejemplo, que $(|m_\alpha - m_{\alpha+1}| > threshold)$, el algoritmo detecta un cambio de producto durante el proceso de producción (que se representa con un cambio de color en los rectángulos de la figura).

Con este primer paso, el sistema determina el número de clústeres, es decir, el número de productos distintos que han sido elaborados durante el turno analizado.

La segunda parte del algoritmo aparece reflejada en las filas que se muestran más abajo en la figura, ya que una vez detectados los cambios de producto, MobiSea realiza un proceso recursivo que permite determinar de forma exacta la señal de sellado final de un producto, y, por tanto, el comienzo del siguiente. Para ello, MoBiSea reduce a la mitad en cada iteración, el número de señales a analizar (*isize*) y recalcula las modas de cada nuevo intervalo, para así poder detectar la última señal de sellado de un tipo de producto, y por tanto, el cambio a otro distinto.

En resumen, el algoritmo se compone de las siguientes fases:

- **Recolección de Señales:** El algoritmo recoge un conjunto de señales de la base de datos, creando una secuencia de valores de señal: $s[1], s[2], \dots, s[n]$.
- **Cálculo de la Moda:** A continuación, el algoritmo divide esta secuencia de señales en intervalos de tamaño *isize* y calcula la moda de cada intervalo. Esto produce una nueva secuencia de modas $m[1], m[2], \dots, m[n/isize]$.
- **Detección de Cambios de Moda:** El algoritmo identifica los índices *i* en esta secuencia de modas, donde $|m[i] - m[i + 1]| > threshold$. Estos índices se

almacenan en el array *mcs*.

- Búsqueda Binaria para Cambios de Producto: Finalmente, el algoritmo busca el cambio exacto de producto dividiendo repetidamente el intervalo de búsqueda a la mitad, utilizando un enfoque de búsqueda binaria. Esto es equivalente a buscar el índice mínimo i en el array *mcs* tal que $|m[i]-m[i+1]| > threshold$, pero restringiendo la búsqueda a intervalos cada vez más pequeños.

5.2.1. Ejemplo de funcionamiento de MoBiSea

Teniendo en cuenta lo anterior, la Tabla 5.1 muestra, a modo de ejemplo de funcionamiento de MobiSea, algunas de las señales recogidas por nuestro sistema en el turno 5. En concreto, muestra las señales desde la número 701 a la 850. Las tres primeras columnas muestran los valores de Signal Id, Timestamp y Sealing Time, respectivamente, para el caso de las señales de la 701 a la 750, y el resto de columnas muestran la misma información, pero en el caso de las señales 751-800 y 801-850.

Si suponemos un interval size de 50, MobiSea calculará la moda correspondiente a cada uno de dichos intervalos (o grupos de señales), con el objetivo de compararlas, y ver si la diferencia es mayor que el *threshold* fijado. De este modo podrá determinar si se ha producido un cambio de moda, y cuándo ha sucedido, es decir, cuándo hay un cambio de producto.

La Figura 5.2 muestra los valores de la moda de los valores de los tiempos de sellado (*Sealing Time*) en el turno 5 con un tamaño de intervalo de 50 (es decir, se agrupan las 1671 señales de sellado en grupos de 50 señales. Como se puede observar, la variación en la moda de los tiempos de sellado nos permite determinar los distintos tipos de productos que se han elaborado en dicho turno. En este ejemplo tenemos 3 tipos de producto distintos, que se corresponden con los tiempos de sellado de 8.38, 7.34 y 6.05 segundos, respectivamente. En concreto, se observa el cambio de moda entre los intervalos de señales 701-750 y 751-800, así como en los intervalos 1151-1200 y 1201-1250.

Una vez determinado el número de clústeres del turno, el algoritmo debe detectar entre qué intervalos se producen los cambios de moda, y así poder determinar finalmente el instante en el que se produce el cambio de producto.

En el ejemplo en el que se basa la validación de la propuesta (turno 5), se detecta un primer cambio de producto entre los grupos de 50 señales que van de 701 a 750, y de 751 a 800, por lo que reduce a la mitad el *isize* para poder ajustar y determinar dónde se produce el cambio exactamente. En el otro cambio de moda (señales 1151 a 1200, y 1201 a 1250), se realizaría de la misma forma.

El resultado de reducir el tamaño del intervalo, es obtener cuatro grupos de 25 señales en los que se vuelve a calcular la moda (ver Figura 5.3). Como se observa, el cambio de moda se produce entre los intervalos de 751-775 y 776-800.

La Tabla 5.2 muestra los datos de cada una de las agrupaciones de tiempos de sellado y las frecuencias para determinar la moda con *isize* = 25.

MobiSea volvería a reducir a la mitad el tamaño de *isize*, y recalcularía la moda de las señales comprendidas entre la 751 y la 800. La Figura 5.4 presenta

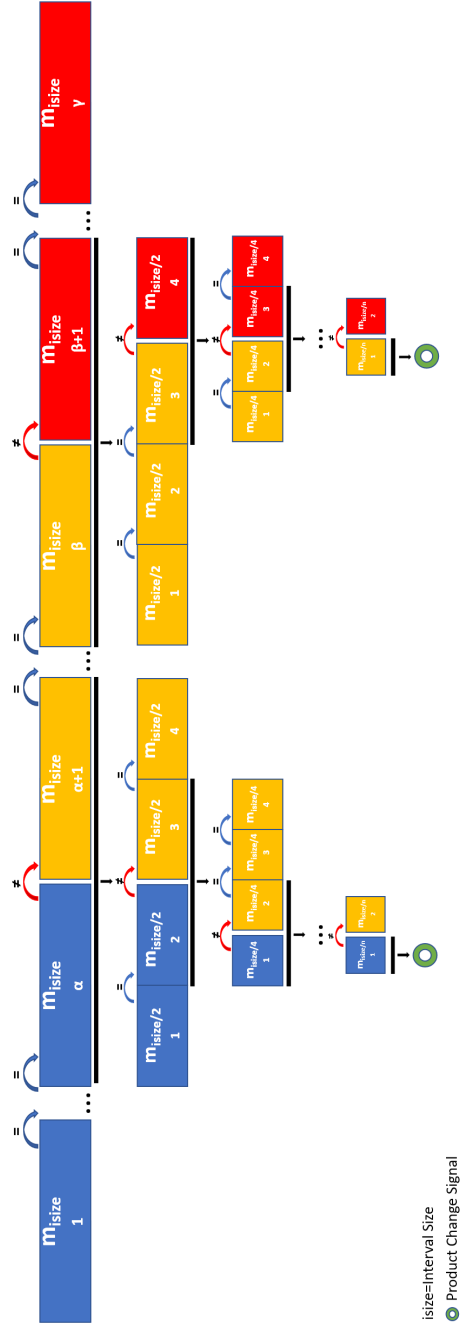


Figura 5.1: Funcionamiento de MoBiSea.

CAPÍTULO 5. DETECCIÓN AUTOMÁTICA DE PRODUCTOS: MOBISEA

Tabla 5.1: Extracto de identificación de señales, timestamp y tiempos de sellado en el turno 5

Signals 701 to 750			Signal 751 to 800			Signal 801 to 850		
Signal ID	Timestamp	Sealing Time (s.)	Signal ID	Timestamp	Sealing Time (s.)	Signal ID	Timestamp	Sealing Time (s.)
701	14:32:43.6	8.38	751	14:46:07.0	8.38	801	15:31:20.2	7.34
702	14:32:55.1	8.38	752	14:46:18.5	8.38	802	15:31:31.0	7.39
703	14:33:06.6	8.38	753	14:46:30.0	8.38	803	15:31:41.8	7.39
704	14:33:18.1	8.38	754	14:46:41.5	8.38	804	15:31:52.6	7.34
705	14:33:29.6	8.38	755	14:46:53.0	8.38	805	15:32:03.4	7.39
706	14:33:41.1	8.38	756	14:47:04.5	8.38	806	15:32:14.2	7.34
707	14:33:52.6	8.38	757	14:47:16.0	8.38	807	15:32:25.0	7.39
708	14:34:04.1	8.38	758	14:47:27.4	8.3	808	15:32:35.8	7.34
709	14:34:15.6	8.38	759	14:47:38.9	8.3	809	15:32:46.6	7.39
710	14:34:27.1	8.38	760	14:47:50.3	8.38	810	15:32:57.4	7.34
711	14:34:38.6	8.38	761	14:48:01.8	8.38	811	15:33:08.2	7.39
712	14:34:50.1	8.38	762	14:48:13.3	8.38	812	15:33:19.0	7.34
713	14:35:01.6	8.38	763	14:49:24.9	8.38	813	15:33:29.8	7.39
714	14:35:13.1	8.38	764	14:49:54.4	8.38	814	15:33:40.6	7.39
715	14:35:24.6	8.38	765	14:50:05.9	8.38	815	15:33:51.4	7.34
716	14:35:36.1	8.38	766	14:50:17.4	8.38	816	15:34:02.2	7.39
717	14:35:47.6	8.38	767	14:50:28.9	8.38	817	15:34:13.0	7.34
718	14:35:59.1	8.38	768	14:50:40.4	8.38	818	15:34:23.8	7.39
719	14:36:10.5	8.3	769	14:50:51.9	8.38	819	15:34:34.6	7.34
720	14:36:22.0	8.38	770	14:51:03.4	8.38	820	15:34:45.4	7.39
721	14:36:33.5	8.38	771	14:51:14.9	8.38	821	15:35:10.1	7.34
722	14:36:45.0	8.38	772	14:51:26.4	8.38	822	15:35:20.9	7.34
723	14:36:56.5	8.38	773	14:51:37.9	8.3	823	15:35:31.7	7.34
724	14:37:08.0	8.38	774	14:51:49.3	8.38	824	15:35:42.5	7.39
725	14:37:19.5	8.38	775	14:52:00.8	8.38	825	15:35:53.3	7.34
726	14:37:31.0	8.38	776	14:52:12.3	8.38	826	15:36:04.1	7.39
727	14:37:42.5	8.38	777	14:52:23.8	8.38	827	15:36:14.9	7.34
728	14:37:54.0	8.38	778	14:52:35.3	8.38	828	15:36:25.7	7.34
729	14:38:05.5	8.38	779	15:27:22.6	7.34	829	15:36:36.5	7.39
730	14:38:17.0	8.38	780	15:27:33.4	7.34	830	15:36:47.3	7.34
731	14:38:28.5	8.38	781	15:27:44.2	7.39	831	15:36:58.1	7.34
732	14:38:40.0	8.38	782	15:27:55.0	7.34	832	15:37:08.9	7.39
733	14:38:51.5	8.38	783	15:28:05.8	7.39	833	15:37:19.7	7.34
734	14:39:03.0	8.38	784	15:28:16.6	7.34	834	15:37:30.5	7.39
735	14:39:14.5	8.38	785	15:28:27.4	7.39	835	15:37:41.3	7.34
736	14:39:26.0	8.1	786	15:28:38.2	7.34	836	15:37:52.1	7.34
737	14:40:01.4	8.1	787	15:28:49.0	7.39	837	15:38:02.9	7.39
738	14:40:12.8	8.0	788	15:28:59.8	7.34	838	15:38:13.7	7.34
739	14:40:24.3	8.38	789	15:29:10.6	7.39	839	15:38:24.5	7.34
740	14:40:35.7	8.3	790	15:29:21.4	7.39	840	15:38:35.3	7.39
741	14:40:47.2	8.38	791	15:29:32.2	7.34	841	15:38:46.1	7.34
742	14:44:23.5	8.38	792	15:29:43.0	7.39	842	15:38:56.9	7.39
743	14:44:35.0	8.38	793	15:29:53.8	7.34	843	15:39:07.7	7.34
744	14:44:46.5	8.38	794	15:30:04.6	7.39	844	15:39:18.5	7.39
745	14:44:58.0	8.38	795	15:30:15.4	7.34	845	15:39:29.3	7.34
746	14:45:09.5	8.38	796	15:30:26.2	7.39	846	15:39:40.1	7.34
747	14:45:21.0	8.38	797	15:30:37.0	7.34	847	15:39:50.9	7.39
748	14:45:32.5	8.38	798	15:30:47.8	7.39	848	15:40:01.7	7.34
749	14:45:44.0	8.38	799	15:30:58.6	7.34	849	15:40:12.5	7.39
750	43902.6152	8.38	800	15:31:09.4	7.39	850	15:40:23.3	7.34

5.2. MOBISEA: MODE BINARY SEARCH

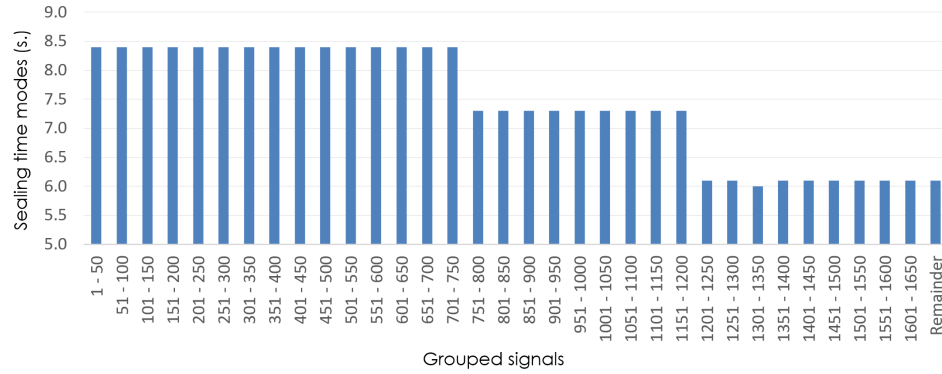


Figura 5.2: Modas de los tiempos de sellado agrupados en el turno 5 (isize = 50).

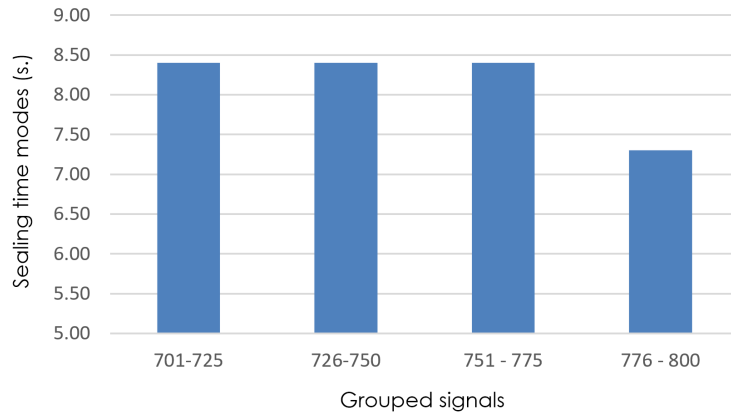


Figura 5.3: Detalle de modas de los tiempos de sellado (señales agrupadas 701-800) en el turno 5, con isize = 25.

Tabla 5.2: Grupos analizados con isize = 25

Group	Mode	Frequency
751 - 775	8.38	22
776 - 800	7.34	11
801 - 825	7.34	13
826 - 850	7.34	15

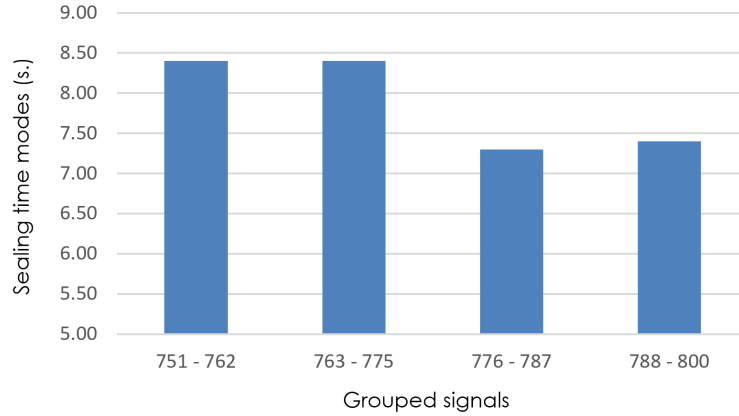


Figura 5.4: Detalle de modas de los tiempos de sellado (señales agrupadas 751-800) en el turno 5, con $isize = 12$.

Tabla 5.3: Grupos analizados con $isize = 12$

Group	Mode	Frequency
751 - 762	8.38	10
763 - 775	8.38	12
776 - 787	7.34	5
788 - 800	7.39	7

el resultado de dicho proceso, mostrando las modas de los intervalos analizados ($isize = 12$). Como se observa, el algoritmo detectaría que el cambio de producto se produce entre los grupos de señales 763-775 y 776-787.

El proceso se repetiría reduciendo el tamaño de los intervalos hasta que $isize = 1$ ya que, en ese momento, MobiSea identifica cuándo se produce exactamente el cambio de producto en el proceso de elaboración, que en el ejemplo mostrado, se produce exactamente en la señal 779.

Las Figuras 5.5, 5.6, 5.7 y 5.8 muestran cómo se irían recalculando las modas de los tiempos de sellado, de manera que se pueda observar cuándo se produce un cambio significativo en su valor, y por tanto ir reduciendo el conjunto de señales a analizar hasta encontrar dicho cambio.

La Tabla 5.3 muestra los datos de cada uno de los grupos analizados con $isize = 12$, con las frecuencias que determinan la moda de cada intervalo. Los grupos analizados con $isize = 6$, $isize = 3$, $isize = 2$ e $isize = 1$, se muestran en las Tablas 5.4, 5.5, 5.6 y 5.7 respectivamente.

5.2. MOBISEA: MODE BINARY SEARCH

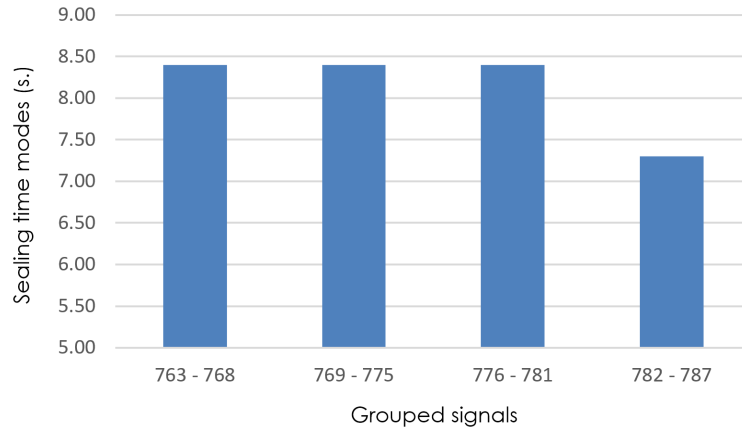


Figura 5.5: Detalle de modas de los tiempos de sellado (señales agrupadas 763-787) en el turno 5, con isize = 6.

Tabla 5.4: Grupos analizados con isize = 6

Group	Mode	Frequency
763 - 768	8.38	6
769 - 775	8.38	6
776 - 781	8.38	3
782 - 787	7.34	3

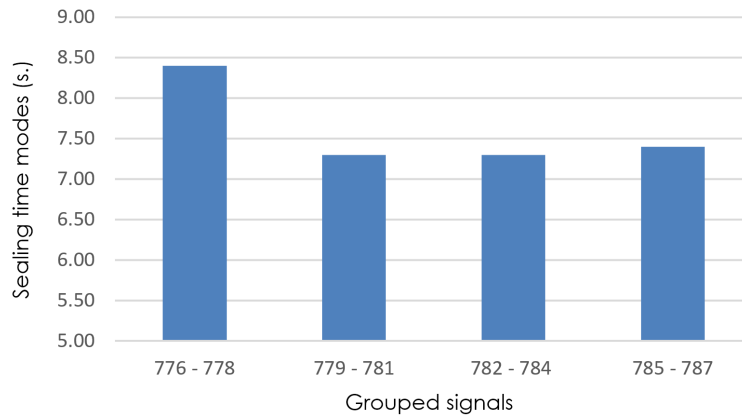


Figura 5.6: Detalle de modas de los tiempos de sellado (señales agrupadas 776-787) en el turno 5, con isize= 3.

Tabla 5.5: Grupos analizados con isize = 3

Group	Mode	Frequency
776 - 778	8.38	3
779 - 781	7.34	2
782 - 784	7.34	2
785 - 787	7.39	2

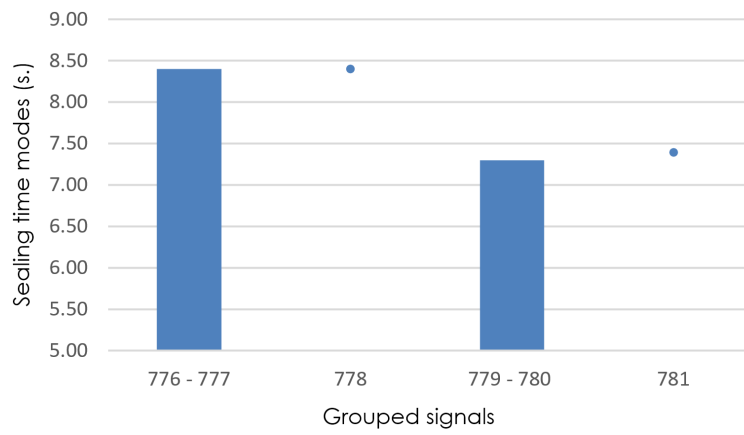


Figura 5.7: Detalle de modas de los tiempos de sellado (señales agrupadas 776-781) en el turno 5, con isize = 2.

Tabla 5.6: Grupos analizados con isize = 2

Group	Mode	Frequency
776 - 777	8.38	2
778	8.38	1
779 - 780	7.34	2
781	7.39	1

Tabla 5.7: Grupos analizados con isize = 1

Group	Mode	Frequency
778	8.38	1
779	7.34	1

5.2. MOBISEA: MODE BINARY SEARCH

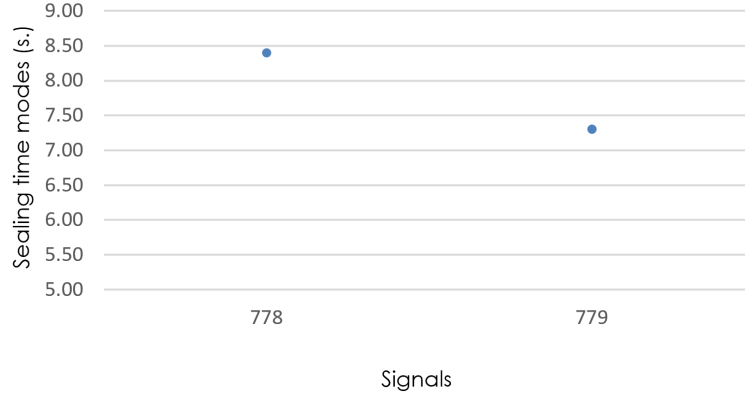


Figura 5.8: Detalle de modas de los tiempos de sellado (señales 778 y 779) en el turno 5, con $isize = 1$.

Tabla 5.8: Clústeres determinados por MoBiSea al variar el tamaño del intervalo (25, 50, 100, 300 y 500) y el threshold (0.10, 0.05 y 0.01). Los valores incorrectos aparecen en rojo

Shift	Real Clusters	Interval Size 25 threshold:			Interval Size 50 threshold:			Interval Size 100 threshold:			Interval Size 300 threshold:			Interval Size 500 threshold:		
		0.10	0.05	0.01	0.10	0.05	0.01	0.10	0.05	0.01	0.10	0.05	0.01	0.10	0.05	0.01
1	1	1	1	3	1	1	1	1	1	1	1	1	1	1	1	1
2	4	4	4	9	4	4	8	4	4	6	4	4	6	3	3	4
3	3	3	3	8	3	3	5	3	3	5	3	3	4	3	3	3
4	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
5	3	3	3	6	3	3	5	3	3	3	3	3	3	2	2	2
6	3	2	2	4	2	2	3	2	2	2	2	2	2	2	2	2

5.2.2. Validación de MoBiSea

En este apartado se presenta la validación de MoBiSea en el entorno industrial en el que se ha desplegado el sistema.

La Tabla 5.8 muestra los resultados reales (que conocemos gracias a que hemos monitorizado todo el proceso), así como los valores estimados por MoBiSea, en términos de distintos productos elaborados, para los seis turnos analizados. En lo relativo a MoBiSea, mostramos los clústeres detectados cuando variamos, tanto el tamaño del intervalo ($isize = 25, 50, 100, 300$ y 500), como la sensibilidad en la diferencia entre las modas ($threshold = 0.10, 0.05$ y 0.01). Destacar que los valores erróneos aparecen en rojo.

Como se puede observar, MoBiSea es capaz de obtener el número de clústeres (es decir, de tipos de productos distintos que han sido elaborados) en la mayoría de los turnos, y para las diferentes configuraciones. Sin embargo, hay que destacar el turno 6, en el que MoBiSea únicamente acierta para los valores de $isize = 50$ y un $threshold = 0.01$, aunque en el resto de turnos con ese mismo threshold existe una diferencia notable con la realidad. El motivo es que, en dicho turno, se elaboran

Tabla 5.9: Tiempos de ejecución de MoBiSea al variar el tamaño del intervalo

Time (s.)	Interval Size				
	25	50	100	300	500
Real	0.357	0.344	0.353	0.337	0.351
User	0.635	0.603	0.618	0.581	0.606
Sys	0.072	0.070	0.072	0.069	0.073
User + System	0.707	0.673	0.690	0.650	0.679

3 productos distintos, pero de uno de ellos únicamente se fabrican 30 unidades, lo que provoca que MobiSea no funcione bien para valores de *isize* mayores a 50.

Por otra parte, también se puede observar que cuando el valor de *threshold* es muy pequeño (i.e., 0.01), el algoritmo tampoco detecta correctamente los cambios de moda en los intervalos analizados, lo que provoca que estime de manera incorrecta los diferentes productos elaborados.

Teniendo en cuenta todo lo anterior, consideramos que el valor de *isize* deberá ser 50, ya que nos asegura un correcto funcionamiento en todo tipo de escenarios, excepto en aquéllos en los que se elaboran muy pocos productos de un mismo tipo. Además, el *threshold* recomendado es de 0.05, ya que es el menor valor posible que nos garantiza que vamos a ser capaces de identificar el número correcto de clústeres en la gran mayoría de los casos.

Una vez que hemos comprobado los valores más recomendables para los parámetros de *isize* y *threshold*, vamos a comprobar cómo afecta el tamaño de los intervalos contemplados en MobiSea en el tiempo de ejecución del algoritmo. En concreto, la Tabla 5.9 presenta los tiempos de ejecución de nuestra propuesta cuando variamos la variable *isize*.

Como se observa, el tamaño de los intervalos, no afecta en gran medida al tiempo de ejecución, ya que éste es muy similar para todos los tamaños analizados.

Por tanto, a modo de resumen, los resultados obtenidos en términos de tiempo de ejecución refuerzan la idea de que utilizar un *isize* = 50 es la mejor opción, dado que nos asegura obtener unos datos precisos, sin comprometer el tiempo de ejecución necesario para obtener la solución.

Finalmente, las Figuras 5.9 y 5.10 muestran, de manera global, las señales de sellado recogidas por el sistema, y cómo se agruparían las señales teniendo en cuenta los diferentes tamaños de intervalo. Más en detalle, la Figura 5.9 muestra en azul los tiempos de sellado para las señales que se reciben correctamente en el dispositivo, y en rojo las señales erróneas debidas a las interferencias electromagnéticas. Además, se puede observar el análisis de señales que permite detectar los cambios de producto (en este caso, quesos enteros, cuartos de queso, y cuñas). En concreto, se produce un cambio de producto a las 15:30, y otro a las 19:35.

La Figura 5.10, por su parte, permite apreciar en detalle el funcionamiento de MobiSea para poder determinar el punto exacto de cambio de producto que se produce entre las 14:30 y las 16:00 horas. Dado que el MobiSea detecta que hay un cambio de moda (i.e., la diferencia entre ellas es mayor que el *threshold* fijado) entre los dos grupos de 50 señales, éstos se dividen en grupos de 25 señales, y se determina entre cuáles de ellos se vuelve a producir un cambio de moda. El

5.2. MOBISEA: MODE BINARY SEARCH

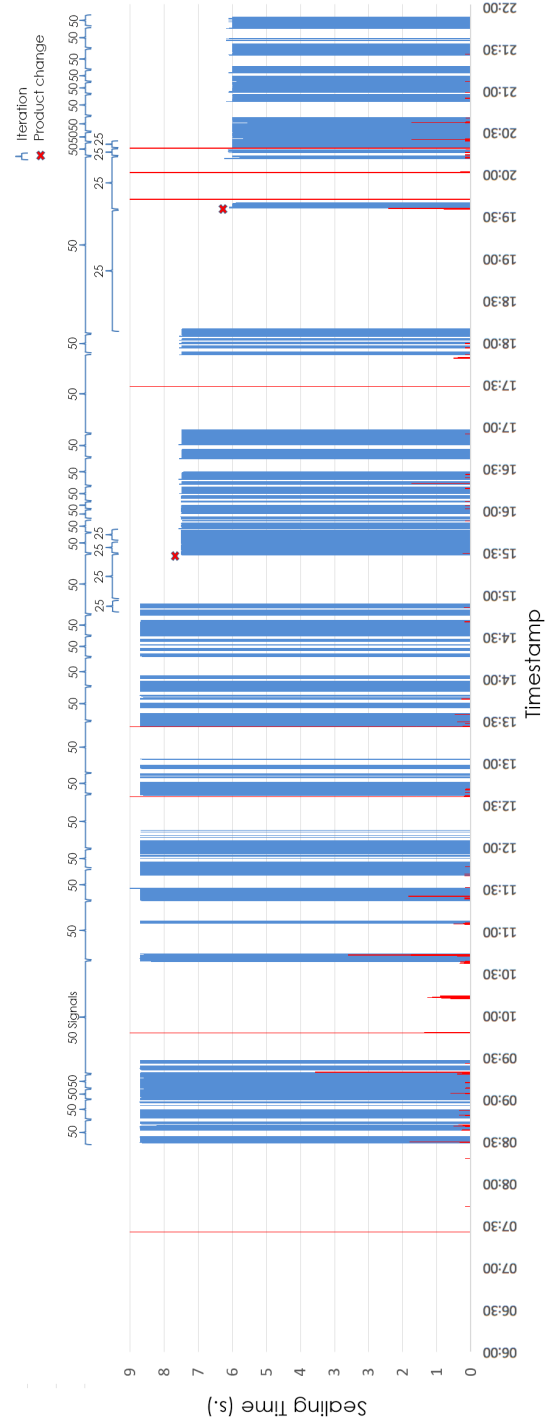


Figura 5.9: Ejemplo de todas las señales recopiladas durante el turno 5.

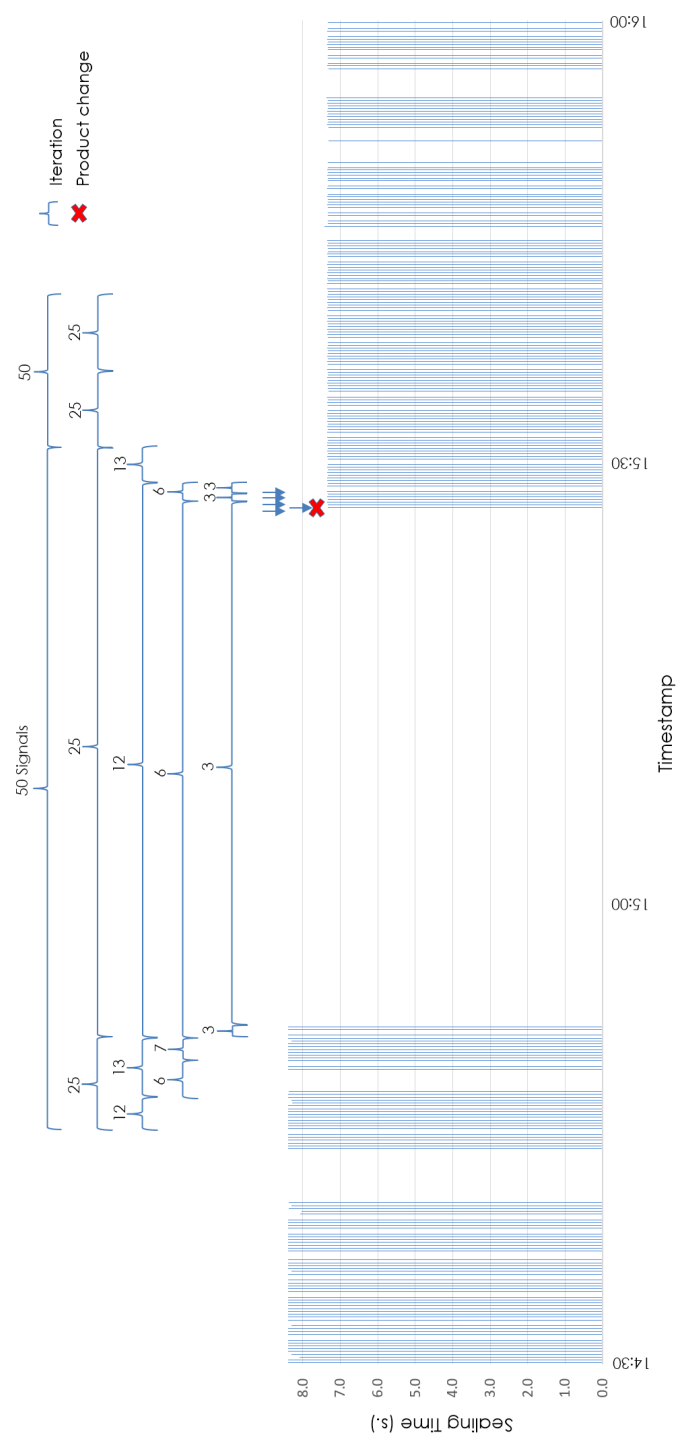


Figura 5.10: Detalle del análisis del cambio de producto.

proceso se repite, como se ha comentado anteriormente, hasta que se determina la señal exacta en la que se produce el cambio de producto.

5.3. Comparativa con otros algoritmos de agrupamiento

Con el objetivo de comprobar el rendimiento de MoBiSea con respecto a los mecanismos de agrupamiento presentados en la Sección 2.6.1, se ha realizado el análisis de seis turnos de producción distintos, comparando los valores obtenidos por MoBiSea con los obtenidos por el x-means, el k-means y el k-rms. Además, para confirmar su correcto funcionamiento y su precisión, también lo comparamos con los datos reales.

Las dos primeras métricas analizadas son el número de clústeres detectados por cada algoritmo y la posición de los centroides. Es importante señalar que el k-means y el k-rms no son capaces de determinar el número de clústeres, sino que es un parámetro que el usuario debe fijar. Por ello, para poder realizar la comparativa, se ha utilizado el número de clústeres reales. Respecto a los centroides, en el caso de MoBiSea, éstos vendrán dados por el valor de la moda de los tiempos de sellado de cada uno de los productos elaborados.

La tercera métrica a tener en cuenta es el número de señales agrupadas en cada uno de los clústeres por cada uno de los algoritmos de agrupamiento, lo que también nos permite comparar su funcionamiento. Cabe destacar que, este dato incluye las señales erróneas debidas a las EMIs con un tiempo de sellado mayor a 1 segundo, que serán filtradas posteriormente por el mecanismo DBF.

Finalmente, dado que el objetivo es conocer el momento exacto en el que se produce el cambio de tipo de producto (necesario para el correcto funcionamiento del mecanismo de filtrado DBF) también comparamos las señales exactas de inicio y final de tipo de producto, que determina cada uno de los algoritmos.

Teniendo en cuenta esto, las Tablas 5.10 - 5.12 muestran los datos obtenidos de las métricas anteriormente comentadas (i.e., el número de clústeres, la posición de los centroides, el número de señales de cada producto, así como los valores de inicio y fin de cada clúster). Además, se presentan los valores reales, de forma que se puedan medir los errores cometidos por cada uno de los algoritmos analizados (k-means, x-means, k-rms y MoBiSea).

Como se puede observar en la Tabla 5.10, el número de clústeres que determina el algoritmo x-means difiere del número de clústeres reales en cuatro de los seis turnos analizados. Cabe destacar los turnos 1, 2 y 5, en los que el x-means determina un número de clústeres que difiere bastante con el valor real. Además, se observa que los errores no siguen un patrón, es decir, x-means falla en el número de clústeres, tanto por exceso como por defecto. Sin embargo, en todos estos casos, el número de clústeres estimados por MoBiSea coincide con los valores reales, exceptuando en el turno 6, en el que MoBiSea no detecta correctamente los tres tipos de producto elaborados, debido a que el segundo de ellos consta únicamente de 30 productos, y por ello agrupa dichas señales con todas las del tercer clúster. A pesar de que el x-means sí que detecta correctamente los tres tipos de produc-

CAPÍTULO 5. DETECCIÓN AUTOMÁTICA DE PRODUCTOS: MOBISEA

Tabla 5.10: Comparativa entre los valores reales (Clusters / Center) y los resultados obtenidos por x-means, k-means, k-rms y MobiSea.

Shift	Clusters / Center				
	Real	x-means	k-means	k-rms	MoBiSea
1	1 / 7.34	1 / 7.42	1 / 7.30	1/7.33	1 / 7.34
	-	1 / 7.27	-	-	-
	-	1 / 7.24	-	-	-
2	1 / 7.34	1 / 7.08	1 / 7.22	1 / 7.04	1 / 7.34
	2 / 8.38	-	2 / 8.22	2 / 7.79	2 / 8.38
	3 / 5.96	-	3 / 5.73	3 / 6.21	3 / 5.96
	4 / 8.38	-	4 / 7.55	4 / 7.26	4 / 8.38
3	1 / 8.29	1 / 8.34	1 / 8.36	1 / 8.33	1 / 8.29
	2 / 6.05	2 / 6.08	2 / 6.23	2 / 6.13	2 / 6.05
	3 / 7.34	3 / 7.23	3 / 7.21	3 / 7.23	3 / 7.34
	-	4 / 7.12	-	-	-
4	1 / 7.34	1 / 7.23	1 / 7.23	1 / 7.28	1 / 7.34
	2 / 8.38	2 / 8.26	2 / 8.26	2 / 8.25	2 / 8.38
5	1 / 8.38	1 / 7.54	1 / 8.49	1 / 8.39	1 / 8.38
	2 / 7.34	-	2 / 7.69	2 / 7.57	2 / 7.34
	3 / 6.05	-	3 / 6.18	3 / 6.17	3 / 6.05
6	1 / 5.96	1 / 5.93	1 / 5.93	1 / 5.93	1 / 5.96
	2 / 8.38	2 / 7.47	2 / 7.46	2 / 7.11	-
	3 / 7.34	3 / 7.36	3 / 7.35	3 / 7.36	2 / 7.34

5.3. COMPARATIVA CON OTROS ALGORITMOS DE AGRUPAMIENTO

Tabla 5.11: Comparativa entre los valores reales (Number of signals (error)) y los resultados obtenidos por x-means, k-means, k-rms y MobiSea.

Shift	Number of signals (error)				
	Real	x-means	k-means	k-rms	MoBiSea
1	2046	479 (76.59 %)	2046 (0.00 %)	2046 (0.00 %)	2046 (0.00 %)
	-	996	-	-	-
	-	571	-	-	-
2	771	2588 (235.67 %)	735 (4.67 %)	1056 (36.96 %)	771 (0.00 %)
	401	-	427 (6.48 %)	537 (33.92 %)	401 (0.00 %)
	914	-	696 (23.85 %)	485 (46.94 %)	914 (0.00 %)
	502	-	730 (45.42 %)	510 (1.59 %)	502 (0.00 %)
3	533	533 (0.00 %)	533 (0.00 %)	533 (0.00 %)	533 (0.00 %)
	581	581 (0.00 %)	677 (16.52 %)	602 (3.62 %)	581 (0.00 %)
	696	497 (28.59 %)	600 (13.79 %)	675 (3.02 %)	696 (0.00 %)
	-	199	-	-	-
4	981	977 (0.41 %)	977 (0.41 %)	1002 (2.14 %)	981 (0.00 %)
	1235	1239 (0.33 %)	1239 (0.33 %)	1214 (1.70 %)	1235 (0.00 %)
5	778	1671 (114.78 %)	525 (32.52 %)	641 (17.61 %)	778 (0.00 %)
	444	-	697 (56.98 %)	581 (30.85 %)	444 (0.00 %)
	449	-	449 (0.00 %)	449 (0.00 %)	449 (0.00 %)
6	888	888 (0.00 %)	888 (0.00 %)	735 (17.23 %)	888 (0.00 %)
	30	505 (1583.33 %)	505 (1583.33 %)	658 (2093.33 %)	-
	1116	641 (42.56 %)	641 (42.56 %)	641 (42.56 %)	1146 (2.69 %)

Tabla 5.12: Comparativa entre los valores reales (Start / Finish) y los resultados obtenidos por x-means, k-means, k-rms y MobiSea.

Shift	Start / Finish				
	Real	x-means	k-means	k-rms	MoBiSea
1	1 / 2046	1 / 479	1 / 2046	1 / 2046	1 / 2046
	-	480 / 1475	-	-	-
	-	1476 / 2046	-	-	-
2	1 / 771	1 / 2588	1 / 735	1 / 1056	1 / 771
	772 / 1172	-	736 / 1162	1057 / 1593	772 / 1172
	1773 / 2086	-	1163 / 1858	1594 / 2078	1773 / 2086
	2087 / 2588	-	1859 / 2588	2079 / 2588	2087 / 2588
3	1 / 533	1 / 533	1 / 533	1 / 533	1 / 533
	534 / 1114	534 / 1114	534 / 1210	534 / 1135	534 / 1114
	1115 / 1810	1115 / 1611	1211 / 1810	1136 / 1810	1115 / 1810
	-	1612 / 1810	-	-	-
4	1 / 981	1 / 977	1 / 977	1 / 1002	1 / 981
	982 / 2216	978 / 2216	978 / 2216	1003 / 2216	982 / 2216
5	1 / 778	1 / 1671	1 / 525	1 / 641	1 / 778
	779 / 1222	-	526 / 1222	642 / 1222	779 / 1222
	1223 / 1671	-	1223 / 1671	1223 / 1671	1223 / 1671
6	1 / 888	1 / 888	1 / 888	1 / 735	1 / 888
	889 / 918	889 / 1393	889 / 1393	736 / 1393	-
	919 / 2034	1394 / 2034	1394 / 2034	1394 / 2034	889 / 2034

5.3. COMPARATIVA CON OTROS ALGORITMOS DE AGRUPAMIENTO

tos que fueron elaborados, sorprendentemente, estima erróneamente el número de unidades elaboradas de cada uno de ellos (o lo que es lo mismo, el número de señales de cada clúster). Además, en lo relativo a los centroides, vemos cómo MoBiSea detecta de forma satisfactoria los tiempos de sellado de los diferentes tipos de producto que han sido elaborados en todos los turnos, a excepción del turno 6, por sus características especiales anteriormente comentadas.

Respecto al número de señales de cada clúster, es decir, el número de sellados que se han recibido para cada uno de los tipos de productos elaborados, vemos que obviamente, para el caso del x-means, los errores se producen de forma mayoritaria cuando se había determinado un número de clústeres erróneo. Sin embargo, también vemos que estima correctamente (o con un error mínimo) el número de señales en algunos clústeres.

Para el caso del k-means, el error cometido es relativamente pequeño en la mitad de los turnos, aunque dicho error es más destacable para los turnos 2, 5 y especialmente el 6. En concreto en el turno 2, el error cometido en los clústeres 3 y 4, es del 23.85 % y del 45.42 %, respectivamente. Por su parte, en el turno 5, el error es mayor en los clústeres 1 y 2, con un error del 32.52 % y del 56.98 %, respectivamente. Finalmente, en el turno 6, los clústeres 2 y 3 presentan un error incluso mayor, llegando al 1583.33 % para el segundo clúster.

En el caso de k-rms, los errores más significativos también ocurren en los turnos 2, 5 y 6, especialmente en este último. En el turno 2, el clúster 3 presenta un error del 46.94 %, seguido del clúster 1 con un error del 36.96 %. En el turno 5, el mayor error está en el clúster 2 con un error del 30.85 % y finalmente, el clúster 2 en el turno 6 presenta un error del 2093.33 %.

Por otra parte, las Figuras 5.11 - 5.16 presentan de manera visual los valores que aparecen en las Tablas 5.10 - 5.12, a excepción del error cometido, para cada uno de los seis turnos analizados. En concreto, dichas figuras permiten observar los valores determinados por los métodos de agrupamiento analizados, en lo relativo a los diferentes clústeres, el número de señales de cada uno de ellos, así como los instantes de inicio y fin de cada uno de los distintos productos.

En lo que respecta al turno 1, la Figura 5.11 muestra cómo el x-means estima tres clústeres diferentes, cuando en realidad sólo se elabora un producto, tal y como detecta el MoBiSea.

En la Figura 5.12, relativa al turno 2, puede observarse la discrepancia que existe entre el x-means y los otros tres algoritmos, el k-means, k-rms y el MoBiSea, puesto que el x-means detecta un solo tipo de producto, cuando realmente se han elaborado tres distintos. Eso sí, nótese que el producto B se fabrica en dos instantes distintos dentro de ese turno. Es decir, se elabora el producto A, el B, el C y luego el producto B otra vez. Aunque a los algoritmos k-means y k-rms se les ha indicado manualmente el número de clústeres real, se observa la diferencia entre el inicio y fin de cada uno de los tipos de producto determinada por el k-means, k-rms y el MoBiSea, que también para este turno, estima de manera exacta dichos valores.

También se muestran discrepancias entre el x-means, el k-means, el k-rms y MoBiSea en el turno 3 (ver Figura 5.13). En concreto, se observa la diferencia en términos de tipos de productos detectados por el x-means (i.e., cuatro), mientras que MoBiSea detecta tres. Además, se evidencia la diferencia en cuanto a

CAPÍTULO 5. DETECCIÓN AUTOMÁTICA DE PRODUCTOS: MOBISEA

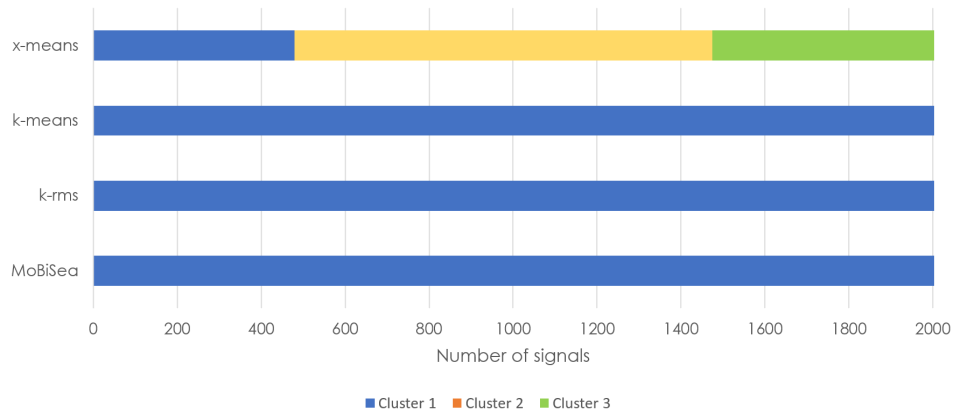


Figura 5.11: Productos estimados por x-means, k-means, k-rms y MoBiSea en el turno 1.

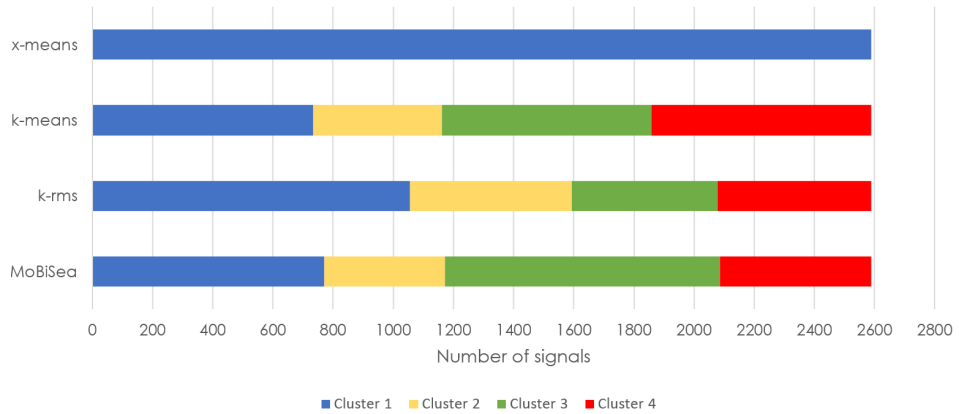


Figura 5.12: Productos estimados por x-means, k-means, k-rms y MoBiSea en el turno 2.

5.3. COMPARATIVA CON OTROS ALGORITMOS DE AGRUPAMIENTO

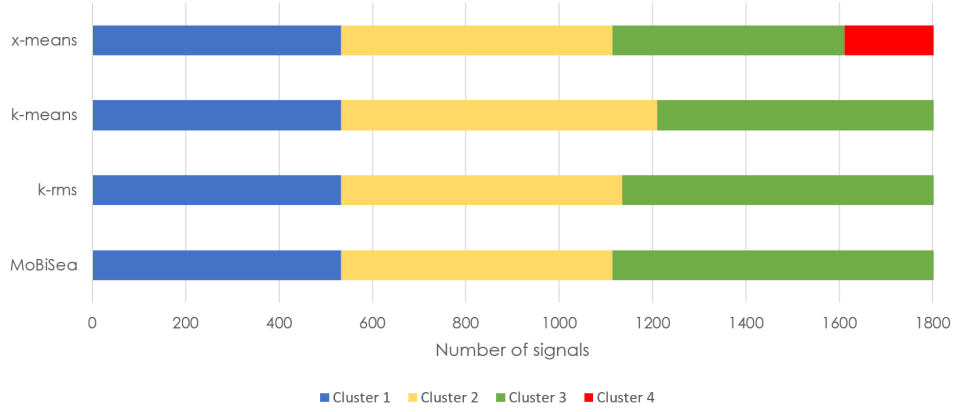


Figura 5.13: Productos estimados por x-means, k-means, k-rms y MoBiSea en el turno 3.

los instantes de inicio y final en la elaboración de los productos que estiman los mecanismos k-means y x-means, con las que detecta el MoBiSea, que coinciden plenamente con la realidad. Mientras, el k-rms es el que más se aproxima a los resultados obtenidos por MoBiSea, que coinciden plenamente con los datos reales.

La Figura 5.14 muestra los datos recogidos durante el turno 4, en el que existe una gran similitud entre los cuatro algoritmos. No obstante, cabe destacar que a la hora de estimar el número de señales agrupadas en cada uno de los clústeres, tanto el x-means como el k-means, cometen un pequeño error (inapreciable en la figura), mientras que en el caso de k-rms, sí que se puede observar el error cometido a la hora de agrupar los productos elaborados en el primer clúster.

De la misma manera, la Figura 5.15 muestra cómo el x-means determina un único clúster para el turno 5, mientras que MoBiSea estima de manera correcta que se elaboraron tres tipos distintos de producto. Además, vemos cómo el k-means y k-rms agrupan de manera incorrecta las señales que forman parte de cada uno de los clústeres, especialmente en el caso de los dos primeros productos; si bien, cabe destacar que el error cometido por el k-rms es sensiblemente menor que el cometido por el k-means.

Finalmente, la Figura 5.16 muestra como el x-means acierta a la hora de determinar el número de clústeres, mientras que el MoBiSea solo estima dos de los tres tipos de producto que fueron elaborados en el turno 6. Sin embargo, es necesario destacar que ninguno de los otros tres algoritmos de agrupamiento (es decir, el x-means, el k-means y el k-rms) tampoco determinan correctamente el número de señales de los clústeres 2 y 3. En el caso de k-rms, llama la atención cómo tampoco lo hace bien para el clúster 1. No obstante, cabe destacar que en este turno se produce un hecho poco común, ya que únicamente se elaboraron 30 unidades del segundo producto, lo que dificulta en gran manera el correcto funcionamiento de los algoritmos, incluido el MoBiSea.

CAPÍTULO 5. DETECCIÓN AUTOMÁTICA DE PRODUCTOS: MOBISEA

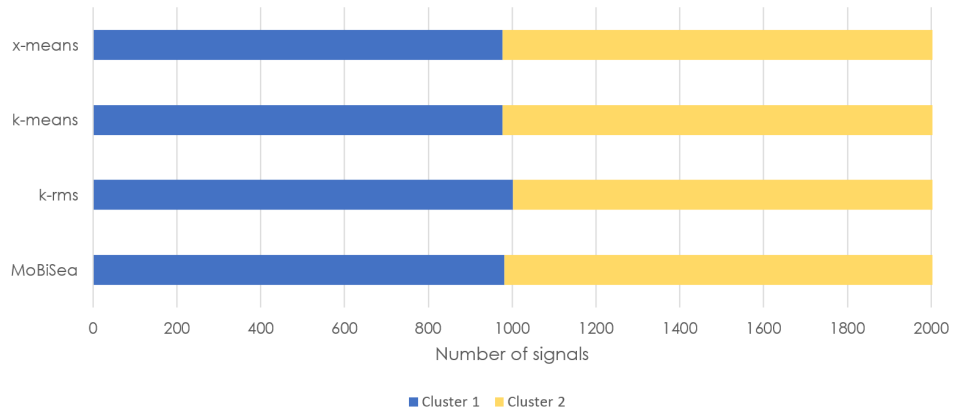


Figura 5.14: Productos estimados por x-means, k-means, k-rms y MoBiSea en el turno 4.

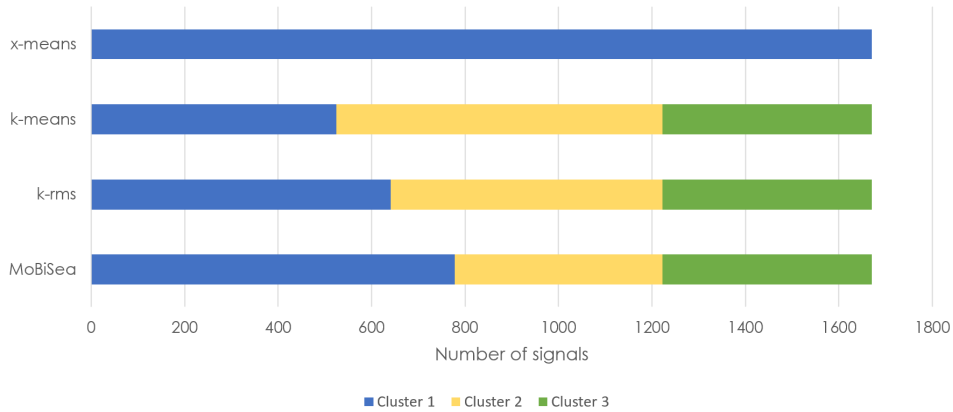


Figura 5.15: Productos estimados por x-means, k-means, k-rms y MoBiSea en el turno 5.

5.4. INTEGRACIÓN DE MOBISEA EN EL MECANISMO DE FILTRADO DBF

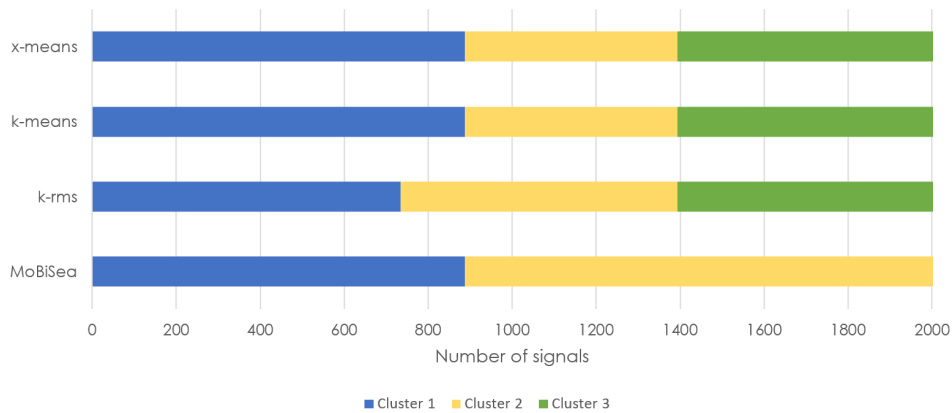


Figura 5.16: Productos estimados por x-means, k-means, k-rms y MoBiSea en el turno 6.

5.4. Integración de MoBiSea en el mecanismo de filtrado DBF

Tal y como se ha indicado anteriormente, MoBiSea ha sido diseñado para permitir que el mecanismo DBF funcione de manera autónoma, es decir, que no sea necesario que los operarios introduzcan de forma manual en el sistema, los instantes de inicio y fin de cada tipo de producto, o lo que es lo mismo, los momentos en los que se produce un cambio de producto en el proceso de fabricación.

Una vez presentado en detalle y validado el algoritmo MoBiSea, en esta sección se va a presentar cómo quedaría integrado dentro del mecanismo de filtrado DBF, de forma que éste último pueda trabajar sin necesidad de supervisión humana.

La Figura 5.17 muestra cómo quedarían finalmente las distintas fases del funcionamiento del DBF. En primer lugar, el mecanismo DBF descartará las señales erróneas que no atienden al orden lógico que muestran las señales de sellado, es decir, un 0 seguido de otro 0 o dos 1's consecutivos (`filterLogicSignals`). Además, también realizará el pre-filtrado de todas las señales con un tiempo de sellado inferior a un segundo (`getCandidateSignalsDB`), dado que se ha evidenciado durante la puesta en marcha del sistema que ningún sellado dura menos de un segundo.

En una segunda fase, entrará en acción el algoritmo MoBiSea, que tras analizar todas las señales de sellado que hayan superado el proceso de pre-filtrado, determinará de manera inicial el número de tipos de productos distintos que han sido elaborados durante el turno analizado, así como los instantes de inicio y fin de cada uno de los mismos, evitando que los operarios tengan que introducir esos datos de forma manual.

Finalmente, la tercera fase consistirá en el filtrado de las señales erróneas restantes, es decir, aquellas que no han sido pre-filtradas. Para ello, se identificarán los valores de tiempos de sellado que más se repiten (moda), se establecerán los límites inferior y superior al aplicar el offset fijado, y se eliminarán todas las señales cuyo tiempo de sellado sea mayor o menor al rango establecido. De esta forma, se

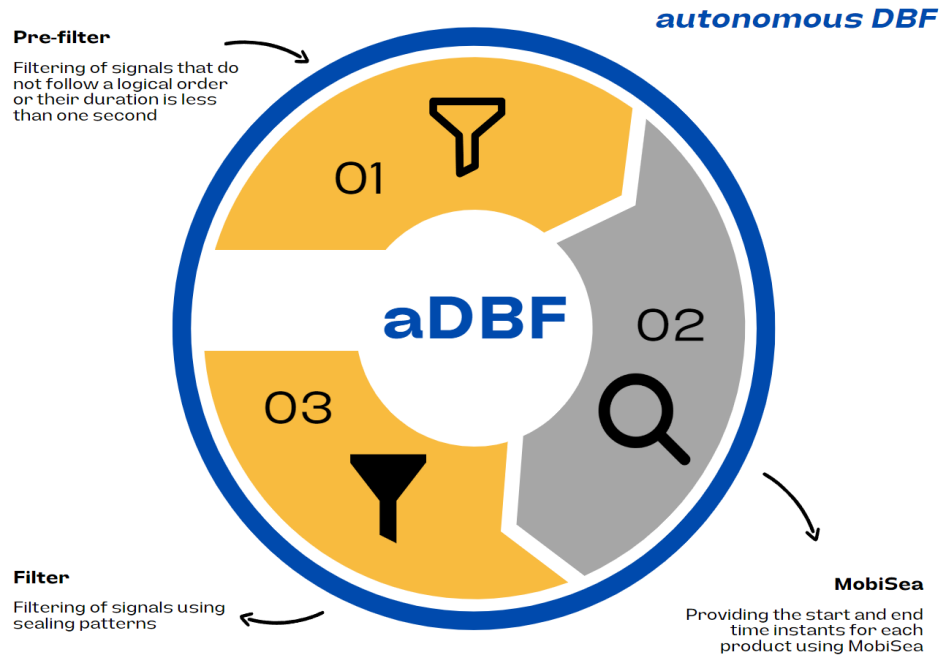


Figura 5.17: Esquema de integración de MoBiSea en DBF.

obtendrán las señales consideradas válidas para cada uno de los distintos productos elaborados en dicho turno, lo que permite estimar de forma precisa el número de productos que realmente han sido elaborados durante todo el turno analizado.

Como vemos, MoBiSea queda perfectamente integrado en el mecanismo de filtrado DBF, convirtiéndolo en un mecanismo de filtrado completamente autónomo.

5.5. Conclusiones

En este capítulo se ha presentado una propuesta centrada en la automatización del mecanismo de filtrado DBF. Concretamente, se ha diseñado el algoritmo MoBiSea, capaz de estimar de forma automática los diferentes tipos de producto elaborados, así como el tiempo durante el cual se elabora cada uno de los mismos. Esta información será de utilidad para que el mecanismo de filtrado de señales de sellado DBF pueda funcionar de manera autónoma, sin que los operarios tengan que introducir esos datos en el sistema de forma manual. Además, se ha presentado la validación de MoBiSea en el entorno industrial en el que se ha desplegado el sistema propuesto, comparándolo con otros algoritmos de agrupamiento, como son el k-means, el x-means o el k-rms.

Los resultados obtenidos indican que el número de clústeres estimados por el x-means no es el correcto en la mayoría de los casos analizados. Además, muestran que el k-means, el x-means y el k-rms, no son capaces de estimar correctamente el número de señales en cada uno de los clústeres, y, por tanto, el instante de inicio

5.5. CONCLUSIONES

y fin en la elaboración de cada producto. Por el contrario, MoBiSea es capaz de determinar con precisión el número de clústeres, así como los instantes de inicio y fin de la elaboración de todos los productos y turnos, excepto para el turno 6, que supone un caso singular, ya que únicamente se elaboraron 30 unidades de uno de los productos.

Capítulo 6

Conclusiones, Publicaciones y Trabajo futuro

6.1. Conclusiones

Esta Tesis presenta una solución low-cost para la medición de parámetros relativos a la productividad en entornos industriales, más concretamente de las variables que determinan el OEE, un índice que permite a los responsables de las empresas detectar qué aspectos deben analizar en detalle para poder mejorar la eficacia del proceso productivo, y así incrementar la rentabilidad.

Las interferencias electromagnéticas representan un claro enemigo para los dispositivos de bajo coste aplicados a la industria. La ubicación encapsulada de los dispositivos low-cost en entornos industriales, que conviven con elementos generadores de gran cantidad de ruido eléctrico, se torna como problema a eliminar para que puedan implantarse este tipo de dispositivos más económicos, en sustitución de otros sistemas propietarios, menos vulnerables a las interferencias, pero de mayor coste.

Para solucionar este problema, se han propuesto dos mecanismos capaces de filtrar las interferencias electromagnéticas en las señales recibidas por el dispositivo, es decir, las señales relativas al proceso de sellado y las relativas a la disponibilidad de la máquina. El primero de ellos es el Data Base Filter (DBF), que realiza un filtrado atendiendo a los patrones de sellado convencionales de la máquina y actúa sobre los datos insertados en la base de datos, eliminando los datos erróneos. El segundo, el Smart Coded Filter (SCF), permite filtrar totalmente las interferencias electromagnéticas, obteniendo la disponibilidad real de la máquina.

Hemos podido comprobar la efectividad de nuestra propuesta, tras su implantación en diferentes líneas de una industria láctea, evidenciando la escalabilidad y robustez de nuestro sistema. Los resultados obtenidos reportan una gran fiabilidad, teniendo un error máximo en la obtención de señales de sellado del 0.93 % tras la aplicación del mecanismo de filtrado DBF, y en la señal de disponibilidad una precisión total, es decir, un error del 0 %.

Por otra parte, el mecanismo de filtrado DBF precisaba de cierta automati-

zación en sus funciones, puesto que era necesario introducir los datos relativos a la elaboración de los distintos productos (específicamente, los instantes de inicio y final de su elaboración), de forma manual por parte de los operarios. Por ello, esta Tesis presenta el algoritmo MoBiSea, capaz de estimar de forma automática los tiempos de sellado de los productos, así como los instantes de inicio y fin en la elaboración de cada producto. Además, MoBiSea es capaz de estimar de forma precisa el número de productos elaborados en cada turno, con un error máximo del 2.69 %.

Para validarlo, se han comparado los resultados obtenidos con MoBiSea con otros algoritmos de clusterización como el k-means, el x-means y el k-rms. La precisión de nuestro algoritmo es total, ya que los valores obtenidos coinciden exactamente con los reales para todos los turnos estudiados, excepto para el turno 6, que presenta un caso singular, dado que sólo se fabricaron 30 unidades de uno de los productos. Sin embargo, los datos obtenidos por x-means, k-means y k-rms presentan resultados erróneos, a pesar de incluso indicarle previamente al k-means y al k-rms el número de clústeres correcto.

6.2. Publicaciones relacionadas con la Tesis

El trabajo de investigación relacionado con esta Tesis ha resultado en seis publicaciones; entre ellas tenemos dos artículos de revista, ambas indexadas en la base de datos de Journal Citation Reports (JCR), dos congresos internacionales de reconocido prestigio y dos congresos nacionales. A continuación presentamos cada uno de los trabajos escritos, junto a una breve descripción de los mismos.

6.2.1. Revistas

- [HSM⁺21] A. C. Herrero, J. A. Sanguesa, F. J. Martinez, P. Garrido and C. T. Calafate, “Mitigating Electromagnetic Noise when Using low-cost Devices in Industry 4.0”, in *IEEE Access*, vol. 9, pp. 63267-63282, 2021, doi: 10.1109/ACCESS.2021.3074588.

La transición hacia la Industria 4.0 requiere una gran inversión en dispositivos y mecanismos que permitan la interconectividad entre las personas, máquinas y procesos. En este trabajo se presenta un sistema de bajo coste basado en Raspberry Pi, para la medición en tiempo real del Overall Equipment Effectiveness (OEE), y se proponen dos mecanismos de filtrado de Interferencias Electromagnéticas (EMIs) para poder medir el OEE de forma precisa. El primer mecanismo de filtrado de EMIs, el Data Base Filter (DBF), se ha diseñado para recoger de forma precisa las señales de sellado. En concreto, este mecanismo opera sobre la base de datos, filtrando las señales erróneas que hayan sido insertadas en ella. El segundo mecanismo, el Smart Coded Filter (SCF), se emplea para el filtrado de señales erróneas en las mediciones de disponibilidad de las máquinas.

Hemos validado nuestra propuesta en varias líneas de producción de una industria alimentaria, y los resultados obtenidos demuestran el correcto fun-

cionamiento de nuestro sistema, que reduce considerablemente los costes de implantación en comparación a sistemas propietarios con similares funciones.

Los resultados obtenidos, tras la implantación del sistema propuesto en entornos industriales reales, presentan un error medio del -0.43 % y un Root Mean Square Error (RMSE) de 4.85 en las señales de sellado, y un error del 0 % en la señal de disponibilidad, lo que permite realizar una estimación precisa del OEE.

IEEE Access es una revista de acceso abierto, multidisciplinar y en línea, que presenta los resultados de investigaciones o desarrollos originales en todos los campos de interés de IEEE. Sus características distintivas son una revisión por pares y la disponibilidad de sus artículos de forma gratuita para todos los lectores.

IEEE Access está indexada en el JCR del año 2021 con un factor de impacto de 3.476, y una posición 79/164 (Q2) en la categoría *Computer Science, Information Systems*.

- [HSG⁺23] A. C. Herrero, J. A. Sanguesa, P. Garrido, F. J. Martinez, and C. T. Calafate, “MoBiSea: a Binary Search algorithm for product clustering in Industry 4.0”, *Electronics* 2023, 12(15), 3262. <https://doi.org/10.3390/electronics12153262>

Los sistemas propietarios utilizados para la modernización de la Industria 4.0 suelen implicar costes económicos elevados. Debido a esto, existe una corriente incipiente en el uso de dispositivos low-cost, capaces de sustituir a estos sistemas propietarios con un menor coste, pero con las mismas funcionalidades. Sin embargo, estos dispositivos low-cost suelen presentar problemas de interferencias electromagnéticas (EMIs) debidas al encapsulamiento en cuadros eléctricos y su convivencia con otros dispositivos electromecánicos.

En este artículo presentamos MoBiSea (Mode Binary Search), un algoritmo que ha sido especialmente diseñado para su uso en un sistema low-cost de recogida automática de los datos de productividad industrial. En concreto, los datos de productividad se obtienen a partir de las señales de disponibilidad y sellado de las máquinas de sellado termoplástico existentes en las líneas de producción en la industria agroalimentaria. MoBiSea ha sido diseñado para agrupar las señales de sellado, lo que nos permite identificar qué productos han sido fabricados. Además, el algoritmo determina de manera no supervisada, sin la necesidad de intervención por parte de los operarios, el inicio y fin de la producción de cada producto, es decir, el instante en el que se produce un cambio de producto.

Finalmente, comparamos nuestro algoritmo, basado en la búsqueda binaria, con tres mecanismos de agrupamiento como son el k-means, el x-means y el k-rms, obteniendo unos resultados mejores, ya que somos capaces de identificar el número de productos con un error máximo del 2.69 %.

Electronics es una revista internacional, revisada por pares y de acceso abierto, sobre la ciencia de la electrónica y sus aplicaciones, publicada quincenal-

mente en línea. Sus áreas temáticas son circuitos y dispositivos eléctricos, microelectrónica y tecnología de la computación, ciencia e ingeniería de la computación, comunicaciones y procesamiento de información, comunicaciones de ingeniería eléctrica, procesamiento de señales, tecnología de mediciones, ingeniería de sistemas de microondas y electrónicos, microelectrónica y optoelectrónica, bioelectrónica, electrónica de potencia y sistemas de energía. En el JCR del año 2022, la revista *Electronics* tiene un Factor de Impacto de 2.9, con una posición en el ranking de 131/275 (Q2), en la categoría *Engineering, Electrical & Electronic*.

6.2.2. Congresos internacionales

- [HMG⁺20] A. C. Herrero, F. J. Martinez, P. Garrido, J. A. Sanguesa and C. T. Calafate, “An interference-resilient IIoT solution for measuring the effectiveness of industrial processes”, the 46th Annual Conference of the IEEE Industrial Electronics Society (IECON), 2020, pp. 2155-2160, doi: 10.1109/IECON43393.2020.9254454.

El desarrollo y despliegue del llamado Internet Industrial de las Cosas (IIoT) ha aumentado significativamente las capacidades de control y seguimiento de las empresas y, por tanto, su productividad potencial. En este artículo, proponemos el uso de dispositivos Raspberry Pi en entornos industriales para medir parámetros de productividad. Nuestra propuesta puede recopilar de manera económica y eficiente datos relacionados con la disponibilidad y productividad de maquinaria industrial. Sin embargo, dado que los dispositivos de bajo coste son propensos a sufrir los efectos negativos de las interferencias electromagnéticas, adicionalmente proponemos una alternativa para prevenir las alteraciones de señal causadas por las mismas. Más específicamente, proponemos un mecanismo de filtrado llamado Smart Coded Filter (SCF), que elimina las señales erróneas causadas por interferencias electromagnéticas y, por lo tanto, mejora considerablemente la precisión al estimar la métrica de disponibilidad.

Los resultados obtenidos demuestran que nuestro dispositivo de bajo coste provisto con el SCF ignora por completo el 100 % de los datos de disponibilidad erróneos, al tiempo que reduce hasta en un 70 % la cantidad de registros almacenados en la base de datos.

La *Annual Conference of the IEEE Industrial Electronics Society* (IECON) es una conferencia centrada en temas de la industria contemporánea que abarcan, desde la electrónica, los controles y la fabricación, hasta las comunicaciones y la inteligencia computacional. IECON es la conferencia anual más importante de la Industrial Electronics Society. Su objetivo es crear un foro en el que científicos e ingenieros de todo el mundo puedan presentar los resultados de sus últimas investigaciones e ideas en el campo de la electrónica industrial, así como posibles contribuciones al desarrollo sostenible y la conservación del medio ambiente.

CAPÍTULO 6. CONCLUSIONES, PUBLICACIONES Y TRABAJO FUTURO

De acuerdo con Google scholar, esta conferencia tiene un h5-index de 27.

- [HSM⁺23] A. C. Herrero, J. A. Sanguesa, F. J. Martinez, P. Garrido, and C. T. Calafate, “aDBF: an autonomous electromagnetic noise filtering mechanism for industrial environments”, the 32nd International Conference on Computer Communications and Networks (ICCCN 2023), pages 1-7, Waikiki Beach, Honolulu, HI, USA, 2023, doi: 10.1109/ICCCN58024.2023.10230132.

El uso de sistemas propietarios en la Industria 4.0 suele conllevar elevados costes económicos. Para solucionar este problema, cada vez es más popular el uso de dispositivos de bajo coste, que presentan capacidades similares. Sin embargo, estos dispositivos son propensos a sufrir los efectos negativos de las interferencias electromagnéticas, debido a su ubicación en cuadros eléctricos junto a otros dispositivos electromecánicos. Para resolver este problema, este artículo presenta el autonomous Data Base Filter (aDBF).

El aDBF es un mecanismo mejorado de filtrado de interferencias electromagnéticas, capaz de eliminar las señales erróneas generadas por el ruido electromagnético. El aDBF ha sido específicamente diseñado para determinar de forma autónoma (es decir, sin necesidad de supervisión o intervención de ningún operario), tanto el número de tipos de productos diferentes elaborados en una línea de producción, como los momentos en los que comienza y finaliza su proceso de fabricación. En concreto, aDBF pasa por tres etapas: (i) la de pre-filtrado, (ii) la de detección de cambios de producto, y (iii) la de identificación de señales válidas. Los resultados obtenidos, tras validar nuestra propuesta en tres turnos de fabricación diferentes, demuestran que el mecanismo de filtrado aDBF funciona con gran precisión, ya que el error máximo cometido es del 0.93 %.

La *International Conference on Computer Communications and Networks* (ICCCN) es un destacado evento global dedicado a exhibir ideas innovadoras y progreso pionero en el ámbito de las comunicaciones y redes informáticas. Este foro fomenta la colaboración entre investigadores y profesionales que están unidos por un objetivo común para mejorar la comunicación y la red mediante innovaciones científicas y tecnológicas. El principal énfasis de la conferencia reside en mostrar nuevos y originales hallazgos de investigación en las áreas de diseño, implementación y aplicación de comunicaciones y redes informáticas.

Esta conferencia, según el GII-GRIN-SCIE (GGS) Conference Rating, es Class 3. De acuerdo al Computing Research & Education (CORE) ranking del año 2021 está indexada como CORE B, y finalmente, según Google scholar, esta conferencia tiene un h5-index de 26.

6.2.3. Congresos nacionales

- [HMGS19] A. C. Herrero, F. J. Martinez, P. Garrido, and J. A. Sanguesa, “Medición de la eficiencia industrial mediante dispositivos de bajo coste”, Jornadas SARTECO 2019, pp. 569-576.

La cuarta revolución industrial, conocida como Industria 4.0, es una realidad creada a partir de la evolución e innovación tecnológica. Aspectos como la conectividad entre máquinas, personas y productos, el análisis en tiempo real de multitud de variables y el uso de técnicas basadas en Inteligencia Artificial, entre otras, han dotado a la industria de una enorme versatilidad, permitiendo la mejora continua en los procesos. Por contra, la implantación de novedades tecnológicas en la industria suele acarrear una gran inversión y quizá sea éste el motivo principal por el que en muchos ámbitos, no se haya llegado a implantar.

En el presente trabajo, proponemos un dispositivo de bajo coste basado en el uso de Raspberry Pi, capaz de calcular el índice de la efectividad global de la maquinaria industrial (OEE). El uso de nuestro sistema permitirá mejorar la capacidad de reacción de la empresa, pues posibilitará detectar los aspectos que se deben mejorar para aumentar globalmente la productividad.

Los resultados obtenidos demuestran que el sistema propuesto es robusto y que las mediciones son precisas, habiendo reducido notablemente el coste necesario, especialmente si lo comparamos con otras soluciones similares basadas en dispositivos propietarios.

SARTECO, la Sociedad Española de Arquitectura y Tecnología de Computadores es una asociación con carácter científico y académico. Contribuye con el desarrollo científico y tecnológico español en el área de la Arquitectura y Tecnología de Computadores, promueve la investigación, la innovación y la transferencia de tecnologías. Una de las actividades que organiza SARTECO, son las Jornadas de Paralelismo (JP) y las Jornadas de Computación Empotrada y Reconfigurable (JCER), en los que se enmarcan tanto la publicación aquí mencionada, como la que se presenta a continuación.

- [HMG⁺22] A. C. Herrero, F. J. Martinez, J. A. Sanguesa, P. Garrido, and C. T. Calafate, “Estimación automática de los productos elaborados en la Industria 4.0”, Jornadas SARTECO 2022, pp. 887-892.

Los sistemas propietarios utilizados para la modernización de la Industria 4.0 suelen requerir unos costes económicos elevados. Debido a esto, existe una corriente incipiente en el uso de dispositivos low-cost, capaces de sustituir a estos sistemas propietarios con un menor coste, pero con unas funcionalidades similares. Sin embargo, estos dispositivos low-cost suelen presentar problemas de interferencias electromagnéticas (EMIs) debidas al encapsula-

miento de dichos dispositivos en cuadros eléctricos y su convivencia con otros dispositivos electromecánicos.

En este artículo presentamos un algoritmo que ha sido especialmente diseñado para su uso en un sistema low-cost de recogida automática de los datos de productividad industrial. El algoritmo determina de manera no supervisada, es decir, sin la necesidad de intervención por parte de los operarios, el inicio y fin de la producción de cada producto mejorando las técnicas empleadas habitualmente para este tipo de entornos.

6.3. Trabajo futuro

Una vez alcanzados los diferentes objetivos de esta Tesis Doctoral de forma satisfactoria, consideramos que existen una serie de líneas de investigación que quedan abiertas y que podrán realizarse a partir de ahora.

Como futuras líneas de trabajo, en primer lugar, se pretende analizar el comportamiento de toda la arquitectura low-cost, presentada en esta Tesis, en otras áreas de producción industrial, ya que consideramos que tanto el sistema propuesto, los mecanismos de filtrado DBF y SCF, así como el algoritmo MoBiSea, pueden ser utilizados en otras cadenas de producción.

Por otro lado, dado que se desea estimar el OEE de forma no supervisada, y en este momento es posible cuantificar de forma automática dos de las tres variables necesarias para determinar el OEE (i.e., la Disponibilidad y el Rendimiento), queda por resolver la obtención de datos relativos a la variable Calidad, ya que en la actualidad, dichos datos se introducen de forma manual. Para ello, será necesario determinar de forma automática el tiempo perdido debido a la elaboración de productos defectuosos. En este sentido se trabajará en un dispositivo low-cost capaz de leer los códigos de barras de las cajas terminadas, para poder determinar la diferencia de productos correctamente finalizados frente a los esperados.

Otro objetivo futuro, es realizar mediciones de otras métricas dentro de la Industria 4.0, utilizando para ello dispositivos low-cost. Además, se pretende probar tanto la arquitectura propuesta, como los mecanismos de filtrado, con otros dispositivos low-cost distintos a la Raspberry Pi (por ejemplo, Arduino o NodeMCU).

Habiendo alcanzado todos los objetivos predefinidos, consideramos que el objetivo final de esta Tesis se ha alcanzado con éxito, y así concluimos esta disertación.

Bibliografía

- [Bis95] C. M. Bishop. *Neural networks for pattern recognition*. Oxford university press, 1995.
- [Bly20] Blynk. we make Internet of Things simple, 2020. Disponible en <https://blynk.io/>.
- [BYZX17] Y. Bai, L. Yang, G. Zhang, and Y. Xu. An improved binary search RFID anti-collision algorithm. In *12th International Conference on Computer Science and Education (ICCSE)*, pages 435–439, 2017.
- [COM⁺15] M. Cruz, P. Olete, C. Morales, C. González, B. Cendón, and A. Hernández. Las tecnologías IoT dentro de la Industria Conectada 4.0. *Gobierno de España, Ministerio de Industria, Energía y Turismo, Escuela de Organización Industrial (EOI)*. Disponible en: <http://a.eoi.es/industria4>, 2015.
- [CSPA18] G. Caiza, M. Saeteros, C. Pillajo, and G.V. Arévalo. Development of an IEC-61499 PID control for industrial processes applications. *IEEE Colombian Conference on Communications and Computing (COLCOM)*, pages 1–5, 2018.
- [CZC19] H. Chen, W. Zhang, and R. Cheng. Electromagnetic interference reduction from printed circuit boards. In *2019 IEEE 3rd Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC)*, pages 1045–1048, 2019.
- [DBAF18] L. S. Dalenogare, G. B. Benitez, N. F. Ayala, and A. G. Frank. The expected contribution of Industry 4.0 technologies for industrial performance. *International Journal of production economics*, 204:383–394, 2018.
- [DHS73] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern classification and scene analysis*, volume 3. Wiley New York, 1973.
- [Ele21] Mouse Electronics. Industrial EMC – is Raspberry Pi the victim of EMI?, 2021. Disponible en <https://www.mouser.com/pdfDocs/brainboxespiandemc.pdf>.

BIBLIOGRAFÍA

- [FA12] L. J. Ferrer Arnau. Introducción a la compatibilidad electromagnética. In *VIII Jornades de Conferències d'Enginyeria Electrònica*, pages 13–16, 2012.
- [FSAD19] M. Fekri Sari and S. Avakh Darestani. Fuzzy overall equipment effectiveness and line performance measurement using Artificial Neural Network. *Journal of Quality in Maintenance Engineering*, 25(2):340–354, 2019.
- [GD20] A. Garain and D. Das. K-rms algorithm. *Procedia Computer Science*, 167:113–120, 2020. International Conference on Computational Intelligence and Data Science.
- [GNZ⁺18] C. A. Garcia, J. E. Naranjo, T. P. Zambrano, D. Lanás, and M. V. Garcia. Low-cost cyber-physical production systems for industrial control robots under IEC 61499. In *2018 IEEE 23rd International Conference on Emerging Technologies and Factory Automation (ETFA)*, volume 1, pages 1281–1284, Sep. 2018.
- [Gob15] Gobierno de España. La transformación digital de la industria española. Informe preliminar. *Industria Conectada 4.0*, 4:120, 2015. Disponible en: <https://www.industriaconectada40.gob.es/SiteCollectionDocuments/informe-industria-conectada40.pdf>.
- [GSS21] H. D. Ghael, L. Solanki, and G. Sahu. A review paper on Raspberry Pi and its applications. *International Journal of Advanced Engineering and Management*, 2(12):225–227, 2021.
- [HLL17] J. Hochdörffer, C. Laule, and G. Lanza. Product variety management using data-mining methods — reducing planning complexity by applying clustering analysis on product portfolios. In *IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)*, pages 593–597, 2017.
- [HMG⁺20] A. C. Herrero, F. J. Martínez, P. Garrido, J. A. Sanguesa, and C. T. Calafate. An interference-resilient IIoT solution for measuring the effectiveness of industrial processes. In *46th Annual Conference of the IEEE Industrial Electronics Society (IECON)*, pages 2155–2160, 2020.
- [HMG⁺22] A. C. Herrero, F. J. Martínez, P. Garrido, J. A. Sanguesa, and C. T. Calafate. Estimación automática de los productos elaborados en la industria 4.0. In *Jornadas SARTECO*, pages 1–6, 2022.
- [HMGS19] A. C. Herrero, F. J. Martínez, P. Garrido, and J. A. Sanguesa. Medición de la eficiencia industrial mediante dispositivos de bajo coste. In *Jornadas SARTECO*, pages 569–576, 2019.

- [HSG⁺23] A. C. Herrero, J. A. Sanguesa, P. Garrido, F. J. Martinez, and C. T. Calafate. MoBiSea: A binary search algorithm for product clustering in Industry 4.0. *Electronics*, 12(15), 2023.
- [HSM⁺21] A. C. Herrero, J. A. Sanguesa, F. J. Martinez, P. Garrido, and C. T. Calafate. Mitigating electromagnetic noise when using low-cost devices in industry 4.0. *IEEE Access*, 9:63267–63282, 2021.
- [HSM⁺23] A. C. Herrero, J. A. Sanguesa, F. J. Martinez, P. Garrido, and C. T. Calafate. aDBF: an autonomous electromagnetic noise filtering mechanism for industrial environments. In *32nd International Conference on Computer Communications and Networks (ICCCN 2023)*, pages 1–7, Waikiki Beach, Honolulu, HI, USA, 2023.
- [HYA18] A. Hüseyinli, O. Yildiz, and M. A. Akcayol. Specification based automatic product categorization from unstructured data. In *26th Signal Processing and Communications Applications Conference (SIU)*, pages 1–4, 2018.
- [IKH10] K. Imamura, N. Kubo, and H. Hashimoto. Automatic moving object extraction using x-means clustering. In *28th Picture Coding Symposium*, pages 246–249, 2010.
- [Ins22] Instituto Nacional de Estadística. Internet de las cosas (IoT), 2022. Disponible en: <https://www.ine.es/jaxi/Datos.htm?tpx=49862>.
- [Ish00] T. Ishioka. Extended k-means with an efficient estimation of the number of clusters. In *Intelligent Data Engineering and Automated Learning-IDEAL 2000. Data Mining, Financial Engineering, and Intelligent Agents*, pages 17–22. Springer Berlin Heidelberg, 2000.
- [JAD17] A. E. Jacob, N. Ashodariya, and A. Dhongade. Hybrid search algorithm: Combined linear and binary search algorithm. In *International Conference on Energy, Communication, Data Analytics and Soft Computing (ICECDS)*, pages 1543–1547, 2017.
- [Jol21] J. W. Jolles. Broad-scale applications of the Raspberry Pi: A review and guide for biologists. *Methods in Ecology and Evolution*, 2021.
- [Kau16] S. Kaushik. An introduction to clustering and different methods of clustering. *Analytics Vidhya*, 3, 2016.
- [Knu99] D. E. Knuth. The art of computer programming. *Sorting and searching*, 3:426–458, 1999.
- [LFK⁺14] H. Lasi, P. Fettke, H.G. Kemper, T. Feld, and M. Hoffmann. Industry 4.0. *Business & Information Systems Engineering*, 6(4):239–242, 2014.

BIBLIOGRAFÍA

- [LWHZ16] C. Li, J. Wu, Y. Huang, and W. Zhu. Characterization of change in microcontroller susceptibility during accelerated aging. In *Asia-Pacific International Symposium on Electromagnetic Compatibility (AP EMC)*, volume 1, pages 751–754. IEEE, 2016.
- [Mar22] Marcos Conde. Queserías de Zamora, 2022. Disponible en: <https://www.marcosconde.com/>.
- [Min21] Ministerio de Industria Comercio y Turismo. Programas de apoyo Secretaría General de Industria y de la PYME, 2021. Disponible en: <https://www.industriaconectada40.gob.es/programas-apoyo/Paginas/programas.aspx>.
- [Min22] Ministerio de Industria Comercio y Turismo. HADA - herramienta de autodiagnóstico digital avanzada, 2022. Disponible en: <https://hada.industriaconectada40.gob.es/hada/>.
- [MJ16] A. P. Mynster and P. T. Jensen. EMC for the IoT. In *Electromagnetic Compatibility-EMC EUROPE, 2016 International Symposium on*, pages 144–149. IEEE, 2016.
- [MKV⁺17] V. Mach, S. Kovář, J. Valouch, M. Adámek, and R. M. S. Silva. Electromagnetic compatibility of Raspberry Pi development platform in near and far-field. In *Progress in Electromagnetics Research Symposium-Fall (PIERS-Fall)*, pages 2466–2472. IEEE, 2017.
- [Nak88] S. Nakajima. *Introduction to TPM: total productive maintenance*. Productivity Press, 1988.
- [NM18] F. Noorbehbahani and S. Mansoori. A new semi-supervised method for network traffic classification based on x-means clustering and label propagation. In *2018 8th International Conference on Computer and Knowledge Engineering (ICCKE)*, pages 120–125, 2018.
- [NY18] N. Z. Noor Hasnan and Y. M. Yusoff. Short review: Application areas of industry 4.0 technologies in food processing sector. In *2018 IEEE Student Conference on Research and Development (SCOReD)*, pages 1–6, 2018.
- [Obs22] Observatorio Nacional de Tecnología y Sociedad (2022). Uso de tecnologías digitales por empresas en España. *Ministerio de Asuntos Económicos y Transformación Digital*, 2022.
- [OGPS18] M. O. Ojo, S. Giordano, G. Procissi, and I. N. Seitanidis. A review of low-end, middle-end, and high-end IoT devices. *IEEE Access*, 6:70528–70554, 2018.
- [Omr22] Omron. MY4IN 24VAC (S), 2022. Disponible en: <https://industrial.omron.es/es/products/MY4IN-24VAC-S>.

- [Ora22] Oracle. Cloud Applications and Cloud Platform, 2022. Disponible en: <https://www.oracle.com/>.
- [OZAD17] N. A. Othman, M. R. Zainodin, N. Anuar, and N.S. Damanhuri. Remote monitoring system development via Raspberry Pi for small scale standalone PV plant. In *7th IEEE International Conference on Control System, Computing and Engineering (ICCSCE)*, pages 360–365. IEEE, 2017.
- [PAMBEU20] J. C. Peralta-Abarca, B. Martínez-Bahena, and J. Enríquez-Urbano. Industria 4.0. *Inventio*, 16(39):1–7, 2020.
- [Pau06] C. R. Paul. *Introduction to electromagnetic compatibility*, volume 184. John Wiley & Sons, 2006.
- [Pod18] Poder Industrial, 2018. Disponible en: <https://poderindustrial.com/mexico-requiere-de-cuatro-cuatro-pilares-para-avanzar-a-la-industria-4-0/>.
- [QZL⁺08] X.H. Qiu, Y. Zhao, S.J. Li, N.Q. Jiang, and X.H. Wu. An conducted electromagnetic interference EMI noise source modeling method using Hilbert transform. In *International Conference on Microwave and Millimeter Wave Technology*, volume 3, pages 1438–1441. IEEE, 2008.
- [Ras22] Raspberry Foundation. Raspberry Pi, 2022. Disponible en: <https://www.raspberrypi.com/>.
- [RBK19] S. Ramalingam, K. Baskaran, and D. Kalaiarasan. IoT enabled smart industrial pollution monitoring and control system using Raspberry Pi with blynk server. In *International Conference on Communication and Electronics Systems (ICCES)*, pages 2030–2034, 2019.
- [RCM⁺18] J. I. Rodríguez, J. M. Cueva, C.E. Montenegro, J. J. Rainer, and R. González. Metamodel for integration of Internet of Things, Social Networks, the cloud and Industry 4.0. *Journal of ambient intelligence and humanized computing*, 9(3):709–723, 2018.
- [RIAI16] A. Raza, A.A. Ikram, A. Amin, and A.J. Ikram. A review of low cost and power efficient development boards for iot applications. In *Future Technologies Conference (FTC)*, pages 786–790, 2016.
- [RNASK20] U. Rahamathunnisa, M.K. Nallakaruppan, A. Anith, and K.S. Sendhil Kumar. Vegetable disease detection using k-means clustering and SVM. In *6th International Conference on Advanced Computing and Communication Systems (ICACCS)*, pages 1308–1311, 2020.
- [Sch18] Schneider Electric. Compatibilidad Electromagnética (CEM), 2018. Disponible en <https://product-help.schneider-electric.com/>.

BIBLIOGRAFÍA

- [SDAP18] S. Sankaran, K. Deshmukh, M. B. Ahamed, and S.K. Pasha. Recent advances in electromagnetic interference shielding properties of metal and carbon filler reinforced flexible polymer composites: a review. *Composites Part A: Applied Science and Manufacturing*, 114:49–71, 2018.
- [SPAI15] J. Siswanto, A. S. Prabuwo, A. Abdullah, and B. Idrus. Automatic image segmentation using Sobel operator and k-means clustering: A case study in volume measurement system for food products. In *International Conference on Science in Information Technology (ICSITech)*, pages 13–18, 2015.
- [Sta22] Statista. Internet of Things Spain - Previsión de mercado en España, 2022. Disponible en: <https://es.statista.com/outlook/tmo/internet-of-things/spain#revenue>.
- [STM19] STMicroelectronics. STM32, 2019. Disponible en: <https://www.st.com/en/microcontrollers-microprocessors/stm32-32-bit-arm-cortex-mcus.html>.
- [STSS19] C. Seguna, L. Tanti, J. Scerri, and K. Scicluna. A low-cost real time monitoring system for an industrial mini-climatic chamber. In *IECON 2019-45th Annual Conference of the IEEE Industrial Electronics Society*, volume 1, pages 3045–3050. IEEE, 2019.
- [TJP⁺13] J. M. Thomassin, C. Jerome, T. Pardo, C. Bailly, I. Huynen, and C. Detrembleur. Polymer/carbon based composites as electromagnetic interference EMI shielding materials. *Materials Science and Engineering: R: Reports*, 74(7):211–232, 2013.
- [ULM22] ULMA Packaging. Diseño y fabricación de máquinas de envasado, 2022. Disponible en: <https://www.ulmapackaging.com/>.
- [Vuy19] G. M. Vuyyuru. KLP’s search algorithm - a new approach to reduce the average search time in binary search. In *4th International Conference on Electrical, Electronics, Communication, Computer Technologies and Optimization Techniques (ICEECOT)*, pages 185–190, 2019.
- [Wil06] R. M. Williamson. Using overall equipment effectiveness: the metric and the measures. *Strategic Work System, Inc*, pages 1–6, 2006.
- [XQL⁺19] P. Xiao, Y. Qiu, Z. Liu, L. Bian, and G. Li. A method of analyzing the impact of the wiring parameters on the electromagnetic coupling to PCB inside electronic equipment. In *12th International Workshop on the Electromagnetic Compatibility of Integrated Circuits (EMC Compo)*, pages 1–3, 2019.

- [YSY⁺10] Z. Yang, X. Shaojun, S. Yan, L. Xiaoquan, D. Yinghua, J. Ningqiu, L. Shijin, and Y. Wei. High performance EMI noise separating network and its application in power electronics EMC. *Journal of Nanjing University of Aeronautics & Astronautics*, 42(1):37–42, 2010.