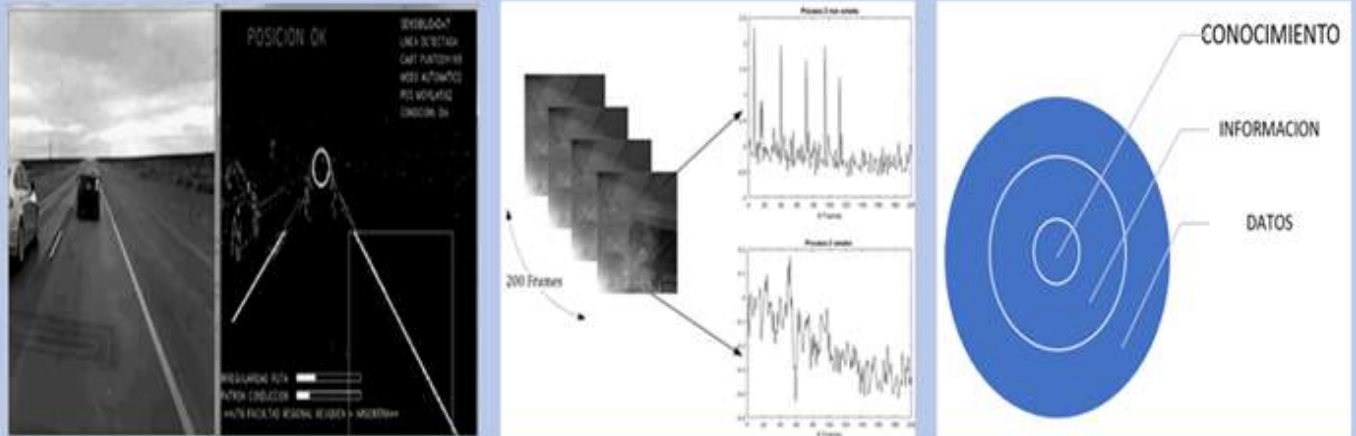


REPRESENTACIÓN SIMPLIFICADA DE SEÑALES UNIDIMENSIONALES Y BIDIMENSIONALES PARA LA EXTRACCIÓN DE INFORMACIÓN EN TIEMPO REAL



Universidad Nacional de Mar del Plata

Tesis del Doctorado en Ingeniería, orientación Electrónica

Msc. Gustavo Monte

Directores: Dra. Virginia Ballarín

Dr. Juan Ignacio Pastore

2022



RINFI se desarrolla en forma conjunta entre el INTEMA y la Biblioteca de la Facultad de Ingeniería de la Universidad Nacional de Mar del Plata.

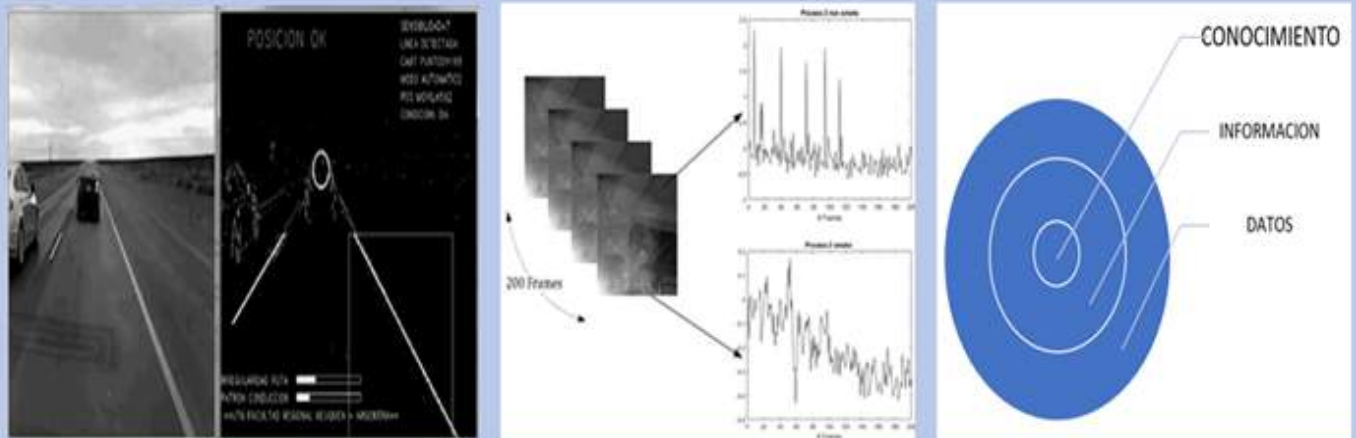
Tiene como objetivo recopilar, organizar, gestionar, difundir y preservar documentos digitales en Ingeniería, Ciencia y Tecnología de Materiales y Ciencias Afines.

A través del Acceso Abierto, se pretende aumentar la visibilidad y el impacto de los resultados de la investigación, asumiendo las políticas y cumpliendo con los protocolos y estándares internacionales para la interoperabilidad entre repositorios



Esta obra está bajo una [Licencia Creative Commons Atribución-
NoComercial-CompartirIgual 4.0 Internacional](https://creativecommons.org/licenses/by-nc-sa/4.0/).

REPRESENTACIÓN SIMPLIFICADA DE SEÑALES UNIDIMENSIONALES Y BIDIMENSIONALES PARA LA EXTRACCIÓN DE INFORMACIÓN EN TIEMPO REAL



Universidad Nacional de Mar del Plata

Tesis del Doctorado en Ingeniería, orientación Electrónica

Msc. Gustavo Monte

Directores: Dra. Virginia Ballarín

Dr. Juan Ignacio Pastore

2022

Tabla de contenido

<u>Aportes de esta tesis</u>	8
<u>Capítulo 1. Introducción</u>	14
1.1. Redundancia	16
1.2. Datos, información y conocimiento	16
<u>Capítulo 2. Muestreo completo de señales sensoriales unidimensionales</u>	23
2.1. Introducción	23
2.2. Descripción de los algoritmos	30
2.3. Ejemplo del algoritmo aplicado a una señal empleando MatLab®	39
2.4. Extracción de características de la señal	42
2.4.1. Máximos locales	42
2.4.2. Mínimos locales	42
2.4.3. Puntos de inflexión	42
2.4.4. Codificación de una señal sinusoidal	42
2.5. Patrones de comportamiento	43
2.5.1. Supersegmentos	43
2.5.2. Detección de ruido impulsivo	45
2.5.3. Estimación de tendencia	46
2.5.4. Estimación de ancho de banda para segmentos de señal	47
2.5.5. Estimación de frecuencia y fase de un patrón oscilatorio	48
2.6. Análisis de los errores de interpolación	52
2.7. Representación formal de la señal sensorial	52
2.8. Reconstrucción de la señal a partir de las muestras esenciales	53
2.9. Estrategias de selección de muestras esenciales	56
2.9.1. Condición de sobremuestreo	57
2.9.2. Condición de no sobremuestreo	60
2.10. Conclusiones	61

<u>Capítulo 3. Aplicaciones en señales unidimensionales</u>	63
3.1. Filtrado de señales empleando representación MCT	63
3.2. Análisis de señales de vibraciones para detectar y clasificar fallas en rodamientos	67
3.3 Análisis de señales biológicas: Aplicación de la representación MCT a la detección de fibrilación ventricular en señales de electrocardiograma	72
3.3.1. Detección de patrones QRS	76
3.3.2. Conclusiones	81
3.4. Determinación de la velocidad de un móvil procesando el sonido pasante	81
3.4.1 Introducción	82
3.4.2. Características de las señales sonoras	83
3.4.3. Algoritmo detector de envolvente	85
3.4.4. Algoritmo discriminador de sonido	85
3.4.5. Combinación de los dos algoritmos y filtrado	87
3.4.6. Filtrado MCT y análisis de la pendiente	89
3.4.7 Resultados experimentales	91
3.4.8. Conclusiones	92
<u>Capítulo 4. Aplicaciones en imágenes y video</u>	92
4.1. Aplicación en video: Estimación en tiempo real de la posición de un vehículo en ruta	92
4.1.1. Introducción	92
4.1.2. Procesamiento de puntos notables o píxeles esenciales	96
4.1.3. Algoritmos para la detección de líneas	99
4.1.4. Validación del algoritmo propuesto	102
4.1.5. Resultados experimentales	104
4.1.6. Implementación para los ensayos	106
4.1.7. Conclusiones	108
4.2. Aplicación en imágenes: Procesamiento de Imágenes Térmicas aplicado a cámaras infrarrojas de Bajo Costo	109
4.2.1. Introducción a las cámaras infrarrojas	109
4.2.2. Procesamiento de la imagen térmica	113
4.2.3. Máximos de la imagen	113
4.2.4 Algoritmos de extracción de características	113
4.2.5. Validación de la escena	114
4.2.6. Análisis térmico de frames validados	114
4.2.7. Resultados experimentales sobre imágenes reales	114
4.2.8. Conclusiones	118
<u>Capítulo 5. Conclusiones y trabajo futuro</u>	119

5.1. Conclusiones generales	119
5.2 Trabajo futuro y desarrollos en curso	120
5.2.1. Señales sensoriales industriales	121
5.2.2. Procesamiento de señales sísmicas	124
5.2.3. Detección de humo en video	126
<u>Referencias</u>	130
<u>Anexo A</u>	135

Tabla de figuras

Figura. 1.1. La señal como fuente de datos, información y conocimiento.	17
Figura 1.2. Los cuatro pilares de la IoT	20
Figura 2.1. La información que transporta una señal generalmente se encuentra oculta. Procesar una señal implica extraer la información embebida en el núcleo de ella.	23
Figura 2.2. El conversor analógico digital es el elemento interfase híbrido (entrada analógica- salida digital)	24
Figura 2.3. El muestreo inteligente debe “ver” a la señal completa, no solo una muestra a digitalizar.	27
Figura 2.4. El muestreo completo debe proporcionar además del valor de la muestra información de la relación entre ellas y el instante de ocurrencia.	29
Figura 2.5 Interpolación lineal. Proceso inicial, se interpola la segunda muestra y se la compara con la real.	33
Figura 2.6 Interpolación lineal. Se interpola la tercera muestra y se la compara con la real.	34
Figura 2.7. Interpolación lineal. El proceso se repite hasta alcanzar la cota de error	35
Figura 2.8. Los tres tipos de segmentos que se generan cuando no se supera la cota de error.	35
Figura 2.9. Los cinco tipos de segmentos que se generan cuando se alcanza la cota de error.	35
Figura 2.10. Algoritmo para la segmentación y etiquetado en tiempo real MCT.	38
Figura 2.11. Ruido de distribución normal de ancho de banda de 300 Hz. Frecuencia de muestreo 7500 Hz.	40
Figura 2.12. Primeras 100 muestras de la señal de la Fig.11 en donde se observa claramente el sobremuestreo.	40
Figura 2.13 Muestras esenciales para un error de interpolación de 0.01	40
Figura 2.14 Muestras esenciales para un error de interpolación de 0.01	41

Figura 2.15 Ejemplo de oscilación estable. Los puntos indican muestras esenciales. $clase()=\{g,g,g,e,e,e,f,f,f,d,d,g,g,e,e,f,d,g\}$.	43
Figura 2.16. Los dos patrones de supersegmentos estables.	44
Figura 2.17. Predicción del valor estacionario de la señal basada en los cambios de pendiente en los segmentos tipo “g”.	45
Figura 2.18. La tendencia es calculada empleando los máximos y mínimos locales y la posición temporal de ellos.	46
Figura 2.19. Diagrama de flujo para determinar la presencia de señales periódicas y estimar su frecuencia y fase empleando MCT iterativamente.	48
Figura 2.20. Señal periódica 50 Hz contaminada con ruido.	49
Figura 2.21. Gráficos de la evolución del cálculo de la frecuencia de la señal y la varianza de la posición de los máximos.	50
Figura 2.22. Señal periódica original, con ruido y reconstituida después de 50 iteraciones con error de interpolación $1 \text{ E-}05$.	52
Figura 2.23. Evolución de la varianza de la posición de los máximos para el caso de una señal aperiódica.	54
Figura 2.24. Las cuatro trayectorias simplificadas entre muestras esenciales.	56
Figura 2.25. Funciones de generación de la señal entre las muestras esenciales. Éstas se adaptan a la longitud de segmento y los valores inicial y final coinciden con las muestras izquierda y derecha.	57
Figura 2.26. En función de los valores de dos parámetros α y β se generan los segmentos que unen las muestras esenciales.	57
Figura 2.27. El muestreo MCT se inicia con tres muestras.	58
Figura 2.28. Dos muestras adicionales son adquiridas y se interpola y compara la tercera muestra.	59
Figura 2.29. Sigüientes dos muestras $x[6]$ y $x[7]$.	59
Figura 2.29. Doble control de interpolación para el caso de no poder contar con sobremuestreo de la señal.	61
Figura 3.1 Modulo del espectro de una $\text{sinc}(x)$ de 15 Hz de ancho de banda y los espectros resultantes con iteraciones en pasos de 10 hasta 1000 con error de interpolación tendiente a cero ($1.0 \text{ E-}08$).	65
Figura 3.2 Modulo del espectro de una $\text{sinc}(x)$ de 15 Hz de ancho de banda y los espectros resultantes con iteraciones en pasos de 1 hasta 10 con error de interpolación ($1.0 \text{ E-}02$).	65
Figura 3.3 Sismo de 6.9 Mw ocurrido el 24 de abril del 2017, cerca de la costa en la región central de Chile, registrado en el observatorio Las Campanas (LCO).	66
Figura 3.4 Primeros 10 segundos de la señal de la figura 3.3	67
Figura 3.5 Señal de entrada y salida al filtro MCT con 2500 iteraciones. Error de interpolación de $1.0\text{E-}04$.	67
Figura 3.6. Señal de vibración del acelerómetro para un rodamiento sin defecto.	68
Figura 3.7. Zoom de 30 milisegundos de la señal de un rodamiento sin defecto.	69
Figura 3.8. Señal de vibración para un rodamiento con un defecto de pista interno de 7 milésimas de pulgada. Frecuencia de muestreo de 48 Khz.	70

Figura 3.9 Zoom de un segmento de 50 milisegundos sobre la señal de la Fig. 3.8	70
Figura 3.10 Detección de envolvente empleando los máximos locales sobre la señal de la fig. 3.9	70
Figura 3.11. Puntos obtenidos para una falla de pista interna de 7, 14 y 21 mil, comparados con un rodamiento normal.	72
Figura 3.12. Puntos obtenidos para un tamaño de defecto de 14 mils para los distintos tipos de defecto junto a los puntos normales.	72
Figura 3.13. Forma ideal de la señal ECG.	75
Figura 3.14. Forma de onda real ECG. MIT-BIH Normal Sinus Rhythm Database record N° 16265, base de datos PhysioNet.	76
Fig. 3.15 Algoritmo MCT aplicado a una señal real ECG (426m.dat from MIT-BIH Malignant Ventricular Arrhythmia database, error de interpolación error=3, Fs=250 Hz.	77
Figura 3.16. Patrón candidato QRS.	78
Figura 3.17. Detección QRS (+) aplicada a la señal 426m.dat de la base de datos: MIT-BIH Malignant Ventricular Arrhythmia database, error de interpolación=3, Fs=250 Hz.	79
Figura 3.18. Zoom de la parte baja de la figura 3.17 para resaltar los patrones descartados.	79
Figura 3.19. Transición a un episodio VF, de la base de datos MIT-BIH Malignant Ventricular Arrhythmia database, error de interpolación=3, Fs=250 Hz, Record 426.	79
Figura 3.20 Frecuencia instantánea en latidos por minuto para la señal de la figura 3.19.	79
Figura 3.21. Algoritmo para la detección de VF (fibrilación Ventricular)	81
Figura 3.22 Superior: Captura de un móvil a 60 KPH. Inferior: Zoom de la parte creciente de la información de velocidad. Frecuencia de muestreo: 44100 Hz.	83
Figura 3.23. Superior señal de un móvil a 65 Km/h. Inferior envolvente empleando los máximos locales del proceso MCT según el algoritmo del punto 3.4.3.	85
Figura 3.24 Salida del algoritmo discriminador de sonido de móvil aplicado a la señal de la Fig. 3.23.	86
Figura 3.25 Señal resultante del producto de las señales de los dos algoritmos.	87
Figura 3.26 Señal de entrada y salida de un filtro de promedio móvil de 50 muestras. Nótese la disminución de la pendiente de la señal de salida.	88
Figura 3.27 Señal de entrada y salida de un filtro de mediana, ventana=50 muestras. Nótese la preservación de la pendiente.	88
Figura 3.28. Superior: Señal de salida del filtro mediana filtrada con el filtro MCT. Inferior: zoom de la pendiente de subida. Filtro MCT con reconstrucción por interpolación lineal, cantidad de iteraciones N=5, error=0.1%.	90
Figura 3.29. Pendiente hallada del perfil sonoro en función de la velocidad. La pendiente es calculada como el promedio de los segmentos tipo "d"	91

previos al máximo en el punto de inflexión “dg”.	
Figura 3.30. Captura de 40 segundos. En los primeros 10 segundos pasan dos móviles distintos. El resto de la señal es ruido provocado por el viento e interferencias propias de una ciudad.	92
Figura 4.1. Información multisensorial obtenida del procesamiento de video.	95
Figura 4.2a. Imagen de una ruta en 256 niveles de grises, 640x 480 píxeles.	98
Figura 4.2b. Máximos locales de la imagen de la Fig. 4.2a para un error de interpolación de 5/256.	98
Figura 4.2c. Máximos locales de la imagen de la Fig. 4.2a para un error de interpolación de 20/256.	99
Figura 4.3. Subimágenes de búsqueda derecha e izquierda. Se detectan líneas solo en estas zonas.	99
Fig. 4.4. Mascaras empleadas para el lado izquierdo y derecho de la imagen. En este caso $k=13$ direcciones de búsqueda para un ángulo de 90 grados.	100
Figura 4.5. Algoritmo de detección de líneas sobre una región particular. La máscara cambia según si es región derecha o izquierda.	100
Figura. 4.6. Superior: imagen sintética de prueba. Inferior izquierda: imagen con ruido tipo sal y pimienta con parámetro de intensidad de 0.6. Inferior derecha: zoom de una parte de una línea que resalta la dificultad de la detección.	103
Figura 4.7 1) Píxeles esenciales para segmentos clase “d”. 2) Segmentos clase “e”.3) Segmentos clase “f”. 4) Segmentos clase “g”. 5) Segmentos asociados a máximos. 6) Segmentos asociados a mínimos. Por último, en 7) La dirección principal encontrada en donde claramente se destaca la original.	104
Figura 4.8 Superior de izquierda a derecha: Imagen de una línea desmarcada. Imagen binaria mediante umbral simple. Segmentos clase e y la mejor línea estimada. Inferior de izquierda a derecha: Imagen con poco contraste sobre la línea tomada en un atardecer. Segmentación de máximos y la mejor línea estimada.	106
Figura 4.9 Superior izquierda, captura nocturna. Superior derecha, píxeles esenciales máximos. Inferior izquierda. Puntos alineados detectados superpuestos en la imagen. Inferior derecha, zoom de la detección y el trazado de la recta más importante.	107
Figura. 4.10. Superior imagen con sol de frente y los píxeles esenciales asociados a clases “e” y “f”. Inferior todas las rectas candidatas y la recta ganadora.	108
Figura 4.11 Interfase visual implementada para los ensayos.	108
Figura 4.12 Interfase visual implementada para los ensayos. En este caso se ubica al móvil en la mano contraria.	109
Figura 4.13 Espectro electromagnético. La banda infrarroja se divide en dos sub-bandas principales: Short Wave IR correspondiente a una longitud de onda de 2 a 5 μm y Long Wave IR desde 8 a 12 μm .	111
Figura 4.14. Valores aproximados de emisividad para distintos materiales.	112
Figura 4.15. Imagen térmica, obtenida con nuestro sistema de adquisición, de	113

un transformador de distribución de intemperie. El rectángulo más frío observado corresponde a la placa identificatoria metálica de baja emisividad. Se observa una distribución homogénea de temperaturas.	
Figura 4.16 Regiones vecinas centradas en el máximo absoluto de la imagen.	116
Figura 4.17 Imagen térmica del disipador bajo ensayo.	117
Figura 4.18 Evolución temporal del máximo MMax de la imagen disipador	117
Figura 4.19 Imagen térmica de un motor en el cual se puede medir la temperatura de los bobinados por las rejillas de ventilación. Capturada con el sistema de adquisición propuesto.	118
Figura 5.1 Secuencia de cálculo para la muestra interpolada empleando solo tres muestras de la señal	123
Figura 5.2 Algoritmo bajo el paradigma de máquina de estados para el cálculo en tiempo real de los segmentos.	124
Figura 5.3 representación de la señal como una matriz de comportamiento vs rango.	125
Figura 5.4 Matriz de probabilidad de transiciones obtenida para una señal de caudal.	126
Figura 5.5 Evaluación de AlfaMCT para los cuatro sismos de la tabla en la figura en función del número de iteraciones del numerador. Número de iteraciones de denominador fijado en cinco.	125
Figura 5.6 Diagrama en bloques propuesto para la detección. Se fusiona la información de N procesos, tomando como información de entrada la salida de los descriptores y la estimación del fondo.	128

Aportes de esta Tesis

Esta tesis se enmarcó en los proyectos de investigación “*Segmentación, análisis y detección de objetos y texturas en imágenes y secuencias de videos*” desarrollados en el Laboratorio de Procesamiento Digital de Imágenes del Instituto de Investigaciones Científicas y Tecnológicas en Electrónica (ICyTE), dependiente de la Facultad de Ingeniería de la Universidad Nacional de Mar del Plata y de CONICET y en el proyecto de investigación y desarrollo “*Desarrollo de sistemas y algoritmos para la extracción de conocimiento aplicado a sensores inteligentes*” de la Facultad Regional Del Neuquén, Universidad Tecnológica Nacional.

Esta tesis propone una forma distinta de representar la información embebida en una señal. La idea principal de la representación propuesta se basa en la unión de trayectorias simplificadas manteniendo la estructura intrínseca de la señal. Esta representación genera una estructura que minimiza la información redundante presente en la señal, facilitando los procesos de extracción de conocimiento. La redundancia, no solo enlentece los procesos subsiguientes, sino que además genera confusión, “ruido”, para el entendimiento deseado.

Dado que la señal es representada por una secuencia de segmentos de funciones, escaladas, dilatadas o contraídas que convergen a la señal, el comportamiento macroscópico de ella es rápidamente inferido observando la secuencia y las características de los segmentos. Para lograr aplicaciones reales de extracción de conocimiento es necesario procesar la señal y estos procesos, en general intensivos en computo, atentan en contra de las aplicaciones en tiempo real. Esta situación conflictiva es resuelta mediante la eliminación en tiempo real de muestras de la señal cuyo valor pueda ser inferido de su entorno cercano. En el marco de esta tesis llamaremos tiempo real cada vez que se obtenga información deseada de manera tal que no introduzca demoras significativas en la obtención de la función objetivo.

Los algoritmos propuestos en esta tesis logran este objetivo de una manera simple y eficaz. Simple porque a partir de la observación de como la señal se aparta de un comportamiento lineal se infieren las trayectorias simplificadas, empleando

operaciones lógicas y aritméticas. La redundancia se elimina en el proceso de adquisición de la señal.

Es importante puntualizar que el objetivo de la tesis no es comprimir la señal, sino ~~que~~ conservar la estructura informativa de la señal minimizando la redundancia para facilitar los procesos de inferencia de conocimiento.

La estructura propuesta para la representación de la señal ha demostrado una eficiente inferencia de conocimiento como se puede comprobar en las aplicaciones abordadas, las cuales han sido y están siendo implementadas en sistemas embebidos para lograr aplicaciones reales.

Todos los aportes de esta tesis se sustentan en los siguientes trabajos:

1. “***Sensor Signal Preprocessing Techniques for Analysis and Prediction. Proceedings***”. Gustavo. Monte. IECON 2008. 34th Annual Conference of the IEEE Industrial Electronics Society: IEEE. 2008 vol.34 n°1. p1788 - 1793. ISSN 1553-572X.
2. “***Identificación de Señales de Sensores Industriales Mediante Secuencias de Patrones Elementales***”. Gustavo Monte, Pablo Liscovsky, Hector Kessel, Damian Marasco, Alejandro Hossian. XIII Reunión de Trabajo en Procesamiento de la Información y Control”. Rosario País: Argentina. Año 2009.
3. “***A Proposal of a New Standard for Sensor Signal Analysis***”. Gustavo Monte IEEE- ICIT 2010. International Conference of Industrial Technology. ISBN 978-1-4244-5697-0. Valparaiso. Chile. 2010.
4. “***Muestreo y Adquisición Inteligente de Señales Sensoriales en Sistemas Embebidos***”. Gustavo Monte, Héctor Kessel, Norberto Scarone, Cesar Almendra. CASE 2011. Congreso Argentino de Sistemas Embebidos. UTN Regional Buenos Aires. Buenos Aires. Argentina. 2011.

5. “*A novel time-domain signal processing algorithm for real time ventricular fibrillation detection*”. G. Monte, N. Scarone, P. Liscovsky. Journal of Physics: Conference Series Volume 332. J. Phys.: Conf. Ser. 332 012015 doi:10.1088/1742-6596/332/1/012015. 2011.
6. “*Detección Robusta de Líneas en Video Orientada a la Navegación Robótica*”. Gustavo Monte, Norberto Scarone, Damian Marasco, Walter Lagos, Fernando Pereira. IEEE -ARGENCON 2012. Córdoba. Argentina. 2012.
7. “*Standard of Things, First Step: Understanding and Normalizing Sensor Signals*”. Gustavo Monte, Victor Huang, Pablo Liscovsky, Damian Marasco, Ariel Agnello. IECON 2013. IEEE Industrial Electronics Society, 39th Annual Conference. ISBN: 978-1-4799-0223-1. Viena, Austria. 2013.
8. “*Robust Line Detection in Video Oriented to Robotic Navigation*”. G. Monte; N. Scarone; D. Marasco; W. Lagos; F. Pereyra. IEEE Latin America Transactions Electrical Electronics Engineers. vol.11 n°1. p609 - 615. ISSN 1548-0992. 2013.
9. “*Sistema de Adquisición y Procesamiento Inteligente de Señales Biológicas*”. Norberto Scarone, Damián Marasco, Nicolás Castro, Gustavo Monte. IV Congreso de Microelectrónica Aplicada. Bahía Blanca. Argentina. 2013.
10. “*Normalizing Transducer Signals: An Overview of a Proposed Standard*”. Gustavo Monte, Victor Huang, Antonio Pietrosanto, Vincenzo Paciello, Zheng Liu, Franceso Abate. I2MCT- 2014. IEEE International Instrumentation and Measurement Technology Conference. ISBN: 978-1-4673-6386-0. Montevideo. Uruguay. 2014.
11. “*Muestreo Integral de Señales Sensoriales*”. Gustavo Monte, Juan Pastore, Virginia Ballarin, Norberto Scarone. Biennial Congress of Argentina ARGENCON-2014. ISBN: 978-1-4799-4269-5. CD-ROM ISBN: 978-1-4799-4270-1. Print on Demand (PoD) ISBN: 978-1-4799-4268-8. Bariloche. Argentina. 2014.
12. “*A Comparison Between Sensor Signal Preprocessing Techniques*”. F. Abate; V. Huang; G. Monte; V. Paciello. IEEE SENSORS JOURNAL. vol n°99. ISSN 1530-437X. 2015.

13. “*Using Algorithms on Smart Transducer: An IEEE Standard Perspective*”. Z. Liu; Banakhr F; G. Monte; V. Huang. IEEE - Sensors Journal: IEEE-Inst Electrical Electronics Engineers. vol n°99. ISSN 1530-437X. 2015.
14. “*Real time transducer signal features extraction: A standard approach*”. Monte, G; Abate, F; Huang, V; Paciello V; Pietrosanto A. Industrial Informatics (INDIN). <http://dx.doi.org/10.1109/INDIN.2015.7281740>. Electronic ISBN: 978-1-4799-6649-3. Print ISSN: 1935-4576. Electronic ISSN: 2378-363XA. 2015.
15. “*Estimación de la velocidad de un móvil mediante el procesamiento del sonido pasante*”. Gustavo Monte Norberto Scarone, Damián Marasco, Ariel Agnello. VI Congreso de Microelectrónica Aplicada. San Justo, Argentina. 2015.
16. “*Preliminary analysis of a Real Time Segmentation and Labeling algorithm*”. Abate F, V; Paciello V; Pietrosanto A; Monte G. Environmental, Energy and Structural Monitoring Systems (EESMS). Trento, Italia. Electronic ISBN: 978-1-4799-8215-8. CD-ROM ISBN: 978-1-4799-8214-1. Print on Demand (PoD) ISBN: 978-1-4799-8216-5. 2015.
17. “*Uncertainty of a real time segmentation and labeling algorithm in signal period measurement*”. F. Abate, A. Pietrosanto, V. Paciello, G. Monte. IEEE 25th International Symposium on Industrial Electronics (ISIE). Electronic ISBN: 978-1-5090-0873-5. USB ISBN: 978-1-5090-0872-8. Electronic ISSN: 2163-5145. San Jose, California, USA. 2016.
18. “*ISO/IEC/IEEE P21451-001 standard for signal treatment of sensory data*”. Zheng Liu; Gustavo Monte; Victor Huang. IEEE 25th International Symposium on Industrial Electronics (ISIE). Electronic ISBN: 978-1-5090-0873-5. USB ISBN: 978-1-5090-0872-8. Electronic ISSN: 2163-5145. San Jose, California, USA. 2016.
19. “*Smart Sensing of Systems Thermal Behavior Using Low-Cost Infrared Cameras*”. Gustavo Monte; Damián Marasco; Lucas Solorzano; Javier Vallejos; Ruben Bufanio. 45th Annual Conference of the IEEE Industrial Electronics Society. The IECON 2019. Lisboa, Portugal 14-17. 2019.

20. “*Smart Sensor Efficient Signal Processing for Earthquake Early Detection*”. Francesco Abate, Vincenzo Paciello, Gustavo Monte António Espírito Santo. IEEE M&N 2019, doi: 10.1109/IWMN.2019.8805009. Catania, Italia. 2019.
21. “*Validación de Señales Sensoriales en Sistemas de Instrumentación para la Toma de Decisiones en Procesos Críticos*”. Gustavo Monte, Damián Marasco, Ruben Bufanio, Pablo Liscovsky. 1er Congreso de seguridad de procesos en la industria de oil & gas. Neuquen. Argentina. 2019.
22. “*Edge Sensor Signal Processing Algorithms for Earthquake Early Detection*”. M. A. Ugwiri, M. Carratú, G. Monte, A. Espírito-Santo and V. Paciello, Dubrovnik, Croatia. IEEE - International Instrumentation and Measurement Technology Conference (I2MTC). pp. 1-6, doi: 10.1109/I2MTC43012.2020.9129617. 2020.

A partir del trabajo de investigación desarrollado durante la presente tesis se lideró un grupo mundial de desarrollo del estándar “*IEEE Recommended Practice for Signal Treatment Applied to Smart Transducers*” in IEEE Std 21451-001-2017, vol., no., pp.1-109, 20 Dec. 2017, doi: 10.1109/IEEESTD.2017.8227039, que se encuentra aprobada desde diciembre del 2017.

1 INTRODUCCION

RESUMEN

La extracción de la información y conocimiento presente en una señal es y seguirá siendo una temática de interés para la comunidad científica. La representación de una señal, muestreo, para su procesamiento en sistemas de cómputo no es adecuada para el entendimiento y no facilita los procesos de inferencia de conocimiento. Básicamente por una razón: se desconoce que muestras transportan información y cuales son redundantes. En este capítulo se presenta el estado del arte y se justifica la propuesta de esta tesis como una contribución al entendimiento de señales, sobre todo para inferencias en tiempo real.

La extracción de las características ideales que reflejen el contenido intrínseco de una imagen es aún un desafío en el campo del procesamiento de imágenes [Dong ping Tian, 2013]. Si bien durante los últimos 60 años se vienen realizando investigaciones para la interpretación de la información presente en una escena, sólo en los últimos años se han logrado avances importantes. Recientes investigaciones en aprendizaje profundo (*Deep Learning*) empleando redes convolucionales, han logrado un avance significativo en el entendimiento de una escena compleja [Jianlong fu, Yong Rui, 2017], [Willett, R. 2017]. Para lograr este discernimiento es necesario realizar un entrenamiento con cientos de miles de imágenes, imitando la fuente de información de un ser humano en los primeros años de su vida. No se conoce exactamente como el cerebro representa y procesa la información, pero sí que recibe un estímulo visual constante a lo largo de los primeros años de vida, adquiriendo y procesando cientos de miles de imágenes que es la fuente de información primaria que posibilita el entendimiento. Otra particularidad es que seguramente en nuestro cerebro, las señales se almacenan y procesan de manera muy distinta en comparación como lo hacen las computadoras. Las computadoras que el hombre ha creado son muy eficientes para procesar datos,

por lo tanto, las señales del mundo real son convertidas en una secuencia ordenada de bits, en señales unidimensionales mediante muestreo uniforme y en imágenes muestreadas sobre una grilla rectangular, para que puedan ser procesadas. Transformar una señal, de manera tal que pueda ser procesada por una computadora, entorpece la extracción de información y conocimiento de una escena.

En señales unidimensionales el problema es sumamente más acotado y se han desarrollado numerosas técnicas que posibilitan el entendimiento de la información transportada en ella. Estas herramientas incluyen cambios de dominios de la variable independiente como, por ejemplo, análisis espectral, wavelets, transformadas especiales (Hilbert, Gabor, Karhunen–Loève), procesos no lineales como redes neuronales, lógica difusa y operaciones morfológicas. Cada una de ellas presenta fortalezas y debilidades en algún aspecto de la extracción de la información [Lyons Richard, 2011], [Mertins Alfred, 1999], [Quinquis André, 2008].

En la gran mayoría de las aplicaciones reales, las señales son digitalizadas mediante muestreo uniforme. Este muestreo es el más utilizado por dos razones: en primer lugar, es simple ya que captura la señal en un instante determinado, sin importar el pasado o el futuro, y genera un valor correspondiente a un alfabeto finito que representa al valor de la señal. Segundo, la variable independiente es representada como el índice de la muestra y de esta forma encuentra una correspondencia directa hacia el dominio digital.

También existen otros tipos de muestreo como son el muestreo no uniforme [Marvasti F,2000], por cruce de nivel [Karen Guan et al., 2008] y recientemente muestreo compresivo [Candès E.,l Wakin M. 2008]. Este último, se basa en la transformación de la señal en otros dominios donde esta señal es representada mediante una menor cantidad de muestras adquiridas con muestreo aleatorio. La propuesta de esta tesis se encuadra en el muestreo compresivo y obtiene información de la resultante de eliminar parte de la redundancia como se explica en el siguiente punto.

1.1 Redundancia

Todas las señales poseen algún grado de redundancia. La comunicación entre sistemas, incluyendo seres humanos, no podría haberse desarrollado sin ella. La redundancia es la forma que tiene la naturaleza de lograr una efectiva comunicación y posibilitar el aprendizaje. Por ejemplo, la transmisión oral, escrita y visual entre seres humanos está repleta de redundancia. Todo mensaje entre un emisor y un receptor posee redundancia y genera robustez ante la pérdida de alguna parte del mensaje.

La redundancia presente en una fuente de información es lo que posibilita que nos equivoquemos y poder aprender nuevos conceptos. En otras palabras, la redundancia genera grados de libertad para la interpretación y es necesaria para detectar los errores en la extracción de características o patrones sobre la información de una señal.

No existe comunicación sin redundancia porque no existe canal sin algún tipo de distorsión. En este fenómeno de selección natural, lo exitoso ha sido la comunicación redundante. Entonces, toda señal que tomemos del mundo real tendrá una fuerte componente redundante. Los procedimientos presentados en esta tesis se basan en analizar eficazmente la redundancia y minimizarla. La gran diferencia con los métodos existentes de compresión de datos es que se mantiene la estructura de información de la señal. Al hacerlo, la componente de información queda al descubierto y facilita la extracción de conocimiento como se describe a continuación.

1.2 Datos, información y conocimiento

Las computadoras son procesadoras de datos por excelencia, así las hemos creado. Se ha avanzado exponencialmente en la capacidad de memoria y en la velocidad de procesamiento, pero conceptualmente éstas realizan las mismas operaciones desde su creación hace décadas. Para adaptar las imágenes y señales al modo de

procesamiento elemental que posee una computadora, se han creado unidades de información acordes. Los datos en una imagen son pixeles mientras que en una señal unidimensional son muestras cuantificadas. Una señal genera datos que conlleva información de la cual se puede extraer conocimiento. Estas tres estructuras básicas, datos, información y conocimiento se encuentran embebidas en la señal.

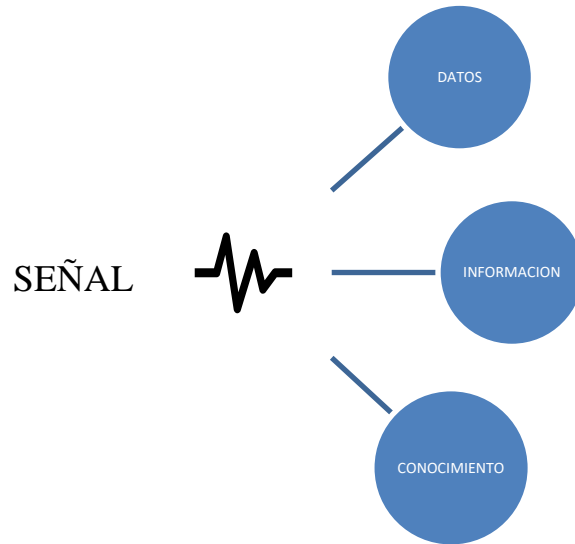


Figura. 1.1. La señal como fuente de datos, información y conocimiento.

Los datos son la fuente primaria de generación de una señal y se encuentran adaptados para ser procesados en los sistemas digitales que el hombre ha creado. Los datos son perfectamente cuantificables y los sistemas digitales procesan y almacenan datos en forma sumamente eficiente.

No existe un procesamiento único para extraer conocimiento y es debido principalmente a dos motivos: primero, a la naturaleza subjetiva del conocimiento y segundo a la innumerable cantidad de funciones objetivo. *En cambio, la redundancia es común a todas las formas de comunicación.*

El enfoque clásico es medir la cantidad de información por la incertidumbre de los símbolos de la fuente. Como la incertidumbre es cuantificable en base a la probabilidad de ocurrencia de los símbolos, entonces la información se puede medir por su entropía. Si no se liga el concepto de información a ninguna forma de extracción de patrones o características y se lo relaciona con la minimización de redundancia, se puede cuantificar la información. Este enfoque brinda un sentido más amplio ya que la redundancia es común a todas las formas de la información. Si se minimiza la redundancia, los datos presentan un porcentaje mayor de información, entendiendo a la información como: “*el significado dado a los datos por la forma en que se interpretan*” [<https://www.thefreedictionary.com/information>].

¿Qué es lo que sirve de esos datos? La respuesta a esta pregunta conlleva a la siguiente pregunta, ¿para qué? Naturalmente surge el concepto de **conocimiento** o entendimiento y no es posible cuantificarlo en forma sencilla. Podríamos definir conocimiento como “información procesada para un objetivo específico”. Por ejemplo, algoritmos para encontrar líneas o formas circulares en una imagen. Como el conocimiento es subjetivo y con múltiples aspectos, procesar una señal para obtenerlo requiere de algoritmos específicos y dedicados (ej. reconocer caras, formas humanas, etc.), por lo tanto, no existe un procesamiento universal para la extracción de conocimiento.

El muestreo de una imagen empleando una grilla rectangular y el muestreo uniforme en señales unidimensionales, no proporcionan una estructura eficiente para la extracción de información. Es una estructura de datos que manejan eficientemente las computadoras, pero no necesariamente es la mejor forma de representar la información. Lo único que garantiza, si se respetan condiciones de muestreo, es que esos datos contienen toda la información presente en la fuente.

La mayoría de los algoritmos desarrollados parten de los datos, píxeles, para extraer conocimiento. Se parte de píxeles hacia la detección del patrón esperado. Este salto necesita una gran cantidad de procesamiento, sobre todo si es necesaria robustez en la detección. Este esquema de procesamiento se conoce como *Botton-up*. Por otro

lado, el procesamiento *Top-down* requiere disponer de un conocimiento previo del o los objetos que se buscan en la imagen. Un ejemplo claro de esta metodología es la técnica de contornos activos o “*Snakes*” [Kass, M., et al. ,1988], técnica que cuantifica cuanto tuvo que deformar un contorno para poder compararlo con uno conocido.

Dada la ineficiencia de la representación de una señal o imagen en una computadora, encontramos en la literatura incontables algoritmos que, con el objetivo de disminuir el tiempo de procesamiento, logran una representación de la información diferente mediante el preprocesamiento de los datos. Un ejemplo clásico es el concepto de imagen integral introducido en [Viola, P, Jones, M. 2001], que genera una representación intermedia, sin un fin específico, que acelera los procesos posteriores.

Un enfoque muy atractivo es partir de datos no redundantes. La redundancia entorpece los procesos de entendimiento o extracción de conocimiento. Volviendo al ejemplo de la comunicación desde el punto de vista de los seres humanos, nuestro cerebro ha aprendido a través de miles de años a minimizar la necesaria redundancia de las señales, para inferir conocimientos a partir de información y no de datos. El problema de partir de datos no redundantes es que, al comprimir los datos con las técnicas conocidas, se destruye la estructura de la información.

Esta tesis propone una forma distinta de representar la información embebida en una señal basada en la unión de trayectorias simplificadas en lugar de muestras a la vez que mantiene la estructura intrínseca de la señal. Esta estructura minimiza la redundancia presente en la señal, por lo tanto, facilita los procesos de extracción de conocimiento. La redundancia, no solo entorpece los procesos, sino que además genera confusión, “ruido”, para la interpretación del conocimiento buscado. Como en esta propuesta la señal es representada por una secuencia de segmentos de funciones, escaladas, dilatadas o contraídas que ajustan a la señal, el comportamiento macroscópico de ella es rápidamente inferido observando la secuencia y las características de estos segmentos. La paradoja es que se requiere procesar la señal para determinar qué es redundancia. Esta situación conflictiva es resuelta mediante el descarte en tiempo real de muestras de la señal cuyo valor pueda ser inferido de su

entorno cercano. Una ventaja adicional del sistema de representación propuesto es que relaciona la información que se encuentra en un entorno a través de la forma de las trayectorias. La redundancia eliminada es la que se genera en la adquisición de la señal, pudiendo existir patrones redundantes que son detectados por la secuencia de trayectorias.

La información presente en una señal digital se puede inferir de la forma en la cual se relacionan las muestras. El valor de la muestra, único valor que proporciona el muestreo uniforme, es solo una pequeña parte de la información de la señal.

Durante el transcurso del desarrollo de esta tesis, se propuso esta metodología como una norma para sensores inteligentes. Se consolidó la propuesta y dirigiendo un grupo mundial de desarrollo se aprobó la norma IEEE 21451-001-2017 *“Recommended Practice for Signal Treatment Applied to Smart Transducers”* en diciembre del 2017. En esta norma se propone la estructura de la información sensorial propuesta en esta tesis para el tratamiento de la señal sensorial en el punto de adquisición.

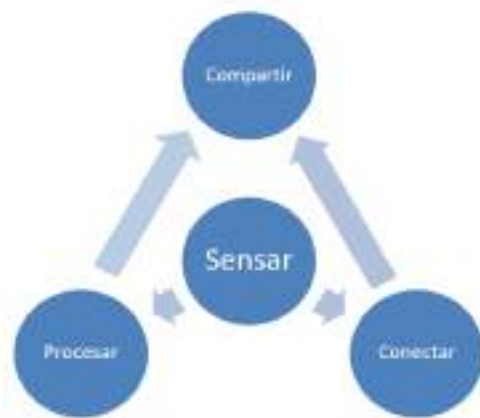


Figura 1.2. Los cuatro pilares de la IoT

Toda la información que obtenemos del mundo real proviene de los sensores. Por lo tanto, los resultados de la tesis impactan en forma directa en la sociedad ya que tratamos con la señal independientemente del tipo de sensor. El procesamiento busca

obtener sistemas más seguros y eficientes en donde el sensor es realmente inteligente porque se da cuenta de lo que está observando y midiendo.

El mundo se encuentra totalmente interconectado, y esta conexión ha alcanzado a todas las cosas. La IoT (*Internet Of Things*) está revolucionando, no solo la forma de hacer negocios, sino nuestra interacción con el mundo. La mayoría de esas cosas poseen sensores y surgen nuevas posibilidades de tratamiento de señales sensoriales. La IoT se encuentra definida por cuatro acciones, como vemos en la Fig. 1.2:

- SENSAR
- PROCESAR
- CONECTAR
- COMPARTIR

El simple hecho de intercambiar dos acciones, por ejemplo:

SENSAR → CONECTAR → PROCESAR o
SENSAR → PROCESAR → CONECTAR

cambia totalmente los paradigmas de la IoT. Surgen conceptos como *Edge computing* [Weisong Shi et al ,2016], *Fog computing* [Songqing Chen et al., 2017] y *Cloud computing* [Mohsin Nazir, 2012] los cuales generan topologías completamente diferentes.

La acción más compleja de lograr es la de COMPARTIR y surge naturalmente la pregunta ¿Que se debe compartir? Está claro que debe ser información o conocimiento. Esta tesis fundamenta que se debe sensar, procesar para extraer información y conocimiento en el punto de adquisición para luego conectar y compartir. La estructura de la señal sensorial propuesta permite compartir conocimiento (información específica para el cual el sensor fue diseñado) y además permite compartir información que puede ser de interés para otro ente o disponible para una minería de datos.

2 MUESTREO COMPLETO DE SEÑALES SENSORIALES UNIDIMENSIONALES

RESUMEN

La conversión de una señal analógica en discreta es un proceso fundamental en los sistemas actuales. La señal por digitalizar es un proceso aleatorio, lo único que se conoce de antemano en la mayoría de los casos, es su limitación en ancho de banda, que le impide tomar valores totalmente arbitrarios entre muestras. Al aumentar la frecuencia de muestreo se restringe la aleatoriedad hasta quedar reducida a un espacio de escasas dimensiones que permite realizar un análisis preciso sobre la señal. Aprovechando la capacidad actual de los sistemas embebidos, en esta tesis se presentan técnicas y algoritmos con el objetivo de sumar inteligencia al proceso de muestreo de una señal analógica. Empleando algoritmos de sobremuestreo e interpolación adaptativa se logra analizar la señal, inferir estados, predecir comportamientos y validar la señal digitalizada en el momento de la adquisición.

2.1 Introducción

Una señal es un símbolo, un gesto u otro tipo de signo que informa o avisa de algún evento. Ejemplo de señales podrían ser las luces de un semáforo, los gestos de una persona, etc. Dentro de un contexto científico, una señal describe fenómenos físicos como una función matemática con respecto a variables independientes. El tiempo es la variable independiente que más frecuentemente emplearemos, pero no es la única posible. Por ejemplo, si tomamos la presión atmosférica en función de la altitud de un avión, la variable independiente es la altitud y la presión experimenta valores en función de ella. Los sistemas basan sus decisiones en función de las señales de los sensores incluidos en él. Las señales provenientes de los sensores son la fuente de información disponible para cumplir el objetivo de diseño de un

sistema de instrumentación y/o control. Estas señales proporcionan la abstracción del mundo real. Más “valiosa” es la señal cuanto más información se encuentra embebida en ella. El concepto clave es la “información” que recibe el sistema. La Fig. 2.1 representa este concepto en forma gráfica. Esta información se encuentra inmersa en la señal, por lo tanto, generalmente es necesario procesarla. Procesar una señal significa: realizar una operación matemática sobre ella con el fin de transformarla o de extraer información embebida.

Aún en señales analógicas, no es necesario conocer el valor de la señal todo el tiempo. La máxima velocidad de cambio de una señal determina el intervalo de tiempo mínimo que se necesita conocer a la señal para poder inferir su comportamiento dentro él.

Una señal es una entidad cuantificable de alguna forma de información.

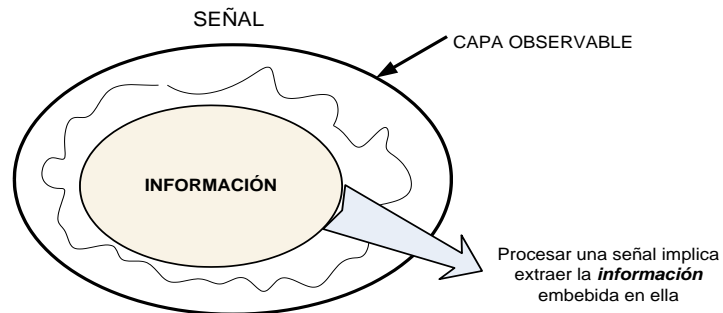


Figura 2.1. La información que transporta una señal generalmente se encuentra oculta. Procesar una señal implica extraer la información embebida en el núcleo de ella.

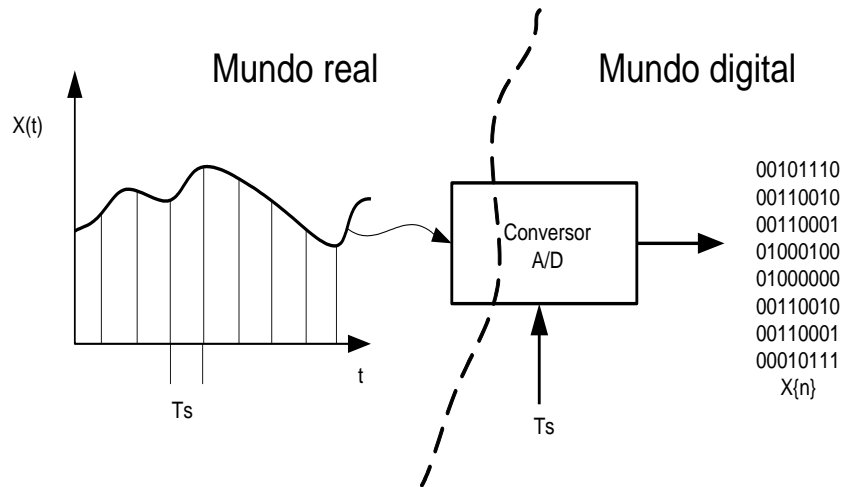


Figura 2.2. El convertor analógico digital es el elemento interfase híbrido (entrada analógica- salida digital) que permite la representación de las señales reales en el mundo digital. En este caso, muestreo uniforme, que transforma la señal en una secuencia ordenada de números.

Si la señal en el mundo real es analógica, debe ser convertida en una representación digital que pueda ser entendida por el sistema digital.

La conversión de una señal en digital es realizada por el convertor analógico – digital (A/D) que es el elemento principal de interfase entre el mundo real y el “mundo digital” como se observa en la Fig. 2.2. El convertor A/D es un sistema híbrido ya que posee como entrada una señal continua y entrega a su salida una señal discreta. La forma más simple, pero no la única, de conversión es el muestreo uniforme [Oppenheim, 2013]. Otros tipos de muestreo incluyen muestreo por cruce de nivel [Sayiner N., Viswanathan T.R, 1996] y muestreo no uniforme [Marvasti F, 2000] y más recientemente muestreo compresivo [Candès E J. Wakin M. B,2008].

En el caso de muestreo uniforme, a intervalos regulares de T_s segundos, se toman muestras de la señal y se convierten en un código correspondiente a un alfabeto finito, es decir una cantidad limitada de representaciones posibles. También el esquema más simple es el de dividir el rango dinámico del sensor en M divisiones de igual longitud, proceso que se conoce como cuantificación uniforme. Otros tipos de cuantificación se adaptan a la función densidad de probabilidad de la señal esperada. Es decir, para los valores más probables de la señal se disminuye la amplitud de la cuantificación, ver por ejemplo un trabajo clásico en [Max J.,1960].

El pasaje del mundo real al digital tiene dos particularidades importantes. Primero, la señal que existe para todo tiempo es considerada solamente cada T_s segundos. Segundo la señal analógica, que es libre de tomar cualquier valor de amplitud, es forzada a una representación finita en el mundo digital, transformándose en una secuencia ordenada de números. El primer aspecto fue abordado por H. Nyquist [Nyquist H.,1928] y se conoce ampliamente como el teorema de muestreo. El teorema establece que no es necesario conocer la señal todo el tiempo para obtener una representación digital fidedigna siempre y cuando la señal posea un límite máximo de contenido espectral. El teorema determina que debemos tomar muestras a una frecuencia de por lo menos el doble de la máxima frecuencia presente en la señal para señales no repetitivas. Es un límite teórico que necesita filtros reconstructores ideales para obtener sin error el valor de la señal entre las muestras. Como resultado del proceso de muestreo uniforme se obtiene una secuencia ordenada de números que representan la señal, siempre y cuando se haya respetado el teorema de muestreo para evitar el solapamiento espectral en la señal muestreada, conocido comúnmente como “*aliasing*”

Hasta aquí se ha puntualizado un proceso convencional de digitalización. La secuencia ordenada de números es la señal en el mundo digital a partir de la cual se deberán tomar todas las decisiones en el sistema.

Un sensor inteligente con capacidad de procesamiento embebida debería procesar el vector de muestras digitales con el fin de tomar conciencia de la señal que obtuvo.

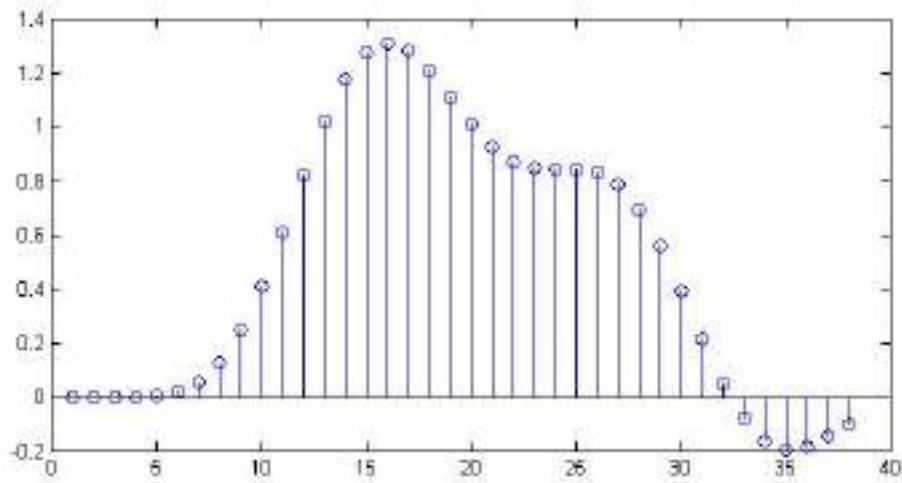


Figura 2.3. El muestreo inteligente debe “ver” a la señal completa, no solo una muestra a digitalizar.

Recordemos que el proceso de muestreo es el único punto de conexión a través del cual se infiere el mundo real. En un sistema de control, todas las acciones estarán basadas en esta información, por lo tanto, es un proceso vital. En la Fig. 2.3 se observa una señal que ha sido muestreada con muestreo uniforme a una frecuencia mayor que la cota mínima de Nyquist.

Un proceso de muestreo inteligente implica ver a la señal en su conjunto, no solo una muestra. *El muestreo uniforme solo nos brinda la secuencia ordenada de números, en donde la variable “tiempo” queda implícita en el orden de la muestra.*

Si un ser humano observa la Fig. 2.3 podría sacar conclusiones tales como:

- La señal es suave.
- Comienza a incrementarse desde cero hasta alcanzar un máximo.
- Luego disminuye, se estabiliza y vuelve a caer hasta alcanzar un mínimo.

Estas conclusiones las podemos realizar porque vemos a la señal en su conjunto, no solo muestras. En todas estas conclusiones, lo que el observador ha hecho mentalmente fue ir comparando muestras entre sí y tener una “visión” completa de la porción de la señal. Ante esto surge la siguiente pregunta ¿no sería más

provechoso que el proceso de muestreo nos brinde una plataforma para inferir en forma eficiente la descripción de la señal?

Este es el principal objetivo de la investigación presentada en esta tesis: ***proponer un proceso de muestreo que capture la relación entre muestras, y a partir de esta relación, inferir comportamientos tal cual lo haría un observador mirando la señal en su conjunto.***

Otra observación importante que podemos hacer, desde el punto de vista del observador, sobre la Fig. 2.3 es que la cantidad de muestras es excesiva. ¿Cómo nos damos cuenta? Porque el cambio entre muestra y muestra es coherente, pequeño y permite inferir la trayectoria de la señal sin ambigüedades y con poco esfuerzo computacional, por ejemplo, una línea entre muestra y muestra es suficiente para conocer la señal entre ellas.

La señal proveniente del sensor genera un proceso aleatorio. Los parámetros que se conocen previamente son la limitación en ancho de banda, impuesta por un filtro *antialiasing*, y su rango dinámico. En general, no se conoce la función densidad de probabilidad de la señal. Las características deseables de un sensor inteligente con respecto a la señal que adquiere serían:

- Determinar la frecuencia de muestreo óptimo.
- Reconocer la forma de la señal.
- Predecir valores.
- Validar la señal.
- Marcar particularidades.
- Almacenar valores pasados en forma comprimida.
- Detectar patrones en forma automática y/o predeterminada.

El valor de la muestra en un instante es solo una parte de la información presente en la señal y si no se comete *aliasing* es una representación fidedigna de la señal analógica original. Sin embargo, la información presente en una señal está embebida en cómo se relacionan muestras en diferentes instantes de tiempo. La

Fig. 2.4 sugiere una representación más completa de la señal digital, en donde se conozca desde el proceso de muestreo la relación entre muestras.

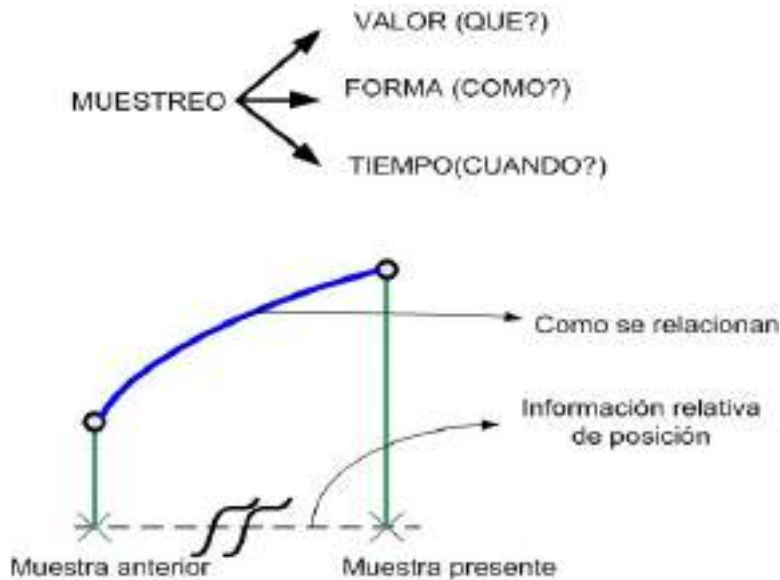


Figura 2.4. El muestreo completo debe proporcionar además del valor de la muestra información de la relación entre ellas y el instante de ocurrencia.

El muestreo uniforme proporciona sólo el valor de la señal en un instante particular. Más información explícita es necesaria, especialmente como se relacionan las muestras entre sí para conformar una plataforma de inferencia de conocimiento.

La idea principal es representar la señal como una concatenación de funciones conocidas en lugar de secuencia de muestras. En una condición de sobremuestreo, el valor de la muestra pierde importancia relativa mientras que la comparación de su valor con muestras pasadas permite inferir comportamientos de trayectoria. Bajo esta condición, las trayectorias de la señal son suaves y se propone a la señal como originada por una función generatriz la cual se contrae, dilata y adapta para seguir la trayectoria de la señal.

En las secciones siguientes se desarrollan los algoritmos propuestos en busca de las características anteriores. Se mantiene el muestreo uniforme pero la señal se sobremuestra con respecto a la frecuencia de Nyquist. Se obtiene un vector redundante a partir del cual se disparan todos los algoritmos. El primero es una

segmentación en tiempo real basada en interpolación lineal adaptiva. Luego los segmentos son etiquetados reflejando el comportamiento dentro de él. Con esta información se detectan secuencias de segmentos que identifican comportamientos globales. Estos tres procesos conforman la base para el análisis de la señal muestreada y se propone como un muestreo inteligente o completo.

2.2 Descripción de los algoritmos

De acuerdo con el teorema de muestreo, debemos tomar dos muestras, como mínimo, en el tiempo correspondiente al periodo de la componente de frecuencia máxima de la señal, asumiendo periodicidad. Si comenzamos a aumentar la frecuencia de muestreo, surgen dos preguntas: ¿Hasta qué valor tiene sentido aumentarla? y ¿Qué beneficios brinda este incremento? Las respuestas a estos interrogantes surgen del siguiente análisis para el caso particular de una señal senoidal.

Sea $x(t)$ la señal a ser muestreada, limitada en banda a f_{max} Hertz. Consideremos el caso particular en que $x(t)$ es una señal sinusoidal de máxima amplitud y frecuencia. El muestreo es uniforme y se divide el rango dinámico $2A$ en 2^N divisiones, donde N es la cantidad de bits del conversor A/D.

$$x(t) = A \text{sen}(2\pi f_{max} t) \quad (1)$$

Donde A es la mitad de rango dinámico del conversor. Si transcurren Δt segundos tenemos:

$$x(t + \Delta t) = \text{sen}(2\pi f_{max}(t + \Delta t)) \quad (2)$$

Por lo tanto:

$$x(t + \Delta t) - x(t) = A [\text{sen}(2\pi f_{max}(t + \Delta t)) - \text{sen}(2\pi f_{max}(t))] \quad (3)$$

El miembro izquierdo de (3) corresponde al cambio de la señal entre dos instantes de muestreo. Si fijamos este cambio a L bits, obtenemos:

$$\frac{2A}{2^N} L = A [\text{sen}(2\pi f_{\max}(t + \Delta t)) - \text{sen}(2\pi f_{\max}(t))] \quad (4)$$

La máxima velocidad de cambio de la sinusoidal ocurre en los cruces por cero. Tomando $t=0$ para simplificar obtenemos de (4):

$$2^{-N+1} L = \text{sen}(2\pi f_{\max} \Delta t) \quad (5)$$

Dado que $1/\Delta t$ es la frecuencia de muestreo f_s , el argumento del seno es mucho menor que uno para una condición de sobremuestreo y teniendo en cuenta que $\text{sen}(x) \approx x$ para x pequeños obtenemos:

$$\frac{f_s}{f_{\max}} = \frac{\pi 2^N}{L} \quad (6)$$

Por ejemplo, empleando la ecuación 6, para $N=8$ bits obtenemos una frecuencia de muestreo de aproximadamente 400 veces la de Nyquist para que solo cambie un bit menos significativo entre muestra y muestra.

Dada una resolución del conversor, la ecuación 6 determina la máxima frecuencia de muestreo que tiene sentido aplicar ya que valores mayores darían como resultado la misma muestra. La ecuación 6 se determinó para una señal sinusoidal pura de máxima amplitud y frecuencia y en la mayor velocidad de cambio de la señal. Considerando una señal real y teniendo en cuenta que el contenido espectral es en general decreciente con la frecuencia, una fracción del máximo obtenido es razonable.

Entre los beneficios que brinda el sobremuestreo se encuentra una reducción efectiva del ruido de cuantificación [Madapura J M,2008]. Es importante resaltar que la señal que podía tomar infinitos valores entre dos puntos en el mundo analógico es forzada a un alfabeto finito. *Más aun, empleando la restricción del cambio a solo un LSB, el espacio queda reducido a tres dimensiones, el mismo valor o más menos un LSB.*

Bajo condiciones de sobremuestreo se reduce la aleatoriedad de la señal. Visto de otra manera se genera redundancia entre muestras vecinas. Los algoritmos que describen el muestreo inteligente parten de los siguientes conceptos:

- *No todas las muestras poseen la misma cantidad de información [Marten D. van der Laan,1995].*
- *En condiciones de sobremuestreo, las muestras pierden valor relativo si su valor puede ser obtenido mediante interpolación de sus muestras vecinas.*
- *Las muestras que no pueden ser obtenidas mediante una combinación lineal de sus vecinas, transportan en algún sentido más información y las llamaremos muestras esenciales.*
- *Las muestras esenciales marcan límites naturales de segmentos de la señal con un comportamiento uniforme.*

El sobremuestreo tiene como inconveniente que se generan demasiadas muestras, la mayoría de las cuales son innecesarias. Por lo tanto, lo primero que se necesita es un algoritmo para eliminar muestras redundantes. El algoritmo que se describe a continuación compara la trayectoria de la señal con una lineal. Cuando la señal se aparta de la lineal, medido por un error de interpolación, se termina un segmento de la señal y se clasifica según la forma de apartarse de la trayectoria lineal.

Se describe a continuación el proceso de muestreo en tiempo real. El algoritmo de segmentación comienza interpolando linealmente la muestra 2 de la 1 y 3. Se

calcula el error como la diferencia entre la muestra 2 interpolada, $x'[2]$ y la real $x[2]$.

Paso 1: Se esperan tres muestras $x[1]$, $x[2]$ y $x[3]$.

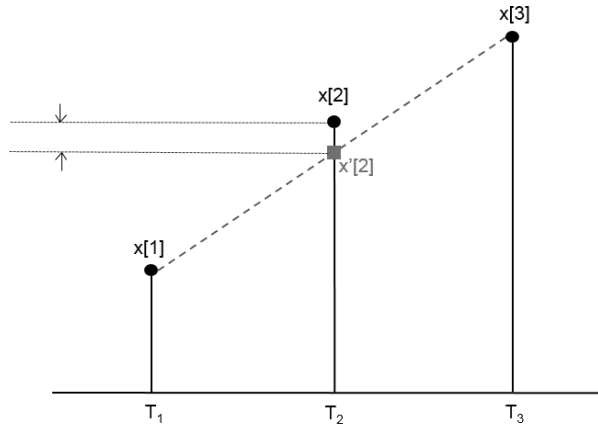


Figura 2.5 Interpolación lineal. Proceso inicial, se interpola la segunda muestra y se la compara con la real.

La muestra 2 interpolada es $x'[2] = \frac{x[1] + x[3]}{2}$ y se calcula el error de interpolación como la diferencia entre la muestra real y la interpolada.

$$err = x'[2] - x[2] \quad (7)$$

Si el error es menor que una cota de error, que notaremos E_{max} , la muestra 2 no transporta información ya que su valor puede ser calculado mediante interpolación lineal de sus muestras vecinas. El signo del error es guardado ya que será el indicador de la trayectoria de la señal real. Si el error es positivo significa que la trayectoria de la señal real se encuentra por debajo de la lineal, si es cero la trayectoria es lineal y para el caso de error negativo, la señal real se encuentra por encima de la interpolada.

El próximo paso es interpolar la muestra 3 mediante las muestras 1 y 5.

Paso2: Se esperan dos muestras, $x[4]$ y $x[5]$. Ahora $x'[3] = \frac{x[1] + x[5]}{2}$, Fig. 2.6.

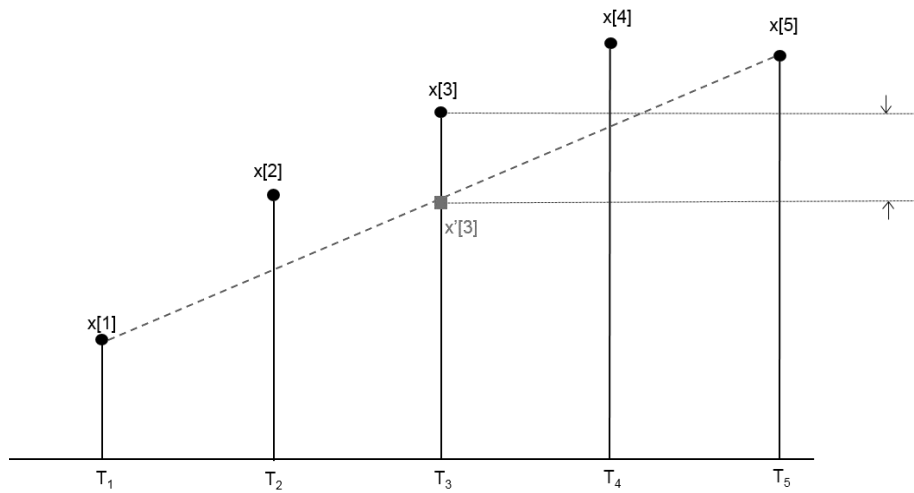


Figura 2.6 Interpolación lineal. Se interpola la tercera muestra y se la compara con la real

Se vuelve a calcular el error como:

$$err = x'[3] - x[3] \quad (8)$$

Se almacena el signo del error y si el error supera la cota máxima de error se da por concluido el segmento, de lo contrario se vuelven a esperar dos muestras más.

El proceso se repite hasta que se satisface la condición de error o se alcanza una longitud de segmento máxima, Fig. 2.7.

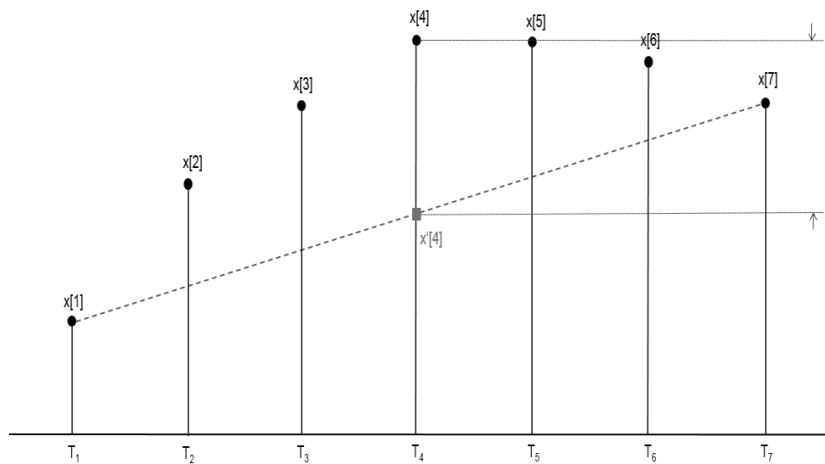


Figura 2.7. Interpolación lineal. El proceso se repite hasta alcanzar la cota de error

Las dos muestras de los extremos constituyen las muestras esenciales y marcan los límites del segmento. Se almacenan en un vector que llamaremos *marcas()*, los valores de las muestras esenciales y en un vector *tiempos()*, los índices de las muestras esenciales. Como se mencionó anteriormente, en el caso que no se supere la cota de error, se termina el segmento cuando se alcanza un número máximo de muestras y se da por terminado el segmento. En este caso el comportamiento de la señal dentro del segmento fue lineal y en función de los valores de las muestras esenciales de izquierda y derecha se clasifica el segmento como “a”, “b” o “c” como se muestra en la Fig. 2.8.

Clase	Segmento
a	
b	
c	

Figura 2.8. Los tres tipos de segmentos que se generan cuando no se supera la cota de error.

Cuando se supera el error de interpolación, se culmina un segmento determinado en sus extremos por dos muestras esenciales. Como cada vez que se interpoló la señal se registraron los errores, se conoce el subespacio en donde reside la señal real entre las dos muestras de los extremos. Este subespacio está acotado por la cota de error con respecto a la trayectoria lineal entre las muestras esenciales. La señal fue controlada, a través del sobremuestreo, lo suficientemente rápido o sea mucho más rápido que la velocidad de cambio, lo que genera trayectorias suaves. Las mismas se pueden clasificar en los tipos que se muestran en la Fig. 2.9.






Clase	Segmento
d	
e	
f	
g	
h	

Figura 2.9. Los cinco tipos de segmentos que se generan cuando se alcanza la cota de error.

Un tercer vector, notado por *clase()*, determina el comportamiento de la señal dentro del segmento estableciendo ocho clases de comportamiento denominadas: a, b, c, d, e, f, g, h. El ejemplo que se fue desarrollando en los párrafos anteriores es un segmento clase “g”. El vector *clase()* es el que relaciona las muestras y permite inferir comportamientos de la señal de la misma forma que lo haría un observador. El algoritmo de segmentación se queda solo con las muestras necesarias para describir y reconstituir la señal. La única exigencia es realizar sobremuestreo para

garantizar trayectorias suaves y que el vector $clase()$ sea representativo de la señal. En una señal real, con sobremuestreo y con el error de interpolación tendiendo a cero, la probabilidad de ocurrencia de los segmentos a, b c y h tiende a cero. ***Por lo tanto, solo cuatro segmentos son importantes para describir la señal: d, e, f, g.***

A este proceso de muestreo lo llamaremos de ahora en adelante **MCT** (**M**arcas, **C**lase, **T** tiempo)

En la fig. 2.10 se muestra el diagrama de flujo del algoritmo para segmentación y etiquetado MCT. Los parámetros de entrada son el error de interpolación y la longitud máxima del segmento. La determinación de la clase de segmento es realizada por la mayoría de los signos de interpolación. Esta forma de clasificación permite tratar a señales ruidosas como se describe más adelante. Los segmentos tipo “h” solo se darían en la condición de igual cantidad de signos positivos que negativos. Ya que este tipo de segmento no aporta información, la solución propuesta para evitarlos es esperar una interpolación más para desempatar. Remarquemos que, si el error de interpolación es pequeño y existe condición de sobremuestreo, los segmentos tipo “h” tienden a desaparecer.

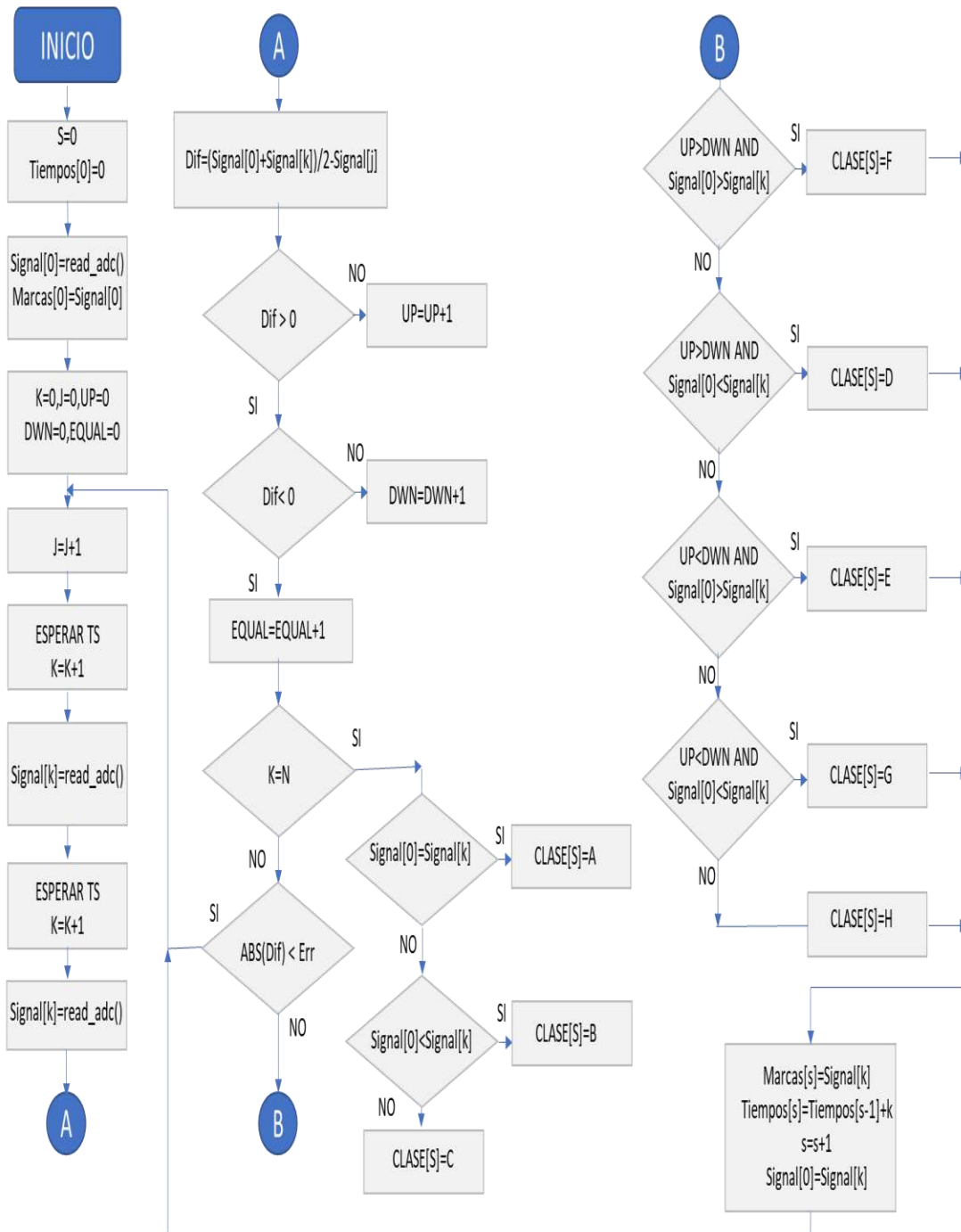


Figura 2.10. Algoritmo para la segmentación y etiquetado en tiempo real MCT.

El algoritmo requiere solo unas pocas operaciones: una suma, una resta y una división por dos (lo que podría implementarse fácilmente como un desplazamiento en un registro binario) en cada iteración. Por lo tanto, es simple de implementar en microcontroladores y puede ejecutarse en tiempo real.

2.3 Ejemplo del algoritmo aplicado a una señal empleando MatLab

Para visualizar el resultado de aplicar el algoritmo de segmentación MCT a una señal empleamos el software de procesamiento MatLab®. Se genera una señal arbitraria con sobremuestreo, limitada en ancho de banda y se aplica el algoritmo. Se genera una señal aleatoria de ruido de distribución Gausiana con media cero y dispersión unitaria y se filtra con un pasabajos con ancho de banda de 300 Hz.

```
>> Fs=7500 ; definimos una frecuencia de muestreo de 7500 Hz
>> t=0:1/Fs:0.01 ; el vector tiempo hasta 10 milisegundos.
>> [b,a]=butter(9,300/(FS/2)); coeficientes de un filtro Butterworth de noveno
orden.
Posteriormente se genera un ruido de distribución Gausiana y se filtra:
>> r=randn(size(t));
>> y=filter(b,a,r); y es el ruido filtrado
>> plot(y)
```

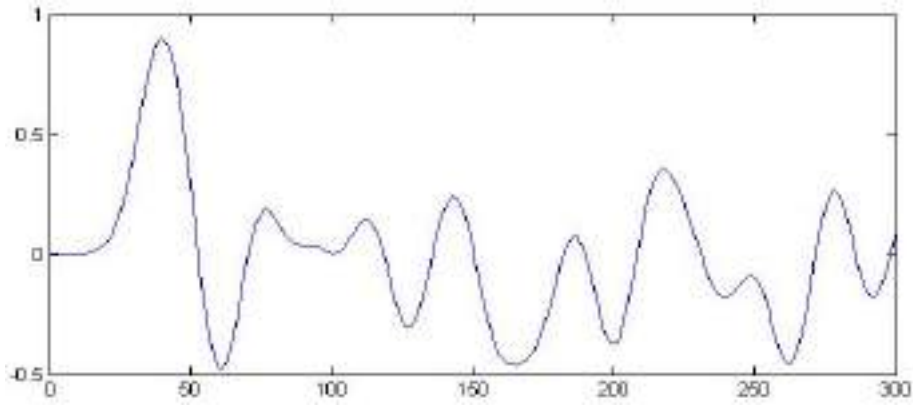



Figura 2.11. Ruido de distribución normal de ancho de banda de 300 Hz. Frecuencia de muestreo 7500 Hz.

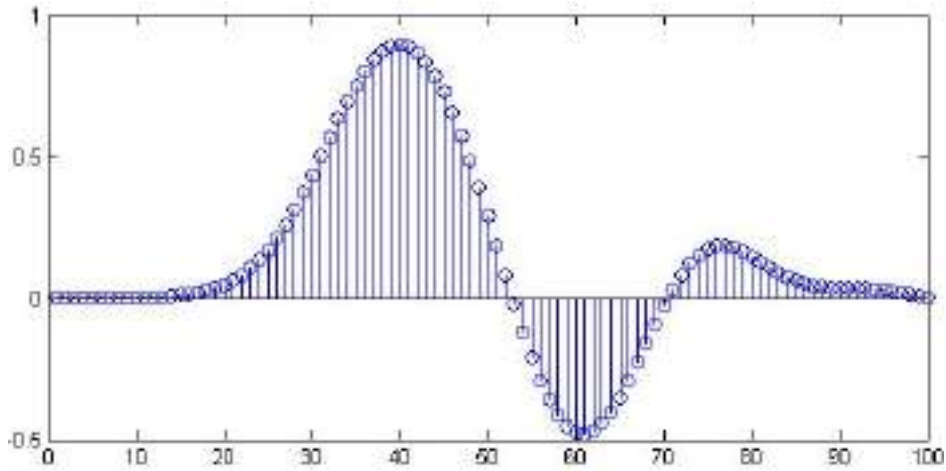


Figura 2.12. Primeras 100 muestras de la señal de la Fig. 2.11 en donde se observa claramente el sobremuestreo.

Se aplica el algoritmo de segmentación a la señal de la Fig. 2.12.

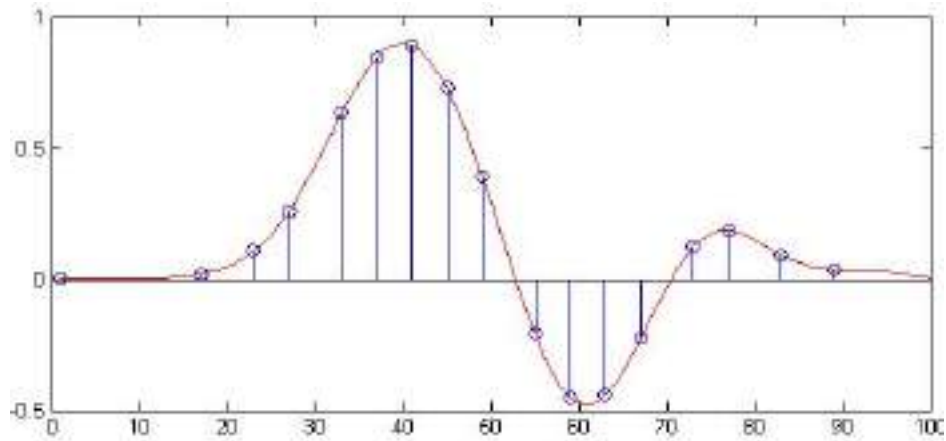


Figura 2.13 Muestras esenciales para un error de interpolación de 0.01.

La señal de la Fig. 2.13 es ahora descrita por el vector de clases:

Clase()= ddddggeeefddggef

Marcas()= {0.0000 0.0221 0.1065 0.2593 0.6351 0.8456 0.8916 0.7293 0.3911 -0.2089 -0.4500 -
0.4422 -0.2269 0.1234 0.1852 0.0912 0.0384 0.0000 0.0221 0.1065 0.2593 0.6351 0.8456 0.8916
0.7293 0.3911 -0.2089 -0.4500 -0.4422 -0.2269 0.1234 0.1852 0.0912 0.0384}

Tiempos()={1 17 23 27 33 37 41 45 49 55 59 63 67 73 77 83 89}.

Las muestras esenciales para ese error de interpolación son 17.

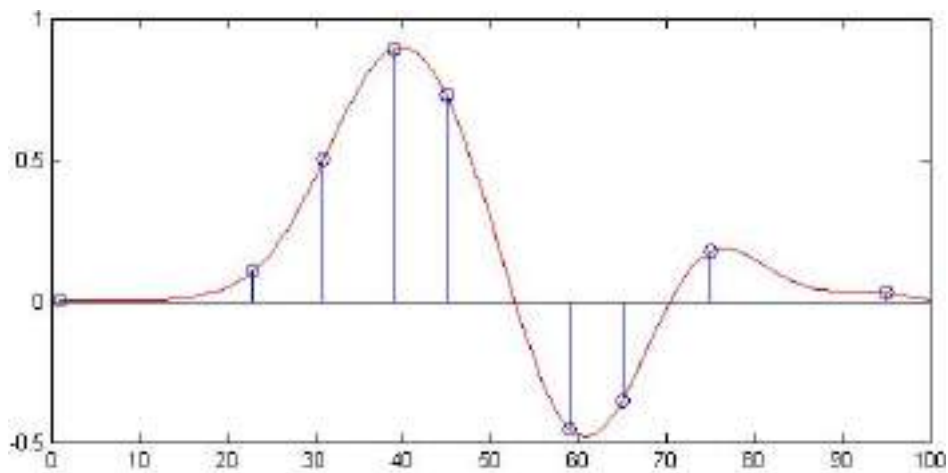


Figura 2.14. Muestras esenciales para un error de interpolación de 0.05.

Si cambiamos el error de interpolación a $\text{err}=0.05$ obtenemos los siguientes vectores:

Clase()=ddgeedge

Marcas()= { 0.0000 0.1065 0.5021 0.8938 0.7293 -0.4500 -0.3520 0.1741 0.0283}

Tiempos()={1 23 31 39 45 59 65 75 95}.

Ahora las muestras esenciales para ese error de interpolación son 8, figura 2.14.

2.4 Extracción de características de la señal

La información almacenada en el vector *clase()* nos proporciona la relación entre muestras esenciales y contiene toda la información sobre la forma de la señal, es decir, nos permite “ver” la señal que estamos muestreando. Además, el vector no tiene ninguna relación con las escalas de amplitud o tiempo, almacena solo la información cualitativa, es decir, es invariante a la amplitud y a la escala de tiempo.

Diversas características de la señal pueden extraerse fácilmente procesando los elementos de este vector. Se basa simplemente en buscar una secuencia de ocurrencia de clases.

2.4.1 *Máximos locales:*

Los máximos locales ocurren en la unión de los segmentos: “**df**”, “**de**”, “**gf**” y “**ge**”.

2.4.2 *Mínimos locales:*

Los mínimos locales ocurren en la unión de los segmentos: “**eg**”, “**ed**”, “**fg**” y “**fd**”
Los máximos y mínimos locales, que son obtenidos directamente del proceso de muestreo brindan gran cantidad de información como, por ejemplo: tendencia, estimación del periodo, estimación del ancho de banda dinámico, detección de ruido impulsivo y detección de patrones. Los máximos y los mínimos son controlados por el error de interpolación.

2.4.3 *Puntos de inflexión:*

Se produce en la unión de los segmentos: “**ef**”, “**fe**”, “**dg**” y “**gd**”.

2.4.4 *Codificación de una señal sinusoidal:*

Una señal sinusoidal queda caracterizada por una secuencia “**gefd**”. Repeticiones de la misma clase respetando la secuencia también es un patrón válido, por ejemplo “**gggeeefffd**”.

Analizando los valores de los máximos y mínimos se detecta fácilmente el periodo de oscilación y si las oscilaciones son estables, decrecientes o crecientes. En la Fig. 2.15 se muestra una oscilación estable.

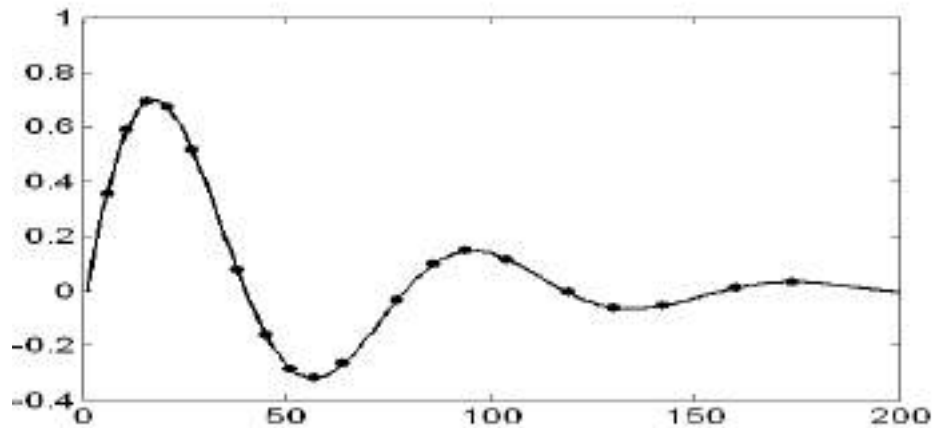


Figura 2.15 Ejemplo de oscilación estable. Los puntos indican muestras esenciales.
`clase()={ g,g,g,e,e,e,f,f,f,d,d,g,g,e,e,f,d,g}`.

Los puntos de inflexión ocurren en los cruces por cero de una senoidal pura. Si la senoidal se encuentra montada sobre una señal de más baja frecuencia, estos puntos siguen reflejando las transiciones entre la parte positiva y la negativa. Un ejemplo de aplicación sería la eliminación del corrimiento de la línea base dada por la unión de los puntos de inflexión.

2.5 Patrones de comportamiento

2.5.1 Supersegmentos

La ocurrencia consecutiva de segmentos de una misma clase indica un comportamiento homogéneo. Dado que son cuatro los segmentos importantes, “d”, “e”, “f” y “g” tenemos cuatro clases de supersegmentos. En la Fig. 2.16 se muestran los dos patrones que responden a características estables, es decir, la señal tiende a un valor finito. Como el muestreo reconoce este tipo de comportamiento, el valor de estado estacionario puede ser estimado. Un ejemplo de

aplicación de este tipo de predicción sería la de un sensor de temperatura que, estando a temperatura ambiente, es colocado en el lugar de medición. La temperatura del objeto puede ser estimada sin tener que esperar que el sensor alcance la temperatura final.

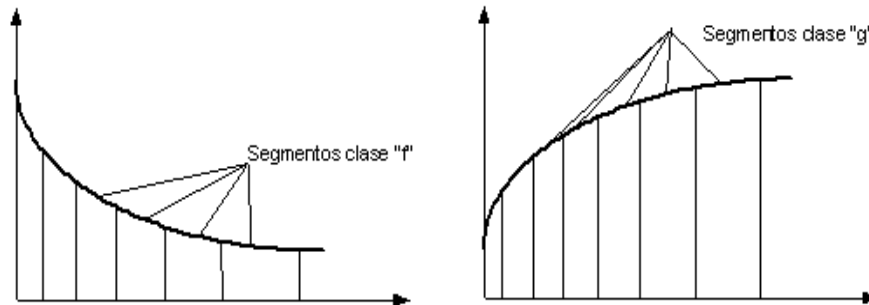


Figura 2.16. Los dos patrones de supersegmentos estables.

La expresión general de exponencial en la Fig. 14 es:

$$x(t) = X_{ss} - (X_{ss} - X_i) e^{-t/\tau} \quad (9)$$

Donde X_{ss} es el valor en estado estacionario y X_i el valor inicial. Tenemos dos incógnitas X_{ss} y τ , el valor de estado estacionario y la constante de tiempo, por lo tanto, planteamos la ecuación en dos puntos de la curva que llamamos a y b.

La derivada de $x(t)$ respecto de en $t=0$ es:

$$\left. \frac{dx}{dt} \right|_{t=0} = \frac{X_{ss} - X_i}{\tau} \quad (10)$$

La pendiente de los segmentos puede emplearse como la aproximación de la derivada en (10) aplicada en los puntos a y b, ver Fig. 2.17. Si consideramos la derivada en el punto a, y empleando este punto como valor inicial obtenemos:

$$\frac{X_{ss} - X_{ia}}{\tau} = M_a \quad (11)$$

Donde $M_a = Y_a/X_a$ es la aproximación de la pendiente en el punto a . Repitiendo el análisis ahora en el punto b , y empleando este punto como valor inicial obtenemos:

$$\frac{X_{ss} - X_{ib}}{\tau} = M_b \quad (12)$$

Donde $M_b = Y_b/X_b$ es ahora la aproximación de la pendiente en el punto b .

Resolviendo (11) y (12) obtenemos (13) que nos permite calcular el valor de estado estacionario en función de los vectores $marcas()$ y $tiempos()$.

$$X_{ss} = \frac{M_a X_{ib} - M_b X_{ia}}{M_a - M_b} \quad (13)$$

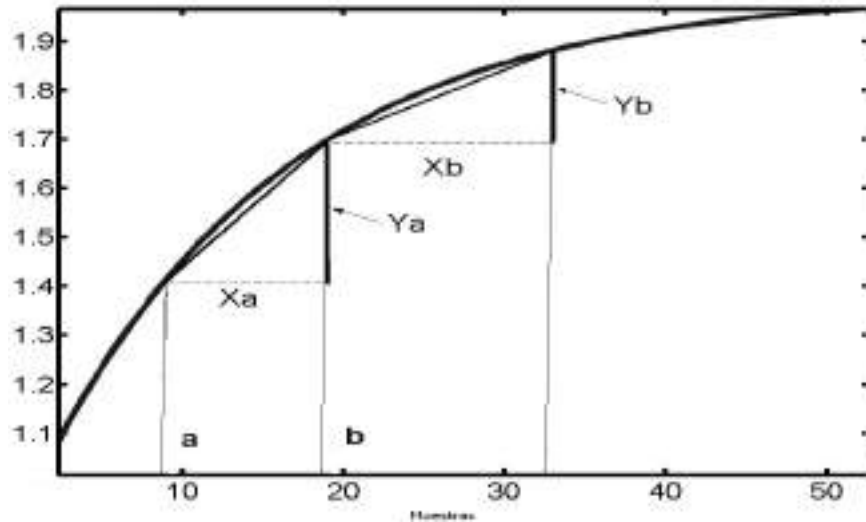


Figura 2.17. Predicción del valor estacionario de la señal basada en los cambios de pendiente en los segmentos tipo “g”.

2.5.2 Detección de ruido impulsivo

El ruido impulsivo se define como la interferencia en una señal causada por un ruido durante un breve intervalo de tiempo. Una brusca reducción de la longitud del segmento inicia el algoritmo de detección. Esto indica la presencia de ruido. Para ser considerado como ruido impulsivo, el cambio en la amplitud debe ser significativo. Una vez que el algoritmo se ha iniciado, se busca el primer máximo o mínimo. A continuación, se calcula el cambio de amplitud a partir del valor antes

de la reducción de la longitud. La longitud del ruido impulsivo se calcula a partir de la diferencia temporal entre el primer segmento y el último. Por ejemplo, la secuencia "dddfff" indica la aparición de un máximo, y la longitud de impulso es la diferencia entre la posición temporal de la última "f" y la primera "d".

2.5.3 Estimación de tendencia

La tendencia se calcula teniendo en cuenta el incremento o decremento de los máximos y mínimos en una ventana temporal. La tendencia creciente se normaliza a 1 cuando todos los nuevos máximos son más grandes que los anteriores en la ventana. Por otro lado, la tendencia es decreciente y normalizada a -1, cuando todos los nuevos máximos son menores que los anteriores en la ventana. El mismo procedimiento se ejecuta para mínimos. En la Figura 2.18, T_x es el índice temporal y Max_x es el valor de los máximos y Min_x el de los mínimos.

La tendencia de los máximos y mínimos se calcula utilizando (14) y (15).

$$Trend_{max} = \frac{\sum_{max}(T_{x+1}-T_x)(Max_{x+1}-Max_x)}{\sum_{max}(T_{x+1}-T_x)|Max_{x+1}-Max_x|} \quad (14)$$

$$Trend_{min} = \frac{\sum_{min}(T_{x+1}-T_x)(Min_{x+1}-Min_x)}{\sum_{min}(T_{x+1}-T_x)|Min_{x+1}-Min_x|} \quad (15)$$

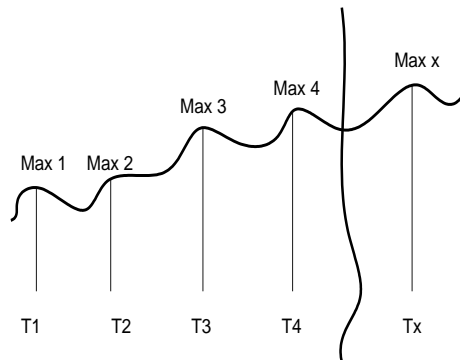


Figura 2.18. La tendencia es calculada empleando los máximos y mínimos locales y la posición temporal de ellos.

El rango de las expresiones (14) y (15) es de -1 a 1. La tendencia total es el promedio de los máximos y los mínimos:

$$Trend = \frac{Trend_{max} + Trend_{min}}{2} \quad (16)$$

Disponer de la tendencia de los máximos y los mínimos por separado proporciona información adicional. Si la tendencia de los máximos es positiva y la tendencia de los mínimos es negativa significa que la señal posee oscilaciones crecientes. Si la tendencia de los máximos es negativa y la tendencia de los mínimos es positiva significa que la señal posee oscilaciones estables. Es importante remarcar que en los dos casos la tendencia total puede ser similar. Si los signos de las tendencias de los máximos y mínimo son iguales, la tendencia total es el promedio de las dos.

2.5.4 Estimación del ancho de banda para segmentos de la señal

El contenido espectral de una señal sensorial, en general, varía en función del tiempo. El filtro *antialiasing* proporciona solo una cota superior. Si por ejemplo aplicamos la STFT (*Short Time Fourier Transform*) a ventanas de la señal obtendríamos distintos contenidos espectrales. El tiempo de separación entre máximos y mínimos locales es un estimador del contenido espectral en una porción de la señal. Como este indicador es obtenido directamente desde el muestreo, proporciona un mecanismo para marcar regiones de la señal para otros procesos de orden superior. Como ejemplo de aplicación de este indicador se puede mencionar el marcado de regiones para analizar el contenido espectral y la sincronización de ventanas de observación con las características de la señal.

Si consideramos al tiempo entre máximos locales como una variable aleatoria, la varianza de ella proporciona información acerca de la regularidad de patrón de señal analizado. Por ejemplo, varianza elevada implica un comportamiento irregular o ruidoso y varianza tendiente a cero, un patrón oscilante uniforme.

2.5.5 Estimación de frecuencia y fase de un patrón oscilatorio

Como se menciona en el párrafo anterior, analizando las características de los máximos y mínimos de una señal es posible determinar la presencia de un patrón oscilatorio y estimar la frecuencia y fase. Al ser MCT un proceso en el dominio temporal, es sensible al ruido y atenta contra la determinación de frecuencia. Para ser aplicado a señales reales es necesario filtrar la señal. Al iterar el muestreo MCT es decir, reconstituir la señal muestreada y aplicar nuevamente MCT, es posible lograr un filtrado pasabajos cuya respuesta en frecuencia es controlada por el número de iteraciones y el error de interpolación. Luego de cada iteración se analiza la señal resultante en busca de una señal periódica detectada por la varianza de la posición temporal de los máximos tendiendo a cero. Si la varianza no tiende a cero y se estabiliza en un valor finito entonces no existe una frecuencia regular en la ventana de la señal analizada. Como la iteración MCT mantiene la posición temporal de las muestras, no existe un corrimiento de fase por lo tanto es posible computar la fase de la señal periódica.

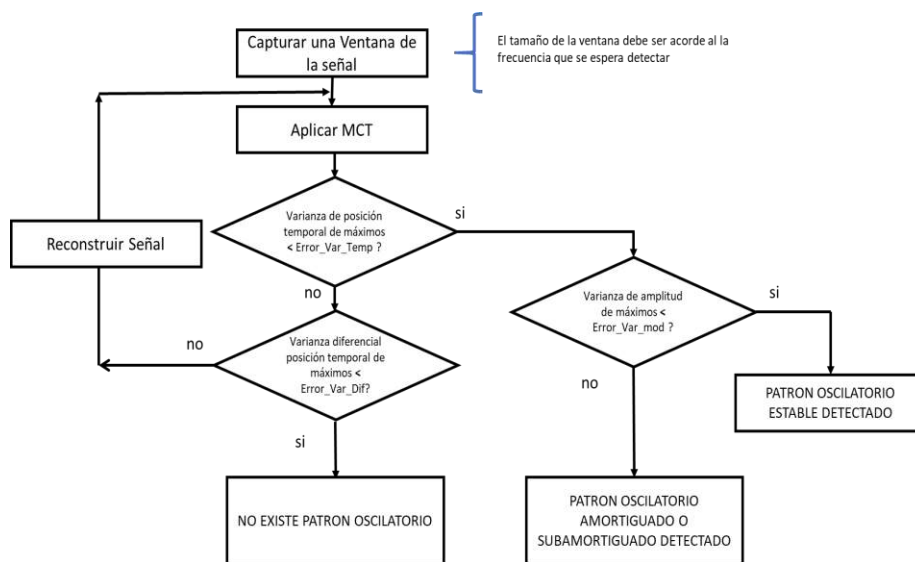


Figura 2.19. Diagrama de flujo para determinar la presencia de señales periódicas y estimar su frecuencia y fase empleando MCT iterativamente.

En la figura 2.19 se observa el diagrama de flujo para la detección y cómputo de un patrón oscilatorio. La no existencia de una señal oscilatoria es determinada

empleando el cambio en la varianza de la posición temporal de los máximos. Cuando este cambio relativo se estabiliza en un valor alto implica que no se ha encontrado un patrón oscilatorio regular. El algoritmo fue probado para señales ruidosas, aperiódicas y periódicas empleando MatLab. Se generó la siguiente señal:

```

Fs=7500; % frecuencia de muestreo
t=0:1/Fs:0.1; % 100 ms de ventana de observación
x=sin(2*pi*50.0*t+pi/4); % señal periódica 50 Hz
r=rand(size(t)); % Ruido
xr=x+0.4*r; % Señal contaminada con ruido

```

En la Fig. 2.20 se observa la señal ruidosa y en la Fig. 2.21 la evolución del cálculo de la frecuencia y la varianza de la posición temporal de los máximos. El error de interpolación fue seteado a 1.0 E-05. La frecuencia hallada fue de 49.89 Hz.

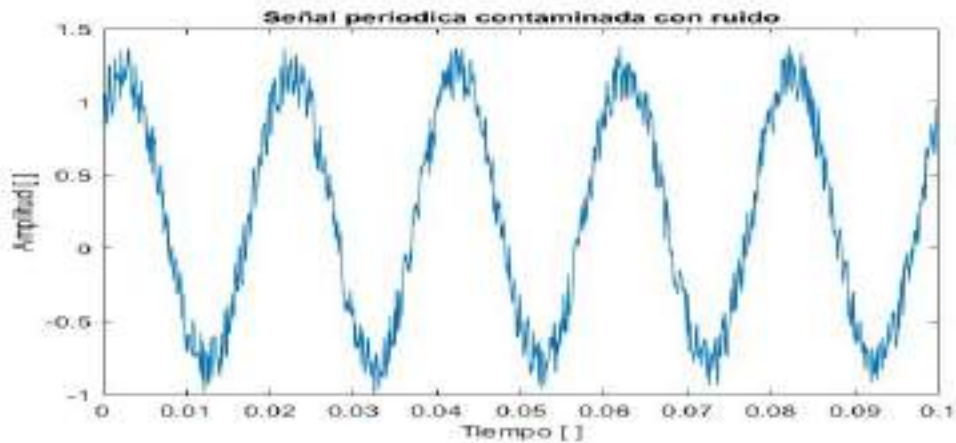


Figura 2.20. Señal periódica 50 Hz contaminada con ruido.

En la Fig. 2.22 se observa la señal periódica original de 50 Hz y la obtenida luego de la detección. Nótese que se detecta sin corrimiento temporal.

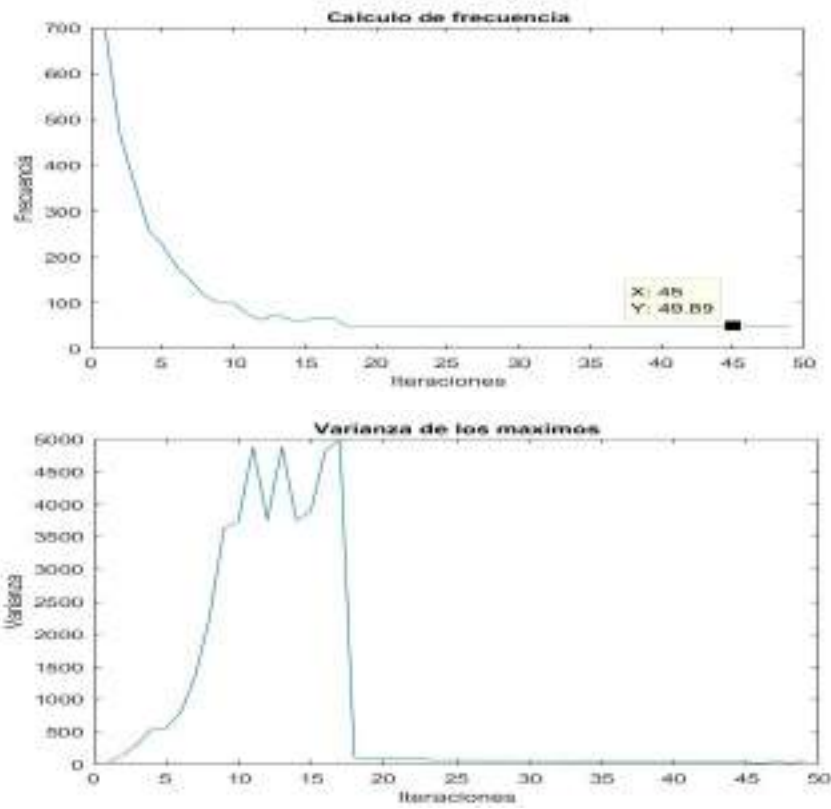


Figura 2.21. Gráficos de la evolución del cálculo de la frecuencia de la señal y la varianza de la posición de los máximos.

Se realizó la prueba con una señal aperiódica.

$F_s=7500$; % frecuencia de muestreo

$t=0:1/F_s:0.1$; % 100 ms de ventana de observación

$r=rand(size(t))$; % Señal de ruido

$xr=r$; % Señal de test ruidosa

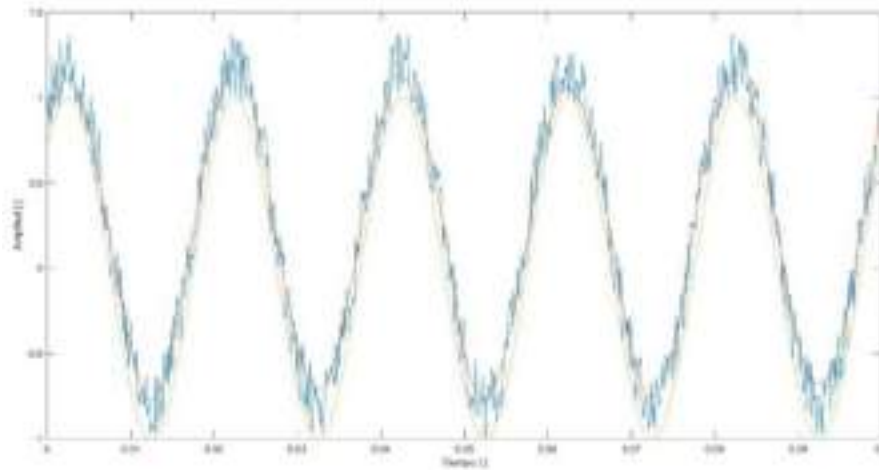


Figura 2.22. Señal periódica original, con ruido y reconstituida después de 50 iteraciones con error de interpolación $1 \text{ E-}05$.

En la Fig. 2.23 se observa la evolución de la varianza de la posición temporal de los máximos para el caso de una señal aperiódica. El valor de varianza no disminuye o podría estabilizarse en un valor, pero no cercano a cero, indicando la falta de periodicidad de la señal bajo análisis.



Figura 2.23. Evolución de la varianza de la posición de los máximos para el caso de una señal aperiódica.

2.6 Análisis de los errores de interpolación

Sea $f(t)$ una señal real y limitada en banda. Esta señal cumple con las condiciones para ser expresada en desarrollo en serie. Mediante este desarrollo podemos comparar la señal interpolada con la real en los instantes de comparación para calcular los errores que se producen.

El desarrollo en serie de Taylor en el instante $t+\Delta t$, conociendo la función en el instante t es:

$$f(t + \Delta t) = f(t) + f'(t)\Delta t + \frac{1}{2!}f''(t)\Delta t^2 + \frac{1}{3!}f'''(t)\Delta t^3 + \dots \quad (17)$$

La primera aproximación por interpolación lineal se produce entre la función en $f(t+\Delta t)$ y la aproximación lineal dada por:

$$\frac{f(t)+f(t+2\Delta t)}{2} \quad (18)$$

La diferencia se compara con el máximo error de interpolación tolerado. Si no se supera la cota de error se continúa interpolando.

La segunda comparación se realiza entre:

$$\frac{f(t) + f(t + 4\Delta t)}{2} \langle \rangle f(t + 2\Delta t) \quad (19)$$

La expresión general de la comparación de la señal real con la interpolada hasta que termina el segmento es:

$$\frac{f(t)+f(t+2k\Delta t)}{2} \langle \rangle f(t + k\Delta t) \quad (20)$$

Con $k=1,2,3\dots$ hasta la máxima longitud de segmento o el valor de k que genera un error mayor que la cota.

Desarrollando la serie para las funciones expresadas en (20), simplificando y agrupando términos comunes obtenemos:

$$err = \sum_{n=2}^{\infty} f^{(n)}(t) \frac{k^n}{n!} \Delta t^n (1-2^{n-1}) \quad (21)$$

Como $\Delta t = \frac{1}{F_s}$, donde F_s es la frecuencia de muestreo:

$$err = \sum_{n=2}^{\infty} f^{(n)}(t) \frac{k^n}{F_s^n n!} (1-2^{n-1}) \quad (22)$$

Notemos que el error de interpolación depende inversamente de la frecuencia de muestreo. En la expresión del error se nota claramente la relación entre la velocidad de cambio de la señal, expresada por las derivadas de orden n de la señal y la frecuencia de muestreo.

2.7 Representación formal de la señal sensorial

Se denomina alfabeto a un conjunto finito Σ de símbolos. Una señal discreta o cadena ω es una secuencia de símbolos del alfabeto Σ . Llamaremos Σ^* al conjunto de todas las señales discretas y finitas sobre Σ . La cantidad de símbolos que componen una señal $\omega \in \Sigma^*$ se denomina la longitud de la señal y se representa como $|\omega|$. Las posiciones de cada símbolo en una señal ω se numeran de $1, \dots, |\omega|$, denotando $\omega[i]$ el símbolo en la posición i -ésima de ω .

En contexto de la representación planteada en esta tesis se propone representar una señal $h \in \Sigma^*$ a partir del conjunto $\Sigma_0 = \{a, b, c, d, e, f, g, h\}$ que se muestran en las Figuras 2.8 y 2.9. Es decir, dada una señal $h \in \Sigma^*$ se puede escribir como combinación lineal como $h = \prod_{i=1}^N \alpha_i \bullet x_i$ con $x_i \in \Sigma_0$. Donde, α_i es una transformación lineal (expansión, contracción, traslación) y \prod denota una concatenación de funciones en Σ_0 .

Máximos locales: Los máximos locales ocurren en la unión de los segmentos:

“ $\prod(d, f)$ ”, “ $\prod(d, e)$ ”, “ $\prod(g, f)$ ” y “ $\prod(g, e)$ ”

Mínimos locales: Los mínimos locales ocurren en la unión de los segmentos:

“ $\prod(e, g)$ ”, “ $\prod(e, d)$ ”, “ $\prod(f, g)$ ” y “ $\prod(f, d)$ ”

Los máximos y los mínimos son controlados por el error de interpolación.

Puntos de inflexión: Se produce en la unión de los segmentos:

“ $\prod(e, f)$ ”, “ $\prod(f, e)$ ”, “ $\prod(d, g)$ ” y “ $\prod(g, d)$ ”

Codificación de una señal sinusoidal: Se produce cuando se detecta una secuencia $g \bullet e \bullet f \bullet d$. Repeticiones de la misma clase respetando la secuencia también es un

patrón válido, por ejemplo $g \bullet g \bullet g \bullet g \bullet e \bullet f \bullet f \bullet f \bullet d \bullet d \bullet d$

Analizando los máximos y los mínimos se detecta fácilmente el periodo de oscilación y si las oscilaciones son estables, decrecientes o crecientes.

2.8 Reconstrucción de la señal a partir de las muestras esenciales

Mediante el muestreo propuesto se captura una información importante de lo que ocurre entre las muestras esenciales. Ésta es el subespacio en donde reside la señal original. Este subespacio se encuentra acotado por los errores de interpolación que se comprueban cada $2 \Delta t$ segundos. Si la frecuencia de muestreo es mucho mayor que el ancho de banda de la señal y si el error de interpolación es pequeño, la señal es de trayectoria suave en el subespacio donde reside.

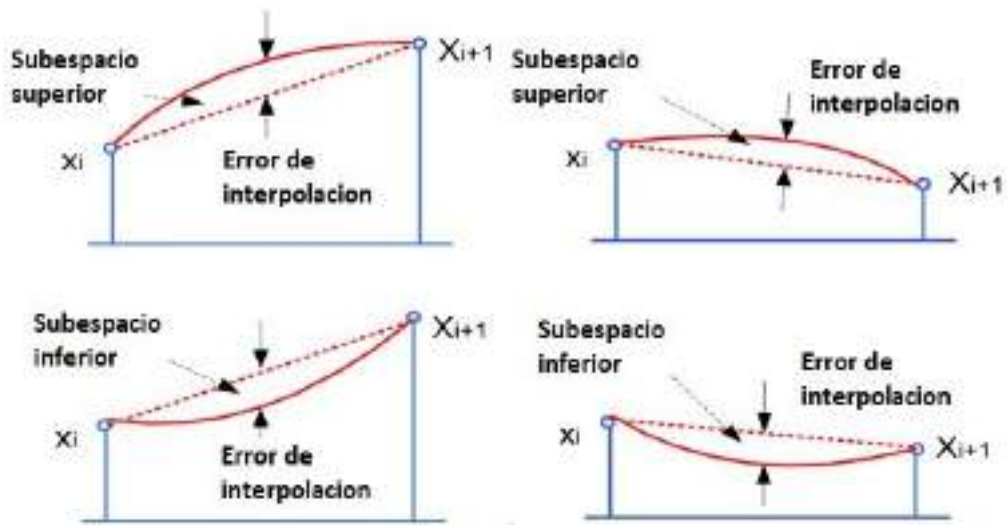


Figura 2.24. Las cuatro trayectorias simplificadas entre muestras esenciales.

En la Fig. 2.24 se observan las cuatro trayectorias de la señal entre muestras esenciales.

La señal reconstituida puede ser descrita como una combinación lineal de funciones elementales escaladas en tiempo y desplazadas.

$$f(t) = \sum c_h \varphi(t / a_k - Ti) \quad (23)$$

Donde $\varphi(t)$ es la función normalizada de generación. Dos formas básicas generan la señal, una para el subespacio inferior a la recta entre las muestras esenciales y otra para el superior como se observa en la Fig. 2.25. En la Figura 2.26 se observan los segmentos que se generan en función de los parámetros α y β .

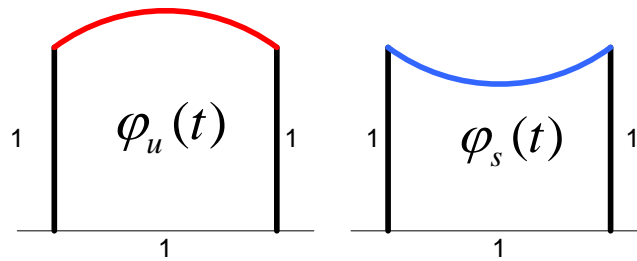


Figura 2.25. Funciones de generación de la señal entre las muestras esenciales. Éstas se adaptan a la longitud de segmento y los valores inicial y final coinciden con las muestras izquierda y derecha.

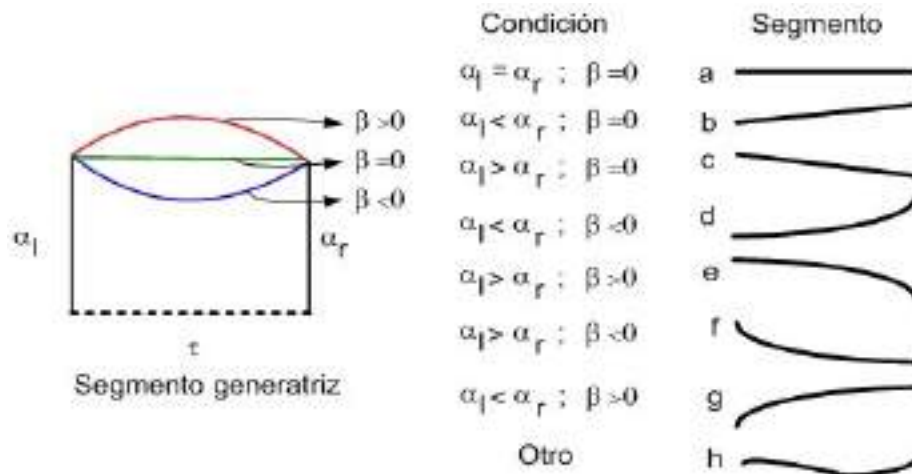


Figura 2.26. En función de los valores de dos parámetros α y β se generan los segmentos que unen las muestras esenciales.

Si se emplea interpolación lineal para reconstituir la señal, las dos funciones convergen a la recta que unen las muestras esenciales. Esta sencilla reconstrucción es importante para aplicaciones prácticas ya que es fácilmente implementada en sistemas embebidos.

Dado que la señal pasa exactamente por las muestras esenciales, hay continuidad entre el inicio y el fin de segmentos y la primera derivada coincide entre segmentos adyacentes, polinomios *Spline* pueden ser empleados para reconstituir la señal.

2.9 Estrategias de selección de muestras esenciales

El algoritmo MCT requiere una condición de sobremuestreo. En esta sección, se sugieren diferentes estrategias para el algoritmo de segmentación y etiquetado, sobre todo cuando no es posible lograr la condición de sobremuestreo. Los casos

pueden ser debido a que las muestras ya fueron adquiridas o para analizar una señal en donde no se puede controlar la resolución.

2.9.1 Condición de sobremuestreo

Sobremuestreo significa adquirir muestras a una velocidad mucho mayor que la frecuencia de Nyquist. En este contexto y en relación con el error de interpolación, la condición de sobremuestreo significa que cualquier muestra entre dos muestras puede ser reemplazada por una muestra interpolada lineal y el error de sustitución será menor que el error de interpolación.

Algoritmo:

Paso 1: Adquirir tres muestras $x[1]$, $x[2]$ and $x[3]$.

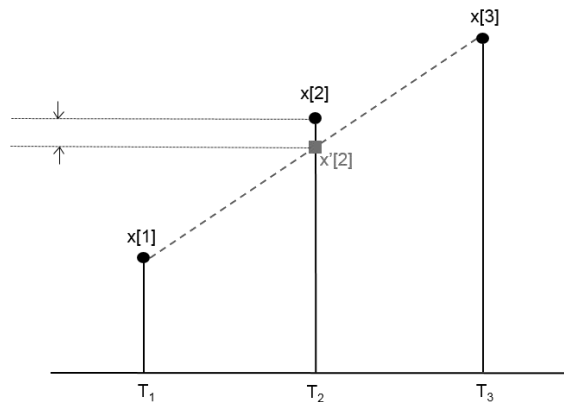


Figura 2.27. El muestreo MCT se inicia con tres muestras.

La muestra 2 interpolada es:

$$x'[2] = \frac{x[1] + x[3]}{2}$$

Calcular el error como la diferencia entre el interpolado y el real.

$$err = x'[2] - x[2]$$

Si el error es menor que un valor prefijado significa que la muestra 2 no lleva información ya que su valor puede obtenerse de sus vecinos. Esto es importante, ya

que con el sobremuestreo el valor de cada muestra no es relevante, solo importa cómo se relaciona con sus vecinos. A partir de esta operación, el signo de error se acumula para seguir la evolución de la señal real.

Paso 2: Adquirir dos muestras más e interpolar la muestra 3 de 1 y 5.

$$x'[3] = \frac{x[1] + x[5]}{2}$$

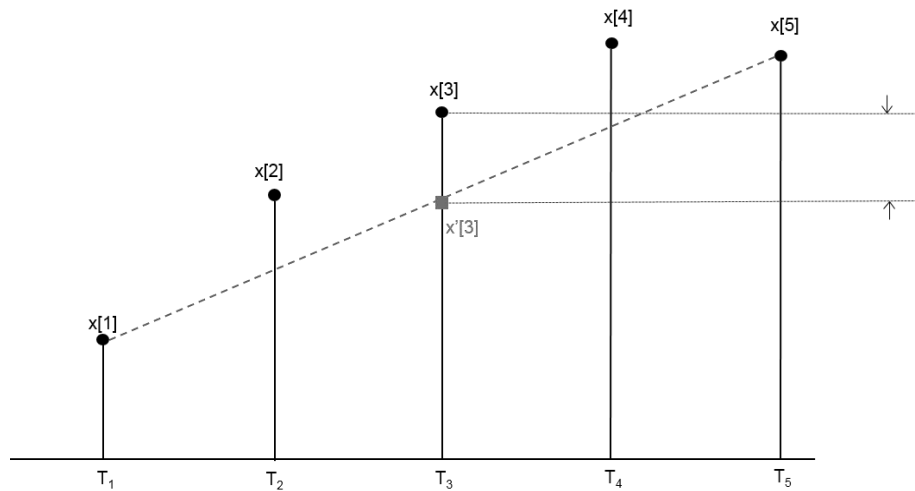


Figura 2.28. Dos muestras adicionales son adquiridas y se interpola y compara la tercera muestra.

El error de interpolación ahora es:

$$err = x'[3] - x[3]$$

El proceso se repite hasta que se satisface la condición de error o se alcanza una longitud de segmento máxima.

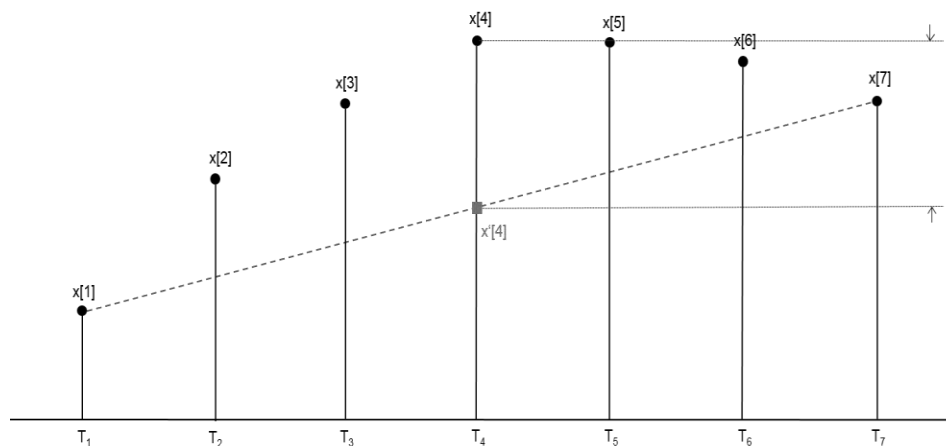


Figura 2.29. Sigüientes dos muestras $x[6]$ y $x[7]$.

Supongamos que la condición de error se satisface con la muestra 7. El segmento ha terminado. La pregunta es: ¿Cuál es la muestra que debe ser marcada como importante y que marcará el inicio del nuevo segmento? La respuesta es que, si tenemos sobremuestreo no hay una muestra individual que haya causado que el segmento termine. Hay dos candidatos:

Muestra 7. Algoritmo estándar

Muestra 4. Algoritmo Volver atrás

Analicemos ambas opciones.

Algoritmo estándar

La muestra 7 es la muestra derecha del segmento y la muestra izquierda del siguiente segmento. El algoritmo es extremadamente simple ya que todos los punteros (el puntero de muestra izquierdo, el puntero de interpolación y el último puntero de muestra) se restablecen para el nuevo segmento y el proceso de muestreo comienza desde el paso 1.

Algoritmo Volver atrás

La muestra 4 es la muestra derecha del segmento y la izquierda del nuevo segmento. El algoritmo es más complejo ya que la primera interpolación para el siguiente segmento incluye las muestras 4, 5 y 6; muestras que ya se han adquirido y que deben ser almacenadas en memoria. Esto implica un buffer de memoria para las muestras. Además, se necesitan tres punteros (muestra izquierda, interpolación y último puntero de muestra) y no se pueden restablecer para el nuevo segmento. Hay una necesidad de comprobar si las muestras se almacenan en la memoria o la necesidad de esperar el instante de muestreo.

Por lo tanto, con una condición de sobremuestreo, la primera opción, algoritmo estándar, proporciona:

- Segmentos más grandes,
- Algoritmo más simple.

Mientras que la segunda opción, algoritmo volver atrás, proporciona:

- Menor error de reconstrucción.
- Algoritmo más complejo.

Como conclusión, ya que este algoritmo debe ser apto para tiempo real y lo suficientemente simple para ser embebido en el punto de adquisición de la señal, el Algoritmo *Estándar* es la mejor opción con sobremuestreo.

2.9.2 Condición de no sobremuestreo

Si no hay sobremuestreo, el valor de la muestra es más importante que cómo se relaciona con sus vecinos. En una condición extrema la señal segmentada es la señal de entrada (todos los segmentos son de longitud uno). Para el algoritmo estándar, en la condición de sobremuestreo, el segmento más corto es de dos muestras, mientras que, para el algoritmo de Volver atrás, es una muestra. Por lo tanto, si no se satisface el sobremuestreo, debe usarse el algoritmo *volver atras*. Se debe incluir un control adicional. En el método de muestreo propuesto para las dos opciones, siempre se obtienen dos muestras y luego se realizan los cálculos. Puesto que, en esta condición, cada muestra es importante, es necesario comprobar mediante interpolación lineal si la penúltima muestra puede ser reemplazada por una muestra interpolada lineal.

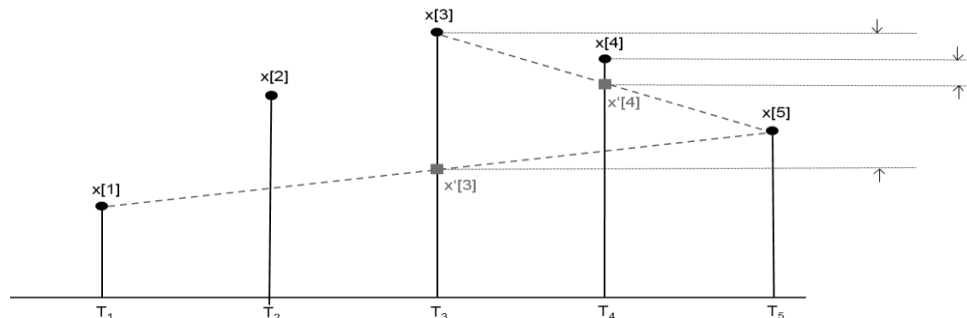


Figura 2.30. Doble control de interpolación para el caso de no poder contar con sobremuestreo de la señal.

Por ejemplo, cuando se adquieren las muestras 4 y 5, Fig. 2.28, se deben cumplir dos condiciones de error.

$$x'[3] = \frac{x[1] + x[5]}{2} \quad x'[4] = \frac{x[3] + x[5]}{2}$$

$$err1 = x'[3] - x[3], \quad err2 = x'[4] - x[4]$$

El error de interpolación considerado para la interpolación es el máximo entre $err1$ y $err2$.

2. 10 Conclusiones

El propósito de esta técnica de muestreo y procesamiento es extraer información básica de la señal de entrada de la misma manera que un ser humano podría proporcionarla observando la señal. Esta técnica emula a un observador, en el sentido de que relaciona muestras para inferir un comportamiento global, detectando, por ejemplo, un mínimo o un máximo, o inferir parámetros de forma de señal.

Esta técnica basa su análisis en la hipótesis de que la información necesaria no está en una sola muestra, sino que está contenida en la relación entre ellas. El objetivo de esta técnica es almacenar solo las muestras significativas, con las cuales es posible mantener información sobre la señal, y de las cuales es posible extraer diversas características simples pero importantes. De esta forma, es posible tener el mismo contenido de información, eliminando la redundancia de una señal de entrada sobremuestreada.

Al mismo tiempo que la señal se adquiere, se implementa una clasificación de las muestras crudas, que describe el comportamiento de la señal. El algoritmo puede dividirse en dos partes: la fase de segmentación y etiquetado logrando en los tres vectores MCT información cuantitativa y cualitativa.

3 APLICACIONES EN SEÑALES UNIDIMENSIONALES

RESUMEN

En este capítulo se presenta el empleo de la representación MCT como fuente de información para adquirir conocimiento en señales unidimensionales. Se desarrollan diferentes aplicaciones que van desde filtrado, análisis de señales de vibración, señales de audio, hasta señales biomédicas. El capítulo detalla aplicaciones básicas y complejas en donde se analizan patrones de comportamiento que evidencian las características objetivo.

3.1 Filtrado de señales empleando representación MCT

Como se mencionó en el capítulo anterior, al iterar el muestreo MCT es decir, reconstituir la señal muestreada y aplicar nuevamente MCT, es posible lograr un filtrado pasabajos cuya respuesta en frecuencia es controlada por el número de iteraciones y el error de interpolación. El filtrado es consecuencia de la reconstrucción empleando trayectorias simplificadas. Para cuantificar las características del filtrado se procedió a filtrar una señal testigo con distintos valores de error de interpolación y número de iteraciones. Una señal $\text{sinc}(x)$ de 15 Hz, muestreada a 100 Hz, se eligió como señal patrón. En la Fig. 3.1 se observan los espectros de la señal original junto con los resultantes de aplicar iteraciones con un error de interpolación cercano a cero, $1.0E-08$.

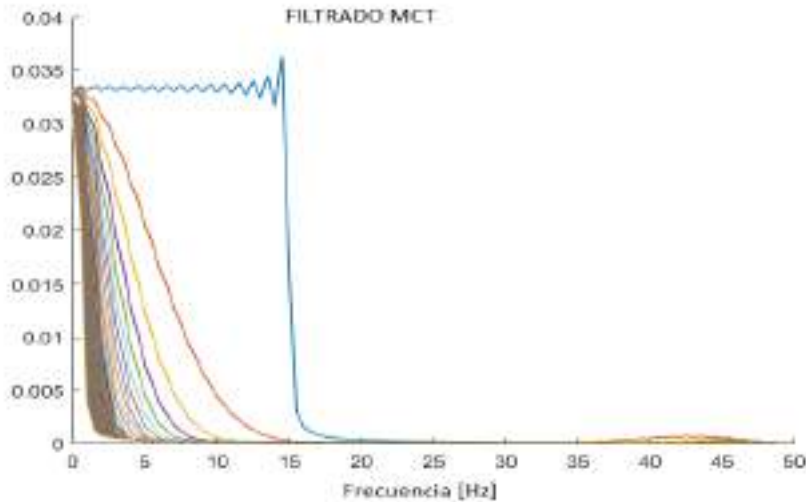


Figura 3.1 Modulo del espectro de una $\text{sinc}(x)$ de 15 Hz de ancho de banda y los espectros resultantes con iteraciones en pasos de 10 hasta 1000 con error de interpolación tendiente a cero ($1.0 \text{ E-}08$).

En la Fig. 3.2 se observa las características del filtro para un mayor error de interpolación, $1.0 \text{ E-}02$, y con iteraciones de una hasta 10 en pasos de uno.

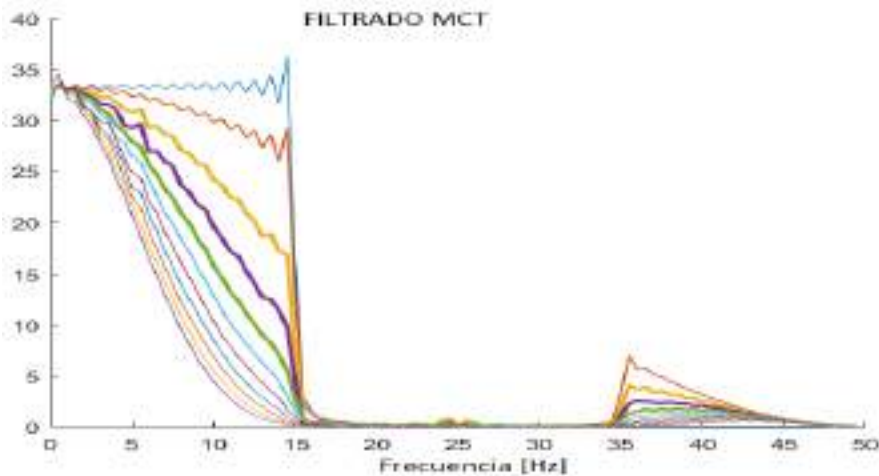


Figura 3.2 Modulo del espectro de una $\text{sinc}(x)$ de 15 Hz de ancho de banda y los espectros resultantes con iteraciones en pasos de 1 hasta 10 con error de interpolación ($1.0 \text{ E-}02$).

El ruido espectral que se observa en la Fig. 3.2 en torno a los 35 Hz es debido a que la reconstrucción de la señal se realizó con rectas que unen las muestras esenciales. Tres aspectos son importantes de destacar del filtrado MCT. Primero, el filtrado no posee retardo o desplazamiento de fase. Esto permite operaciones algebraicas simples, muestra a muestra, sobre la señal resultante. Por ejemplo, podríamos

obtener el contenido de alta frecuencia de la señal substrayendo la señal iterada con respecto a la original. El segundo aspecto es que no presenta transitorios. Y la tercera característica es que luego de cada iteración es posible analizar la señal para, por ejemplo, tomar decisiones si se ha alcanzado el filtrado esperado o si es necesario seguir iterando. Es posible en cada iteración analizar la señal ya que se dispone de la terna MCT resultante. Es importante destacar que la iteraciones pueden limitar las aplicaciones en tiempo real.

Para ejemplificar el filtrado se procedió a filtrar señales de sismógrafos sobre terremotos reales. En señales sísmicas el filtrado pasabajo es un proceso fundamental [Yamamoto S, et al, 2008] en especial en algoritmos de detección temprana analizando las ondas **P** (ondas Primarias). La figura 3. 3 muestra la señal de aceleración de un terremoto de magnitud 6.9 Mw. Las señales fueron obtenidas de la base de datos interactiva **IRIS** (Incorporated Research Institutions for Seismology) Wilber3 [<http://ds.iris.edu/wilber3/>]. La figura 3.4 muestra los primeros 10 segundos de la señal de la anterior figura. Los primeros segundos son importantes en la estimación de la magnitud del sismo ya que las ondas **P** viajan a mayor velocidad que las ondas **S** (secundarias). Estas ondas son las que generan los desplazamientos terrestres, provocando daños en función de la magnitud del sismo.

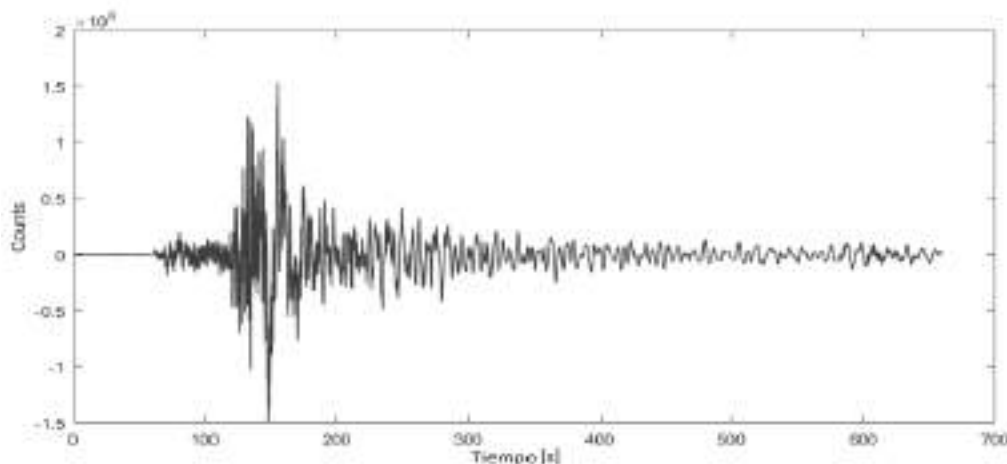


Figura 3.3 Sismo de 6.9 Mw ocurrido el 24 de abril del 2017, cerca de la costa en la región central de Chile, registrado en el observatorio Las Campanas (LCO).

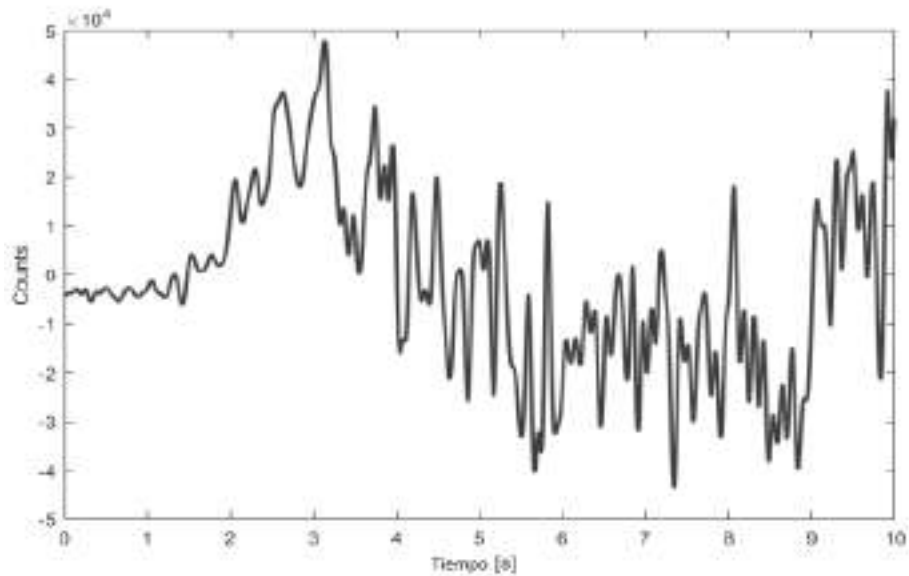


Figura 3.4 Primeros 10 segundos de la señal de la figura 3.3

El filtrado MCT se aplicó a la señal de la Fig. 3.4 con distinto número de iteraciones y error de interpolación. La Fig. 3.5 muestra la señal filtrada para un gran número de iteraciones, en este caso 2500. De esta señal filtrada es posible inferir la magnitud del sismo asociado a las ondas S.

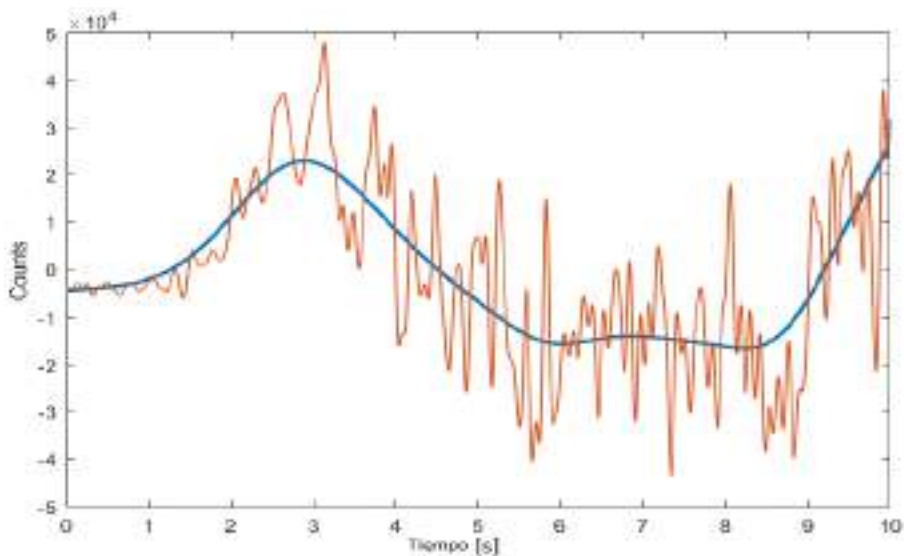


Figura 3.5 Señal de entrada y salida al filtro MCT con 2500 iteraciones. Error de interpolación de 1.0E-04.

3.2 Análisis de señales de vibraciones para detectar y clasificar fallas en rodamientos

La estructura de información MCT presentado en el capítulo anterior se utilizó para el procesamiento de señales de vibración con el objetivo de detectar y aislar fallas en rodamientos. Para esta aplicación se utilizó la base de datos de la Case Western Reserve University [<http://csegroups.case.edu/bearingdatacenter/pages/welcome-case-western-reserve-university-bearing-data-center-website>]. Las señales incorporadas en esta base fueron todas muestreadas a una frecuencia de 48 kHz. En la Figura 3.6 se muestra una señal típica del acelerómetro en condiciones normales sin defecto. La Fig. 3.7 muestra 30 ms de la señal para evidenciar el patrón de comportamiento.

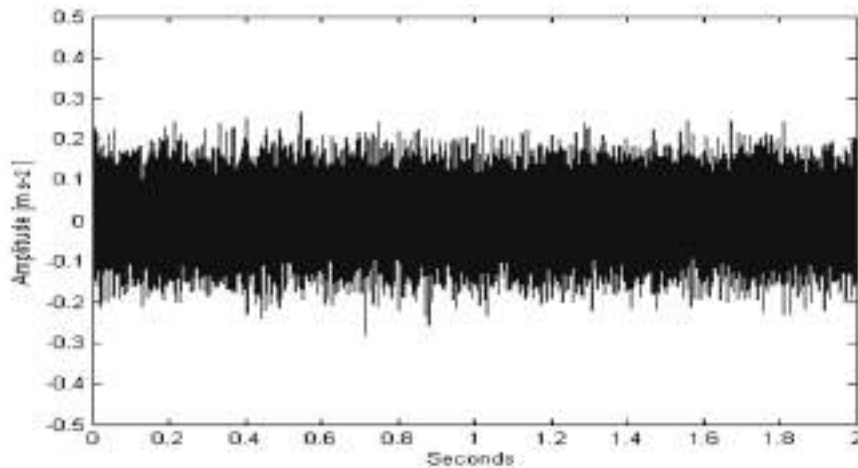


Figura 3.6. Señal de vibración del acelerómetro para un rodamiento sin defecto.

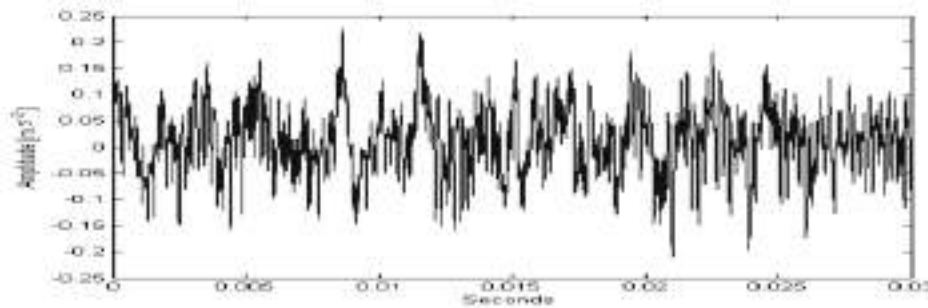


Figura 3.7. Zoom de 30 milisegundos de la señal de un rodamiento sin defecto.

Una señal de vibración sin defectos se caracteriza por no presentar un patrón predominante de comportamiento, es decir, la señal presenta un ruido aleatorio de distribución uniforme cuya única componente determinística está asociada a la velocidad de rotación. Los máximos y mínimos locales de la estructura MCT reflejan este mismo comportamiento. Por esta razón en esta condición no hay segmentos predominantes.

En la base de datos empleada, las fallas cuantificables en los rodamientos se generaron mediante mecanizado por electro-descarga. Los defectos fueron introducidos en las bolillas, en la pista interna y en la pista externa. Para esta aplicación se utilizaron las señales para diámetros de falla de 7, 14 y 21 mil (milésimas de pulgada).

En la figura 3.8 se muestra una señal obtenida con un defecto de pista interna de 7 mil, y en la figura 3.9, un zoom de 50 ms de la señal. Se puede observar que la señal tiene la apariencia de una señal portadora modulada. Al aplicar la estructura MCT a este tipo de señal, los máximos locales, se corresponden con los máximos locales de la señal envolvente que permitirán detectar y clasificar el tipo de modulación de señal.

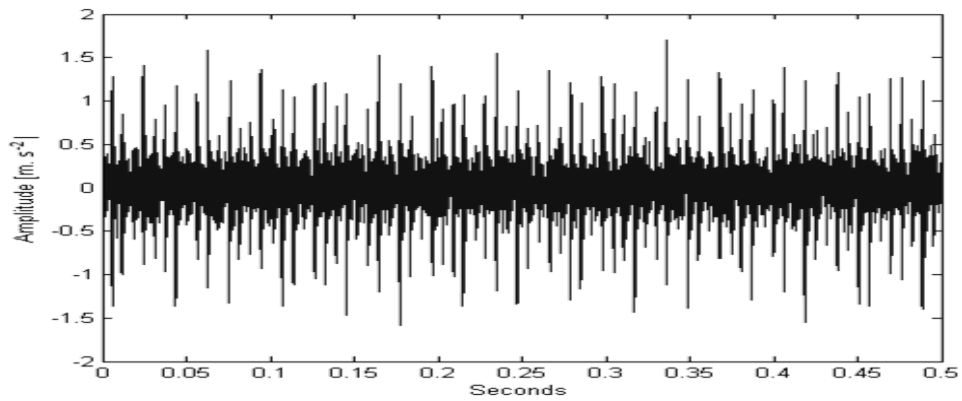


Figura 3.8. Señal de vibración para un rodamiento con un defecto de pista interno de 7 milésimas de pulgada. Frecuencia de muestreo de 48 kHz.

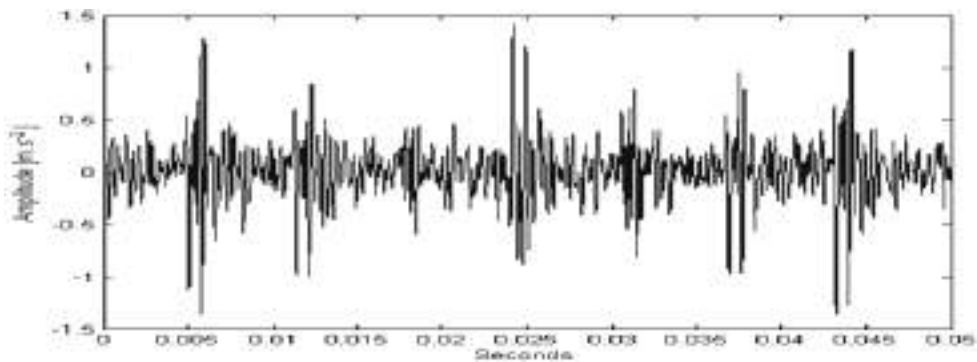


Figura 3.9 Zoom de un segmento de 50 milisegundos sobre la señal de la Fig. 3.8

La figura 3.10 muestra la envolvente de la señal de la fig. 3.9 empleando los máximos locales luego de aplicar el algoritmo MCT.

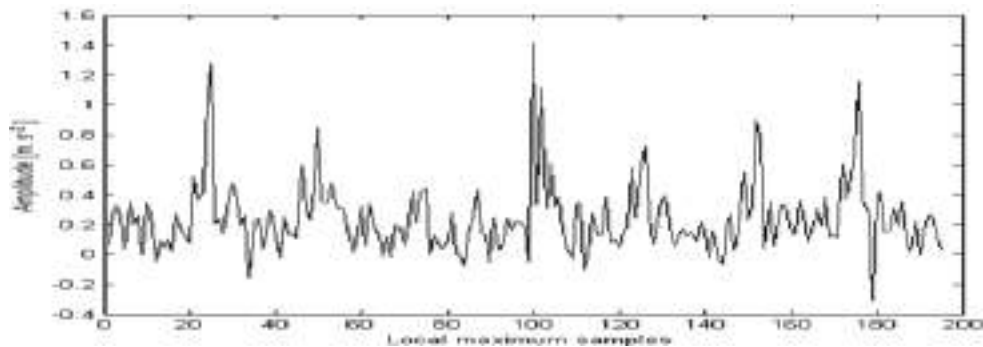


Figura 3.10 Detección de envolvente empleando los máximos locales sobre la señal de la fig. 3.9

El procedimiento basado en la estructura MCT para detectar y clasificar fallas en rodamientos se puede resumir en los siguientes pasos:

- 1- Se divide la señal en segmentos de 500 ms de señal, 24000 muestras. En aplicaciones en tiempo real, se toman 24 k muestras.
- 2- Se forma una señal con los máximos locales que llamaremos “NewSignal”. Se calcula el valor medio de la señal construida a partir de los máximos locales. Esta variable aleatoria la llamamos “MeanMaxMCT”.
- 3- Se aplica el algoritmo MCT a NewSignal y se repite el paso 2. La nueva señal generada la llamamos “NewMaxSignal” y el nuevo valor medio “MeanNewMaxMCT”.
- 4- Se calcula la varianza de NewMaxSignal. Esta nueva variable aleatoria la llamaremos VarMaxMCT.

En las Figuras 3.11 y 3.12, se muestran dos ejemplos de la clasificación obtenida. Se grafican los resultados obtenidos cada 500 milisegundos en un plano cartesiano ubicando en las abscisas la variable MeanMaxMCT y en el eje de ordenadas a VarMaxMCT. La posición de cada punto en el plano brinda información sobre el estado del rodamiento y el tipo de defecto.

La Figura 3.11 muestra la clasificación obtenida para la falla de una pista interna para 7, 14 y 21 mil. Así mismo en el mismo gráfico se presentan los resultados para un rodamiento normal.

En la Figura 3.12 se observan los puntos para un tamaño fijo de defecto, en particular 14 mils, para los diferentes tipos de defecto: de pista interna, externa y de bolillas. Al igual que en la figura anterior, se presentan los puntos para rodamientos normales.

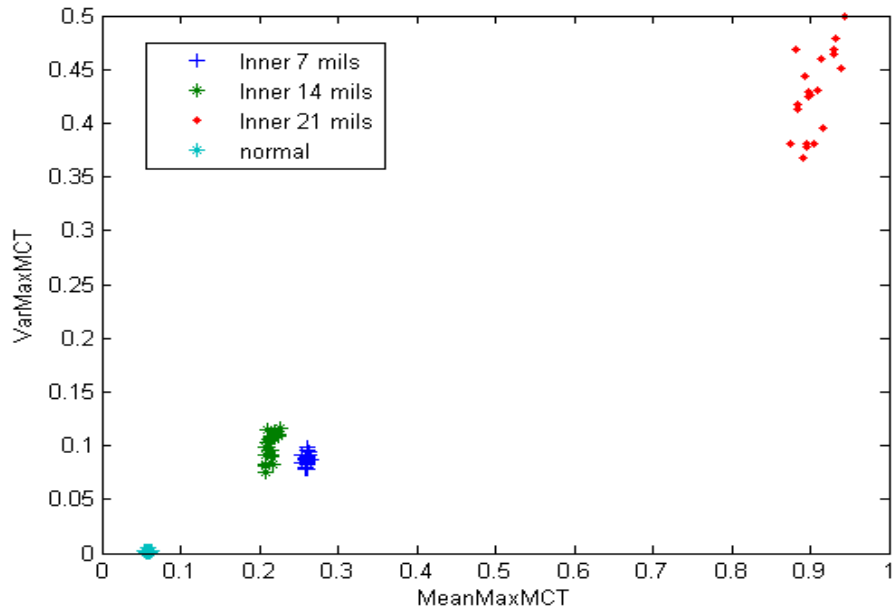


Figura 3.11. Puntos obtenidos para una falla de pista interna de 7, 14 y 21 mil, comparados con un rodamiento normal.

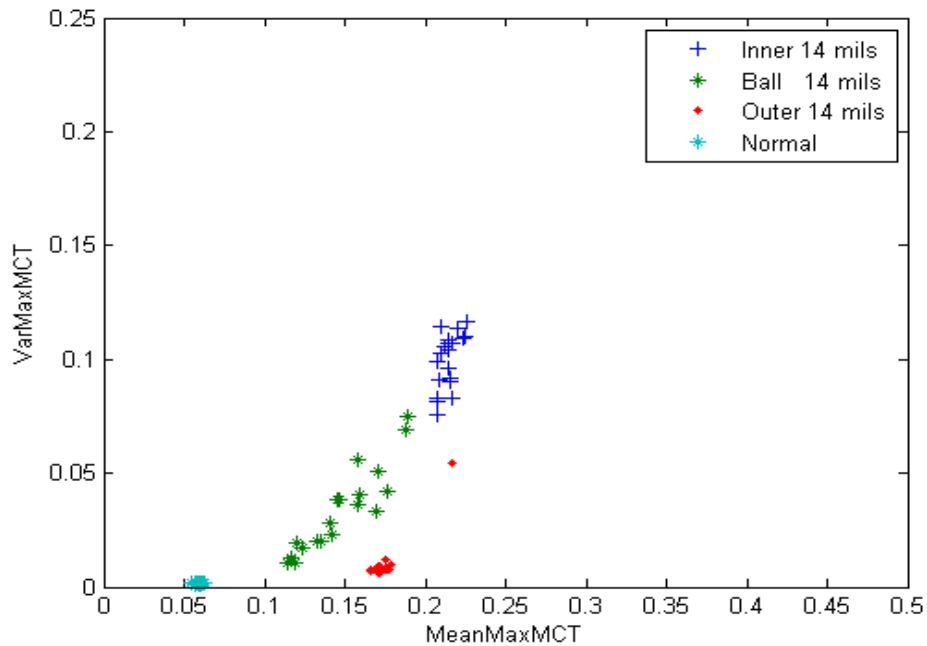


Figura 3.12. Puntos obtenidos para un tamaño de defecto de 14 mils para los distintos tipos de defecto junto a los puntos normales.

La relación entre MeanMaxMCT y VarMaxMCT proporciona información referida al tipo de defecto. La distancia al origen, el módulo, proporciona información sobre la intensidad del daño. Además, el valor MeanMaxMCT es información

correlacionada con la cantidad patrones organizados, mientras que la señal “NewMaxSignal” proporciona información acerca de la forma de la envolvente. En este caso particular se usó a la varianza como indicador.

3.3 Análisis de señales biológicas: Aplicación de la representación MCT a la detección de fibrilación ventricular en señales de electrocardiograma

Las señales eléctricas producidas por el corazón en cada latido son conocidas desde mediados del siglo XIX. En 1901, William Einthoven publicó el documento "String Galvanometer", que es considerado el primer electrocardiógrafo [Geddes L, Roeder R, 2002]. A lo largo de estos años, científicos, médicos y luego cardiólogos han aprendido a inferir el estado de salud del sistema cardiovascular analizando las formas de onda. Las señales de ECG son variantes en el tiempo. Por lo tanto, toda transformación de dominio requiere analizar la señal a través de ventanas. Esta ventana puede incluir ruido mezclado y superpuesto a las señales normales. Como resultado, la señal debe preprocesarse para descartar regiones de señales no deseadas o para procesar regiones de señales esperadas.

La fibrilación ventricular (FV) es una clase grave de arritmia que requiere una detección inmediata y correcta para salvar vidas. En condiciones normales, las actividades eléctricas en el corazón están sincronizadas. Bajo ciertas circunstancias, este sincronismo se pierde y los ventrículos entran en un estado conocido como fibrilación [Valentinuzzi M E, 2010]. Como resultado, existe un comportamiento caótico que muestra actividad no coordinada con bombeo de sangre pobre o inexistente. El colapso y la muerte cardíaca repentina seguirán en minutos a menos que la ayuda médica sea proporcionada inmediatamente. Esto requiere aplicar en el corazón un dispositivo llamado desfibrilador [<http://www.americanheart.org/presenter.jhtml?identifier=4784>]. Si se trata a tiempo, la FV puede convertirse en ritmo sinusal normal.

Los algoritmos de detección FV han sido estudiados durante más de cuatro décadas. Se ha aplicado un gran número de técnicas de procesamiento de señales

para la detección de patologías en señales de Electrocardiograma (ECG). En el dominio del tiempo, los algoritmos más utilizados son el algoritmo de intervalos de cruce de umbral (ICT) [Thakor N V, Zhu Y S, and Pan K Y, 1990] y el algoritmo de Tompkins [Pan J and Tompkins W J, 1985]. En el ICT las decisiones se basan en el número y la posición de los cruces de señales a través de un determinado umbral. El algoritmo de Tompkins se basa en la detección del complejo QRS utilizando información de la pendiente, la amplitud y el ancho de éste. También existen algoritmos de autocorrelación que utilizan las periodicidades dentro del ECG para inferir parámetros de la señal [Chen S, Thakor N, and Mower M, 1987]. Por otra parte, en el dominio de la frecuencia, el algoritmo espectral (SPEC) [Barro S. et al, 1989] analiza el cambio de patrones durante FV en bandas de frecuencia mediante el análisis de Fourier. La transformada wavelet [Szilagy S., et al, 2003] y más recientemente la transformada de Hilbert [Amann A. et al, 2005], ambas se han propuesto para dar solución a este problema. Técnicas de inteligencia artificial como redes neuronales [Clayton R et al, 1994] y algoritmos genéticos [Poli R et al, 1995], se han utilizado con buenos resultados, pero requieren etapas de entrenamiento.

Cada uno de los métodos tiene ventajas y desventajas. Las técnicas aplicadas en el dominio del tiempo son más eficientes que otras, pero son muy sensibles al ruido presente en toda señal biológica. Las transformaciones de señal a otros dominios logran la detección de patrones específicos, pero implican un alto costo computacional y pérdida de precisión temporal.

La representación de esta tesis MCT, para el procesamiento de señales de ECG es simple, eficiente y no requiere ningún tipo de procesamiento previo. Además, al ser procesada en el dominio temporal se detectan exactamente la ocurrencia de los eventos esperados. Como se demuestra a continuación estos algoritmos pueden usarse para clasificar patrones normales de anormales y detectar patologías o simplemente marcar regiones para un procesamiento posterior. Estas técnicas se pueden aplicar a cualquier señal, como resultado conforman una plataforma homogénea para el análisis.

Las señales biomédicas representan un desafío para las técnicas de procesamiento de señales digitales. En primer lugar, la relación señal / ruido es generalmente pobre. En segundo lugar, aparecen señales no deseadas como el ruido electromiográfico, deriva de la línea de base, interferencia de la línea eléctrica, interferencia de la radiofrecuencia y ruido del interfaz de piel-electrodo. Tales interferencias pueden degradar las señales de ECG, por lo tanto, el algoritmo debe ser robusto y tener la capacidad de detectar estas señales evitando regiones ruidosas dado que la interferencia es siempre excesivamente alta en este tipo de señales. La figura 3.13 muestra una señal ECG ideal simplificada, donde las características importantes son la forma, la posición relativa y la duración de las ondas esperadas. La figura 3.14 muestra una forma de onda de ECG real después de ser filtrada y muestreada a la frecuencia de 128 Hz.

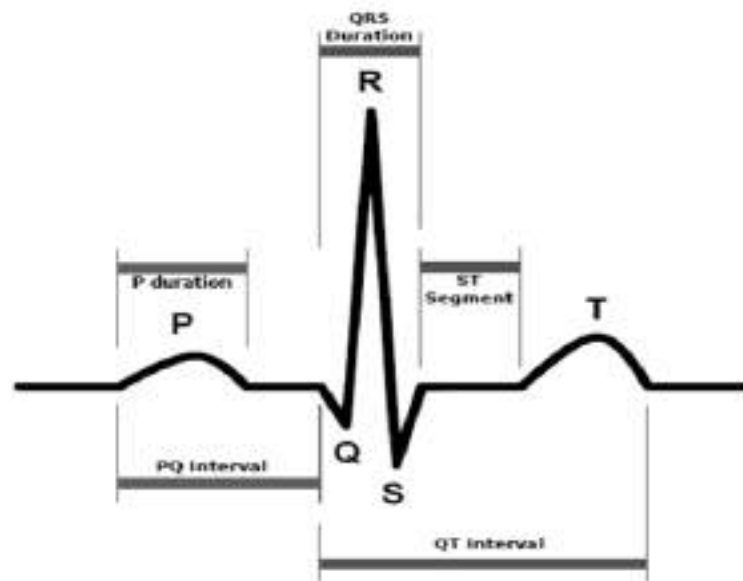


Figura 3.13. Forma ideal de la señal ECG.

Para la experimentación con señales reales y validación de los algoritmos propuestos se trabajó con el banco de señales PhysioNet, disponible en:

[<http://www.physionet.org/physiobank/database/mitdb>] Massachusetts Institute of Technology, MIT-BIH arrhythmia database.

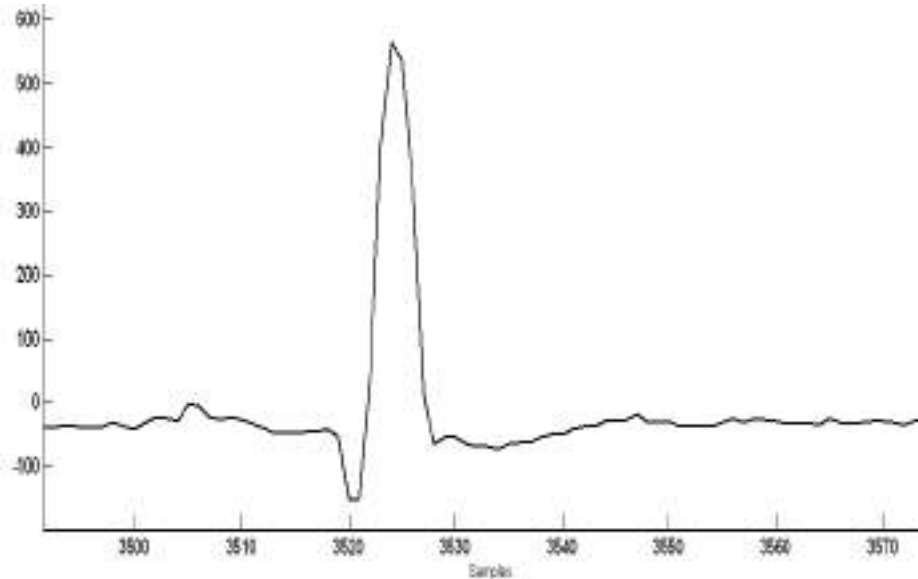


Figura 3.14. Forma de onda real ECG. MIT-BIH Normal Sinus Rhythm Database record N° 16265, base de datos PhysioNet.

Una de las características útiles para detectar el comportamiento caótico de la señal de FV es la falta de regularidad en el intervalo de tiempo entre un pico R y el siguiente R (PeriodoRR). Una característica adicional es el incremento en la frecuencia del ritmo cardíaco más allá de los límites normales. En los episodios de taquicardia ventricular (TV), la frecuencia de latidos cardíacos es alta, pero hay regularidad en el período RR. Dado que el intervalo de tiempo RR es una variable aleatoria, usaremos la desviación estándar del tiempo RR obtenido mediante la estructura MCT, como otro parámetro para detectar la posible presencia de FV. Por lo tanto, es necesaria una detección precisa del conjunto de ondas que representan la despolarización de los ventrículos, también conocido como complejo QRS.

En la siguiente sección se presenta un algoritmo basado en estructura MCT para la detección del complejo QRS.

3.3.1 Detección de patrones QRS

El algoritmo utiliza la información proporcionada por los vectores pertenecientes a la estructura MCT. El vector clase (n) de la estructura MCT contiene los máximos y mínimos de la señal segmentada. La Fig. 3.15 muestra un complejo QRS real y la salida de los procesos de segmentación y etiquetado. La línea roja es la concatenación lineal de las muestras etiquetadas. Esta concatenación no es parte del proceso, sólo se muestra para observar los segmentos y su capacidad de representar la señal real.

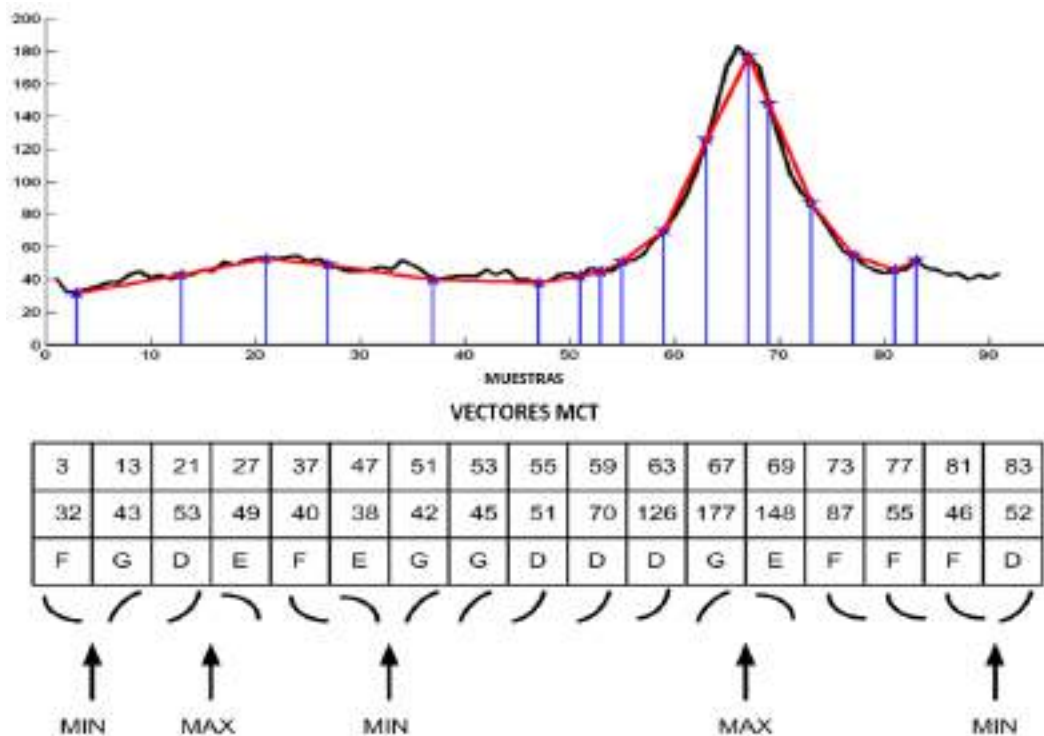


Fig. 3.15 Algoritmo MCT aplicado a una señal real ECG (426m.dat from MIT-BIH Malignant Ventricular Arrhythmia database, error de interpolación error=3, Fs=250 Hz. Parte inferior, los vectores MCT destacando los máximos y mínimos.

La detección del complejo QRS comienza detectando secuencias de min-max-min. A su vez, estas secuencias deben cumplir cinco condiciones para calificar como un complejo QRS normal:

- 1- La sección de elevación debe ser segmento dominante "d".
- 2- La sección descendente debe ser segmento dominante "f".
- 3- La diferencia de los valores mínimos debe ser una fracción de la diferencia entre el valor máximo y el promedio de los valores mínimos.
- 4- La diferencia entre la altura y la base debe ser mayor que un valor preestablecido.
- 5- La energía del impulso, estimada como el área de un triángulo debe estar dentro de un rango.

La Figura 3.16 muestra un patrón candidato para ser analizado, condición de la secuencia min-max-min.

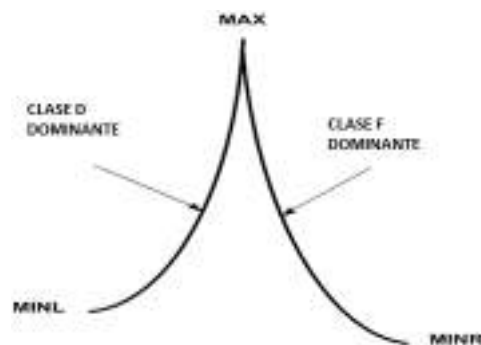


Figura 3.16. Patrón candidato QRS.

La Figura 3.17 muestra los resultados experimentales de la detección del complejo QRS para una señal de la base de datos. Luego de ajustar los valores correspondientes a las 5 condiciones anteriores, se pudo comprobar visualmente

que el algoritmo propuesto tiene una excelente detección de los patrones normales, incluso con deriva de línea y ruido presente.

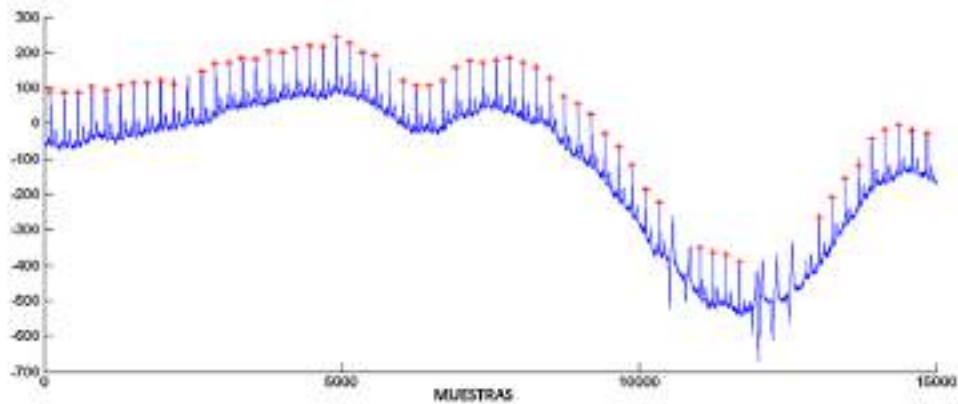


Figura 3.17. Detección QRS (+) aplicada a la señal 426m.dat de la base de datos: MIT-BIH Malignant Ventricular Arrhythmia database, error de interpolación=3, $F_s=250$ Hz.

Cada vez que el algoritmo no detecta un complejo QRS es posible estar en presencia de otras patologías. Por ejemplo, la figura 3.18 muestra un zoom de la parte inferior de la figura 3.17 donde el complejo QRS no es detectado debido a la diferencia excesiva entre los valores mínimos que podrían indicar la contracción ventricular prematura (CVP).

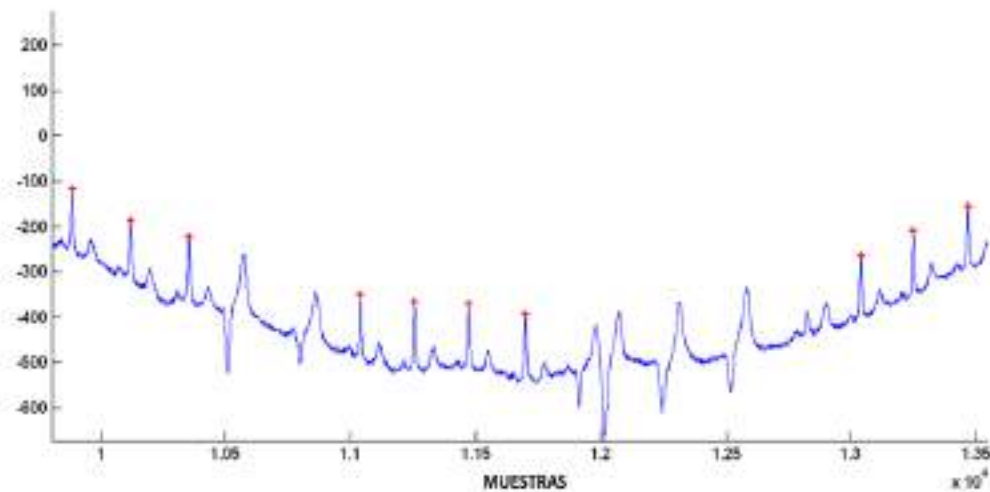


Figura 3.18. Zoom de la parte baja de la figura 3.17 para resaltar los patrones descartados.

La Figura 3.19 muestra los resultados de la aplicación del algoritmo del punto anterior a una señal ECG con FV. Se puede observar el cambio de patrón que muestra el incremento del contenido de frecuencia y una forma irregular.

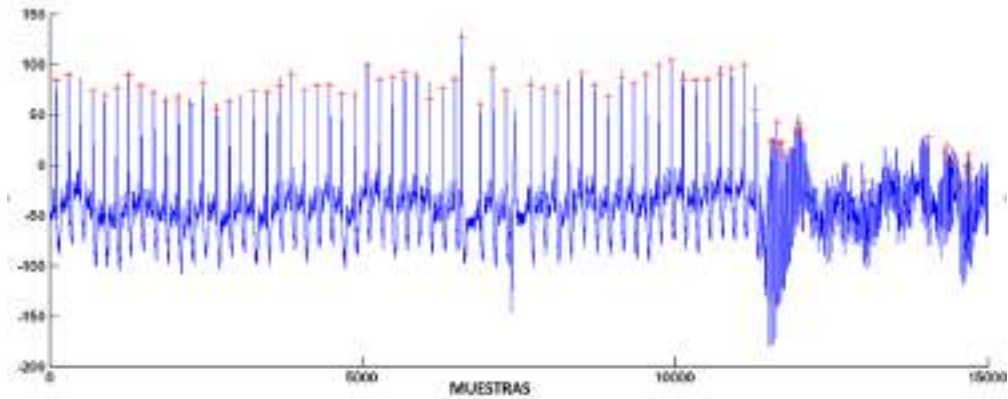


Figura 3.19. Transición a un episodio VF, de la base de datos MIT-BIH Malignant Ventricular Arrhythmia database, error de interpolación=3, $F_s=250$ Hz, Record 426.

La Figura 3.20 muestra la frecuencia de latido cardíaco estimada por el tiempo transcurrido entre complejos QRS detectados. Durante el episodio de la FV, la detección del complejo QRS sigue el comportamiento caótico de la señal. Si la FV persiste, la detección tiende a cero por la imposibilidad de generar contracciones.



Figura 3.20 Frecuencia instantánea en latidos por minuto para la señal de la figura 3.19.

En la figura 3.21 se muestra un diagrama en bloques del algoritmo propuesto para la detección de la FV. Hay dos ventanas temporales, una para el episodio FV y otra para el estado normal. Una vez que se ha detectado una frecuencia en un rango de 200-500 BPM el algoritmo adopta el estado temporal de ventana de FV, permaneciendo en estado de alerta. En este estado, el episodio de la FV se confirma a partir de la evaluación de dos parámetros. En primer lugar, la varianza estimada del latido cardiaco debe ser alta o el ritmo cardiaco debe tender a cero debido a que el algoritmo no detecta pulsos. Si no se cumple ninguna de estas condiciones, se abre la ventana temporal para volver al estado normal. Estas ventanas están en el rango de 2 a 5 segundos.

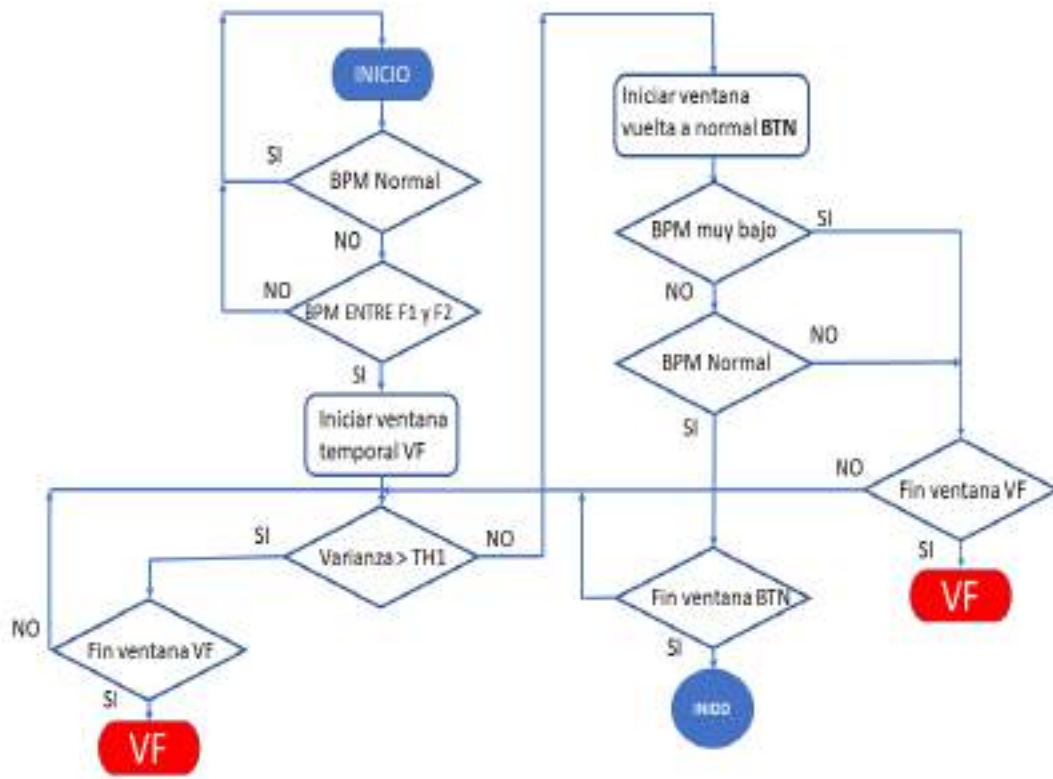


Figura 3.21. Algoritmo para la detección de VF (fibrilación Ventricular)

3.3.2 Conclusiones

Se utilizó exitosamente la estructura MCT propuesta en esta tesis para analizar señales de ECG. En esta sección, se enfocó el procesamiento en la detección de fibrilación ventricular. Estos algoritmos pueden ser vistos como un proceso de muestreo completo seguido de un tratamiento morfológico de la señal que podría ser utilizado para el análisis de otras patologías cardíacas, ya que clasifica los patrones QRS como normal / anormal. A diferencia de los métodos propuestos en la bibliografía para resolver este tipo de problemas, las señales se procesaron sin ningún tipo de prefiltrado.

3.4 Determinación de la velocidad de un móvil procesando el sonido pasante

3.4.1 introducción

Bajo el concepto de ciudades inteligentes, planificación urbana y semáforos inteligentes, un dato sumamente importante es la densidad de tránsito en tiempo real. Con esta información es posible mejorar los flujos vehiculares actuando sobre los tiempos de habilitación de paso de semáforos, realizar desvíos automáticos y realizar estadísticas de la dinámica de la densidad vehicular en función de día y hora.

Los centros urbanos se encuentran superpoblados, lo que requiere una constante planificación para mejorar las condiciones de desplazamiento en vehículos. Esta aplicación propone medir el flujo vehicular y estimar la velocidad de un móvil mediante el ruido captado por un micrófono empleando la estructura MCT.

3.4.2 Características de las señales sonoras

La Fig.3.22 muestra una captura del sonido de un móvil a 60 km/h. Se observa la forma de la envolvente que posee la información de la velocidad. Se desprende de la Figura 3.22 que la estimación de la velocidad no es un proceso sencillo por dos motivos; el ruido propio del ambiente y la naturaleza ruidosa del crecimiento de la señal a medida que el móvil se aproxima al micrófono, como se aprecia en el zoom de la Figura 3.23.

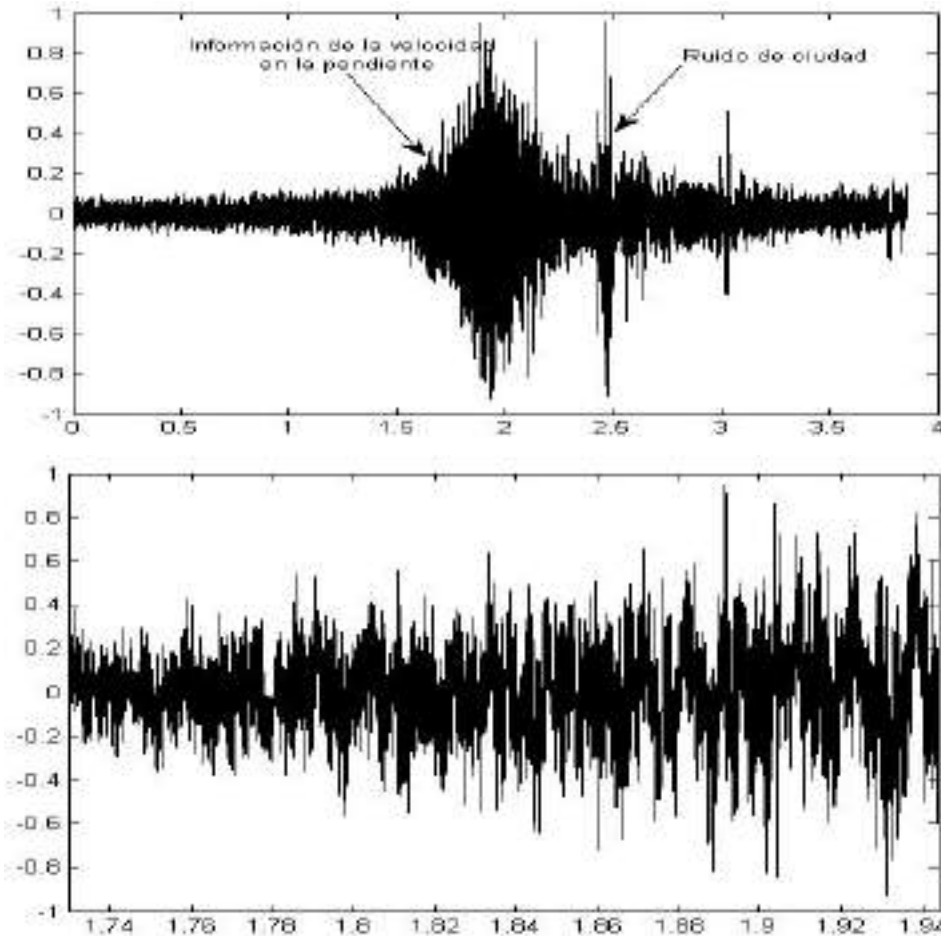


Figura 3.22 Superior: Captura de un móvil a 60 km/h. Inferior: Zoom de la parte creciente de la información de velocidad. Frecuencia de muestreo: 44100 Hz.

El crecimiento de la intensidad sonora tiene una forma exponencial y el contenido espectral es amplio y uniforme. A continuación, se describen los algoritmos empleados para la estimación del pasaje de un móvil y su velocidad.

3.4.3 Algoritmo detector de envolvente

El primer procesamiento es determinar la envolvente ya que en su forma se encuentra la información de aproximación del vehículo. El algoritmo propuesto es el siguiente:

- Sobremuestrear la señal.
- Cada T_i segundos:
- Aplicar muestreo MCT con error de interpolación bajo, del orden del 0.1%.
- Obtener los máximos locales.
- Calcular como salida, el valor promedio de los máximos.

En los resultados experimentales mostrados en esta tesis se emplearon los siguientes parámetros: $F_s=44100$ kHz, $T_i=20$ ms. En la Fig. 3.23 se muestra el resultado aplicado sobre una señal real.

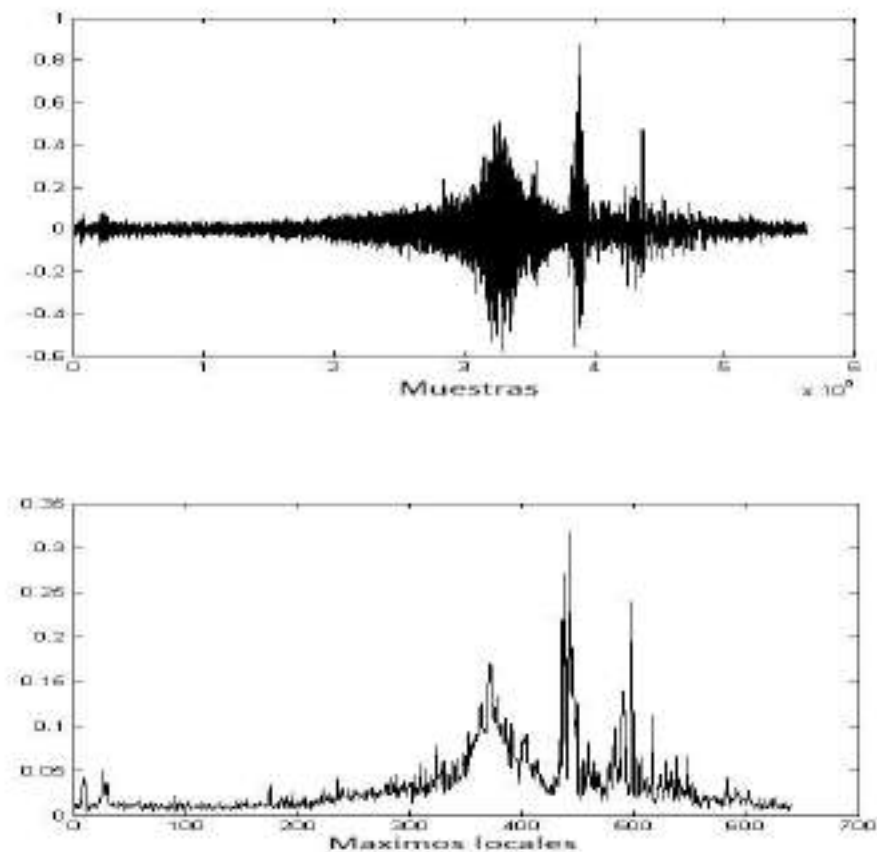


Figura 3.23. Superior señal de un móvil a 65 Km/h. Inferior envolvente empleando los máximos locales del proceso MCT según el algoritmo del punto 3.4.3.

Se observa en la Fig. 3.23 que se captura la información de envolvente, pero es muy sensible al ruido en amplitud.

Nótese que por cada 882 muestras se obtiene un resultado. En la Fig. 3.23, 600 valores máximos promedio representan aproximadamente 5 segundos de señal. Dos observaciones importantes. La forma de la envolvente es capturada, pero el algoritmo es sensible al ruido en amplitud. Además, la envolvente es todavía ruidosa como para calcular la pendiente.

3.4.4 Algoritmo discriminador de sonido

Se buscó un algoritmo basado en la estructura MCT que sea sensible a las características del sonido generado por un móvil. Se eligió como variable aleatoria a la cantidad de máximos locales por unidad de tiempo, un estimador del contenido espectral. Se ensayó el algoritmo MCT, pero con un error grande de interpolación de manera tal que, energía espectral pero con pequeña amplitud no genere una salida apreciable al disminuir la cantidad de máximos detectados. Por lo tanto, son minimizados segmentos de señal en donde la cantidad de máximos con alto valor de señal es baja y en donde la cantidad de máximos es alta, pero con intensidad baja.

Algoritmo discriminador sonido móvil:

- Sobremuestrear la señal.
- Cada T_i segundos:
- Aplicar muestreo MCT con error de interpolación alto, del orden del 2%.
- Obtener los máximos locales.
- Calcular como salida, la cantidad de los máximos hallados.

En la Fig. 3.24 se muestra la respuesta del algoritmo a la señal de la Fig. 3.25.

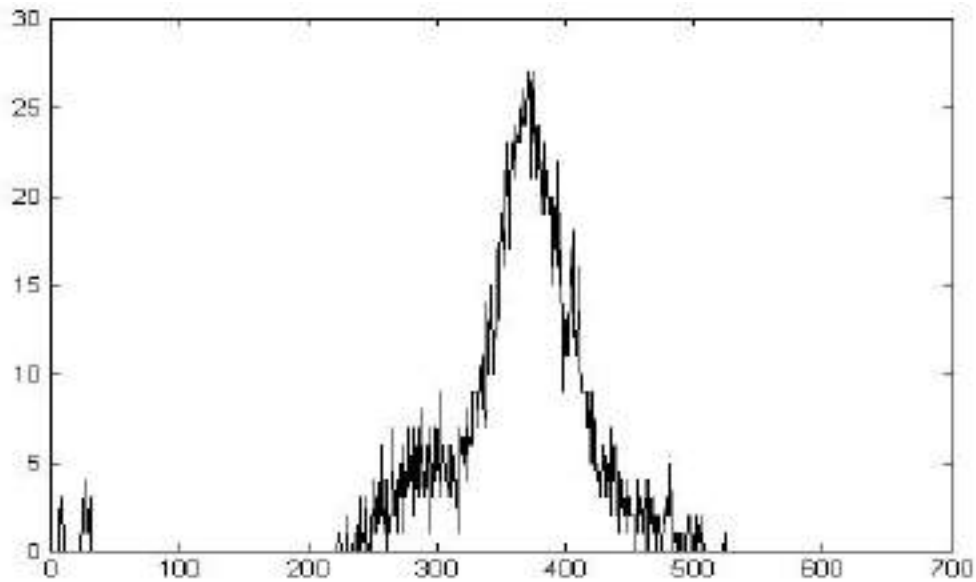


Figura 3.24 Salida del algoritmo discriminador de sonido de móvil aplicado a la señal de la Fig. 3.23.

El algoritmo discriminador elimina señales no deseadas, entre las más importantes; ruido provocado por el viento y ruidos típicos de ciudad.

3.4.5 Combinación de los dos algoritmos y filtrado

Se ensayaron diferentes propuestas para combinar los dos algoritmos previos, la más simple y efectiva fue la de multiplicar punto a punto los resultados de los dos algoritmos. De esta forma se enfatizan los segmentos en donde hay amplitud de envolvente con características de ruido tipo móvil. La Fig. 3.25 muestra la señal resultante del producto correspondiente a la señal de la Fig. 3.23.

El proceso restante es filtrar la señal para eliminar ruido antes de calcular la pendiente. Los filtros lineales pasabajos corrompen la pendiente de la señal y por lo tanto destruyen la información útil.

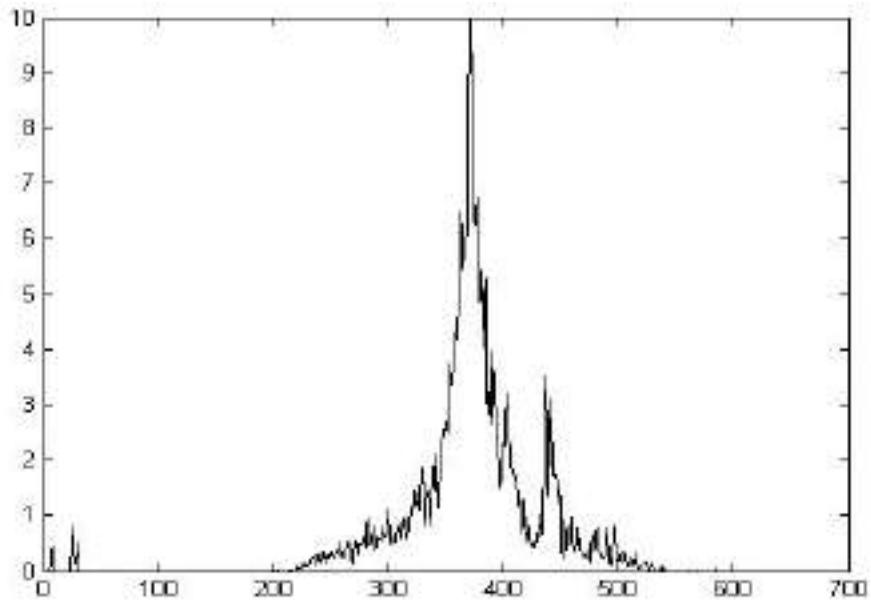


Figura 3.25 Señal resultante del producto de las señales de los dos algoritmos.

En la Fig. 3.26 se observa la señal de entrada y salida de un filtro de promedio móvil de una ventana de 50 muestras en donde se observa el cambio de la pendiente de la señal de salida, lo cual resulta inaceptable.

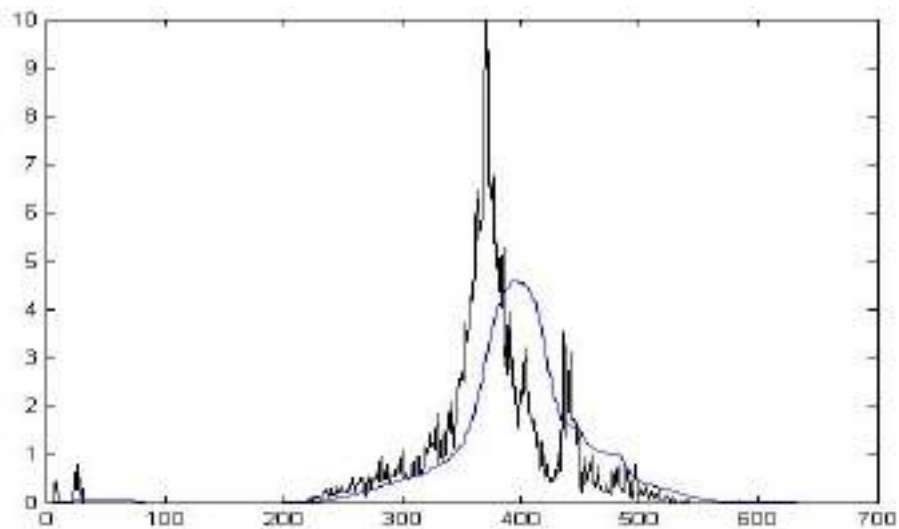


Figura 3.26 Señal de entrada y salida de un filtro de promedio móvil de 50 muestras. Nótese la disminución de la pendiente de la señal de salida.

El filtro de mediana, que es simple de implementar, resulta ideal para esta aplicación. Su principal virtud es lograr un filtrado pasabajos sin afectar las pendientes de las transiciones.

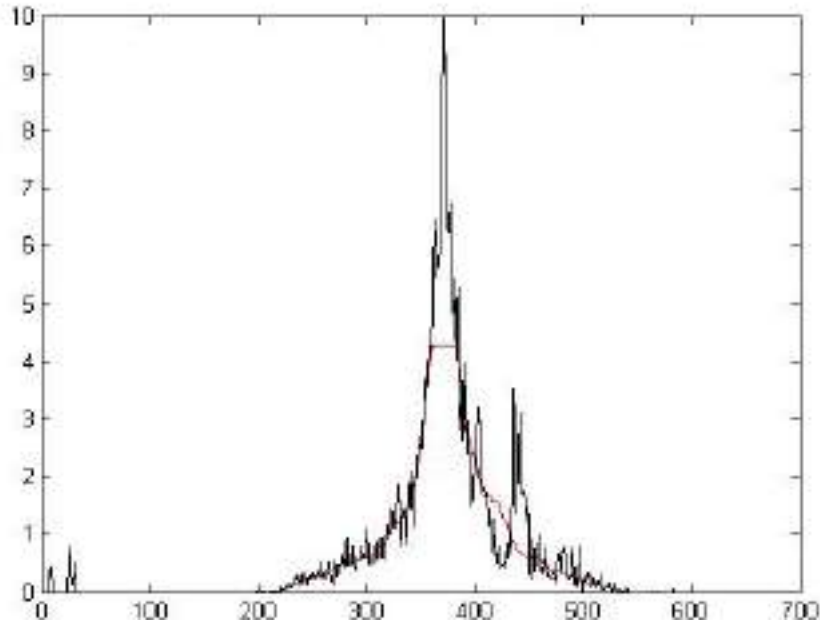


Figura 3.27 Señal de entrada y salida de un filtro de mediana, ventana=50 muestras. Nótese la preservación de la pendiente.

En la Fig. 3.27 se observa que la forma de la punta de la señal de entrada es destruida por los dos tipos de filtros, pero el filtro de mediana mantiene intacta la pendiente.

3.4.6 Filtrado MCT y análisis de la pendiente

El proceso de muestreo MCT permite reconstituir, empleando la información de los tres vectores, la señal entre las muestras esenciales. Dado que la señal es reconstituida empleando trayectorias simplificadas, se pierde información en alta frecuencia, como se detalla en la sección 3.1. La señal puede volver a muestrearse iterando el proceso un número arbitrario de veces. La señal de salida del filtrado mediano es iterada con el muestreo MCT un número N de veces hasta lograr una suavidad en la trayectoria que es inferida por la secuencia de clases. La Fig. 3.28

muestra la respuesta del filtro mediana y la salida de filtro de iteración MCT para cinco iteraciones.

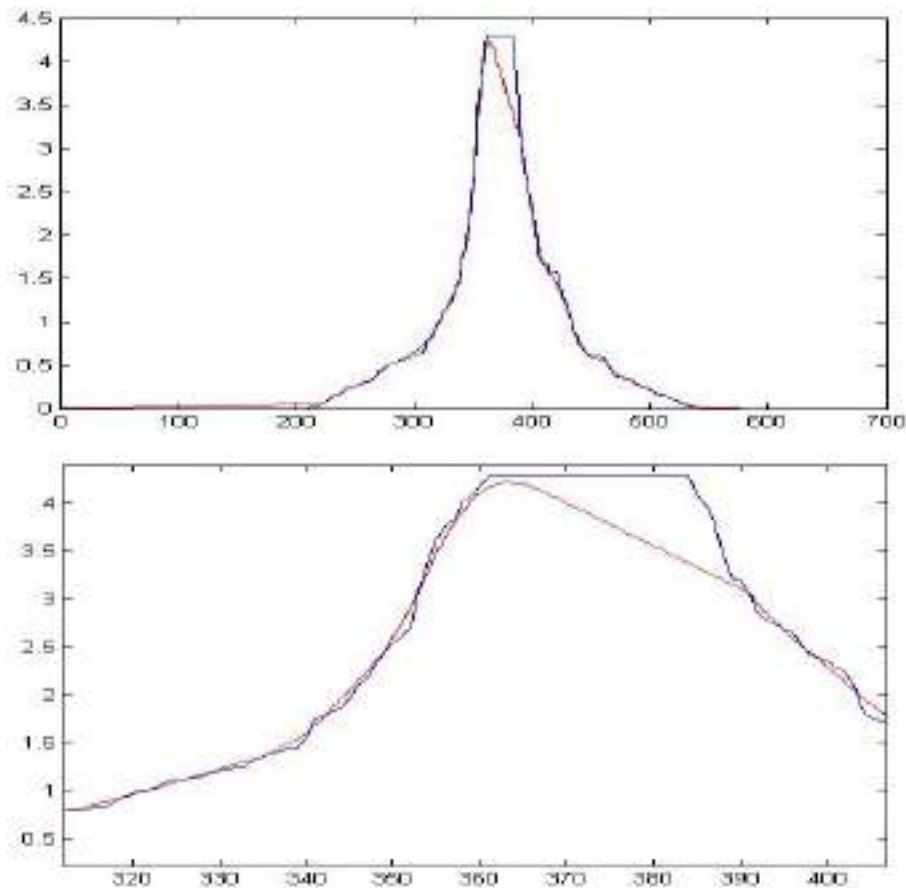


Figura 3.28. Superior: Señal de salida del filtro mediana filtrada con el filtro MCT. Inferior: zoom de la pendiente de subida. Filtro MCT con reconstrucción por interpolación lineal, cantidad de iteraciones $N=5$, $\text{error}=0.1\%$.

Se comprueba empíricamente que el filtrado por iteración MCT no destruye la pendiente de la señal para un error de interpolación menor al 0.1% del rango de la señal. La salida del filtro MCT es analizada para determinar la pendiente. La secuencia de clases de la señal de la Fig. 3.28 es:

Clase=[d,d,d,d,g,g,d,d,g,d,d,d,d,d,g,g,g,g,e,e,ef,f,e,f,f,e,f,f,f]

La unión de los segmentos “ge” determina la posición del máximo. La pendiente es calculada empleado los segmentos tipo “d” previos a los del tipo “g”.

3.4.7 Resultados experimentales

Se realizaron capturas de sonidos de vehículos a velocidades entre 40 y 70 km/h, empleando diferentes móviles. Los datos fueron capturados a una frecuencia de muestreo de 44100 KHz, empleando directamente la placa de audio estándar de computadoras portátiles. El micrófono empleado fue del tipo *Electret* de bajo costo.

Todos los algoritmos ensayados se basan en la estructura de muestreo MCT, salvo el filtro de mediana. La función del filtro de mediana es la de eliminar ruido que tiene las características de un móvil pasante, pero que no se sostiene el tiempo suficiente para ser considerado un móvil. La combinación de los algoritmos propuestos proporciona una detección muy confiable.

En la Fig. 3.29 se observan valores de pendiente hallados en función de distintas velocidades en donde la pendiente fue calculada empleando el promedio de los segmentos tipo “d” previos al primer segmento tipo “g” que está involucrado en el máximo. Es importante remarcar que la unión de un segmento “d” con uno “g” es un punto de inflexión.

Los segmentos tipo “g” previos al máximo son consecuencia del filtrado MCT en la zona en donde el filtro de mediana ha producido un recorte abrupto. Todos los algoritmos son aptos para ser implementados en microcontroladores de uso general de bajo costo.

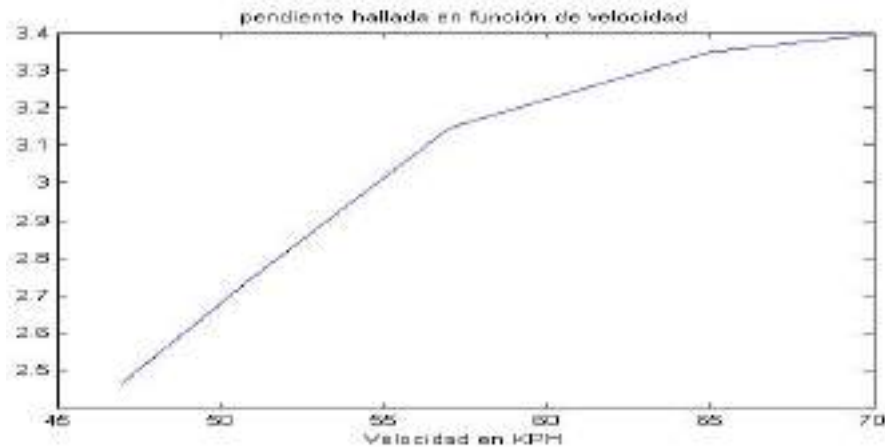


Figura 3.29. Pendiente hallada del perfil sonoro en función de la velocidad. La pendiente es calculada como el promedio de los segmentos tipo “d” previos al máximo en el punto de inflexión “dg”.

En la Fig. 3.30 se observa 40 segundos de captura en la cual pasan dos vehículos en la primera mitad y la señal de salida del algoritmo completo. El resto de la señal es ruido provocado por el viento y otros tipos de sonidos comunes en una ciudad. Se observa que el ruido es prácticamente eliminado. Los picos por encima de un umbral son considerados como vehículos pasantes.

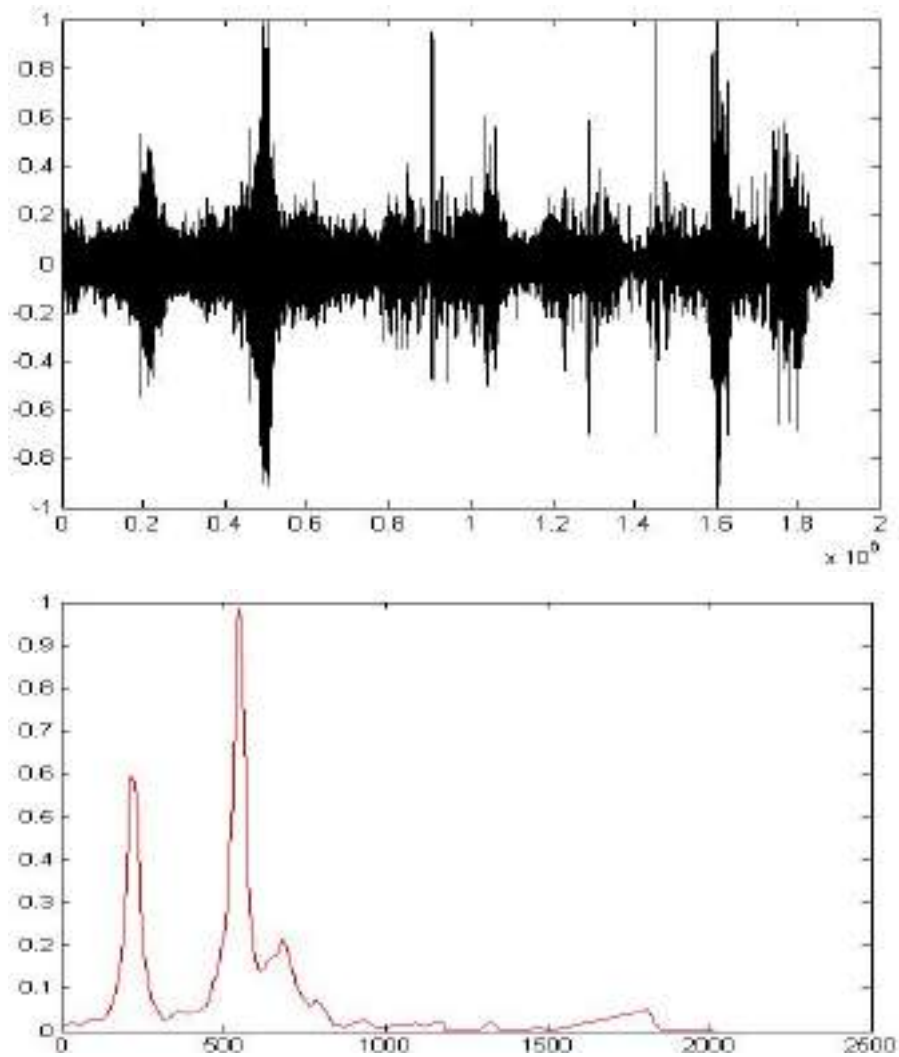


Figura 3.30. Captura de 40 segundos. En los primeros 10 segundos pasan dos móviles distintos. El resto de la señal es ruido provocado por el viento e interferencias propias de una ciudad.

Los resultados presentados fueron procesados en computadoras personales empleando software de simulación y empleando capturas de señales reales.

3.4.8 Conclusiones

La congestión en los centros urbanos será uno de los grandes desafíos que tendrá que enfrentar el hombre en los próximos años. Se requerirá de soluciones realmente inteligentes que exceden largamente a simples semáforos. La implementación de soluciones tecnológicas se ven afectadas por dos principales problemas, el costo y el consumo energético. El procesamiento por sonido tiene la fortaleza que es de muy bajo costo y que el sensado es totalmente pasivo, lo que permite implementaciones portátiles o de bajo consumo.

Los sensores inteligentes basados en procesamiento de video, las rutas y los móviles inteligentes son los paradigmas que seguramente encontrarán soluciones a esta problemática. Mientras tanto, los algoritmos propuestos permiten sumar inteligencia a un semáforo que controla una intersección que tiene una densidad vehicular que varía en un mismo día desde 100 autos/hora a 1200 autos/hora y mantiene los mismos parámetros operativos para las dos condiciones extremas.

4 APLICACIONES EN IMÁGENES Y VIDEO

RESUMEN

En este capítulo se muestran aplicaciones sobre imágenes y videos de la extensión a datos n-dimensionales de la representación MCT propuesta en esta tesis. Para esta extensión, una imagen es considerada como un arreglo independiente de señales unidimensionales fila o columna. Esta representación permite obtener aplicaciones en tiempo real factibles de ser implementadas en microcontroladores de bajo costo.

Junto a una descripción completa de cada aplicación abordada en esta tesis, se presentan los resultados para imágenes sintéticas y reales en distintas condiciones ambientales a fin de validar los algoritmos propuestos, destacando la bondad de la estructura de la información MCT para la inferencia en tiempo real de conocimiento.

4.1 Aplicación en video: Estimación en tiempo real de la posición de un vehículo en ruta.

4.1.1 Introducción

Los sensores proporcionan la interfase de los sistemas con el mundo real. Ellos son cada vez más inteligentes ya que incorporan inteligencia en el punto de medición. Los microcontroladores “*all inclusive*” nos ofrecen todo lo que necesitamos para convertir a un sensor en un transductor inteligente. Además de esta migración de tecnología hacia el punto de medición, la potencia de procesamiento disponible en la actualidad ha permitido que otros elementos asuman el rol de sensores al mutar su objetivo. Este es el caso de las cámaras de video. La secuencia de imágenes de un video proporciona tanta información que la salida no es más una imagen, sino

información multisensorial. Mediante procesamiento de imágenes se reconocen objetos, rostros, personas, gestos, movimientos, colores, texturas y todo lo que permita la riqueza de la información presente en una imagen. El video agrega la dimensión temporal a las imágenes resultando en una fuente de información de naturaleza diversa.

De los datos suministrados por una cámara de video enfocada en lo que ve el conductor de un vehículo en una ruta se pueden obtener diferentes clases de información. En primer lugar, promediando el nivel de intensidad de los píxeles de la parte superior de la imagen se podría determinar si la conducción es diurna, nocturna o bajo condiciones de baja visibilidad. Además, detectando líneas en las imágenes es posible estimar el punto de fuga de la perspectiva, a partir del cual se determina el patrón de manejo y se estima la posición del móvil dentro de la ruta. Por otro lado, observando la evolución temporal de los píxeles esenciales de la representación MCT de la imagen, se puede estimar la velocidad del vehículo y obtener información del tiempo de manejo del conductor sin detenerse. Otra información importante se obtiene a partir del desplazamiento del punto de fuga en una dirección perpendicular al desplazamiento del vehículo. Esta información proporciona la irregularidad de la ruta que fusionado con la información de velocidad podría indicar una situación de peligro. Finalmente, la suciedad del parabrisas se puede estimar mediante el promedio a largo plazo de la secuencia de imágenes dada la persistencia de las manchas.

La Figura.4.1 resume las principales extracciones de conocimiento a las cuales se les pueden agregar otras dependiendo de la aplicación real como por ejemplo estimación de niebla, observando la falta de transiciones abruptas en la imagen. Además, las estimaciones se pueden fusionar para obtener nueva información.

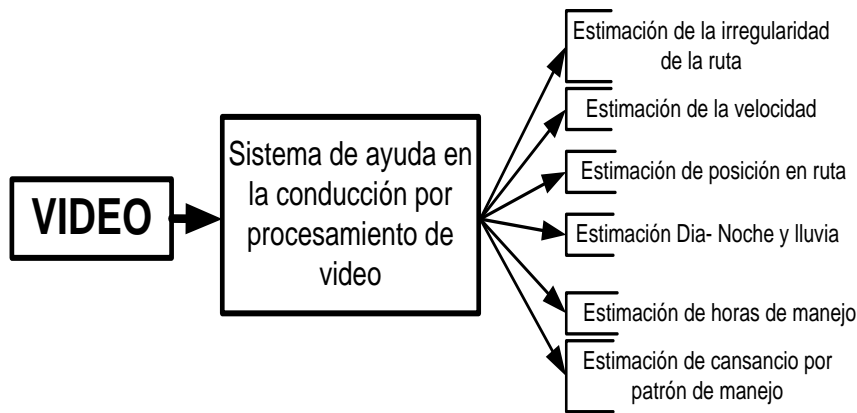


Figura 4.1. Información multisensorial obtenida del procesamiento de video.

La riqueza de la información presente en un video tiene como contrapartida la desventaja de un alto costo de procesamiento. Por otro lado, el análisis de imágenes del exterior en entornos no estructurados es también un desafío, especialmente para sistemas en tiempo real. La variabilidad extrema de las condiciones encontradas en una ruta sumado a la excesiva información a procesar, pueden hacer inviable un desarrollo a costos razonables.

Las estadísticas revelan que por lo menos el 25% de los accidentes en ruta se deben a distracciones [Governors Highway Safety Association, 2006], aunque se intuye que ese porcentaje es aún mayor dado que no siempre el conductor asume su error. Entre las distracciones más comunes se pueden mencionar el uso del teléfono celular o dispositivos electrónicos, GPS, cambio de estación de la radio, el mate, los carteles y otros eventos anormales. Si a estos elementos le sumamos el cansancio, los microsueños, y las largas y monótonas rutas, tan comunes en Argentina, este porcentaje se incrementa notablemente.

La detección de líneas en una imagen es un proceso importante hacia el entendimiento de la escena. La transformada de Hough es un método ampliamente difundido para la detección de líneas y curvas. La excesiva carga computacional que implica ha impulsado la investigación de mejoras de la transformada para aplicaciones en tiempo real [D. Ben-Tzvi and M.B. Sandler, 1990], [Kalviainen H, et al,1995]. Se emplea intensivamente en detección de líneas para conducción autónoma, ver por ejemplo [G.T. Fan, B. Li, Q. Han, et al,2020], donde la

transformada es acotada en ángulo y distancia dentro de la ROI (*Region of Interest*).

Un proceso sumamente sensible y que es un factor común en el estado del arte, es el proceso de binarización de la imagen. La elección del umbral no es un proceso simple debido a las condiciones cambiantes de un ambiente exterior. Se emplean técnicas de umbrales adaptivos en función de estadísticas locales [Cheng H Y, Tseng P T, et al,2006], también hay desarrollos con umbrales empleando Fuzzy K means como por ejemplo en [Toan Minh Hoang et al, 2017].

La estructura de la información presentada en esta tesis permite diseñar algoritmos eficientes a partir de la representación MCT para la asistencia en la conducción en ruta, viables y a bajo costo. No es necesario el proceso de binarización ya que los puntos notables de la imagen se adaptan automáticamente a las condiciones de iluminación.

La detección de líneas resulta más importante aún ya que el escenario previsto es una vista en perspectiva en la cual los objetos se alinean naturalmente hacia el punto de fuga. En una ruta sin marcar, los bordes también generan puntos alineados. Empleando la información de píxeles esenciales se puede estimar la alineación en la imagen sin la detección de líneas marcadas en la ruta.

Para esta aplicación se desarrollaron algoritmos basados en la representación MCT a partir de la cual se extrae la información para estimar las distintas funciones objetivo de interés. Los algoritmos se basan en la extracción de puntos notables o esenciales de la imagen. Para esta aplicación se ha considerado solamente una restricción: la existencia de un camino ya sea ruta señalizada o camino sin marcación. El algoritmo propuesto debe proporcionar buenos resultados para las condiciones severas que exige una ayuda en la navegación: vibraciones del vehículo, día, noche, lluvia, niebla, distintas condiciones de luz solar en intensidad y posición. Toda la información es obtenida de una sola cámara en espectro visible en escala de grises.

La obtención de los puntos notables o píxeles esenciales en una imagen es el proceso a partir del cual se desprende la extracción de la información objetivo. Finalmente se brindan resultados experimentales para distintas condiciones ambientales y las conclusiones.

4.1.2 Procesamiento de puntos notables o píxeles esenciales

En el procesamiento de imágenes es usual binarizar la imagen para su posterior tratamiento. Si bien esto simplifica los procesos posteriores, la elección del valor del umbral es crítica en imágenes de entornos exteriores. Este valor determina que es objeto y que es fondo y el valor óptimo depende fuertemente de la imagen. Esta dependencia requiere de “inteligencia” para hallarlo. Los métodos que brindan mejores resultados se basan en la información proporcionada por el histograma, entropía, correlación espacial y la forma local [Mehmet Sezgin, Bulent Sankur, 2004]. En una aplicación de procesamiento de video en tiempo real, dos aspectos son primordiales. Primero, realizar un preprocesamiento para eliminar los píxeles que no aportan información y segundo que este preprocesamiento sea la fuente de información para los ulteriores procesos de decisión. El principal aporte que se hace en esta tesis para la presente aplicación es la independencia de la elección de un valor umbral, ya que píxeles esenciales son obtenidos directamente en cada cuadro de la imagen sin ningún preprocesamiento previo. El algoritmo propuesto es una extensión directa del presentado en el capítulo 2, apartado 2.2, para el tratamiento de señales unidimensionales.

Sea $I[f,c]$ la matriz de $M \times N$ que representa los niveles de intensidad de un cuadro del video en 2^b niveles de grises. Realizando una exploración de filas obtenemos M señales unidimensionales $x[i]$ con $i = 0:N-1$ para las M filas. En cada una de estas señales se obtiene la representación MCT. En este contexto significa una transición de trayectoria específica hacia un nivel de gris alto o bajo. En el algoritmo propuesto, se interpolan píxeles de la señal y se comparan con los píxeles de la imagen. Como resultado de esta comparación se genera un error vectorial, donde su magnitud y signo son relevantes. A diferencia de los métodos tradicionales que

realizan una comparación contra un umbral, en esta propuesta la señal se compara a sí misma. Si el error es bajo, menor que una cota, ese pixel no conlleva información significativa y es eliminado para futuros cálculos y se continúa interpolando con sus píxeles vecinos. Cuando el error supera esta cota, el pixel es etiquetado como esencial y se le asocia, teniendo en cuenta la mayoría de los signos de error de interpolación, una clasificación acorde al tipo de segmento es generado.

La trayectoria desde un pixel esencial a otro esencial aporta información de la forma de la señal entre estos dos puntos. El signo del error de interpolación brinda la ubicación de la trayectoria y se la compara con la recta que une los dos píxeles esenciales. Si la mayoría de los errores de interpolación son positivos, es decir, los píxeles reales están por encima de los interpolados, se asocia al espacio entre estos dos píxeles un tipo de comportamiento.

La probabilidad de ocurrencia de los segmentos “a”, “b” y “c” (ver figura 2.8 del capítulo 2) es cero si el máximo segmento permitido excede al tamaño de la fila. El segmento tipo “h” no aporta información y se evita la ocurrencia decidiendo por el segmento más probable. Por lo tanto, con los segmentos “d”, “e”, “f” y “g” se construye el sistema de representación MCT. La nueva estructura de la información de la imagen incluye el valor y la posición de los píxeles esenciales y la clase del segmento asociado. A partir de la secuencia de segmentos obtenidos es posible obtener más información como se desarrolló en el capítulo 3. En el caso particular de la detección de líneas, los máximos y los mínimos son importantes. Los máximos se encuentran en la unión de segmentos “de”, “df”, “ge”, y “gf”. Los mínimos en la unión de los segmentos “ed”, “fd”, “eg”, y “fg”.

Los máximos y los mínimos son controlados por el error de interpolación. En la Fig. 4.2a. se muestra una imagen de una ruta en 256 niveles de grises, en la Fig. 4.2b una imagen de los puntos máximos obtenidos para un error de interpolación con un valor de $5/255$ y en la Fig. 4.2c un error de $20/255$. El algoritmo demostró ser eficiente y entrega la información necesaria para los demás procesos. Se puede observar que la densidad de máximos y mínimos disminuye sobre el camino, y las líneas presentan detecciones a lo largo de su recorrido, Fig. 4.2c. Si bien el

algoritmo se centra en la detección de las líneas, si éstas no estuvieran presentes, sería posible poder discriminar entre camino y no camino, estimando la densidad y alineación de píxeles esenciales de las regiones para un error de interpolación grande.



Figura 4.2a. Imagen de una ruta en 256 niveles de grises, 640x 480 píxeles.

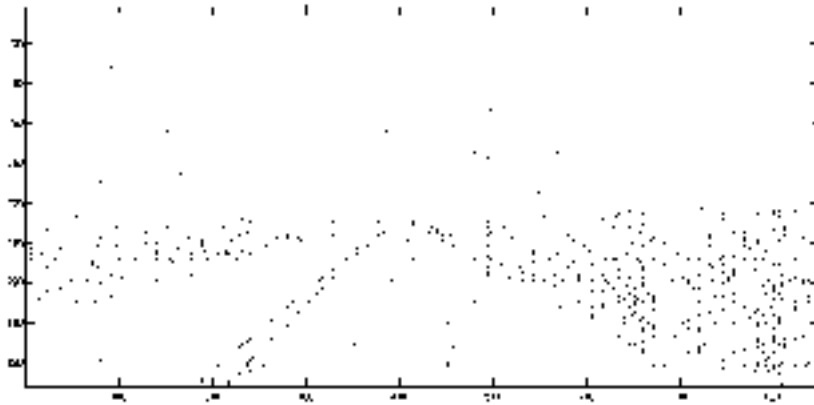


Figura 4.2b. Máximos locales de la imagen de la Fig. 4.2a para un error de interpolación de $5/255$.

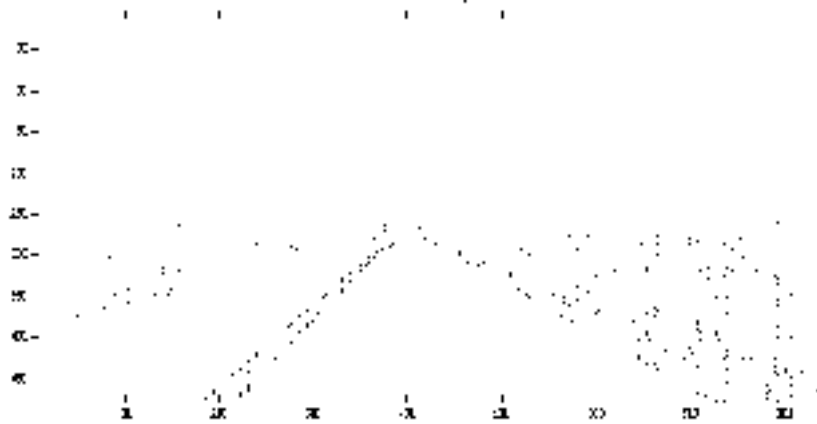


Figura 4.2c. Máximos locales de la imagen de la Fig. 4.2a para un error de interpolación de 20/255.

4.1.3 Algoritmos para la detección de líneas

Los algoritmos descritos en esta sección se adaptan particularmente al tipo de escenario que se espera encontrar. En la parte izquierda de este tipo de imágenes (ver Fig. 4.2a), las líneas de interés tendrán un ángulo con respecto a la horizontal entre 0 y 90 grados, mientras que en la parte derecha de éstas será entre 90 y 180 grados. Teniendo en cuenta estos ángulos, se definió una máscara distinta para izquierda y derecha que busca tres puntos notables alineados en k direcciones para un ángulo de 90 grados como se observa en la Fig. 4.3.



Figura 4.3. Subimágenes de búsqueda derecha e izquierda. Se detectan líneas solo en estas zonas.

En la Figura 4.4 se presenta la máscara de búsqueda para $k=13$ direcciones en 90 grados. El valor de k debe estar en concordancia con la calidad de la imagen. Para imágenes de baja resolución, menor a $\frac{1}{4}$ VGA (160 x 120), 7 direcciones son suficientes según resultados experimentales.

Una segunda observación importante es que la zona de búsqueda de la parte izquierda y derecha de la imagen debe ser el sector inferior de la imagen solamente, ya que es lo que el móvil va a recorrer en el futuro cercano como se observa en la Fig. 4.3. Además, de esta manera, se disminuye el costo computacional y se evitan problemas por la distorsión de líneas debido imperfecciones de las cámaras de video.

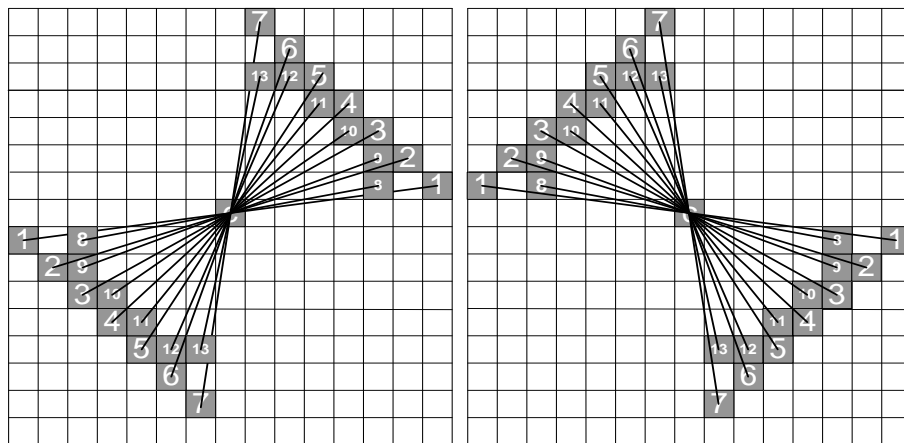


Fig. 4.4. Máscaras empleadas para el lado izquierdo y derecho de la imagen. En este caso $k=13$ direcciones de búsqueda para un ángulo de 90 grados.

El algoritmo comienza buscando tres puntos alineados en cada una de las k direcciones. Para cada alineación encontrada se genera una línea en una matriz auxiliar en donde se suman los resultados. En esta matriz se acumulan las coincidencias de alineaciones detectadas. Una vez explorada la zona, se suman las coincidencias en cada dirección. En la Fig. 4.5 se brinda el algoritmo en pseudocódigo.

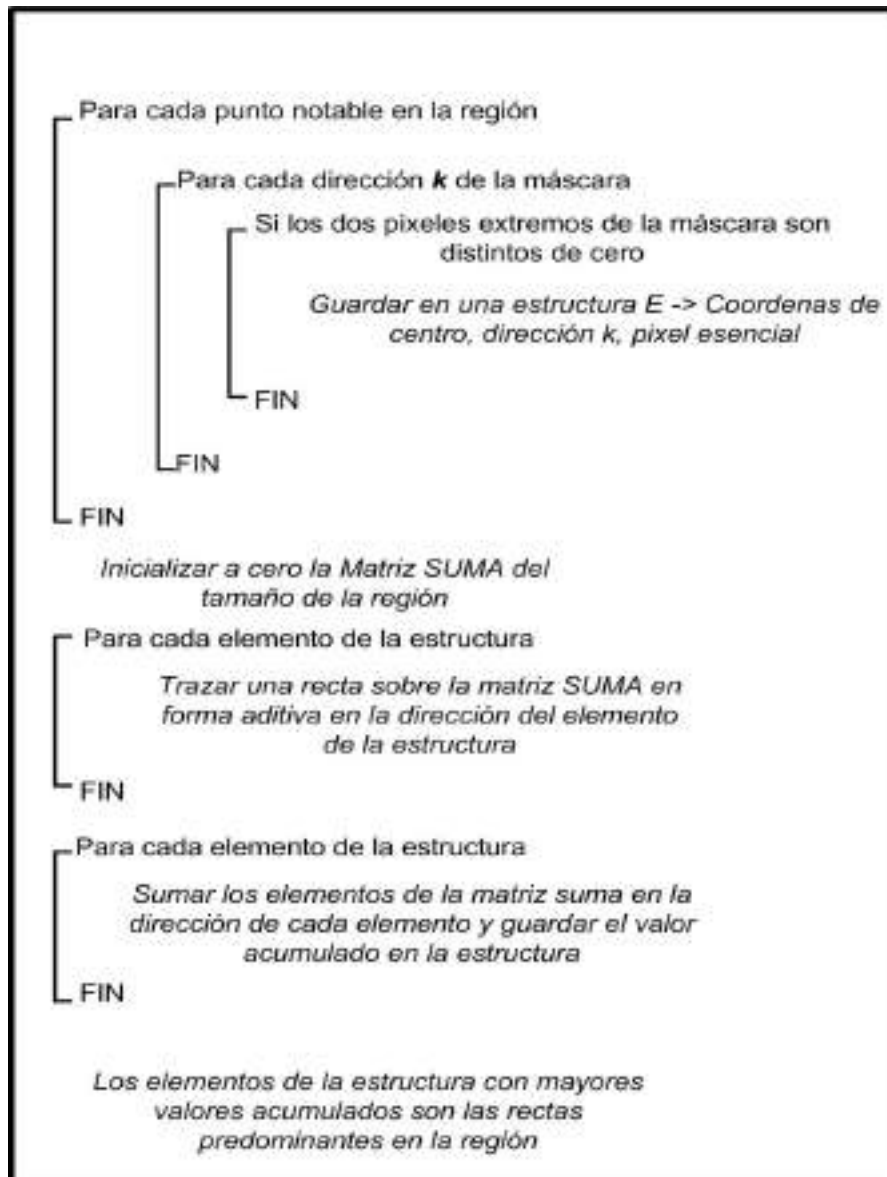


Figura 4.5. Algoritmo de detección de líneas sobre una región particular. La máscara cambia según si es región derecha o izquierda.

La acumulación de coincidencias genera las líneas predominantes en la región de búsqueda. En la región derecha la línea predominante más cercana al vehículo es la línea que posibilita el control de la navegación. Con la línea predominante de la región izquierda se obtiene el punto de fuga de la imagen.

4.1.4 Validación del algoritmo propuesto

Para validar el algoritmo propuesto y ajustar parámetros funcionales, se ensayó primero sobre imágenes sintéticas. Se generó una imagen con una línea como se observa en la Fig. 4.6 y se le agregó ruido tipo sal y pimienta [Jayaraman, et al. 2009]. Si bien este tipo de ruido es la peor condición para el algoritmo, es la que más se asemeja a los bordes de la ruta ya sea pasto o piedras en situaciones reales. En la Fig. 4.6 también se muestra un zoom de una porción de la línea donde aún para el cerebro humano es complejo de segmentar.

Se aplicó el proceso de segmentación a la imagen ruidosa de la Fig. 4.6 y en la Fig. 4.7 se muestran los resultados para las cuatro clases de segmentos además de los máximos y mínimos.

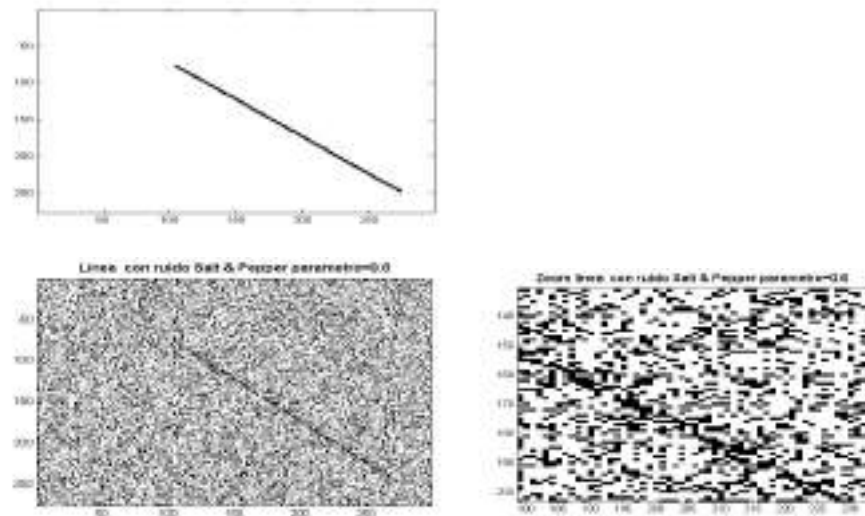


Figura. 4.6. Superior: imagen sintética de prueba. Inferior izquierda: imagen con ruido tipo sal y pimienta con parámetro de intensidad de 0.6. Inferior derecha: zoom de una parte de una línea que resalta la dificultad de la detección.

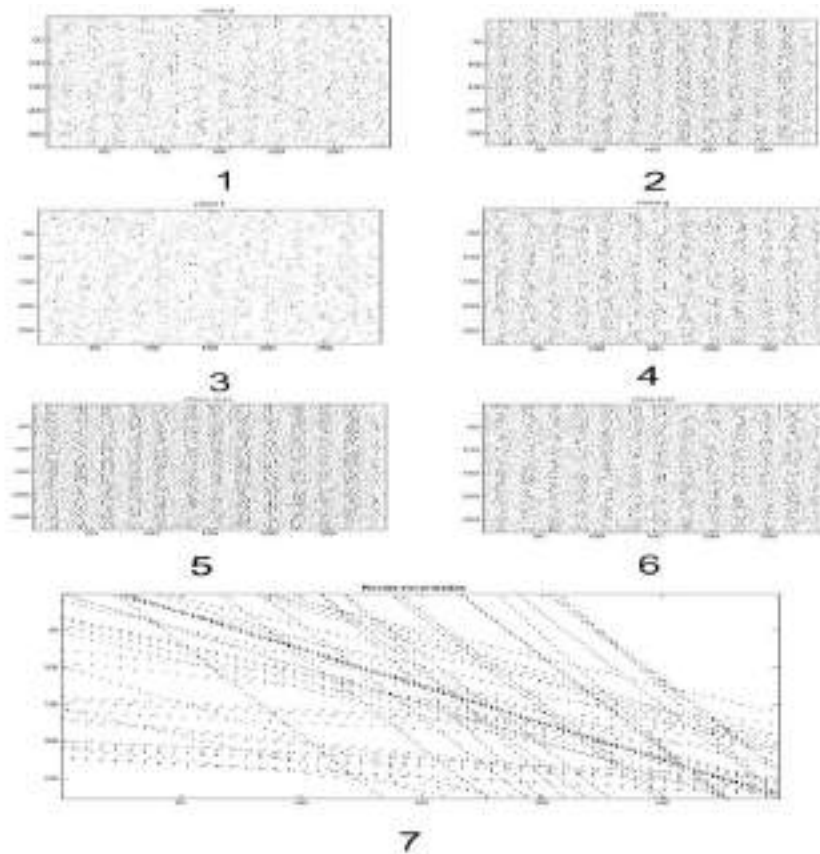


Figura 4.7 1) Píxeles esenciales para segmentos clase “d”. 2) Segmentos clase “e”.3) Segmentos clase “f”. 4) Segmentos clase “g”. 5) Segmentos asociados a máximos. 6) Segmentos asociados a mínimos. Por último, en 7) La dirección principal encontrada en donde claramente se destaca la original.

Si bien desde el punto de vista teórico, se hubiera esperado que los segmentos asociados a mínimos hubieran sido lo que aportaran información más clara para el algoritmo de búsqueda, se detectó que los segmentos tipo “d” tienen mejor desempeño que los mínimos. La marcación de la línea es similar, pero los segmentos tipo “d”, simplifican más la imagen donde no hay una línea definida. En el caso de líneas blancas los mejores segmentos son los del tipo “e”. Estas importantes conclusiones fueron corroboradas con imágenes reales cuyos resultados se presentan en la siguiente sección.

4.1.5 Resultados experimentales

Se ejecutaron los algoritmos sobre imágenes reales tomadas desde el interior de un vehículo basadas en una sola cámara con resolución 640 x480 en 256 niveles de grises.

Se realizaron pruebas en distintos tipos de ruta bajo distintas condiciones a fin de asegurar la robustez del sistema, imágenes con bajo contraste y en situaciones de diferentes condiciones ambientales como sol de frente, atardecer y noche. Dos clases de procesamiento se destacan. La primera consiste en procesar la imagen directamente sin convertirla previamente en binaria. Los pixeles esenciales más importantes son los máximos para las líneas blancas o zonas más brillantes que se alinean. El segundo consiste en convertir la imagen en binaria mediante un simple umbral para luego aplicar el algoritmo de detección de líneas empleando segmentos clase “e”. La elección del umbral no es crítica y es preferible un umbral superior ya que es mejor perder parte de la definición de la línea a que se desdibuje el contorno. En la Fig. 4.8 se brindan resultados de ambas propuestas para líneas con escaso contraste o despintadas.

Bajo condiciones de buena visibilidad ambas propuestas se desempeñan correctamente. Pero en condiciones de bajo contraste, como ser: sol de frente, amanecer, atardecer, penumbra y noche, es conveniente procesar la imagen sin aplicar binarización. En la Fig. 4.8 el procesamiento de la imagen inferior no detecta la línea si la imagen se binariza con un umbral. Por lo tanto, en todos los resultados experimentales no se empleó binarización y el algoritmo fue testado en condiciones extremas de visibilidad y contraste.

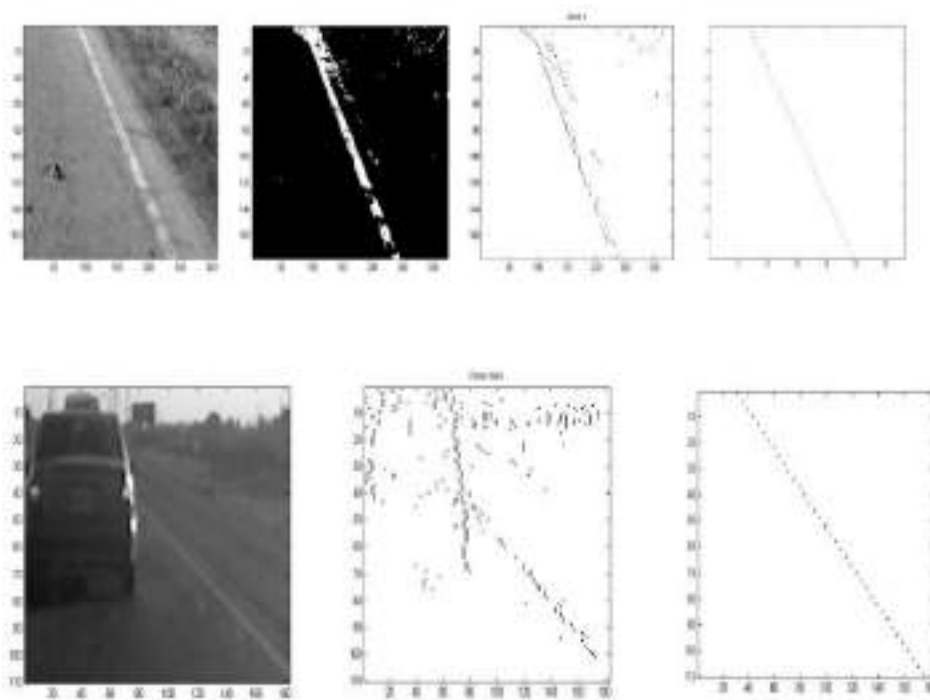


Figura 4.8 Superior de izquierda a derecha: Imagen de una línea desmarcada. Imagen binaria mediante umbral simple. Segmentos clase e y la mejor línea estimada. Inferior de izquierda a derecha: Imagen con poco contraste sobre la línea tomada en un atardecer. Segmentación de máximos y la mejor línea estimada.

En el caso de las imágenes de prueba nocturnas, dado que éstas presentan menor contraste, se debe disminuir el error de interpolación de la segmentación. La Fig. 4.9 muestra resultados para una detección de línea derecha con escasa visibilidad. El algoritmo implementado ajusta automáticamente el error en base a la cantidad de píxeles esenciales en la zona de búsqueda.

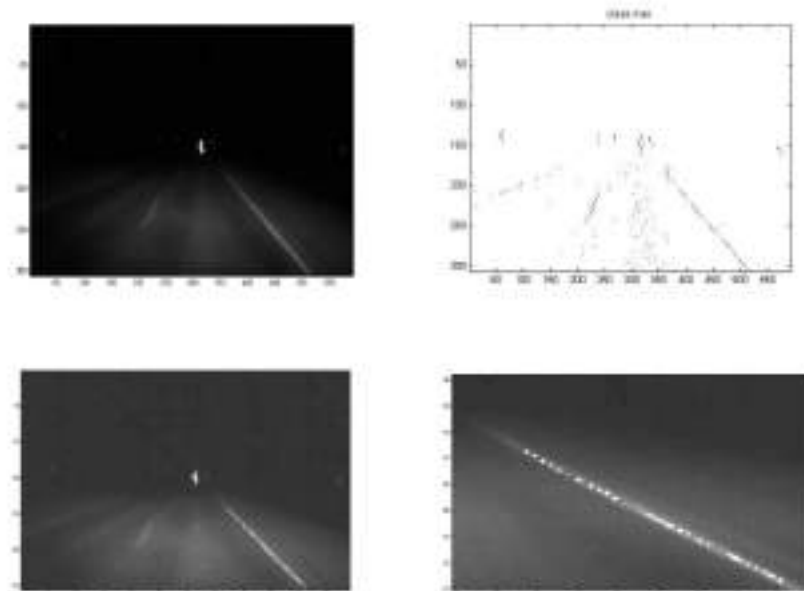


Figura 4.9 Superior izquierda, captura nocturna. Superior derecha, pixeles esenciales máximos. Inferior izquierda. Puntos alineados detectados superpuestos en la imagen. Inferior derecha, zoom de la detección y el trazado de la recta más importante.

La peor condición de visibilidad ocurre al tener sol de frente. Se ensayaron distintas tomas llegando a la conclusión que los segmentos “e” y “f”, segmentos descendientes, son los mejores indicadores. Esto se debe a que el sol se refleja fuertemente en la ruta y no fuera de ella implicando una fuerte caída de nivel de intensidad de los pixeles fuera de la ruta. La Fig. 4.10 presenta resultados sobre una condición extrema para la búsqueda de la recta derecha. Es importante destacar la similitud de esta situación con la imagen sintética mostrada en la Fig. 4.7.

4.1.6 Implementación para los ensayos

Se desarrollaron los algoritmos en lenguaje C, implementados sobre la plataforma Netbeans y empleando la librería OpenCV solo para la captura de los *frames*. En el link [https://www.youtube.com/watch?v=A4nJ_eoBATA] se puede observar un video que corre en tiempo real sobre una PC que entrega información del desempeño del algoritmo de detección de líneas. Además, se le incorporó información de

ubicación del móvil en la ruta, patrón de manejo, irregularidad, condiciones de visibilidad y detección de proximidad de banquina.

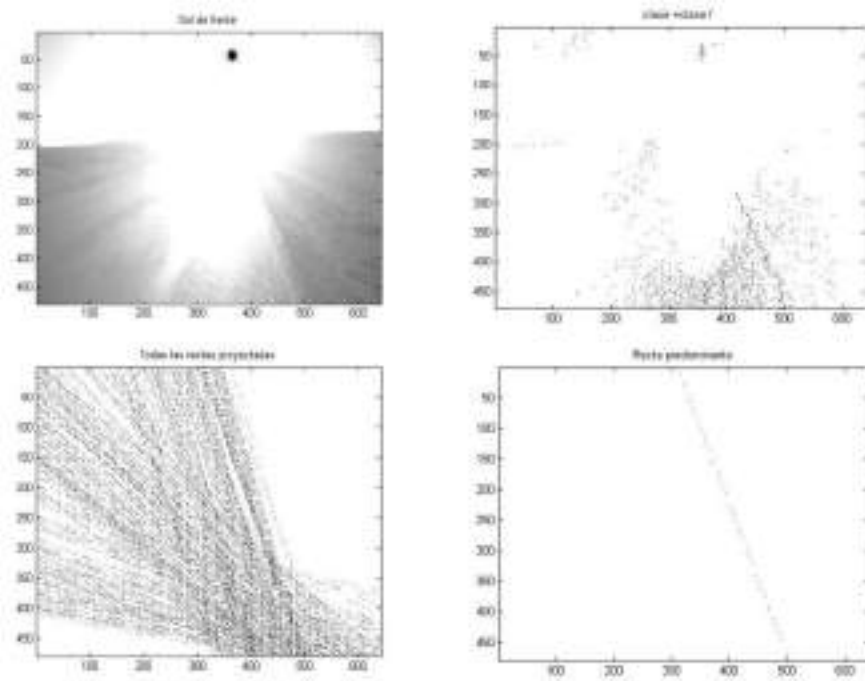


Figura. 4.10. Superior imagen con sol de frente y los pixeles esenciales asociados a clases “e” y “f”. Inferior todas las rectas candidatas y la recta que acumula mayor cantidad de coincidencias.



Figura 4.11 Interfase visual implementada para los ensayos.



Figura 4.12 Interfase visual implementada para los ensayos. En este caso se ubica al móvil en la mano contraria.

4.1.7 Conclusiones

Procesar información de video en tiempo real para el entendimiento de escenarios correspondiente a ambientes exteriores, sigue siendo un desafío. La disparidad de condiciones de luz, desde la noche hasta la plena luz del día junto a la cantidad de información a procesar, requiere de abstracciones inteligentes para llegar a una solución robusta. Contar con superprocesadores no garantiza alcanzar el objetivo. Los algoritmos presentados extraen de manera simple la información en una imagen, mediante los pixeles esenciales, bajando los órdenes de magnitud de los datos procesando sólo los relevantes. Los pixeles esenciales representan entre el 1 y el 10% de los pixeles para este tipo de imágenes. Las clases “d”, “e”, “f”, “g”, los máximos y mínimos se obtienen a medida que se leen los pixeles de la imagen. En la misma exploración matricial es posible trazar las líneas para cada alineación encontrada. Esos datos esenciales son la fuente de ingreso para otros procesos de orden superior. En esta aplicación se utilizaron los algoritmos para detectar líneas u objetos alineados. Empleando la estructura MCT es posible extraer más información como, por ejemplo, estimación de velocidad, flujo óptico y objetos sobre la ruta.

4.2 Aplicación en imágenes: Procesamiento de Imágenes Térmicas aplicado a cámaras infrarrojas de Bajo Costo

En esta sección se presenta el desarrollo de un sistema de adquisición y procesamiento de imágenes térmicas que emplea cámaras IR de bajo costo. Este sistema está orientado al monitoreo permanente de un proceso o dispositivo crítico que requiere monitoreo frecuente de su evolución térmica. La adquisición y procesamiento de un *frame* es llevada a cabo en un microcontrolador de 16 bits, generando una solución en un chip. En las secciones subsiguientes, se describe primero la naturaleza de una imagen térmica y la tecnología de las cámaras infrarrojas poniendo énfasis en sus virtudes y debilidades. Luego se describen los algoritmos empleados en el análisis de la imagen y como estos son implementados en el microcontrolador. Por último, se presentan resultados experimentales y las conclusiones.

4.2.1 Introducción a las cámaras infrarrojas

Cámaras termográficas

Una cámara termográfica es un dispositivo que detecta el patrón de emisión de la escena de visión en el espectro de la longitud de onda infrarroja. En la Fig. 4.13 se observa el espectro electromagnético detallando las bandas correspondientes a la zona infrarroja. La cámara empleada en los ensayos para este trabajo posee una sensibilidad menor a 50 mK (miligrados Kelvin), en la región LWIR (Long Wave Infrared) desde 8 a 14 μm , [Flir,2019].

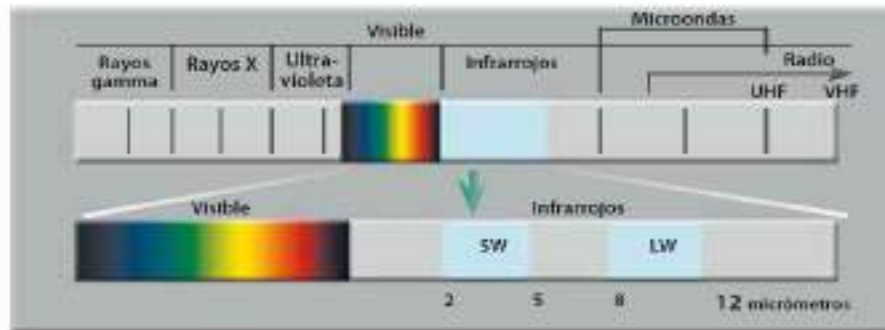


Figura 4.13 Espectro electromagnético. La banda infrarroja se divide en dos sub-bandas principales: Short Wave IR correspondiente a una longitud de onda de 2 a 5 μm y Long Wave IR desde 8 a 12 μm .

Las cámaras termográficas están compuestas básicamente de una lente transparente a la energía infrarroja y de una matriz de $M \times N$ elementos sensibles que traducen esta energía a una señal eléctrica. Por lo tanto, las cámaras IR proporcionan la radiación de energía promedio en cada subdivisión de la escena según se proyecta a cada elemento de la matriz de sensores a través del sistema óptico. En el caso particular de la cámara empleada, la resolución es de 80×60 elementos sensibles. Cada elemento detector es un microbolómetro de óxido de vanadio, en el cual la temperatura varía según la radiación incidente. Los cambios de temperatura producen un cambio proporcional en la resistencia de cada detector. Mediante circuitos de sensado y conversores A/D la información matricial es enviada en forma secuencial mediante protocolos normalizados descritos en [Flir,2019].

Radiación y temperatura superficial

La potencia de radiación de una superficie real está dada por la ley de Stefan-Boltzmann:

$$\epsilon = \epsilon \sigma T^4 \quad (24)$$

Donde ϵ es la emisividad, $\sigma = 5.67 \cdot 10^{-8} \text{ W/m}^2\text{K}^4$, es la constante de Stefan-Boltzmann y T la temperatura de la superficie en grados Kelvin.

La emisividad es la relación de emisión referida a la emisión del cuerpo negro ideal, para cual toma el valor unitario. La emisividad es baja para metales pulidos.

Como la emisividad es determinada por las características de la superficie del objeto, un metal pintado cambia su emisividad a la correspondiente a la pintura empleada. La Fig. 4.14 presenta el valor de emisividad para algunos materiales comunes.

MATERIAL	EMISIVIDAD ϵ
Cuerpo negro	1
Piel humana	0.98
Agua	0.95
Cerámica	0.95
Barro	0.95
Cemento	0.95
Grava	0.95
Papel	0.95
Plástico	0.95
Goma	0.95
Madera	0.95
Cobre(sulfatado)	0.68
Acero Inoxidable	0.1
Cobre pulido	0.02
Aluminio pulido	0.05

Figura 4.14. Valores aproximados de emisividad para distintos materiales.

Si no se conoce la emisividad del objeto en la escena, la ecuación (24) presenta dos incógnitas; ϵ y T , ya que las cámaras son sensibles a la radiación en el espectro infrarrojo.

Este análisis simplificado conduce a dos formas de emplear la termografía: análisis cualitativo y cuantitativo. La técnica de la termografía cualitativa consiste en detectar gradientes térmicos y patrones anormales en la escena de inspección. El defecto causante de este patrón térmico se localizará por comparación con otros cuerpos de esa misma condición. Una de las ventajas de esta técnica es que no necesita una medida exacta de la temperatura, sino que la clasificación normal/anormal se basa en análisis comparativo.

La termografía cuantitativa obtiene la temperatura en cada punto de la escena, lo que implica conocer la emisividad del objeto. La gran mayoría de las aplicaciones

se basan en análisis cualitativo dado que la información del comportamiento térmico del objeto es función de la distribución espacial de temperaturas relativas. En aplicaciones fijas, como la propuesta en esta aplicación, permite además obtener la distribución temporal al observar la misma escena a lo largo del tiempo. Esta dimensión adicional posibilita detectar condiciones anormales de operación o sobrecarga calculando los gradientes temporales de regiones de la escena.

Para ilustrar estos conceptos, en la Fig. 4.15 se observa la imagen térmica de un transformador de distribución de intemperie capturada con nuestro sistema de adquisición. El transformador posee una distribución uniforme de temperaturas, pero se observa una zona rectangular más fría. Esta zona corresponde a la placa metálica de identificación de la maquina eléctrica que al tener un coeficiente de emisividad bajo se observa, erróneamente como más fría.

En una aplicación de adquisición fija, los elementos de baja emisividad son conocidos de antemano, por lo tanto, pueden ser descartados del análisis del comportamiento térmico.

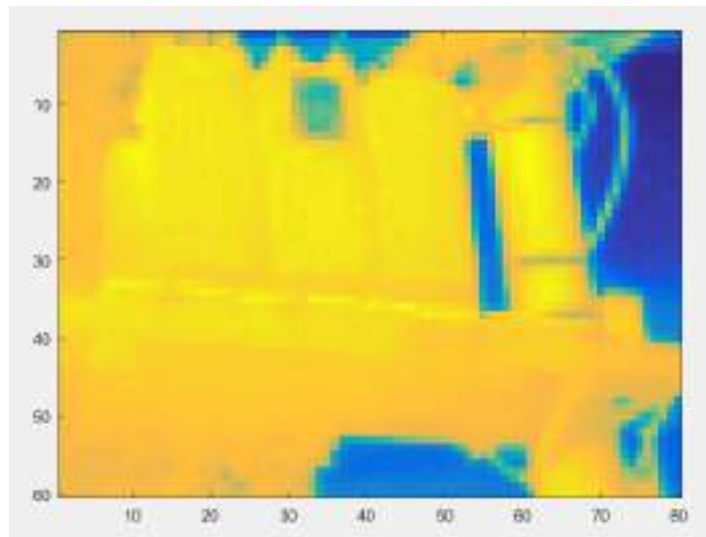


Figura 4.15. Imagen térmica, obtenida con nuestro sistema de adquisición, de un transformador de distribución de intemperie. El rectángulo más frío observado corresponde a la placa identificatoria metálica de baja emisividad. Se observa una distribución homogénea de temperaturas.

4.2.2 Procesamiento de la imagen térmica

Condiciones de borde

El procesamiento sobre la imagen térmica fue diseñado para ser realizado empleando los recursos del microcontrolador. Como no se conoce de antemano la escena para la aplicación final y para que sea genérico se implementaron algoritmos que extraen características comunes a numerosas aplicaciones. Además, existe la posibilidad de enviar por el puerto serial el *frame* completo para ser procesado en forma remota.

En general, en una imagen térmica, es útil determinar el valor absoluto de puntos calientes, el gradiente espacial de temperatura centrado en estos puntos y la evolución temporal de ellos.

4.2.3 Máximos de la imagen

El muestreo MCT proporciona de manera inmediata los máximos de la imagen, puntos calientes. En nuestro caso particular, se considera a la imagen como un arreglo de 60 segmentos de señales unidimensionales de 80 elementos. Cada *frame* consta de 4800 píxeles. Solo los píxeles esenciales son almacenados en memoria, quedando una cantidad considerablemente menor, que es función de la escena y del error de interpolación prefijado. Se procesan las filas de la imagen en forma secuencial para la obtención de los píxeles esenciales y los máximos locales controlados por un error de interpolación.

4.2.4 Algoritmos de extracción de características

El *frame* adquirido es subdividido en 16 regiones de 20 x15 píxeles. Esta subdivisión es realizada por dos motivos. Primero, simplifica el análisis del comportamiento térmico y lo adapta a los recursos de microcontrolador y segundo, se obtiene una firma digital de la escena que es empleada para validarla, teniendo en cuenta que los procesos térmicos son lentos y el objetivo del adquisidor es observar procesos

en forma continua.

Para cada región i,j con $i=1, 2, 3,4$ y $j=1, 2, 3,4$ se realizan los siguientes procesos.

1. **Cantidad de pixeles esenciales. (Cpix)**
2. **Valor medio de los pixeles esenciales. (Mpix)**
3. **Cantidad de máximos. (CMax)**
4. **Máximo de mayor valor. (MMax)**
5. **Diferencia entre la media de los máximos y la media de los pixeles esenciales. (DifMpixMax)**

Por lo tanto, se obtienen 80 indicadores que describen la escena, son sencillos de calcular y son empleados para validarla y analizarla.

4.2.5 Validación de la escena

Validar la escena en este contexto significa que se está adquiriendo la misma imagen o que hay energía térmica en juego en la escena. La diferencia entre dos *frames* seguidos obtenida en un tiempo mucho menor que las constantes de tiempo térmicas involucradas, debe ser pequeña. Dos motivos principales generan grandes diferencias. Primero que no exista energía térmica significativa en la escena, por ejemplo, la maquina está apagada y segundo que la escena cambió en forma abrupta, por ejemplo, se interpuso algún objeto o persona. Para validar una escena se debe obtener una diferencia menor a un umbral, Thd , en una cantidad de frames consecutivos Kd . Para la validación se emplean los procesos 1 y 2 según las siguientes ecuaciones:

$$dif1 = \sum |Cpix_{i,j,n} - Cpix_{i,j,n-1}| \quad (25)$$

$$dif2 = \sum |Mpix_{i,j,n} - Mpix_{i,j,n-1}| \quad (26)$$

Donde $i=1, 2, 3,4$; $j=1, 2, 3,4$ y n es el frame actual y $n-1$ el anterior, con una

diferencia temporal mucho menor que las constantes térmicas involucradas. Un nuevo *frame* será validado si se cumple la siguiente inecuación:

$$dif1 + dif2 < Thd \quad (27)$$

para Kd *frames* consecutivos.

4.2.6 Análisis térmico de *frames* validados

Se ejecutan dos procesos básicos: el crecimiento temporal de los máximos y la evolución espacial de la energía térmica en cada región. Se calculan los valores resultantes para cada uno de los 5 procesos en cada región. En base a los resultados, se seleccionan las regiones que requieren atención para su seguimiento. Por ejemplo, en una región sin máximos no se realiza ningún calculo posterior. Se busca el máximo absoluto de la imagen y se registra la evolución temporal de las regiones vecinas en sucesivos *frames*, ver Fig. 4.16.

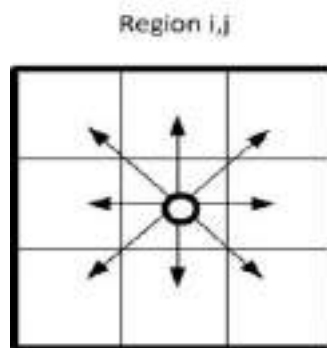


Figura 4.16 Regiones vecinas centradas en el máximo absoluto de la imagen.

4.2.7 Resultados experimentales sobre imágenes reales

Se procesaron imágenes térmicas de un disipador de un transistor de potencia. Se adquirieron 30 *frames* a intervalos de 20 segundos. En la Fig.4.17 se observa la imagen térmica del disipador y en la Fig.4,18 la evolución temporal del máximo MMax de la imagen.

En la Fig.4.19 se presenta el estado térmico inicial y final del disipador luego de

los 10 minutos de ensayo. Se calcularon los 5 procesos y es importante destacar la importancia del proceso 5, DifMpixMax, ya que un aumento de ese valor implica que el calor se está concentrando en pequeñas superficies o que la velocidad de generación de calor es mayor a la de evacuación. En la Fig. 4.20 se observa la captura de la imagen térmica de un motor, que a través de las rejillas de ventilación se puede inferir la temperatura del bobinado.

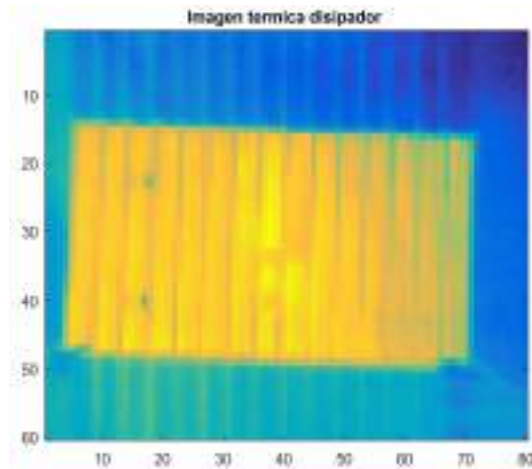


Figura 4.17 Imagen térmica del disipador bajo ensayo.

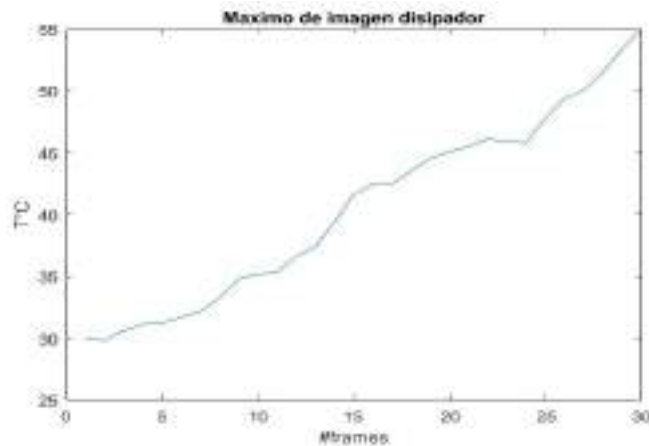


Figura 4.18 Evolución temporal del máximo MMax de la imagen disipador

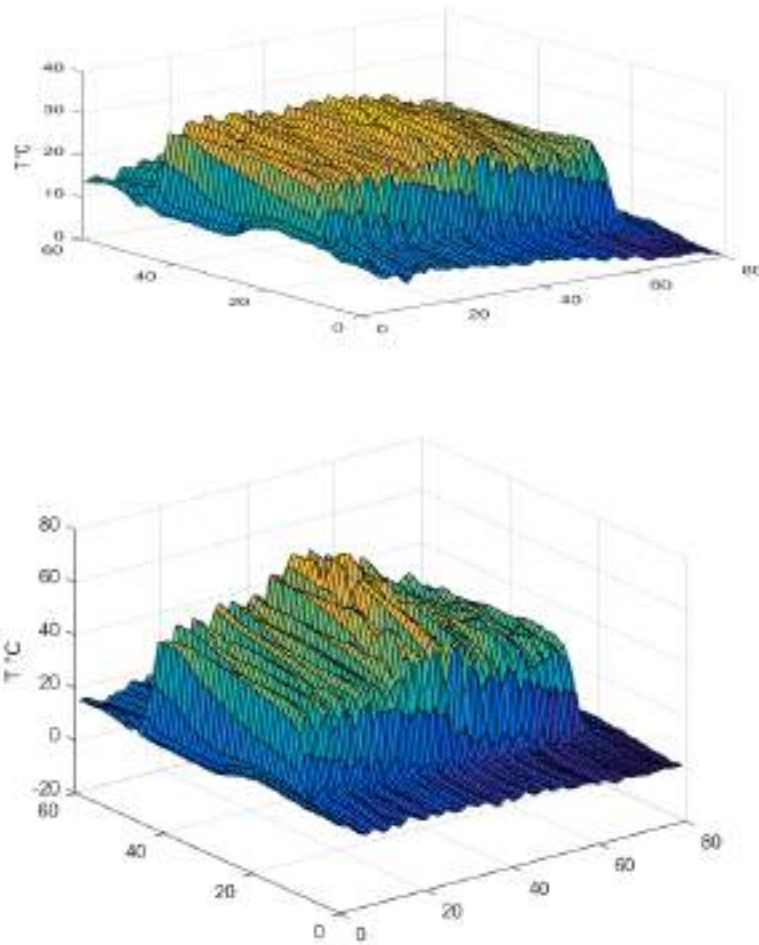


Figura 4.19 Imagen térmica del disipador al principio (superior) y al final del ensayo (inferior).

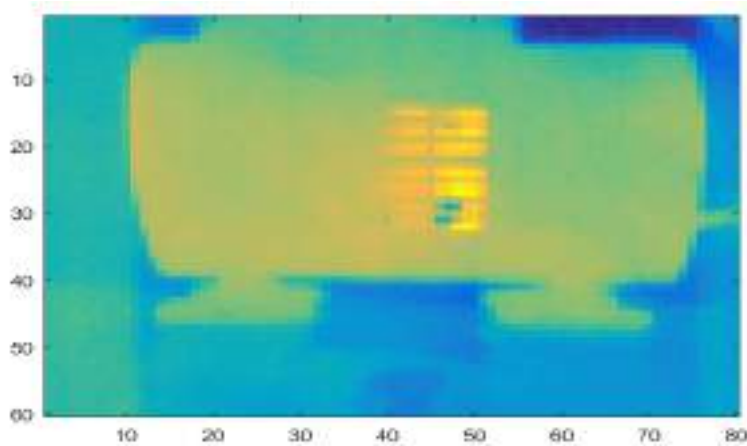


Figura 4.20 Imagen térmica de un motor en el cual se puede medir la temperatura de los bobinados por las rejillas de ventilación. Capturada con el sistema de adquisición propuesto.

4.2.8 Conclusiones

La termografía presenta la gran ventaja de observar un proceso o dispositivo sin contacto. En el caso presentado en este capítulo, la resolución es de 80 x 60 puntos. Aunque es baja, es como observar al proceso con 4800 termómetros. Al bajar los costos de las cámaras infrarrojas aumentan la cantidad de aplicaciones con capturas fijas que permiten evaluar el comportamiento térmico con la filosofía de un sensor inteligente. Las aplicaciones son innumerables, no solo para inferir patrones anormales, sino que además para validar el correcto desempeño de un sistema, por ejemplo, verificar que alcanza el régimen permanente con normalidad.

Se proponen 5 indicadores de comportamiento térmico y el seguimiento de regiones que presentan características de evolución térmica.

En algunas aplicaciones también son importantes los valores mínimos de una imagen, como por ejemplo en estudios de eficiencia energética. Los máximos y los mínimos locales son resultados inmediatos del muestreo MCT y puede ser implementada en microcontroladores de mediana complejidad.

5 CONCLUSIONES Y TRABAJO FUTURO

RESUMEN

Este capítulo se presentan las conclusiones generales teniendo en cuenta la experiencia acumulada durante el desarrollo de la tesis, sobre todo por las aplicaciones ideadas e implementadas. Además, se presentan los desarrollos actuales y las líneas de investigación futuras en análisis de señales sísmicas, análisis de señales sensoriales en procesos continuos e interpretación de imágenes.

5.1 Conclusiones generales

La estructura de información sensorial propuesta en esta tesis ha demostrado, a través de aplicaciones diversas, ser eficaz y simple para inferir en base a ella el conocimiento deseado. En todos los ámbitos, el conocimiento de las señales propicia mejora en la toma de decisiones. Por ejemplo, en un entorno industrial permite validar estados operativos, detectar patrones anormales y obtener sistemas más seguros como se describe en el siguiente punto.

Vivimos en un mundo altamente interconectado que ha alcanzado a todas las cosas y seres humanos e incluye aspectos jamás pensados. Creemos que la propuesta de representación de esta tesis es un aporte a lo que se dado a llamar “*Edge computing*” que es el paradigma de procesamiento con más probabilidades de convertirse en el predilecto en un futuro cercano. Este paradigma propicia la extracción de información y conocimiento en el punto de adquisición de las señales.

5.2 Trabajo futuro y desarrollos en curso

5.2.1 Señales sensoriales industriales

Todas las acciones realizadas por un sistema de instrumentación están basadas en las señales provenientes de sensores. Los sistemas digitales dirigidos por sensores convierten la percepción en información sobre la cual los operadores y sistemas actúan. Una percepción errónea puede generar graves accidentes, impacto medioambiental y pérdidas económicas. Siempre existe una probabilidad de falla, por lo tanto, los sistemas de instrumentación buscan tener más fuentes de información generando redundancia de hardware y software para minimizar la probabilidad de tomar decisiones incorrectas, obteniendo sistemas más seguros, con probabilidades de fallas muy bajas. Los sistemas de instrumentación son cada vez más complejos y la capacidad de procesamiento se encuentra cada vez más distribuida. El hardware ha experimentado un crecimiento exponencial en capacidad de procesamiento y capacidad de memoria. Pero el software industrial no se ha desarrollado acordeamente aun, sobre todo en el entendimiento de las señales sensoriales.

Los algoritmos y formas de representar a las señales sensoriales propuesta en esta tesis se incorporarán en forma local en PLCs (Controladores Lógicos Programables) y dispositivos de campo para analizar el comportamiento de las señales y aprender su comportamiento normal. A partir de esta inferencia, es posible estimar estados de señal, estadísticas, autoaprendizaje y el tipo de comportamiento normal/ anormal / ilógico.

Los resultados de los módulos serán consultados por los algoritmos de instrumentación y control con el objetivo de sumar información, obteniendo una decisión que minimiza la probabilidad de error. Los módulos propuestos correrán en paralelo a los sistemas de instrumentación, por lo tanto, se adaptan a todo proceso para sumar herramientas lógicas para la toma de decisiones generando sistemas más seguros. Además, estos módulos brindaran información en situaciones de falla que provocan la parada del proceso controlado, facilitando la

detección del problema dado que los módulos han observado las señales involucradas en el proceso y aportan elementos de juicio para detectar anomalías.

Las señales sensoriales industriales tienen embebida la información de sistema físico que representan, por lo tanto, la trayectoria de estas señales es acorde a la dinámica del sistema asociado. Analizar estas señales permite inferir el comportamiento del sistema que representan, validar su representación y detectar posibles estados anómalos. La representación propuesta en esta tesis se adapta perfectamente para el seguimiento de trayectorias.

Para optimizar su implementación en PLCs, se implementó una variante del muestreo MCT que solamente retiene en memoria tres muestras: la inicial del segmento y las dos últimas. Mediante la secuencia de operaciones descritas en la figura 5.1 se calcula la muestra interpolada para realizar la comparación con la real. Con esta técnica se eliminan los segmentos a,b,c y es apto para señales que varían lentamente en el tiempo o que permanecen en un mismo valor por un lapso de tiempo prolongado.

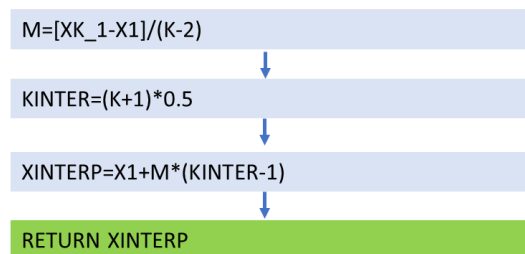


Figura 5.1 Secuencia de cálculo para la muestra interpolada empleando solo tres muestras de la señal.

Donde X_1 es la primera muestra del segmento, X_{k-1} es la muestra pasada y K es la longitud del segmento. La figura 5.2 muestra el diagrama de flujo en máquina de estados para la segmentación en tiempo real.

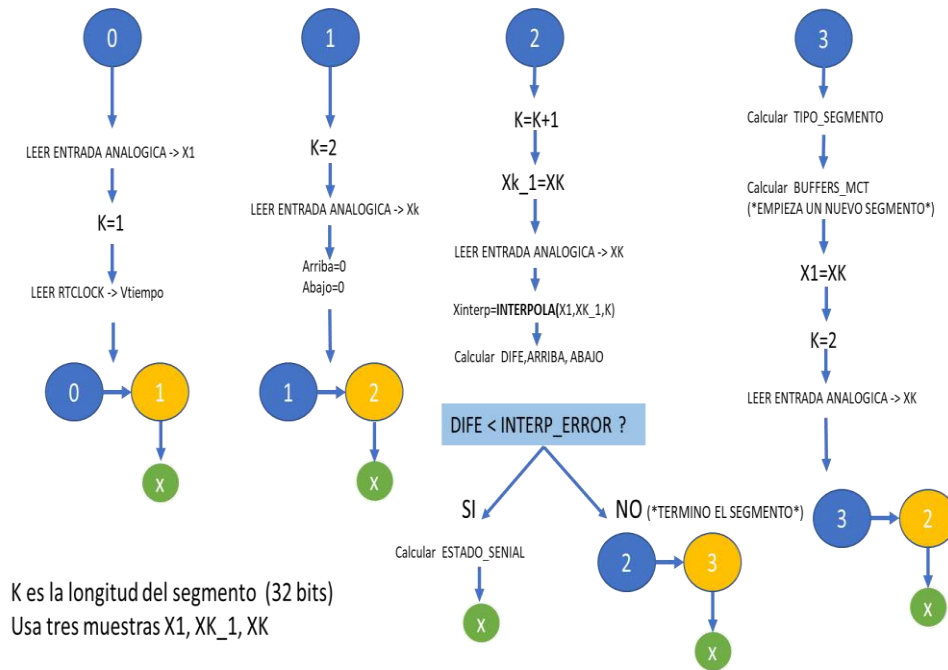


Figura 5.2 Algoritmo bajo el paradigma de máquina de estados para el cálculo en tiempo real de los segmentos.

En un proceso industrial, sobre todo en los de operación continua, las señales sensoriales exhiben un patrón operativo normal que puede ser aprendido por un sistema que observa estas señales. Para este objetivo se agregó una variable que clasifica la señal con respecto al rango. Ahora la representación de la señal puede sintetizarse como se observa en la figura 5.3.

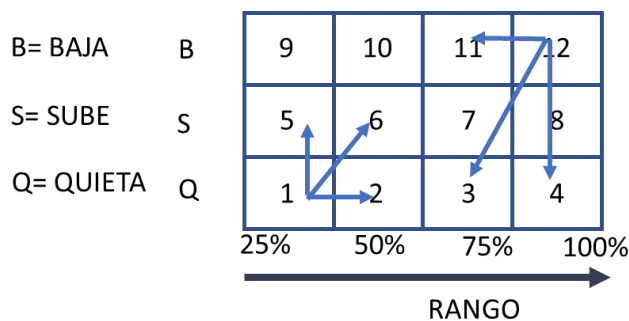


Figura 5.3 representación de la señal como una matriz de comportamiento vs rango.

La señal es clasificada en 12 estados. El rango de la señal es dividido en cuatro partes y cada una de ellas en tres condiciones: señal subiendo, bajando y en reposo. Recordemos que la señal sensorial transporta la dinámica del sistema físico que representa. Por lo tanto, la dinámica de cambio de estados, las probabilidades de transición y los tiempos de permanencia identificarán al sistema que la señal representa.

La dinámica de la señal es capturada por las siguientes variables aleatorias:

T_m = Tiempo mínimo promedio de permanencia en cada estado.

P_{ij} = Probabilidad de cambiar del estado **i** al **j**.

Los algoritmos propuestos tendrán una etapa de aprendizaje en la cual estiman los valores normales de las variables aleatorias. Se considera que el algoritmo aprendió la dinámica de la señal cuando se estabilizan los valores. En base a las probabilidades de transición, el algoritmo detecta comportamiento anormal de la señal.

Durante la etapa de **aprendizaje**, se estiman las probabilidades cambiar del estado **i** al **j** y los tiempos de permanencia. Durante la etapa de **análisis** se evalúan los tiempos de permanencia y las probabilidades acumuladas en promedio.

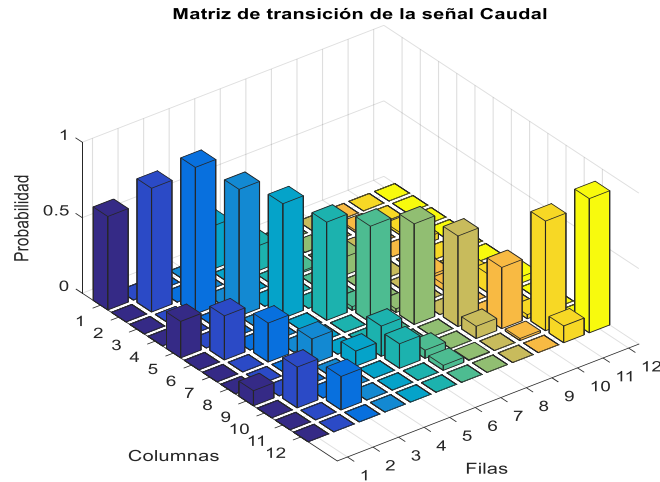
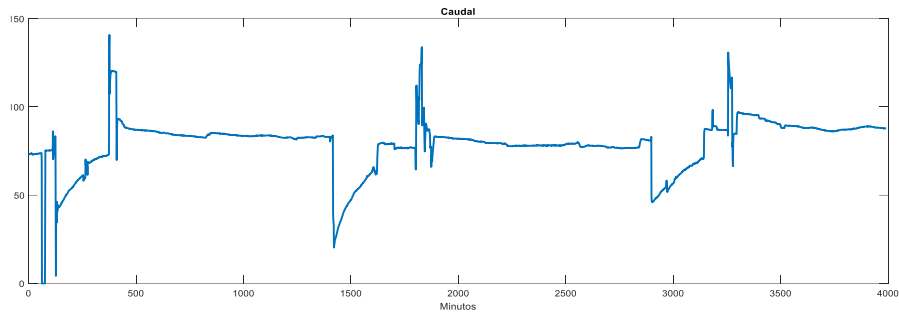


Figura 5.4 Matriz de probabilidad de transiciones obtenida para una señal de caudal.

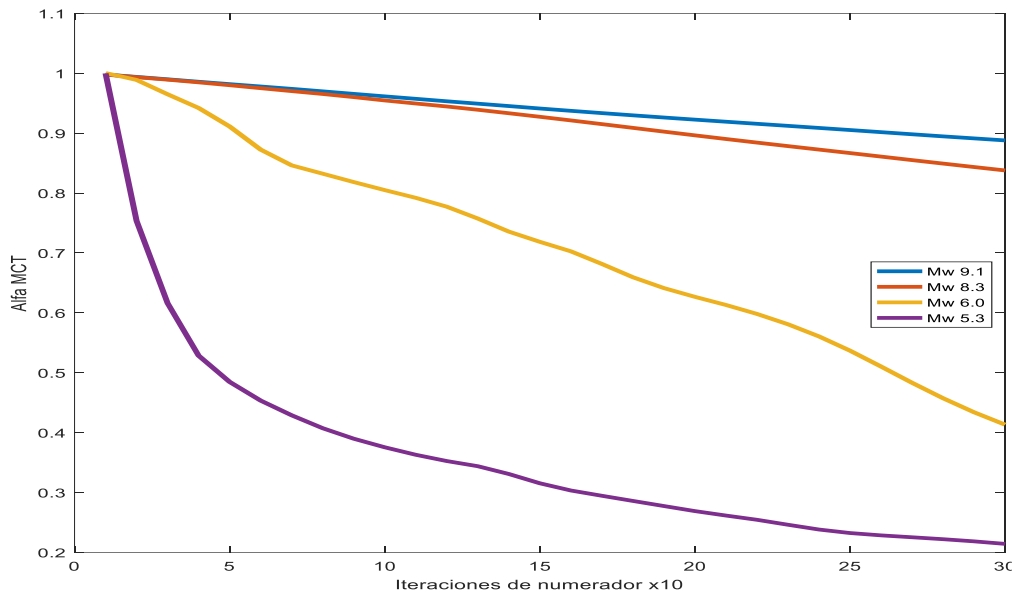
5.2.2 Procesamiento de señales sísmicas

Se continuó trabajando con las señales sísmicas para detección temprana de eventos basado en el análisis de los primeros tres segundos de la onda **P**. Los resultados preliminares han sido publicados en [Abate,2019] y han mostrado un excelente desempeño en la inferencia de la magnitud del terremoto que se acerca al punto de adquisición. Se ha definido un coeficiente **alfaMCT** como el cociente de la energía obtenida de la señal filtrada en forma iterativa con distintos niveles de iteración, según la técnica descrita en el capítulo 3. Este coeficiente presenta correlación con la magnitud del terremoto que se aproxima observado por las futuras ondas tipo **S**.

$$\alpha_{MCT} = \frac{Energia(x_p, Err, Iter_N)}{Energia(x_p, Err, Iter_D)}$$

Donde x_p es la Ventana de tres segundos de la señal sísmica, Err es el error de interpolación de muestreo MCT, $Iter_N$ numero de iteraciones del numerador del filtrado MCT e $Iter_D$ numero de iteraciones del denominador. Para los primeros resultados, la varianza fue empleada como estimador de la energía, pero otros estimadores podrían ser usados.

En la figura 5.5 se observa el desempeño de α_{MCT} en función de numero de iteraciones de numerador. Obsérvese como se logra sensibilidad para sismos de diferente magnitud.



Evento, fecha	M
Honshu, Japan 11/03/2011	9.1
Taokachi-Oki, 9/26/2003	8.0
North Miyagi, 7/26/2003	6.0
San Juan, Argentina, 12/31/2018	5.3

Figura 5.5 Evaluación de AlfaMCT para los cuatro sismos de la tabla en la figura en función del número de iteraciones del numerador. Número de iteraciones de denominador fijado en cinco.

5.2.3 Detección de humo en video

En la literatura existen diferentes enfoques para la detección de humo, donde dos líneas de investigación se destacan. En la primera se analiza el comportamiento caótico de humo, su desplazamiento y evolución, mientras que en la segunda se realiza un análisis indirecto, analizando la distorsión del fondo de la imagen causada por la presencia del humo. En [Toreyin B, 2005] se emplean wavelets para cuantificar la pérdida de contenido de alta frecuencia debido a la presencia de humo, sumada a la ocurrencia temporal de la pérdida de fondo. Para evitar falsas detecciones se emplea fusión sensorial. En el trabajo [Torabnezhad M, 2013] se combina la información del espectro visible con el infrarrojo. La presencia de humo en su estado inicial degrada los bordes de la imagen de fondo. Esta característica de atenuación temporal es muy empleada como indicador humo en estado inicial ya que la extracción de bordes de una imagen es una técnica muy utilizada. Otra manera de caracterizar el movimiento del humo es a través del concepto de flujo óptico. En [Yu Chunyu, 2010] se utiliza el algoritmo de Lucas Kanade para la detección del humo combinado con una red neuronal *backpropagation* para discernir entre las distintas fuentes de flujo óptico. Una muy completa descripción de los métodos de detección de fuego y humo es realizada por [Çetin, 2016] donde se destacan fortalezas y debilidades de cada uno de los métodos más referenciados en la bibliografía.

La detección en espacios abiertos es aún un desafío por la variabilidad de la escena y la presencia de factores perturbadores como ser viento, polvo, movimiento y condiciones de iluminación no controlables y completamente variables. La detección del fuego es relativamente más sencilla que la del humo ya que la gama de colores que presenta es confinada a un subespacio que junto a las características espaciales, temporales y de textura logran una certera detección. El fuego presenta tonalidades entre el rojo, naranja y amarillo. Para la detección de color se emplea mayormente modelos de color que diferencian la crominancia de la luminancia. Estos modelos logran clasificadores lineales entre píxeles asociados a fuego y no

fuego. En la detección conjunta de fuego y humo, la fusión de descriptores es primordial para lograr disminuir los falsos positivos. Lamentablemente, la fusión de descriptores atenta contra la eficiencia y la determinación en tiempo real ya que varios procesos debe ejecutarse además de la lógica de fusión de las fuentes de información.

El método propuesto en esta tesis se diferencia, de los mencionada en este estado del arte, en el empleo de algoritmos basados en la información esencial de la imagen en un instante. Esta representación simplificada contiene descriptores que capturan la esencia de la información necesaria para la inferencia de regiones que pertenecen a humo o a fuego. La obtención de los descriptores genera una salida que es de órdenes de magnitud menor con respecto a la matriz imagen. Si bien, la información de entrada es un *frame* del video, el procesamiento puede realizarse por filas que es la forma habitual que una cámara digital utiliza. Implementado de esta forma, no es necesario almacenar en memoria un *frame* completo al mismo tiempo. A partir de los descriptores, que como ya se mencionó forman una estructura compacta, se puede almacenar el tipo de ocurrencia y su posición en el *frame*. Por lo tanto, el método descrito es adaptable para ejecutarse en sistemas embebidos con medianos requerimientos logrando un costo razonable.

En cada *frame* se obtienen los pixeles esenciales y se realizan los procesos que se observan en la figura 5.6

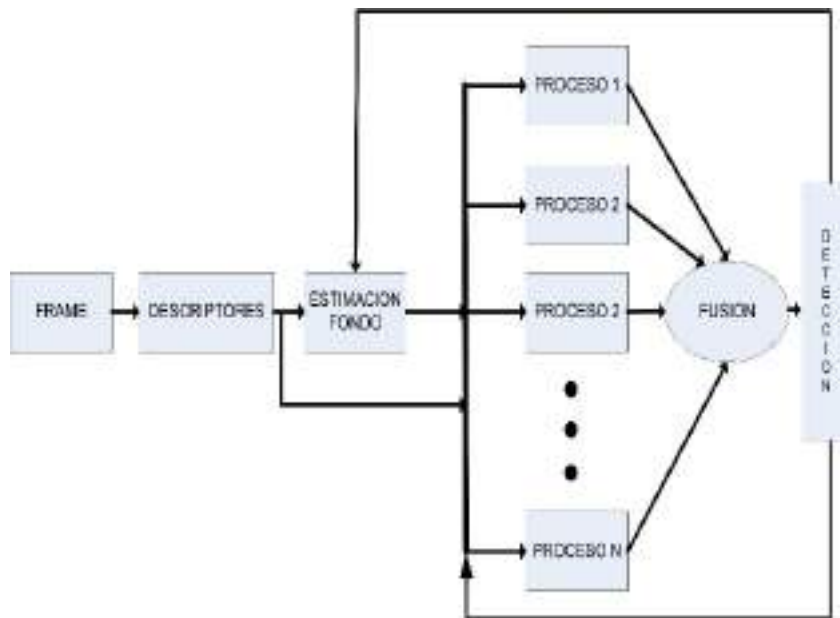


Figura 5.6 Diagrama en bloques propuesto para la detección. Se fusiona la información de N procesos, tomando como información de entrada la salida de los descriptores y la estimación del fondo.

El proceso clave que es la estimación de fondo, se realiza utilizando únicamente los píxeles esenciales. El humo es inferido por la manera que afecta a las recientes estimaciones del fondo. A través del humo el fondo es modulado.

Cada *frame* del video es subdividido en celdas de $M \times N$ píxeles. El humo es inferido como fusión de los siguientes procesos aplicados a cada celda:

1. Cambio en el valor promedio de los descriptores con respecto a la estimación de fondo.
2. Disminución del contenido en alta frecuencia de la celda observando la disminución de la varianza de valores de intensidad de los descriptores.
3. Cambio en la cantidad de descriptores en la celda.
4. Filtro de persistencia temporal.
5. Filtro de persistencia espacial.

Resultados preliminares de la detección de humo han sido publicados en [Monte, 2017]. Actualmente se están implementando estos procesos en un sistema embebido de bajo costo para lograr una red de sensores de detección de humo para espacios abiertos no supervisados. Si bien en [Çetin, 2016] se destacan diferentes métodos con sus fortalezas y debilidades, ninguno de ellos es comparable con la propuesta realizada en esta tesis ya que la única validación posible es estrictamente cualitativa. Esto se debe a que la forma de representar la información embebida se basa en una unión de trayectorias simplificadas minimizando la información redundante presente y disminuyendo fuertemente los procesos de extracción de conocimiento y consecuentemente los tiempos de procesamiento.

REFERENCIAS

- Amann A, Tratnig R and Unterkofler K S, 2005 A New Ventricular Fibrillation Detection Algorithm for Automated External Defibrillators, *Computers in Cardiology Applications*. John Wiley & Sons Ltd.
- Barro S, Ruiz R, Cabello D, and Mira J 1989 Algorithmic sequential decision-making in the frequency domain for life threatening ventricular arrhythmias and imitative artefacts: a diagnostic system. *J Biomed Eng*, 11(4):320–8.
- Beyrouthy, T., Fesquet, L., & Rolland, R. (2015). Data sampling and processing: Uniform vs. non-uniform schemes. 2015 International Conference on Event-Based Control, Communication, and Signal Processing (EBCASP).
- Candès E J. Wakin M. B. An introduction to Compressive Sampling.. *IEEE Signal Processing Magazine*. pp 21-30.2008.
- Çetin Enis, Bart Merci, Osman Günay, Behçet Uğur Töreyn, Steven Verstockt. “Methods and techniques for fire detection : signal, image and video processing perspectives”. London, UK : Academic Press.Elsevier, 2016.
- Chen S, Thakor N V, and Mower M M 1987 Ventricular fibrillation detection by a regression test on the autocorrelation function. *Med Biol Eng Comput*, 25(3):241–9.
- Cheng H Y, J B S, Tseng P T, et al. Lane detection with moving vehicles in the traffic scenes [J]. *IEEE transactions on intelligent transportation systems*, 2006, 7(4): 571-582. DOI:10.1109/tits.2006.883940.
- Clayton R H, Murray A and Campbell R W 1994 Recognition of Ventricular Fibrillation Using Neural Networks, *Medical and Biological Engineering and Computing*, Vol 32(2) 217-220.
- D. Ben-Tzvi and M.B. Sandler. A combinatorial Hough Transform. *Pattern Recognition Letters*, 11(3):167–174, March 1990.

F. Abate, A. Espírito-Santo, G. Monte and V. Paciello, "Smart Sensor Efficient Signal Processing for Earthquake Early Detection," 2019 IEEE International Symposium on Measurements & Networking (M&N), Catania, Italy, 2019, pp. 1-6.

G. Monte, J. I. Pastore, V. Ballarin, D. Marasco and P. Liscovsky, "Smoke detection using simplified descriptors of video information," 2017 IEEE International Conference on Industrial Technology (ICIT), Toronto, ON, 2017, pp. 1070-1075.

Geddes L A and Roeder R A 2002 The First Electronic Electrocardiograph Cardiovascular Engineering Vol 2 Number 2 pp 73-79.

Governors Highway Safety Association. "Distracted Driving: What Research Shows and What States Can Do".2006.

Governors Highway Safety Association. "Distracted Driving: What Research Shows and What States Can Do".2006.

G.T. Fan, B. Li, Q. Han, et al., "Robust lane detection and tracking based on machine vision," ZTE Communications, vol. 18, no. 4, pp. 69–77, Dec. 2020. doi: 10.12142/ZTECOM.202004010.

H. A. Rahim, U. U. Sheikh, R. B. Ahmad and A. S. M. Zain. "Vehicle Velocity Estimation for Traffic Surveillance System". World Academy of Science, Engineering and Technology Vol:4 2010-09-22.

Heikki Kalviainen, Petri Hirvonen, Lei Xu, Erkki Oja, Probabilistic and non-probabilistic Hough transforms: overview and comparisons, Image and Vision Computing, Volume 13, Issue 4, May 1995, Pages 239-252, ISSN 0262-8856, DOI: 10.1016/0262-8856(95)99713-B.

http://ds.iris.edu/wilber3/find_event.

<http://www.americanheart.org/presenter.jhtml?identifier=4784>.

http://www.cesvi.com.ar/SeguridadVial/Novidades/Seguridad_distraccion.aspx.

<http://www.flir.com/cores/content/?id=66257>

Jayaraman; et al. (2009). Digital Image Processing. Tata McGraw Hill Education. p. 272. ISBN 9781259081439.

- Jianlong fu, Yong Rui. (2017) Advances in deep learning approaches for image tagging *Microsoft Research, No. 5*, Dan Ling Street, Haidian District, Beijing, P. R. China.
- Karen M. Guan,¹ Suleyman S. Kozat,² and Andrew C. Singer¹ (2008). Adaptive Reference Levels in a Level-Crossing Analog-to-Digital Converter *EURASIP Journal on Advances in Signal Processing* Volume 2008, Article ID 513706.
- Kass, M.; Witkin, A.; Terzopoulos, D. (1988). Snakes: Active contour models. *International Journal of Computer Vision*.
- Ling-fei Mo ; Chun-fang Qin ; Xiao-fen Tang “Velocity Analysis for UHF RFID Vehicle License Plate” *Optoelectronics and Image Processing (ICOIP)*, 2010 International Conference on Volume: 2: 2010 pp. 722 – 725.
- Lucas B. and T. Kanade, An iterative image registration technique with an application to stereo vision. *Proceedings of Imaging Understanding Workshop*. 1981, pages 121--130.
- Yu Chunyu, Fang Jun, Wang Jinjun, Zhang Yongming Video Fire Smoke Detection Using Motion and Color Features.. *Fire Technology*. July 2010, Volume 46, Issue 3, pp 651-663
- Lyons Richard (2011). *Understanding Digital Signal Processing* (3rd Edition). Prentice Hall.
- Madapura J Murthy “Achieving Higher ADC Resolution Using Oversampling” AN1152, Microchip Technology Inc. 2008.
- Maduro, C. ; Batista, K. ; Peixoto, P. ; Batista, “Estimation of vehicle velocity and traffic intensity using rectified images” *J. Image Processing*, 2008. *ICIP 2008*. 15th IEEE International Conference on DOI: 10.1109/ICIP.2008.4711870, 2008 , pp. 777 – 780.
- Mandic, D., Golz, M., Kuh, A., Obradovic, D., & Tanaka, T. (Eds.). (2008). *Signal Processing Techniques for Knowledge Extraction and Information Fusion*.
- Marten D. van der Laan. *Signal sampling techniques for data acquisition in process Control*. Thesis Rijksuniversiteit Groningen. - 1995. ISBN 90-367-0502-9.
- Marvasti F. *Nonuniform Sampling, Theory and Practice* (2000). Ed., Kluwer Academic/Plenum Publishers, New York.

Marvasti F., Nonuniform Sampling, Theory and Practice, Kluwer Academic/Plenum Publishers, New York, 2000.

Massachusetts Institute of Technology, MIT-BIH arrhythmia database
<http://www.physionet.org/physiobank/database/mitdb>.

Mehmet Sezgin, Bulent Sankur "Survey over image thresholding techniques and quantitative performance evaluation" Journal of Electronic Imaging 13(1), 146–165. January 2004.

Mehmet Sezgin, Bulent Sankur "Survey over image thresholding techniques and quantitative performance evaluation" Journal of Electronic Imaging 13(1), 146–165. January 2004.

Mertins Alfred (1999) Signal Analysis: Wavelets, Filter Banks, Time-Frequency Transform.

Mohsin Nazir, (2012) Cloud Computing: Overview & Current Research Challenges IOSR Journal of Computer Engineering (IOSR-JCE) ISSN: 2278-0661, ISBN: 2278-8727 Volume 8, Issue 1 (Nov. - Dec. 2012).

Monte, G. "Sensor Signal Preprocessing Techniques for Analysis and Prediction" Industrial Electronics, 2008. IECON 2008. 34th Annual Conference of IEEE. pp 1788-1793. ISBN 978-114244-1766-7.

Nyquist, Harry (1928). Certain Topics in Telegraph Transmission Theory. Transactions of the A. I. E. E. 47: 617-644.

Oppenheim A, Schafer R. (2013) Discrete-Time Signal Processing: Pearson. International Edition.

Pan J and Tompkins W J. 1985 A real-time QRS detection algorithm. IEEE Trans Biomed Eng, 32(3):230–6.

Poli R, Cagnoni S and Valli, G 1995 Genetic design of optimum linear and nonlinear QRS detectors, IEEE Transaction On Biomedical Engineering, Vol. 42, pp. 1137-1141.

Quinquis André (2008). Digital Signal Processing using MATLAB. Wiley 2008.

S. M. Szilagy, Z. Benyo, L. Szilagy and L David 2003 Adaptive wavelet transform based ECG waveforms detection, Proceedings of 25th Annual IEEE EMBS International Conference, No. 24, pp. 12-15.

Sayiner N., Viswanathan T.R., "A Level- Crossing Sampling Scheme for A/D Conversion", IEEE Transactions on Circuits and Systems II, vol. 43, pp. 335-339, April 1996.

Siemens."Soluciones inteligentes para el tráfico de hoy y del mañana " <https://www.swe.siemens.com/spain/web/es/industry/mobility/Documents/traffic.pdf>.

Songqing Chen, Tao Zhang, Weisong Shi, (2017). Fog Computing. IEEE Internet Computing Volume: 21, Issue: 2, Mar.-Apr. 2017.

Thakor N V, Zhu Y S, and Pan K Y 1990 Ventricular tachycardia and fibrillation detection by a sequential hypothesis testing algorithm. IEEE Transaction on Biomedical Engineering, 37(9):837–43.

Toan Minh Hoang, Na Rae Baek, Se Woon Cho, Ki-Wan Kim, Kang Ryoung Park: "Road Lane Detection Robust to Shadows Based on a Fuzzy System Using a Visible Light Camera Sensor". Sensors 17(11): 2475 (2017)

Torabnezhad M, Ali Aghagolzadeh, HadiSeyedarabi, "Visible and IR image fusion algorithm for short range smoke detection," IEEE International Conference on Robotics and Mechatronics, 2013.

Toreyin B, Yigithan Dedeoglu, Cetin A. Enis. Wavelet based real-time smoke detection in video. 13th European Signal Processing Conference (EUSIPCO2005).Antalya, Turkey: Curran Associates, 2005.

Valentinuzzi M E 2010 Cardiac Fibrillation-Defibrillation: Clinical and Engineering Aspects. In Series on Bioengineering and Biomedical Eng. World Scientific Publishers, Singapore.

Viola, P., & Jones, M. (2001). Robust real-time face detection. Proceedings Eighth IEEE International Conference on Computer Vision. ICCV.

Weisong Shi, Jie Cao, Lanyu Xu (2016) Edge Computing: Vision and Challenges. IEEE Internet of Things Journal.

Willett, R. (2017). Signal representations in modern signal processing. 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).

Yamamoto S. , Rydelek P. , Horiuchi S., Wu C. , Nakamura H., “On the estimation of seismic intensity in earthquake early warning systems,” *Geophysical Research Letters*, vol. 35, n. 7, April 2008, doi:10.1029/2007GL033034 .

ANEXO A

Estandar IEEE 21451-001-2017

(resumen)

IEEE Recommended Practice for Signal Treatment Applied to Smart Transducers

IEEE Industrial Electronics Society

Sponsored by the
Industrial Electronics Society Standards Committee

IEEE Instrumentation and Measurement Society

Sponsored by the
Sensor Technology Committee
and the
Signals and Systems in Measurement Committee

IEEE
3 Park Avenue
New York, NY 10016-5997
USA

IEEE Std 21451-001™-2017

IEEE Recommended Practice for Signal Treatment Applied to Smart Transducers

Sponsors

**Industrial Electronics Society Standards
Committee** of the
IEEE Industrial Electronics

Society and

Sensor Technology Committee
and
**Signals and Systems in Measurement
Committee** of the
IEEE Instrumentation and Measurement Society

Approved 23 March 2017

IEEE-SA Standards Board

Abstract: Signal processing algorithms and data structure are defined in this recommended practice in order to share and to infer signal and state information of an instrumentation or control system. These algorithms are based on the signal and transducers attached to the system. This recommended practice also defines a set of transducer signal treatment services based on signal treatment algorithms as application programming interface (API), which is used for applications to use or call these transducer signal treatment services.

Keywords: IEEE 1451™, IEEE 21451™, oversampling, segmentation, serial, signal shape analysis, smart actuator, smart filtering, smart sensor, smart transducer, time-domain signal analysis

The Institute of Electrical and Electronics Engineers, Inc.
3 Park Avenue, New York, NY 10016-5997, USA

Copyright © 2017 by The Institute of Electrical and Electronics Engineers, Inc.
All rights reserved. Published 20 December 2017. Printed in the United States of America.

IEEE is a registered trademark in the U.S. Patent & Trademark Office, owned by The Institute of Electrical and Electronics Engineers, Incorporated.

CORBA is a registered trademark of Object Management Group, Inc. in the United States and/or other countries.

Java is a registered trademark of Oracle and/or its affiliates.

Microsoft, Visual C++, Visual C#, Visual Studio, and Windows are registered trademarks of Microsoft Corporation in the United States and/or other countries.

PDF: ISBN 978-1-5044-4054-7 STD22608
Print: ISBN 978-1-5044-4055-4 STDPD22608

IEEE prohibits discrimination, harassment, and bullying.

For more information, visit <http://www.ieee.org/web/aboutus/whatis/policies/p9-26.html>.

No part of this publication may be reproduced in any form, in an electronic retrieval system or otherwise, without the prior written permission of the publisher.

Important Notices and Disclaimers Concerning IEEE Standards Documents

IEEE documents are made available for use subject to important notices and legal disclaimers. These notices and disclaimers, or a reference to this page, appear in all standards and may be found under the heading “Important Notices and Disclaimers Concerning IEEE Standards Documents.” They can also be obtained on request from IEEE or viewed at <http://standards.ieee.org/IPR/disclaimers.html>.

Notice and Disclaimer of Liability Concerning the Use of IEEE Standards Documents

IEEE Standards documents (standards, recommended practices, and guides), both full-use and trial-use, are developed within IEEE Societies and the Standards Coordinating Committees of the IEEE Standards Association (“IEEE-SA”) Standards Board. IEEE (“the Institute”) develops its standards through a consensus development process, approved by the American National Standards Institute (“ANSI”), which brings together volunteers representing varied viewpoints and interests to achieve the final product. IEEE Standards are documents developed through scientific, academic, and industry-based technical working groups. Volunteers in IEEE working groups are not necessarily members of the Institute and participate without compensation from IEEE. While IEEE administers the process and establishes rules to promote fairness in the consensus development process, IEEE does not independently evaluate, test, or verify the accuracy of any of the information or the soundness of any judgments contained in its standards.

IEEE Standards do not guarantee or ensure safety, security, health, or environmental protection, or ensure against interference with or from other devices or networks. Implementers and users of IEEE Standards documents are responsible for determining and complying with all appropriate safety, security, environmental, health, and interference protection practices and all applicable laws and regulations.

IEEE does not warrant or represent the accuracy or content of the material contained in its standards, and expressly disclaims all warranties (express, implied and statutory) not included in this or any other document relating to the standard, including, but not limited to, the warranties of: merchantability; fitness for a particular purpose; non-infringement; and quality, accuracy, effectiveness, currency, or completeness of material. In addition, IEEE disclaims any and all conditions relating to: results; and workmanlike effort. IEEE standards documents are supplied “AS IS” and “WITH ALL FAULTS.”

Use of an IEEE standard is wholly voluntary. The existence of an IEEE standard does not imply that there are no other ways to produce, test, measure, purchase, market, or provide other goods and services related to the scope of the IEEE standard. Furthermore, the viewpoint expressed at the time a standard is approved and issued is subject to change brought about through developments in the state of the art and comments received from users of the standard.

In publishing and making its standards available, IEEE is not suggesting or rendering professional or other services for, or on behalf of, any person or entity nor is IEEE undertaking to perform any duty owed by any other person or entity to another. Any person utilizing any IEEE Standards document, should rely upon his or her own independent judgment in the exercise of reasonable care in any given circumstances or, as appropriate, seek the advice of a competent professional in determining the appropriateness of a given IEEE standard.

IN NO EVENT SHALL IEEE BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO: PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE PUBLICATION, USE OF, OR RELIANCE UPON ANY STANDARD, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE AND REGARDLESS OF WHETHER SUCH DAMAGE WAS FORESEEABLE.

Translations

The IEEE consensus development process involves the review of documents in English only. In the event that an IEEE standard is translated, only the English version published by IEEE should be considered the approved IEEE standard.

Official statements

A statement, written or oral, that is not processed in accordance with the IEEE-SA Standards Board Operations Manual shall not be considered or inferred to be the official position of IEEE or any of its committees and shall not be considered to be, or be relied upon as, a formal position of IEEE. At lectures, symposia, seminars, or educational courses, an individual presenting information on IEEE standards shall make it clear that his or her views should be considered the personal views of that individual rather than the formal position of IEEE.

Comments on standards

Comments for revision of IEEE Standards documents are welcome from any interested party, regardless of membership affiliation with IEEE. However, IEEE does not provide consulting information or advice pertaining to IEEE Standards documents. Suggestions for changes in documents should be in the form of a proposed change of text, together with appropriate supporting comments. Since IEEE standards represent a consensus of concerned interests, it is important that any responses to comments and questions also receive the concurrence of a balance of interests. For this reason, IEEE and the members of its societies and Standards Coordinating Committees are not able to provide an instant response to comments or questions except in those cases where the matter has previously been addressed. For the same reason, IEEE does not respond to interpretation requests. Any person who would like to participate in revisions to an IEEE standard is welcome to join the relevant IEEE working group.

Comments on standards should be submitted to the following address:

Secretary, IEEE-SA Standards Board
445 Hoes Lane
Piscataway, NJ 08854 USA

Laws and regulations

Users of IEEE Standards documents should consult all applicable laws and regulations. Compliance with the provisions of any IEEE Standards document does not imply compliance to any applicable regulatory requirements. Implementers of the standard are responsible for observing or referring to the applicable regulatory requirements. IEEE does not, by the publication of its standards, intend to urge action that is not in compliance with applicable laws, and these documents may not be construed as doing so.

Copyrights

IEEE draft and approved standards are copyrighted by IEEE under U.S. and international copyright laws. They are made available by IEEE and are adopted for a wide variety of both public and private uses. These include both use, by reference, in laws and regulations, and use in private self-regulation, standardization, and the promotion of engineering practices and methods. By making these documents available for use and adoption by public authorities and private users, IEEE does not waive any rights in copyright to the documents.

Photocopies

Subject to payment of the appropriate fee, IEEE will grant users a limited, non-exclusive license to photocopy portions of any individual standard for company or organizational internal use or individual, non-commercial use only. To arrange for payment of licensing fees, please contact Copyright Clearance Center, Customer Service, 222 Rosewood Drive, Danvers, MA 01923 USA; +1 978 750 8400. Permission to photocopy portions of any individual standard for educational classroom use can also be obtained through the Copyright Clearance Center.

Updating of IEEE Standards documents

Users of IEEE Standards documents should be aware that these documents may be superseded at any time by the issuance of new editions or may be amended from time to time through the issuance of amendments, corrigenda, or errata. An official IEEE document at any point in time consists of the current edition of the document together with any amendments, corrigenda, or errata then in effect.

Every IEEE standard is subjected to review at least every ten years. When a document is more than ten years old and has not undergone a revision process, it is reasonable to conclude that its contents, although still of some value, do not wholly reflect the present state of the art. Users are cautioned to check to determine that they have the latest edition of any IEEE standard.

In order to determine whether a given document is the current edition and whether it has been amended through the issuance of amendments, corrigenda, or errata, visit the IEEE Xplore at <http://ieeexplore.ieee.org/> or contact IEEE at the address listed previously. For more information about the IEEE-SA or IEEE's standards development process, visit the IEEE-SA Website at <http://standards.ieee.org>.

Errata

Errata, if any, for all IEEE standards can be accessed on the IEEE-SA Website at the following URL: <http://standards.ieee.org/findstds/errata/index.html>. Users are encouraged to check this URL for errata periodically.

Patents

Attention is called to the possibility that implementation of this standard may require use of subject matter covered by patent rights. By publication of this standard, no position is taken by the IEEE with respect to the existence or validity of any patent rights in connection therewith. If a patent holder or patent applicant has filed a statement of

assurance via an Accepted Letter of Assurance, then the statement is listed on the IEEE-SA Website at <http://standards.ieee.org/about/sasb/patcom/patents.html>. Letters of Assurance may indicate whether the Submitter is willing or unwilling to grant licenses under patent rights without compensation or under reasonable rates, with reasonable terms and conditions that are demonstrably free of any unfair discrimination to applicants desiring to obtain such licenses.

Essential Patent Claims may exist for which a Letter of Assurance has not been received. The IEEE is not responsible for identifying Essential Patent Claims for which a license may be required, for conducting inquiries into the legal validity or scope of Patents Claims, or determining whether any licensing terms or conditions provided in connection with submission of a Letter of Assurance, if any, or in any licensing agreements are reasonable or non-discriminatory. Users of this standard are expressly advised that determination of the validity of any patent rights, and the risk of infringement of such rights, is entirely their own responsibility. Further information may be obtained from the IEEE Standards Association.

Participants

At the time this recommended practice was completed, the Signal Treatment for Smart Transducer Working Group had the following membership:

Gustavo Monte, *Chair*

Ruqiang Yan, *Vice Chair*

Francesco Abate
Allen Chen
Antonio Espirito Santo
Sabrina Guia

Rodica Holonec
Victor Huang
Kang Lee
Zheng Liu
Vincenzo Paciello

Antonio Pietrosanto
Helena Ramos
Eugene Song
Hubert Zangl

The following members of the individual balloting committee voted on this recommended practice. Balloters may have voted for approval, disapproval, or abstention.

Mihaela Albu
Frank Basciano
William Bloethe
Riccardo Brama
Dietmar Bruckner
Paul Cardinal
Cheng-Jen Chen
Keith Chow
Helena Maria Geirinhas Ramos
Randall Groves
Werner Hoelzl
Rodica Holonec
Victor Huang

Atsushi Ito
Robert Johnson
Kenneth Karg
John Kay
Kang Lee
Michael Maytum
Tremont Miao
Gustavo Monte
Arthur Neubauer
Michael Newman
Vincenzo Paciello
Sergio Rapuano

Robert Robinson
Bartien Sayogo
Christopher Searles
Robert Seitz
Suresh Shrimavle
Jeremy Smith
Yuyin Song
Joseph Stanco
Thomas Strasser
John Vergis
Stephen Webb
Ruqiang Yan
Oren Yuen

When the IEEE-SA Standards Board approved this recommended practice on 23 March 2017, it had the following membership:

Jean-Philippe Faure, *Chair*
Gary Hoffman, *Vice Chair*
John D. Kulick, *Past Chair*
Konstantinos Karachalios, *Secretary*

Chuck Adams
Masayuki Ariyoshi
Ted Burse
Stephen Dukes
Doug Edwards
J. Travis Griffith
Michael Janezic

*Member Emeritus

Thomas Koshy
Joseph L. Koepfinger*
Kevin Lu
Daleep Mohla
Damir Novosel
Ronald C. Petersen
Annette D. Reilly

Robby Robson
Dorothy Stanley
Adrian Stephens
Mehmet Ulema
Phil Wennblom
Howard Wolfman
Yu Yuan

Introduction

This introduction is not part of IEEE Std 21451-001-2017, IEEE Recommended Practice for Signal Treatment Applied to Smart Transducers.

IEEE Std 21451-001, a recommended practice, defines signal processing algorithms and data structure in order to share and infer signal and state information of instrumentation or control systems. These algorithms are based on acquired signals and the transducers attached to the systems. This recommended practice also defines the commands and responses for requesting information and algorithms for shape analysis, such as exponential, sinusoidal, impulsive noise, noise, and tendency.

IEEE Recommended Practice for Signal Treatment Applied to Smart Transducers

1. Overview

1.1 Scope

This recommended practice defines signal processing algorithms and data structure in order to share and to infer signal and state information of an instrument or control system. These algorithms are based on their own signal and also on the transducers attached to the system. The recommended practice also defines the commands and replies for requesting information and algorithms for shape analysis such as exponential, sinusoidal, impulsive noise, noise, and tendency.

1.2 Purpose

The purpose is to define a standardized and universal framework that allows smart transducers to extract features of the signal being generated and measured. With the definition of these practices, the raw data can be converted into information and then into knowledge. In this context, knowledge means understanding of the nature of the transducer signal. This understanding can be shared with the system and other transducers in order to form a platform for sensory knowledge fusion.

1.3 Conformance

Conformance to IEEE Std 21451-001™ requires that all non-optional clauses be implemented in the vendor device.

Several keywords are used to differentiate among various levels of requirements and optionally, as follows:

shall: The key word “shall” indicates a mandatory requirement. Designers are required to implement all such mandatory requirements to promote interoperability with other products that conform to the specifications for IEEE Std 21451-001.

recommended: Recommended is a key word indicating flexibility of choice with a strong preference alternative. The word “should” has the same meaning.

should: Should is a key word indicating flexibility of choice with a strong preference alternative. The phrase “it is recommended” has the same meaning. **may:** May is a key word that indicates flexibility of choice with no implied preference.

2. Normative references

The following referenced documents are indispensable for the application of this document (i.e., they must be understood and used; therefore, each referenced document is cited in text and its relationship to this document is explained). For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments or corrigenda) applies.

IEEE P1451.2/D32 (April 2016), Draft Standard for a Smart Transducer Interface for Sensors and Actuators—Serial Point-to-Point Interface.¹

IEEE Std 1451.5™-2007, IEEE Standard for a Smart Transducer Interface for Sensors and Actuators—Wireless Communication Protocols and Transducer Electronic Data Sheet (TEDS) Formats.^{2,3}

ISO/IEC/IEEE 21450:2010, Information technology — Smart transducer interface for sensors and actuators — Common Functions, Communication Protocols, and Transducer Electronic Data Sheet (TEDS) formats.⁴

ISO/IEC/IEEE 21451-7:2011, Information technology — Smart transducer interface for sensors and actuators — Part 7: Transducer to radio frequency identification (RFID) systems communication protocols and Transducer Electronic Data Sheet (TEDS) formats.

3. Definitions, abbreviations, and units

3.1 Definitions

For the purposes of this document, the following terms and definitions apply. The *IEEE Standards Dictionary Online* should be consulted for terms not defined in this clause.⁵ **impulsive noise:** Interference on a signal caused by a short period of noise.

¹ Numbers preceded by P are IEEE authorized standards projects that were not approved by the IEEE-SA Standards Board at the time this publication went to press. For information about obtaining drafts, contact the IEEE.

² The IEEE standards or products referred to in this clause are trademarks owned by The Institute of Electrical and Electronics Engineers, Incorporated.

³ IEEE publications are available from The Institute of Electrical and Electronics Engineers (<http://standards.ieee.org/>).

⁴ ISO/IEC publications are available from the International Organization for Standardization (<http://www.iso.org/>), the International Electrotechnical Commission (<http://www.iec.ch>), and the American National Standards Institute (<http://www.ansi.org/>).

⁵ *IEEE Standard Dictionary Online* is available at <http://dictionary.ieee.org>.

oversampling: A sampling frequency that is much greater than the Nyquist frequency required to avoid aliasing in uniform sampling. **tendency:** Long-term behavior of a transducer signal.

uniform sampling: In a continuous-time to discrete-time signal conversion, when the time interval between samples is constant.

3.2 Abbreviations

CORBA® Common Object Request Broker Architecture^{6, 7}

CPU	central processing unit
ECG	electrocardiogram
IDL	Interface Definition Language
LSB	least significant bit
MCT	mark, class, and time (i.e., the three vectors that represent the sampled transducer signal)
NCAP	network-capable application processor
POM	point of measurement (in a sensor)
QRS	a wave pattern related to electrocardiograms (i.e., wave Q, wave R, and wave S)
RTSAL	real-time segmentation and labeling
TADS	transducer algorithm data sheet
TEDS	transducer electronic data sheet (see IEEE Std 1451.0™-2007 [B3], 3.1.37) ⁸
TIM	transducer interface module (see IEEE Std 1451.0-2007, 3.1.33)

3.3 Units

UInt32	unsigned 32-bit integer
UInt16	unsigned 16-bit integer
UInt8	unsigned 8-bit integer
Float32	a single-precision real number, i.e., a 32-bit binary sequence
Float64	a double-precision real number, i.e., a 64-bit binary sequence
_String	string data type
_Boolean	Boolean data type
UInt32Array	an arbitrary number of 32-bit signed integers
UInt16Array	an arbitrary number of 16-bit signed integers
UInt8Array	an arbitrary number of 16-bit signed integers

⁶ CORBA is a registered trademark of Object Management Group, Inc. in the United States and/or other countries.

⁷ This information is given for the convenience of users of this standard and does not constitute an endorsement by the IEEE of these products. Equivalent products may be used if they can be shown to lead to the same results.

⁸ The numbers in brackets correspond to the numbers of the bibliography in Annex A.

Float32Array an arbitrary number of single-precision real numbers
Float64Array an arbitrary number of double-precision real numbers
StringArray an arbitrary number of `_String` data types
BooleanArray an arbitrary number of `_Boolean` data types

4. Need for the recommended practice

4.1 Introduction

The real world is inferred through sensor signals. The sensors and actuators can do much more than what humans usually use them for, i.e., performing their basic functions. The expanded application of embedded systems has direct impact on transducer design. The all-inclusive microcontrollers and the application-specific integrated circuits offer everything needed to turn a regular transducer into a smart transducer, with the integration of a sensor element into a single unit. General sensor signal processing today is basically the same as 10 years ago. While the processing power and memory size of the central processing unit (CPU) have grown exponentially, the costs of sensor signal processing remain relatively constant. Taking into account the size, power consumption, cost, etc., a paradigm shift in the sensor world is on the horizon: the signal will be processed entirely at the point of measurement (POM). Although today this is only relevant to sensors with complex signal processing related to specific issues, it will occur for most transducers in the future, even for small and low-cost sensors. However, this issue is out of the scope of this recommended practice.

No universal treatment for transducer signals allows signal synthesis and analysis and a dialogue among transducers to validate the measurement processes and to provide a platform for data mining. This recommended practice allows smart transducers from different manufacturers to interchange signal information in order to infer states of the instrumentation system, and such capability helps to achieve higher reliability and better real-world inference.

Once the sensor signal is digitized, a great amount of data will be generated. If there are dozens or hundreds of sensors, it may be desirable not to interchange these data because, most of the time, there is no need to do it. Besides, the data flow is restricted by the transmission channel capacity. Since the trend is to process the signal in the POM instead of raw data, there will be a need to interchange knowledge of the sensor signal. Therefore, taking sensor fusion into account, there is a requirement for processing these data from the transducer to obtain data and then convert them into knowledge. For some specific applications, this knowledge can be

obtained directly from samples. For multiple objectives, an intermediary platform is needed to simplify and speed up the signal processing as illustrated in Figure 1. This signal treatment should be carried out in a normalized scheme in order to allow interchange of knowledge among transducers. In this context, knowledge means understanding of the nature of the transducer signal. *This understanding must be shared with the system and transducers in order to build a platform for sensory knowledge fusion.* This point is the main objective of this recommended practice.

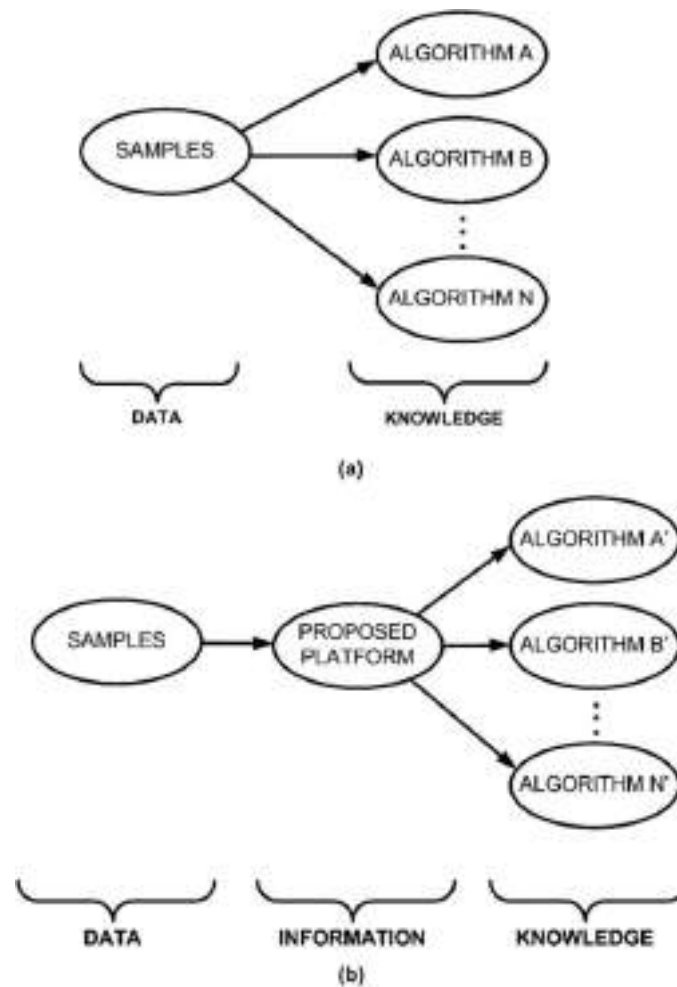


Figure 1 —Intermediary platforms. (a) Classical approach – from data to knowledge. (b) IEEE 21451-001 scheme – from data to information to knowledge.

4.2 A Simple example of application

A simple example may help to understand the aforementioned concepts. For example, a tank contains an actuator, e.g., a heater, and a temperature sensor. As part of the process, the temperature of the liquid inside the tank must be raised. Therefore, the actuator receives a command to turn on the heater. If everything is correct, the heater resistance is in range, there is liquid in the tank, and the temperature sensor works properly. The liquid temperature will start to increase, and it will be measured by the sensor. A simple question with a yes/no answer may validate the whole process. The actuator asks the sensor if the temperature signal exhibits an exponential growth. If the answer is a yes, the normal function of the entire system, i.e., actuator-liquid-sensor, is validated. If not, a problem is detected immediately after it occurs. The traditional way to detect an abnormal function is to trigger an alarm due to a timeout. If the actuator is a brand A and the temperature sensor is a brand B, a standardized sensor signal treatment is necessary to make this example application a reality. In addition, since the temperature sensor knows its signal shape, it can predict the steady-state value or the time to reach a desired value.

4.3 Outline of the recommended practice

Standard-based approaches offer many advantages over proprietary solutions, especially when the application is universal. What kind of knowledge should be included in this recommended practice? As a first approximation, it should be basic information of sensor signal shape, impulsive noise, oscillations, noise, mean, and tendency. The aim is to include similar information that may be provided for an observer to monitor the signal. The following clauses present the approach:

- Clause 5 describes the functional specification of smart transducers.
- Clause 6 presents the algorithm description with emphasis on the universal treatment that can be applied to all sensor signals. These algorithms implement a smart sampling process. The segmentation and labeling process is to obtain the raw information to infer a global behavior and to extract important features of the signal.
- Clause 7 presents the recommended practice structure that includes representative signal shape parameters.
- Clause 8 presents the first layer of the advanced feature extraction that may be embedded into a smart transducer.
- Clause 9 describes the services required from the first layer algorithms.
- Clause 10 presents the prediction algorithm.
- Clause 11 describes the algorithm for filtering.
- Clause 12 presents the compression algorithm.

- Clause 13 presents the algorithm to be implemented for a complex pattern analysis.
- Clause 14 describes the capability to load a user-defined application code.
- Clause 15 presents the structure for a transducer algorithm data sheet (TADS).
- Clause 16 describes the application programming interface (API).
- Clause 17 provides guidance for implementing the set of standardized transducer signal treatment services defined in this recommended practice.

For the convenience of the reader, variable names have been italicized in this document.

5. Smart transducer functional specification

5.1 ISO/IEC/IEEE 21451 reference model

This clause provides a description of the functions to be included in ISO/IEC/IEEE 21451 smart transducers. The reference model presented in Figure 2 shows the relationships among the various documents of the ISO/IEC/IEEE 21451 family of standards. The model is used to show the relationship between each standard and to define what is in each of the standards. It is not intended to imply a required implementation of devices that comply with the standards. Subclauses 5.2 through 5.10 give a further description of the elements shown in Figure 2.

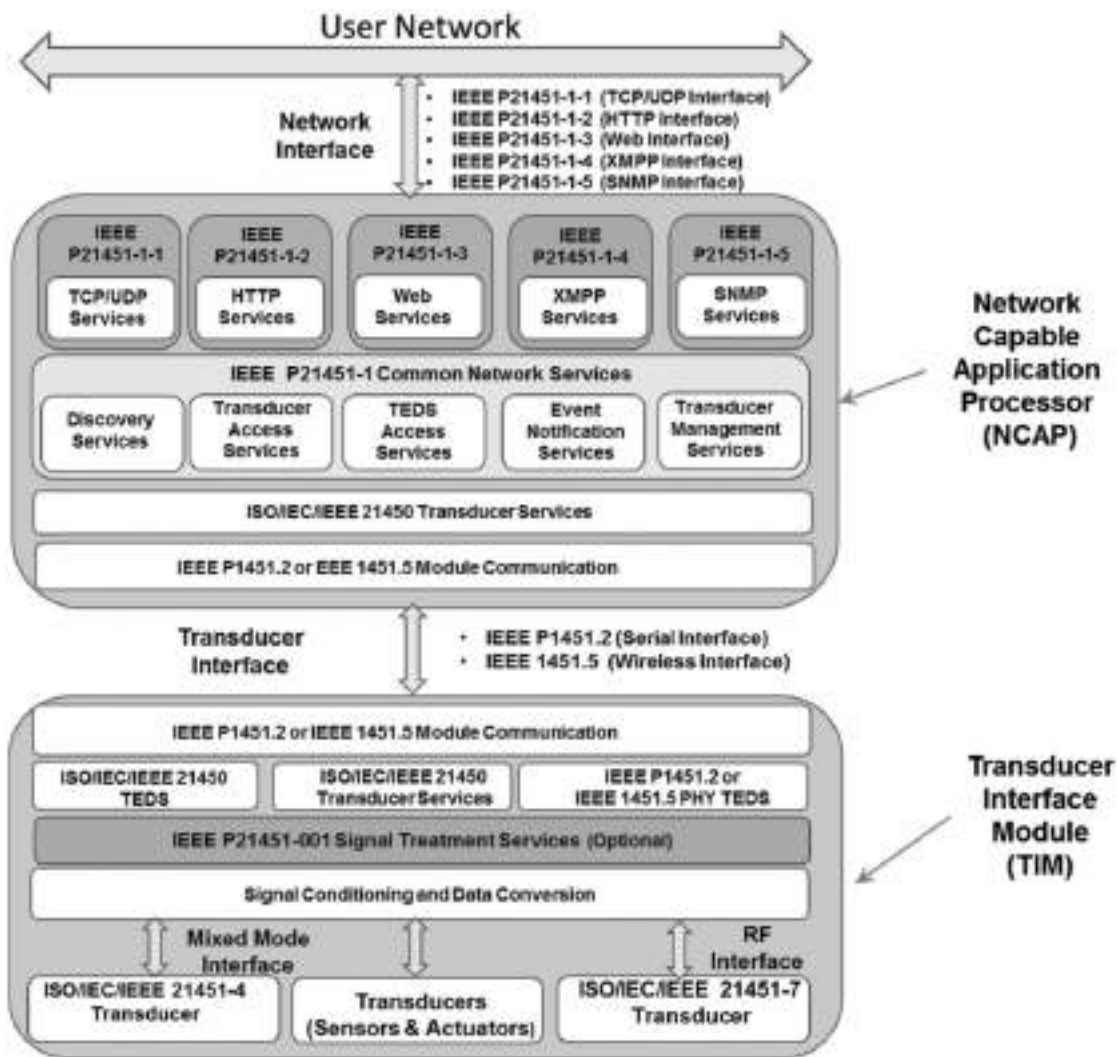


Figure 2 —ISO/IEC/IEEE 21451 family reference model

5.2 User's network

The user's network is beyond the scope of the ISO/IEC/IEEE 21451 family of standards. The networkcapable application processor (NCAP), shown in Figure 2, has the network interface. Different NCAPs are required for networks using different physical communications media or protocols. The network types are not defined in ISO/IEC/IEEE 21451.

5.3 NCAP network interface

IEEE P21451-1 [B1], shown in Figure 2, defines a set of common network services for ISO/IEC/IEEE 21451 smart transducers. The services consist of discovery

services, transducer data access services, transducer electronic data sheet (TEDS) access services, event notification services, and transducer manager services in order to achieve sensor data interoperability at the network level. The NCAP network interface shown in Figure 2 includes IEEE P21451-1-1, IEEE P21451-1-2, IEEE P21451-1-3, IEEE P21451-1-4 [B2], and IEEE P21451-1-5 services, which implement IEEE P21451-1 common network services in IP, HTTP, Web Services, XMPP, and SNMP services, respectively.

5.4 ISO/IEC/IEEE 21450 NCAP transducer services

The ISO/IEC/IEEE 21450 transducer services are common services for transducers. These functions include the command/response sets and TEDS. Features are included in the standard to support capabilities, such as triggering and synchronous sampling of sensors, but detailed specification is defined in each relevant document of the family of standards.

5.5 ISO/IEC/IEEE 21450/IEEE 1451.X NCAP module communication interface

The module communication interface is used to pass information across the interface between the ISO/IEC/IEEE 21450 NCAP services and the IEEE 1451.X NCAP module communication. IEEE P1451.2 and IEEE Std 1451.5-2007 are examples of the IEEE 1451.X standards.⁹

5.6 Physical interface between NCAP and TIM

The physical interface between the NCAP and the transducer interface module (TIM) is defined in the other documents of the IEEE 1451.X family of standards. Functions needed across this interface, such as synchronization, if applicable, are described, but not their implementation. IEEE P1451.2 and IEEE Std 1451.5-2007 are examples of the IEEE 1451.X standards.

5.7 ISO/IEC/IEEE 21450/IEEE 1451.X TIM communication interface

The ISO/IEC/IEEE 21450/IEEE 1451.X TIM communication interface is logically the complement of the ISO/IEC/IEEE 21450/IEEE 1451.X NCAP module communication interface. The ISO/IEC/IEEE 21450/ IEEE 1451.X TIM communication interface is defined in the corresponding documents of the IEEE

⁹ Information on normative references can be found in Clause 2.

1451.X family of standards. IEEE P1451.2 and IEEE Std 1451.5-2007 are examples of the IEEE 1451.X standards.

5.8 ISO/IEC/IEEE 21450 TIM transducer services

The ISO/IEC/IEEE 21450 TIM transducer services are logically the complement of the ISO/IEC/IEEE 21450 NCAP transducer services.

5.9 IEEE 21451-001 TIM transducer signal treatment services

This recommended practice defines a set of transducer signal treatment services, which include signal processing algorithms (signal segmentation and labeling algorithm, signal features extraction and/or detection algorithm, and advanced feature extraction or filtering algorithm) and data structure in order to share and to infer signal and state information of an instrument or control system. These algorithms are based on the acquired signal and also on the transducers attached to the system. This recommended practice also defines an application programming interface (API), which is a set of request/response pairs discussed in detail in Clause 13.

5.10 Signal conditioner and data conversion

The details of the signal conditioning and data conversion functions are beyond the scope of this recommended practice.

5.11 Transducers

Transducers may be classified into three categories:

- ISO/IEC/IEEE 21451-4 transducers
- ISO/IEC/IEEE 21451-7 transducers
- General transducers

For ISO/IEC/IEEE 21451-4 transducers, the interface between transducers and signal conditioning is the

ISO/IEC/IEEE 21451-4 mixed-mode interface (MMI). For the ISO/IEC/IEEE 21451-7 transducers, the interface between transducers and signal conditioning is the ISO/IEC/IEEE 21451-7 radio frequency (RF) interface. For general transducers, the interface between transducers and signal conditioning is beyond the scope of this recommended practice.