

Instituto Tecnológico de Costa Rica

Escuela de Ingeniería Electrónica



Sistema de detección de patrones acústicos de disparos de armas de fuego y motosierras en bosques tropicales basado en un algoritmo de correlación de *bitstream*.

Informe de Proyecto de Graduación para optar por el título de Ingeniero en Electrónica con el grado académico de Licenciatura

Rolando Adrián Moraga Mora

19 de junio de 2016

INSTITUTO TECNOLÓGICO DE COSTA RICA
ESCUELA DE INGENIERÍA ELECTRÓNICA
PROYECTO DE GRADUACIÓN
ACTA DE APROBACIÓN

Defensa de Proyecto de Graduación
Requisito para optar por el título de Ingeniero en Electrónica
Grado Académico de Licenciatura
Instituto Tecnológico de Costa Rica

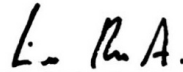
El Tribunal Evaluador aprueba la defensa del proyecto de graduación denominado Sistema de detección de patrones acústicos de disparos de armas de fuego y motosierras en bosques tropicales basado en un algoritmo de correlación *bitstream*, realizado por el señor Rolando Adrián Moraga Mora y, hace constar que cumple con las normas establecidas por la Escuela de Ingeniería Electrónica del Instituto Tecnológico de Costa Rica.

Miembros del Tribunal Evaluador



Ing. Esteban Baradín Méndez

Profesor lector



Ing. Leonardo Rivas Arce

Profesor lector



Ing. Alfonso Chacón Rodríguez

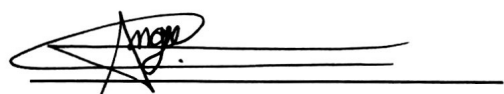
Profesor asesor

Cartago, 16 de junio 2016

Declaro que el presente Proyecto de Graduación ha sido realizado, en su totalidad, por mi persona, utilizando y aplicando literatura referente al tema e introduciendo conocimientos propios.

En los casos en que he utilizado material bibliográfico, he procedido a indicar las fuentes mediante citas.

En consecuencia, asumo la responsabilidad total por el trabajo de graduación realizado y por el contenido del correspondiente informe final.



Rolando Adrian Moraga Mora
Cédula: 1-1505-0664

Resumen

Costa Rica tiene un cuarto del territorio nacional que corresponde a áreas de bosques protegidas, las cuales son constantemente atacadas por actos ilegales como la caza y tala ilegal afectando la flora y fauna. Determinar dónde y cuándo se da un acto ilegal resulta una tarea difícil de lograr para los guardabosques dada la gran extensión de área.

Este proyecto propone la prueba de concepto de un detector acústico que indique cuándo ocurre un evento de caza o tala ilegal. Se propone una solución basada en un algoritmo de correlación cruzada implementado en una FPGA. El problema a resolver se parte en dos: por un lado un algoritmo de detección basado en teoría de señales y por otro lado un módulo encargado de la adquisición de datos. Ambos módulos se describen en el ambiente de programación gráfica LabView y se portan a un myRIO de National Instruments para su verificación.

El primer módulo consiste en un detector binario de patrones acústicos, basado en un comparador de umbral adaptativo del cual se obtiene una curva COR (Característica Operativa del Receptor) o ROC (más conocido en la literatura por sus siglas en inglés), que grafica su sensibilidad como detector binario. Este detector requiere una etapa previa procesamiento basada en un algoritmo de correlación cruzada que provee de la información a analizar. El desarrollo final de este módulo forma parte de otro proyecto paralelo, por lo que aquí se usa una descripción de alto nivel del algoritmo para fines de referencia.

El segundo módulo desarrollado en este proyecto es la etapa de adquisición de datos. Debe emularse aquí un modulador sigma-delta, que luego será sustituido por un circuito real. Así se proveerá de los datos necesarios para el algoritmo de correlación.

Palabras clave: ADC, Curva ROC, FPGA, LabVIEW, Modulador Sigma-Delta, Patrones Acústicos.

Abstract

Costa Rica has a fourth of its national territory that corresponds to protected forest areas, which area constantly foraged by poachers and illegal loggers. Determining where and when does an illegal act happen is a hard task to accomplish for the rangers given the big extension of protected area.

This project proposes a proof of concept of an acoustic detector that indicates when does an event of illegal hunting or logging happen. The solution proposed is based on a cross-correlation algorithm implemented in a FPGA. The design problem is partitioned in two: a detection algorithm based on Signal Theory and a module devoted to data acquisition. Both modules are graphically implemented on LabVIEW and ported to a National Instruments myRIO for their testing.

The first implemented module consists of a binary detector of acoustics patterns, based on an adaptive threshold comparator, from which a ROC (Receiver Operator Characteristic) curve can be obtained. This type of curves plots the sensibility of a binary detector. The input signal to this detector is pre-processed with a particular algorithm to enhance the detectability of the signal. In this case, a cross-correlation processing is used. Since the FPGA porting of this algorithm is being developed in a parallel project, a high level implementation is used as reference model.

The second module developed in this project is the data acquisition stage. Here a sigma-delta modulator must be emulated, and then it will be substituted by a real circuit. This will provide the necessary data for the cross-correlation algorithm.

Keywords: Acoustic Patterns, ADC, FPGA, LabVIEW, ROC Curve, Sigma-Delta Modulator.

*A mi querida familia en especial a mi madre, hermano, primos,
novia y a mi padre para que en descanse en paz finalmente.*

Muchas gracias por estar ahí siempre.

Agradecimientos

A todas las personas que me acompañaron en mi travesía durante la universidad, que estuvieron ahí en todos los buenos y malos momentos, reuniones de estudios y fiestas, pero en especial a Rebeca Castro Artavía, Luis Ricardo Codero Aguilar, Jenifer Brenes Cassasola, David Pacheco Serrano, Stephanie Arias Rodríguez, Jorge Sequeira Rojas y Josué Mora Castro.

Al Ing. Alfonso Chacón Rodríguez asesor de mi proyecto de graduación por toda su ayuda.

Por último un agradecimiento especial a todos los compañeros de junta directiva, fiscalía de la Asociación de Estudiantes de Ingeniería Electrónica (AESETEC) y a todos los otros estudiantes y funcionarios comprometidos con el movimiento estudiantil.

Toda la experiencia obtenida con ustedes es fundamental en mi vida cotidiana.

Rolando Adrián Moraga Mora

Cartago, 19 de junio de 2016

Índice general

Índice de figuras	v
Índice de tablas	ix
Lista de abreviaciones	xi
1 Introducción	1
1.1 La tala y caza ilegal en los bosques tropicales de Costa Rica	1
1.2 Sistema de detección propuesto	2
2 Meta y objetivos	5
2.1 Meta	5
2.2 Objetivos	5
2.2.1 Objetivo general	5
2.2.2 Objetivos específicos	5
3 Metodología	7
3.1 Estudio de teoría	7
3.2 Aprendizaje de las herramientas	7
3.3 Prueba de concepto con modelos en alto nivel	8
3.4 Prueba de concepto con señales reales	8
3.5 Implementación en hardware	8
3.6 Verificación de resultados	8
4 Marco teórico	9
4.1 Generalidades de los sonidos en un bosque	9
4.1.1 Acústica de un disparo	9

4.1.2	Acústica de las motosierras	11
4.2	Modulación sigma-delta	14
4.2.1	Representación en <i>bitstream</i>	15
4.2.2	Frecuencia de muestreo	16
4.2.3	Modulador sigma-delta de segundo orden.	16
4.3	Correlación cruzada	18
4.3.1	Definición	18
4.3.2	Interpretación gráfica	18
4.4	Detector de umbral	20
4.4.1	Detector de umbral adaptativo	20
4.5	Curva ROC	24
4.6	LabVIEW	28
4.6.1	¿Qué es LabVIEW?	28
4.6.2	Instrumentos virtuales	28
4.6.3	Módulo LabVIEW FPGA	31
4.6.4	Modo de ejecución de VIs	32
4.6.5	Método de referencia a VI FPGA	34
5	Sistema de detección de patrones acústicos	35
5.1	Modulador Sigma-Delta	37
5.1.1	Modelo en alto nivel	37
5.1.2	Implementación hardware de la etapa de adquisición	38
5.2	Pre-procesamiento	40
5.3	Creación de plantillas	41
5.3.1	Plantilla de disparo	41
5.3.2	Plantilla de motosierra	42
5.4	Detector de umbral adaptativo	43
5.4.1	Modelo en alto nivel	43
5.4.2	Traslado a la FPGA	44
6	Resultados y análisis	49
6.1	Emulación del modulador Sigma-Delta	49
6.2	Implementación en FPGA del módulo de detección	54

6.3	Verificación contra señales reales	58
6.3.1	Pruebas de detección de disparos	59
6.3.2	Pruebas de detección de motosierras	63
6.3.3	Otras pruebas	66
6.3.4	Curvas ROC	67
7	Conclusiones y recomendaciones	71
	Bibliografía	73
	Índice alfabético	79

Índice de figuras

1.1	Diagrama general del sistema de detección.	3
4.1	Onda de choque “N” de un disparo	10
4.2	Grabación de un disparo de un arma calibre 22 a 30 metros de distancia del micrófono.	11
4.3	Onda de choque de una grabación de un disparo de un arma calibre 22 a 30 metros de distancia del micrófono.	11
4.4	Grabación de una motosierra a 30m de distancia del micrófono.	12
4.5	Grabación de una motosierra a 90m de distancia del micrófono.	12
4.6	Grabación de viento en un bosque.	12
4.7	Grabación de lluvia en un bosque.	13
4.8	Grabación de una motosierra a 30m de distancia del micrófono.	13
4.9	Grabación de una motosierra a 90m de distancia del micrófono.	13
4.10	Grabación de viento en un bosque.	14
4.11	Grabación de lluvia en un bosque.	14
4.12	Modulador Sigma-Delta de primer orden.	15
4.13	Modulador sigma-delta de segundo orden.	17
4.14	SNR contra OSF para moduladores sigma-delta de orden 1 a 5.	17
4.15	Interpretación gráfica de la correlación.	19
4.16	Detector de umbral adaptativo.	21
4.17	Promedio de una señal de ruido.	22
4.18	Promedio de una señal de ruido y un pico.	23
4.19	Detector de umbral adaptativo para picos neuronales basado en la energía de la señal.	23

4.20	Detector de umbral adaptativo para picos neuronales basado en el valor cuadrático medio.	24
4.21	Gráfico de una curva ROC con la <i>diagonal de no discriminación</i>	25
4.22	Matriz de confusión.	26
4.23	Curva ROC para un experimento con 7 valores de umbral diferentes.	27
4.24	Panel frontal de LabVIEW.	28
4.25	Paleta de Controles de LabVIEW.	29
4.26	Diagrama de bloques de LabVIEW.	30
4.27	Paleta de Controles de LabVIEW.	31
4.28	Ejemplo de flujo de datos en LabVIEW.	33
4.29	Ejemplo de flujo de datos en LabVIEW FPGA.	33
4.30	Funciones de interfase FPGA.	34
5.1	Diagrama detallado del sistema de detección.	35
5.2	VI de un conversor sigma delta de segundo orden. Sólo se utilizó en este proyecto el modulador, pues únicamente se requería la señal bitstream.	37
5.3	Esquemático del circuito de aplicación típico para el AD7405.	39
5.4	Circuito impreso del circuito típico para el AD7405.	39
5.5	Fotografía del circuito impreso creado AD7405.	40
5.6	Paleta de VI's de operaciones de señal.	40
5.7	VI de correlación cruzada.	40
5.8	Plantilla de disparo generada a partir de 6 muestras patrones de armas de fuego calibre 22, 32 y 38 grabadas a 30m de distancias a ángulos de 0° y 90° del micrófono.	42
5.9	Plantilla de motosierra generada a partir de 6 muestras patrones de motosierras grabadas a 30 y 90m de distancias a ángulos de 0°, 90° y 180° del micrófono.	43
5.10	Modelo en alto nivel del detector de umbral adaptativo	43
5.11	VI del detector de umbral adaptativo	44
5.12	Código fuente del comparador de nivel y multiplicador por K	45
5.13	Código fuente para calcular media móvil	45
5.14	Código fuente del control del índice del arreglo	46
5.15	Código fuente de entra de datos y rectificador	46

5.16	Código fuente gráfico del VI del detector completo	47
6.1	Resultado de conversión <i>bitstream</i> para una señal de entrada de -1V con un OSF de 8.	50
6.2	Recuperación de una señal de un pulso después de convertida en <i>bitstream</i>	51
6.3	Recuperación de una señal sinusoidal gaussiana después de convertida en <i>bitstream</i>	52
6.4	Recuperación de una señal sinusoidal después de convertida en <i>bitstream</i>	52
6.5	Recuperación de una señal quasi-aleatoria de Halton después de convertida en <i>bitstream</i>	53
6.6	Recuperación de una señal de ruido después de convertida en <i>bitstream</i>	53
6.7	Comparacion de funciones no disponibles en LabVIEW y su análogo en LabVIEW FPGA	55
6.8	Detección de disparo utilizando con la correlación provista por la biblioteca de procesamiento de señales de LabVIEW.	56
6.9	Detección de disparo usando el modelo de correlación bitstream implementado en LabVIEW por [5]	56
6.10	VI de conversión de un archivo de audio a texto y viceversa.	59
6.11	Detección de disparo a 30m de una carabina calibre 22.	59
6.12	Detección de disparo a 30m de una pistola 9mm.	60
6.13	Detección de disparo a 30m de una escopeta calibre 12mm.	60
6.14	Detección de disparo a 90m de una carabina calibre 22.	60
6.15	Detección de disparo a 90m de un revolver calibre 38.	61
6.16	Detección de disparo a 90m de una pistola 9mm.	61
6.17	Detección de disparo a 250m de una revolver calibre 38.	61
6.18	Detección de disparo a 250m de una escopeta calibre 12.	62
6.19	Media móvil y correlación para un disparo a 30m de una carabina calibre 22.	62
6.20	Detección de 4 disparos seguidos.	63
6.21	Detección de una motosierra a 30m.	63
6.22	Detección de una motosierra a 90m.	64
6.23	Detección de una motosierra a 90m.	65
6.24	Detección de otra motosierra a 90m.	65
6.25	Detección de lluvia en un bosque tropical.	66

6.26 Detección de viento en un bosque tropical.	66
6.27 Detección de aves cantando en un bosque tropical.	67
6.28 Detección un disparo y motosierra.	67
6.29 Curva ROC para la detección de disparos.	68
6.30 Curva ROC para la detección de motosierras.	68

Índice de tablas

4.1	Ciclo de trabajo de una señal en bitstream para un valor de tensión dado. . .	15
4.2	Posibles salidas de un detector binario	26
5.1	Descripción de los bloques del sistema de detección de patrones acústicos. . .	36
5.2	Señales del sistema de detección de patrones acústicos.	36
5.3	Entradas y salidas para el VI del modulador sigma-delta de segundo orden. .	38
5.4	E/S del VI del detector de umbral adaptativo	44
6.1	Prueba del módulo sigma-delta con valores constantes de -1V a 1V en pasos de 0.5V con un OSF de 8.	51
6.2	Resultados obtenidos de SNR para señales variantes en el tiempo con un OSF de 64.	54
6.3	Recursos utilizados en el SoC del NI myRIO por el VI de FPGA del detector	57
6.4	Comparación de recursos utilizados por la función de <i>Quotient and Remainder</i> y <i>Logical Shift</i>	58
6.5	ROC para la detección de disparos.	69
6.6	ROC para la detección de motosierras.	69

Lista de abreviaciones

ACC: Del inglés Accuracy.

ADC: Del inglés Analog to Digital Converter.

CFAR: Del inglés Constant False-Alarm Rate.

DCILab Laboratorio de Diseño de Circuitos Integrado.

FPGA: Del inglés Field Programmable Gate Array.

FN: Falsos Negativos.

FP: Falsos Positivos.

ITCR: Instituto Tecnológico de Costa Rica.

MINAE: Ministerio Nacional de Ambiente y Energía.

OSF: Del inglés Oversampling Factor.

PCM: Del inglés Pulse Code Modulation.

RFP: Razón de Falsos Positivos.

RIO: Del inglés Reconfigurable Input Out.

ROC: Del inglés Receiver Operator Characteristic.

RVP: Razón de Verdaderos Positivos.

SCTL: Del inglés Single-Cycle Timed Loop.

SiRPA: Sistema de Reconocimiento de Patrones Acústicos.

SNR: Del inglés Signal-to-Noise Ratio.

SoC: Del inglés System on Chip.

VI: Del inglés Virtual Instrument.

VIE: Vicerrectoría de Investigación y Extensión.

VN: Verdaderos Negativos.

VP: Verdaderos Positivos.

Capítulo 1

Introducción

1.1 La tala y caza ilegal en los bosques tropicales de Costa Rica

Costa Rica es uno de los países territorialmente más pequeños del mundo, pero con unas de las biodiversidades más grandes a nivel mundial, aproximadamente un 25.1% de el territorio nacional se encuentra bajo alguna categoría de bosques protegidos [25] los cuales se encuentran amenazados constantemente por personas que talan y/o cazan de forma ilegal.

El ecoturismo es uno de las principales actividades económicas de nuestro país. De hecho Costa Rica es uno de los países pioneros en el ecoturismo, considerándose extremadamente popular por extranjeros con el objetivo de visitar las áreas protegidas o parques nacionales. Este es uno de los motivos por el cual es fundamental velar por la protección de los bosques protegidos [30].

En el Índice de Competitividad en Viajes y Turismo, Costa Rica tomó el primer lugar en América Latina y el puesto 42 a nivel mundial. En la región del Caribe también cuenta con los mayores índices per cápita de visitantes extranjeros, dejando alrededor de \$2200 millones anuales en ingresos al país y contribuyendo aproximadamente en un 8.1% en el producto interno bruto, además del aporte en empleos en el sector turístico [2].

Existe un mercado ilegal de madera a nivel mundial, y Costa Rica juega un papel muy importante dentro de este mercado, dada la gran variedad de árboles exóticos y especies

en peligro de extinción que tienen los bosques nacionales, por lo mismo se invierte una gran cantidad de dinero y esfuerzo logístico por parte de las autoridades nacionales para combatir esta situación ya que además atenta a los hábitats naturales que existen en los bosques [34] [4].

Contemplando lo anterior, Costa Rica en Diciembre del 2012 aprobó la Ley de Conservación de Vida Silvestre, en la que prohíbe cualquier tipo de caza (deportiva y comercial), tráfico y posesión como mascotas de especies silvestres [31]. A pesar de que se aprobó esta ley siguen ocurriendo eventos graves como la extracción de árboles protegidos o el tráfico de animales [28].

El Minae pretende contrarrestar esta situación con más plazas y nuevas políticas [35]. Estas medidas únicamente elevan sustancialmente los gastos y ni siquiera medidas paliativas como la reforestación están teniendo mayor efecto, lo cual sugiere que el uso de tecnología sería de gran ayuda para poder combatir eficientemente ésta problemática.

La ineficiencia en la detección de caza y tala ilegal debido a la alta densidad de área de zonas protegidas causa que se sigan cometiendo estos delitos y además aumenten. Por éste motivo se han desarrollado diferentes sistemas con el propósito de detectar este tipo eventos basados en el reconocimiento de patrones acústicos. En el 2011 se desarrolló en la VIE del ITCR, un proyecto adscrito a la Escuela de Electrónica conocido como SiRPA, que propuso un clasificador para el reconocimiento de disparos y motosierras en zonas boscosas, y varios subsistemas analógicos y digitales de detección. [6, 42, 44, 46, 51].

Además, si una señal de audio corresponde a un disparo o no es el único problema asociado, detalles como localizar la ubicación de donde se origina un disparo también lo es, por lo que se han hecho proyectos para detectar el ángulo de una señal de audio respecto a varios micrófonos [40] y minimizando la energía al máximo de los algoritmos de localización, pero con una velocidad adecuado para detectar fenómenos impulsivos [1, 41, 48].

1.2 Sistema de detección propuesto

Esta solución está basada en investigación previa en donde se validó el uso de diferentes algoritmos para la detección de patrones acústicos, entre estos el de correlación cruzada

[47, 49]; sin embargo, en este proyecto a diferencia de estos trabajos previos se propone usar el formato de *tren de pulsos* o *bitstream* porque este tipo de representación optimiza el uso de recursos hardware y por ende el consumo de potencia [43].

Los requerimientos de este sistema están relacionados sobre todo con el consumo de potencia y área para una potencial integración en un IC de muy baja potencia. No obstante, antes de realizar ese paso, deben integrarse una prueba de concepto en FPGA de todo el sistema que permita verificar contra una muestra relevante de sonidos su funcionalidad.

Considerando los requerimientos planteados, se propuso un sistema con tres bloques principales como se muestra en la Figura 1.1, donde:

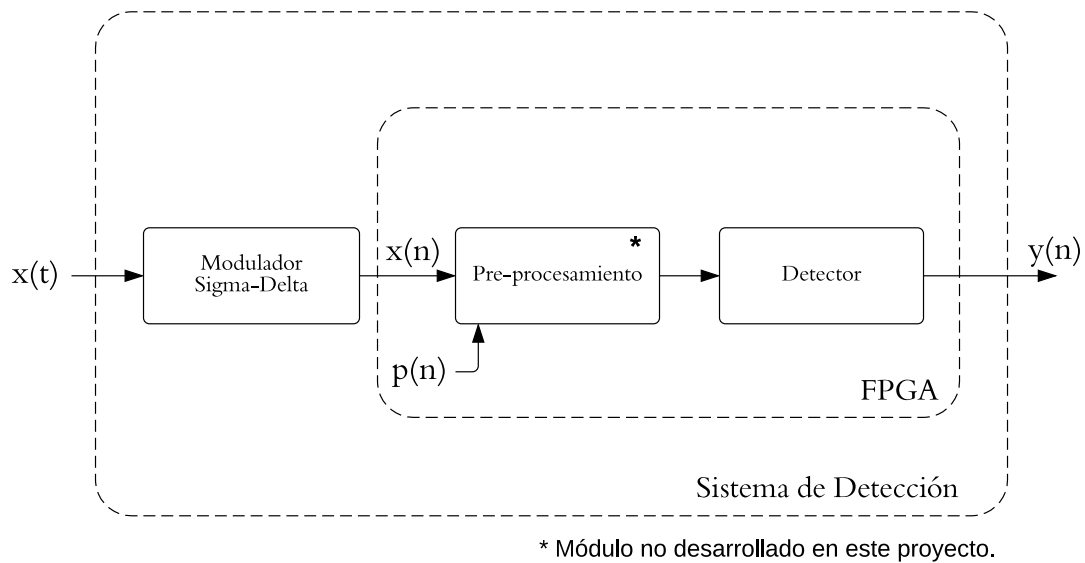


Figura 1.1: Diagrama general del sistema de detección.

- $x(t)$: es la señal de entrada acústica en el tiempo.
- $p(n)$: es el patrón acústico o plantilla (en formato *bitstream*) contra la que se realiza la comparación de la señal $x(n)$.
- El modular sigma-delta genera el patrón de *bitstream* de la señal a correlacionar.
- El bloque de “pre-procesamiento” calcula la correlación cruzada (bloque no implementado en esta tesis, pero al que se le definen los requerimientos de interfaz para su posterior integración desde otro proyecto paralelo a éste; para esta tesis se usó un módulo de alto nivel de LabVIEW para suplir este bloque [5]).

- El detector de umbral adaptativo se alimenta del algoritmo de correlación cruzada y tiene una salida binaria que indica si se da una detección o no. Éste bloque se explica más a fondo en el capítulo 2.

Capítulo 2

Meta y objetivos

2.1 Meta

Este proyecto pretende realizar una prueba de concepto de un sistema digital de detección de patrones acústicos basado en un algoritmo de correlación cruzada para su potencial fabricación en un circuito integrado.

2.2 Objetivos

2.2.1 Objetivo general

Implementar en FPGA una prueba de concepto de detección de patrones acústicos de disparos y motosierras en ambientes boscosos tropicales, basado en un algoritmo de correlación cruzada *bitstream*.

2.2.2 Objetivos específicos

1. Emular la etapa de adquisición de datos modulados en sigma-delta, para alimentar la unidad de detección.
2. Implementar en FPGA el módulo de detección basado en teoría de detección de señales.

3. Verificar el módulo de detección contra señales reales de disparos y motosierras grabadas en un bosque lluvioso tropical.

Capítulo 3

Metodología

3.1 Estudio de teoría

Se partió de los requerimientos solicitados por el DCILab mencionados en la sección 1.2 y con ellos se identificaron áreas de conocimiento importantes de estudiar (moduladores sigma-delta, correlación cruzada y detectores de umbral adaptativo) para comprender el funcionamiento de los tres bloques mostrados en la Figura 1.1 y desarrollar la solución, por lo que se estudiaron varias publicaciones y artículos relacionados con el tema, además de varios de los artículos y tesis que se han producido en el DCILab sobre este tema.

3.2 Aprendizaje de las herramientas

Para llevar a cabo el desarrollo de este proyecto se procedió a utilizar LabVIEW con el objetivo de realizar un modelo en alto nivel de todo el sistema por lo que fue necesario realizar tutoriales sobre la herramienta disponibles vía web para estudiantes activos de la carrera de ingeniería en electrónica del ITCR [18]. Además se utilizó el módulo de LabVIEW FPGA .Por esta razón se hicieron tutoriales para este módulo disponibles en el sitio web oficial de National Instruments [16] .

3.3 Prueba de concepto con modelos en alto nivel

Se implementó en LabVIEW los módulos del proyecto con el fin de validar el funcionamiento de cada uno de los módulos en alto nivel antes de realizar una implementación en hardware. En esta etapa se utilizó señales sinusoidales para realizar dicha validación.

3.4 Prueba de concepto con señales reales

Luego se procedió a realizar pruebas con las plantillas creadas y las señales de audio reales de la base de datos que tiene el DCILab de grabaciones reales de motosierras, disparos y sonidos ambientales de bosque.

En esta etapa se realizó una gran cantidad de pruebas con el fin de determinar la validez de la plantilla creada y los parámetros de ganancia (K), ventana de la media móvil (N) (ver Tabla 5.4) para el correcto funcionamiento del sistema de detección y las plantillas.

3.5 Implementación en hardware

Para la implementación en físico se utilizó el módulo de LabVIEW FPGA y un myRIO, en esta parte se hicieron las debidas modificaciones para trasladar el código de LabVIEW para que sea compatible con este módulo.

Se diseñó y se construyó un PCB con el circuito de aplicación típico para un modulador sigma-delta de segundo orden de 16bits analógico, modelo AD7405.

3.6 Verificación de resultados

Luego de depurar por medio de simulaciones la versión de LabVIEW FPGA del detector umbral adaptativo, se realizaron las pruebas estimulando con las señales reales ya mencionadas a la FPGA por medio del método de referencia de LabVIEW que comunica el computador anfitrión con la FPGA. De esta manera se puede estimular con las grabaciones almacenadas en la computadora y recopilar fácilmente los resultados obtenidos en la FPGA.

Capítulo 4

Marco teórico

En este capítulo se muestran los fundamentos teóricos, físicos y matemáticos utilizados en este proyecto, así como una explicación del funcionamiento básico de la herramienta de software principal LabVIEW y su módulo de FPGA myRIO de la empresa National Instruments, utilizada a lo largo del desarrollo del este proyecto.

4.1 Generalidades de los sonidos en un bosque

En esta sección se explica sobre las señales acústicas típicas de los bosques tropicales, motosierras, disparos y los detalles relevantes para el algoritmo de detección utilizado en este proyecto.

4.1.1 Acústica de un disparo

Las grabaciones de los disparos de armas de fuego proveen información acerca de la localización del arma con respecto al micrófono, la velocidad y trayectoria del proyectil, la dificultad yace en interpretar los sonidos debido al eco (solapamiento de las reflexiones acústicas) del disparo.

Un arma convencional utiliza pólvora como explosivo para impulsar la bala afuera del cañón de la pistola. Los gases calientes en expansión causados por un disparo generan una perturbación acústica que dura de 3 a 5 milisegundos y se propaga en el aire a la velocidad del

sonido dependiendo del tamaño de la carga, la masa de bala y otros factores [26].

La *onda de choque* acústica de la bala tiene un máximo positivo de sobrepresión que aumenta muy rápido seguido de un mínimo de subpresión, mientras la onda de choque se propaga el comportamiento no lineal del aire causa una perturbación en la presión que forman una onda de choque con forma “N” que tiene un inicio rápido y una rampa a un mínimo de presión, generalmente duran al menos $200 \mu\text{s}$ dependiendo del tamaño de la bala a como se muestra en la Figura 4.1. Es importante aclarar que este patrón puede variar en amplitud y duración si se utiliza armas de diferentes calibres.

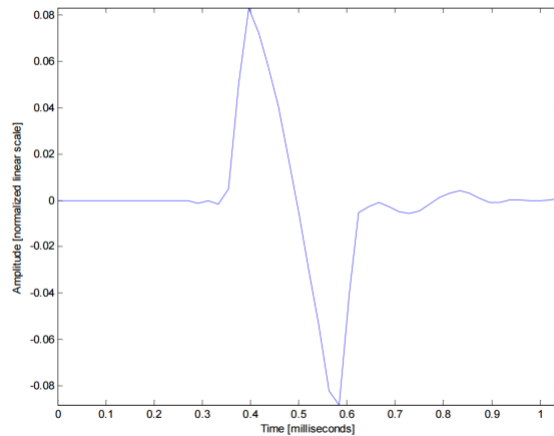


Figura 4.1: Onda de choque “N” de un disparo. Tomado de [26].

En la Figura 4.2 se muestra la forma de onda completa de un disparo real de un arma calibre 22 grabado a una distancia de 30m, en esta figura se puede apreciar el impulso generado en un disparo. La onda de choque N se encuentra aproximadamente en el segundo 0.21, tal a como se muestra en la Figura 4.3 .

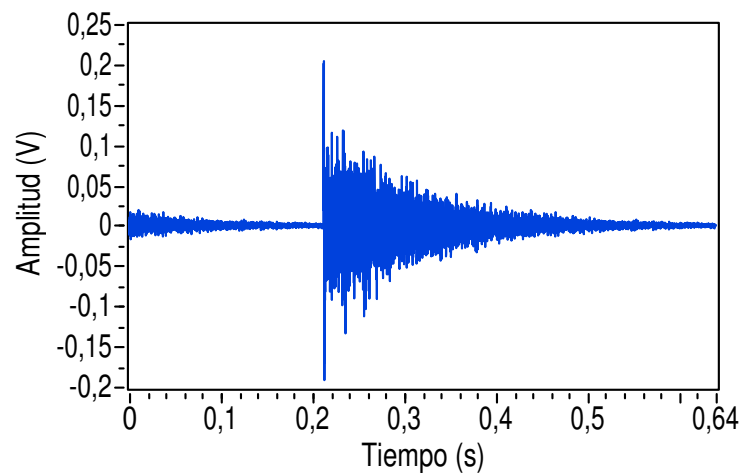


Figura 4.2: Grabación de un disparo de un arma calibre 22 a 30 metros de distancia del micrófono.

En la Figura 4.3 se muestra la onda de choque “N” después de ampliar el impulso del disparo mostrado en la Figura 4.2.

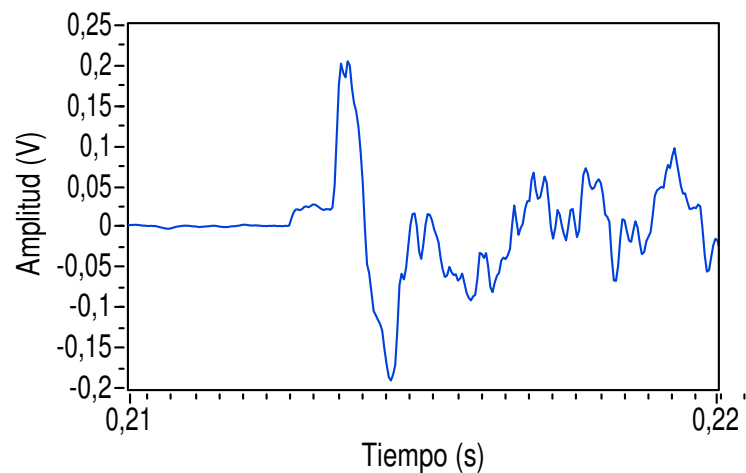


Figura 4.3: Onda de choque de una grabación de un disparo de un arma calibre 22 a 30 metros de distancia del micrófono.

4.1.2 Acústica de las motosierras

La acústica que gira en torno a las motosierras es muy diferente a la expuesta en la sección 4.1.1 para los disparos, ya que no hay una firma acústica que se presente únicamente en un instante determinado.

Las Figuras 4.4 y 4.5 representan grabaciones reales de motosierras en un bosque tropical, a primera vista se pueden confundir con señales de ruido o con sonidos típicos de un bosque

como se muestra en las Figuras 4.6 y 4.7 que representan una grabación de viento y lluvia respectivamente.

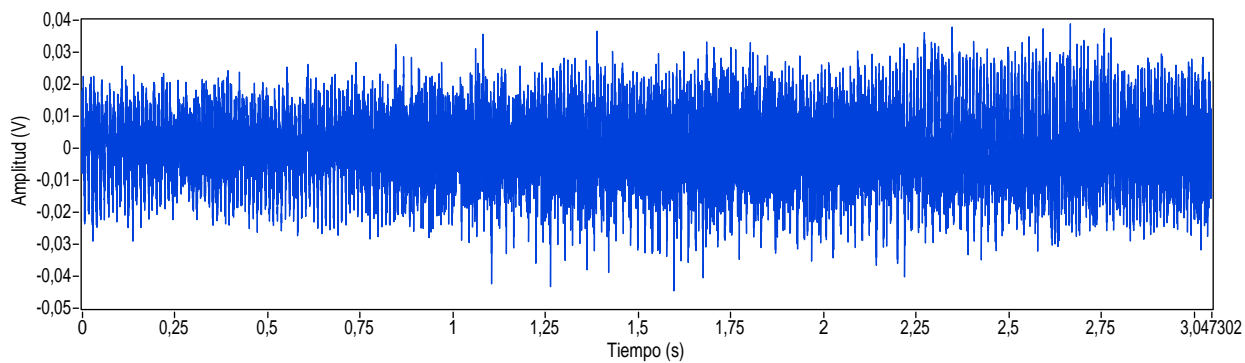


Figura 4.4: Grabación de una motosierra a 30m de distancia del micrófono.

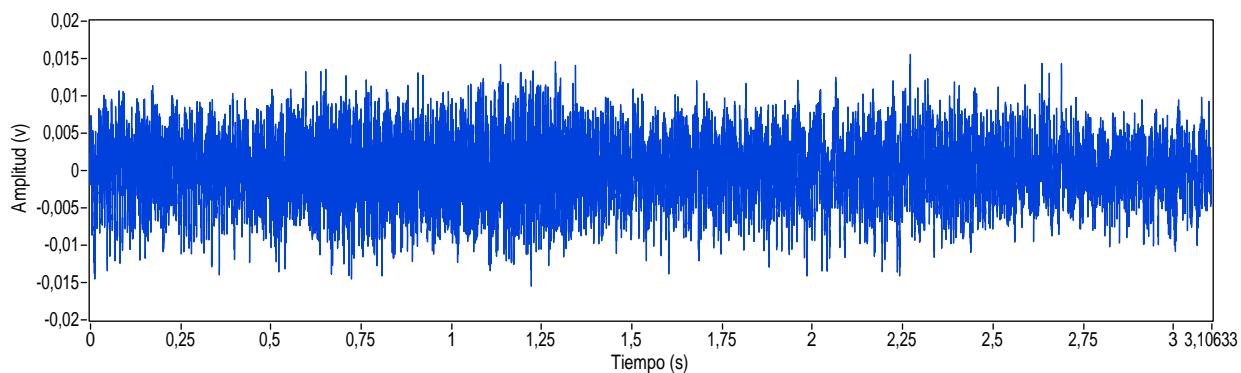


Figura 4.5: Grabación de una motosierra a 90m de distancia del micrófono.

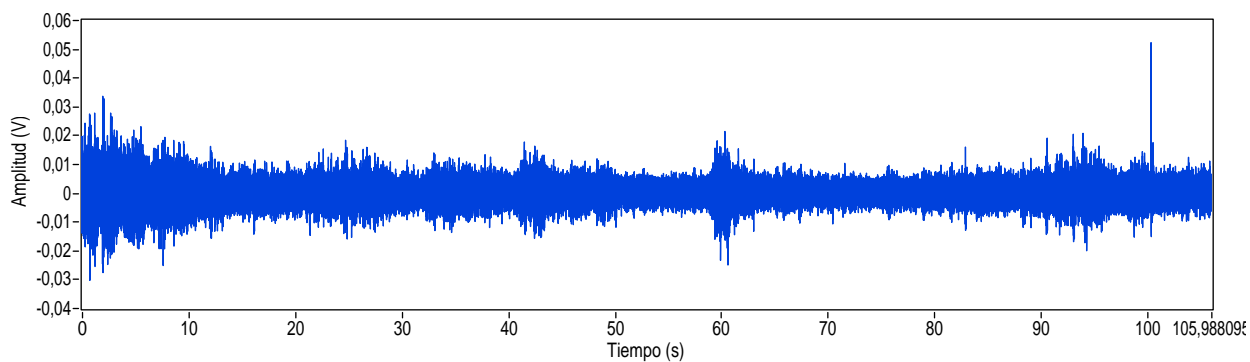


Figura 4.6: Grabación de viento en un bosque.

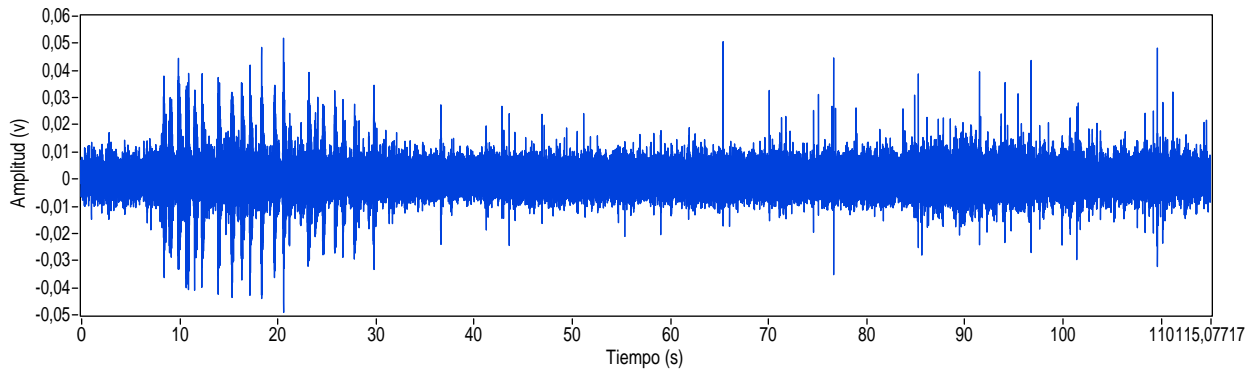


Figura 4.7: Grabación de lluvia en un bosque.

La diferencia fundamental de una señal acústica generada por una motosierra contra el ruido común del entorno de un bosque es que, el motor genera una señal mucho de picos de mayor frecuencia que se traduce en sonidos repetitivos [37].

Para ejemplificar lo anterior se presentan las Figuras 4.8 a 4.11 que corresponden a las figuras anteriores (4.4 a 4.7) ampliadas en la escala del tiempo con igual proporción.

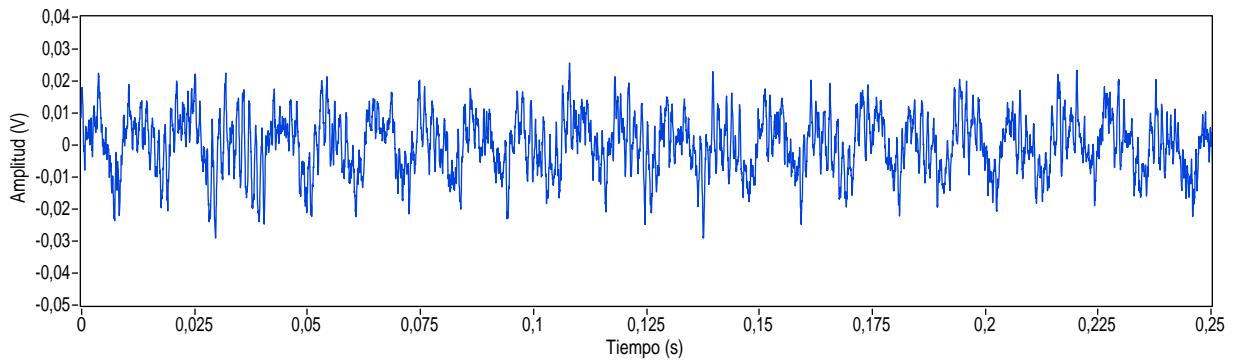


Figura 4.8: Grabación de una motosierra a 30m de distancia del micrófono.

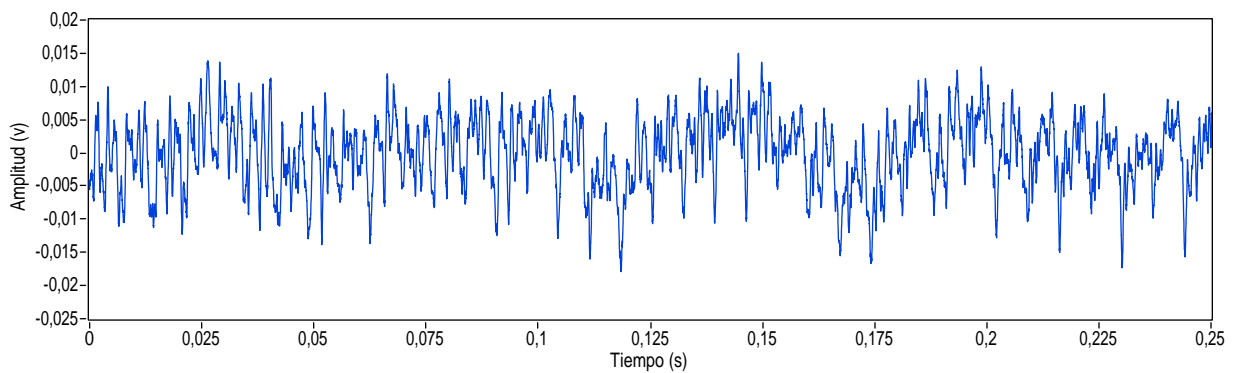


Figura 4.9: Grabación de una motosierra a 90m de distancia del micrófono.

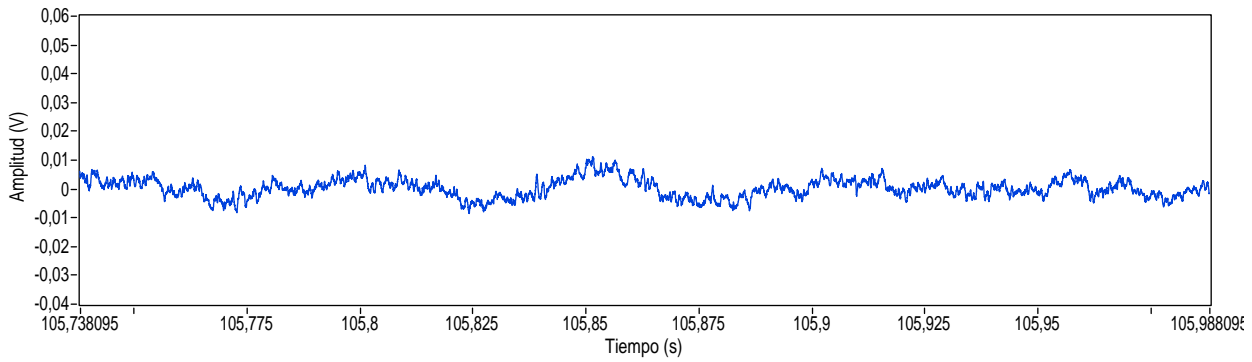


Figura 4.10: Grabación de viento en un bosque.

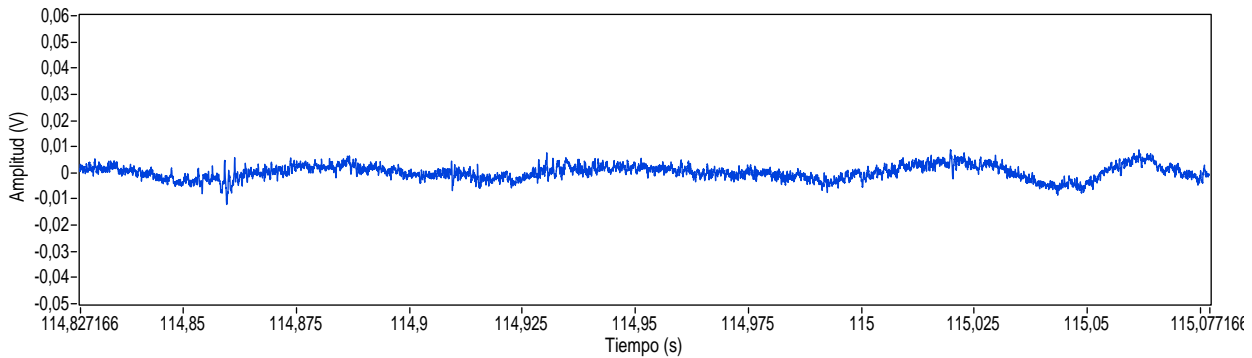


Figura 4.11: Grabación de lluvia en un bosque.

Se puede apreciar claramente que las Figuras 4.8 y 4.9 a diferencia de las Figuras 4.10 y 4.11 muestran picos con una mayor intensidad y más frecuencia.

4.2 Modulación sigma-delta

Los sistemas de procesamiento digital de señales por lo general se ven limitados por la precisión de la señal que es obtenida en digital y los moduladores *sigma-delta* se utilizan en un tipo particular de convertidores de analógico a digital para propósitos de alto desempeño y resolución.

Aunque la modulación sigma-delta fue introducida en el año 1962, no se dio gran uso hasta en los últimos años debido al incremento en el uso de técnicas digitales en el área de comunicación y aplicaciones de audio se aumentó el interés en estos ADCs de alto desempeño y precisión [33].

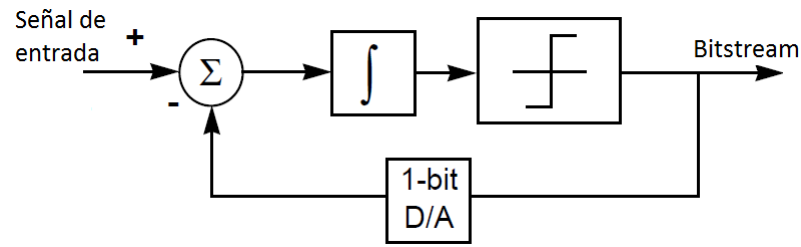


Figura 4.12: Modulador Sigma-Delta de primer orden. Figura tomada de [33]

En la Figura 4.12 se muestra un diagrama de bloques general de un modulador sigma-delta de primer orden, este tipo de moduladores produce una salida conocida como *bitstream*.

4.2.1 Representación en *bitstream*

Este tipo de representación en *bitstream* consiste en una señal de bits en serie, donde su valor medio total representa la media de la señal de entrada. Este concepto se asemeja al de modulación de ancho de pulso (PWM), pero en *bitstream*, el ancho de pulso de un 1-bit no cambia, pero sí cambia el ciclo de trabajo general del total de bits en serie [8].

Para ejemplificar esto observe la Tabla 4.1 y considere caso de una señal de entrada analógica con un rango de valores de tensión de $-V_{in}$ a $+V_{in}$.

Tabla 4.1: Ciclo de trabajo de una señal en bitstream para un valor de tensión dado.

Tensión de entrada	Ciclo de trabajo de la señal en <i>bitstream</i>	<i>Bitstream</i> de 4 bits
$+V_{in}$	100%	1111
$+V_{in}/2$	75%	0111
0	50%	0101
$-V_{in}/2$	25%	0100
$-V_{in}$	0%	0000

Por este motivo se puede usar simplemente un *filtro pasa bajas* analógico o digital para recobrar la señal de entrada, pues es solo cuestión de recuperar el valor medio de la señal en PCM para conocer su valor binario equivalente en paralelo. En el caso de este proyecto, no obstante, se pretende realizar el procesamiento directamente sobre la señal codificada en *bitstream*.

4.2.2 Frecuencia de muestreo

El *teorema de muestro de Nyquist* establece que una señal se debe de muestrear con una frecuencia de al menos el doble de la frecuencia máxima de la señal de entrada, un ADC que convierte con exactamente el doble es conocido como *convertidor de Nyquist* o *ADC de Nyquist*. Sin embargo, es posible utilizar frecuencias mayores a ésta para muestrear, este fenómeno se conoce como *sobremuestrear*.

De acuerdo a Nyquist, si una señal se muestrea a 48kHz, señales de hasta 24kHz pueden pasar sin *aliasing*. Si ésta misma señal de 24kHz se muestrea a 96kHz se dice que está sobremuestreada por un factor de dos. Dicho de otra forma sea $f_{Nyquist}$ la frecuencia de muestreo de Nyquist y una señal se muestrea a $Nf_{Nyquist}$; se dice que está N veces sobremuestreada.

Los moduladores sigma-delta requieren muestrear a frecuencias mayores a la estipulada por Nyquist con el fin de obtener una mayor cantidad de bits en el bitstream de un *mismo* valor de entrada, con esto se mejora la aproximación del valor medio y se reduce el *ruido cuantización* [3].

De acuerdo a la teoría de ADCs, durante el proceso de convertir o cuantizar una señal analógica que por definición tiene resolución infinita a un rango finito de valores en un sistema numérico finito se introduce un error conocido como “*error de cuantización*”. Este error suele ser insignificante en señales grandes en amplitud pero conforme la señal de entrada se reduce en amplitud este error se vuelve mas significativo, que es el caso en las señales de audio [33].

4.2.3 Modulador sigma-delta de segundo orden.

Otra manera en vez de incrementar el factor de sobremuestreo para reducir el ruido es utilizar un modulador sigma-delta de mayor orden. El *bitstream* generado por un modulador de mayor orden reduce el *error de conversión* intrínscico del *bitstream* y además el ruido de cuantización se traslada a frecuencias mayores a las de interés. De esta manera se obtiene una mejor relación de señal-a-ruido (SNR) sin aumentar la cantidad de bits [3, 29, 33].

En la Figura 4.13 se muestra un ejemplo del diagrama de bloques para un modulador sigma-delta de segundo orden, el cual agrega una etapa de retroalimentación adicional de un inte-

gradador. Según Reis, con un modulador de segundo orden se obtiene una mejora de 20dB en SNR por cada duplicación de factor de sobremuestreo.

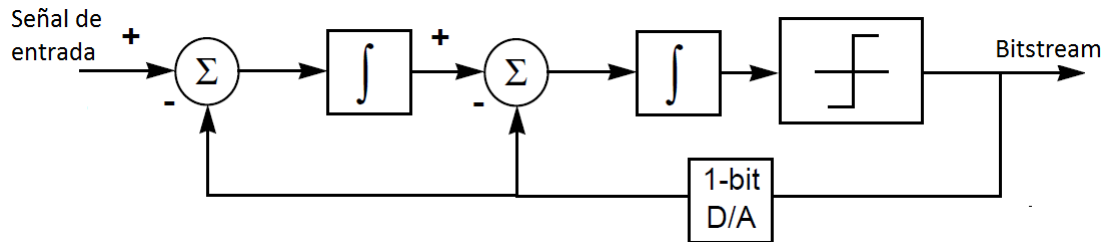


Figura 4.13: Modulador sigma-delta de segundo orden. Figura tomada de [33].

En la Figura 4.14 se muestra una gráfica teórica del SNR contra el factor de sobremuestreo (OSF) para moduladores sigma delta de orden 1 hasta 5.

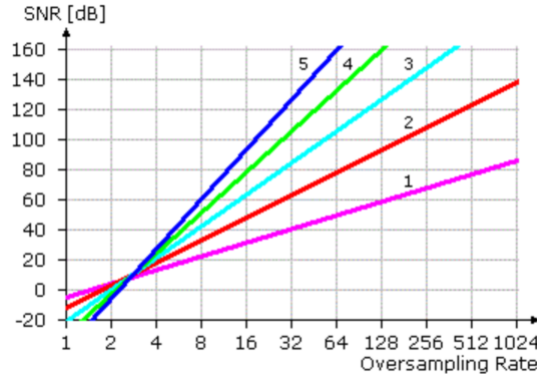


Figura 4.14: SNR contra OSF para moduladores sigma-delta de orden 1 a 5. Figura tomada de [3].

Los moduladores sigma-delta de tercer orden no se pueden obtener con “solo” agregar etapas adicionales de integradores como en el caso de el de segundo orden. El sistema se vuelve inestable por el cambio de fase ocasionado por los integradores, pero es factible obtener moduladores de hasta quinto orden por otros métodos. Estos son utilizados en la industria del audio [3].

4.3 Correlación cruzada

4.3.1 Definición

La operación de *correlación* cruzada suele conocerse como correlación y es frecuentemente utilizada para cuantificar el grado de similitud entre dos señales o procesos. Por este motivo es común en el uso de reconocimiento de patrones [38, 39].

Formalmente la correlación de $x(t)$ con $y(t)$ está definida por

$$R_{xy}(\tau) = (x \star y)(\tau) = \int_{-\infty}^{\infty} \bar{x}(\tau)y(t + \tau)d\tau \quad (4.1)$$

Donde \star es el operador de correlación y $\bar{x}(\tau)$ el conjugado completo de $x(\tau)$. De manera similar para funciones discretas, la correlación de $x(n)$ y $y(n)$ se define como

$$R_{xy}(k) = (x \star y)(k) = \sum_{n=-\infty}^{\infty} x(n)y(n - k) \quad (4.2)$$

Se puede definir también la auto-correlación cruzada o simplemente auto-correlación, que es la correlación de una señal consigo misma o con una señal idéntica a ella, como se vio en (4.1) y (4.2), la auto-correlación para el caso específico del tiempo discreto se puede definir como

$$R_{xx}(k) = (x \star x)(k) = \sum_{n=-\infty}^{\infty} x(n)x(n - k) \quad (4.3)$$

4.3.2 Interpretación gráfica

En la Figura 4.15 se muestra la interpretación gráfica de la correlación y auto-correlación. En la izquierda se muestra $f \star g$ donde $f(t) \neq g(t)$ y a la derecha se muestra la auto-correlación de $f(t)$.

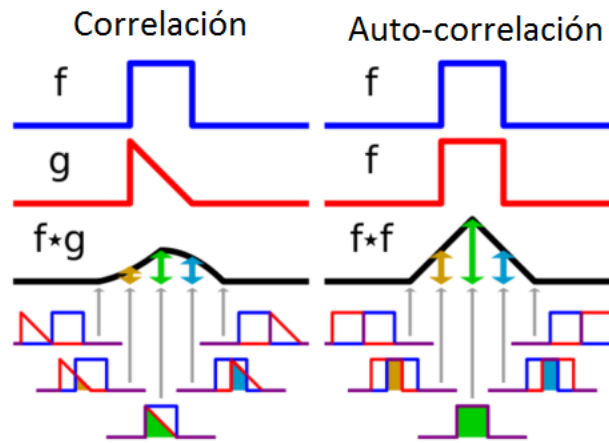


Figura 4.15: Interpretación gráfica de la correlación. Figura tomada de [11].

Se puede observar que de la correlación de dos señales iguales o muy parecidas entre sí se obtiene como resultado un máximo mayor que el de la correlación de dos señales diferentes. Este principio es el motivo del porqué se utiliza la correlación para la detección de patrones, gracias al máximo que genera cuando detecta la similitud de dos señales.

Otros usos

Es importante mencionar que la correlación cruzada además de utilizarse para el reconocimiento de patrones también se puede utilizar para medir el retardo de dos señales.

Sea

$$x(t) = s(t) \quad (4.4)$$

una señal en el dominio del tiempo y

$$y(t) = s(t + D) \quad (4.5)$$

la misma señal $x(t)$ pero con un retardo de D , es posible determinar este retardo encontrando el tiempo en el que sucede el máximo de $R_{xy}(\tau)$.

$$D = \operatorname{argmax}(R_{xy}(\tau)) \quad (4.6)$$

Un uso común de la correlación para determinar retardo se da cuando se tiene un arreglo de sensores y se desea determinar el retardo entre ellos o también medir el retardo de propagación de las señales de audio en un arreglo de micrófonos. Esta operación es sin embargo

muy pesada, y por ello se buscan alternativas como la de [50] para simplificarla. Una opción entonces interesante es la que se explora en este proyecto, pues la correlación *bitstream* es mucho menos pesada algorítmicamente, como demuestran en [43].

4.4 Detector de umbral

Los detectores de umbral consisten en un comparador de nivel convencional, en donde al darse una señal con una intensidad mayor al umbral especificado éste se activa. Sin embargo, este tipo de comparadores solamente es útil en ambientes controlados en donde el nivel de ruido o de señales indeseadas es constante y el nivel de la señal a detectar no varía y supera al del ruido.

Cuando se tiene un ambiente en donde el ruido es variable e inclusive puede sobrepasar el nivel de la señal del evento a detectar, este tipo de detectores de umbral constante no es útil puesto que puede activar el sistema y dar alarmas falsas.

Para el caso específico del que trata este proyecto en un bosque se puede encontrar una gran variedad de sonidos: personas hablando, sonidos de pájaros cantando, agua de lluvia o ríos fluyendo, viento, entre otros; además la intensidad de ellos puede variar, por lo que basarse únicamente por el pico generado al darse un disparo o el sonido de una motosierra como se ve en la sección 4.1.1 y 4.1.2 respectivamente no es fiable, por éste motivo se emplea el uso de detectores de *umbral adaptativo*.

4.4.1 Detector de umbral adaptativo

Estos detectores se basan en el principio de los algoritmos de tasa constante de falsas alarmas o CFAR por sus siglas en inglés, que utilizan un algoritmo adaptativo por medio de un estudio estadístico del entorno en el cual desean detectar un fenómeno impulsivo en presencia de ruido y/o interferencias [24].

Se utilizan comúnmente en la industria biomédica como detectores de picos neuronales, en la medicina en equipos que realizan electroencefalogramas o electrocardiogramas, radares y en cualquiera otra aplicación relacionada con la detección de eventos impulsivos [7, 10, 24, 32, 45].

En la Figura 4.16 se muestra un diagrama de la arquitectura típica de estos detectores, el cual compara el nivel de la señal $x(t)$ preprocesada con el umbral adaptativo. El umbral es calculado generalmente con una media móvil amplificada por una constante ajustable que determina la sensibilidad del detector completo.

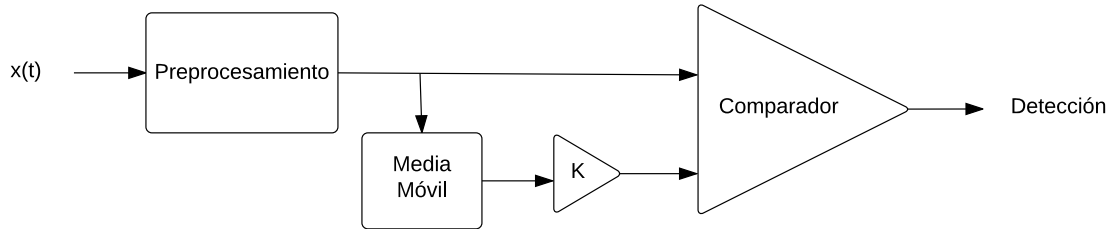


Figura 4.16: Detector de umbral adaptativo.

Pre-procesamiento

Existen muchos tipos de técnicas de pre-procesamiento utilizado en este tipo de detectores. Estos tienen típicamente la función general de resaltar los picos existentes en la señal a analizar ya que típicamente los sonidos ambientales en el bosque tienden a ser periódicos, suaves y no tan impulsivos respecto a los sonidos de motosierras y disparos.

Umbral adaptativo

El umbral adaptativo se calcula con la media móvil amplificada K veces de los últimos n segundos o muestras de la señal pre-procesada. El fin de esta técnica es obtener el nivel de ruido de la señal de entrada mediante el media de la señal y amplificarla para que no supere el máximo de la señal y no active al comparador, como es el caso que se observa en la Figura 4.17.

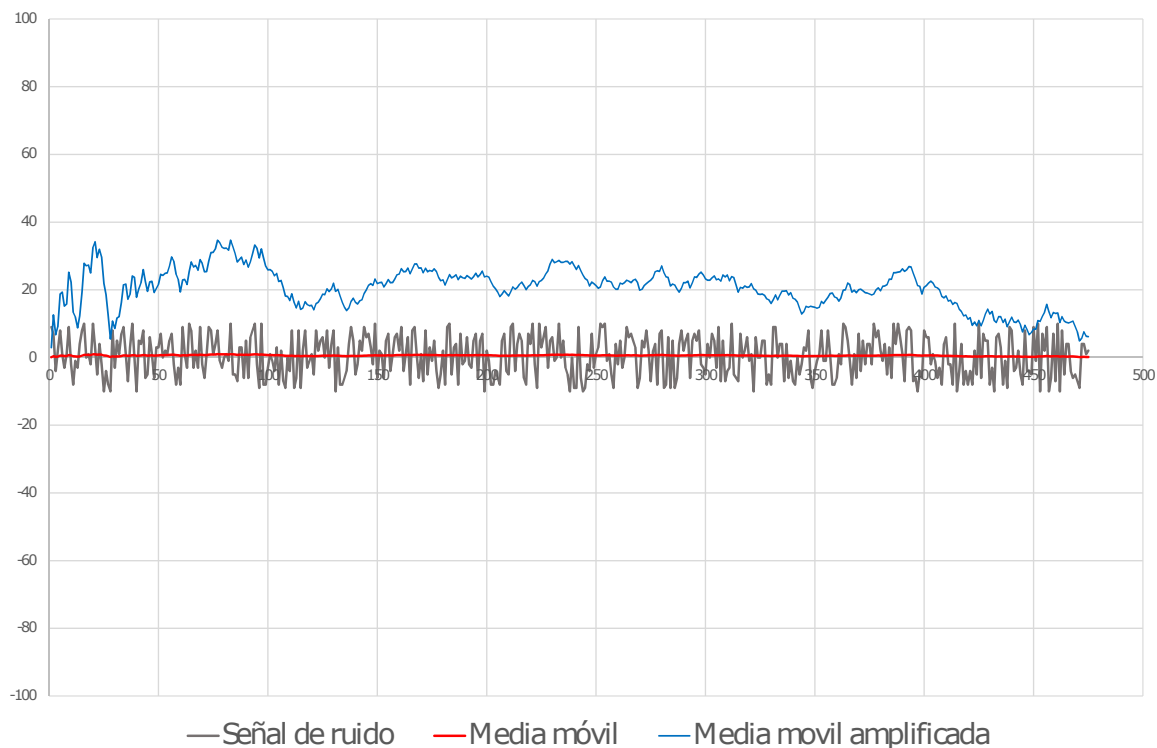


Figura 4.17: Promedio de una señal de ruido.

Dada la característica propia de la *media estadística*, ésta no va a registrar rápidamente el valor o efecto de un pico existente en una señal de entrada. Por ello el resultado es que el valor de la media móvil inmediato es menor al valor del pico activando de esta manera el comparador, tal como se muestra en la Figura 4.18 que representa la misma señal de ruido de la Figura 4.17 pero con la existencia de un pico.

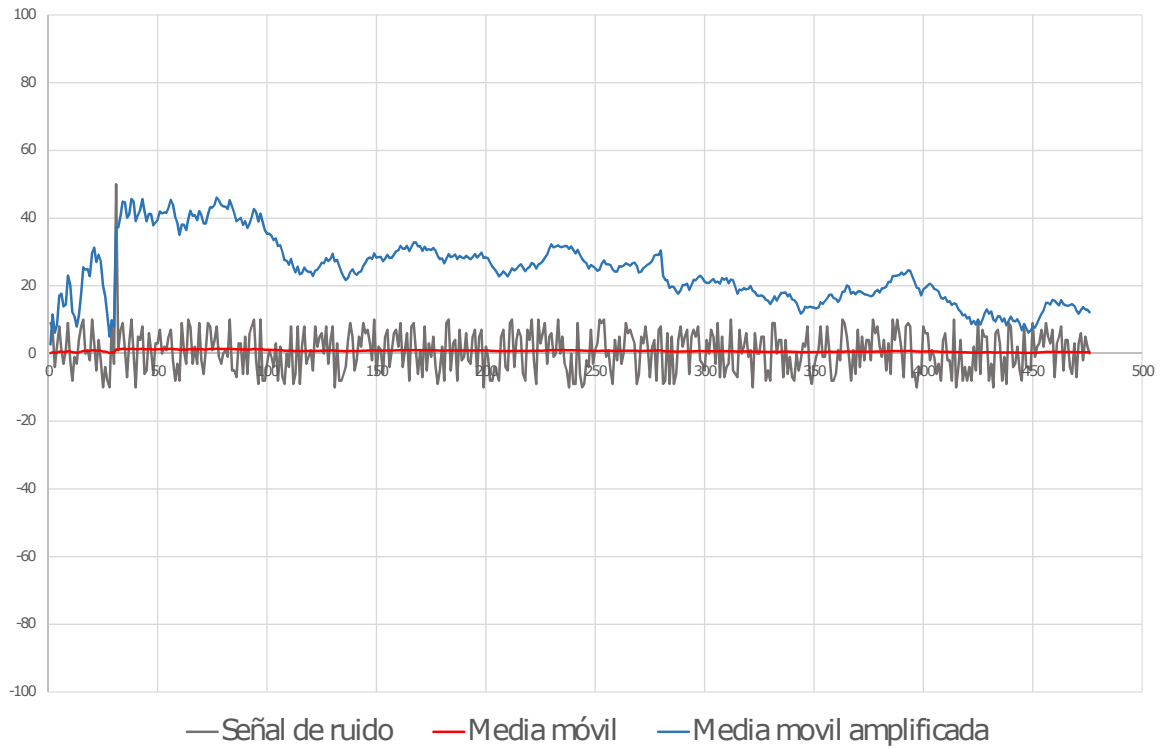


Figura 4.18: Promedio de una señal de ruido y un pico.

En las Figuras 4.19 y 4.20 se muestran el comportamiento de un detector de umbral adaptativo en dos implementaciones reales con diferentes preprocesamientos para detectar picos neuronales.

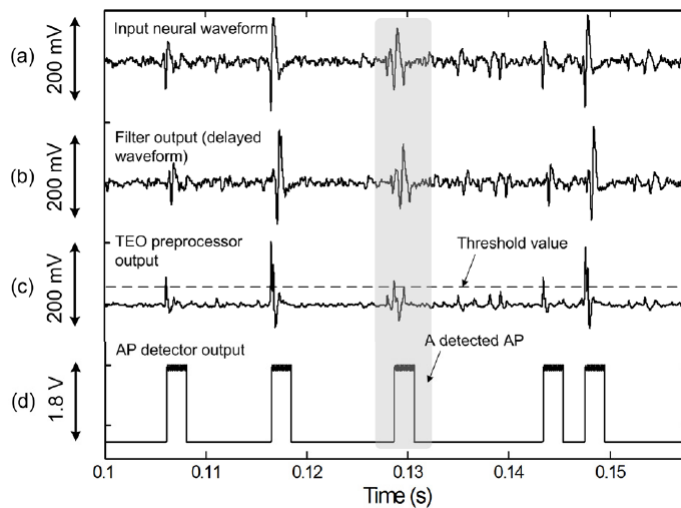


Figura 4.19: Detector de umbral adaptativo para picos neuronales basado en la energía de la señal. Figura tomada de [45].

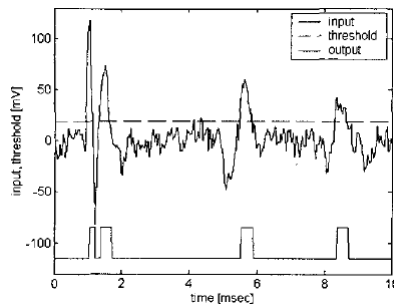


Figura 4.20: Detector de umbral adaptativo para picos neuronales basado en el valor cuadrático medio. Figura tomada de [10].

4.5 Curva ROC

Las *curvas característica operativa del receptor* o más conocidas como “*curva ROC*” por sus siglas en inglés, se originan en la *teoría de detección de señales*. Esta teoría provee mecanismos para tomar decisiones en presencia de cierta incertidumbre y para efectos de este proyecto servirán para tener la capacidad de discernir o detectar la señal deseada en presencia de ruido [12].

Los ROC se propusieron durante la segunda guerra mundial por ingenieros en radares e ingenieros eléctricos para detectar objetos enemigos aéreos correctamente con radares y no confundirlos con manadas de aves en Pearl Harbor [9]. En la actualidad se utiliza en la medicina, psicología, radiografías, biometría, análisis de mercadeo y muchas otras áreas [36].

De acuerdo a Heeger, las curvas ROC son una representación gráfica que ilustra el rendimiento de un clasificador binario conforme el umbral de discriminación cambia. Estas curvas grafican la sensibilidad contra la especificidad para cualquier clasificador binario. Dicho de otra forma, se grafica la razón de verdaderos positivos (RVP) contra la razón de falsos positivos (RFP), para un valor de umbral o criterio de discriminación específico. Un ejemplo de una gráfica ROC se muestra en la Figura 4.21.

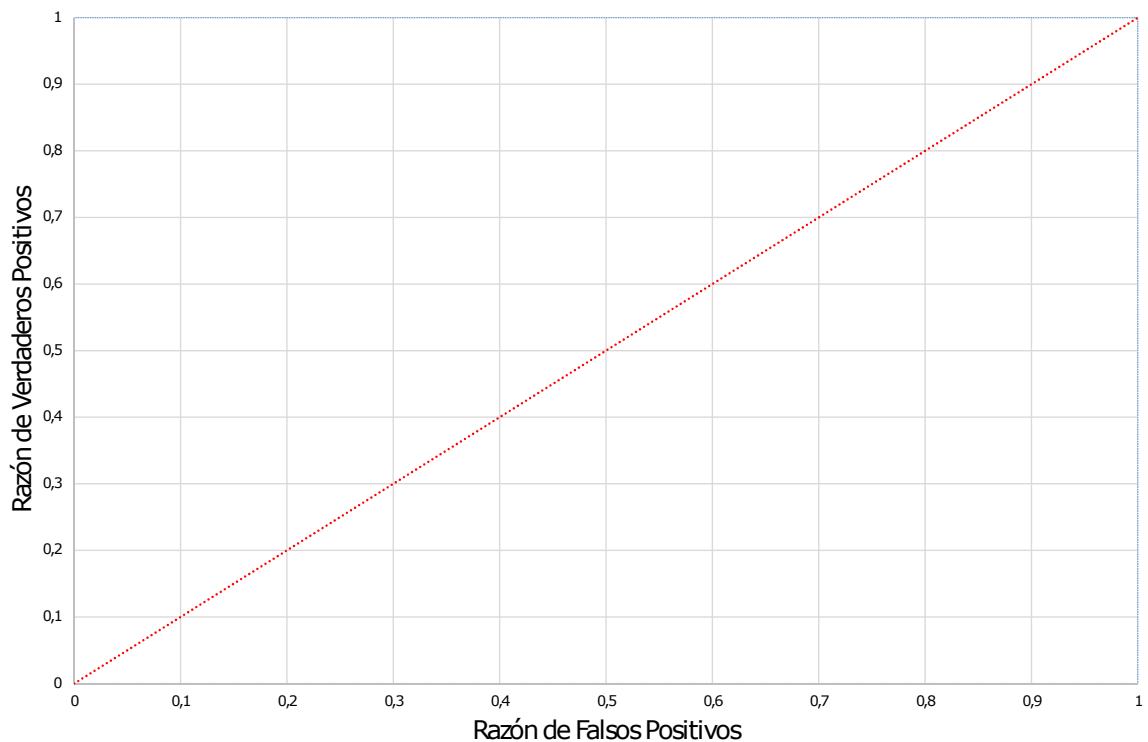


Figura 4.21: Gráfico de una curva ROC con la *diagonal de no discriminación*.

La diagonal en rojo mostrada en la Figura 4.21 es conocida como *diagonal de no discriminación*. Cualquier par ordenado de (RFP,RVP) que se encuentre sobre la diagonal se considera clasificado de forma aleatoria, mientras que puntos ubicados sobre esta línea representan que el detector clasifica mejor que de manera aleatoria y puntos por debajo indican que clasifica peor que aleatoriamente.

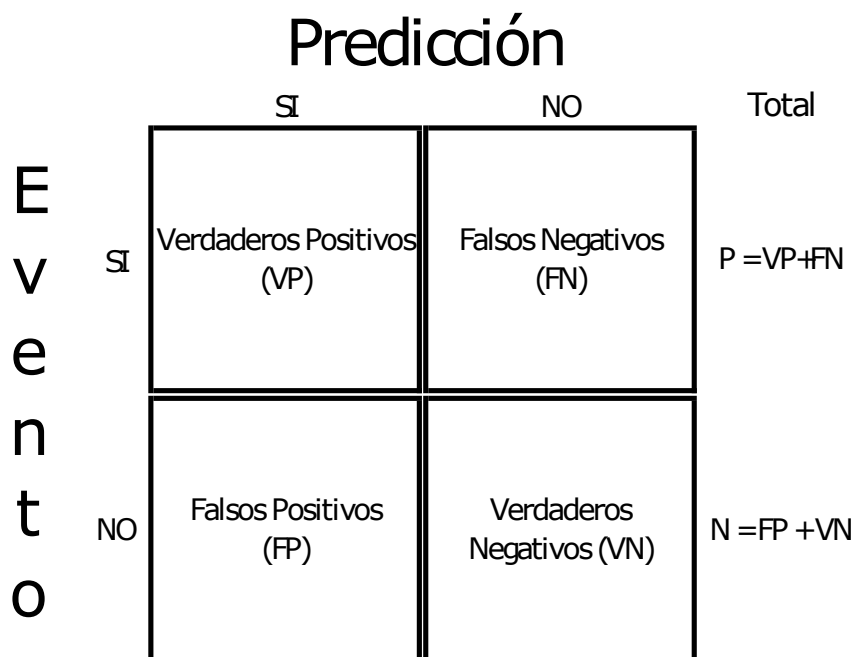
Dicho esto, el punto (0,1) se le llama una clasificación perfecta ya que implica que no hay ningún falsa alarma y por otra parte el 100% de las detecciones son correctas, entre más se alejen los puntos de la diagonal de no discriminación hacia el punto (0,1) representa que el detector es un buen clasificador.

La ocurrencia de cualquier evento clasificado de forma binaria en presencia de incertidumbre genera realmente 4 posibles respuestas: un **verdadero positivo (VP)**, que ocurre cuando se detecta un evento que realmente ocurrió; un **falso positivo (FP)**, cuando se detectó un evento que no ocurrió realmente (falsa alarma); un **verdadero negativo (VN)**, cuando no se detectó un evento que no ocurrió; y un **falso negativo (FN)**, cuando no se detectó un evento que realmente ocurrió (ver Tabla 4.2 para un resumen [27]).

Tabla 4.2: Posibles salidas de un detector binario

Tipo de predicción	Interpretación	Resultado del detector	Evento
VP	Verdadero Positivo	Positivo	Positivo
FP	Falso Positivo	Positivo	Negativo
VN	Verdadero Negativo	Negativo	Negativo
FN	Falso Negativo	Negativo	Positivo

De lo anterior también se puede formular una *Matriz de confusión* 2x2 como se muestra en la Figura 4.22 para clasificar las predicciones del detector.

**Figura 4.22:** Matriz de confusión.

Esta matriz ayuda a visualizar fácilmente como se obtienen los pares ordenados (RFP,RVP) para la construcción de la curva ROC por cada umbral de detección dado.

La razón de falsos positivos se calcula cómo

$$RFP = \frac{FP}{FP + VN} \quad (4.7)$$

y la razón de verdaderos positivos cómo

$$RVP = \frac{VP}{VP + FN} \quad (4.8)$$

donde los denominadores corresponden a la totalidad de predicciones positivas y negativas respectivamente.

Adicionalmente se puede definir también la *precisión de una curva ROC* que se calcula cómo

$$P = \frac{VP + VN}{P + N} \quad (4.9)$$

donde este valor indica cuál punto de la curva ROC es realmente el que brinda mejores resultados en la detección.

En la Figura 4.23 se muestra una curva ROC de un estudio hipotético generada a partir de siete valores de umbral diferentes. Se puede apreciar que el punto (0.15,0.85) es el más alejado de la diagonal de no discriminación y más cercano al clasificador perfecto. El valor de umbral utilizado para ese experimento en específico sugiere ser el que mejor discrimina en este clasificador.

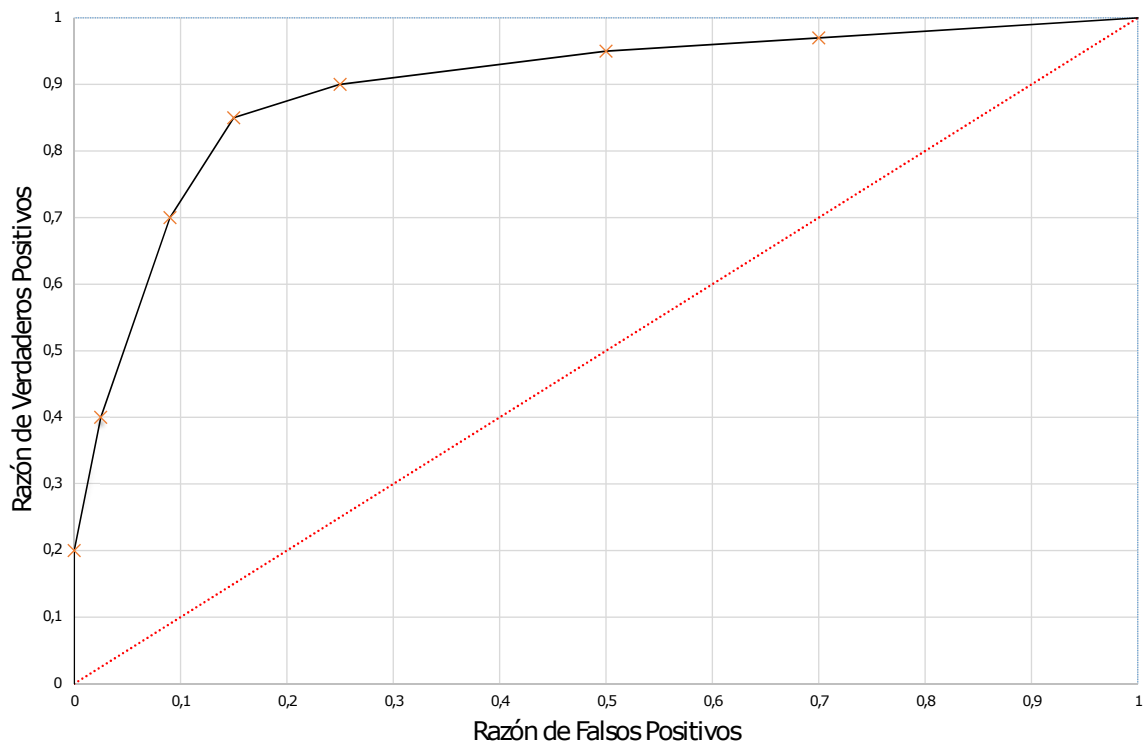


Figura 4.23: Curva ROC para un experimento con 7 valores de umbral diferentes.

4.6 LabVIEW

4.6.1 ¿Qué es LabVIEW?

LabVIEW o “Laboratory Virtual Instruments Engineering Workbench” es una de software de desarrollo basado en programación gráfica creada por la empresa National Instruments. Esta herramienta ofrece la capacidad de resolver problemas de manera más rápida y eficiente, además ofrece una amplia integración con hardware [20].

4.6.2 Instrumentos virtuales

Los programas creados en LabVIEW se conocen como instrumentos virtuales o VIs por sus siglas en inglés. Principalmente los VIs están compuestos por dos ventanas conocidas como *Panel frontal* y *Diagrama de bloques* [21].

Panel frontal

La ventana del panel frontal es la interfaz de usuario. Esta ventana por lo general tiene apariencia de instrumentos físicos o de equipo electrónico como osciloscopios y multímetros, de donde el nombre de instrumento virtual, tal y como se muestra en la Figura 4.24.

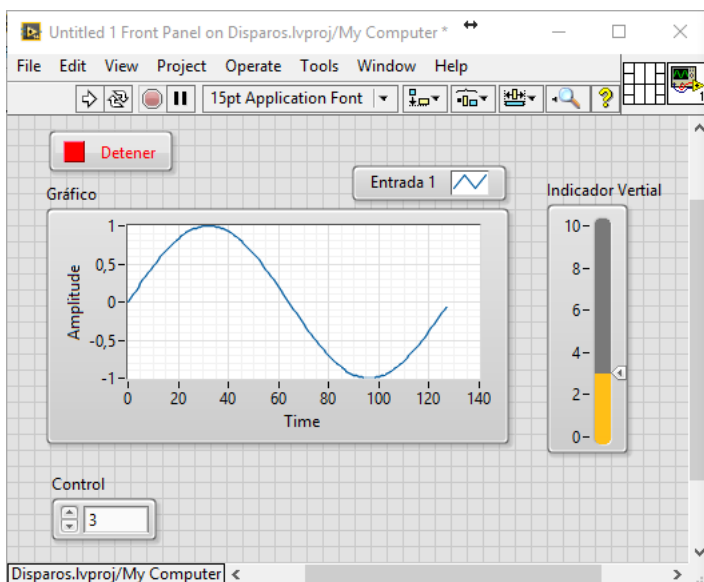


Figura 4.24: Panel frontal de LabVIEW.

Paleta de controles

La paleta de controles contiene los controles e indicadores que utiliza para crear el panel frontal. En la Figura 4.25 se muestra un ejemplo de la paleta de controles.

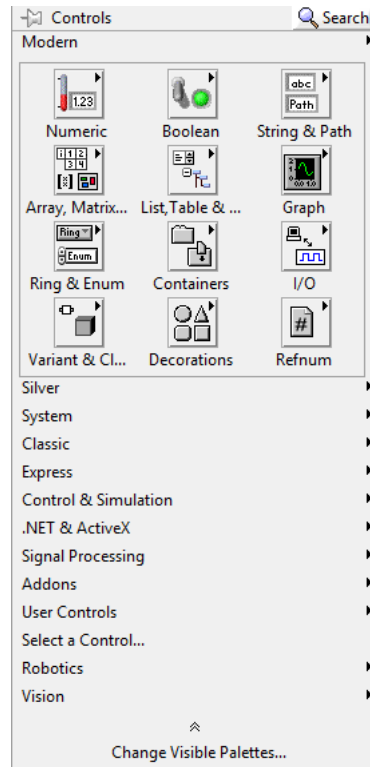


Figura 4.25: Paleta de Controles de LabVIEW.

Diagrama de bloques

El código de programación gráfico de LabVIEW se encuentra contenido en esta ventana, que puede contener subVIs, funciones, constantes, estructuras y cables; en la Figura 4.26 se muestra un ejemplo de un diagrama de bloques.

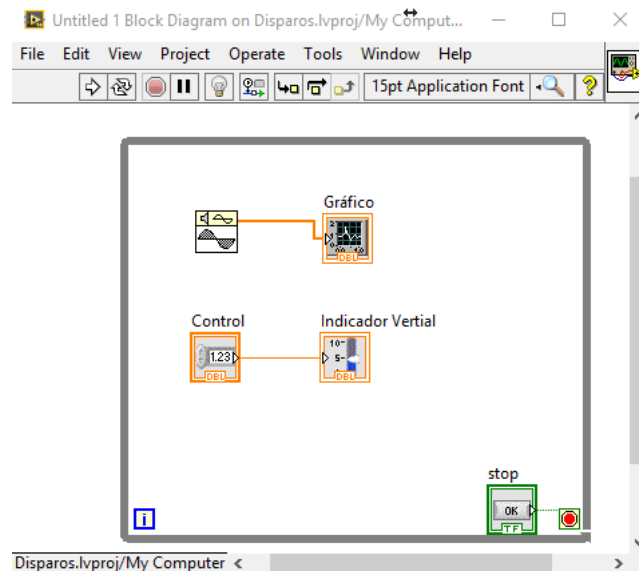


Figura 4.26: Diagrama de bloques de LabVIEW.

Paleta de funciones

La paleta de funciones contiene todos los subVIs, funciones y constantes que se utiliza para programar gráficamente. Este contenido puede variar dependiendo de la versión de LabVIEW, dependencias o bibliotecas instaladas. En la Figura 4.25 se muestra una paleta de funciones.

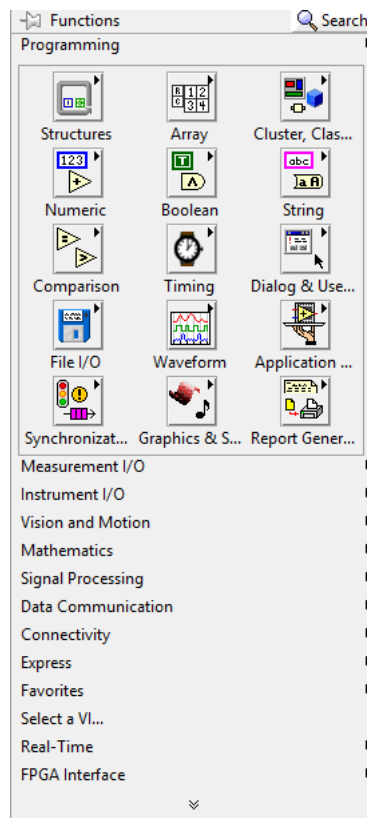


Figura 4.27: Paleta de Controles de LabVIEW.

4.6.3 Módulo LabVIEW FPGA

LabVIEW cuenta con un módulo de FPGA el cual extiende el alcance de la plataforma de desarrollo gráfico de LabVIEW para que el código gráfico pueda ser compilado y programar una FPGA o un hardware de E/S reconfigurables (NI provee una línea de hardware llamada RIO que ofrece este soporte de FPGA). Este módulo expande muchas de las facilidades que LabVIEW tiene para programar en FPGA, puesto que los lenguajes de descripción de hardware sin lugar a duda son más complicados que utilizar esta herramienta.

No obstante, código gráfico utilizado en LabVIEW no es completamente compatible con este módulo por lo que no siempre es posible sintetizar el mismo código. Para ello, hay que tomar lo siguiente para adaptar el código:

- Estructura de datos: no se pueden emplear números en coma flotante.
- Funciones: no todas las funciones están disponibles , tal como las de procesamiento de señal o algunas de matemática como dividir o sumar un arreglo.

- Recursos de la unidad de destino: dependiendo de la solución de RIO utilizado, el SoC disponible va a tener un número específico de recursos por lo que un VI grande puede no alcanzar en el dispositivo de destino.
- Estructuras: las estructuras cíclicas no anidadas se ejecutan en paralelo, por lo que se debe hacer uso de estructuras de control de secuencia o tiempo para la correcta ejecución del código.
- Arreglos: el uso de arreglos grandes no es recomendado, pues los arreglos mayores de 32bits que se usen en el panel frontal requieren dos flip-flops por cada bit en lugar de uno [23] y para arreglos mayores de 300bytes se aconseja utilizar otro método de almacenamiento como memorias o FIFOs [13].
- Sincronización: se recomienda de ser posible el uso de estructuras de lazo temporizadas de un único ciclo o SCTL por sus siglas en inglés. Estas estructuras fuerzan al código a que se ejecuten en un solo ciclo de reloj, ello además ayuda a reducir recursos [19].

Para mayor información y/o dudas se recomienda consultar el manual de LabVIEW FPGA [17] que se encuentra en el sitio oficial de National Instruments.

4.6.4 Modo de ejecución de VIs

Ejecución de VIs en la computadora de desarrollo

LabVIEW sigue un modelo de flujo de datos particular para ejecutar VIs, lo que significa que un nodo del diagrama no se ejecuta hasta que esté todas las entradas listas. Por ejemplo, en la Figura 4.28 la resta no se ejecuta hasta que la suma esté lista. Ello es análogo a lenguajes de programación como Visual Basic y C++, entre otros que siguen un flujo de ejecución secuencial conforme el código descende línea por línea u otros lenguajes que se ejecutan en paralelo.

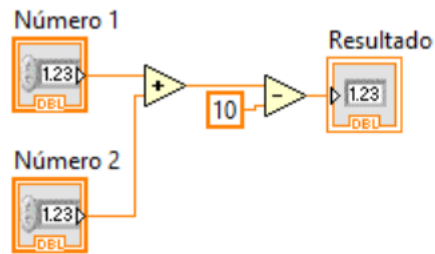


Figura 4.28: Ejemplo de flujo de datos en LabVIEW.

Ejecución de VIs en la FPGA

LabVIEW FPGA añade código adicional para forzar el flujo de datos en cadena tal y como se muestra en la Figura 4.29. Aquí se puede observar lógica que se utiliza para almacenar los resultados de las funciones y para hacer funcionar el resto del flujo de datos en cadena forzado.

Es evidente el consumo de recursos que este tipo de ejecución ocasiona, para solucionar esto se recomienda utilizar estructuras SCTLs. Sin embargo, un mayor número de funciones se encuentran excluidas en este tipo de estructuras.

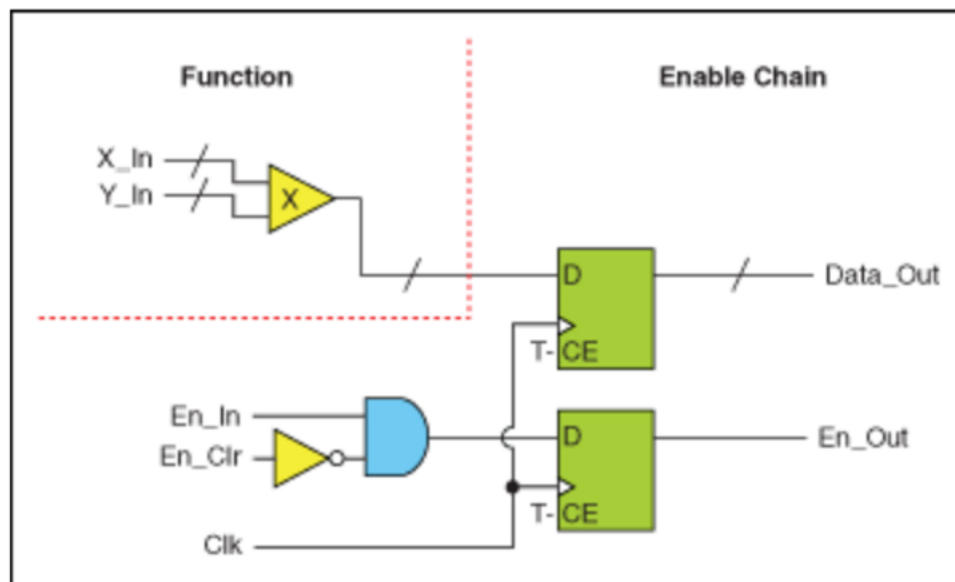


Figura 4.29: Ejemplo de flujo de datos en LabVIEW FPGA. Figura tomada de [14].

4.6.5 Método de referencia a VI FPGA

Este método se utiliza para comunicar un VI que se ejecuta en la FPGA con un VI que se ejecuta en la computadora de desarrollo de manera interna. De esta manera es posible estimular las entradas del VI de FPGA con funciones o subVIs que no están disponibles en el módulo de FPGA, así como la recolección de datos generados en la FPGA.

En la Figura 4.30 se muestra la paleta de funciones de interfase FPGA disponibles para implementar este método. Sin embargo para comprender cómo utilizar este método consultar el manual de comunicación de FPGA con el computador para más información [22].

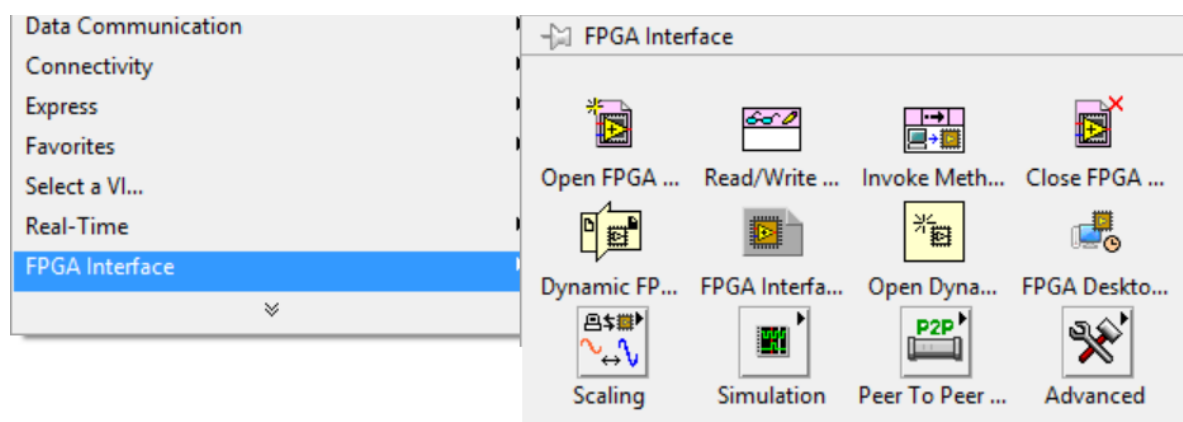


Figura 4.30: Funciones de interfase FPGA.

Capítulo 5

Sistema de detección de patrones acústicos

En la sección 1.2 se introdujo una descripción general de la solución con el uso de bloques funcionales del sistema de detección propuesto. En las siguientes secciones de este capítulo se explica detalladamente cada uno de los bloques funcionales o módulos del sistema implementado.

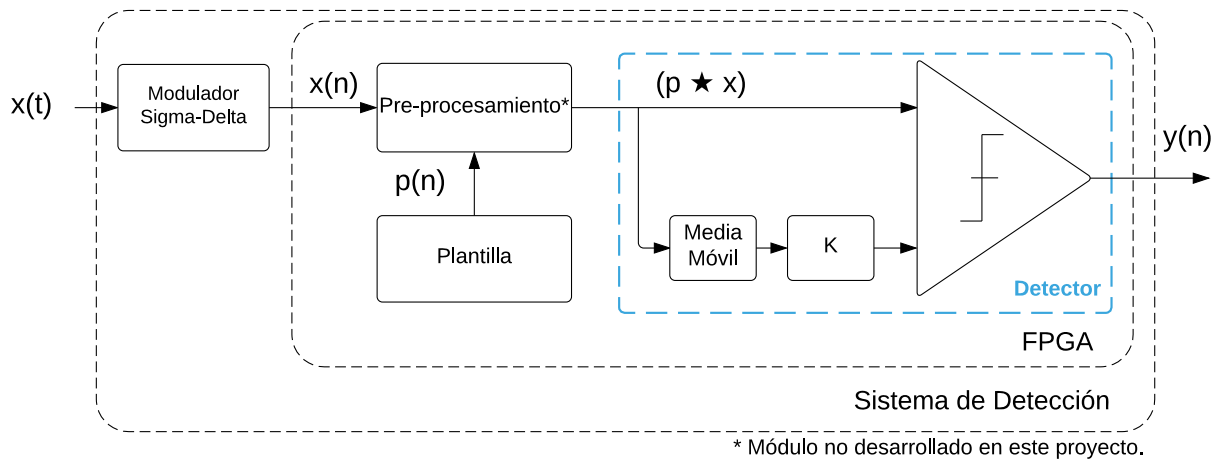


Figura 5.1: Diagrama detallado del sistema de detección.

La Figura 5.1 se muestra un diagrama detallado de la solución implementada. Es importante aclarar que el bloque pre-procesamiento no fue implementado en hardware o desarrollado en este proyecto, ni se fue posible utilizar un módulo en hardware puesto que este módulo se desarrolló en paralelo a este proyecto. Sin embargo, es mostrado en este capítulo como

parte de la solución puesto que es vital para comprender el funcionamiento y justificar los módulos que conforman la solución de este sistema de detección de patrones acústicos. Para fines de esta solución y como se explica en la sección 5.2 se utilizó un módulo de alto nivel para simular la respuesta de este bloque.

En la Tabla 5.1 y 5.2 se muestra una descripción de las E/S del sistema y los módulos respectivamente.

Tabla 5.1: Descripción de los bloques del sistema de detección de patrones acústicos.

Módulo	Descripción
Modulador Sigma-Delta	Es el ADC que convierte la señal $x(t)$ a <i>bitstream</i> .
Pre-procesamiento	Este es el bloque que calcula la correlación cruzada entre la señal $x(n)$ y $p(n)$.
Plantilla de disparo	Es en donde se almacena el patrón o plantilla $p(n)$, que se procesa junto con la señal $x(n)$.
Detector de umbral adaptativo	Este detector se compone de los siguientes tres bloques: <ul style="list-style-type: none"> - Media Móvil: Calcula la media móvil de la correlación cruzada de $x(n)$ y $p(n)$. - K: Es una constante que amplifica K veces la media móvil. - Comparador: Corresponde a un comparador de nivel digital.

Tabla 5.2: Señales del sistema de detección de patrones acústicos.

Señal	Descripción
$x(t)$	Señal de audio analógica
$x(n)$	Señal de audio digital
$p(n)$	Plantilla de los patrones acústicos a detectar
$x \star p$	Correlación de $x(n)$ y $p(n)$
$y(n)$	Resultado del detector de umbral

5.1 Modulador Sigma-Delta

5.1.1 Modelo en alto nivel

La función principal de este módulo es de simular un modulador sigma-delta para poder obtener una representación en *bitstream* de una señal cualquiera en el dominio del tiempo. Esta señal es la que estimula el módulo de pre-procesamiento de correlación. Para llevar a cabo esto se utilizó LabVIEW y como ejemplo un VI de un modulador sigma-delta de primer orden disponible en la comunidad oficial de desarrolladores de National Instruments [15].

Cómo se vio en la sección 4.2; se mejora la relación de señal a ruido (SNR) conforme se utiliza un modulador de mayor orden. Por este motivo y porque además se pretende utilizar para la implementación final de este sistema; un modulador sigma-delta analógico de segundo orden para la implementación en hardware final se usó el VI que se muestra en la Figura 5.2 de un modulador sigma-delta de segundo orden de acuerdo a la topología mostrada en la sección 4.2.3.

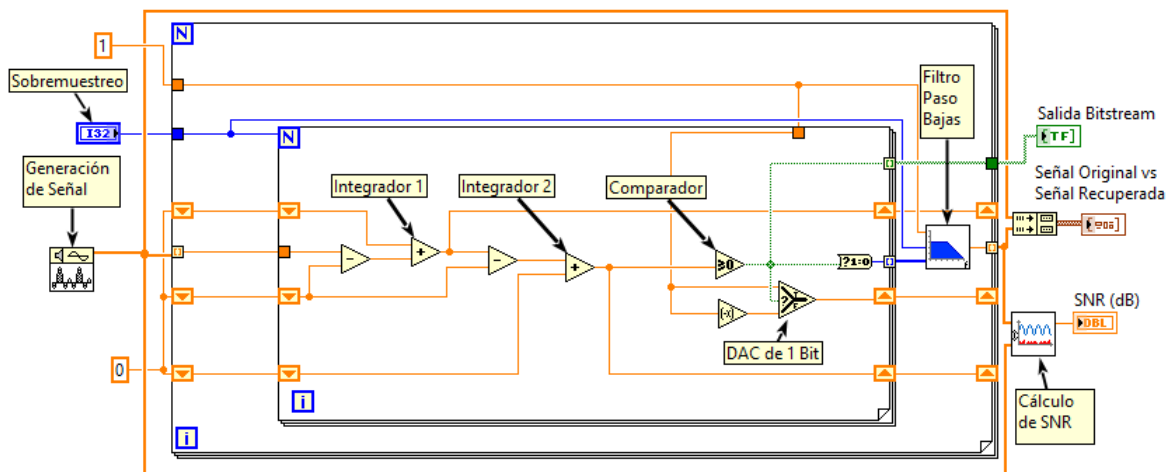


Figura 5.2: VI de un conversor sigma delta de segundo orden. Sólo se utilizó en este proyecto el modulador, pues únicamente se requería la señal bitstream.

Se puede apreciar que el VI anterior, además del modulador sigma-delta, también incluye un filtro pasa bajas para obtener la señal codificada en PCM (y usar entonces como ADC el VI). Puesto que el procesamiento en este proyecto será *bitstream*, esta es la salida utilizada. La de PCM sin embargo puede servir como entrada a un módulo de correlación estándar para fines de verificación del prototipo inicial.

En la Tabla 5.3 se muestran las entradas y salidas del VI del modulador de sigma-delta creado en LabVIEW.

Tabla 5.3: Entradas y salidas para el VI del modulador sigma-delta de segundo orden.

E/S	Descripción
Señal	Señal en el dominio del tiempo a convertir
Factor de sobremuestreo	Factor al cual se va a sobremuestrear la señal
Salida en <i>bitstream</i>	Señal en en formato <i>bitstream</i>
SNR	Razón de señal-a-ruido en dB
Gráficas	Gráficas de la señal original vs la recuperada

5.1.2 Implementación hardware de la etapa de adquisición

Puesto que el fin del sistema es detectar en el campo disparos, se desarrolló la unidad de adquisición de datos a partir de un modulador sigma-delta comercial. Aunque no fue posible por razones de tiempo comprobar el funcionamiento del sistema completo con esta unidad, se ofrecen aquí los detalles de la unidad. Este hardware queda disponible para la continuación del proyecto en la etapa de traslado completo a una FPGA.

Se desarrolló una tarjeta de circuito impreso (PCB). Luego de estudiar algunos modelos AD7403, AD7405, ACPL-C797 se decidió por razones de costo y resolución usar el IC de la compañía Analog Devices, AD7405 que es un modulador sigma-delta de segundo orden de 16bits, con un SNR típico de 88dB (esta cifra se derivó en parte de la hoja de datos del AD7405, pues éste posee un SNR de 88dB).

En la Figura 5.3 se muestra el circuito de aplicación típico para este IC extraído de la hoja de datos del mismo.

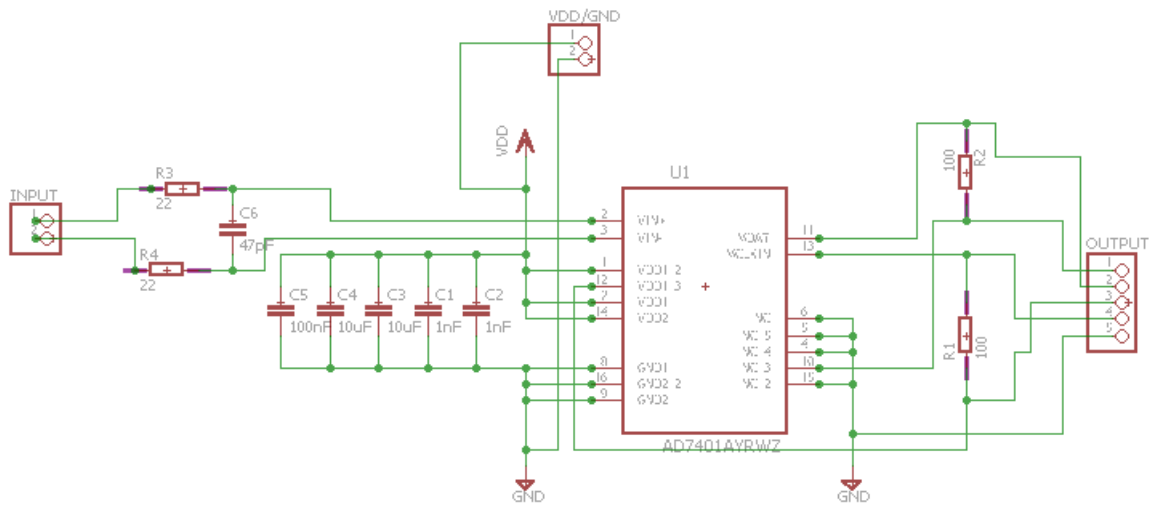


Figura 5.3: Esquemático del circuito de aplicación típico para el AD7405. El componente mostrado AD7401 es compatible a nivel de pines con el AD7405, y ya está dentro de la biblioteca de Eagle.

Se utilizó la herramienta de software de EagleCAD para diseñar el circuito impreso y se obtuvo el plano o *layout* que se muestra en la Figura 5.4.

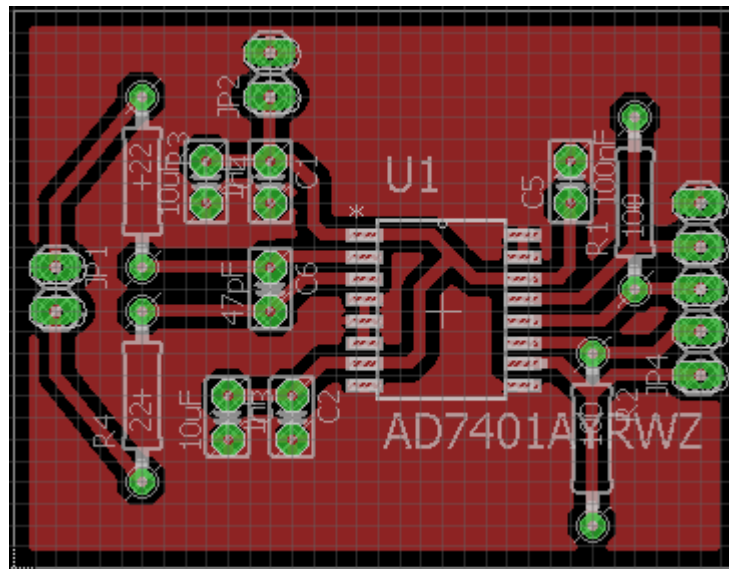


Figura 5.4: Circuito impreso del circuito típico para el AD7405.

Finalmente se realizó el impreso y se soldaron los componentes necesarios. Una fotografía del resultado se muestra en la Figura 5.5.

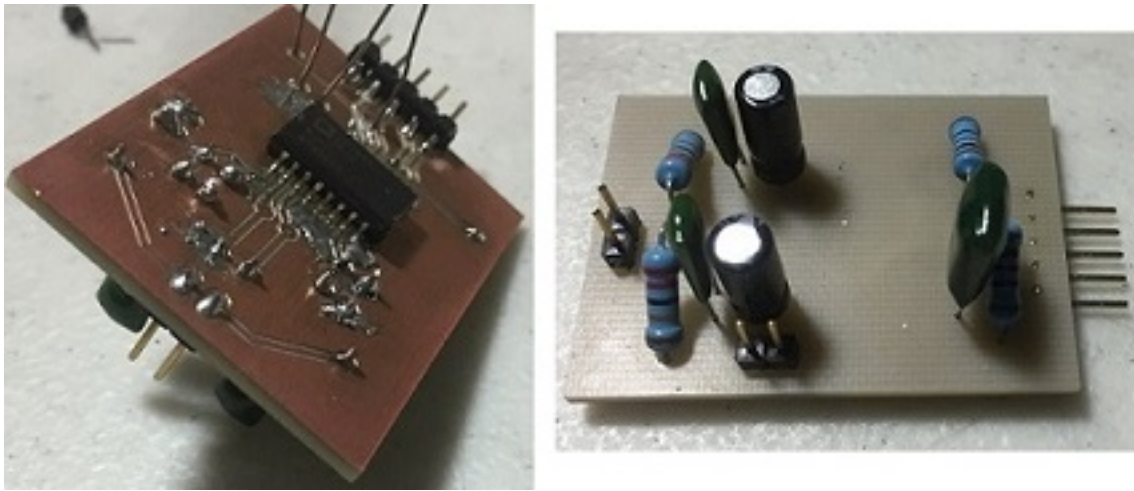


Figura 5.5: Fotografía del circuito impreso creado AD7405.

5.2 Pre-procesamiento

Este módulo se encarga de calcular la correlación cruzada de dos señales y corresponde al pre-procesamiento necesario de la señal $x(t)$ para alimentar al módulo del detector de umbral adaptativo que se explica en la sección 5.4.

Es importante de resaltar es que en un inicio se utilizó el modelo de alto nivel de LabVIEW del módulo de pre-procesamiento disponible en la paleta de “Procesamiento de Señal de LabVIEW” tal a como se muestra en la Figura 5.6.

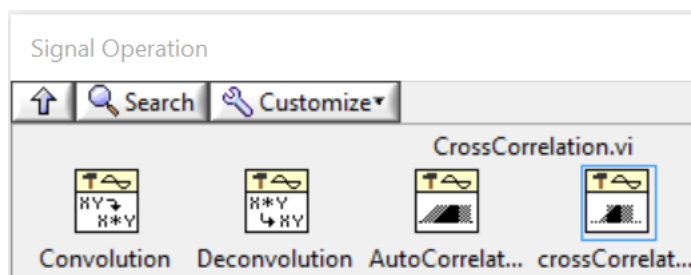


Figura 5.6: Paleta de VI's de operaciones de señal.

El VI de correlación cruzada cuenta con dos entradas (X y Y) y el resultado es la salida mostrada como R_{xy} , tal y como se muestra en la Figura 5.7.



Figura 5.7: VI de correlación cruzada.

Como se mencionó anteriormente este módulo se desarrolló en paralelo a este proyecto, por lo que probar el modulador sigma-delta y el detector de umbral adaptativo con este módulo implementado en a nivel de HDL en la FPGA no fue posible en el plazo del proyecto. Como alternativa, se utilizó más tarde el modelo de correlación *bitstream* generado en el proyecto paralelo [5]. Puesto que dicho modelo es el de referencia usado en el proyecto paralelo, las interfaces deben ser las mismas, al igual que la funcionalidad. Ello asegura que al sustituir por el modelo HDL final, no habrán diferencias funcionales.

Para efectos de este proyecto y poder comunicar el modelo en alto nivel con el detector de umbral en hardware se utilizó el método de referencia de LabVIEW a VI de FPGA expuesto en la sección 4.6.4.

5.3 Creación de plantillas

A continuación se explica el método utilizado para la generación de las plantillas de disparos y motosierras. Se recurrió a la metodología expuesta en [43], donde se generaron tres plantillas para la detección de patrones “QRS”, ondas “P” y “T” provenientes de los latidos del corazón que se usa en los electrocardiogramas.

El método consiste en extraer diferentes muestras del patrón deseado y promediar las muestras para así obtener una plantilla que representa la media de todos los patrones.

5.3.1 Plantilla de disparo

Se utilizó como base en el patrón acústico de un disparo u onda de choque N expuesto en la sección 4.1.1 para generar esta plantilla. Se utilizó la base de datos de grabaciones de disparos reales y se extrajo la onda de choque N de seis muestras de diferente calibre grabadas a 30m de distancia y se realizó el promedio de dichas muestras. En la Figura 5.8 se muestra en rojo el patrón obtenido.

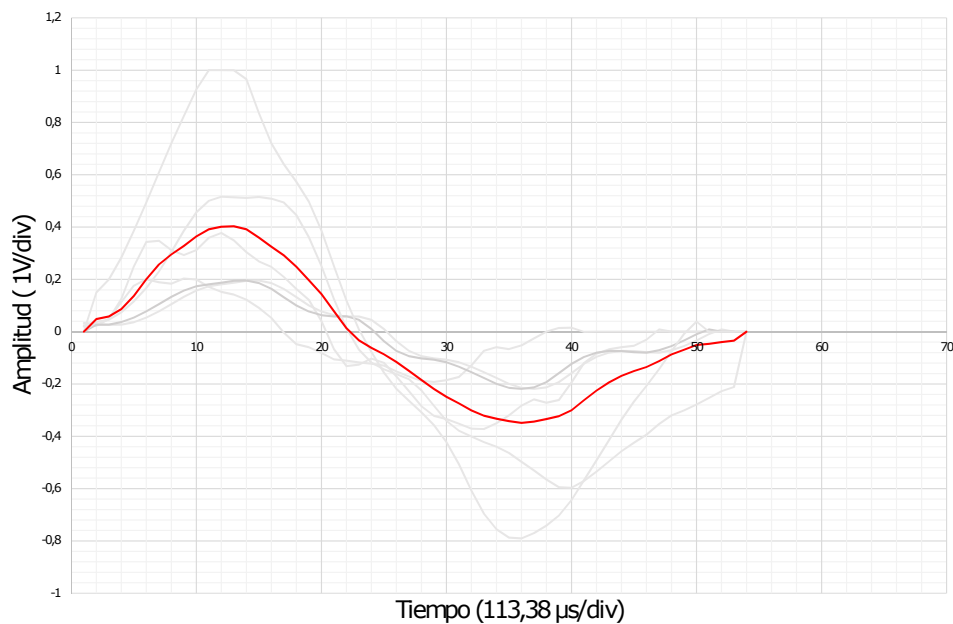


Figura 5.8: Plantilla de disparo generada a partir de 6 muestras patrones de armas de fuego calibre 22, 32 y 38 grabadas a 30m de distancias a ángulos de 0° y 90° del micrófono.

5.3.2 Plantilla de motosierra

Para generar esta plantilla se partió del hecho mostrado en la sección 4.1.2, en donde se observa que el patrón característico de las motosierras se basa en picos consecutivos muy periódicos.

Se utilizó el mismo método y se extrajeron seis muestras de picos de motosierras. Después de obtener el promedio de dichas muestras se obtuvo la plantilla mostrada en la Figura 5.9 con el color rojo.

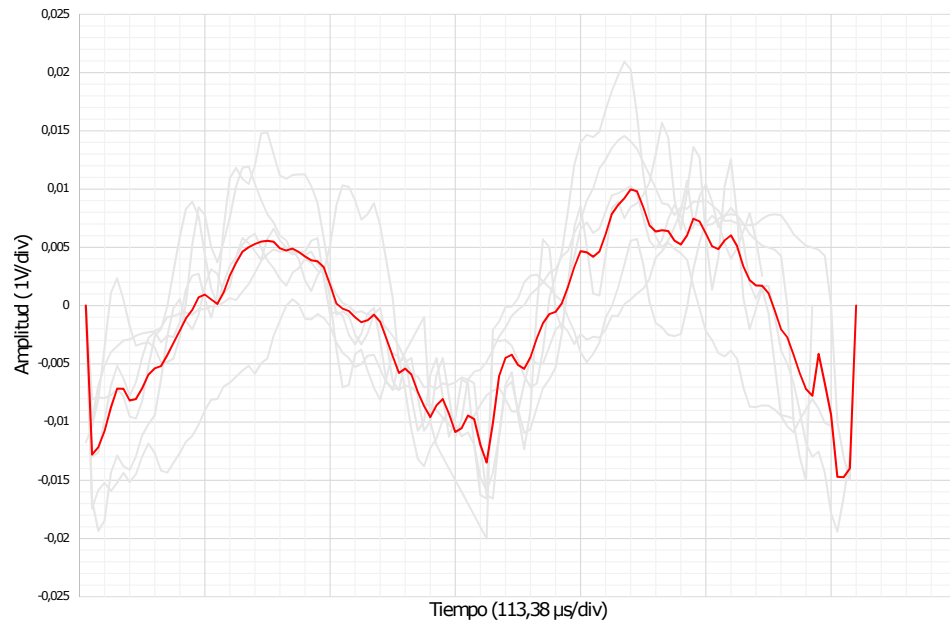


Figura 5.9: Plantilla de motosierra generada a partir de 6 muestras patrón de motosierras grabadas a 30 y 90m de distancias a ángulos de 0° , 90° y 180° del micrófono.

5.4 Detector de umbral adaptativo

5.4.1 Modelo en alto nivel

La Figura 5.10 muestra el VI del detector de umbral adaptativo creado basado en la arquitectura mostrada en la sección 4.4.1.

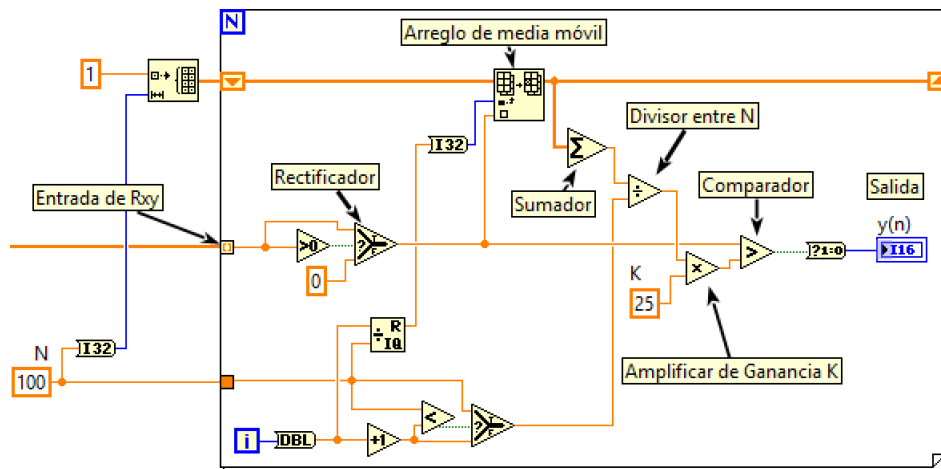


Figura 5.10: Modelo en alto nivel del detector de umbral adaptativo

Aquí se puede apreciar a diferencia de la topología general de un detector de umbral adaptativo el uso de un rectificar de media onda. Este rectificador se utiliza para que el detector

únicamente sea sensible a la correlación de una señal con otra, cuando están en fase. Un valor negativo de correlación indica que hay un grado similitud pero están invertidas en fase. También se puede apreciar que los valores para calcular la media móvil se almacenan en un arreglo para luego ser sumados y se utiliza un divisor para finalmente calcular la media de los valores. El color anaranjado representa que se utiliza coma flotante para todos los cálculos de éste módulo.

5.4.2 Traslado a la FPGA

Se procedió a realizar un diseño de una versión en digital implementado en una FPGA. Se usó LabVIEW con la extensión de FPGA para programar un SoC Xilinx Zynq 7010 embebido un National Instruments myRIO.

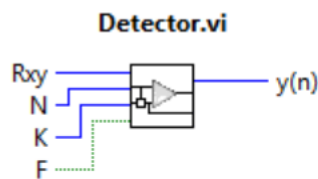


Figura 5.11: VI del detector de umbral adaptativo

La Figura 5.11 muestra el icono del VI del detector de umbral creado en LabVIEW junto con sus E/S.

En la Tabla 5.4 se describe el funcionamiento de las E/S de este módulo.

Tabla 5.4: E/S del VI del detector de umbral adaptativo

E/S	Descripción
Rxy	Resultado de la correlación $x(n)$ y $p(n)$
N	Ventana para el cálculo de la media móvil
K	Constante de ganancia que amplifica la media móvil
F	Bandera que indica que hay un dato nuevo de Rxy listo
$y(n)$	Salida del detector

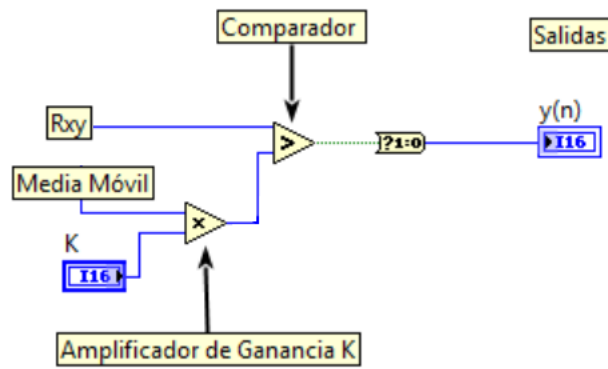


Figura 5.12: Código fuente del comparador de nivel y multiplicador por K

La figura 5.12 muestra el código fuente gráfico utilizado para implementar el comparador de nivel en la FPGA. Nótese el comparador booleano con dos entradas. Una de ellas es el valor de la correlación cruzada de $x(n)$ y $p(n)$ y la otra corresponde a la media móvil amplificada K veces, a la salida del comparador se convierte el valor booleano (falso/verdadero) del comparador a un entero (0/1) y se obtiene la señal señal $y(n)$.

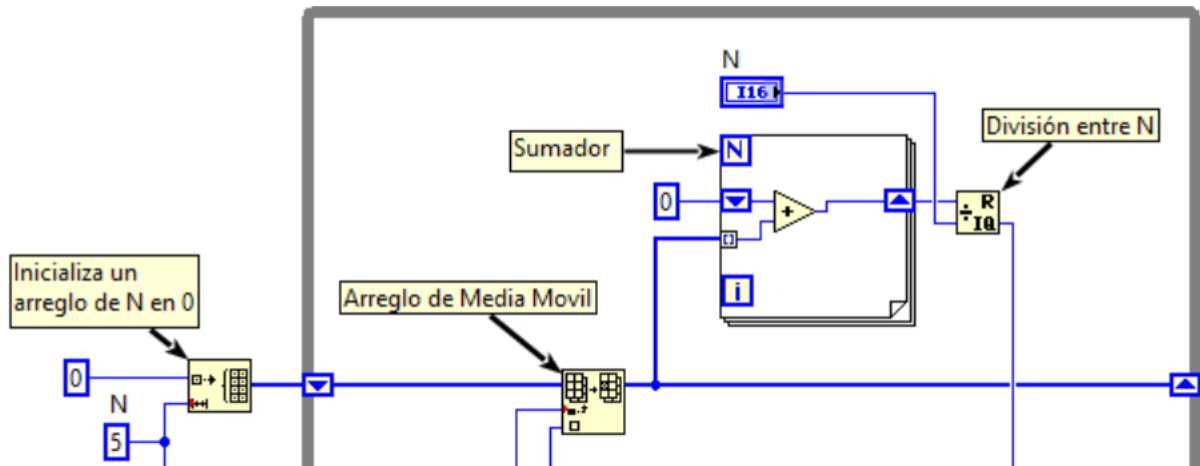


Figura 5.13: Código fuente para calcular media móvil

En la figura 5.13 se muestra el código fuente gráfico que se utilizó para realizar el cálculo de la media móvil. Para esta sección se utilizó un arreglo inicializado con ceros y de tamaño N . Con un análisis realizado se observó que la media móvil calcula el promedio de los últimos N datos de un conjunto de datos, por lo que en este arreglo se almacenan los últimos N datos para luego ser sumados en la etapa del sumador y luego se divide entre N .

Un detalle importante es que se debe determinar el valor de N previo a compilar el código para FPGA, puesto que LabVIEW FPGA no admite arreglos de tamaño variable. De ahí el

valor constante N a la izquierda.

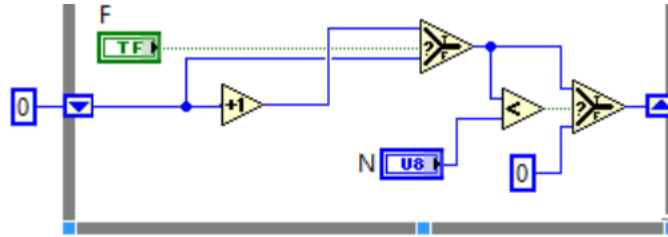


Figura 5.14: Código fuente del control del índice del arreglo

La Figura 5.14 muestra un contador que se activa cuando la bandera F de dato nuevo es verdadera y se reinicia una vez que llega a contar hasta N. Esto es para controlar el índice de lectura o escritora del arreglo en el cual el dato para calcular la media móvil se guarda.

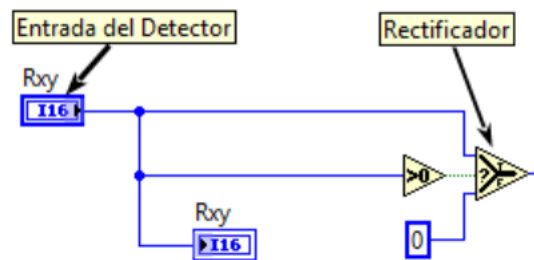


Figura 5.15: Código fuente de entrada de datos y rectificador

En la Figura 5.15 se muestra la sección del código de fuente del VI del detector de umbral en donde se observa la entrada de los datos de la correlación ($x \star p$) y un rectificador para obtener únicamente los valores de correlación cuando las señales están en fase.

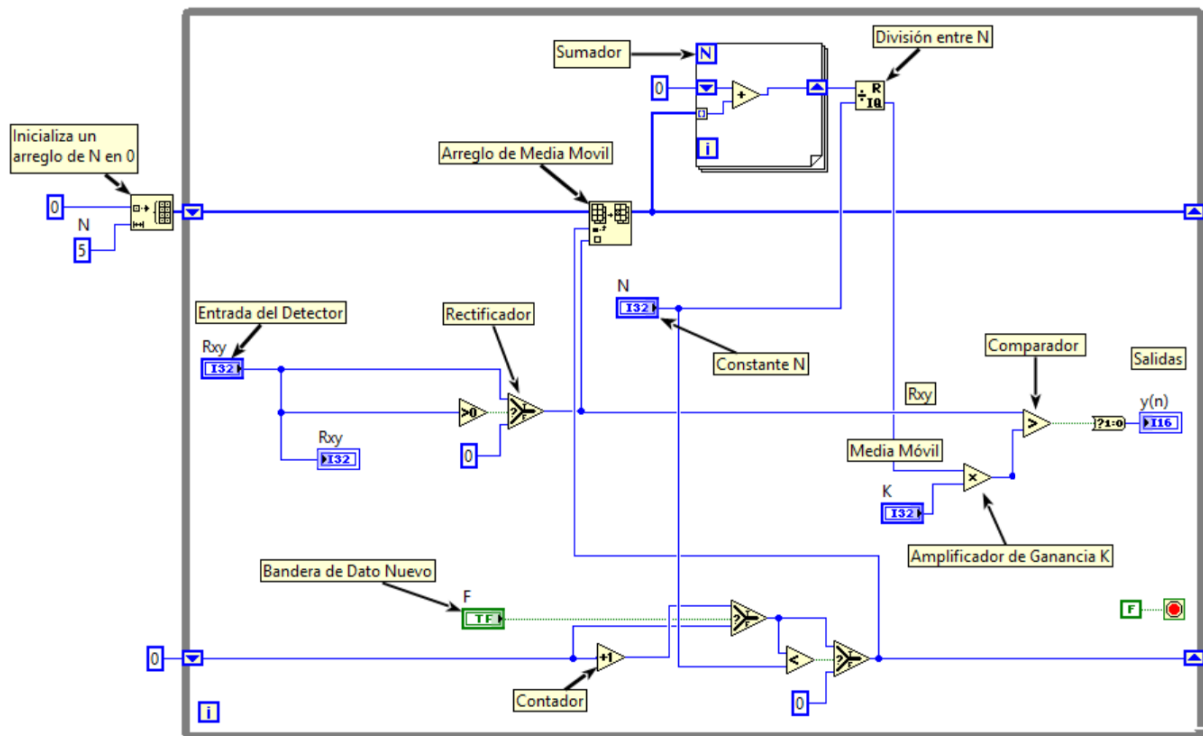


Figura 5.16: Código fuente gráfico del VI del detector completo

Finalmente en la Figura 5.16 se observa el código de fuente gráfico completo del VI del detector de umbral adaptativo en LabVIEW FPGA.

Capítulo 6

Resultados y análisis

En éste capítulo se muestran los resultados y el análisis de cada etapa del proyecto ordenados de acuerdo a los objetivos mencionados en la sección 2.2.2 .

6.1 Emulación del modulador Sigma-Delta

Este modulador fue desarrollado con el fin de poder convertir una señal analógica a formato de *bitstream* para que se puede diseñar y alimentar la etapa de pre-procesamiento que corresponde al correlador en este tipo de representación. Se siguió una arquitectura básica de un modulador sigma-delta de segundo orden tal y como se mostró en la sección 5.1.1.

Para realizar las siguientes pruebas se utilizó un OSF de 64, ya que de acuerdo al comportamiento teórico de éste modulador para este factor de sobremuestreo el SNR es de aproximadamente 80dB.

Se muestra a continuación un ejemplo sencillo de conversión a *bitstream* para señales de entrada de -1, -0.5, 0, 0.5 y 1V. Se aprecia en la Figura 6.1 que -1V y 1V es el rango de conversión. En este caso en particular se utiliza un OSF de 8 porque es suficiente para representar estos valores de forma exacta, al ser valores constantes.

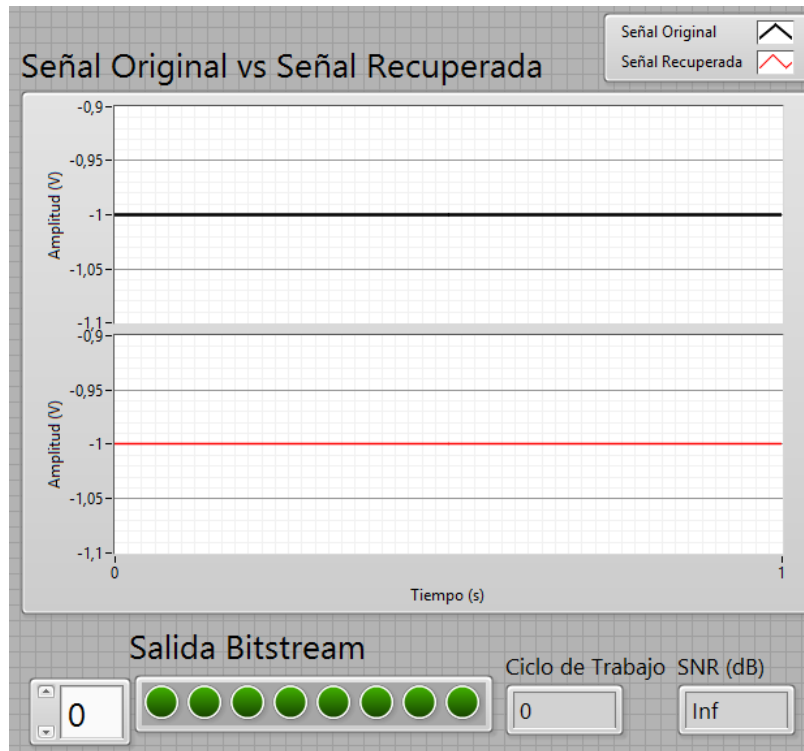


Figura 6.1: Resultado de conversión *bitstream* para una señal de entrada de -1V con un OSF de 8.

Se puede observar en la Figura 6.1 que la señal de entrada es -1V y la señal recuperada después del filtro es también -1V además la representación en *bitstream* al ser el extremo inferior del rango de conversión corresponde a 0s lo que da como resultado un ciclo de trabajo de 0%.

Por otro lado el SNR es infinito porque en

$$SNR = 10 \log_{10} \left(\frac{P_{Signal}}{P_{Noise}} \right) \quad (6.1)$$

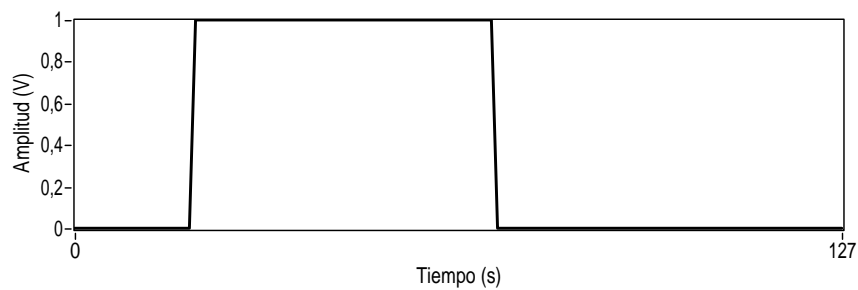
la potencia del ruido (P_{Noise}) es de 0.

Tabla 6.1: Prueba del módulo sigma-delta con valores constantes de -1V a 1V en pasos de 0.5V con un OSF de 8.

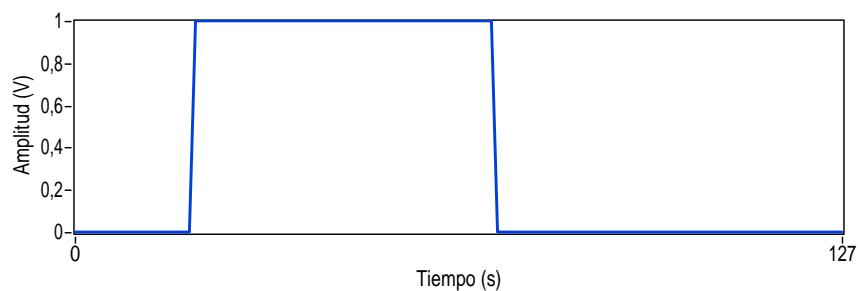
Tensión de entrada (V)	Representación en bitstream	Ciclo de trabajo (%)	SNR (dB)
1	11111111	100	Infinito
0.5	10111101	75	Infinito
0	10011001	50	Indefinido
-0.5	01000010	25	Infinito
-1	00000000	0	Infinito

Las siguientes figuras (de 6.2 a 6.6) muestran los resultados obtenidos utilizando señales analógicas como excitación del modulador, su recuperación después de convertida con un OSF de 64 y sus respectivos SNR. En este caso se omitió la representación en *bitstream* por razones obvias de espacio y comprensión. Para obtener las señales utilizadas en estas pruebas se utilizaron VIs de LabVIEW de la paleta de Generación de señales.

La Figura 6.2, muestra otro caso simple de la recuperación de una señal de un pulso después de convertida por el modulador sigma-delta.

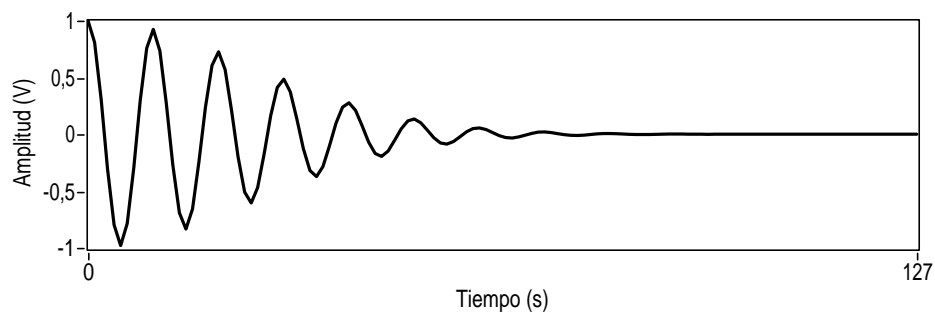


(a) Señal original

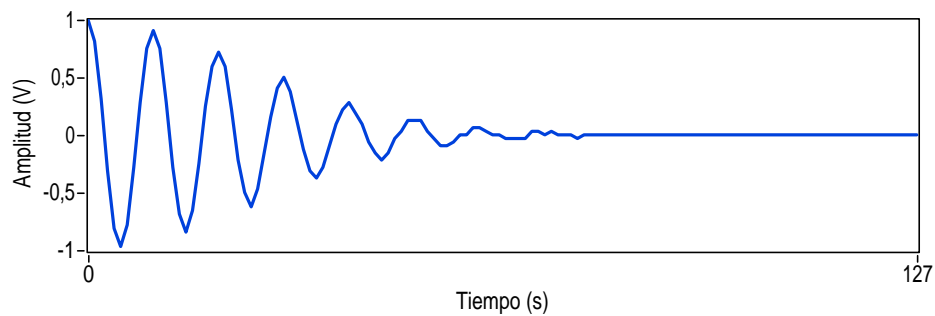


(b) Señal recuperada

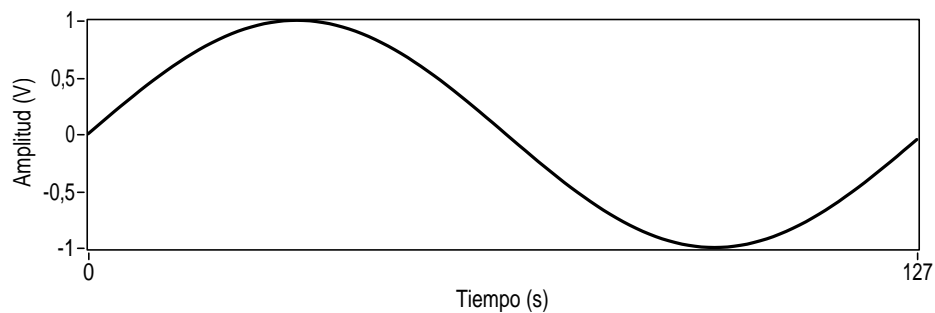
Figura 6.2: Recuperación de una señal de un pulso después de convertida en *bitstream*.



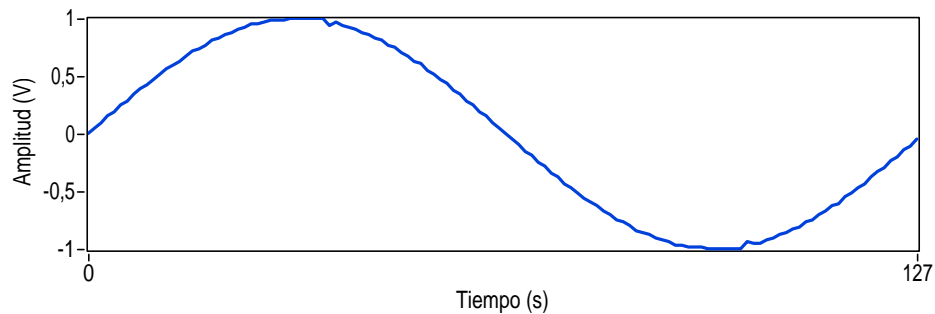
(a) Señal original



(b) Señal recuperada

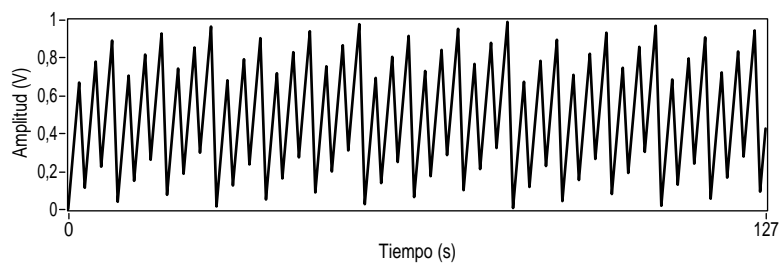
Figura 6.3: Recuperación de una señal sinusoidal gaussiana después de convertida en *bitstream*.

(a) Señal original

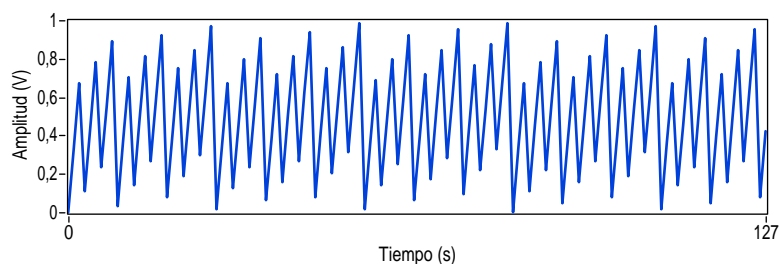


(b) Señal recuperada

Figura 6.4: Recuperación de una señal sinusoidal después de convertida en *bitstream*.

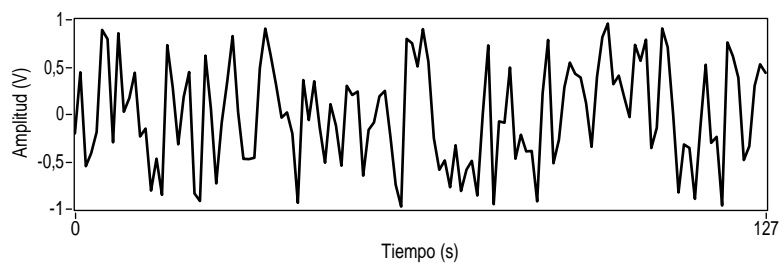


(a) Señal original

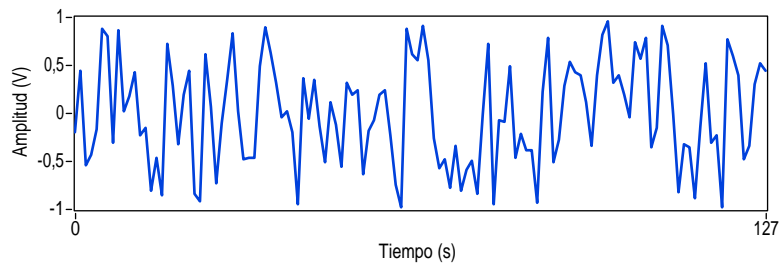


(b) Señal recuperada

Figura 6.5: Recuperación de una señal quasi-aleatoria de Halton después de convertida en *bitstream*.



(a) Señal original



(b) Señal recuperada

Figura 6.6: Recuperación de una señal de ruido después de convertida en *bitstream*.

La Tabla 6.2 resume los resultados obtenidos de SNR para diferentes señales variantes en el tiempo convertidas a *bitstream* con un modulador sigma-delta con un OSF de 64.

Tabla 6.2: Resultados obtenidos de SNR para señales variantes en el tiempo con un OSF de 64.

Tipo de señal	SNR (dB)
Pulso	Infinito
Sinusoidal Gaussiana	73.73
Sinusoidal	77.5
Quasi-aleatoria de Halton	77.88
Ruido	77.17

El SNR de infinito que se obtuvo en los diferentes escenarios es porque en este ambiente simulado las señales utilizadas para excitar el modulador son ideales y no tienen ruido por lo que su recuperación es exacta.

6.2 Implementación en FPGA del módulo de detección

En la sección 5.4 se mostró el VI del modelo en alto nivel en la Figura 5.10 y el VI de FPGA en la Figura 5.16. Las diferencias fundamentales de ambas versiones de VIs yacen en las incompatibilidades de LabVIEW FPGA para compilar exactamente el mismo código vistas en la sección 4.6.3, a saber: uso de enteros en vez de números en coma flotante, funciones no disponibles y el uso de un contador con una bandera en la versión de FPGA para sincronizar el procesamiento del cuándo existe un nuevo valor de correlación.

La Figura 6.7 muestra las funciones de sumar arreglo y división que se utilizaron en el modelo de alto nivel y no son compatibles con LabVIEW FPGA y la sustitución utilizada para realizar la misma función en la FPGA.

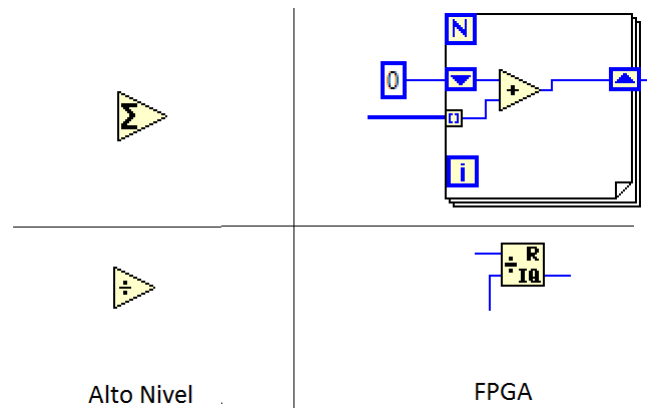


Figura 6.7: Comparación de funciones no disponibles en LabVIEW y su análogo en LabVIEW FPGA

El hecho de utilizar números enteros no es problema porque el módulo de correlación cruzada en *bitstream* tiene una salida en binario por lo que se encarga de solucionar el problema numérico. En el caso del modelo inicial, se parte del módulo original de correlación de LabVIEW mientras que en la prueba ya real en FPGA se usa el modelo bitstream de [5]. El primero usa representación en coma flotante, mientras el otro usa notación en coma fija y, por ello, es menos preciso. Aún así, como se ve en las Figuras 6.8 y 6.9, la precisión será suficiente para obtener resultados satisfactorios.

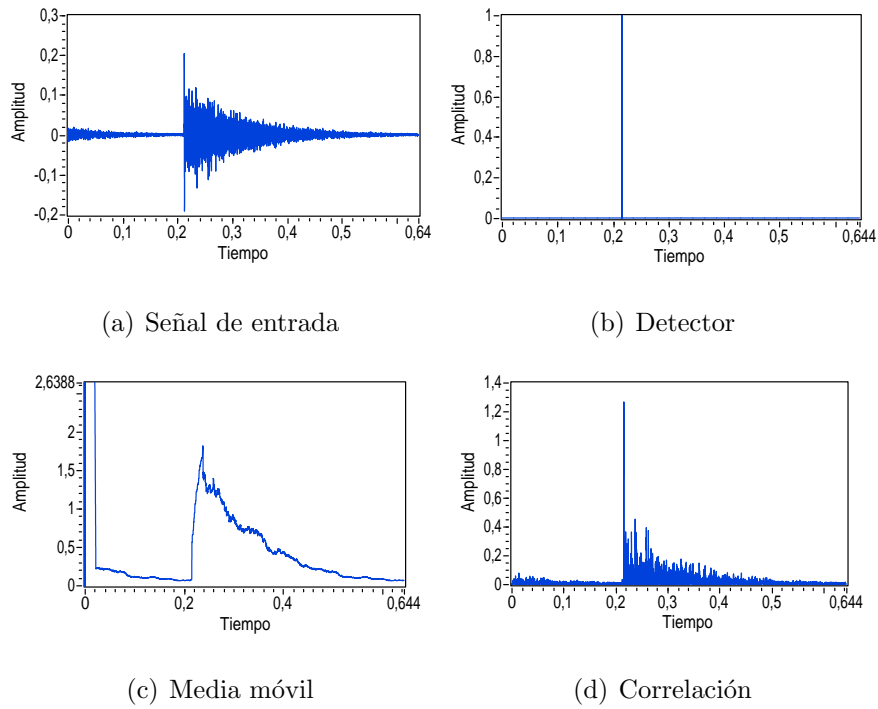


Figura 6.8: Detección de disparo utilizando con la correlación provista por la biblioteca de procesamiento de señales de LabVIEW.

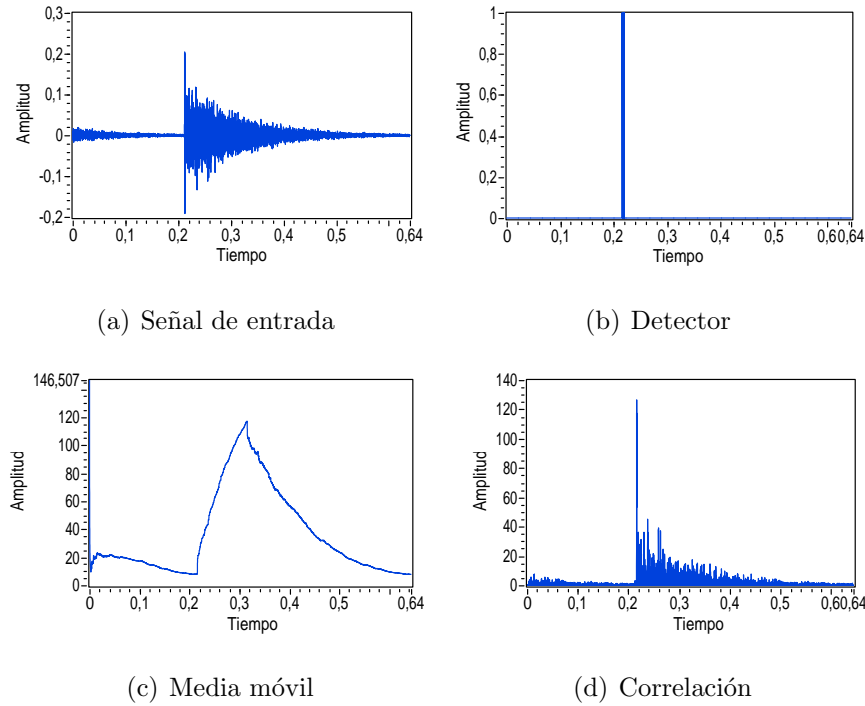


Figura 6.9: Detección de disparo usando el modelo de correlación bitstream implementado en LabVIEW por [5]. Las diferencias con la Figura 6.8 en parte se deben a la resolución en coma fija de esta último modelo.

Después de realizar varias pruebas comparando ambos modelos, se verifica que las diferencias no afectan la respuesta de módulo del detector porque la característica principal que se requiere para el debido funcionamiento de éste módulo es la forma de onda de la media móvil al producirse un valor alto de correlación. Se verá luego en el análisis final que la funcionalidad del sistema es más que satisfactoria.

La Tabla 6.3 muestra los recursos utilizados, extraídos del reporte del compilador de LabVIEW FPGA. Se observa un excesivo uso de recursos, en parte debido al uso de arreglos y por el método tipo de ejecución de flujo de cadena de datos característico de LabVIEW. Para forzar este tipo de flujo de código se utiliza lógica extra a la visualizada en el código gráfico, tal como se explicó en la sección 4.6.4. A pesar de la existencia de métodos para optimizar y eliminar ésta lógica para reducir recursos, como el uso de estructuras SCTL no fue posible utilizarlas porque la función de *Quotient and Remainder* para sustituir a la división en FPGA no admite el uso de STCLs.

Tabla 6.3: Recursos utilizados en el SoC del NI myRIO por el VI de FPGA del detector

Utilización de dispositivo	Utilizados	Totales	Porcentaje (%)
Slice Registros	18601	35200	52.8
Slice LUTs	15814	17600	89.9
Block RAMs	15	60	25.0
DSP48s	4	80	5.0

Una alternativa a utilizar la función de *Quotient and Remainder* es utilizar la función de desplazamientos a la izquierda o *Logical Shift* para dividir en potencias de dos. Esta función si es permitida con las estructuras SCTLs pero como se observa en la Tabla 6.4, el ahorro de recursos no es sustancial. Esta prueba compara únicamente un VI de FPGA con ambas funciones de forma independiente.

Tabla 6.4: Comparación de recursos utilizados por la función de *Quotient and Remainder* y *Logical Shift*

Utilización de dispositivo	<i>Quotient and Remainder</i> (%)	<i>Logical Shift</i> en SCTL (%)	Diferencia (%)
Slice Registros	24.1	21.7	2.4
Slice LUTs	47.3	42.4	4.9
Block RAMs	25.0	25.0	0
DSP48s	0	0	0

6.3 Verificación contra señales reales

En esta sección además de analizar los resultados obtenidos del módulo de detección, se explican además los aspectos más relevantes del proceso empleado para la obtención de resultados que se relacionan con el funcionamiento de dicho módulo.

Para realizar estas pruebas se excitó el sistema con grabaciones de sonidos reales de disparos, motosierras y sonidos ambientales de un bosque tropical, grabadas a distancias de 30m, 90m y 250m del origen del evento con ángulos de 0°, 90° y 180° del micrófono.

Cada archivo de sonido originalmente contenía solo información ya sea del momento que se origina un disparo o de la motosierra en acción. Excitar el módulo de detección en “frío”, o sea, cuando el sistema apenas está empezando a operar, tiende a dar falsos positivos en prácticamente cualquier caso, puesto que el umbral adaptativo con el que inicia el sistema por defecto es de 0. Cualquier valor de correlación positivo sería mayor que éste umbral generando de esta manera un falso positivo. Para disminuir la sensibilidad a este problema se optó por pre-cargar los registros del cálculo de umbral adaptativo con un valor alto para evitar una falsa alarma apenas entra a funcionar el sistema. Además es falso asumir que hay total silencio en un bosque y luego se genera inmediatamente un disparo o se cortan árboles, porque al menos existe una interacción del ambiente con el micrófono antes de. Por ello a los sonidos originales se les agregó al inicio fragmentos de sonidos de viento, lluvia o aves para simular las condiciones normales de un bosque y que de ésta manera se pueda realizar el cálculo del umbral adaptativo con el estudio adecuado del sonido ambiental previo a la generación de un evento de interés a detectar.

Para solucionar lo anterior y hacer las pruebas correctamente se creó un VI para exportar

los sonidos a archivos de texto en donde se facilita agregar al inicio información de otros sonidos y otro VI en donde se puede convertir un archivo de texto a una señal, la Figura 6.10 muestra los VIs creados.

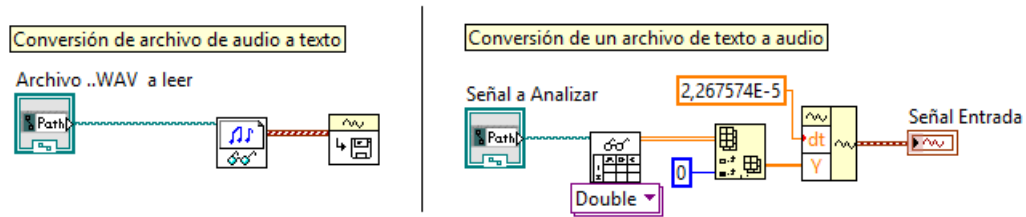


Figura 6.10: VIs de conversión de un archivo de audio a texto y viceversa.

6.3.1 Pruebas de detección de disparos

Las Figuras de 6.11 a 6.19 representan pruebas de detección obtenidas para disparos de armas de fuego de calibres diferentes y a distancias distintas. Se utilizó para estas pruebas una constante de ganancia K de 25 con una ventana que calcula el valor de umbral adaptativo con los últimos 100ms de señal.

En estas figuras se muestra la forma de onda del respectivo disparo a la izquierda con la salida del detector a la derecha, en donde en el instante que se genera el impulso en la señal del disparo se activa el detector. Es importante resaltar que se utilizó una plantilla para detectar ambos tipos de eventos (disparos o motosierras) creada a partir de las dos plantillas mencionadas en la sección 5.3 para realizar estas pruebas.

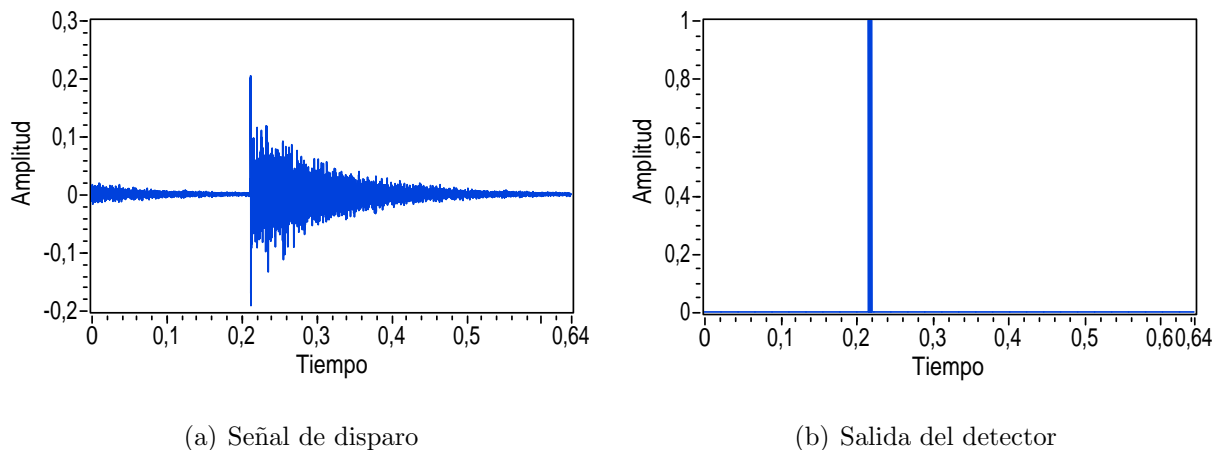


Figura 6.11: Detección de disparo a 30m de una carabina calibre 22.

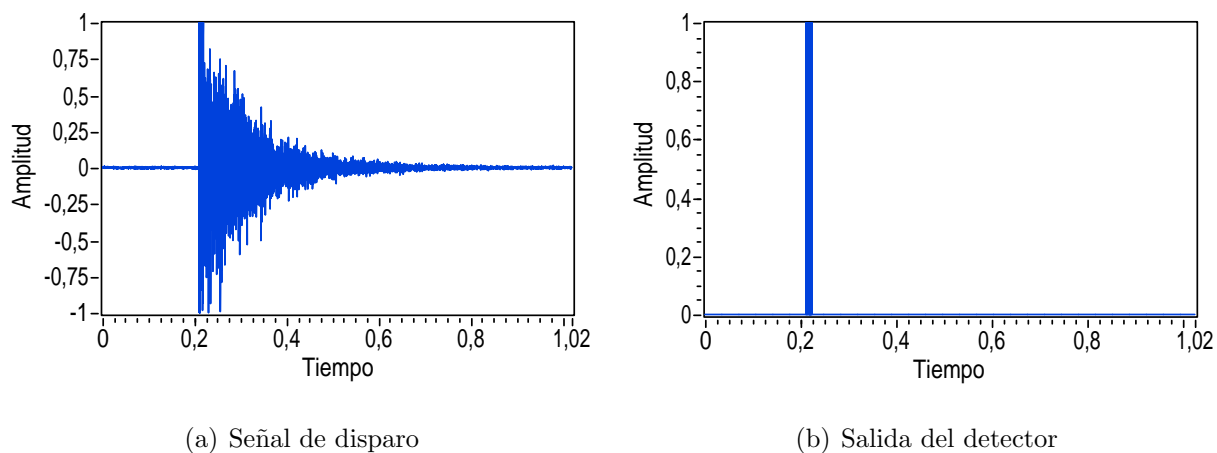


Figura 6.12: Detección de disparo a 30m de una pistola 9mm.

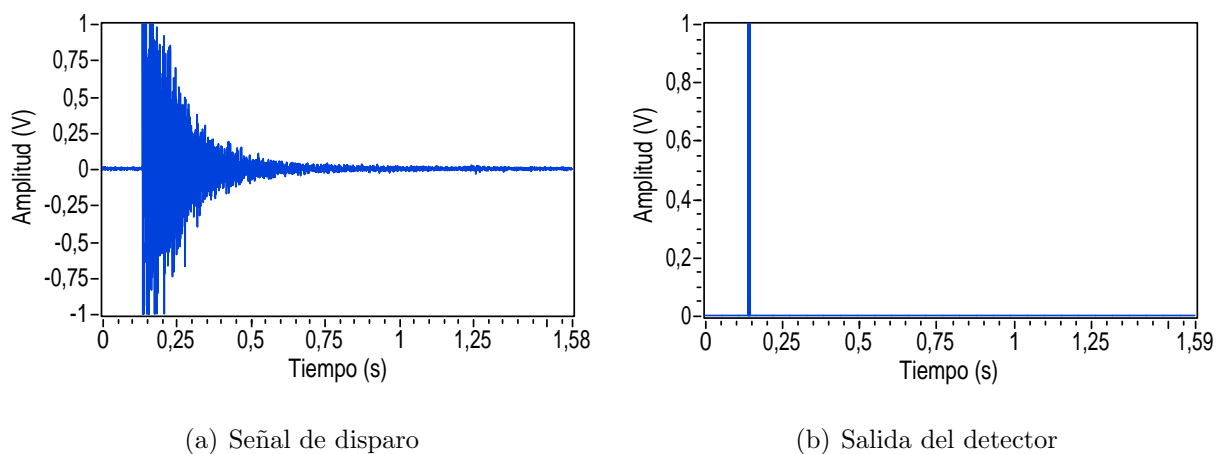


Figura 6.13: Detección de disparo a 30m de una escopeta calibre 12mm.

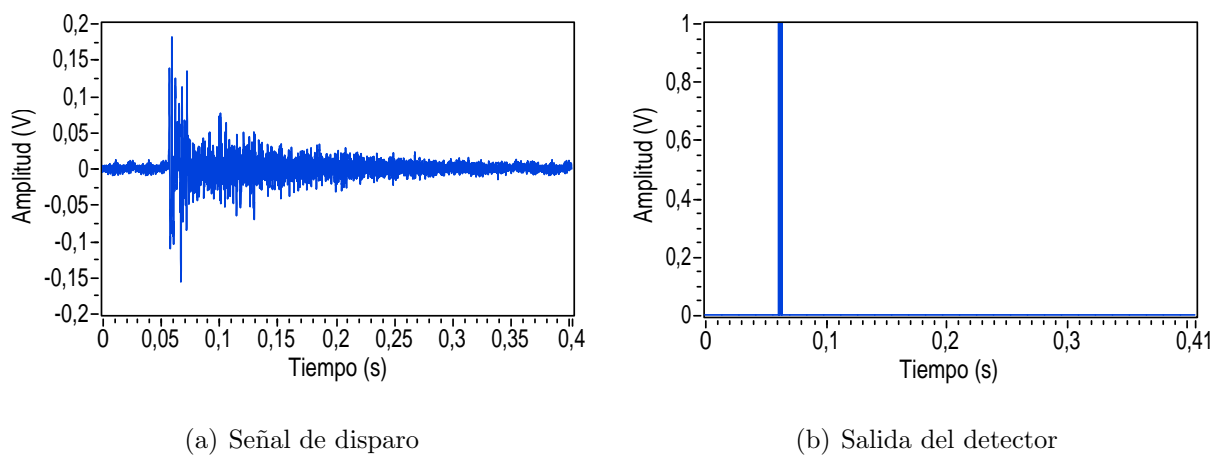
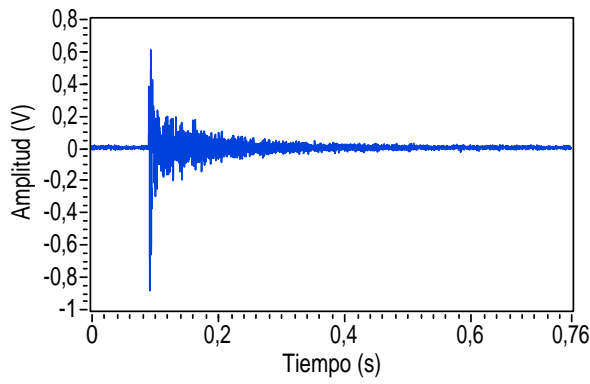
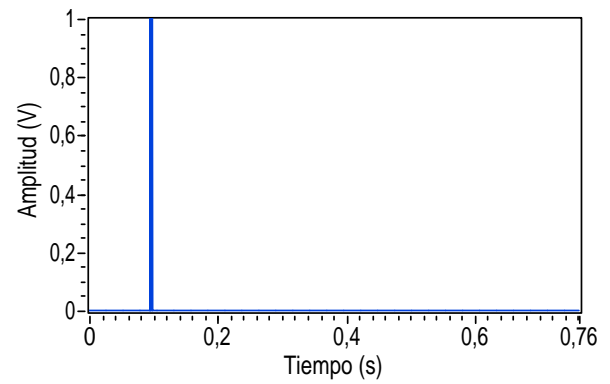


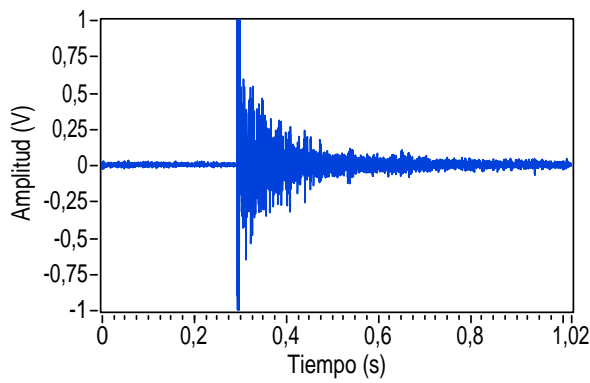
Figura 6.14: Detección de disparo a 90m de una carabina calibre 22.



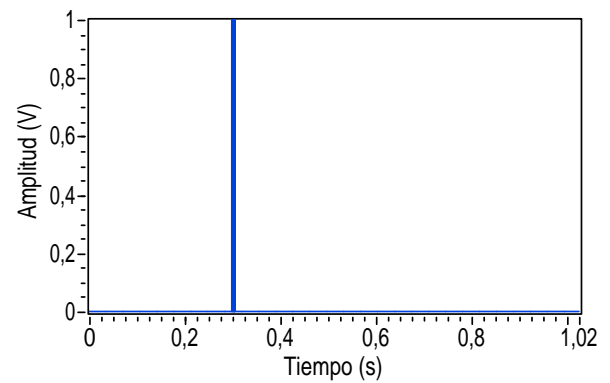
(a) Señal de disparo



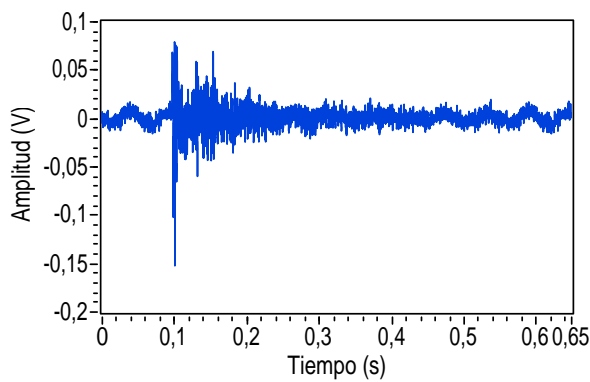
(b) Salida del detector

Figura 6.15: Detección de disparo a 90m de un revólver calibre 38.

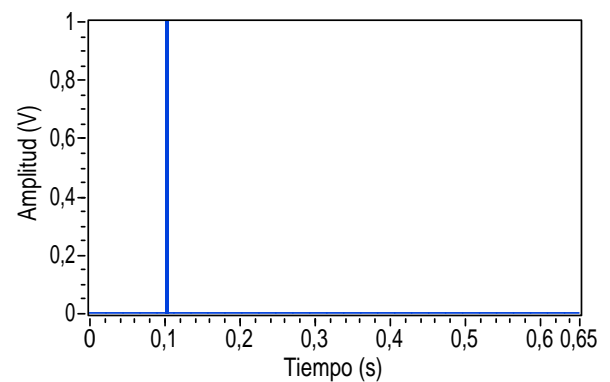
(a) Señal de disparo



(b) Salida del detector

Figura 6.16: Detección de disparo a 90m de una pistola 9mm.

(a) Señal de disparo



(b) Salida del detector

Figura 6.17: Detección de disparo a 250m de una revólver calibre 38.

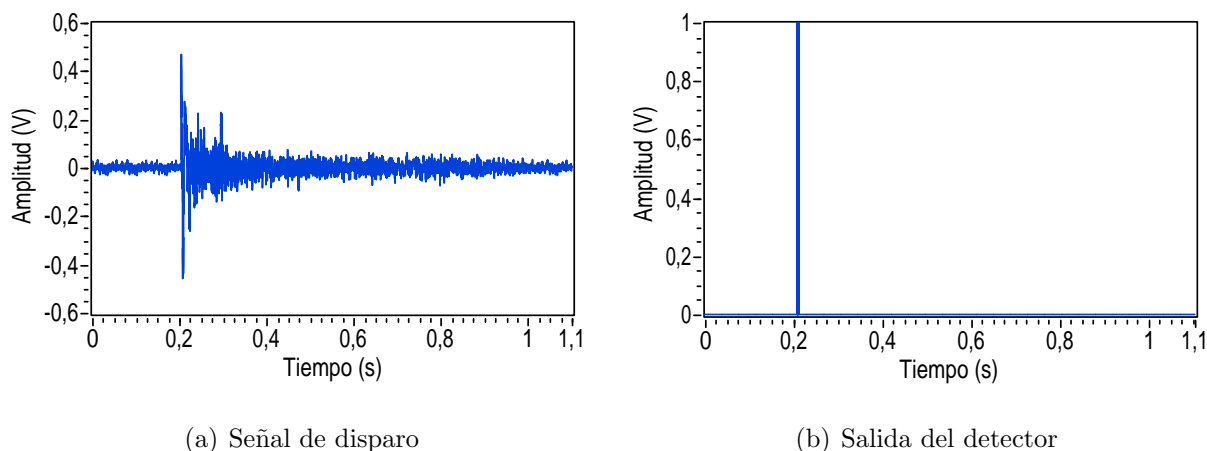


Figura 6.18: Detección de disparo a 250m de una escopeta calibre 12.

La Figura 6.19 muestra el comportamiento de la correlación del patrón con la señal de un disparo a 30m de una carabina calibre 22 y el cálculo de la media móvil. Puede notarse cómo los bloques de correlación y media móvil interaccionan para producir la señal binaria justo en el momento en que la correlación es máxima. Finalmente en la Figura 6.20 se muestra la detección de 4 disparos continuos.

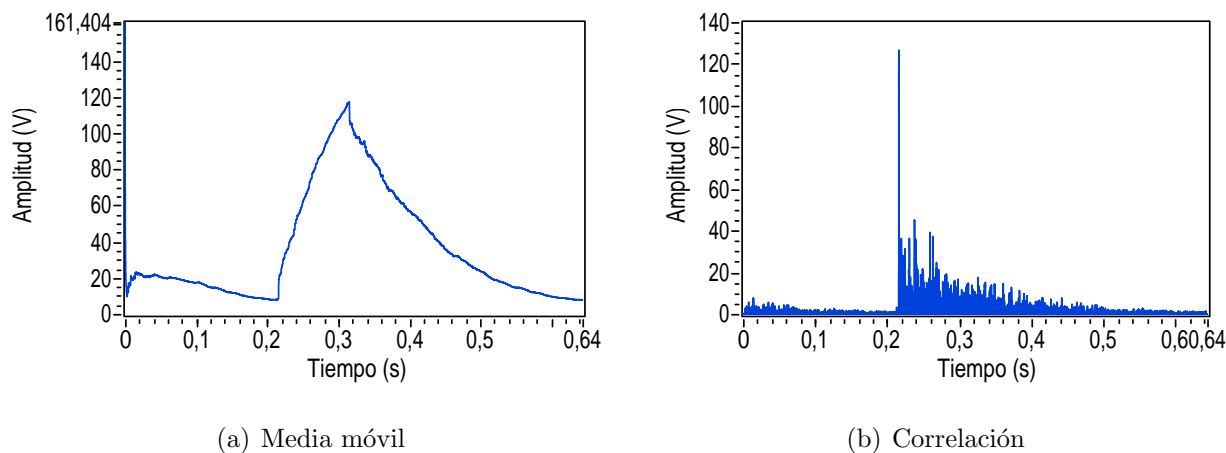


Figura 6.19: Media móvil y correlación para un disparo a 30m de una carabina calibre 22.

Se omite mostrar más resultados similares a los de la Figura 6.19 puesto que las otras pruebas de disparos exhiben el mismo comportamiento.

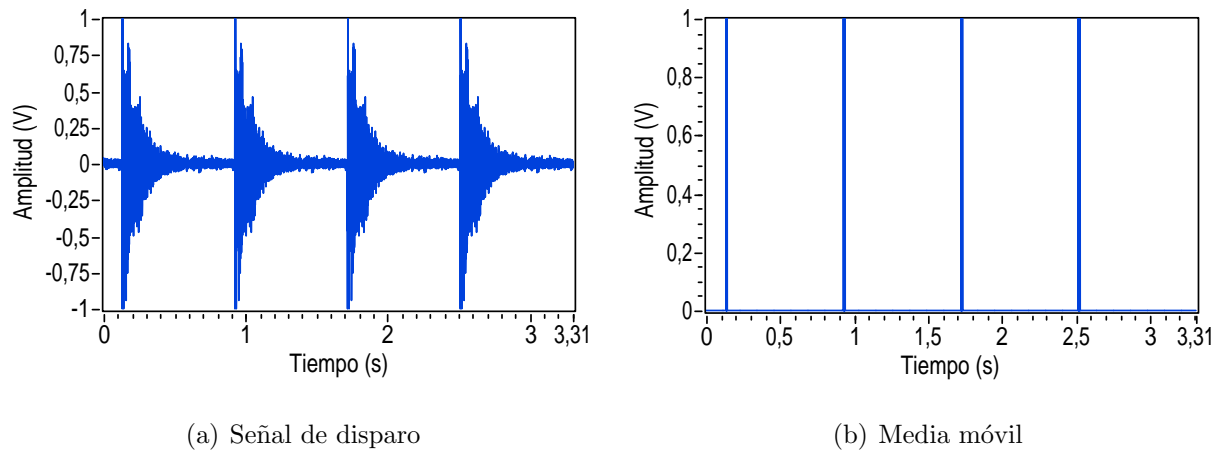


Figura 6.20: Detección de 4 disparos seguidos.

6.3.2 Pruebas de detección de motosierras

Al igual que en la sección 6.3.1, ésta muestra los resultados de detección de motosierras. Se tiene el mismo escenario en donde los parámetros del detector K y la ventana de para el cálculo de la media de la señal son los mismos.

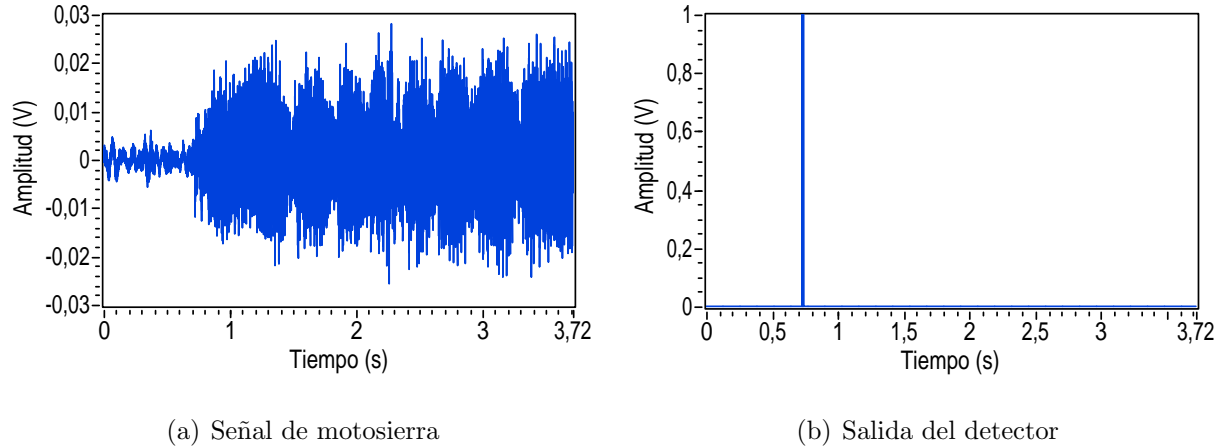


Figura 6.21: Detección de una motosierra a 30m.

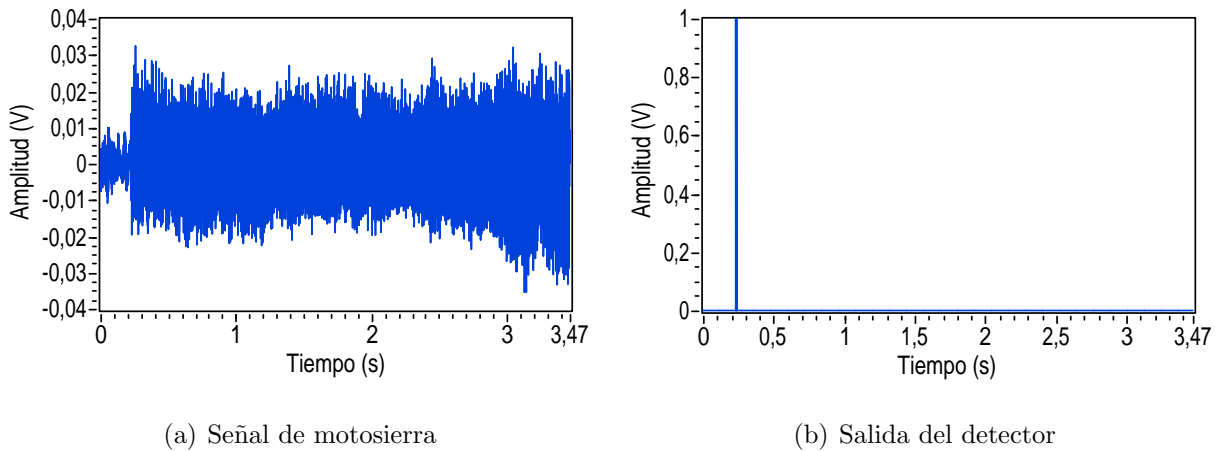
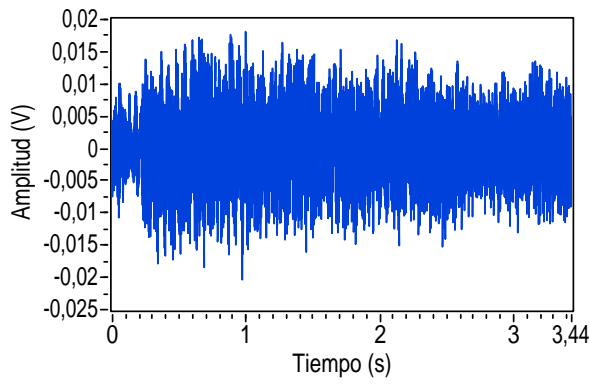


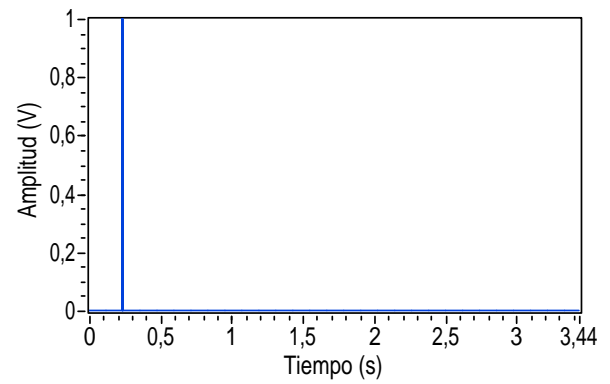
Figura 6.22: Detección de una motosierra a 90m.

Las Figuras 6.21 y 6.22 muestran la detección de dos motosierras a distancias diferentes. La detección se da únicamente al inicio del arranque de la motosierra y no a lo largo de todo el sonido a como es intuitivo pensarlo, porque el patrón es relativamente constante durante todo el sonido de la motosierra. Ello causa que el valor medio de la señal cambie de manera abrupta únicamente al inicio y no durante el resto de la señal. Pero debe recordarse que este esquema de detección es usado precisamente para detectar cambios abruptos o anormales en una variable. Es precisamente por ello que son eficientes si son ajustados correctamente para discriminar entre eventos típicos (sonidos no muy fuertes) y aquellos que no serían naturales en el ambiente.

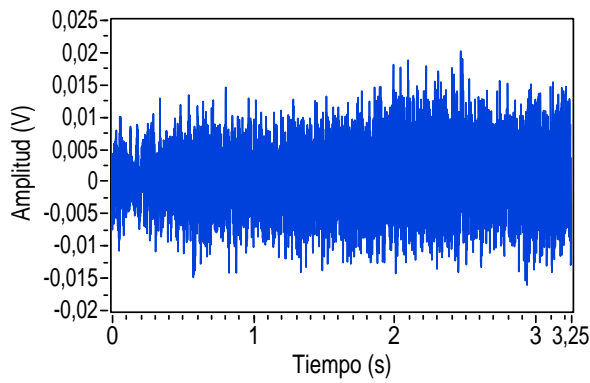
En las siguientes Figuras 6.23 y 6.24 se puede apreciar detecciones correctas para motosierras a 90m de distancia. En este caso en particular, debido a la distancia, podría pensarse sin embargo que la motosierra podría confundirse con el sonido ambiente. Aquí es donde el pre-procesamiento de la correlación realza las diferencias entre la motosierra y otros eventos, dándole más discriminación al detector. Nótese al respecto la Figura 6.27.



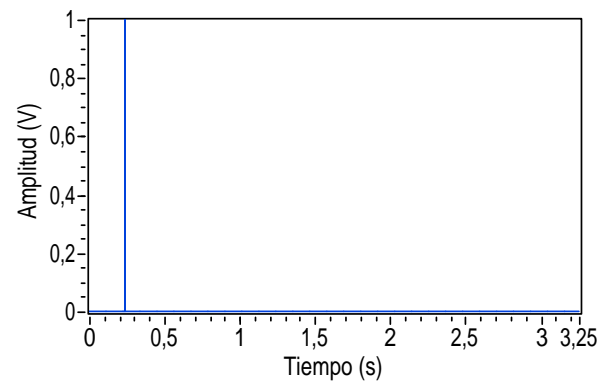
(a) Señal de motosierra



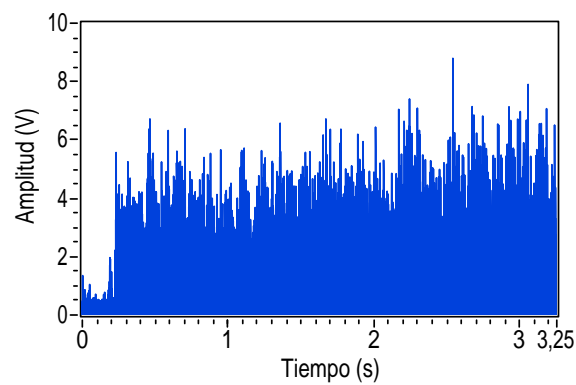
(b) Salida del detector

Figura 6.23: Detección de una motosierra a 90m.

(a) Señal de motosierra



(b) Salida del detector



(c) Correlación

Figura 6.24: Detección de motosierra a 90m. Nótese como al encenderse la motosierra, sube fuertemente el valor en la correlación.

6.3.3 Otras pruebas

Finalmente las Figuras 6.25, 6.26 y 6.27 muestran casos en donde se obtienen verdaderos negativos cuando se estimula el sistema con sonidos que no corresponden a disparos y/o motosierras.

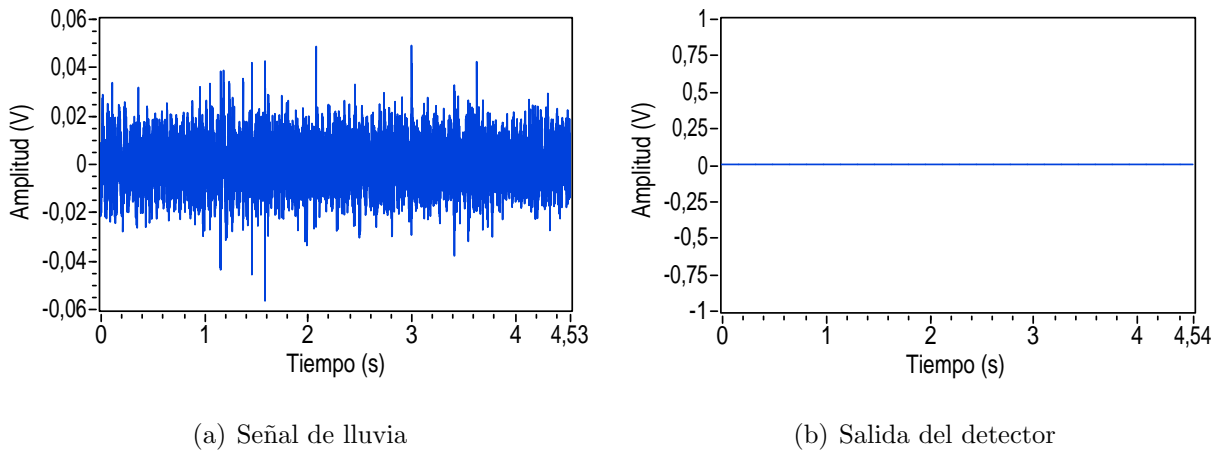


Figura 6.25: Detección de lluvia en un bosque tropical.

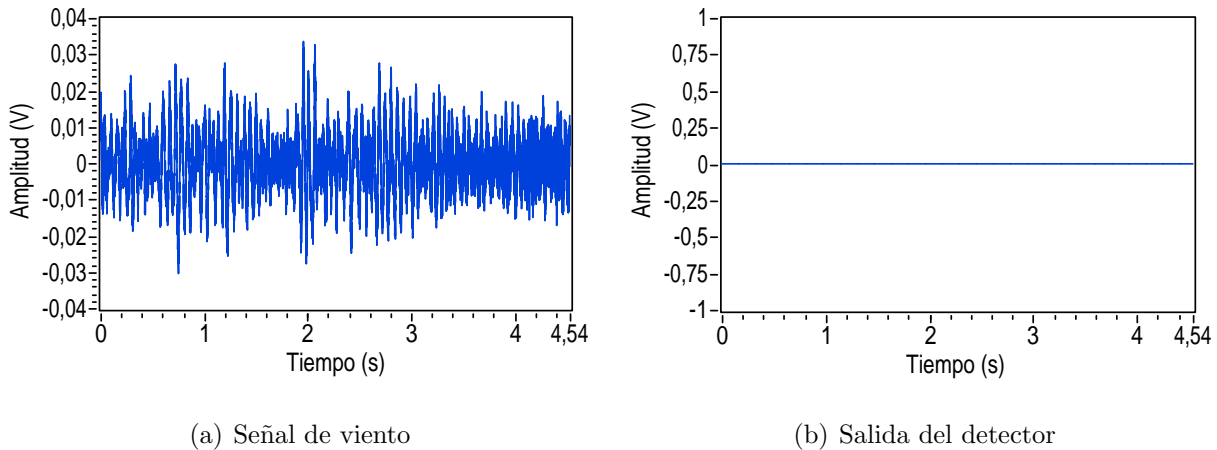


Figura 6.26: Detección de viento en un bosque tropical.

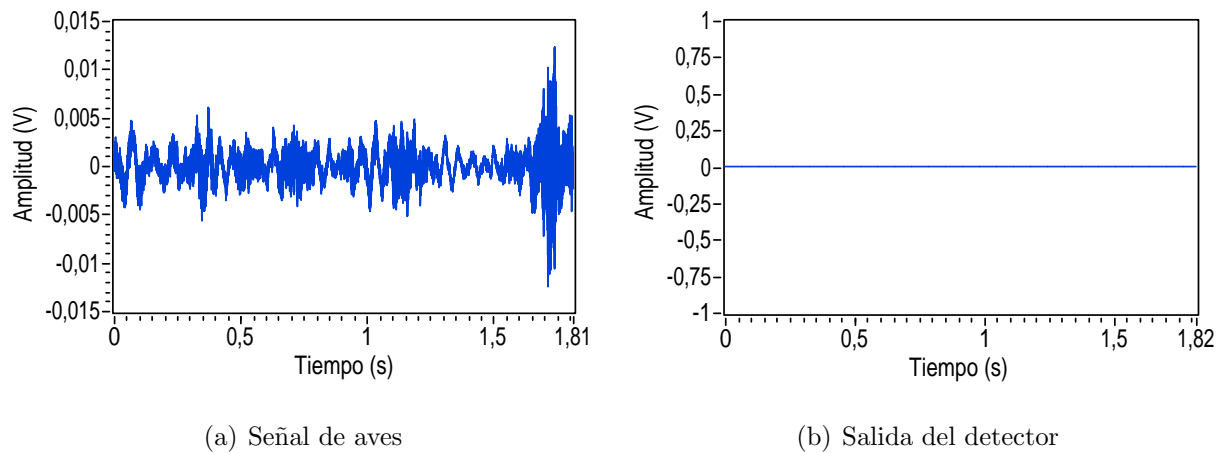


Figura 6.27: Detección de aves cantando en un bosque tropical.

En la figura 6.28 se muestra un estímulo continuo de aves cantando, un disparo a 30m, viento y una motosierra a 30m y se observa que el detector se activó en el momento adecuado en cada caso.

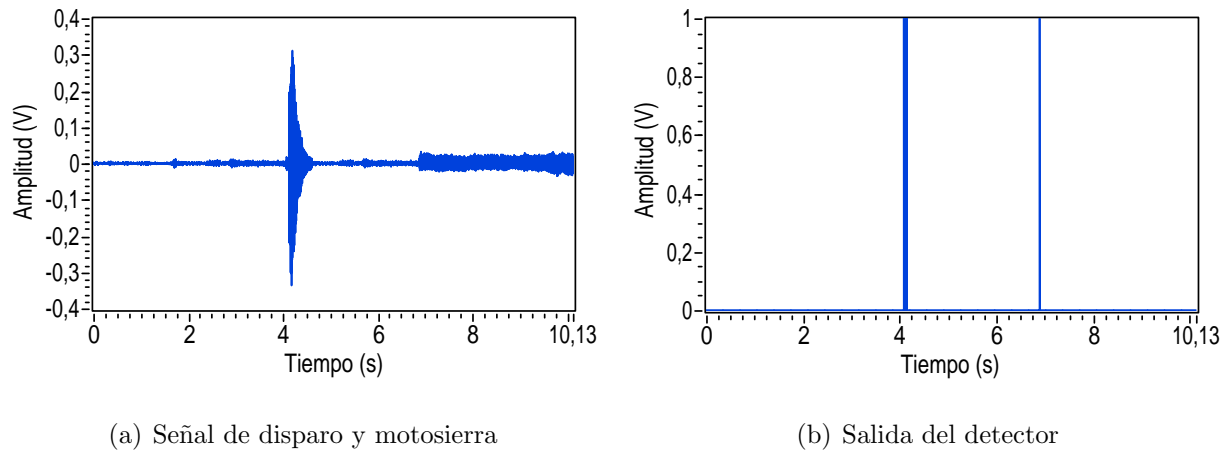


Figura 6.28: Detección un disparo y motosierra.

6.3.4 Curvas ROC

En las Figuras 6.29 y 6.30 se muestra las curvas ROC para la detección de disparos y motosierras respectivamente. Se utilizaron 83 señales de audio en total, con 33 de disparos, 29 de motosierras, y 21 de sonido ambiental típico. Para mejorar los verdaderos negativos, se usaron a las motosierras como muestras negativas para el proceso de detección de disparos, y viceversa.

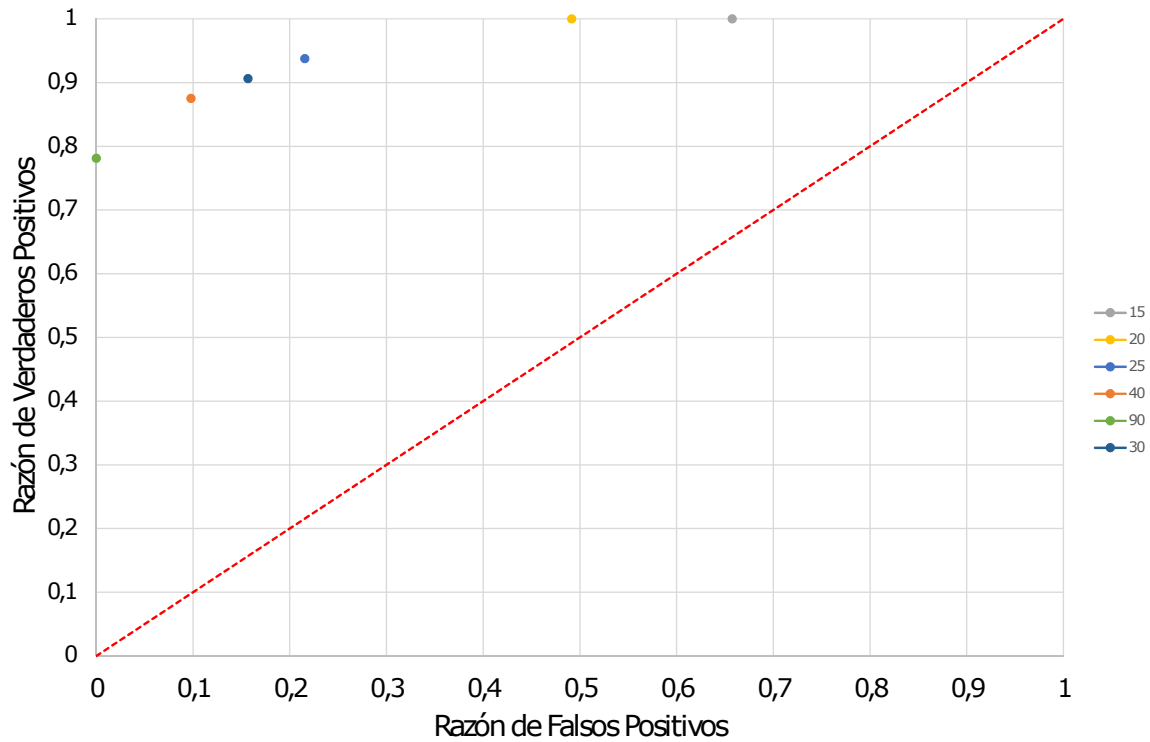


Figura 6.29: Curva ROC para la detección de disparos.

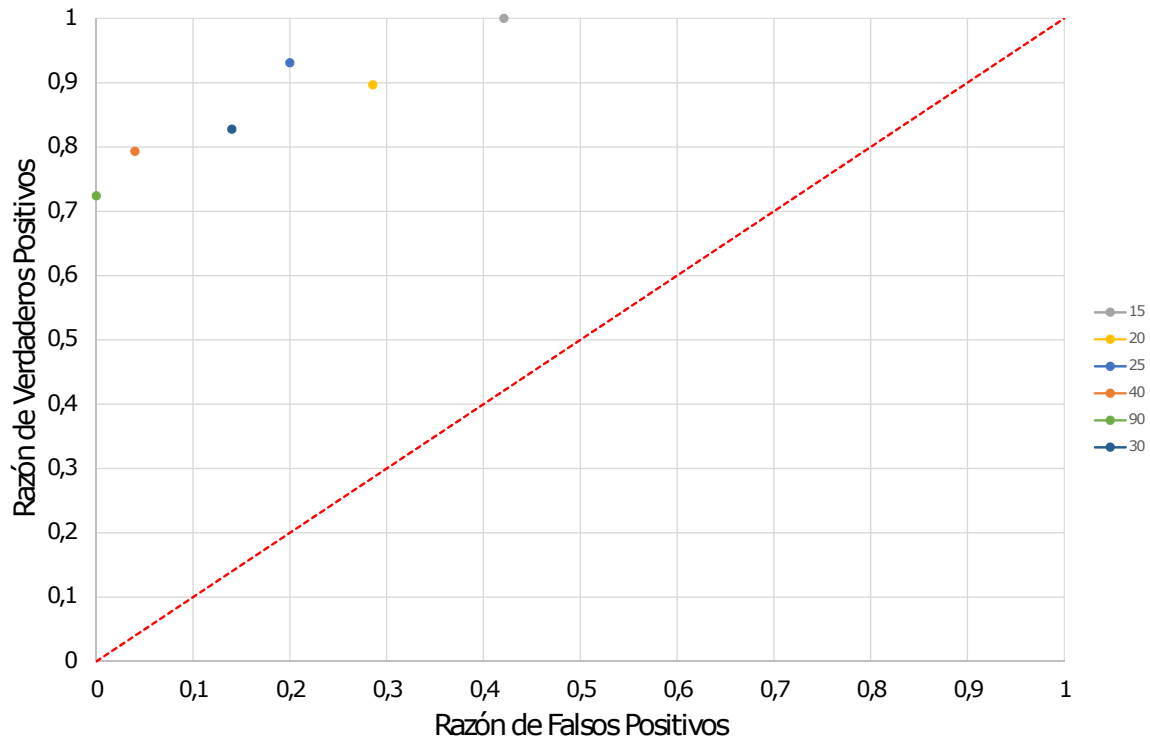


Figura 6.30: Curva ROC para la detección de motosierras.

En las Tablas 6.5 y 6.6 se muestran los pares ordenados (RFP,RVP) las curvas ROC para la detección de disparos y motosierras respectivamente.

Tabla 6.5: ROC para la detección de disparos.

Constante K	RFP	RVP
15	0.65	1
20	0.49	1
25	0.21	0.93
30	0.15	0.90
40	0.09	0.87
90	0	0.78

Tabla 6.6: ROC para la detección de motosierras.

Constante K	RFP	RVP
15	0.42	1
20	0.28	0.89
25	0.20	0.93
30	0.14	0.82
40	0.04	0.79
90	0	0.72

El efecto de la K en el rendimiento del detector es muy importante debido a que ajusta la sensibilidad. Una constante muy baja hace que el detector por un lado detecte la mayoría o la totalidad de eventos verdaderos, pero aumentando demasiado la detección de falsos positivos. Un valor muy alto de K vuelve al detector demasiado selectivo, más bien. El valor de K óptimo dependerá de dónde deseamos usar el detector. Si tenemos una etapa de clasificación posterior al detector (como es en el caso del sistema SiRPA), es de mayor interés no poseer muchos falsos negativos, pues el clasificador eliminará los falsos negativos excesivos. Ello implica un umbral más bajo. Para el caso de los disparos, se calculó un valor de K de 25 determinado experimentalmente después de un largo proceso de estimular el detector con los sonidos de la base de datos para valores de K diferentes. De igual manera para las motosierras una K de 25 o 30 se obtienen los mejores resultados de detección.

La mayor cantidad de falsos negativos obtenidos en las pruebas se dan en las muestras de sonidos a 180° y a largas distancias. El motivo principal es porque el micrófono utilizado es

direccional y a 180° se tiene menor sensibilidad. Ello sugiere que un método de mejorar la detección es utilizando una red de micrófonos.

Capítulo 7

Conclusiones y recomendaciones

Conclusiones

- LabVIEW resultó una herramienta muy útil para modelar sistemas complejos e implementarlos rápidamente en FPGAs de NI para realizar pruebas de concepto.
- Se ha comprobado empíricamente que el uso de un sistema de detección de señal basado en umbral adaptativo es altamente eficiente para detectar patrones de señales de disparos y motosierras en un ambiente boscoso. En particular, las pruebas han verificado que se alcanzan tasas de reconocimiento de hasta un 93% para una ganancia de 25.
- Se ha comprobado empíricamente que la correlación cruzada en *bistream* es una manera eficiente de construir un detector de patrones acústicos. Esta resalta de forma impulsiva los patrones de interés en la señal de entrada.
- Se puede mejorar el SNR aumentando el orden del modulador sigma-delta o el factor de sobremuestreo.
- Se ha comprobado empíricamente que el detector de umbral adaptativo excluye eficientemente el ruido ambiental y detecta los eventos de interés. Sin embargo, aumentar K mejora la razón de falsos positivos pero empeora la razón de verdaderos positivos y si se disminuye ocurre lo opuesto.
- Es necesario que el sistema alcance un estado de reposo o estacionario antes de volverse

un detector eficiente. Para simular este efecto en las pruebas en laboratorio, conviene entonces excitar al sistema con algún nivel de ruido o sonido ambiente antes de iniciar con las pruebas reales en sí.

- La construcción de un gráfico ROC es un proceso extenso que se volverá estadísticamente determinante solo si se le somete a una gran cantidad de pruebas para un valor de umbral específico.
- Con el uso de una red amplia de micrófonos se podría mejorar el rendimiento del detector, puesto que con muestras muy lejanas y a ángulos de 180° se obtuvo mayor cantidad de falsos negativos.

Recomendaciones

- Utilizar desplazamientos a la derecha para dividir en potencias de dos en el filtro de media móvil.
- Verificar el rendimiento del detector utilizando plantillas creadas con una mayor cantidad de muestras y más periodo de la señal.
- Estudiar si mejora la detección si se utilizar algún método de rectificación completa o de cuadrado (energía) a la salida del bloque de correlación.
- Estudiar la potencial mejora en detección utilizando un control de ganancia automático (AGC) previo a la etapa del modulador sigma-delta.

Bibliografía

- [1] P. Julián y P. Myolesi A. Chacón-Rodríguez, F. Martin-Pirchio. A verilog hdl digital architecture for delay calculation. *Latin American Applied Research Review.*, 37(1):41–46, Enero 2007.
- [2] E. Bachinelo. *Crónicas de la vida columnas, ensayos y comentarios 2004-2013*. Xlibris Corporation, Estados Unidos, 2013. URL <https://books.google.co.cr/books?isbn=1503566900>.
- [3] U. Beis. An introduction to delta sigma converters. *U. Beis*, 2007. URL <http://www.beis.de/Elektronik/DeltaSigma/DeltaSigma.html>.
- [4] D. Castro. Costa rica lucha contra las byas de tala ilegal. *Centro de Investigación de Crimen Organizado*, 2013. URL <http://es.insightcrime.org/noticias-del-dia/costa-rica-lucha-contras-las-byas-de-tala-ilegal>.
- [5] J. Mora Castro. Implementación de algoritmo para calcular la correlación entre dos señales para detectar disparos en el bosque. Tesis, Insituto Tecnológico de Costa Rica, 2016.
- [6] Vicerectoría de Investigación y Extensión Instituto Tecnológico de Costa Rica. Cartera de proyectos del tec [online]. 2013 [visitado el 19 de junio de 2016]. URL <http://www.tec.ac.cr/sitios/Vicerrectoria/vie/Paginas/proyectos%20activos.aspx>.
- [7] D. Anastasio de la Mata Moya. *DISEÑO DE DETECTORES ROBUSTOS EN APLICACIONES RADAR*. PhD thesis, Universidad de Alcalá, 2012. URL <http://dspace.uah.es/dspace/bitstream/hyle/10017/17143/Dise%C3%B1o%20de%20Detectores%20Robustos%20en%20Aplicaciones%20Radar.pdf?sequence=1>.

- [8] Analog Devices. Sigma-delta adc tutorial [online]. 2016 [visitado el 19 de junio de 2016]. URL <http://www.analog.com/en/design-center/interactive-design-tools/sigma-delta-adc-tutorial.html#relatedInformation>.
- [9] F. Gorunescu. *Data Mining Concepts, Models y Techniques*, volume 12. Springer, 2011. URL <https://books.google.co.cr/books?isbn=3642197213>.
- [10] R. R. Harrison. A low-power integrated circuit for adaptive detection of action potentials in noisy signals. In *Engineering in Medicine y Biology Society, 2003. Proceedings of the 25th Annual International Conference of the IEEE*, volume 4, pages 3325–3328 Vol.4, Sept 2003.
- [11] Thomas Haslwanter. Introduction to biosignal analysys [online]. 2014 [visitado el 19 de junio de 2016]. URL <http://work.thaslwanter.at/BSA/html/Features.html>.
- [12] D. Heeger. Signal detection theory. *Center for Neural Science*, Noviembre 1997. URL <http://www.cns.nyu.edu/~david/hyouts/sdt-advanced.pdf>.
- [13] National Instruments. How can i optimize/reduce fpga resource usage y/or increase speed? [online, visitado el 19 de junio de 2016]. URL <http://digital.ni.com/public.nsf/allkb/311C18E2D635FA338625714700664816>.
- [14] National Instruments. Dataflow y the enable chain in fpga vis (fpga module) [online]. 2016 [visitado el 19 de junio de 2016]. URL http://zone.ni.com/reference/en-XX/help/371599H-01/lvfpgaconcepts/fpga_sctl_y_enablechain/.
- [15] National Instruments. First order delta sigma vi [online]. 2016 [visitado el 19 de junio de 2016]. URL <https://decibel.ni.com/content/docs/DOC-13486>.
- [16] National Instruments. Getting started with labview fpga [online]. 2016 [visitado el 19 de junio de 2016]. URL <http://www.ni.com/tutorial/14532/en/>.
- [17] National Instruments. Labview 2015 fpga module help [online]. 2016 [visitado el 19 de junio de 2016]. URL <http://zone.ni.com/reference/en-XX/help/371599L-01/>.
- [18] National Instruments. Ni online training [online]. 2016 [visitado el 19 de junio de 2016]. URL www.ni.com/self-paced-training.

- [19] National Instruments. Single-cycle timed loop faq for the labview fpga module [online]. 2016 [visitado el 19 de junio de 2016]. URL <http://digital.ni.com/public.nsf/allkb/722A9451AE4E23A586257212007DC5FD>.
- [20] National Instruments. Software de desarrollo de sistemas ni labview [online]. 2016 [visitado el 19 de junio de 2016]. URL <http://www.ni.com/labview/esa/>.
- [21] National Instruments. Software de desarrollo de sistemas ni labview [online]. 2016 [visitado el 19 de junio de 2016]. URL <http://www.ni.com/getting-started/labview-basics/esa/environment>.
- [22] National Instruments. Using a host vi to communicate with the fpga target (fpga module) [online]. 2016 [visitado el 19 de junio de 2016]. URL http://zone.ni.com/reference/en-XX/help/371599H-01/lvfpgaconcepts/pfi_comm/.
- [23] National Instruments. Why shouldn't i use large arrays when developing fpga applications? [online]. 2016 [visitado el 19 de junio de 2016]. URL <http://digital.ni.com/public.nsf/allkb/80C9808E454DCBC286257093005B9EB6>.
- [24] S. Louis L. *Statistical Signal Processing - Detection, Estimation, y Time Series Analysis*. Addison-Wesley, 1991.
- [25] M. De Leganés. Costa rica prohíbe la caza. *Ecologistas en Acción*, Diciembre 2012. URL <http://www.ecologistasenaccion.org/article24711.html> [visitado el 19 de junio de 2016].
- [26] R. C. Maher. Summary of gun shot acoustics. *Universidad del Estado de Montana*, 2006. URL http://www.montana.edu/rmaher/publications/maher_aac_0406.pdf.
- [27] MedCalc. Roc curve analysis in medcalc [online]. 2016 [visitado el 19 de junio de 2016]. URL <https://www.medcalc.org/manual/roc-curves.php>.
- [28] H. Morales. Aumenta tala ilegal de árboles en guanacaste. *La voz de Guanacaste*, 2014. URL <http://www.vozdeguanacaste.com/es/articulos/2014/08/18/aumenta-tala-ilegal-de-arboles-en-guanacaste>.
- [29] J. Resis. How sigma delta modulators achieve high performance. *University of London*, 2007.

- [30] Go Visit Costa Rica. Ecoturismo en costa rica. *Ecoturismo*, 2016. URL <http://www.govisitcostarica.co.cr/travelInfo/ecotourism.asp>.
- [31] INBio Costa Rica. Conservación in situ: áreas silvestres protegidas. *Los esfuerzos de conservación de la biodiversidad*, 2015. URL http://www.inbio.ac.cr/es/biod/estrategia/Paginas/esfuerzos_conservar01.html.
- [32] A. Chacón Rodríguez. *CIRCUITOS INTEGRADOS DE BAJO CONSUMO PARA DETECCIÓN Y LOCALIZACIÓN DE DISPAROS DE ARMAS DE FUEGO*. PhD thesis, Universidad Nacional de Mar del Plata, 2009.
- [33] Ph. D. Sangil Park. *Principles of Sigma-Delta Modulation for Analog-to-Digital Converters*. Motorola, 2000.
- [34] M. Soto. Autoridades intentan frenar tala y extracción de animales. *La Nación*, 2015. URL http://www.nacion.com/vivir/ambiente/Operativos-frenaran-tala-extraccion-animales_0_1531646833.html.
- [35] M. Soto. Tala ilegal desluce esfuerzos de reforestación de costa rica. *La*, 2015. URL http://www.nacion.com/vivir/ambiente/Tala-ilegal-desluce-esfuerzos-reforestacion_0_1494050591.html.
- [36] John A. Swets. *Signal Detection Theory y ROC Analysis in Psychology y Diagnostics*. Psychology Press, 1996. URL <https://books.google.co.cr/books?isbn=1317779711>.
- [37] L. Petrică V. yrei, H. Cucu. Considerations on developing a chain-saw intrusion detection y localization system for preventing unauthorized logging. *Journal of Electrical y Electronic Engineering*, 3:202–207, Diciembre 2015. URL <http://article.sciencepublishinggroup.com/html/10.11648.j.jeee.20150306.15.html#paper-content-5>.
- [38] Emilio Soria Olivas Marcelino Martines Sober José Vicente Francés Villora Gustavo Camps Valls. *Tratamiento Digital de Señales - Problemas y Ecoturismo ejercicios resultados*. 84-205-3559-1. Pearson Prentice Hall, 2003. URL <https://books.google.co.cr/books?isbn=8420535591>.

- [39] Inc. Wolfram Research. Cross-correlation [online, visitado el 19 de junio de 2016]. URL <http://mathworld.wolfram.com/Cross-Correlation.html>.
- [40] R. Cerdas-Robles y A. Rodríguez y A. Chacon-Rodríguez y P. Julián. Design of an idm-based determinant computing unit for a 130nm low power cmos asic acoustic localization processor. In *Circuits Systems (LASCAS), 2015 IEEE 6th Latin American Symposium on*, pages 1–4, Feb 2015.
- [41] A. Chacon-Rodriguez y F. Martin-Pirchio y S. Sanudo y P. Julian. A low-power integrated circuit for interaural time delay estimation without delay lines. *IEEE Transactions on Circuits y Systems II: Express Briefs*, 56(7):575–579, July 2009.
- [42] R. Pereira-Arroyo y F. Nicaragua-Guzmán y A. Chacón-Rodríguez. Design of an operational transconductance amplifier applying multiobjective optimization. In *Argentine School of Micro-Nanoelectronics Technology y Applications (EAMTA), 2010*, pages 12–17, Oct 2010.
- [43] O. E. Liseth y H. A. Hjortly y T. S. "Bassen" Lye. Power efficient cross-correlation beat detection in electrocardiogram analysis using bitstreams. In *2009 IEEE Biomedical Circuits y Systems Conference*, pages 237–240, Nov 2009.
- [44] C. Salazar-García y L. Alfaro-Hidalgo y M. Carvajal-Delgado y J. Montero-Aragón y R. Castro-Gonzalez y J. A. Rodríguez y A. Chacon-Rodríguez y P. Alvarado-Moya. Digital integrated circuit implementation of an identification stage for the detection of illegal hunting y logging. In *Circuits Systems (LASCAS), 2015 IEEE 6th Latin American Symposium on*, pages 1–4, Feb 2015.
- [45] B. Gosselin y M. Sawan. An ultra low-power cmos action potential detector. In *2008 IEEE International Symposium on Circuits y Systems*, pages 2733–2736, Mayo 2008.
- [46] C. Gómez Víquez L.Á. Li Huang O. Villalta Gutiérrez A. Chacón Rodríguez y P. Alvarado Moya. An embedded test system for acoustic pattern recognition intended for environmental monitoring y protection in tropical rain forest reserves. In *In Proceedings of the Third Embedded Technology Conference 2012*, pages 17–21, Febrero 2012.

- [47] A. Chacon-Rodriguez y P. Julian. Evaluation of gunshot detection algorithms. In *Micro-Nanoelectronics, Technology y Applications, 2008. EAMTA 2008. Argentine School of*, pages 49–54, Sept 2008.
- [48] A. Chacon-Rodriguez y P. Julian y F. Masson. Fast y low power integrated circuit for impulsive sound localisation using kalman filter approach. *Electronics Letters*, 46(7):533–534, April 2010.
- [49] A. Chacon-Rodriguez y P. Julian y L. Castro y P. Alvarado y N. Hernyez. Evaluation of gunshot detection algorithms. *IEEE Transactions on Circuits y Systems I: Regular Papers*, 58(2):363–373, Feb 2011.
- [50] F. N. M. Pirchio y P. Julian y P. S. Myolesi y A. Chacon-Rodriguez. An adaptive cross-correlation derivative algorithm for ultra-low power time delay measurement. In *2007 IEEE International Symposium on Circuits y Systems*, pages 4016–4019, May 2007.
- [51] A. Chacon-Rodriguez y S. Li y M. Stanaćević y L. Rivas y E. Baradin y P. Julian. Low power switched capacitor implementation of discrete haar wavelet transform. In *Circuits y Systems (LASCAS), 2012 IEEE Third Latin American Symposium on*, pages 1–4, Feb 2012.

Índice alfabético

ADC de Nyquist, 16
aliasing, 16
bitstream, 3, 15
convertidor de Nyquist, 16
correlación, 18
curva ROC, 24
curvas característica operativa del receptor,
24
diagonal de no discriminación, 25
error de conversión, 16
error de cuantización, 16
filtro pasa bajas, 15
Matriz de confusión, 26
media estadística, 22
onda de choque, 10
precisión de una curva ROC, 27
ruido cuantización, 16
sigma-delta, 14
sobremuestrear, 16
teoría de detección de señales, 24
teorema de muestro de Nyquist, 16
tren de pulsos, 3
umbral adaptativo, 20