



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

SISTEMA DE VIDEO VIGILANCIA POR DETECCIÓN
DE ACTIVIDAD ACÚSTICA

T E S I S

QUE PARA OBTENER EL TÍTULO DE:

INGENIERO ELÉCTRICO - ELECTRÓNICO

P R E S E N T A

YVES MAILLARD QUIROZ

TUTOR

M.I. LARRY HIPÓLITO ESCOBAR SALGUERO



CIUDAD UNIVERSITARIA, CDMX, 2020

Para mi madre, mi familia y mis maestros

Agradecimientos

Esta tesis no se pudiera haber llevado a cabo sin el apoyo incondicional de mi madre, mi hermana y toda mi familia. Gracias a su constante esfuerzo, aliento y motivación es que este trabajo se pudo realizar.

Agradezco a la UNAM y en especial a la Facultad de Ingeniería por la educación que me brindaron y por facilitarme lo necesario para poder realizar esta tesis y todos mis estudios a lo largo de estos años.

Expreso especial gratitud a mi asesor, el maestro Larry Hipólito Escobar, por transmitirme sus conocimientos, por su guía y consejo durante todas las etapas implicadas en la realización de este trabajo. Sin su orientación y ayuda esta tesis no hubiera sido posible.

Aprecio y agradezco a todos mis amigos por su soporte y la confianza que depositaron en mí a lo largo de estos años; por todos los momentos agradables que me brindaron y por su constante colaboración y motivación.

Finalmente agradezco a todas las personas que me apoyaron durante mi formación académica y que confiaron en mí.

Resumen

Los sistemas de vigilancia están presentes en una gran cantidad de lugares y se utilizan para el monitoreo constante de diversos lugares y fenómenos debido a que ayudan a detectar eventos no habituales o indeseados y permiten a los operadores tomar las medidas necesarias en caso que ocurran. Esta tesis presenta la implementación de un sistema de vigilancia utilizando un DSP. El sistema de vigilancia es capaz de detectar en tiempo real la presencia de estruendos o ruidos anormales mediante un algoritmo de detección de actividad acústica y permite controlar el movimiento de una cámara de vigilancia en base a la detección realizada. Además ofrece la visualización en tiempo real del video producido por la cámara a través de un monitor LCD. Los algoritmos fueron realizados en lenguaje C y cargados al DSP; en el trabajo se explica detalladamente el funcionamiento de cada uno de los algoritmos, se incluyen los diagramas de flujo correspondientes a ellos y se describe la forma en que se implementaron. También se presentan los resultados de un conjunto de pruebas realizadas para evaluar el desempeño de los algoritmos bajo diferentes condiciones.

Abstract

Surveillance systems are present in a large variety of places and are constantly being used to monitor different locations and phenomena because they help detecting unwanted or non-usual events and allow the users to take the necessary actions in case something happens. This thesis presents the development of a surveillance system using a DSP. The surveillance system is capable to detect the presence of loud or abnormal noises in real-time using an acoustic activity detection algorithm, also allows to control the movement of a surveillance camera based on the detection made. The systems also offers the display of real-time video produced by the surveillance camera through a LCD Monitor. The algorithms were written in C programming language and loaded to the DSP; the present work explains in detail the principle of operation of each algorithm and includes the flow charts associated to each algorithm as well as the way they were implemented. Moreover, the results obtained from a set of tests are presented, these results allow to evaluate the performance of the algorithms under different conditions.

Índice general

Agradecimientos	v
Resumen	vii
Abstract	ix
1. Introducción	1
1.1. Sistemas de vigilancia	1
1.2. Inconvenientes comunes en sistemas de vigilancia	4
1.3. Propuesta frente a la problemática presentada	4
1.4. Objetivo	4
1.5. Breve descripción de capítulos	4
2. Servomotores y PWM	7
2.1. Control y retroalimentación	7
2.2. Elementos de los servomecanismos	10
2.3. Servomotores	11
2.3.1. Motores de DC	11
2.4. Modulación por ancho de pulso (PWM)	13
2.4.1. Control de movimiento de motores utilizando PWM	17
2.5. Control de servomotores	22
2.6. Resumen	26
3. Detección de actividad acústica y de voz	27
3.1. Ondas y sonido	27
3.1.1. Ecuación de onda	28

3.1.2.	Sonido y su ecuación	30
3.1.3.	Intensidad y potencia sonoras	34
3.2.	Procesamiento de audio y voz	35
3.2.1.	Señales de audio y voz	35
3.2.2.	Sonidos voceados, no voceados y ruido	35
3.2.3.	Conversión analógica-digital	36
3.2.4.	Análisis en tiempo corto	39
3.3.	Métodos de detección de actividad acústica y voz (VAD)	40
3.3.1.	Métodos basados en energía	41
3.3.2.	Tasa de cruces por cero	44
3.3.3.	Métodos basados en medidas estadísticas	44
3.3.4.	Métodos basados en reconocimiento de patrones	46
3.3.5.	Métodos basados en análisis espectral	46
3.4.	Resumen	46
4.	Diseño e implementación del sistema	47
4.1.	Requerimientos generales del sistema y objetivos académicos	47
4.2.	Descripción general	48
4.3.	Adquisición de señales	49
4.3.1.	Micrófono	49
4.3.2.	Circuito acondicionador de la señal	50
4.3.3.	Convertidor analógico-digital	54
4.4.	Detección de actividad acústica	56
4.5.	Caracterización y control del servomotor	58
4.5.1.	Generador de PWM	59
4.5.2.	Algoritmo de control	62
4.6.	Visualización de video	65
4.7.	Resumen	66
5.	Pruebas y resultados	67
5.1.	Montaje y colocación del sistema	67
5.2.	Desempeño del algoritmo de detección de actividad acústica	69

5.2.1. Parámetros de desempeño	69
5.2.2. QUT-NOISE-TIMIT para la evaluación de algoritmos de detección de actividad	71
5.2.3. Pruebas del algoritmo y resultados	72
5.3. Movimiento de la cámara	76
5.3.1. Control de posición angular	76
5.3.2. Control de velocidad angular	77
5.4. Resumen	81
6. Conclusiones	83
6.1. Conclusiones generales	83
6.2. Trabajo a futuro	84
6.2.1. Para el algoritmo de detección de actividad acústica	84
6.2.2. Para el algoritmo de control del servomotor	84
6.2.3. Para el sistema de vigilancia completo	84
7. Anexos	85
7.1. Características generales del DSP TMS320F28377S	85
7.2. Curvas características del motor	86
Glosario	89
7.3. Códigos de los programas	91
7.3.1. Código del sistema completo	91
7.3.2. Código para pruebas del detector	100
Bibliografía	110

Índice de figuras

2.1. Sistema en lazo abierto [1]	8
2.2. Sistema en lazo abierto simplificado [1].	8
2.3. Sistema en lazo cerrado [1]	9
2.4. Sistema en lazo cerrado simplificado [1].	9
2.5. Motor de DC simple.	11
2.6. Motor de DC [2] con electroimán.	13
2.7. Señal cuadrada para PWM	13
2.8. Señal de PWM asimétrica [3].	15
2.9. Señal de PWM simétrica [3].	16
2.10. Puente H ideal.	18
2.11. Puente H moviendo un motor de DC.	19
2.12. Voltaje y corriente en el motor de DC.	19
2.13. Modelo equivalente de un motor de DC.	21
2.14. Voltajes y aproximación de corrientes en el motor.	21
2.15. Control de posición utilizando un servomotor	24
4.1. Diagrama de bloques del sistema	49
4.2. Hardware del sistema	49
4.3. Microfono Electret	50
4.4. Etapa de preamplificación	51
4.5. Etapa de amplificación	53
4.6. Nivel de DC	54
4.7. Circuito de acondicionamiento completo	54
4.8. Diagrama de flujo del algoritmo para la detección de actividad acústica	57

4.9. Diagrama de flujo del algoritmo para la generación de ventanas de la señal . . .	58
4.10. Diagrama de flujo del algoritmo para el control	63
5.1. Sistema completo y conexión de sus elementos	68
5.2. Parámetros objetivos de desempeño	70
5.3. MR, FAR y HTER para el ambiente cafetería	74
5.4. MR, FAR y HTER para el ambiente sala y calle	75
5.5. Resultados de tres iteraciones para obtener la posición angular	76
5.6. Promedio de las tres iteraciones para obtener la posición angular	77
5.7. Velocidad angular en función del retraso de tiempo	78
5.8. Velocidad angular en función del incremento de posición angular	79
5.9. Intervalo de comportamiento lineal. Retraso (Delay)	80
5.10. Intervalo de comportamiento lineal. Incremento angular ($\Delta\theta$)	80
7.1. Curva característica de velocidad angular contra "par" para el motor utilizado	87
7.2. Curva característica de velocidad angular contra potencia para el motor utilizado	88

Capítulo 1

Introducción

Los sistemas de vigilancia nos permiten conocer el comportamiento de un fenómeno, proceso o evento deseado, de forma remota y sin la necesidad de que una persona esté presente donde ocurre el evento. Esto resulta de especial utilidad al observar sucesos de larga duración o eventos que ocurren en momentos imprevistos.

La vigilancia se puede realizar de diversas formas y utilizando diferentes equipos como computadoras, cámaras, teléfonos, equipos biométricos, vehículos aéreos, micrófonos, etc. Dependiendo del fenómeno que se requiera vigilar, será el equipo con el que contará el sistema. Existen compañías dedicadas a la fabricación, venta e instalación de equipos de vigilancia; siendo los más comunes aquellos que incluyen micrófonos y cámaras dentro de su funcionamiento.

La mayoría de los sistemas de vigilancia cuentan con elementos de control para las cámaras y algoritmos de procesamiento digital de señales en su funcionamiento, debido a la naturaleza de las señales con las que operan y al espacio reducido que usualmente ocupan. Estos algoritmos son de gran importancia para los sistemas de vigilancia pues permiten mejorar su desempeño y aumentar las capacidades de los mismos.

1.1. Sistemas de vigilancia

Debido a su gran variedad y versatilidad, los sistemas de vigilancia son ampliamente utilizados y se encuentran en una gran cantidad de lugares. Su uso va desde la seguridad en hogares, aeropuertos, bancos, etc. hasta el cuidado de pacientes en clínicas y hospitales. Cada día son más comunes de encontrar debido a que los avances tecnológicos han permitido reducir su costo y los han vuelto más accesibles y fáciles de utilizar para los usuarios.

El diccionario de la Real Academia Española define vigilancia como *”Cuidado y atención exacta en las cosas que están a cargo de cada uno”* [4]. Este cuidado y atención se puede llevar a cabo de una gran variedad de maneras y utilizando diversos dispositivos. Dentro de los sistemas de vigilancia se encuentran los de video vigilancia, que utilizan cámaras y micrófonos para su operación y sobre los cuales está centrado el trabajo presente.

La video vigilancia ha ido evolucionando a lo largo de los años volviéndose cada vez más sofisticada. En el siglo XIX, con el auge de las cámaras portátiles, aparecieron fotógrafos ocultos que observaban y capturaban escenas y personas en lugares comunes. Posteriormente a finales de ese siglo y principios del siglo XX se inventaron las cámaras de video, el fonógrafo y el teléfono dando inicio a los sistemas de video vigilancia modernos [5]. En 1923 se inventa y se hace una demostración de la televisión [6], abriendo camino a estudios sobre transmisión de señales de audio y video, así como mejoras en la grabación de video.

Posteriormente, durante las dos guerras mundiales los sistemas de vigilancia fueron utilizados por los países involucrados con el fin de aventajar a sus rivales y mantener bajo cuidado a los presos. Durante el siglo XX, con el surgimiento de dispositivos semiconductores se realizaron mejoras en la transmisión de señales de audio por teléfono y por radio y se facilitó la transmisión de video; los cuales son algunos elementos presentes en diversos sistemas de vigilancia.

Años después, en la década de los 50's se establecen las primeras comunicaciones utilizando microondas y las transmisiones comerciales de televisión a color. Esto dió origen a los sistemas de video vigilancia como los conocemos hoy en día.

En la segunda mitad del siglo XX, con el surgimiento y evolución de los dispositivos digitales se pudo introducir el procesamiento digital de audio y video en tiempo real incorporado en la mayoría de los sistemas actuales de video vigilancia. Además se mejoraron los elementos que permiten realizar el control de estos sistemas.

Actualmente los sistemas de video vigilancia permiten la visualización del video de las cámaras en celulares, dispositivos inteligentes y computadoras, utilizando direcciones IP. Además cuentan con diversas funciones configurables desde los dispositivos móviles y son de tamaño y precio reducido.

Como se describió en los párrafos anteriores, los avances en comunicaciones, procesamiento y transmisión de señales de audio y video ha estado altamente presentes en los sistemas de video vigilancia. Actualmente, parte de la investigación relacionada a estos sistemas se centra en desarrollar e implementar algoritmos para analizar el video y los datos obtenidos con el fin de detectar eventos de forma automática [7].

Las aplicaciones de los algoritmos previamente mencionados son diversas y tienen diferentes usos. Algunas de estas aplicaciones son la grabación y transmisión de multi video de alta definición (HD), detección y monitoreo de movimiento, codificación para dispositivos móviles, conteo de peatones, entre otras.

Los sistemas de vigilancia tienen tres componentes básicos [8]:

1. Detección: es el conjunto de elementos o dispositivos mediante los cuales se adquieren datos del entorno o fenómeno que se va a vigilar. Estos datos se envían a la unidad de procesamiento y control.
2. Unidad de procesamiento y control: recibe las señales y datos de los sensores, realiza el procesamiento necesario de los mismos y activa los subsistemas de alerta y monitoreo

para el operador.

3. Periféricos de salida y alerta: son los dispositivos encargados de desplegar lo detectado por los sensores y dar aviso al operador del sistema.

Los sistemas de video vigilancia cuentan con los componentes mencionados previamente. Para la detección y monitoreo utilizan cámaras de video, micrófonos y otros sensores para adquirir señales del entorno a vigilar. El procesamiento y control se realiza actualmente mediante computadores o microcontroladores que permiten la operación del sistema en tiempo real. Los periféricos de salida incluyen un monitor que permite visualizar el video de las cámaras y otros dispositivos de alerta y almacenamiento.

La adquisición y procesamiento de la señal de video proveniente de la cámara está más allá del alcance de este trabajo, pero se dará una breve explicación del funcionamiento de esta parte del sistema. Una imagen o video se puede representar de forma digitalizada como un número de "cuadros" por unidad de tiempo. Cada cuadro consta de un número de componentes (usualmente tres colores), cada componente está formado por un conjunto de píxeles con una determinada precisión (número de bits usados para representarlos). Cada componente del cuadro se traza línea por línea en el monitor y la superposición de los componentes forma la imagen a color.

Muchas herramientas y técnicas de análisis de video se centran en la generación del mismo, en extraer y proveer representaciones basadas en contenido en forma de bordes, puntos característicos y otras características [7].

1.2. Inconvenientes comunes en sistemas de vigilancia

Los sistemas de video vigilancia tienen un conjunto de cámaras y sensores para llevar a cabo su propósito. En ocasiones, estos sistemas cuentan con un número reducido de cámaras o el campo de visión de las mismas no es suficiente para observar el evento deseado.

En numerosas ocasiones los sistemas de video vigilancia se mantienen funcionando continuamente o están activos por periodos de tiempo prolongados. Esta forma de operar es comúnmente utilizada, pero poco conveniente ya que no permite detectar cambios en el ambiente o espacio y puede resultar en una mayor demanda de potencia y en un elevado consumo energético; además implica un mayor desgaste a los equipos utilizados en el sistema.

Por otro lado, los operadores de los sistemas de video vigilancia generalmente observan un elevado número de cámaras. Usualmente todas las cámaras están activas y no se monitorean al mismo tiempo o se dejan desatendidas, lo que puede ocasionar que eventos importantes se pierdan de vista.

Los inconvenientes previamente mencionados se pueden superar de distintas formas. El intervalo de visión de la cámara se puede aumentar si ésta no se mantiene estática y se gira de forma controlada; además permite seguir objetos en movimiento. El consumo de energía de las cámaras se puede reducir si solo se activan en periodos cortos de tiempo cuando ocurre el fenómeno que se desea observar o un evento extraordinario; esto también permite centrar la atención del observador cuando se tiene un elevado número de cámaras.

1.3. Propuesta frente a la problemática presentada

El presente trabajo muestra el diseño e implementación de un sistema de vigilancia automático que utiliza una cámara de video, un micrófono y la tarjeta de desarrollo LaunchXL-F28377S de Texas Instruments TM, que cuenta con un DSP para realizar el procesamiento y control. La realización de este sistema de vigilancia implicó el desarrollo e implementación del control de un servomotor usado en el movimiento de la cámara y de un algoritmo de detección de actividad acústica para la activación automática del sistema.

1.4. Objetivo

Analizar, diseñar e implementar el control de movimiento, a partir de detección actividad acústica en un espacio, para un servomotor que girará una cámara de vigilancia acoplada a este último y por medio de un procesador digital de señales.

1.5. Breve descripción de capítulos

A continuación se muestra de forma resumida el contenido de cada capítulo con el fin de ofrecer al lector un panorama general sobre el escrito realizado.

En el *Capítulo 1* se presenta el tema a desarrollar, se especifica el objetivo de la tesis, se da una visión general sobre los sistemas de vigilancia y su funcionamiento, así como algunos inconvenientes presentes en algunos de ellos y como se pueden evitar.

Posteriormente en el *Capítulo 2* se abordan, de forma resumida, los aspectos teóricos relacionados al control de servomotores. Se da una definición de servomecanismos, su importancia y su uso. Se mencionan aspectos teóricos fundamentales de control que sirven de apoyo y para ayudar a comprender lo descrito en secciones posteriores. Se exponen los elementos básicos de un servomecanismo y finalmente se explica el principio de funcionamiento de un servomotor y la forma en la que opera el control del mismo.

El *Capítulo 3* contempla aspectos físicos relacionados a la producción de sonidos y describe formas comúnmente utilizadas en la detección automática de los sonidos. Al inicio del capítulo se dedica una sección que explica de forma general las ondas mecánicas, la generación de sonidos y algunas propiedades relacionadas a éstos. En las subsecciones posteriores se presentan fundamentos teóricos para trabajar con señales de audio y voz dentro de sistemas digitales. Para finalizar el capítulo se describen algunos de los algoritmos comúnmente utilizados para la detección de actividad acústica y de voz por medio de dispositivos digitales.

Dentro del *Capítulo 4* se describe detalladamente el sistema de vigilancia que se implementó. Se muestran diagramas del sistema, las partes que lo conforman y se explica el funcionamiento de cada una de forma extensa. Se mencionan todos los elementos utilizados en la implementación y su uso.

El montaje del sistema, las partes del mismo y diversas pruebas realizadas se muestran en el *Capítulo 5*. En las pruebas se evalúa el desempeño del algoritmo de detección de actividad acústica para diferentes condiciones del entorno. También se evalúa el algoritmo de control y se establece la forma en que se puede mover la cámara utilizando el servomotor.

El *Capítulo 6* comprende las conclusiones del trabajo y menciona como se puede utilizar o incluir lo desarrollado en este trabajo para realizar futuros proyectos o de referencia para trabajos posteriores.

Finalmente, los capítulos y contenido restante se utilizan para presentar las fuentes consultadas en la elaboración del texto y anexos que complementan al trabajo o que pueden servir de referencia para comprender mejor el funcionamiento del sistema.

Capítulo 2

Servomotores y PWM

Los servos o servomecanismos están presentes en una gran cantidad de procesos industriales y domésticos utilizados en la actualidad, van desde el control de temperatura de un horno industrial hasta el posicionamiento de un cabezal de impresora.

Un servo es un sistema que utiliza retroalimentación negativa para inducir una acción que causa que la salida se modifique de acuerdo a la entrada. Los servomecanismos se utilizan usualmente en acciones que requieren de una mayor potencia o precisión a la ejercida por un operador, o que se necesita se realicen de forma automática. La retroalimentación es un término que está altamente relacionado a los servos y a su control y se debe tratar para tener un mejor entendimiento sobre estos.

Este capítulo presenta el estudio y análisis de los servomecanismos, haciendo énfasis en servomotores debido a que son parte fundamental del proyecto que se desarrollará en esta tesis.

2.1. Control y retroalimentación

Los sistemas de control se pueden clasificar en dos tipos: sistemas de control en lazo cerrado o sistemas de control en lazo abierto. La diferencia entre ambos tipos de sistemas recae en la retroalimentación o realimentación del mismo.

La retroalimentación es la recuperación de información acerca del proceso que se está controlando [9]. Generalmente los sistemas de control hacen uso de la retroalimentación para realizar acciones correctivas que ayuden a mantener una salida deseada, también sirve para verificar que el sistema este funcionando adecuadamente y ajustar la salida.

Sistemas de control en lazo abierto. Son sistemas en los que el control no está afectado por la salida producida, en estos sistemas no se retroalimenta la salida ni se compara con la entrada. No permiten compensar errores ocasionados por perturbaciones que se sumen a la señal de actuación del controlador.

La *Figura 2.1* muestra el diagrama de bloques de un sistema general en lazo abierto, cada uno de los bloques es un subsistema con características propias. El transductor de entrada recibe y adecúa la señal de entrada de forma que pueda ser empleada por el controlador. El controlador se encargará de manejar la planta o el proceso. Las perturbaciones son señales agregadas que se suman de forma algebraica a la generada por el controlador o a la salida.

Las perturbaciones pueden llegar a corromper la salida generando una salida no deseada que no se modifica de acuerdo a una señal de error, lo cual representa la desventaja de este tipo de control.

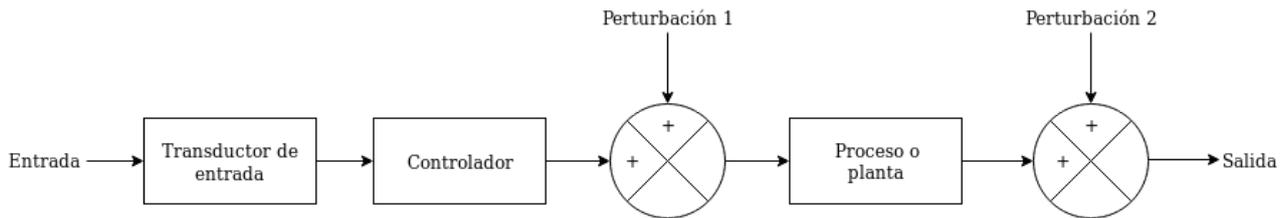


Figura 2.1: Sistema en lazo abierto [1]

Los subsistemas representados en el diagrama de bloques de la *Figura 2.1* tienen una función de transferencia o ganancia asociada a ellos. Si se considera que no existen perturbaciones y tomando como entrada la señal que recibe el controlador, el diagrama de bloques se puede simplificar como se muestra en la *Figura 2.2*.

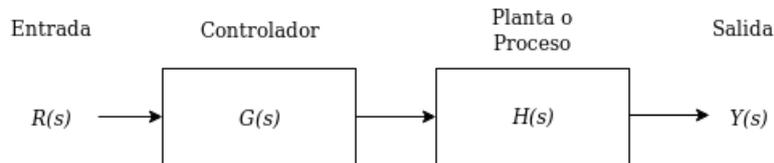


Figura 2.2: Sistema en lazo abierto simplificado [1].

La función de transferencia que relaciona la entrada del sistema $R(s)$ con la salida $Y(s)$ se obtiene multiplicando las ganancias o funciones de transferencia de los subsistemas $G(s)$ y $H(s)$ resultado en

$$\frac{Y(s)}{R(s)} = G(s)H(s) \quad (2.1)$$

Ganancia en lazo abierto

Sistemas de control en lazo cerrado o sistemas retroalimentados. Son aquellos que comparan la salida del sistema con su entrada y utilizan la diferencia entre ambas como medio de control [10]. Los sistemas retroalimentados obtienen la diferencia entre la señal de salida y de entrada y alimentan al controlador con ella. El controlador va a desencadenar una acción que busca reducir la diferencia mencionada o error a cero y llevar a la salida a un valor deseado.

El diagrama de bloques mostrado en la *Figura 2.3* representa un sistema en lazo cerrado típico, en estos sistemas la salida se realimenta a un punto donde se compara con una señal de

referencia. El transductor de salida sirve para obtener y medir la señal de salida, el subsistema de retroalimentación convierte y adapta la señal de salida para que pueda compararse con la entrada. Cada uno de los subsistemas del diagrama de bloques de la *Figura 2.3* tiene una función de transferencia o ganancia asociada.

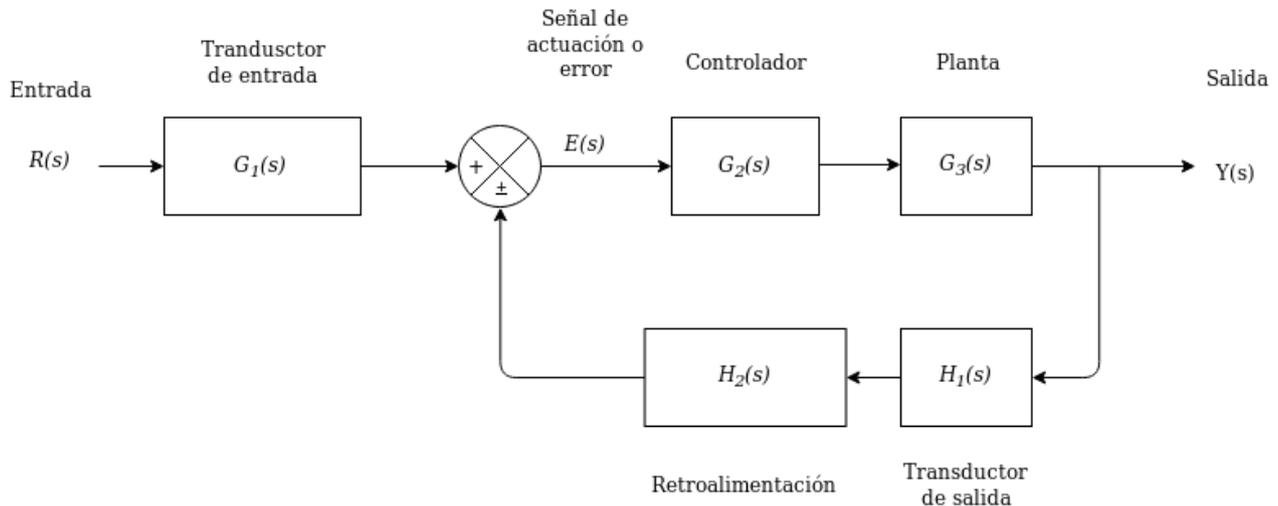


Figura 2.3: Sistema en lazo cerrado [1]

El diagrama de la *Figura 2.3* se puede simplificar para llegar a un modelo como el mostrado en la *Figura 2.4*.

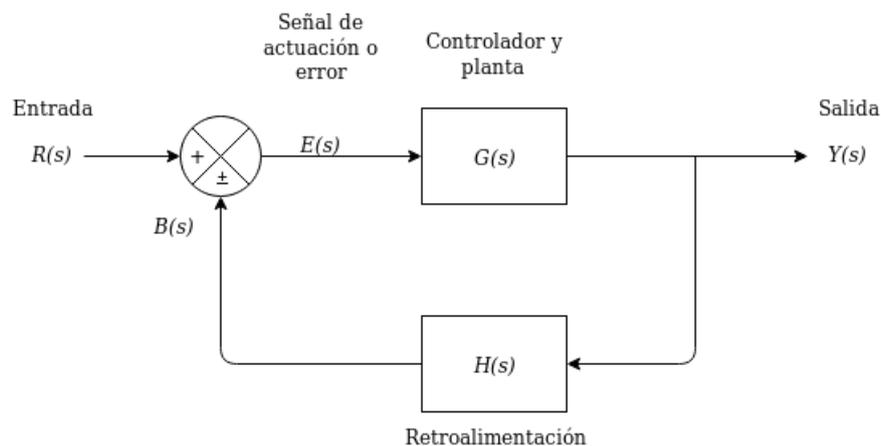


Figura 2.4: Sistema en lazo cerrado simplificado [1].

Si se obtiene la función de transferencia que relaciona el error $E(s)$ con la salida de la retroalimentación $B(s)$, se llega a la expresión

$$\frac{B(s)}{E(s)} = G(s)H(s) \quad (2.2)$$

A esta expresión se le conoce como ganancia de lazo abierto y es similar a la generada en el modelo del sistema de control en lazo abierto, si se considera al subsistema de retroalimentación como la planta.

La ganancia de lazo cerrado o la función de transferencia que relaciona la salida $Y(s)$ y la entrada $R(s)$ se deduce a partir del diagrama de bloques cómo se muestra a continuación; la salida del sistema $Y(s)$ se obtiene realizando el producto de la señal de error con la función de transferencia del controlador y la planta.

$$Y(s) = G(s)E(s) \quad (2.3)$$

Del diagrama de bloques se observa que $E(s) = R(s) + B(s)$ y que $B(s) = H(s)Y(s)$, entonces

$$Y(s) = G(s)[R(s) \pm B(s)] \quad (2.4)$$

$$Y(s) = G(s)R(s) \pm G(s)H(s)Y(s) \quad (2.5)$$

$$Y(s) \pm G(s)H(s)Y(s) = G(s)R(s) \quad (2.6)$$

$$Y(s)[1 \pm G(s)H(s)] = G(s)R(s) \quad (2.7)$$

$$\frac{Y(s)}{R(s)} = \frac{G(s)}{1 \pm G(s)H(s)} \quad (2.8)$$

Ganancia en lazo cerrado

Se observa que en la *Ecuación* (2.8) aparece el producto $G(s)H(s)$, es decir, la ganancia en lazo abierto del sistema.

2.2. Elementos de los servomecanismos

Los servos cuentan con dos elementos básicos: un sumador y un amplificador. Estos son necesarios si se quiere que funcionen como un sistema retroalimentado.

El **sumador** se encarga de sumar la señal proveniente de la realimentación negativa con la de entrada o comando para generar una señal de error que desencadenará una acción para ajustar la salida.

El **amplificador**, como su nombre lo indica, se encarga de amplificar la señal de error. Toma esta señal y la multiplica por un factor de amplificación referido comúnmente como ganancia del sistema.

Al clasificar a los servos como sistemas retroalimentados se sugiere que existan señales que viajan dentro del lazo después de excitar el sistema con una entrada y hasta que se obtiene un equilibrio, es decir, estos sistemas modificaran la salida haciendo que el error disminuya poco a poco hasta que sea nulo.

2.3. Servomotores

Los servomotores son motores dentro de un lazo cerrado de control que permiten obtener desplazamientos, velocidades y aceleraciones angulares precisas. Utilizan un sensor que entrega la señal de realimentación con la que se llega a la posición o velocidad angular deseada. Los servomotores se utilizan en diversas aplicaciones de robótica, telecomunicaciones, transporte industrial, en máquinas de grabado y corte, sistemas de vigilancia, etc.

Los servomotores pueden ser de dos tipos. El primer tipo es el servomotor de corriente directa (DC servo) y el segundo es el servomotor de corriente alterna (AC servo).

2.3.1. Motores de DC

La *Figura 2.5* muestra el diagrama de un motor simple de corriente directa (DC motor). Consiste de un embobinado de alambre que rota en el espacio alrededor de un eje fijo con una determinada velocidad.

Los motores de DC, al igual que los de AC, constan de dos partes. La primera es la parte móvil o rotatoria llamada rotor y la segunda es la parte estática conocida como estator. El rotor está formado por el embobinado con sus extremos abiertos y el conmutador. El conmutador consta de dos segmentos semicirculares conductores a los cuales están adheridos los extremos abiertos del embobinado. Por otro lado, el estator abarca a las escobillas y los imanes que generan un campo magnético \vec{B} permanente.

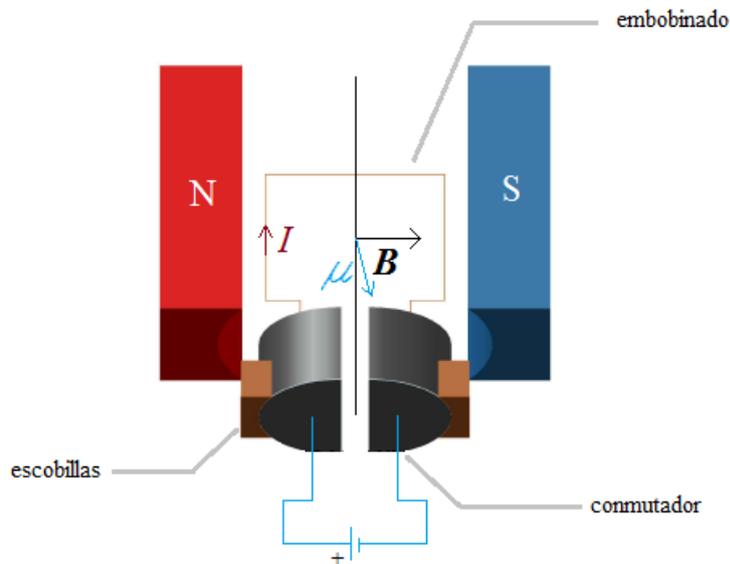


Figura 2.5: Motor de DC simple.

Principio de funcionamiento

Los segmentos del conmutador hacen contacto con las escobillas o terminales, éstas últimas están conectadas a una fuente de voltaje externa. Cuando se energiza la fuente de voltaje una corriente I fluye a través del rotor (bobina y conmutador); esta corriente genera un campo magnético y una fuerza en cada uno de los segmentos del embobinado. Las fuerzas en los segmentos del embobinado opuestos entre sí son de igual magnitud, pero de diferente sentido (par de fuerzas), lo cual genera un momento magnético $\vec{\mu}$. La dirección de este momento magnético estará dada por la regla de la mano derecha, enroscando los dedos en dirección que circula la corriente a través del embobinado [11].

El rotor se encuentra entre los polos opuestos de un imán permanente, por lo que hay un campo magnético \vec{B} que produce un par de torsión $\vec{\tau} = \vec{\mu} \times \vec{B}$ que actúa sobre el rotor. El par de torsión provocará un giro de 90° del rotor y el sentido del mismo dependerá de la orientación del par de torsión. Si la corriente a través del rotor fuera constante, el rotor giraría en sentido contrario y este oscilaría alrededor de una misma posición, pero esto no ocurre por la acción del conmutador. Cuando el rotor gira 90° ambos segmentos del conmutador se encuentran en contacto con las escobillas, ocasionando que no haya diferencia de potencial entre los segmentos del conmutador. Por lo tanto en ese instante no fluye corriente por el rotor y el momento magnético es cero. Debido a la inercia, el rotor continúa girando hasta los 180° . En esta posición otra vez fluye corriente a través del rotor, pero ahora los segmentos del conmutador se encuentran en posición invertida, es decir, la corriente entra por el otro lado del rotor y sale por donde entró inicialmente. Aunque el rotor se haya invertido con respecto a la dirección de la corriente, el momento magnético y el par de torsión están en la misma dirección con respecto al campo magnético, haciendo que vuelva a girar en el mismo sentido repitiendo el proceso descrito anteriormente y completando la rotación de 360° . El conmutador ocasiona que la corriente se invierta cada 180° de giro, pero el par de torsión es unidireccional haciendo que el rotor gire en un mismo sentido.

Los embobinados del rotor de los motores prácticos tienen muchas vueltas *Figura 2.6*; esto incrementa el momento y el par de torsión magnéticos. Además con el fin de minimizar la reluctancia en el circuito magnético del motor, se intercala el embobinado entre una pila de láminas de acero y se deja el menor espacio de aire posible entre el rotor y el estator de la máquina.

El par de torsión de los motores se puede incrementar si se utiliza un campo magnético más intenso, razón por la cual muchos diseños de motores utilizan electroimanes en vez de un imán permanente, como se observa en la *Figura 2.6*. La magnitud del par de torsión se puede mantener constante si el rotor incluye varias bobinas independientes de alambre, orientadas con diferentes ángulos. Para mantener constante el flujo magnético en el hueco de aire entre los polos del imán, se utilizan imanes redondos o curvos.



Figura 2.6: Motor de DC [2] con electroimán.

2.4. Modulación por ancho de pulso (PWM)

La modulación por ancho de pulso es una forma de representar una señal analógica utilizando una señal digital [3]. Las señales de PWM son señales cuadradas periódicas en las que se varía el ancho del pulso cuadrado, es decir, varía el ciclo de trabajo. El ciclo de trabajo de una señal (D) se define como la razón entre el tiempo que una señal periódica está activa (t_{on}) y el periodo de la misma (T).

$$D = \frac{t_{on}}{T} \quad (2.9)$$

Usualmente el ciclo de trabajo se representa mediante un porcentaje, por lo que la Ecuación (2.9) se expresa como:

$$D \% = \frac{t_{on}}{T} * 100 \quad (2.10)$$

Las señales de PWM tienen otras características a considerar que definen su comportamiento además del ciclo de trabajo. El período T de estas señales, y por lo tanto su frecuencia, está siempre fijo; además la amplitud de la señal es constante como se aprecia en la *Figura 2.7*.

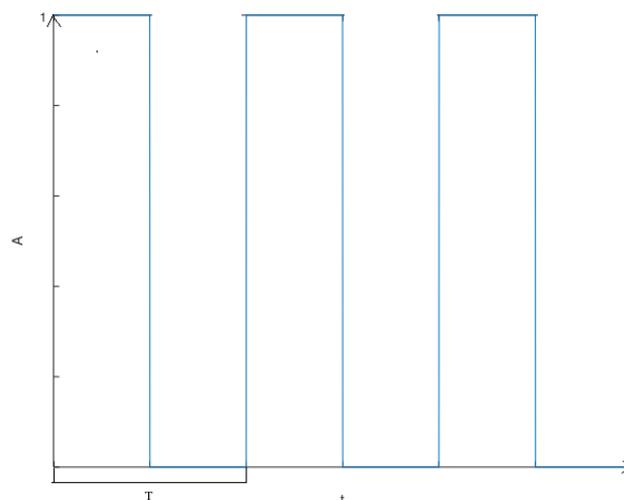


Figura 2.7: Señal cuadrada para PWM

Si una señal digital se enciende y apaga a una frecuencia suficientemente rápida, y con un determinado ciclo de trabajo, el comportamiento de la salida aparentará ser una señal analógica de voltaje constante, debido a que el valor medio de la señal de PWM en un periodo T es el mismo que el valor de la señal analógica. El ancho del pulso será proporcional a la amplitud instantánea de la señal y la energía de la señal de PWM es aproximadamente igual a la energía de la señal original [3].

La señal de PWM se controla o depende de otra señal moduladora. La señal moduladora puede ser una señal analógica de amplitud variable o una secuencia de valores en algún dispositivo, esto último dependerá de como se genere el PWM.

Generación y análisis de señales de PWM

La señal de PWM puede generarse utilizando circuitos analógicos o digitales. Independientemente del tipo de circuito que se utilice para este propósito, el proceso a seguir para obtener el PWM es similar y se describe a continuación:

1. Se establece una base de tiempo. La base de tiempo es una señal periódica que va incrementando o disminuyendo su amplitud de forma controlada y constante. El periodo de esta señal determinará el periodo y la frecuencia de la señal de PWM que se quiere generar.
2. Se crea un valor de comparación o referencia. El valor de comparación o referencia varía en el tiempo y se utiliza para controlar el ancho del pulso de la señal. Si se mantiene fijo, a la salida se tendrá una señal cuadrada con ciclo de trabajo constante.
3. Comparación y salida modulada. El valor de referencia se compara constantemente con la base tiempo, cuando el valor de la base de tiempo rebasa el valor de referencia se activa el pulso de la señal de PWM hasta el final del periodo. Al terminar el periodo la base de tiempo se reinicia, el pulso se desactiva y se vuelve a activar hasta que vuelve a rebasar el valor de comparación.

La mayoría de los dispositivos y circuitos ofrecen gran flexibilidad cuando se genera la señal de PWM. Estos permiten modificar la frecuencia de la señal, la forma de la base de tiempo, el valor de comparación, la acción a tomar cuando se llega al valor de comparación y otros parámetros útiles al trabajar con este tipo de señales.

Al utilizar elementos analógicos en la generación del PWM la frecuencia de éste dependerá directamente de la frecuencia de la señal moduladora, mientras que al utilizar elementos digitales la frecuencia dependerá del oscilador o pulso de reloj que utilice el dispositivo y de la forma en que se configure para su uso.

Los microcontroladores usualmente cuentan con módulos altamente configurables para generar señales de PWM en sus terminales. Estos módulos permiten generar múltiples señales al mismo tiempo, en fase o desfasadas, simétricas y asimétricas ó recortadas. Además ofrecen protección contra cortos circuitos y circuitos de banda muerta [12], lo cual resulta de gran utilidad al trabajar con transistores para manejar motores como se muestra más adelante.

Generador simétrico y asimétrico de PWM

Las señales de PWM pueden ser simétricas o asimétricas dependiendo de cómo sean generadas. Si se utiliza una señal de tipo diente de sierra como base de tiempo, la señal de PWM será asimétrica, por otro lado, si se utiliza una señal triangular, la señal de PWM será simétrica [3].

Señal asimétrica: La señal asimétrica de PWM se observa en la *Figura 2.8* y tiene las características listadas a continuación:

- Permanece en cero antes de que la base de tiempo comience a incrementar.
- Cambia de valor cuando la base de tiempo alcanza el valor de comparación.
- Se mantiene sin cambios hasta el final del periodo.
- Si al final del periodo el nuevo valor de comparación es diferente de cero, cambia de valor.

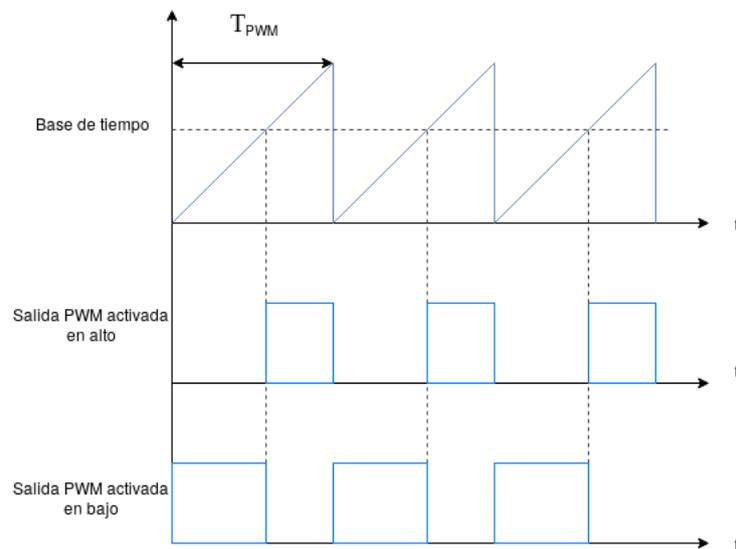


Figura 2.8: Señal de PWM asimétrica [3].

Señal simétrica: La señal simétrica de PWM se observa en la *Figura 2.9* y tiene las características listadas a continuación:

- Los pulsos están centrados respecto de cada periodo de PWM.
- Tiene dos zonas de igual duración donde la señal está inactiva .
- Permanece en cero antes de que la base de tiempo comience a incrementar.
- No cambia hasta que se alcanza el valor de comparación por primera vez.
- Se mantiene sin cambios hasta que el valor de comparación se alcanza por segunda vez cuando la base de tiempo decrementa.

- Si al final del periodo el nuevo valor de comparación es diferente de cero, permanece sin cambios.
- Al generar una señal asimétrica existen dos comparaciones, una durante el incremento de la base de tiempo y otra durante el decremento.

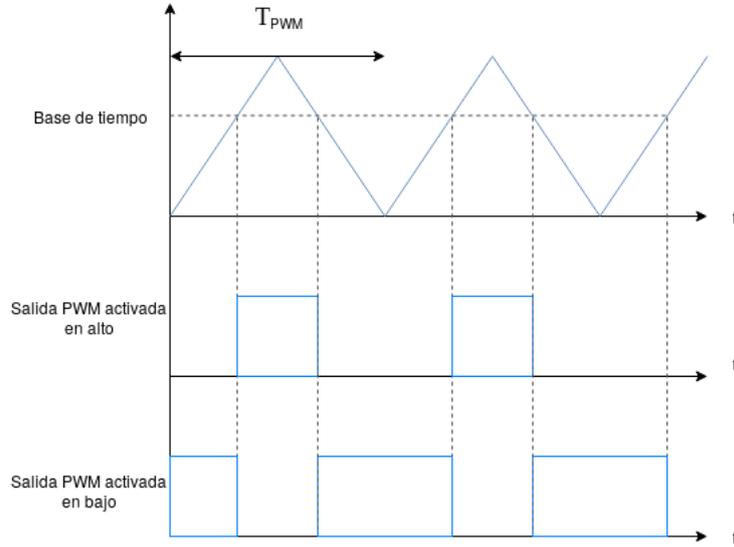


Figura 2.9: Señal de PWM simétrica [3].

Un análisis para obtener el voltaje promedio de este tipo de señales se muestra a continuación. Este voltaje promedio producirá el efecto perceptible por los humanos cuando se conecta la carga a la señal de PWM. El voltaje promedio de una señal periódica continua en intervalos se define como:

$$V_{avg} = \frac{1}{T} \int_{t_0}^T v(t) dt \quad (2.11)$$

La señal de PWM se puede definir en trozos o intervalos de tiempo dentro de un periodo. En el primer intervalo el PWM está encendido y su voltaje es igual al voltaje de la fuente. Es decir, para $t_0 < t < t_{on}$, $v(t) = V_s$. En el segundo intervalo de tiempo el PWM está apagado y su voltaje es nulo: $t_{on} < t < T$, $v(t) = 0$. Por lo que $v(t)$ queda definido como se muestra en la *Ecuación* (2.12):

$$v(t) = \begin{cases} V_s & t_0 < t < t_{on} \\ 0 & t_{on} < t < T \end{cases} \quad (2.12)$$

Considerando $v(t)$ y descomponiendo la integral de la *Ecuación* (2.11), el voltaje promedio se obtiene mediante la siguiente expresión:

$$V_{avg} = \frac{1}{T} \left[\int_{t_0}^{t_{on}} V_s dt + \int_{t_{on}}^T 0 dt \right] \quad (2.13)$$

Resolviendo la integral y considerando $t_0 = 0$ se llega a:

$$V_{avg} = V_s \frac{t_{on}}{T} \quad (2.14)$$

o también

$$V_{avg} = V_s D \quad (2.15)$$

$$D = \frac{t_{on}}{T} \quad (2.16)$$

2.4.1. Control de movimiento de motores utilizando PWM

La señal con la que se controla un motor de DC necesita ser amplificada antes de aplicarse a las terminales del mismo. Usualmente la amplificación se realiza de dos formas, la primera es utilizando un amplificador lineal de potencia y la segunda es utilizando amplificadores de potencia que involucran la conmutación en su funcionamiento. Los amplificadores lineales usualmente son más simples y baratos, pero su eficiencia es regularmente baja (<80 %) [13], además necesitan fuentes bipolares si se quiere mover un motor de forma bidireccional. Si se utilizan motores cuyo uso de potencia es elevado, el calor disipado por los amplificadores puede llegar a dañarlos o a otros componentes en el circuito. Por otra parte, están los amplificadores de potencia que involucran conmutación y emplean señales de PWM para su funcionamiento. Estos son los más usados, por las razones listadas a continuación [10]:

- Son más eficientes. Al utilizar los transistores como interruptores, las pérdidas en las uniones se minimizan, esto significa menor cantidad de pérdidas de potencia en forma de calor y por tanto mayor eficiencia. Además se pueden utilizar transistores y disipadores de calor más pequeños.
- Operan con frecuencias medias y altas. Esto permite reducir el tamaño de los componentes utilizados. Además si la conmutación ocurre a frecuencias superiores al ancho de banda del sistema que accionan, el sistema actúa como un filtro paso bajas que solo responde a los componentes de baja frecuencia de la señal de salida.
- Los transistores incluyen diodos adicionales que se usan como diodos de "freewheeling" o "flyback". Los motores representan una carga inductiva para el circuito que los mueve. Cuando se deja de suministrar potencia al motor para detenerlo, la energía almacenada por la inductancia generada por el devanado del rotor se libera. Esto ocasiona un pico de voltaje negativo a través de las terminales del motor, el cual puede dañar el circuito conectado a éste. Con el diodo de "freewheeling" el pico de voltaje negativo se limita al voltaje de encendido del diodo (0.7V) y la corriente circula a través de éste y no hacia el circuito conectado al motor. Cuando se utilizan MOSFETs para mover los motores, los diodos son propios de las uniones en los transistores.
- Permiten limitar la corriente de encendido. La señal de PWM usada para mover el motor permite que la corriente se eleve lentamente al momento de encenderlo. Esto se mostrará más adelante.

Puentes H

Un puente H es un arreglo de interruptores utilizado para mover motores de forma bidireccional. La *Figura 2.10* muestra un puente H ideal. Consta de 4 interruptores estáticos que permiten o interrumpen el flujo de corriente a la carga conectada entre las terminales de salida A y B.

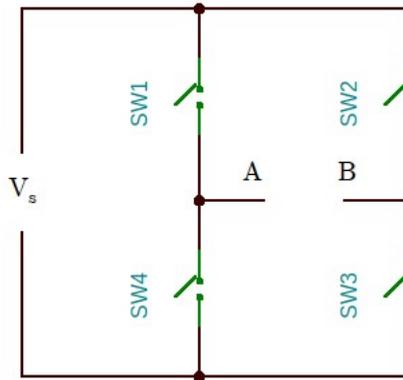


Figura 2.10: Puente H ideal.

La forma en que se usan los puentes H para mover motores se explicará a continuación. Supongamos que se quiere mover un motor sin carga en un mismo sentido y con una velocidad constante. Además el sentido de giro es el mostrado en la *Figura 2.11 (a)* cuando la corriente fluye de B a A.

En el tiempo $t = 0$ todos los interruptores están abiertos. Posteriormente se cierran los interruptores SW2 y SW4 en el tiempo $t = t_0$ con lo que un voltaje de magnitud V_s se aplica en las terminales del motor, *Figura 2.11 (a)*. La corriente en los devanados del motor empieza a incrementar lentamente y el motor se va a empezar a mover. Posteriormente, el interruptor SW4 se abre en el tiempo $t = t_1$ y el interruptor SW1 se cierra en el tiempo $t = t_1 + \delta t$ *Figura 2.11 (b)*. El tiempo δt es un pequeño intervalo de tiempo conocido como "deadband" en el cual los interruptores SW1 y SW4 están abiertos; esto se hace con el fin de evitar cortos circuitos de la fuente. La diferencia de potencial en las terminales del motor es nula y la corriente debida a la energía almacenada por la inductancia del motor fluye a través de los interruptores SW1 y SW2. Esta corriente disminuye lentamente debido a la fuerza contraelectromotriz ("back emf") del motor y circula de forma contraria a como lo hace en el tiempo $t_0 < t < t_1$. Para repetir el proceso, en el tiempo $t = t_2$ se abre el interruptor SW1 y poco después en el tiempo $t = t_2 + \delta t$ se cierra el interruptor SW4. De nuevo el voltaje en las terminales del motor es igual a V_s y la corriente incrementa de nuevo.

$$(a) SW2 \text{ y } SW4 \text{ cerrados } V_m = V_s, \quad (b) SW2 \text{ y } SW1 \text{ cerrados } V_m = 0$$

Análisis de voltajes y corrientes en el motor

Considerando el puente H mostrado en la *Figura 2.10* y para obtener control bidireccional de la rotación del motor, existen dos posibilidades que se pueden presentar con los interruptores.

1. Voltaje aplicado positivo.
 - a) SW2 siempre encendido.
 - b) SW4 encendido en $0 \leq t < t_1$.
 - c) SW1 encendido en $t_1 \leq t < t_2$.
2. Voltaje aplicado negativo.

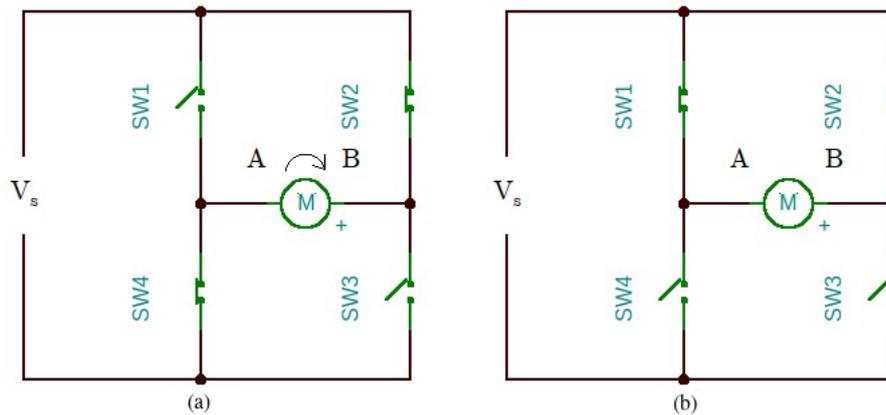


Figura 2.11: Puente H moviendo un motor de DC.

- a) SW1 siempre encendido.
- b) SW3 encendido en $0 \leq t < t_1$.
- c) SW2 encendido en $t_1 \leq t < t_2$.

La *Figura 2.12* muestra los voltajes y corrientes en el motor considerando que el voltaje aplicado en sus terminales es positivo.

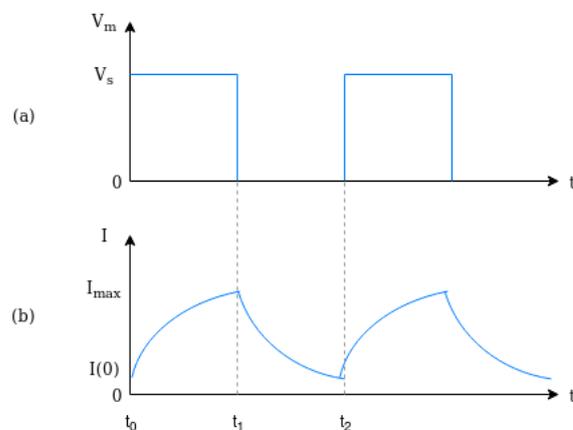


Figura 2.12: Voltaje y corriente en el motor de DC.

La forma de onda de la corriente depende de la carga que se conecte al puente H. Si la carga fuera completamente resistiva, la corriente tendría la misma forma del voltaje.

Los motores son cargas predominantemente inductivas que además presentan cierta resistencia en el embobinado, es decir, se pueden modelar como cargas RL. Esto se mostrará más adelante. Para este tipo de cargas, la forma de la corriente es exponencial como se muestra en la *Figura 2.12* [14]. Un análisis más completo utilizando estas formas de onda se puede encontrar en [14], pero para el presente trabajo se hará un análisis más simple realizando ciertas consideraciones

como se discute en [10].

El voltaje en el motor para el caso positivo está dado por la siguiente expresión y se muestra en la *Figura 2.12 (a)* y en la *Figura 2.14 (a)*.

$$V_m = V_s \quad \text{para } 0 < t < t_1 \quad (2.17)$$

$$V_m = 0 \quad \text{para } t_1 < t < T \quad (2.18)$$

como la señal de voltaje es periódica, puede expresarse utilizando una serie de Fourier [15]

$$V_m = a_0 + \sum_{n=1}^{\infty} a_n \cos(n\omega_s t) + \sum_{n=1}^{\infty} b_n \sin(n\omega_s t) \quad (2.19)$$

Considerando que la señal es par, solo tiene coeficientes a_0 y a_n , por lo que la serie queda:

$$V_m = a_0 + \sum_{n=1}^{\infty} a_n \cos(n\omega_s t) \quad (2.20)$$

Donde $\omega_s = 2\pi f_s$ y

$$a_0 = \frac{1}{T} \int_{-T/2}^{T/2} V_s dt = \frac{t_{on}}{T} V_s \quad (2.21)$$

$$a_0 = DV_s \quad (2.22)$$

Además:

$$a_n = \frac{1}{T} \int_{-T/2}^{T/2} V_s \cos(n2\pi f_s t) dt = \left(\frac{V_s}{n\pi}\right) \sin(n2\pi t_{on}/T) \quad (2.23)$$

$$a_n = \left(\frac{V_s}{n\pi}\right) \sin(n2\pi D) \quad (2.24)$$

Si se elige una frecuencia de conmutación f_s mucho mayor al ancho de banda del motor, entonces solo se considera la componente fundamental a_0 . Cuando se promedia durante muchos ciclos de conmutación se obtiene

$$V_m = a_0 = DV_s \quad (2.25)$$

El análisis de la forma de onda de la corriente se lleva a cabo como se muestra a continuación.

El motor de DC se puede modelar como se muestra en la *Figura 2.13*

Aplicando KVL para obtener el voltaje en las terminales del motor V_m se llega a:

$$V_m = V_R + V_L + V_{bemf} \quad (2.26)$$

Sustituyendo las ecuaciones de los elementos, la ecuación expresión se puede expresar como

$$V_m = i_a R_a + L_a \frac{di_a}{dt} + K_e \omega_0 \quad (2.27)$$

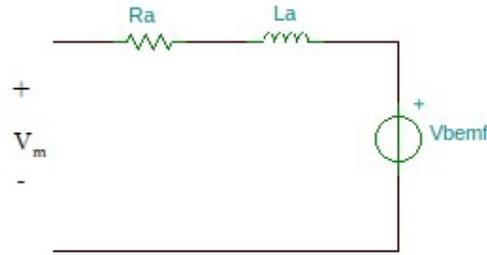


Figura 2.13: Modelo equivalente de un motor de DC.

i_a es la corriente que circula en la armadura o rotor del motor. R_a es la resistencia presente en los devanados y laminaciones del motor. L_a es la inductancia de la armadura y K_e es la constante de la fuerza contraelectromotriz del motor.

La solución de la ecuación (2.27) para la corriente $i_a = I(t)$ resulta en expresiones exponenciales como se mencionó previamente. Haciendo las siguientes consideraciones, el análisis para obtener una expresión de la corriente de salida que circula por el motor se simplifica:

- La forma de la corriente se aproxima a líneas rectas que forman una señal triangular periódica, *Figura 2.14*
- La velocidad del motor en un periodo de conmutación se mantiene constante.
- La corriente que fluye por la resistencia del circuito se puede promediar en un periodo de conmutación.

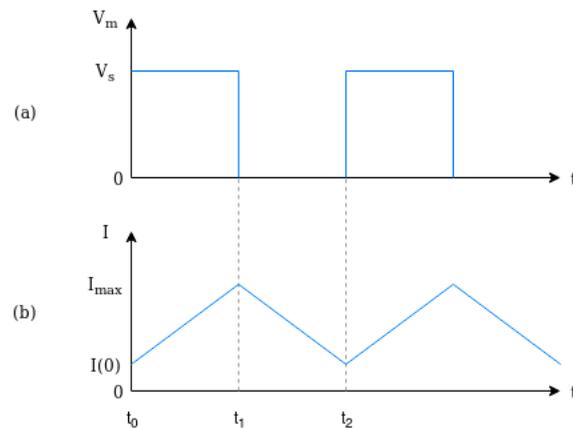


Figura 2.14: Voltajes y aproximación de corrientes en el motor.

La corriente promedio en la resistencia es

$$i_{av} = \frac{1}{t_f} \int_0^{t_f} i(t) dt \quad (2.28)$$

El voltaje $V_{bemf} = K_e \omega_0$ se mantiene constante y considerando la corriente promedio i_{av} en la resistencia, se puede expresar un nuevo voltaje de DC.

$$V_1 = R_a i_{av} + K_e \omega_0 \quad (2.29)$$

Sustituyendo este resultado en la Ecuación (2.27)

$$V_m - V_1 = L \frac{di_a}{dt} \quad (2.30)$$

Resolviendo por separación de variables y descomponiendo la integral en intervalos para $i_a = I(t)$ y recordando que para $0 \leq t < t_1$, $V_m = V_s$ y para $t_1 \leq t < t_2$, $V_m = 0$ entonces

$$I(t) = I(0) + \left(\frac{V_s - V_1}{L}\right)t \quad \text{para } 0 \leq t < t_1 \quad (2.31)$$

$$I(t) = I(t_1) - \frac{V_1}{L}(t - t_1) \quad \text{para } t_1 \leq t < t_2 \quad (2.32)$$

A partir de las expresiones obtenidas se puede conocer la máxima variación o rizo de corriente ΔI que se presentará en los devanados del motor en función de los parámetros conocidos. La variación o rizo de corriente se define como:

$$\Delta I = I(t_1) - I(0) \quad (2.33)$$

Evaluando la Ecuación (2.31) en $t = t_1$, sustituyéndola en (2.33) y considerando que $t_1 = \alpha t_2$ y $V_1 = \alpha V_s$, donde α es el ciclo de trabajo de señal de PWM, se obtiene

$$\Delta I = \left(\frac{V_s t_2}{L}\right)(\alpha - \alpha^2) \quad (2.34)$$

Para obtener el máximo rizo de corriente se deriva (2.34) respecto a α y se iguala a cero. Haciendo esto se obtiene que la máxima variación se obtiene con $\alpha = 1/2$. Sustituyendo este resultado en (2.34) se llega a:

$$\Delta I_{max} = \frac{V_s t_2}{4L} \quad (2.35)$$

2.5. Control de servomotores

Los sistemas retroalimentados comparan dos señales en algún punto para generar una señal que desencadenará una acción dentro del lazo de control. Generalmente se compara la señal de entrada con la variable de salida que se desea controlar para generar una señal de error con la que actuará el sistema. Muchas veces no es posible comparar directamente la variable de entrada con la de salida, por lo que se usan dispositivos para el sensado y adecuación en el intermedio. Los servomotores utilizan diversos sensores para medir la salida y compararla con la entrada. Entre los más comunes están los potenciómetros, los tacómetros, resistores en derivación, encoders, sensores de efecto Hall y transformadores.

Potenciómetros, tacómetros y encoders

Potenciómetros

Los potenciómetros son resistores eléctricos variables. El valor de su resistencia depende de la posición angular o lineal en la que se encuentra un eje unido a ellos. Se utilizan como

transductores electromecánicos que convierten energía mecánica en energía eléctrica. Los potenciómetros cuentan con tres terminales, una de ellas se conecta a un voltaje (usualmente fijo) y la otra a una referencia de voltaje (usualmente tierra). La tercer terminal es donde se toma el voltaje de la salida del potenciómetro, este voltaje dependerá de la posición del eje del potenciómetro y será proporcional al desplazamiento en el mismo.

El servomotor utilizado en este trabajo utiliza un potenciómetro como transductor de salida. Muchos de los servomotores pequeños emplean potenciómetros con el fin de ahorrar espacio y costo al momento de fabricarlos.

Tacómetros

Los tacómetros son dispositivos utilizados para medir la velocidad angular o velocidad de rotación, usualmente de motores y máquinas. Estos dispositivos funcionan como generadores de voltaje, en los cuales este es proporcional a la magnitud de la velocidad en el eje de entrada. Los que se emplean con mayor frecuencia son aquellos que producen una señal de DC en la salida [16].

Encoders

Los encoders o codificadores de velocidad son dispositivos que convierten desplazamientos lineales o rotatorios en una señal digital o en un tren de pulsos. Los codificadores pueden ser absolutos o incrementales. Los absolutos producen como salida un código digital distinto que indica cada incremento menos significativo. Por otra parte, los incrementales proveen un pulso por cada incremento. Los encoders incrementales están formados por cuatro partes básicas: un disco giratorio con una retícula radial formada por líneas opacas alternadas con espacios transparentes, una fuente de luz, una máscara estática y un fotodetector.

Para su funcionamiento, el disco giratorio proyecta su imagen sobre la superficie del o los receptores enmascarados. Los receptores tienen la tarea de detectar las variaciones de luz que se producen con el desplazamiento del disco convirtiéndolas en señales eléctricas, usualmente cuadradas, senoidales o triangulares. Normalmente se proporcionan dos señales cuadradas y desfasadas entre sí en 90° eléctricos. La duración del periodo de la señal cuadrada determinará la velocidad de rotación y el desfase entre las señales determinará el sentido de rotación.

Control de posición

El control de posición y velocidad para servomotores de DC es un tema de amplio interés y cuyo estudio ha sido profundo, pues su aplicación está presente en numerosas áreas como telecomunicaciones, robótica, transporte industrial, automotriz, telefonía y computación, transporte eléctrico, entre otras.

Existen diversas formas para realizar el control de posición de un servomotor. La elección de la manera en que se realizará depende de la aplicación donde se necesite, los componentes a utilizar, el costo del proyecto, la complejidad del sistema, el ruido y perturbaciones externas, etc. Una forma sencilla y comúnmente utilizada en el control de motores pequeños es la descrita a continuación.

La *Figura 2.15* muestra un sistema de control de posición en que la posición angular del eje de salida sigue a un voltaje de referencia. El eje de salida puede ser el eje del motor o parte de un

sistema de engranes conectados al eje del motor. Un potenciómetro se acopla mecánicamente al eje de salida. Este potenciómetro se encarga de proveer un voltaje de retroalimentación v_f , el cual depende de la posición del eje de salida, siendo función lineal del ángulo θ_c y proporcional al desplazamiento del eje [17].

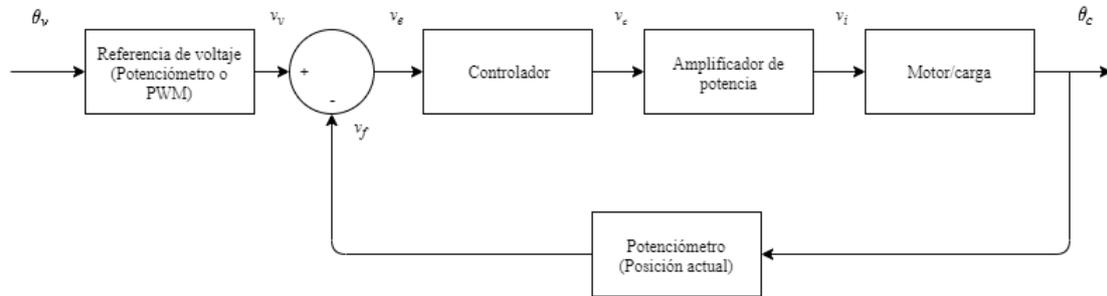


Figura 2.15: Control de posición utilizando un servomotor

El voltaje de retroalimentación proveniente del potenciómetro representa a la posición actual del eje de salida. Este voltaje se compara o resta de un voltaje de referencia v_v que representa la posición deseada. El voltaje de referencia puede obtenerse de diversas fuentes, usualmente de un potenciómetro o de una señal de PWM filtrada. La comparación del voltaje de referencia con el de retroalimentación produce un tercer voltaje llamado de error o de actuación v_e . El error se amplifica y alimenta al controlador, este último genera la señal v_c para que el amplificador de potencia a la salida mueva al eje de salida del motor hasta la posición deseada. Cuando la posición del eje de salida alcanza la posición deseada, es decir, cuando el voltaje de referencia es igual al de retroalimentación, el error se vuelve cero y no se aplica diferencia de potencial a las terminales del motor $v_i = 0$, el eje de salida se mantiene en reposo. Si se trata de mover físicamente al eje de salida se crea inmediatamente la señal de error que genera el par necesario en el motor para que se mueva a la posición deseada.

Los sistemas como el descrito anteriormente usualmente incluyen algún tipo de compensación para evitar respuestas oscilatorias o para mejorar el tiempo de repuesta y minimizar los errores debidos a la fricción. Las compensaciones incluyen agregar redes eléctricas a la entrada del amplificador de potencia o en el control, también se utilizan tacómetros para generar señales de retroalimentación que permitan ajustar el amortiguamiento de la respuesta transitoria [17].

A partir de la *Figura 2.15* y de lo descrito en la sección 2.1 de este capítulo, se puede obtener la función de transferencia del sistema.

La ganancia de lazo abierto $G(s)H(s)$ se obtiene observando que

$$G(s) = \left(\frac{v_c}{v_e} \right) \left(\frac{v_i}{v_c} \right) \left(\frac{\theta_c}{v_i} \right) \quad (2.36)$$

además

$$H(s) = \frac{v_f}{\theta_c} \quad (2.37)$$

$$G(s)H(s) = \left(\frac{v_c}{v_e}\right) \left(\frac{v_i}{v_c}\right) \left(\frac{\theta_c}{v_i}\right) \left(\frac{v_f}{\theta_c}\right) \quad (2.38)$$

La ganancia de lazo cerrado se puede obtener recordando que $A_{CL} = \frac{G(s)}{1+G(s)H(s)}$, resultando en:

$$A_{CL} = \frac{\left(\frac{v_c}{v_e}\right) \left(\frac{v_i}{v_c}\right) \left(\frac{\theta_c}{v_i}\right)}{1 + \left(\frac{v_c}{v_e}\right) \left(\frac{v_i}{v_c}\right) \left(\frac{\theta_c}{v_i}\right) \left(\frac{v_f}{\theta_c}\right)} \quad (2.39)$$

Las funciones de transferencia correspondientes a los cocientes en la ecuación (2.39) dependerán de la dinámica de cada uno de los subsistemas. A continuación se presentarán estas funciones para el sistema propuesto y realizando ciertas consideraciones.

Los potenciómetros presentan una salida de voltaje proporcional a la posición angular de su eje, entonces:

$$v(t) = K_s \theta(t) \quad (2.40)$$

K_s es la constante de proporcionalidad y para un potenciómetro de N vueltas, donde cada vuelta representa un desplazamiento de 2π [rad], está dada por:

$$K_s = \frac{V_{ss}}{2\pi N} \quad (2.41)$$

Donde V_{ss} es el voltaje de referencia al que está conectado el potenciómetro.

Por lo tanto, al aplicar transformada de Laplace \mathcal{L} :

$$\frac{v}{\theta}(s) = K_s \quad (2.42)$$

La función de transferencia para el motor con carga está dada por [10]

$$\frac{\theta}{v}(s) = \frac{nK_T}{L_a J_{eq} s^3 + (B_{eq} L_a + R_a J_{eq}) s^2 + (R_a B_{eq} + n^2 K_T K_e) s} \quad (2.43)$$

Ésta función se obtuvo considerando el modelo del motor previamente mostrado y que éste se encuentra conectado a un tren de engranes y a una carga inercial. n es la relación de transmisión de los engranes, K_T es la constante de par del motor, K_e es la constante de fuerza contraelectromotriz del motor, R_a es la resistencia en la armadura del motor, L_a es la inductancia en la armadura del motor, $B_{eq} = n^2 B_M + B_L$ es la constante de amortiguamiento equivalente y $J_{eq} = n^2 J_M + J_L$ es la constante de inercia equivalente. B_M es el coeficiente de amortiguamiento del motor y B_L el de la carga. J_L es la inercia de la carga y J_M la del rotor del motor [10].

El amplificador de potencia amplifica el voltaje de control y usualmente lo invierte, por lo que su función de transferencia es:

$$\frac{v_o}{v_i}(s) = -K_{pa} \quad (2.44)$$

Finalmente, para el controlador usualmente se utiliza uno de tipo proporcional-derivativo. Esto se debe a que cuando a un motor de DC se le aplica un voltaje constante, la posición de salida incrementa linealmente con el tiempo, es decir, se comporta como la salida de un integrador. Por eso se agrega la acción derivativa de compensación. La acción derivativa aporta un modo de obtener un controlador con alta sensibilidad, permite responder a la velocidad del cambio del error y produce una corrección significativa antes de que la magnitud del error incremente demasiado.

La función de transferencia de los controles proporcional-derivativo es la mostrada a continuación:

$$\frac{v_c}{v_a}(s) = K_p + K_d s \quad (2.45)$$

K_p y K_d son las ganancias de la parte proporcional y derivativa, usualmente la ecuación anterior se expresa de la siguiente manera considerando que $T_d = K_d/K_p$

$$\frac{v_c}{v_a}(s) = K_p(1 + T_d s) \quad (2.46)$$

A partir de las ecuaciones obtenidas se puede obtener otra expresión para la *Ecuación* (2.39), además de realizar análisis de estabilidad y respuesta en frecuencia del sistema.

2.6. Resumen

En el capítulo presente se describieron aspectos sobre el control de motores de corriente directa, el funcionamiento de servomotores de DC y los fundamentos teóricos sobre el control en servomecanismos y que se aplican a servomotores. También se mostró el principio de operación de los motores de corriente directa, los cuales son ampliamente utilizados en la industria. Se realizó un análisis sobre las señales de PWM y se mostró su uso para el control de movimiento de motores. Finalmente se propuso un sistema de control para un motor de DC, un sistema similar a ese se utiliza para la realización del presente trabajo.

Capítulo 3

Detección de actividad acústica y de voz

La detección de actividad acústica y de voz está presente como parte de muchos sistemas de comunicaciones, reconocedores de voz y audio, transmisores en tiempo real, sistemas de vigilancia, dispositivos de asistencia auditiva y multimedia. En telefonía, la detección de actividad de voz permite diferenciar entre los periodos de silencio y de voz y hace posible la transmisión discontinua (DTX). En los dispositivos de ayuda auditiva, ayuda a reducir el ruido y es parte de la mejora de audio que realiza el dispositivo. En los sistemas de vigilancia permite detectar cambios en el entorno y permite la operación discontinua del sistema, ahorrando energía.

Los sistemas de procesamiento y transmisión de voz y de audio siempre se ven afectados por la presencia de ruido ambiental. Existen diversas técnicas que se han desarrollado con el fin de disminuir el efecto del ruido dentro del sistema. Los métodos de detección de actividad acústica y de voz extraen características de la señal de audio entrante y obtienen medidas estadísticas de ésta; las características y medidas permiten conocer las condiciones del ruido ambiental ayudando a mejorar el desempeño del sistema.

Los sistemas electrónicos biológicos utilizan señales que son perceptibles por los seres humanos. Estos sistemas se hacen y configuran de tal forma que detecten fenómenos de manera similar a como lo haría un ser humano. La detección de actividad acústica y de voz mediante sistemas electrónicos digitales permite reconocer fenómenos audibles que pueden ser de importancia sin la necesidad de la presencia de una persona. A partir de la detección se pueden tomar acciones en base a lo detectado.

Antes de abordar la forma en que se realiza la detección de forma automática mediante dispositivos digitales, se presentarán diversos temas de interés asociados a la detección que permitirán facilitar su comprensión.

3.1. Ondas y sonido

Las ondas están presentes en una gran cantidad de fenómenos físicos que percibimos todos los días. Las notas producidas por un instrumento musical, los sismos y las olas son algunos

ejemplos en donde las ondas están presentes.

Los fenómenos descritos en el párrafo anterior involucran a un tipo de ondas conocidas como ondas mecánicas. Una onda mecánica es *una perturbación en un medio donde viaja*. El medio es un material o sustancia y es necesario para que este tipo de ondas se propaguen. Cuando las ondas mecánicas viajan por un medio ocasionan desplazamientos de partículas en el mismo, los desplazamientos dependerán de las características de la onda y el material en el que se propaga.

Las ondas se pueden clasificar bajo distintos criterios. Una forma de clasificar a las ondas mecánicas es de acuerdo al movimiento de las partículas del medio, relativo a la dirección de propagación de la onda. Si los desplazamientos del medio son perpendiculares o transversales a la dirección en que la viaja la onda, la onda es transversal. Si el movimiento de las partículas del medio es en la misma dirección en la que viaja la onda, es una onda longitudinal [11].

3.1.1. Ecuación de onda

Las ondas se pueden describir matemáticamente mediante funciones. La función que se utilice para describir a la onda dependerá de la naturaleza de la misma onda y la forma en la que viaja a través del medio. A continuación se describirá y analizará una función de onda, considerando que la onda es transversal y viaja en una sola dirección.

Una onda produce desplazamientos de las partículas del medio en el que viaja, si la onda es transversal los desplazamientos son perpendiculares a la dirección en la que viaja la onda. Para una partícula del medio, el valor del desplazamiento depende de la partícula que se esté considerando y del instante de tiempo que se esté analizando. Se puede establecer una función de onda para una partícula que se desplace en una sola dirección de la forma $f = y(x, t)$, con esta función se puede calcular el desplazamiento de cualquier partícula en cualquier instante en la dirección x .

Considerando que una partícula se desplace de forma oscilatoria siguiendo un movimiento armónico simple, su movimiento para el punto de origen de la onda estará dado por:

$$y(x = 0, t) = A \cos(\omega t) \quad (3.1)$$

La onda viaja a otro punto en un tiempo $t = x/v$, donde v es la rapidez de la onda y A es la amplitud de la onda. El movimiento de cualquier partícula en x en el instante t es el mismo que el movimiento en el instante anterior $t - x/v$. Sustituyendo en la *Ecuación* (3.1), se puede obtener la función de onda para cualquier punto en x y para cualquier instante de tiempo t resultando en:

$$y(x, t) = A \cos\left(\omega \left(t - \frac{x}{v}\right)\right) = A \cos\left(\omega \left(\frac{x}{v} - t\right)\right) \quad (3.2)$$

además $\omega = 2\pi f$

$$y(x, t) = A \cos\left(2\pi \left(\frac{xf}{v} - ft\right)\right) \quad (3.3)$$

Recordando que $v = \lambda f$ entonces la ecuación anterior se puede escribir como:

$$y(x, t) = A \cos \left(2\pi \frac{x}{\lambda} - 2\pi ft \right) \quad (3.4)$$

La función de onda anterior contiene el término

$$k = \frac{2\pi}{\lambda} \quad (3.5)$$

Este término se conoce como número de onda y es comúnmente utilizado para representar las funciones de onda. Entonces la *Ecuación* (3.4) se puede reescribir de la forma

$$y(x, t) = A \cos(kx - \omega t) \quad (3.6)$$

De la expresión para el número de onda se obtiene $\lambda = 2\pi/k$, combinando esto con $v = \lambda f$ y recordando que $f = \omega/2\pi$ se obtiene

$$v = \frac{\omega}{k} \quad (3.7)$$

La *Ecuación* (3.6) nos permite conocer el desplazamiento de cualquier partícula para cualquier instante de tiempo. Si se desea conocer la velocidad o aceleración de la partícula, basta con derivar respecto al tiempo una o dos veces.

La segunda derivada respecto al tiempo de (3.6) es

$$\frac{\partial^2 y}{\partial t^2} = -A\omega^2 \cos(kx - \omega t) = -A\omega^2 y(x, t) \quad (3.8)$$

De manera similar podemos derivar (3.6) respecto al desplazamiento en x una o dos veces para conocer la pendiente y la curvatura.

La segunda derivada respecto a x de (3.6) es

$$\frac{\partial^2 y}{\partial x^2} = -Ak^2 \cos(kx - \omega t) = -Ak^2 y(x, t) \quad (3.9)$$

Dividiendo (3.8) entre (3.9) y reordenando se obtiene

$$\frac{\partial^2 y}{\partial t^2} = \frac{\omega^2}{k^2} \frac{\partial^2 y}{\partial x^2} \quad (3.10)$$

Finalmente y sabiendo que $v = \omega/k$

$$\frac{\partial^2 y}{\partial x^2} = \frac{1}{v^2} \frac{\partial^2 y}{\partial t^2} \quad (3.11)$$

La *Ecuación* (3.11) es la ecuación de onda en una dimensión. Aunque se obtuvo considerando perturbaciones periódicas senoidales; las perturbaciones no periódicas también la obedecen.

La solución general para (3.11) se llama fórmula de D'Alembert o solución de D'Alembert [15] y se expresa a continuación

$$y(x, t) = \frac{1}{2}[f(x + vt) + f(x - vt)] + \frac{1}{2v} \int_{x-vt}^{x+vt} g(s) ds \quad (3.12)$$

considerando condiciones iniciales

$$y(x, 0) = f(x) \quad , \quad \left. \frac{\partial y}{\partial t} \right|_{t=0} = g(x) \quad (3.13)$$

La ecuación (3.13) es de utilidad cuando se requiere modelar otratar la señal acústica.

3.1.2. Sonido y su ecuación

El sonido es una onda longitudinal en un medio. Usualmente el estudio del sonido se realiza considerando como medio el aire, pero puede propagarse en otros materiales o sustancias. A continuación se describirá la forma en que se producen las ondas longitudinales y sus características para empezar con el estudio del sonido.

La forma en que se generan y propagan las ondas longitudinales se puede entender a través de lo planteado en seguida. Se tiene un tubo lleno de un fluido con un pistón en un extremo. Al empujar el pistón se produce una compresión en el fluido cercano al pistón aumentando la presión y densidad de esta región. La compresión hace que se empuje a la región adyacente y ésta a la que sigue y así sucesivamente formando un pulso de onda. Si el pistón se mueve con un movimiento armónico simple, en el tubo se generarán regiones de compresión y regiones de expansión. En las regiones de compresión la densidad es mayor, mientras que en las de expansión es menor. Estas diferentes regiones representan la propagación de la onda, y la longitud de onda es la distancia entre dos regiones de compresión o de expansión.

Las compresiones y expansiones forman un patrón igual que el de crestas y valles de una onda transversal senoidal, pero cada partícula en el fluido oscila en la dirección de la propagación de la onda y con la misma amplitud y periodo T que el pistón. Para este tipo de ondas se cumple la relación $v = \lambda f$ puesto que son periódicas y la ecuación de onda, considerando para esta última que el desplazamiento no es transversal y , es longitudinal ξ .

Las ondas sonoras, como se mencionó previamente, son ondas longitudinales cuyas componentes en frecuencia se encuentran en el intervalo audible. Este intervalo comprende a las ondas en el intervalo de frecuencias de 20 a 20 000 Hz, aunque también existen ondas de frecuencias mayores o menores que llevan la palabra sonido en su denominación: ultrasonido e infrasonido [11].

Las ondas sonoras se dispersan en todas direcciones a partir de la fuente que las genera, la amplitud depende de la dirección y la distancia a la fuente. La función de onda y la ecuación de onda que las modelan implican tres direcciones o variables espaciales y el tiempo. El análisis realizado a continuación se hará considerando una sola dirección, pero se puede obtener un modelo similar para tres direcciones y en diferentes sistemas de coordenadas.

Las ondas sonoras se pueden describir en términos de variaciones de presión o de desplazamientos de las partículas del medio por el que se propagan. El oído humano funciona detectando tales variaciones de presión. Los micrófonos y dispositivos similares generalmente también

detectan diferencias de presión, existiendo una relación con el desplazamiento de las partículas [11].

Una forma de describir las ondas sonoras a través de las variaciones de presión se muestra a continuación y se describe en [18].

Se considera un cubo de un material con masa constante y un determinado volumen ΔV cuyos lados miden $\Delta x, \Delta y, \Delta z$. Al existir un desplazamiento de las partículas se modifica el volumen del cubo, pero no su masa por lo que por conservación de materia se tiene:

$$\Delta m(t_0) = \Delta m(t_0 + dt) \quad (3.14)$$

En términos de la densidad y el volumen

$$\rho(t_0)\Delta V(t_0) = \rho(t_0 + dt)\Delta V(t_0 + dt) \quad (3.15)$$

Al ser funciones lineales y sabiendo que $\rho(t_0) = \rho_0$ y $\Delta V = \Delta V(t_0)$

$$\rho_0\Delta V = (\rho_0 + d\rho)(\Delta V + dV) = \rho_0\Delta V + \rho_0dV + d\rho\Delta V + d\rho dV \quad (3.16)$$

Ignorando el término $d\rho dV$ por ser muy pequeño en magnitud y reordenando se llega a

$$-\frac{dV}{\Delta V} = \frac{d\rho}{\rho_0} \quad (3.17)$$

Del módulo de compresibilidad $K = \rho_0 dp/d\rho$

$$\frac{d\rho}{\rho_0} = \frac{dp}{K} \quad (3.18)$$

Considerando que el volumen cambia en una sola dirección

$$\frac{dV}{\Delta V} \approx \frac{dx}{\Delta x} \quad (3.19)$$

dx es la diferencia en el cambio de desplazamientos du_x a ambos lados del cubo, por lo que

$$dx = du_x(x + \Delta x) - du_x(x) \quad (3.20)$$

La función diferencial se define como $df = f'(x)\Delta x$, entonces

$$dx = (du_x)'\Delta x \quad (3.21)$$

El desplazamiento depende de la posición en x y del tiempo, por lo que la derivada de la ecuación (3.21) se vuelve parcial. Además sabiendo que la velocidad es la derivada del desplazamiento respecto al tiempo $du_x = v_x dt$ [11] la ecuación anterior resulta en

$$dx = \frac{\partial(du_x)}{\partial x}\Delta x = \frac{\partial v_x}{\partial x} dt \Delta x \quad (3.22)$$

A partir de las ecuaciones (3.22), (3.19), (3.17) y (3.18) se obtiene

$$\frac{dp}{K} = -\frac{\partial v_x}{\partial x} dt \quad (3.23)$$

también

$$\frac{1}{K} \frac{dp}{dt} = -\frac{\partial v_x}{\partial x} \quad (3.24)$$

El término diferencial de la izquierda se puede representar con la ecuación mostrada a continuación

$$\frac{dp}{dt} = \left[\frac{\partial p}{\partial t} + v_x \frac{\partial p}{\partial x} \right] \quad (3.25)$$

considerando la aproximación de baja velocidad $v_x \approx 0$ entonces

$$\frac{dp}{dt} \approx \frac{\partial p}{\partial t} \quad (3.26)$$

finalmente

$$\frac{1}{K} \frac{\partial p}{\partial t} = -\frac{\partial v_x}{\partial x} \quad (3.27)$$

A partir de la segunda ley de Newton $F = ma$ y de $F = -pS$ se tiene

$$\Delta F_x = \Delta m \frac{dv_x}{dt} \quad (3.28)$$

$$\Delta F_x = -\Delta p_x \Delta S \quad (3.29)$$

la diferencial total de una función se puede definir como [19]

$$dy = \frac{\partial y}{\partial x_1} dx_1 + \dots + \frac{\partial y}{\partial x_n} dx_n \quad (3.30)$$

entonces

$$dp_x = \frac{\partial p}{\partial x} dx + \frac{\partial p}{\partial t} dt \quad (3.31)$$

considerando que las variaciones son pequeñas $\Delta p_x \approx dp_x$ y $\Delta x \approx dx$, la ecuación (3.29) se puede expresar

$$\Delta F_x = - \left(\frac{\partial p}{\partial x} \Delta x + \frac{\partial p}{\partial t} dt \right) \Delta S \quad (3.32)$$

Para $dt \rightarrow 0$ y sabiendo que $\Delta S \Delta x = \Delta V$ la ecuación (3.29) se aproxima a

$$\Delta F_x \approx -\frac{\partial p}{\partial x} \Delta V \quad (3.33)$$

combinando la ecuación anterior con (3.28) se obtiene

$$-\frac{\partial p}{\partial x} \Delta V = \Delta m \frac{dv_x}{dt} \quad (3.34)$$

expresando a la masa en términos de la densidad y el volumen $\Delta m = \rho \Delta V$

$$-\frac{\partial p}{\partial x} = \rho \frac{dv_x}{dt} \quad (3.35)$$

Similarmente como se realizó con (3.25), la derivada total de la ecuación anterior se aproxima a la derivada parcial [18]

$$\frac{dv_x}{dt} \approx \frac{\partial v_x}{\partial t} \quad (3.36)$$

Por lo que la ecuación (3.35) está dada por

$$-\frac{\partial p}{\partial x} = \rho \frac{\partial v_x}{\partial t} \quad (3.37)$$

Derivando parcialmente la ecuación (3.37) y sabiendo que $\frac{\partial^2 f}{\partial y \partial x} = \frac{\partial^2 f}{\partial x \partial y}$ para las derivadas parciales mixtas [19] se obtiene

$$-\frac{\partial p}{\partial x} = \rho \frac{\partial v_x}{\partial t} \quad (3.38)$$

$$-\frac{\partial^2 p}{\partial x^2} = \rho \frac{\partial}{\partial t} \left(\frac{\partial v_x}{\partial x} \right) \quad (3.39)$$

Sustituyendo el término de la velocidad proveniente de (3.27) en la ecuación anterior se tiene

$$-\frac{\partial^2 p}{\partial x^2} = \rho \frac{\partial}{\partial t} \left(-\frac{1}{K} \frac{\partial p}{\partial t} \right) \quad (3.40)$$

reordenando

$$\frac{\partial^2 p}{\partial x^2} = \frac{\rho}{K} \frac{\partial^2 p}{\partial t^2} \quad (3.41)$$

Utilizando la ecuación de Newton-Laplace para la velocidad del sonido [20] $c = \sqrt{K}/\rho$

$$\frac{\partial^2 p}{\partial x^2} = \frac{1}{c^2} \frac{\partial^2 p}{\partial t^2} \quad (3.42)$$

Esta ecuación es la ecuación de onda acústica en una dimensión; se observa que es muy similar a la ecuación de onda para ondas transversales obtenida en la sección 3.1.1.

La *Ecuación* (3.42) se puede extender a 3 dimensiones. En coordenadas cartesianas resulta en [21]

$$\nabla^2 p = \frac{1}{c^2} \frac{\partial^2 p}{\partial t^2} \quad (3.43)$$

Las ondas sonoras más simples son aquellas que se desplazan en una sola dirección y cuya función de onda es de tipo senoidal como la vista en la sección 3.1.1

$$p(x, t) = P_a \cos(kx - \omega t) \quad (3.44)$$

Donde P_a es la amplitud en la forma de onda descrita a través de la presión $P_a = p_{max}$.

Las funciones de onda como (3.44) son solución a la ecuación de onda sonora (3.42).

3.1.3. Intensidad y potencia sonoras

Las ondas sonoras y todas las ondas viajeras, portan energía de una región a otra. La energía transportada por un sonido se puede describir a través de la intensidad de la onda I , que es *igual a la rapidez media con que la onda transporta energía, por unidad de área, a través de una superficie perpendicular a la dirección de propagación* [11]. Es decir, es la potencia promedio de la onda por unidad de área (perpendicular a la dirección de propagación de la onda). Las maneras más comunes de expresar la intensidad de una onda sonora es en términos de la amplitud de desplazamiento A o de la amplitud de presión.

La porción de fluido por la que viaja una onda acústica ejerce una fuerza sobre el fluido delante de esta porción. La potencia de una onda se puede expresar como el producto de esa fuerza por la velocidad de las partículas $P = Fv$. Entonces, la potencia por unidad de área $P/A = Fv/A$ en esta onda sonora es igual al producto de p (fuerza por unidad de área) por la velocidad v de la partícula, es decir

$$P/A = pv \quad (3.45)$$

La intensidad es el promedio temporal en un periodo del producto pv

$$I = \langle pv \rangle_T = \frac{1}{T} \int_0^T (pv) dt \quad (3.46)$$

Para ondas senoidales que viajan en una sola dirección como la descrita en las secciones anteriores

$$I = \frac{P_a^2}{2\rho v} \quad (3.47)$$

Donde P_a es la amplitud en la forma de onda descrita a través de la presión $P_a = p_{max}$. La intensidad se expresa en unidades de $[W/m^2]$.

El oído humano es sensible a un amplio intervalo de intensidades de sonido, por lo que usualmente la intensidad se expresa en una escala logarítmica. Este tipo de escalas permiten desplegar, medir y visualizar grandes intervalos de valores de forma conveniente para el observador.

La escala logarítmica de intensidades de sonido denomina escala de nivel de sonido, donde el nivel de sonido se define como:

$$\beta = 10 \log \frac{I}{I_0} [dB] \quad (3.48)$$

I_0 es una intensidad que se toma como referencia y su valor es de $I_0 = 10^{-12} [W/m^2]$ [11]. Este valor corresponde al umbral de intensidad mínima que un humano puede oír considerando ondas con frecuencias medias de alrededor de 1kHz. La unidad que se utiliza para medir la intensidad del sonido es el dB. Los sonidos de intensidad $I = I_0$ tienen un nivel de sonido de 0 dB, mientras que los sonidos de intensidad $I = 1 [W/m^2]$ tienen un nivel de sonido de 120 dB, el cual es el llamado umbral del dolor.

3.2. Procesamiento de audio y voz

3.2.1. Señales de audio y voz

Las señales de audio usualmente son capturadas por micrófonos, éstos convierten los sonidos capturados en señales eléctricas con formas de onda variantes en el tiempo, que dependen de las variaciones de presión en el campo de sonido dentro del que está el microfono. Las señales de audio digitales provienen del muestreo y cuantización de una señal de audio analógica (como la producida por un microfono), utilizando un convertidor analógico digital (ADC).

Las señales de audio presentan formas de onda muy variadas, estas formas de onda pueden ser periódicas o aperiódicas, siendo la forma de onda senoidal la más simple. Las señales de audio periódicas están formadas o se pueden aproximar mediante sumas o conjuntos de señales senoidales, en los cuales existe una componente de frecuencia fundamental y un conjunto de armónicas múltiplos de la componente fundamental. Las amplitudes y fases de las componentes armónicas determinaran el timbre del sonido. Por otro lado, las señales no periódicas están conformadas por tonos senoidales no relacionados armónicamente [22].

La voz se caracteriza por tener regiones tonales y de ruido alternadas, las regiones tonales corresponden a segmentos vocales que ocurren en intervalos silábicos. Las señales de voz en un canal telefónico presentan componentes en frecuencia en el intervalo de 300 Hz a 3kHz, mientras que la música ocupa la mayoría del intervalo audible.

Las señales de audio provienen de procesos no-estacionarios por naturaleza. Es decir, no se pueden representar mediante sumas discretas de señales senoidales armónicas (series de Fourier), su varianza no es independiente del tiempo o su función de autocorrelación no depende unicamente de los desplazamientos en el tiempo. Es decir, son aleatorias y su distribución de probabilidad asociada y propiedades estadísticas varían al ser desplazadas en el tiempo.

El análisis de las señales de audio se simplifica si se supone que las propiedades de la señal cambian lentamente con el tiempo. Para el análisis de la señal se extraen o estiman parámetros o características de la misma, estos parámetros se obtienen a partir del análisis en tiempo corto y se conocen como parámetros de tiempo corto. Los parámetros obtenidos de la señal se relacionan a un modelo de la misma. La señal de voz se aproxima comúnmente por el modelo de fuente-filtro (source-filter) para la producción de voz. Este modelo también sirve para algunos instrumentos musicales, aunque la música usualmente se modela utilizando características de la señal provenientes de la suma de componentes elementales que la conforman [22].

3.2.2. Sonidos voceados, no voceados y ruido

Los métodos de detección de actividad acústica o voz se basan en la distinción entre la señal de audio o voz que se desea detectar y el ruido; usualmente el ruido y la señal están mezclados pero cada uno tiene características diferentes que permiten distinguirlos. Los sonidos contenidos en una señal de voz pueden clasificarse o distinguirse como voceados y no voceados [23], [24].

Sonidos voceados

Los sonidos voceados se producen cuando las cuerdas vocales vibran al pronunciar un fonema, es decir, son oscilatorios y se pueden llegar a representar mediante series de Fourier. Los sonidos voceados tienden a ser más "fuertes" y su energía se concentra en las bajas frecuencias del espectro en frecuencia. Las vocales son algunos sonidos de este tipo.

Sonidos no voceados

Los sonidos no voceados no implican el uso de las cuerdas vocales. Por ejemplo el sonido de la "s" o de la "f" [23]. Los sonidos no voceados tienden a ser más abruptos en su forma y no presentan características claras en el dominio del tiempo y la frecuencia. Usualmente se considera que se comportan como ruido blanco gaussiano en el dominio de la frecuencia y que tienen mayor energía en frecuencias mayores a los sonidos voceados.

Ruido

El ruido se define como cualquier señal indeseada superpuesta que interfiere con la reproducción fiel de la señal deseada en un sistema. Las fuentes de ruido dentro de sistemas que procesan señales de audio son muy variadas. Las más comunes son las presentes en el sistema acústico o en el sistema eléctrico utilizado para el procesamiento. El ruido acústico incluye sonidos ambientales indeseados como los producidos por vehículos, viento y cualquier dispositivo que genere vibraciones audibles. El ruido eléctrico se refiere a todas las señales eléctricas indeseadas superpuestas que siempre están presentes en los sistemas eléctricos.

La mayoría del ruido es aleatorio por naturaleza y se describe mejor a través de sus propiedades estadísticas y su espectro en frecuencia. El ruido térmico es la principal fuente de ruido en los sistemas eléctricos, sobre todo los relacionados a comunicaciones. Este ruido se puede describir como un proceso aleatorio Gaussiano con media cero. Se denomina Gaussiano porque se puede caracterizar estadísticamente a través de la función de densidad de probabilidad (pdf) Gaussiana.

$$p(n) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left[-\frac{1}{2}\frac{n^2}{\sigma^2}\right] \quad (3.49)$$

Donde n es la variable aleatoria y σ^2 es la varianza de n .

Ruido blanco Gaussiano aditivo. El ruido blanco Gaussiano, además de poder modelarse teóricamente a través de una pdf Gaussiana, se supone que tiene una densidad espectral de potencia "plana" para todas las frecuencias (tanto positivas como negativas). Es decir, tiene la misma potencia en todas las frecuencias [25]. Siempre que el ruido tiene una densidad espectral uniforme de esa manera, se le llama ruido blanco. El adjetivo "blanco" hace referencia a la luz blanca la cual radía de forma uniforme en todas las frecuencias del espectro visible.

3.2.3. Conversión analógica-digital

Los convertidores analógico-digital cambian o transforman una señal de voltaje en el tiempo continuo a una secuencia de valores de voltaje en el tiempo discreto. El proceso de conversión se puede dividir en tres etapas: muestreo, cuantización y codificación [26]. Las etapas de

muestreo y cuantización son independientes una de la otra y pueden realizarse por separado.

El muestreo es el proceso de conversión de una señal en el tiempo continuo a una señal en el tiempo discreto. Se obtiene tomando "muestras" de la señal continua en instantes discretos. El tiempo en el que se toma cada muestra se llama intervalo de muestreo T .

La cuantización es la conversión de las muestras de una señal discreta a valores continuos a una señal discreta con valores discretos (señal digital). El valor de cada muestra discreta de la señal se representa por un valor dentro de un conjunto finito de valores posibles.

La codificación es la representación de cada valor discreto mediante una secuencia binaria de b-bits [26].

Las etapas de la conversión se pueden estudiar por separado, pero en la práctica un mismo dispositivo convertidor toma la señal analógica y produce un número codificado en binario.

Muestreo y aliasing

El muestreo más común y sencillo es el muestreo periódico o uniforme y se describe mediante

$$x(n) = x_c(nT) \quad (3.50)$$

Donde $x(n)$ es la señal discreta obtenida al tomar muestras cada T segundos. Como se mencionó previamente, el intervalo de tiempo T entre cada muestra sucesiva se llama periodo de muestreo y su inverso $F_s = 1/T$ se llama frecuencia de muestreo (en Hz) o tasa de muestreo (muestras/s).

Existe una relación entre la frecuencia F de una señal analógica y la frecuencia f de una señal digital. Esta relación lineal es

$$f = \frac{F}{F_s} \quad (3.51)$$

y recordando que $\Omega = 2\pi F$ y $\omega = 2\pi f$

$$\omega = \Omega T \quad (3.52)$$

F_s es la frecuencia de muestreo o frecuencia de Nyquist, a partir de ésta y de la frecuencia de la señal digital f se puede conocer la frecuencia de una señal analógica F .

Las señales continuas y discretas presentan una diferencia fundamental en el dominio de la frecuencia. El intervalo de valores en frecuencia para una señal continua es infinito $-\infty < F < \infty$, mientras que el intervalo de valores en frecuencia para una señal discreta es $-1/2 < f < 1/2$ o $-\pi < \omega < \pi$. Esto implica trasladar un rango infinito de frecuencias F a un intervalo finito para la variable f . Ya que la máxima frecuencia en una señal discreta es $\omega = \pi$ o $f = 1/2$ entonces la frecuencia máxima de la señal analógica para una frecuencia de muestreo F_s es

$$F_{max} = \frac{F_s}{2} \quad (3.53)$$

o

$$\Omega_{max} = \pi F_s \quad (3.54)$$

El trabajar con señales analógicas de frecuencias mayores a F_{max} , produce un efecto conocido como "aliasing". El aliasing genera un traslape o repetición de componentes frecuenciales de la señal digitalizada. Conforme se incrementa la frecuencia de la señal analógica, manteniendo una frecuencia de muestreo constante F_s , el espectro de la señal discreta se va desplazando y debido a que el espectro de este tipo de señales es periódico, las componentes comienzan a traslaparse. Las señales con frecuencias $F = F_0 + kF_s$, donde k es un número entero, generan componentes indistinguibles de F_0 y se vuelven "alias" de la misma. Es decir, el aliasing causa que señales continuas distintas se vuelvan indistinguibles al muestrearse.

Elección de la frecuencia de muestreo

La elección de la frecuencia de muestreo dependerá de las características de la señal analógica que se desea muestrear, en especial de su contenido en frecuencia. Si conocemos el contenido máximo en frecuencia, se puede especificar la frecuencia de muestreo necesaria para convertir la señal. Para algunas señales, como la de voz o video, la máxima frecuencia puede variar ligeramente. Si se desea asegurar que F_{max} no exceda un valor predeterminado, se puede pasar la señal analógica a través de un filtro antialiasing que atenúe las componentes mayores a F_{max} .

F_{max} nos permite elegir la frecuencia de muestreo F_s adecuada. La máxima frecuencia en una señal analógica que puede muestrearse sin generar ambigüedades debidas al aliasing es $F_s/2$, por lo tanto

$$F_s > 2F_{max} \quad (3.55)$$

Esta condición asegura que todas las componentes en frecuencia de la señal analógica se representen correctamente y sin ambigüedades en el dominio de la frecuencia de la señal discreta, es decir, evita el aliasing. A la tasa de muestreo $F = 2F_{max}$ se le conoce como frecuencia de Nyquist.

La elección de la frecuencia de muestreo al trabajar con señales de audio y voz es variada. La voz telefónica usualmente se muestrea utilizando $F_s = 8kHz$, las señales de audio se muestrean utilizando $F_s = 44.1kHz$ para CDs o con $F_s = 48kHz$ para DVDs. Estas frecuencias están relacionadas a las señales de video y se remontan a los inicios del video digital [27].

Cuantización

La cuantización es el proceso de convertir una señal discreta de amplitud continua a una señal digital, expresando cada valor muestreado con un número de dígitos finito [26]. En la cuantización se expresan cantidades infinitamente variables mediante valores discretos o escalonados.

Las señales analógicas tienen valores de amplitud infinitamente variable, es decir, que se expresan mediante un número infinito de dígitos. Los sistemas digitales reales no pueden representar cantidades infinitamente variables, por lo que se realiza una cuantización al momento de representarlas. Los cuantizadores dividen el intervalo de valores de las señales en intervalos de cuantización Q o pasos (*steps*) S idénticos o diferentes. Si los intervalos son idénticos se le llama cuantización uniforme. El cuantizador limita el número de valores que pueden ser re-

presentados a un conjunto de valores discretos finitos, por lo que al representar los valores de una señal continuamente valuada se produce un error.

El error introducido al representar una señal continuamente valuada mediante un conjunto finito de valores discretos se llama error de cuantización o ruido de cuantización.

El error de cuantización es una secuencia e_q definida como la diferencia entre el valor cuantizado $x_q(n)$ y el valor de la muestra $x(n)$

$$e_q(n) = x_q(n) - x(n) \quad (3.56)$$

Los cuantizadores limitan el número de dígitos representados truncándolos o redondeándolos. Usualmente los cuantizadores utilizan el redondeo, el error de cuantización por redondeo está limitado a

$$\frac{-S}{2} \leq e_q(n) \leq \frac{S}{2} \quad (3.57)$$

S es el paso de cuantización. Si x_{max} y x_{min} son los valores máximos y mínimos de $x(n)$ y L el número de niveles de cuantización

$$S = \frac{x_{max} - x_{min}}{L - 1} \quad (3.58)$$

A la diferencia $x_{max} - x_{min}$ se le llama intervalo dinámico de la señal.

La cuantización de las señales analógicas resulta en pérdida de información, resultado de la ambigüedad introducida por el cuantizador. Esta ambigüedad hace que el análisis de la cuantización se complique demasiado y que se use análisis estadístico para su estudio. En la práctica, el error de cuantización se reduce de forma significativa si se eligen suficientes niveles de cuantización.

Codificación

La codificación asigna un número binario único a cada nivel de cuantización. Para L niveles de cuantización se necesitan mínimo L números binarios diferentes para representarlos. Con b bits se pueden crear 2^b números binarios, entonces $2^b \geq L$. A partir del número de niveles de cuantización que se requieran se puede determinar la cantidad de bits del codificador binario utilizando la ecuación

$$b \geq \log_2(L) \quad (3.59)$$

Usualmente el costo de los convertidores comerciales aumenta conforme incrementa el número de bits usados para la codificación y la tasa de muestreo.

3.2.4. Análisis en tiempo corto

Las señales de audio y voz son procesos aleatorios no estacionarios cuyas características cambian en el tiempo, pero lo hacen de forma "lenta" [24]. Por esta razón, si se analizan en periodos cortos de tiempo, se puede considerar que sus características casi no presentan cambios en cada periodo corto. Muchos de los métodos de análisis para este tipo de señales se basan en la

extracción de características de la señal discretizada dividida en "bloques"(frames) de corta duración [28].

Para evitar los efectos ocasionados por el truncamiento de la señal y la discontinuidad presentada al dividir la señal en bloques, se puede multiplicar el bloque escogido por una ventana. La ventana utilizada $w(n)$ multiplica a la secuencia de audio $x(n)$ para formar la secuencia $x_w(m)$ y existen diversas ventanas a utilizar. Esto lleva al principio básico del análisis en tiempo corto, el cual es [28]

$$X_{\hat{n}} = \sum_{-\infty}^{\infty} T[x(m)w(\hat{n} - m)] \quad (3.60)$$

$X_{\hat{n}}$ representa el parámetro o vector de parámetros a analizar en el tiempo \hat{n} . El operador $T[\]$ define la naturaleza de la función de análisis en el tiempo corto y $w(\hat{n} - m)$ representa la secuencia de la ventana desplazada en el tiempo. El producto dentro del operador $T[\]$ es la secuencia $x_w(m)$ mencionada previamente, es decir

$$x_w(m) = x(m)w(n - m) \quad (3.61)$$

Las ventanas utilizadas en el análisis de tiempo corto son variadas, algunas de las ventanas comunmente utilizadas son las siguientes.

Ventana rectangular:

$$w(n) = \begin{cases} 1, & 0 \leq n < N - 1 \\ 0, & \text{otro } n \end{cases} \quad (3.62)$$

Ventana de Hamming:

$$w(n) = \begin{cases} 0.54 - 0.46 \cos\left(\frac{2n\pi}{N-1}\right), & 0 \leq n < N - 1 \\ 0, & \text{otro } n \end{cases} \quad (3.63)$$

Ventana de Hanning:

$$w(n) = \begin{cases} 0.5 \left[1 - \cos\left(\frac{2n\pi}{N-1}\right)\right], & 0 \leq n < N - 1 \\ 0, & \text{otro } n \end{cases} \quad (3.64)$$

La mayoría de los métodos descritos a continuación funcionan en tiempo corto y utilizando ventanas rectangulares.

3.3. Métodos de detección de actividad acústica y voz (VAD)

La detección de actividad acústica y de voz se ha vuelto parte importante en el procesamiento digital de señales pues es utilizado en una gran cantidad de sistemas electrónicos actuales. Usualmente es la primera etapa dentro de un procesamiento subsecuente y su desempeño afecta a las etapas siguientes.

Los algoritmos de detección de actividad acústica y de voz permiten diferenciar entre el ruido

y la señal que se desea detectar de forma automática. El desempeño del algoritmo depende de que tan precisa sea la detección para diversas condiciones de ruido ambiental.

3.3.1. Métodos basados en energía

El cálculo de la energía en tiempo corto es un parámetro que se puede utilizar para diferenciar entre el ruido y la señal acústica que se desea detectar. En estos métodos, la señal de entrada se divide en bloques x_n , si la energía del bloque actual E_n es mayor a un umbral establecido Trh , se decide que se detectó presencia de la señal deseada (VAD=1). Si la energía del cuadro es menor al umbral, el bloque se clasifica sin presencia de actividad de la señal (VAD=0).

La energía en tiempo corto de un bloque, E_n , es igual a la suma del cuadrado de cada muestra dentro del segmento en tiempo corto ("bloque"). Es decir, se calcula de acuerdo a la ecuación (3.65)

$$E_n = \sum_{m=0}^{N-1} x_n(m)^2 \quad (3.65)$$

$x_n(m)$ es el segmento de la señal de audio sobre el que se realiza el cálculo de la energía y está previamente multiplicado por alguna de las ventanas mencionadas en la sección anterior.

$$x_n(m) = x(m)w(n-m) \quad 0 < m < N-1 \quad (3.66)$$

En las expresiones (3.66) y (3.65) $n = 0, 1T, 2T..$, N es la longitud del bloque y T es el desplazamiento del bloque. Además $w(-m) = 0$.

Otra manera de considerar las expresiones anteriores es como se muestra a continuación [29]

$$E_n = \sum_{m=n-N+1}^n [x(m)w(n-m)]^2 \quad (3.67)$$

de la misma manera $n = 0, 1T, 2T..$, N es la longitud del bloque y T es el desplazamiento del bloque.

La decisión sobre la presencia de actividad acústica se toma en base a la comparación de la energía calculada de un bloque E_n con un umbral Trh establecido o propuesto al inicio del método. El umbral puede fijarse en un inicio y mantenerse constante durante todo el proceso de detección o ser adaptable como se muestra en seguida.

Determinación del umbral para la detección de actividad

El valor escogido como umbral es un factor importante que afectará el desempeño del método para diferentes condiciones del entorno. Para la determinación del umbral se consideran las condiciones del ruido en el lugar donde se realiza la detección de actividad acústica; cuando existe actividad acústica, la energía calculada contendrá la suma de la energía del ruido y de la señal, por lo tanto es importante conocer la energía del ruido ambiental para establecer un

umbral inicial.

El valor del umbral inicial se puede establecer de forma arbitraria, pero puede resultar en un detector muy sensible o con desempeño pobre. Por esta razón se propone el siguiente método para establecer el umbral inicial [30], [31], [21]. Se asume que durante un pequeño periodo inicial de tiempo, la señal de audio solo contiene ruido ambiental. El umbral inicial se puede calcular a partir de la energía promedio de los primeros v bloques en los que se divide la señal. Esta energía promedio de los v bloques iniciales se calcula mediante la expresión:

$$E_r = \frac{1}{v} \sum_{i=0}^v E_i \quad (3.68)$$

v es el número de bloques iniciales y E_i se obtiene mediante la Ecuación (3.65) o (3.67). El cálculo de la energía también se puede realizar considerando el valor promedio, valores normalizados o RMS. En [30] y [31] utilizan para su cálculo el valor promedio.

$$E_j = \frac{1}{N} \sum_{i=(j-1)N+1}^{jN} [x(i)]^2 \quad (3.69)$$

Donde j es el j -ésimo bloque bajo consideración, E_j es la energía del cuadro y N la longitud del mismo.

A partir del cálculo de la energía promedio del ruido E_r se puede establecer el umbral de decisión mediante

$$Trh = E_r + k_d E_r \quad (3.70)$$

En la Ecuación (3.70) k_d es un factor que define una tolerancia de la energía para realizar la comparación. Entre mayor sea el valor k_d , la señal de voz deberá contener mayor energía para superar el umbral.

Umbral adaptable

El ruido presente en la señal de audio puede ser no estacionario, por lo que se requiere que el valor del umbral de energía se actualice o adapte de manera proporcional a los cambios en el ruido. Este umbral adaptable se calcula de forma iterativa a partir de valores anteriores del umbral y de las variaciones en el ruido. A continuación se presentarán tres tipos de detectores de actividad con umbral adaptable [30].

Detector lineal basado en energía (LED I)

El umbral se establece mediante la regla

$$E_{rnew} = (1 - p)E_{rold} + pE_{sil} \quad (3.71)$$

E_{rnew} es el valor actualizado del umbral de decisión. E_{rold} es el valor del último umbral establecido, E_{sil} es el valor de la energía para el último cuadro (puede ser el cuadro presente) sin actividad; p es un parámetro entre 0 y 1 ($0 < p < 1$) que se escoge considerando la respuesta

al impulso de (3.71) como un filtro de primer orden [30].

Detector lineal basado en energía con doble umbral (LED II)

El detector anterior calculaba el umbral solamente en los periodos en los que no se detectaba actividad. Este detector establece dos umbrales de forma separada, uno cuando hay actividad acústica o de voz y otro cuando no la hay. El detector funciona como se describe a continuación. El nivel o la energía del ruido se establece mediante las expresiones (3.72) y (3.73)

$$E_{rnew} = \lambda_1 E_{rold} + (1 - \lambda_1) E_j \quad (3.72)$$

$$E_{rnew} = \lambda_2 E_{rold} + (1 - \lambda_2) E_j \quad (3.73)$$

La (3.72) se utiliza en los segmentos con actividad y (3.73) en los segmentos sin actividad. λ_1 y λ_2 son los factores de adaptación y están en los intervalos $0.85 \geq \lambda_1 \geq 0.95$ y $0.98 \geq \lambda_2 \geq 0.999$, estos factores definen filtros paso-bajas [30]. El valor de λ_1 debe ser lo suficiente pequeño para seguir a la variación del ruido, pero mayor a la variación de la señal de audio. λ_2 debe ser lo suficientemente grande para evitar que siga a la variación de la energía de la señal de audio, pero lo suficiente pequeña para adaptarse a variaciones en el ruido de fondo.

Los umbrales para la señal de audio a detectar y el ruido se definen como

$$trh_{silnew} = E_{rnew} + \delta_{sil} \quad (3.74)$$

$$trh_{actnew} = E_{rnew} + \delta_{act} \quad (3.75)$$

Donde $0.1 \leq \delta_{sil} \leq 0.4$ y $0.5 \leq \delta_{act} \leq 0.8$. Estas son constantes usadas para determinar los umbrales. Cuando la energía es mayor al umbral de detección trh_{actnew} , se detecta actividad acústica y cuando el umbral es menor al umbral de ruido trh_{silnew} no se detecta actividad.

Detector lineal adaptable basado en energía (ALED)

El ALED es una mejora del detector lineal basado en energía (LED I). En este último, el coeficiente p se mantiene constante volviéndose insensible a modificaciones estadísticas del ruido. Para compensar esto se calcula el umbral de energía usando estadísticas de los bloques sin actividad acústica.

El método utiliza un buffer de m bloques sin actividad. Cuando se detecta un nuevo bloque sin actividad, se añade al buffer y el más antiguo se remueve. Este método calcula de forma constante la varianza del buffer en términos de la energía

$$\sigma^2 = Var[E_{sil}] \quad (3.76)$$

Los cambios en el ruido de fondo se detectan comparando las variaciones que produce la energía del bloque inactivo actual con las medidas estadísticas de los m bloques inactivos anteriores. Antes de añadir un nuevo cuadro inactivo al buffer, la varianza del buffer está dada por σ_{old}^2 . Una vez que se añade el nuevo bloque y se recalcula la varianza, esta tiene un valor de σ_{new}^2 . Un cambio abrupto en el ruido de fondo significa que

$$\sigma_{new}^2 > \sigma_{old}^2 \quad (3.77)$$

A partir de esto se puede ajustar el valor de p de acuerdo al Cuadro 3.1 [30].

$\sigma_{new}^2/\sigma_{old}^2$	p
$1.25 \leq \sigma_{new}^2/\sigma_{old}^2$	0.25
$1.10 \leq \sigma_{new}^2/\sigma_{old}^2 < 1.25$	0.20
$1.0 \leq \sigma_{new}^2/\sigma_{old}^2 < 1.10$	0.15
$\sigma_{new}^2/\sigma_{old}^2 < 1.0$	0.10

Cuadro 3.1: Tabla de adaptación de p según la varianza

3.3.2. Tasa de cruces por cero

La tasa de cruces por cero (ZCR) es el número de veces que la señal de audio cruza el nivel cero del eje horizontal, es decir, el número de veces que las muestras sucesivas de la señal cambian de signo. También se considera a la tasa de cruces por cero como el número de veces que la muestra cambia los símbolos. La ZCR está definida matemáticamente como:

$$Z_n = \frac{1}{2} \sum_{m=0}^{N-1} |\text{sgn}(x(m)) - \text{sgn}(x(m-1))| \quad (3.78)$$

Donde

$$\text{sgn}(x) = \begin{cases} 1, & x \geq 0 \\ -1, & x < 0 \end{cases} \quad (3.79)$$

Este método se utiliza para detección de actividad de voz. Cuando existen sonidos voceados dentro de la señal, las componentes espectrales de la misma se concentran en regiones de baja frecuencia (<3KHz). Por otro lado, cuando existen sonidos no voceados en la señal, las componentes espectrales de mayor energía aparecen en zonas de alta frecuencia (>8kHz). Entonces, si la Z_n es alta, el cuadro se considera que no está voceado (no se detecta la presencia de voz). Al contrario, si la Z_n es baja, el cuadro esta voceado (se detecta la presencia de voz) [24].

3.3.3. Métodos basados en medidas estadísticas

Estos métodos incorporan estimadores y medidas estadísticas basadas en el espectro en frecuencia de la señal. Parten de la relación señal a ruido (SNR) y utilizan umbrales adaptables para la detección dentro de la señal de entrada. Un método de este tipo se describe en seguida y se encuentra en [31].

Se asume que la señal y el ruido son independientes y la señal es de la forma

$$x(n) = s(n) + v(n) \quad (3.80)$$

Donde $s(n)$ es la señal que se quiere detectar y $v(n)$ es el ruido aditivo en el k -ésimo cuadro.

La relación señal a ruido se define como:

$$SNR_k = \frac{P_{xx,k}(f_l)}{\widehat{P}_{vv,k}(f_l)} - 1 \quad (3.81)$$

$\widehat{P}_{vv,k}(f_l)$ es el valor esperado de la densidad espectral (PSD) del ruido. $P_{xx,k}(f_l)$ es la densidad espectral del k-ésimo cuadro actual para la muestra espectral f_l .

$$\widehat{P}_{vv,k}(f_l) = \frac{1}{K} \sum_{k=0}^{K-1} P_{xx,k}(f_l) \quad (3.82)$$

La Ecuación (3.82) brinda el valor esperado de la densidad espectral promedio calculada sobre un periodo inicial de tiempo en el que no hay actividad de la señal deseada. Este periodo inicial de tiempo se divide en k bloques.

Para el periodo inicial de tiempo cuando se considera que no hay actividad $P_{xx,k}(f_l) = P_{vv,k}(f_l)$ y la relación señal a ruido está dada por

$$SNR_k = \frac{P_{vv,k}(f_l)}{\widehat{P}_{vv,k}(f_l)} - 1 \quad (3.83)$$

Para la toma de la decisión se establecen dos hipótesis y un umbral. Estas hipótesis representan los casos donde hay actividad H_1 y donde no hay actividad H_0 . Estas hipótesis se representan mediante las siguientes ecuaciones:

$$H_0 : SNR_k(f_l) = \frac{P_{vv,k}(f_l)}{\widehat{P}_{vv,k}(f_l)} - 1 \quad (3.84)$$

$$H_1 : SNR_k(f_l) = \frac{P_{vv,k}(f_l) + P_{ss,k}(f_l)}{\widehat{P}_{vv,k}(f_l)} - 1 \quad (3.85)$$

$P_{ss,k}$ es la densidad espectral estimada para la señal en la muestra espectral f_l . El umbral de decisión está dado por

$$\eta'_k(f_l) = \sqrt{2\sigma_{v,k}^2(f_l)} \cdot \text{erfc}^{-1}(2P_{FA}) \quad (3.86)$$

$\sigma_{v,k}^2$ es la varianza de la relación señal a ruido medida durante el periodo cuando no hay actividad $\sigma_{v,k}^2 = E[SNR_k(f_l)^2]$; erfc^{-1} es la función de error complementaria y P_{FA} es la probabilidad de falsa alarma.

Los cuadros en los que está dividida la señal se clasifican como con actividad acustica o de voz o sin actividad basandose en la comparación de la SNR promedio y el umbral de decisión promedio.

Si la SNR promedio es mayor o igual al umbral, se decide H_1 , es decir existe actividad de la señal.

$$\frac{1}{L} \sum_{f_l=0}^{L-1} SNR_k(f_l) > \frac{1}{L} \sum_{f_l=0}^{L-1} \eta'_k(f_l) \rightarrow H_1 \quad (3.87)$$

Si no, H_0 se decide y no hay actividad de la señal.

$$\frac{1}{L} \sum_{f_l=0}^{L-1} SNR_k(f_l) < \frac{1}{L} \sum_{f_l=0}^{L-1} \eta'_k(f_l) \rightarrow H_0 \quad (3.88)$$

3.3.4. Métodos basados en reconocimiento de patrones

Estos métodos aplican el concepto de reconocimiento de patrones para clasificar una señal de voz dada en dos clases voceada o no voceada. Estos métodos extraen características de señal y obtienen estadísticas de las mismas para tomar la decisión, las características medidas o extraídas son la tasa de cruces por cero, la energía, la autocorrelación entre muestras sucesivas, el primer coeficiente predictor proveniente de la codificación lineal predictiva (LPC) y la energía en el error de predicción.

La clasificación de la señal en voceada o no voceada se logra calculando la distancia euclideana con los parámetros extraídos de los segmentos de la señal y asignado cada segmento a la clase con menor distancia. En este método, los parámetros extraídos de segmentos de la señal se almacenan en un vector de parámetros. Posteriormente se calcula el vector de promedios y la matriz de covarianza de los vectores de parámetros. Con el vector y la matriz se calculan las distancias para cada segmento y se toma la decisión sobre la clasificación.

3.3.5. Métodos basados en análisis espectral

La detección de actividad de voz se puede hacer en el dominio de la frecuencia. Existen múltiples métodos utilizados en el dominio de la frecuencia para realizar la detección de actividad acústica. Para la realización de estos métodos se hace uso de la transformada rápida de Fourier (FFT) y la transformada corta de Fourier (STFT).

La transformada corta de Fourier es la transformada de Fourier en el tiempo discreto (DTFT) del segmento de la señal multiplicado por una ventana que se va deslizando $x_n(m) = x(m)w(n-m)$.

Una vez que se obtiene el espectro en frecuencia de la señal, se obtienen parámetros del mismo y con base en estos se toma la decisión de la detección de actividad. En el dominio de la frecuencia también existen los métodos basados en energía y son muy similares a los descritos en el dominio del tiempo. También existen métodos que utilizan características del espectrograma de la señal para la decisión.

3.4. Resumen

En el capítulo se abordaron los fundamentos, principios y métodos básicos utilizados para la detección de actividad acústica en señales de audio discretas. Al inicio del capítulo se realizó una reseña teórica sobre acústica que ayuda a comprender la generación y descripción de señales audibles. Posteriormente se abordaron los fundamentos para discretizar y procesar señales de audio provenientes de micrófonos y finalmente se describieron diversos métodos de detección de actividad acústica para señales de audio digitales.

Capítulo 4

Diseño e implementación del sistema

Los sistemas de vigilancia están conformados por diversos subsistemas y elementos que realizan uno o varios procesos dentro del sistema. Estos subsistemas pueden tener componentes físicos (hardware) y/o programas (software) que se utilizan para llevar a cabo diferentes tareas.

El sistema de vigilancia que se muestra en el presente trabajo cuenta con varios subsistemas que se describen en este capítulo. Se presenta el diseño de cada subsistema y los componentes y algoritmos utilizados en ellos.

Inicialmente se realiza una descripción general del sistema, su funcionamiento, el conjunto de subsistemas que lo forman y como están conectados entre sí. Posteriormente se muestra el diseño de cada subsistema en específico, el hardware y software asociado a ellos y la forma en que operan.

4.1. Requerimientos generales del sistema y objetivos académicos

El sistema de vigilancia debe de ser capaz de detectar de forma correcta, automática y en tiempo real la actividad acústica, además debe trabajar en diferentes entornos y condiciones de ruido ambiental. También debe poder realizar el movimiento de una cámara de vigilancia y permitir visualizar el video proveniente de la misma al momento en el que se realiza la detección. Los requerimientos específicos de cada etapa del sistema y diferentes pruebas y condiciones en las que opera se muestran en las secciones y capítulos siguientes.

La selección y uso del procesador digital de señales (DSP) TMS320F28377S se planteó como objetivo académico al inicio de la realización del presente trabajo, es decir, se propuso evaluar el desempeño y la viabilidad de uso de ese DSP en la aplicación que se muestra.

4.2. Descripción general

El sistema de vigilancia que se implementó consta de cuatro etapas que permiten su funcionamiento. Estas etapas son la adquisición de la señal de audio, la detección de actividad acústica, el control del movimiento del servomotor al cual se encuentra acoplado la cámara de vigilancia y la visualización del vídeo producido por la cámara.

La *Figura 4.1* muestra un diagrama de bloques con las etapas mencionadas anteriormente y los subsistemas pertenecientes a cada etapa. La etapa de adquisición de señal detecta las señales acústicas presentes en el ambiente en el que se encuentra el sistema, las convierte en una señal eléctrica analógica y posteriormente en una señal digital mediante el convertidor analógico-digital(A/D). La etapa de detección de actividad acústica toma la señal discreta proveniente del convertidor analógico-digital y decide si hay actividad acústica presente o solamente ruido. El funcionamiento detallado y el algoritmo para la detección de actividad acústica se presentan en una sección posterior de este capítulo. El servomotor encargado de mover la cámara se controla mediante una señal modulada por ancho de pulso (PWM) generada por software. Finalmente, la cámara produce una señal de video que se visualiza en el monitor LCD.

La etapa de adquisición de la señal requería que se tuviera una respuesta en frecuencia adecuada en el espectro audible, es decir, que se pudiera digitalizar la señal acústica sin sufrir distorsión, considerando componentes de costo reducido. La señal de entrada para esta etapa es la señal acústica presente en el ambiente y la salida es la señal discreta dada por el convertidor analógico-digital. Los valores de voltaje de operación y el diseño de los circuitos se hicieron en base a las limitaciones establecidas por el DSP elegido.

Para la etapa de detección de actividad acústica se requería que se realizara el procesamiento y la toma de decisión en un tiempo menor al que se toma cada muestra, con el fin de evitar pérdidas de información y decisiones erróneas. Como entrada de ésta etapa se tiene la señal de audio discreta proveniente del ADC y almacenada en un buffer de memoria y como salida la señal de decisión, esta es una señal digital con niveles lógicos de voltaje.

El control del servomotor debía ser capaz de mover al motor con una cámara de vigilancia de tamaño reducido en un intervalo de posiciones angulares de 180° . La acción de esta etapa debía de depender de la decisión tomada en la etapa anterior por lo cual se tenía como entrada la señal de decisión y como salida la posición angular de la cámara.

La etapa de visualización del video debía de mostrar el video proveniente de la cámara en tiempo real a través de un monitor. Como se mencionó previamente, la adquisición y procesamiento del video van más allá del alcance de este trabajo.

La *Figura 4.2* presenta el hardware utilizado en la realización del sistema. Se visualiza el hardware asociado a los subsistemas y donde se realiza cada tarea del sistema.

El circuito acondicionador de señal se encarga de adaptar la señal producida en el micrófono para que el A/D pueda convertirla. El convertidor analógico-digital y el generador de la señal de PWM son periféricos incorporados en el procesador digital de señales (DSP) TMS320F28377S de Texas Instruments TM [12], [32], [33]. El TMS320F28377S está montado sobre la tarjeta

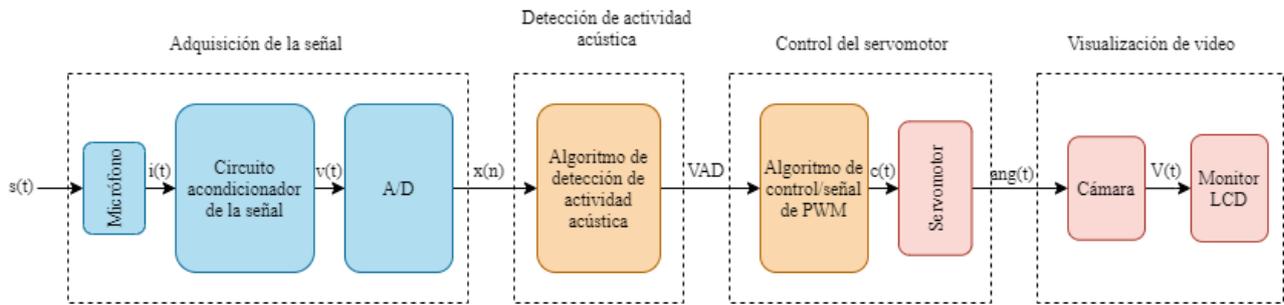


Figura 4.1: Diagrama de bloques del sistema

de desarrollo LaunchXL-F28377s TM. Esta última a su vez está colocada en una tarjeta de expansión para aplicaciones de audio, desarrollada en el laboratorio de Procesamiento Digital de Señales de la Facultad de Ingeniería de la UNAM, la cual contiene el circuito acondicionador de la señal proveniente del micrófono como se ve en la *Figura 4.2*.

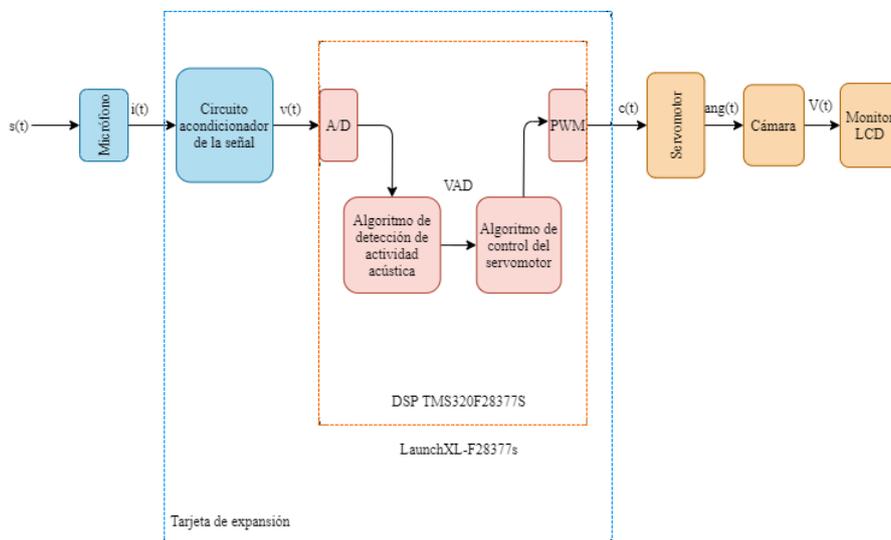


Figura 4.2: Hardware del sistema

4.3. Adquisición de señales

4.3.1. Micrófono

El sensado de las señales acústicas del ambiente se realizó mediante un micrófono *Electret*. Se eligió este micrófono debido a su reducido tamaño, bajo costo y respuesta en frecuencia.

Electret es un material dieléctrico delgado con presencia de carga eléctrica cuasi permanente en su superficie. El término cuasi permanente implica que la carga presente en el material decae muy lentamente conforme pasa el tiempo [34], razón por la cual usualmente se considera que la carga presente en ellos es fija.

En los micrófonos de este tipo, el electret se encuentra alojado entre dos electrodos y la estructura forma un capacitor. Cuando existen variaciones en la presión del aire (ondas sonoras) uno de los electrodos del capacitor se mueve hacia atrás y hacia adelante, modificando la distancia entre los electrodos y por ende la capacitancia del capacitor. Es decir, los micrófonos electret se pueden ver como capacitores variables dentro de un circuito eléctrico.

La señal de voltaje producida en este tipo de micrófonos es pequeña, razón por la cual debe amplificarse para poder ser utilizada por otros circuitos; además, los amplificadores en contacto directo con el micrófono deben tener una impedancia de entrada extremadamente alta. La mayoría de los micrófonos electret tienen un transistor de efecto de campo de unión (JFET) interno que sirve de buffer para la señal producida en el capacitor. La señal de voltaje producida por el sonido modifica el voltaje en la compuerta (gate) del JFET, ocasionando que cambie la corriente que fluye a través del drenaje (drain) y la fuente (source) del JFET [35]. La *Figura 4.3* muestra el circuito equivalente para el micrófono electret que se utilizó en este trabajo.

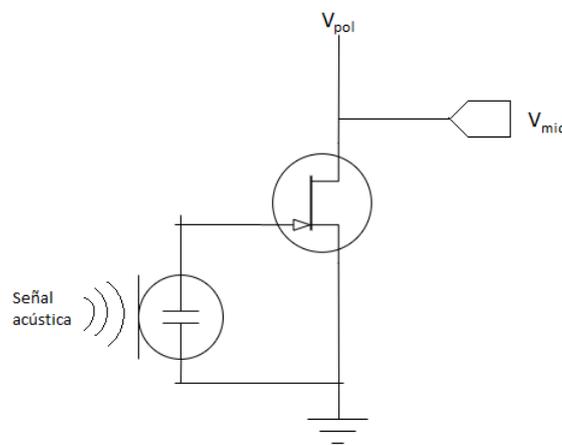


Figura 4.3: Microfono Electret

4.3.2. Circuito acondicionador de la señal

El circuito de acondicionamiento de la señal proveniente del micrófono es parte de la *Tarjeta de expansión* para la LaunchPad LAUNCHXL-F28377S. La tarjeta de expansión fue desarrollada en el Laboratorio de Procesamiento Digital de Señales de la UNAM con el fin de realizar aplicaciones de audio y voz [36].

La *Tarjeta de expansión* se polariza con voltajes mayores a 12V y menores a 35V. Esta tarjeta cuenta con un regulador de voltaje de 12V el cual alimenta al circuito de acondicionamiento utilizado en la realización de este trabajo.

El circuito de acondicionamiento consta de varias etapas que se describirán a continuación. A la entrada del circuito se tiene un *Jack* de 3.5mm para conectar el micrófono electret y la salida del circuito está conectada al ADC_A2 del TMS320F28377S. El circuito utilizado está conectado a la entrada señalada como *Micrófono uno* de la *Tarjeta de expansión*.

La primer etapa del circuito mencionado anteriormente es la etapa de preamplificación. Esta etapa se diseñó a partir de la nota de aplicación [35] de Texas Instruments. La *Figura 4.4* muestra el diagrama esquemático del preamplificador utilizado.

La manera en que opera el circuito se entiende considerando lo explicado a continuación. La corriente que circula hacia el micrófono i_{MIC} tiene una componente de corriente directa I_{DC} necesaria para polarizar el JFET y una componente de corriente alterna i_{ac} causada por el efecto de las ondas sonoras en el micrófono. Es decir, $i_{MIC} = I_{DC} + i_{ac}$. El capacitor C_2 forma un filtro paso altas en combinación con la resistencia R_1 y sirve para eliminar la componente de DC de la corriente i_{MIC} permitiendo que i_{ac} fluya hacia el amplificador operacional. Si la impedancia de C_2 es mucho menor que la de R_1 , entonces i_{ac} fluirá a través de C_2 . El amplificador operacional $U1A$ es un amplificador de transimpedancia, es decir, obtiene un voltaje a partir de la corriente que circula por el micrófono. Se puede obtener la impedancia equivalente de C_1 y R_2 resultando en $Z_{eq} = R_2/(1 + sR_2C_1)$. Haciendo esto y aplicando la ley de voltajes de Kirchoff (LVK), el voltaje de salida del amplificador está V_{OA} dado por:

$$V_{OA} = V_B + \frac{R_2}{1 + sR_2C_1}i_{ac} \quad (4.1)$$

Considerando $sR_2C_1 \ll 1$ entonces:

$$V_{OA} = V_B + R_2i_{ac} \quad (4.2)$$

Donde V_B es el voltaje de offset, dado por la red conectada a la terminal no inversora (+) de $U1A$.

Finalmente a la salida del amplificador se colocó un filtro paso altas para eliminar la componente de DC (offset) de la señal de salida. Se eligió este preamplificador por las razones mencionadas en [35].

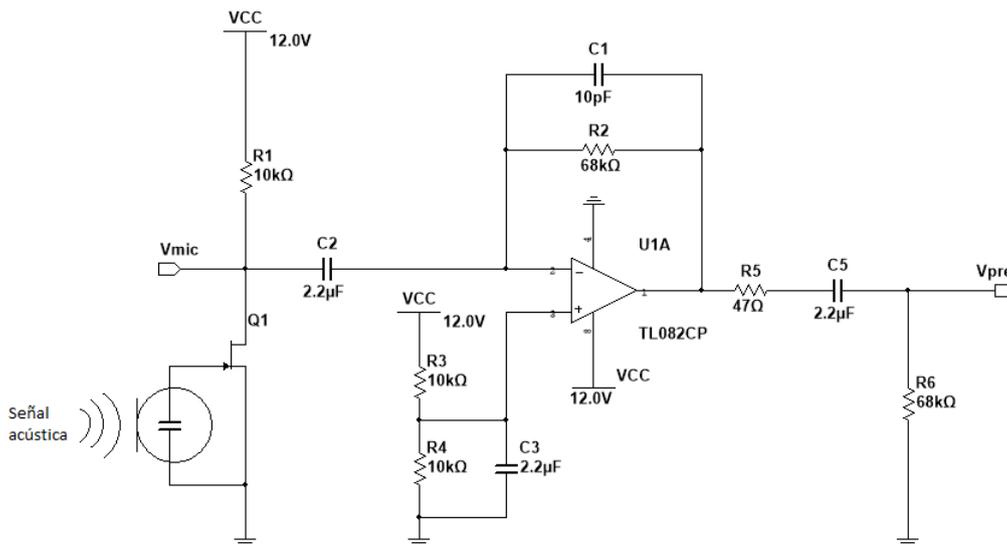


Figura 4.4: Etapa de preamplificación

El valor de cada componente del circuito se obtuvo como se muestra en seguida. La sensibilidad

del micrófono utilizado en amperes por Pascal de presión de aire, medida utilizando una impedancia de $2.2K\Omega$ es de $8.083 \mu A/Pa$. Considerando un nivel de sonido máximo de 100dB SPL correspondientes a una presión de 2 Pa, la corriente máxima de salida del micrófono es de:

$$i_{acm} = (8.083\mu A/Pa)(2Pa) = 16.166\mu A \quad (4.3)$$

La ganancia del amplificador de transimpedancia considerando un voltaje de salida típico de un nivel de línea de audio ($1.228 V_{RMS}$) está dada por R_2 . Esta resistencia se calculó mediante

$$R_2 = \frac{V_{OA}}{i_{acm}} = \frac{1.228V}{16.166\mu A} = 75.9619K\Omega \quad (4.4)$$

Para el circuito utilizado se ajustó el valor de la resistencia R_2 al valor comercial más cercano: $R_2 = 68K\Omega$, con lo que el voltaje de salida se modifica a $V_{OA} = 1.1V$.

El capacitor de realimentación C_1 compensa los efectos producidos por capacitancias parásitas en la terminal inversora del amplificador operacional y en combinación con la resistencia R_2 añade un polo para mejorar la respuesta en frecuencia del amplificador. La localización en frecuencia de este polo está dada por

$$f_p = \frac{1}{2\pi R_2 C_1} \quad (4.5)$$

Elijiendo $C_1 = 10pF$ entonces $f_p = 234.0514kHz$; la frecuencia donde se localiza el polo es suficientemente alta para no afectar la función de transferencia del amplificador en el ancho de banda audible, es decir, cumple con la condición $sR_2C_1 \ll 1$.

R_1 se eligió considerando el voltaje de la fuente, el voltaje de DC V_{mic} con el que opera el micrófono y limitando a que el consumo máximo de corriente de la fuente I_s fuera de 1mA. Aplicando ley de Ohm en R_1 .

$$R_1 = \frac{V_{CC} - V_{mic}}{I_s} = \frac{12 - 10V}{1mA} = 10K\Omega \quad (4.6)$$

R_1 y C_2 forman un filtro paso altas. Para evitar que componentes de corriente directa lleguen al amplificador se elige una frecuencia de corte lo suficientemente alta. En este caso se eligió $f_c = 7Hz$ y entonces:

$$C_2 = \frac{1}{2\pi R_1 f_c} = \frac{1}{2\pi(10K)(7)} = 2.2\mu F \quad (4.7)$$

Las resistencias R_3 y R_4 se eligieron para centrar la salida del amplificador a la mitad del voltaje de polarización y permitirle el máximo "swing" a la señal de audio. El voltaje en la terminal no inversora V_B se calcula utilizando un divisor de voltaje y entonces $R_3 = R_4$ para $V_B = V_{CC}/2 = 6V$. Se eligieron resistencias de 10 K Ω . C_3 forma un filtro paso bajas en combinación con R_3 y R_4 , este filtro elimina parte del ruido térmico introducido por las resistencias o la fuente de alimentación. La frecuencia de corte de este filtro es de 14 Hz, es decir, es menor al rango audible.

La resistencia R_5 limita la corriente a través del capacitor en caso de que un circuito con un

voltaje de DC diferente se conecte a la salida del preamplificador. Esta resistencia va de decenas de Ohms a varias centenas de Ohms, usualmente se elige de 50Ω . Se eligió una resistencia de 47Ω .

C_5 y R_6 forman un filtro paso altas cuya frecuencia de corte está dada por $f_c = 1/2\pi R_6 C_5$. R_6 se elige lo suficientemente grande para poder descargar C_5 rápidamente, en este caso $R_6 = 68k\Omega$. Considerando una $f_c = 1Hz$, entonces:

$$C_5 = \frac{1}{2\pi R_6 f_c} = \frac{1}{2\pi (68k)(1)} \approx 2.2\mu F \quad (4.8)$$

La segunda etapa del circuito de acondicionamiento de la señal es la etapa de amplificación. Esta etapa se muestra en la *Figura 4.5* y consta de un amplificador inversor cuyo voltaje de salida está dado por

$$V_{OD} = -\frac{R_8}{R_7} V_{pre} \quad (4.9)$$

La ganancia de este amplificador es fija y es de

$$G = -\frac{R_8}{R_7} = -10 \quad (4.10)$$

La última etapa del circuito consta de una red que se encarga de agregar un nivel de DC de $1.56V$ a la señal proveniente de las etapas anteriores. Esto se hace para que la señal solo tenga valores positivos de voltaje (en el intervalo de 0 a $3.3V$) y pueda ser convertida por el convertidor analógico-digital del DSP.

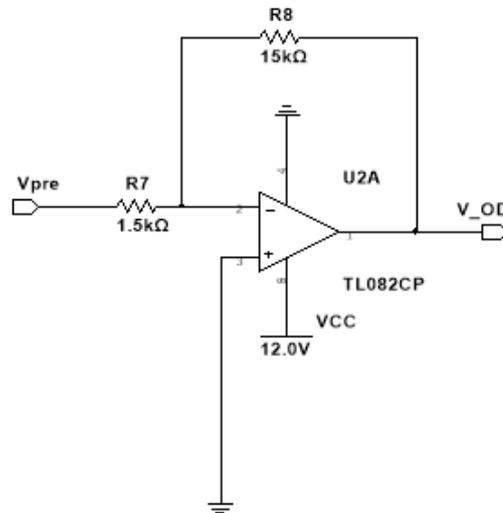


Figura 4.5: Etapa de amplificación

El nivel de DC está definido por las resistencias R_9 y R_{10} y se calcula aplicando un divisor de voltaje con esas resistencias. Esta etapa también forma un filtro paso altas que elimina componentes de DC provenientes de las etapas anteriores y su frecuencia de corte es de $f_c = 55.5Hz$. La *Figura 4.6* muestra el diagrama esquemático de esta etapa.

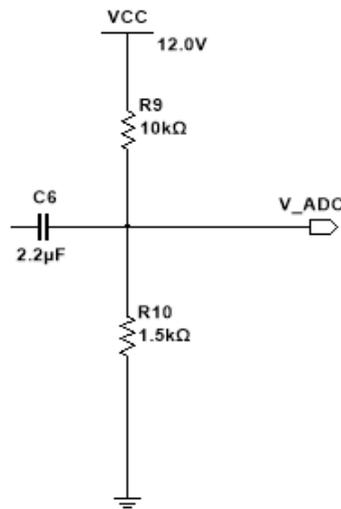


Figura 4.6: Nivel de DC

La Figura 4.7 muestra el diagrama esquemático del circuito de acondicionamiento completo.

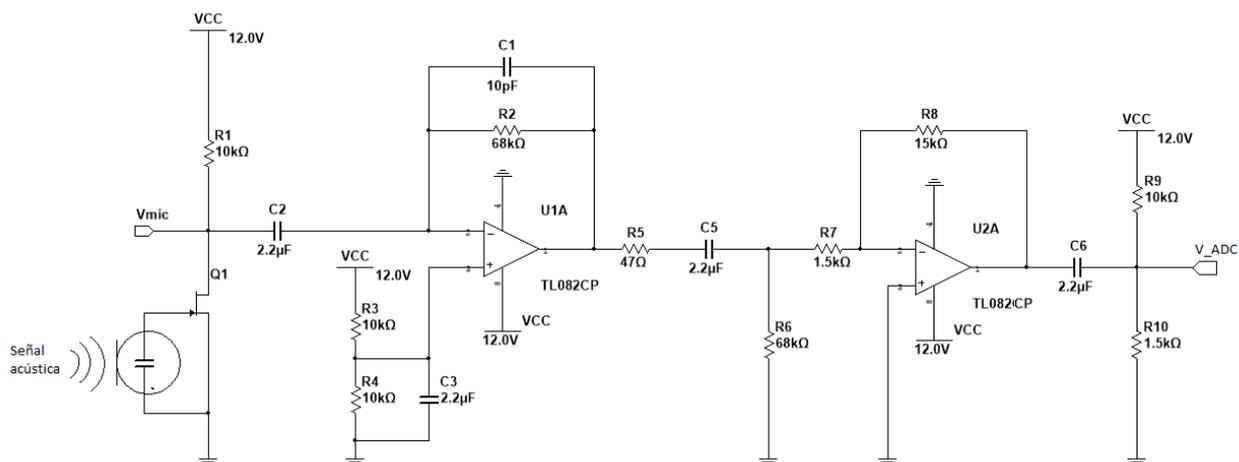


Figura 4.7: Circuito de acondicionamiento completo

4.3.3. Convertidor analógico-digital

El TMS320F28377S cuenta con cuatro convertidores analógico-digital en su arquitectura, pero dependiendo del empaquetado del CPU será el número de ADC que se puedan utilizar. En su empaquetado de 100 pines (100-pin PowerPAD Thermally Enhanced Thin Quad Flatpack (HTQFP)) [32] cuenta con dos convertidores analógico-digital utilizables. Estos convertidores son de aproximaciones sucesivas y cuentan con las siguientes características:

- Resolución elegible y modo de configuración para la señal. Se puede elegir una resolución de 12 o 16 bits para la conversión y el modo diferencial o el modo simple para la señal

de entrada. El modo diferencial resta las señales que llegan a dos pines de entrada del ADC y convierte la resta; el modo simple convierte la señal que llega a un solo pin del ADC. La resolución de 12 bits solo se puede utilizar para el modo simple y la de 16 bits solo para el modo diferencial.

- Referencia de voltaje configurable. Los ADCs cuentan con pines para fijar los voltajes de referencia alto (VREFHI) y bajo (VREFLO) utilizados para realizar la conversión. También permiten fijar estos voltajes usando referencias internas presentes en la tarjeta.
- Múltiples canales de conversión. Cuenta con 16 canales de conversión multiplexados (en modo simple) u ocho canales (en modo diferencial).
- Múltiples conversiones y registros de resultados. Cuenta con 16 registros para guardar los resultados de la conversión. Estos resultados provienen de 16 diferentes SOCs, los cuales se explicarán más adelante.
- Capacidad para generar interrupciones. Se puede generar software para realizar acciones cada vez que el ADC realiza una conversión.
- Bloques de post-procesamiento. Permiten calibrar el offset de la señal, corregir el error de punto de partida, detectar límites y cruces por cero y generar retrasos entre el muestreo y la conversión.
- Detección de corto circuito y circuito abierto.

El ADC está formado por una parte analógica y una parte digital. La parte analógica incluye el multiplexor del canal, el circuito de muestreo y retención (S/H), los circuitos para realizar las aproximaciones sucesivas y las referencias de voltaje. La parte digital configura y controla el ADC, esta parte se compone de los circuitos lógicos para configurar la conversión, los registros de resultados y las interfaces con otros periféricos y módulos del chip.

La configuración del ADC se realiza enteramente por software y cada conversión tiene asociado un inicio de conversión (SOC). Para cada SOC se configuran tres parámetros: el disparo del ADC, el canal a convertir y la ventana de adquisición deseada. Esta ventana de adquisición está relacionada al circuito analógico de muestreo y retención del ADC y se calcula de acuerdo a diferentes componentes del convertidor como el ADC seleccionado, el canal utilizado, la resistencia de salida del circuito conectado al ADC, la resistencia de entrada del ADC, entre otros.

La configuración del ADC utilizado para este trabajo se describe a continuación y el código mediante el cual se realizó dicha configuración se encuentra en la *sección de Anexos*.

La parte digital del ADC requiere de un pulso de reloj para llevar a cabo la conversión, se eligió una frecuencia de 42.2MHz para el pulso de reloj mencionado. Esta frecuencia es menor al límite indicado en la hoja de datos del DSP [32]. Se utilizó el ADC_ A, un solo canal en modo simple y por lo tanto una resolución de 12 bits.

Se decidió utilizar una frecuencia de muestreo de 48 kHz (estándar de audio) para muestrear la señal que se deseaba convertir. Para poder obtener esta frecuencia de muestreo y evitar

asincronías se configuró una señal de PWM a esa frecuencia con la cual empezaba cada conversión del ADC. Cada vez que ocurría un flanco de bajada en la señal de PWM iniciaba una conversión. Esta señal de PWM se puede ver en el GPIO2 del DSP.

El TMS320F28377S permite configurar hasta 16 SOC's, cada SOC tiene asociada cierta prioridad dada por un apuntador llamado *Round Robin Pointer*. Esto resulta de especial importancia cuando se presentan varios SOC's al mismo tiempo. Para este trabajo solo se utilizó el SOC0, y se configuró de acuerdo a lo siguiente: se eligió convertir el canal dos del ADC_ A, se tomó una ventana de adquisición de 16 ciclos de reloj y se escogió disparar el ADC mediante la señal del PWM2. El canal dos del ADC_ A es donde se encuentra conectado el circuito de acondicionamiento, el cálculo de la ventana de adquisición se realizó siguiendo lo mostrado en [12].

Finalmente, se configuró el ADC para que generara una interrupción al final de cada conversión. Lo realizado en la subrutina de interrupción (ISR) corresponde a la *sección 4.4*.

4.4. Detección de actividad acústica

La detección de actividad acústica se realizó mediante un algoritmo, el cual se programó en lenguaje C y se cargó al DSP. El algoritmo se creó a partir de los métodos basados en energía para la detección de actividad acústica descritos en la *Sección 3.3* del presente trabajo, en específico, en el *Detector lineal basado en energía* de umbral adaptable.

La *Figura 4.8* muestra un diagrama de flujo del algoritmo implementado para la detección de actividad acústica y el algoritmo se explica en seguida.

Al inicio se configura el convertidor analógico-digital descrito en *4.3.3* para poder obtener la señal proveniente del micrófono. Después se establece el umbral de energía inicial. Este umbral calcula la energía del ruido inicial presente en el ambiente; el tiempo que se da para calcular el umbral inicial varía de acuerdo al ruido del ambiente, se escogió establecer el umbral durante los primeros 426ms. Para el cálculo de este umbral se utilizaron las ecuaciones *3.68* y *3.69*.

Posteriormente se llena un buffer de datos o una ventana con valores de la señal y se calcula la energía de la misma con la *ecuación 3.65*. Si la energía de la ventana o buffer es mayor a la del umbral establecido $E_n > Thr$ entonces se decide que se detectó actividad acústica $VAD = 1$. Si la energía de la ventana es menor a la del umbral establecido, entonces se decide que no se detectó actividad acústica $VAD = 0$ y se establece un nuevo umbral de energía (ruido) utilizando las ecuaciones *3.71* y *3.70*.

Los elementos del diagrama de flujo encerrados en líneas punteadas pertenecen al control del servomotor explicado en la *sección 4.5*; se incluyeron dentro del diagrama debido a que dependen de la decisión tomada en la detección de actividad y para evitar mostrar el diagrama de flujo de forma discontinua. El primer diamante de decisión corresponde a un ciclo que se realiza de forma infinita.

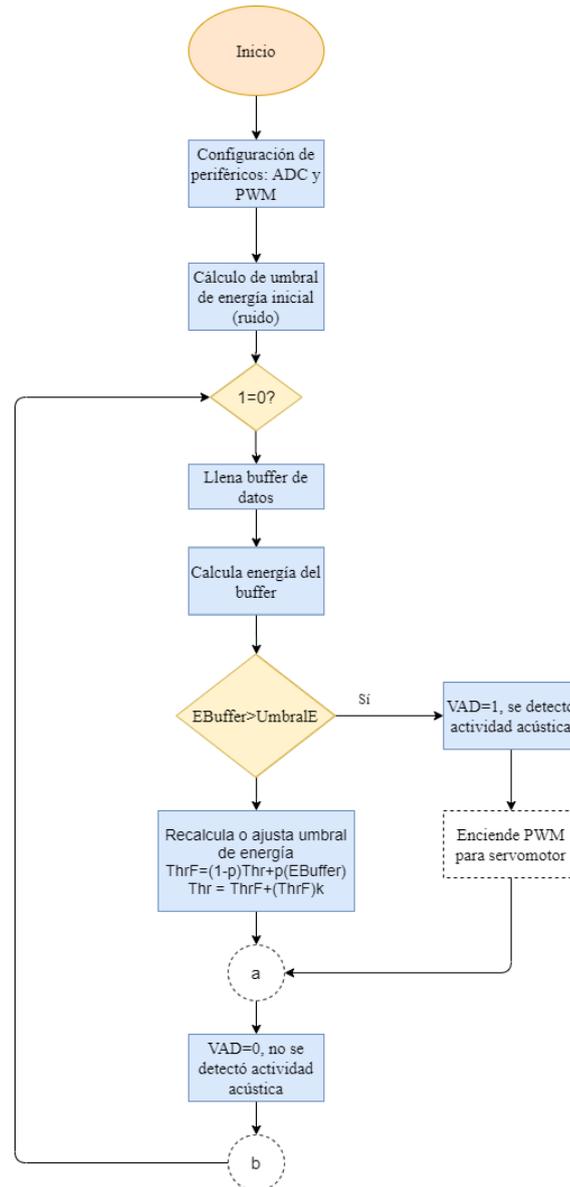


Figura 4.8: Diagrama de flujo del algoritmo para la detección de actividad acústica

El procesamiento de la señal se realiza por ventanas o buffers de datos de corta duración. Para procesar la señal se llena una ventana con datos de la misma y posteriormente se realiza el procesamiento. La *Figura 4.9* muestra el diagrama de flujo para el llenado de las ventanas de datos.

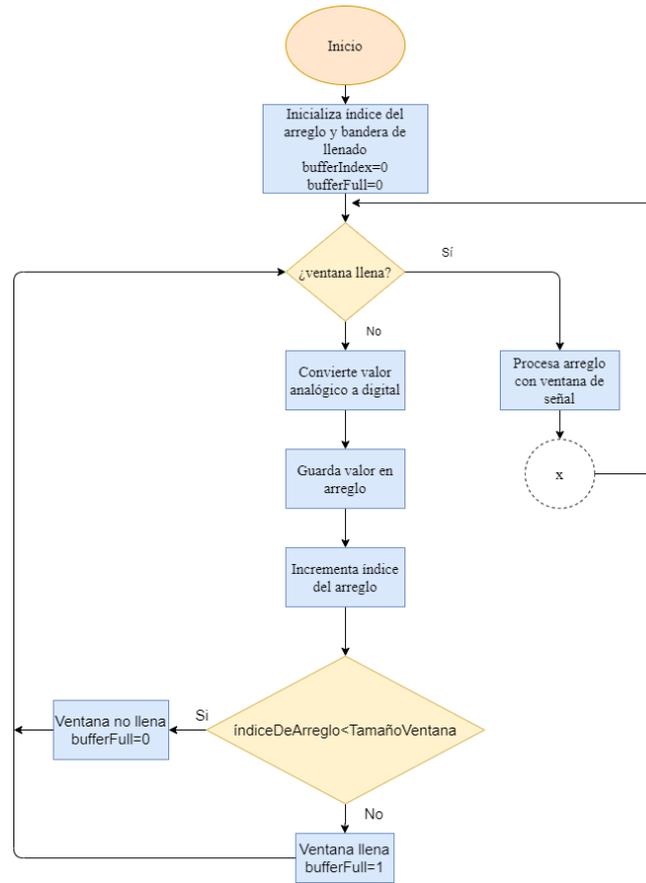


Figura 4.9: Diagrama de flujo del algoritmo para la generación de ventanas de la señal

La interrupción que se dispara al terminar cada conversión del ADC permite llenar la ventana de la señal; en la subrutina de interrupción se guarda cada valor de la señal convertido dentro de un arreglo de N datos. Antes de guardar cada valor en el arreglo se multiplica por tres, se divide entre 4096 y se le resta 1.6. Estas operaciones se realizan para guardar los valores de voltaje analógicos sin componente de DC correspondientes a los números convertidos dentro del arreglo de datos. Para acceder a cada elemento del arreglo se cuenta con un número que sirve de índice, cada vez que se realiza una conversión se incrementa el índice y el voltaje convertido se guarda en la siguiente posición del arreglo. Cuando el índice supera al límite o tamaño de ventana establecido se activa una bandera para indicar que la ventana está llena, una vez que se llena la ventana se realiza todo el procesamiento posterior: cálculo de la energía, decisión de detección de actividad, establecimiento del umbral de decisión, etc. Estas ventanas de corta duración permiten que el sistema opere en tiempo real.

4.5. Caracterización y control del servomotor

El control del servomotor que movía la cámara de vigilancia se realizó mediante un algoritmo programado en lenguaje C, el cuál se cargó al DSP. Antes de realización del algoritmo se obtuvieron diversos parámetros del servomotor que permiten su uso. La obtención de estos parámetros se hizo a partir de la hoja de datos del servomotor [37] y de forma experimental.

El servomotor utilizado fue el HS-422 [37] de HITEC. Este servomotor cuenta con tres terminales de entrada, la primera se utiliza para polarizar el servomotor, la segunda es para la señal de control y la tercera se usa para colocar una referencia de voltaje que usualmente es tierra. De acuerdo a las especificaciones mostradas en [37], el servomotor se polariza con voltajes entre 4.8V y 6V, para el trabajo presente se decidió polarizar con 5V.

La señal de control es una señal de PWM cuya amplitud puede estar entre 3.3V y 6V. La frecuencia de esta señal está en el intervalo de $[300 - 666]Hz$, este intervalo se obtuvo a partir de diversas pruebas experimentales y considerando las especificaciones de [37], donde se indica 666 Hz como frecuencia de operación.

Las pruebas para obtener el intervalo de frecuencia de la señal de PWM se realizaron mediante un generador de funciones que se conectó directamente a la terminal de control del servomotor, en específico, se utilizó el generador *BK PRECISION 4017A, 10 MHz Sweep Function Generator*.

La prueba consistió en colocar una señal cuadrada en la terminal de control del servomotor. Se varió la frecuencia de la señal cuadrada empezando en 60 Hz y terminando en 700 Hz, incrementando la frecuencia en pasos de 5Hz. Para cada frecuencia se varió el ciclo de trabajo de la señal en el intervalo de $D \in [0.1 - 0.9]$ y se observó el comportamiento de una flecha montada sobre el eje del motor. El ciclo de trabajo de la señal introducida en la terminal de control del motor determina la posición angular del eje del motor. Se encontró que en el intervalo de 300 a 670 Hz se lograba realizar un giro de 180° de la flecha montada sobre el motor, es decir, se tenía el intervalo de posiciones angulares completo del eje del motor. Para el presente trabajo se decidió utilizar una frecuencia de 320Hz para la señal de control del servomotor.

Una vez que se determinó el intervalo de frecuencia en el que operaba el servomotor, se determinó la correspondencia entre el ciclo de trabajo y la posición angular dada una frecuencia fija. Utilizando una señal cuadrada con frecuencia $f = 320Hz$ proveniente del generador de funciones se varió su ciclo de trabajo hasta llegar a una posición angular deseada. El intervalo de posiciones angulares fue de $\theta \in [0 - 180]^\circ$ y para cada posición angular se registró el valor del ciclo de trabajo correspondiente. Se repitió esto tres veces y se calcularon los valores promedio para el ciclo de trabajo; con estos valores promedio se utilizó regresión lineal para obtener la función que relacionaba el ciclo de trabajo con la posición angular. La función obtenida fue:

$$D = 0.003389442\theta + 0.196417065 \quad (4.11)$$

Donde θ es la posición angular del eje del servomotor y D es el ciclo de trabajo de la señal de PWM utilizada para controlar el motor.

4.5.1. Generador de PWM

El control del movimiento del motor se realizó por software y utilizando el módulo de PWM incluido como periférico en el DSP TMS320F28377S. El algoritmo utilizado y la configuración del módulo de PWM se explican a continuación.

El DSP TMS320F28377S cuenta con 12 módulos de PWM, cada uno con dos salidas, es decir, se pueden tener hasta 24 señales de PWM al mismo tiempo [12], [32]. Ocho de los módulos de PWM pueden configurarse para producir señales de PWM de alta resolución, es decir, se pueden tener hasta 16 señales de PWM de alta resolución [12], [32].

Los módulos de PWM están conformados por ocho submódulos configurables por software. Los submódulos son los siguientes:

- Base de tiempo. Se encarga de determinar el tiempo en el que ocurren todos los eventos que puede generar el módulo de PWM. Este submódulo está conformado por varios registros y tiene diversas funciones, pero la principal es establecer una cuenta periódica con la que se sincronizan las demás funciones del módulo de PWM. Esta cuenta se realiza a través de un contador de 16 bits bidireccional, es decir, que puede contar de forma ascendente, descendente o ascendente y descendente. El valor al que llega la cuenta antes de reiniciarse se carga en un registro (TBPRD) de 16 bits. La frecuencia de la señal de PWM producida está determinada por el valor cargado en el registro mencionado anteriormente y por la frecuencia del pulso de reloj con el que operan los módulos de PWM. Las ecuaciones 4.12 y 4.13 permiten calcular la frecuencia de la señal de PWM de salida.

$$F_{PWM} = \frac{1}{(TBPRD + 1)T_{TBCLK}} \quad (4.12)$$

$$F_{PWM} = \frac{1}{2(TBPRD)T_{TBCLK}} \quad (4.13)$$

$TBPRD$ es el valor de la cuenta cargado al registro con el mismo nombre, T_{TBCLK} es el periodo de la señal de reloj con la que funcionan los módulos de PWM. Esta última señal depende de la frecuencia de reloj con la que opera el DSP y del preescalador de reloj para los módulos de PWM.

La ecuación 4.12 se utiliza cuando la cuenta es ascendente o descendente en un periodo (PWM asimétrico) y la ecuación 4.13 se utiliza cuando la cuenta es ascendente y descendente en un periodo (PWM simétrico).

Este submódulo también permite controlar la sincronización de las bases de tiempo de los demás módulos de PWM y el desfase entre las señales de PWM de los diferentes módulos.

- Comparador de cuenta. Se encarga de comparar de forma continua el valor de la cuenta de la base de tiempo contra un valor determinado por el usuario (CMPA, CMPB, CMPC O CMPD). Cuando la cuenta de la base de tiempo es igual al valor de comparación establecido se genera un evento elegido. Estos eventos pueden ocurrir una o dos veces en un periodo dependiendo de como se configure la base de tiempo.

Este submódulo puede generar eventos en los tres modos de cuenta. La comparación en el modo de cuenta ascendente o descendente se utiliza para generar señales de PWM simétricas, mientras que la comparación en el modo ascendente-descendiente se utiliza para generar señales de PWM simétricas.

- **Calificador de acciones.** Este submódulo se utiliza para controlar la señal de salida del módulo de PWM. Decide que acción realizar dependiendo del evento que se presente, produciendo las formas de onda conmutadas requeridas en las salidas del módulo de PWM. Es decir, conmuta la salida, dependiendo del evento que se haya presentado, produciendo la señal de PWM.

Las acciones generadas y calificadas son enciende(set), apaga(clear) y conmuta(toggle) y se basan en los siguientes eventos: la cuenta y el periodo de la base de tiempo son iguales ($CTR = PRD$), la cuenta es igual a cero ($CTR = Zero$), la cuenta es igual al valor de comparación A ($CTR = CMPA$), la cuenta es igual al valor de comparación B ($CTR = CMPB$).

El calificador de acciones controla como se comportan las dos salidas del módulo de PWM. Las acciones impuestas y las formas de onda de las salidas son independientes entre ellas. Las posibles acciones impuestas a la salida son las siguientes

1. Establece alto. Coloca un nivel alto de voltaje en la salida A o B del PWM.
 2. Limpia. Coloca un nivel bajo de voltaje en la salida A o B del PWM.
 3. Conmuta. Invierte el nivel de voltaje en la salida A o B del PWM. Si tiene un nivel alto de voltaje, establece un nivel bajo y si tiene un nivel bajo, establece un nivel alto.
 4. Sin cambio. Mantiene los niveles de voltaje presentes a la salida.
- **Generador de tiempo muerto:** Se utiliza para generar un tiempo en el que las salidas de PWM se encuentran en un nivel bajo al mismo tiempo. Se puede realizar mediante el submódulo calificador de acciones, pero este submódulo permite la implementación usando un retraso disparando por flanco con control de polaridad.

Este submódulo resulta de especial importancia cuando se trabaja con pares de señales de PWM conectadas a amplificadores de potencia conmutados, por ejemplo en amplificadores de audio o amplificadores para motores de DC. El submódulo permite generar pares de señales de PWM con banda muerta a partir de una sola entrada, es decir, a partir de la generación de una sola señal de PWM.

- **Troceador de PWM.** Permite que una señal portadora de alta frecuencia module la forma de onda del PWM generado por el calificador de acciones y el generador de tiempo muerto. La frecuencia de la portadora es programable y puede no utilizarse si se desea.

- Zona de rodado. Se encarga de enviar al módulo de PWM seis señales diferentes (TZ1 a TZ6) que indican fallas externas o condiciones de error; el comportamiento del PWM se puede programar ante una condición de falla, forzando la salida a una de las siguientes: salida en alto, salida en bajo, alta impedancia y sin modificación. Este submódulo permite proteger al DSP y a circuitos conectados a las salidas de PWM ante cortos circuitos y condiciones de sobrecorriente. Al igual que el submódulo troceador puede no utilizarse si se desea.
- Disparador de eventos. Se encarga de recibir los eventos generados por los submódulos de base de tiempo, comparador de cuenta y comparador digital. Este submódulo permite generar interrupciones en base a los eventos recibidos y/o un pulso de inicio de conversión para el convertidor analógico-digital. Este submódulo monitorea diferentes condiciones de eventos y se configura para generar la interrupción o el pulso de conversión de acuerdo a un preescalador de eventos. El preescalador cuenta los eventos y decide cuando disparar el ADC o generar la interrupción. Puede realizar esto hasta con 15 eventos, es decir, cada primer evento, segundo evento, tercer evento, etc. hasta el décimo quinto.

La configuración de este submódulo se realiza mediante un conjunto de registros siendo los siguientes dos, los mínimos necesarios para su funcionamiento:

1. ETSEL. Permite seleccionar cual de los posibles eventos disparará la interrupción o el pulso de inicio de conversión del ADC.
2. ETPS. Permite especificar el número de eventos requeridos para disparar una interrupción o un inicio de conversión.

Existen 15 eventos que se pueden identificar y se encuentran especificados en [12]. El número de eventos que han ocurrido se puede conocer a través del registro ETPS[INTCNT].

- Comparador Digital. Se encarga de comparar señales externas al módulo de PWM para generar eventos o acciones que llegan directamente a los submódulos de base de tiempo, disparador de eventos o zona de rodado. Las señales producidas por este submódulo disparan eventos que pueden ser filtrados o llevados a los tres submódulos mencionados anteriormente. A partir de las señales generadas por este submódulo se pueden generar interrupciones por zona de rodado, generar inicios de conversión del ADC, forzar eventos y generar un evento para sincronizar el contador de la base de tiempo.

4.5.2. Algoritmo de control

El algoritmo de control implementado se encarga de generar la señal de PWM necesaria para hacer girar el motor junto con la cámara. El motor gira en el intervalo de $[10 - 170]^\circ$ una vez que se ha detectado actividad acústica. El movimiento del motor se realiza de forma continua, es decir, empieza en la posición angular de 10° , llega a 170° y regresa a 10° en un movimiento continuo. Como se verá más adelante, la forma de operación puede modificarse para mover al motor a alguna posición angular fija deseada y modificar la velocidad angular del movimiento.

La *Figura 4.10* muestra el diagrama de flujo correspondiente al algoritmo de control descrito a continuación. Al inicio se configuró el módulo de PWM y sus respectivos submódulos con

el fin de poder generar la señal deseada en un momento posterior durante la ejecución del programa.

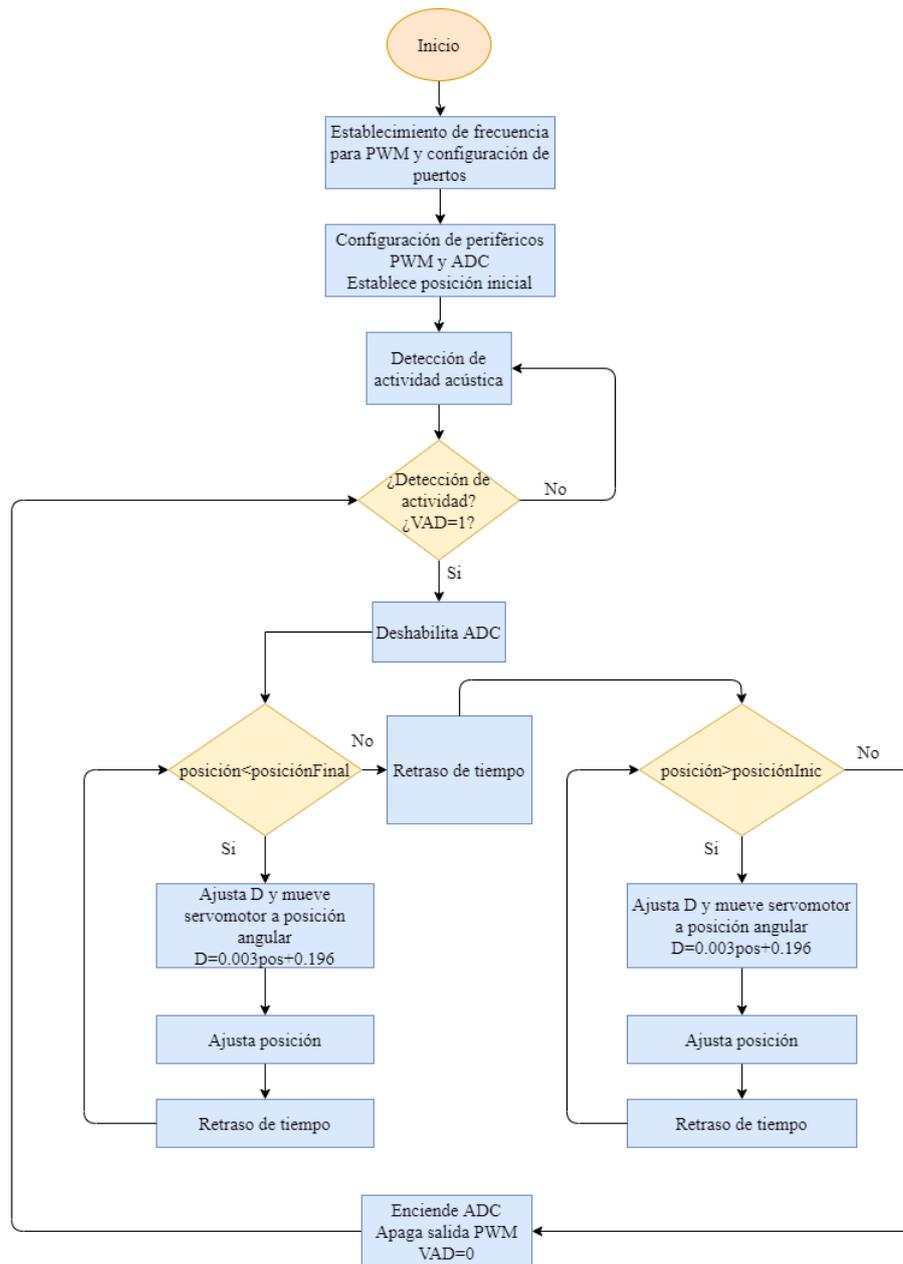


Figura 4.10: Diagrama de flujo del algoritmo para el control

La primer acción realizada para la configuración fue el establecimiento de la frecuencia de la señal de PWM con la que se controlaba el motor. Como se describió en la *sección 4.5* esta frecuencia debe encontrarse en el intervalo $[300 - 670] Hz$ para que el motor pudiera girar en todo el intervalo disponible. Se eligió como frecuencia de la señal 320 Hz, debido a que se encontraba en el intervalo de operación establecido y a que fue la utilizada para obtener la relación entre el ciclo de trabajo y la posición angular de salida.

Posteriormente se configuraron los puertos de propósito general (GPIO) 12 y 13 para ser utilizados como terminales de salida para la señal de PWM. Estos puertos son los correspon-

dientes al módulo PWM7 del TMS320F28377S [32]. Se eligió este módulo de PWM debido a que las salidas para la señal de PWM se encuentran conectadas a pines físicos de la tarjeta LaunchXL-F28377S y a un led RGB, parte de la Tarjeta de expansión, que permitía visualizar el comportamiento de la señal de PWM.

La configuración del módulo PWM7 se llevó a cabo utilizando los registros pertenecientes a los submódulos descritos en la *subsección 4.5.1*. Se estableció que el módulo trabajara con una señal de reloj de 20 MHz y que la cuenta de la base de tiempo fuera unidireccional ascendente. En base a la frecuencia de la señal de reloj (20 MHz) y a la escogida para la señal de PWM (320 Hz) se calculó el valor del periodo que se cargó al registro TBPRD, el cual determina la frecuencia de la señal de PWM. Para el cálculo se partió de la *ecuación 4.12* resultando en:

$$TBPRD = \frac{1}{(F_{PWM})(T_{TBCLK})} - 1 = \frac{F_{TBCLK}}{F_{PWM}} - 1 \quad (4.14)$$

$$TBPRD = \frac{20M}{320} - 1 = 62499 \quad (4.15)$$

Como configuración inicial se estableció un valor de comparación de cero para el submódulo de comparador de cuenta y que el calificador de acciones no realizara nada. Es decir, mantiene la salida del PWM sin cambios y no genera la señal. Esto se modificó posteriormente; los demás submódulos pertenecientes al PWM no fueron utilizados.

Una vez configurado el módulo de PWM se activó el contador de la base de tiempo, es decir, se encendió el módulo. Después de realizar esto, el algoritmo espera la decisión tomada por la detección de actividad acústica. Si no se detecta actividad acústica $VAD = 0$ no se genera la señal de PWM; por otro lado si el algoritmo detecta actividad acústica $VAD = 1$ realiza lo siguiente:

Se deshabilitan los inicios de conversión del convertidor analógico-digital, posteriormente se establece una posición angular inicial a la que se mueve el servomotor. De forma iterativa se incrementaba la posición a la que se mueve el servomotor y se da un tiempo de espera para observar el movimiento. Se repite esto hasta que el servomotor ha llegado a una posición angular final.

Después de que el servomotor llega a la posición final, se da un tiempo de espera y se repite lo descrito anteriormente para regresar al servomotor a su posición angular inicial. Es decir, se mueve el servomotor a una posición angular que decremента de forma iterativa y se da un tiempo de espera entre cada movimiento. Lo descrito anteriormente ocasiona que el servomotor realice un movimiento lento de vaivén con el que se puede observar el entorno mediante la cámara montada sobre el motor.

La duración del tiempo de espera que se da cada vez que se modifica la posición angular, determina la velocidad a la que se mueve el motor, así como el valor del paso con el que se decremента o incrementa la posición. Las posiciones iniciales y finales tienen un valor constante predeterminado, pero se pueden modificar para funcionar de forma variable.

Para realizar el movimiento del servomotor se parte de la *ecuación 4.11*. Cada vez que se modifica la posición angular a la que se mueve el servomotor se recalcula el valor del tiempo

en alto t_H correspondiente de la señal de PWM y se carga este valor a un registro parte del submódulo de comparación de cuenta. Es decir, se recalcula el ciclo de trabajo y se modifica la señal de PWM. El valor cargado al registro esta dado por la *ecuación 4.18*.

$$D = 0.003389442\theta + 0.196417065 \quad (4.16)$$

$$\frac{t_H}{T} = 0.003389442\theta + 0.196417065 \quad (4.17)$$

$$t_H = (0.003389442\theta + 0.196417065)T \quad (4.18)$$

Cuando se modifica el ciclo de trabajo de la señal de PWM, D , se configura el submódulo calificador de acciones para que conmute la salida de PWM cada vez que la cuenta de la base de tiempo llega al valor en el registro del submódulo de comparación de cuenta. La salida B del módulo de PWM no se utiliza, la señal conectada al motor proviene del PWM7A.

4.6. Visualización de video

El dispositivo de salida utilizado en este sistema de vigilancia fue un monitor LCD el cual se encuentra conectado a la cámara de vigilancia. El monitor utilizado fue el modelo LT073W de la marca Vsonic. Este monitor cuenta con una pantalla de cristal líquido (TFT-LCD) de siete pulgadas y tiene una resolución de 480x234 pixeles. Permite visualizar vídeo en tiempo real el cual puede provenir de distintas fuentes. Cuenta con un conector de 3.5mm que sirve de entrada para audio y video (AV/IN) proveniente de una fuente externa. También cuenta con una antena para recibir señales de TV y el monitor permite sintonizar los canales de TV. Además cuenta con bocinas integradas.

El monitor también permite visualizar imágenes o video y escuchar audio en diversos formatos de compresión de medios incluyendo MP3, WMA, RMVB, AVI, VIB y JPEG. Cuenta con una ranura para tarjetas de memoria MMC o SD y un puerto USB que permite introducir medios almacenados en los formatos previamente mencionados para su posterior reproducción en el monitor.

El dispositivo tiene dos conectores para señales de salida. El primero es una salida de audio estereo que permite conectar audifonos o equipos de amplificación de audio. El segundo es una salida de video y audio (AV/OUT).

El monitor se polariza utilizando una fuente de corriente directa de 12V y su consumo de potencia nominal es de 14W.

La cámara utilizada para la realización de este trabajo fue la CCTV-160 de la marca Steren. Es una minicámara alámbrica a color con una lente ajustable tipo cañón. Esta cámara cuenta con un dispositivo de carga acoplada (CCD) utilizado como sensor para registrar las imágenes en la lente [38].

La cámara se montó sobre una base metálica desmontable. Esta base está articulada y permite colocar la cámara en diversas posiciones requeridas.

La señal entregada por la cámara es en banda base 1 Vpp (RCA) en formato NTSC y PAL. El barrido es de 420 líneas, el tamaño de la imagen NTSC producida es de 720x480 pixeles y cuenta con balance automático de blancos.

La polarización de la cámara se realiza mediante una fuente de corriente directa de 12V y cuyo consumo de potencia nominal es de 6W. La temperatura máxima de operación de la cámara es de 80°C y su iluminación mínima es de 0.3 lux. Cuenta solamente con dos conectores: uno para la señal de polarización y otro para la salida de la señal de video producida.

El procesamiento de la señal de video generada por la cámara se lleva a cabo por la cámara misma y por el monitor. Como se mencionó en la *sección 1.1* el procesamiento y la transmisión de la señal de video se encuentran más allá del alcance de este trabajo, por esta razón no se mencionan algoritmos de procesamiento de imágenes o video.

La cámara y el monitor se conectan entre ellos utilizando un cable con conectores RCA para audio y video. Debido a que la señal producida por la cámara solamente era de video, los conectores para audio se utilizan. Además se utiliza un adaptador de RCA a 3.5mm para el monitor LCD.

En el *Capítulo 5* se muestra un diagrama del sistema completo y la conexión de sus elementos.

4.7. Resumen

En este capítulo se presentó una descripción detallada del sistema de vigilancia que se realizó. Se mostró el sistema de vigilancia en su totalidad y cada una de las partes o etapas que lo conforman. Para cada etapa se explicaron aspectos detallados del diseño, la conformación y el funcionamiento de la etapa. Se dieron a conocer los elementos de hardware y software utilizados y la forma en que se interconectaban. Se plasmaron los algoritmos implementados, la base matemática utilizada para diseñarlos y los diagramas de flujo de los mismos para tener un mejor entendimiento de su funcionamiento. En cuanto al DSP, se mostró la forma en que se configuró y usó para poder llevar a cabo las tareas que se necesitaban realizar.

Capítulo 5

Pruebas y resultados

La función principal de los sistemas de vigilancia es poder monitorear adecuadamente un fenómeno durante largos periodos de tiempo y detectar cambios en el mismo para poder intervenir cuando el proceso no se efectúa de la forma deseada.

El desempeño del sistema y su capacidad de monitorear correctamente un fenómeno y los cambios en el mismo dependerá del funcionamiento adecuado de cada una de sus partes y la capacidad de las mismas para decidir frente a los cambios que se presenten al transcurrir el tiempo.

En este capítulo se muestra el comportamiento de cada una de las partes que conforman al sistema de vigilancia en diferentes situaciones y bajo condiciones variadas, esto se hace con el fin de evaluar su funcionamiento y desempeño.

Se presentan las pruebas realizadas a cada parte del sistema y los resultados obtenidos de las mismas. También se incluye la forma en que se realizaron las pruebas y la finalidad de ellas.

Los resultados obtenidos para el algoritmo de detección de actividad acústica y para el movimiento del servomotor incluyen gráficos que facilitan su visualización e interpretación. También se da un análisis y una evaluación objetiva a partir de los resultados obtenidos.

5.1. Montaje y colocación del sistema

Las etapas, elementos de hardware y software que conforman al sistema se muestran y describen en el *Capítulo 4*. La forma en que se encuentran colocados y conectados entre ellos, así como los elementos utilizados para realizar las pruebas del sistema se describen a continuación.

El micrófono se encuentra montado sobre una base que permite ajustar la altura a la que se quiere colocar. Este micrófono está conectado directamente mediante un conector de 3.5mm a la entrada de *Micrófono uno* de la *Tarjeta de expansión* desarrollada en el Laboratorio de Procesamiento Digital de Señales de la UNAM [36]. La tarjeta LAUNCHXL-F28377S que

contiene al DSP donde se programaron y cargaron los algoritmos de detección de actividad acústica y de control del servomotor se encuentra montada sobre la *Tarjeta de expansión* mencionada previamente, a través de un conjunto de conectores incorporados en la tarjeta. En los pines correspondientes al GPIO12 y GND se conectaron un par de alambres; el GPIO12 tenía la salida de PWM utilizada para controlar el servomotor HS-422 de HITECH [37]. Este servomotor cuenta con tres terminales: dos terminales de alimentación y una de control. Las terminales de alimentación se conectaron a una fuente de poder de 5V (y 850mA) y la terminal de control se conectó directamente al pin correspondiente a la salida de PWM producida por la tarjeta (pin 40, GPIO12).

El eje del servomotor está montado sobre una placa acrílica que restringe el movimiento del eje del servomotor, pero permite girar a este último. Sobre el servomotor está colocada la cámara de vigilancia, para esto se montó la cámara sobre una base metálica articulable y esta base es la que se encuentra sobre el servomotor.

La cámara y el monitor se conectaron directamente mediante un cable con conectores RCA como se mencionó en la *subsección 4.6*. El monitor se encuentra montado sobre otra base articulada con la que se puede modificar su posición.

Se ocuparon tres fuentes de alimentación para la realización del sistema y las pruebas del mismo. Una fuente de 15V para la *Tarjeta de Expansión*, una fuente de 12V para la cámara y el monitor LCD y una fuente de 5V para el servomotor. La *Figura 5.1* muestra el diagrama del sistema completo y la conexión de sus elementos.

Para las pruebas y el desarrollo del software se utilizaron, además de lo ya descrito, un par de altvoces, un zumbador y el software *Code Composer Studio TM* de *Texas Instruments TM*.

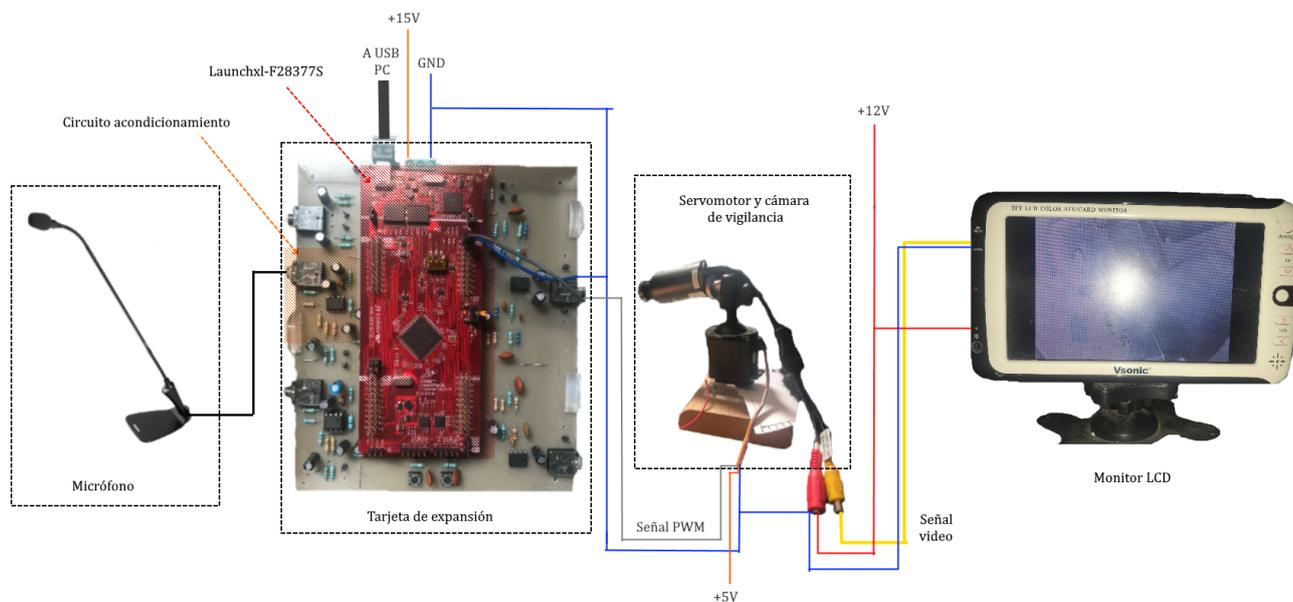


Figura 5.1: Sistema completo y conexión de sus elementos

5.2. Desempeño del algoritmo de detección de actividad acústica

5.2.1. Parámetros de desempeño

El desempeño de los algoritmos de detección de actividad acústica o detección de actividad de voz (VAD) se puede observar a través de un conjunto de parámetros con los que se obtiene el error introducido por el algoritmo durante el proceso de detección. Estos parámetros fueron propuestos por Freeman en [39] y evalúan el desempeño en términos del tiempo que dura la señal de detección de actividad y el grado de recorte introducido. Para poder determinar la severidad del recorte de la señal y el ruido detectado como la señal acústica deseada (detección falsa) se necesita comparar la salida del método de detección contra una decisión de detección ideal. Esta decisión de detección ideal se establece en las condiciones más favorables para la detección, es decir que se tenga una relación señal a ruido alta $SNR > 10$. Para obtener la decisión ideal o de referencia se reproduce la señal en un ambiente silencioso y se marcan los puntos de inicio y fin de la señal. Después se añade una señal de un bit que representa la decisión de la detección ideal de actividad, esta señal adquiere un valor de "0" lógico cuando la señal a detectar no está presente y un valor de "1" lógico cuando si está presente, además coincide perfectamente con los puntos de inicio y fin establecidos para la señal de audio.

Los parámetros de desempeño que se mencionaron previamente son los cuatro descritos a continuación [31], [40], [39], [41]:

- Front End Clipping (FEC): Recorte introducido al pasar del ruido a la actividad acústica, es decir, el número de muestras o ventanas que no se clasificaron correctamente con actividad cuando se pasa de ruido a actividad acústica.
- Midspeech Clipping (MSC): Recorte ocasionado por calificar erróneamente actividad como ruido. Indica el número de muestras que se clasificaron erróneamente como ruido durante un intervalo con actividad acústica.
- Overhang (OVER): Ruido interpretado como la señal debido a que el bit de actividad permaneció encendido al pasar de la señal al ruido, es decir, la cantidad de muestras o ventanas que se clasificaron erróneamente con actividad cuando se pasa de la señal acústica al ruido.
- Noise Detected as Speech (NDS): Ruido interpretado como la señal durante un periodo de silencio. Es la cantidad de muestras o ventanas de ruido que se clasificaron erróneamente con actividad cuando no hay actividad de la señal.

Los cuatro parámetros de desempeño descritos previamente se muestran de forma gráfica en la *Figura 5.2*.

Los parámetros FEC y MSC se pueden agrupar para medir la cantidad de recorte introducido en la señal. OVER y NDS se pueden agrupar para indicar la cantidad de falsas alarmas.

Una forma similar de medir el desempeño del algoritmo de detección de actividad acústica es mediante la tasa de faltas (Missing Rate MR) y la tasa de falsa alarma (Falsa alarm Rate FAR) como se muestra en [42]. La MR mide la proporción de muestras o cuadros con actividad de la

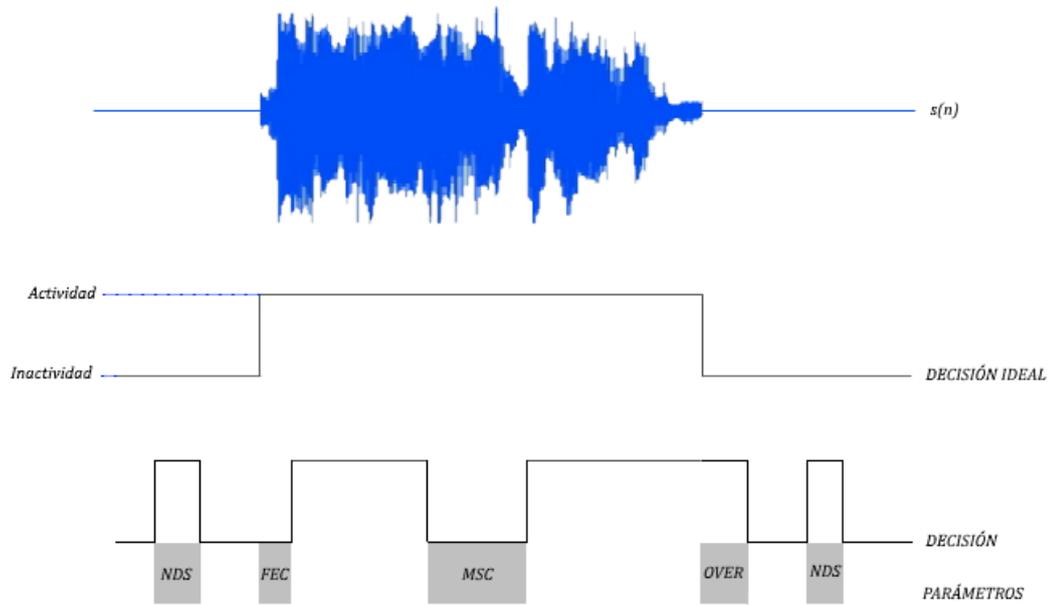


Figura 5.2: Parámetros objetivos de desempeño

señal que no fueron detectados y la FAR mide la proporción de cuadros o muestras de ruido que fueron detectados incorrectamente como la señal. La MR y la FAR pueden combinarse para formar la tasa de error medio total (Half total error rate HTER) [42].

La forma en que se calcularon todos los parámetros previamente descritos se basó en lo mostrado en [31] y se muestra a continuación:

- FEC: se expresa en porcentaje y se obtiene mediante la ecuación 5.1

$$\%FEC = \left(\frac{N_{fc}}{N_s} \right) 100 \quad (5.1)$$

N_{fc} es el número de muestras clasificadas erróneamente como ruido cuando se pasa de un intervalo sin actividad a la actividad acústica. N_s es el total de muestras pertenecientes a la señal que se desea detectar.

- MSC: se expresa en porcentaje y se obtiene mediante la ecuación 5.2

$$\%MSC = \left(\frac{N_{mc}}{N_s} \right) 100 \quad (5.2)$$

N_{mc} es el número de muestras clasificadas erróneamente como ruido en los periodos de actividad acústica.

- OVER: se expresa en porcentaje y se obtiene mediante la ecuación 5.3

$$\%OVER = \left(\frac{N_{ov}}{N_n} \right) 100 \quad (5.3)$$

N_{ov} es el número de muestras detectadas incorrectamente con actividad cuando se pasa de la actividad acústica a un intervalo de inactividad. N_n es el total de muestras pertenecientes a los intervalos de inactividad.

- NDS: se expresa en porcentaje y se obtiene mediante la ecuación 5.4

$$\%NDS = \left(\frac{N_{nds}}{N_n} \right) 100 \quad (5.4)$$

N_{nds} es el número de muestras detectadas incorrectamente con actividad cuando se pasa de la actividad acústica a un intervalo de inactividad.

La tasa de faltas y la tasa de falsa alarma se obtuvieron a partir de los cuatro parámetros de desempeño previamente expuestos. La tasa de faltas se obtuvo sumando el porcentaje de FEC y de MSC; la tasa de falsa alarma se calculó sumando el porcentaje de OVER y de NDS, esto es:

$$MR = FEC + MSC = \left(\frac{N_{fc} + N_{mc}}{N_s} \right) 100 \quad (5.5)$$

$$FAR = OVER + NDS = \left(\frac{N_{ov} + N_{nds}}{N_n} \right) 100 \quad (5.6)$$

La tasa de error medio total es el promedio de MR y FAR

$$HTER = \frac{MR + FAR}{2} \quad (5.7)$$

5.2.2. QUT-NOISE-TIMIT para la evaluación de algoritmos de detección de actividad

El corpus QUT-NOISE-TIMIT [42] consiste en un conjunto de 600 horas de secuencias o señales de ruido ambiental que permiten evaluar algoritmos de detección de actividad de voz o acústica. La colección incluye ruido grabado en cinco escenarios comunes: una cafetería, el interior de una casa, el interior de un automóvil, entornos reverberantes y en intersecciones de calles.

En total el corpus incluye una colección de 20 grabaciones de al menos 30 minutos de ruido grabado en los escenarios mencionados previamente. Estas grabaciones se llevaron a cabo bajo diferentes condiciones, a cada escenario mencionado corresponden cuatro grabaciones. Estas grabaciones se realizaron utilizando una grabadora de mano Zoom H2 que producía como salida archivos con audio estereo en formato WAV, con una frecuencia de muestreo de 48kHz y 16 bits por muestra.

Los escenarios donde se grabó el ruido del corpus se describen brevemente a continuación:

- **Café:** Se realizaron dos grabaciones en una cafetería típica al aire libre y otras dos en una cafetería dentro de un centro comercial. Cada par de grabaciones se realizó en la misma locación pero colocando la grabadora en puntos diferentes dentro del lugar.
- **Casa:** Se realizaron dos grabaciones en la sala y dos en la cocina de una casa. De igual manera cada par de grabaciones se realizó en la misma locación pero colocando la grabadora en puntos diferentes dentro del lugar.
- **Auto:** Se realizaron las cuatro grabaciones dentro del mismo auto pero en diferentes condiciones. Dos con la ventana abierta y dos con la ventana cerrada.
- **Reverberantes:** Se realizaron dos grabaciones dentro de una piscina cerrada y dos dentro de un estacionamiento parcialmente cerrado.
- **Calles:** Se grabaron dos señales en una intersección dentro de la ciudad y otras dos grabaciones en una intersección a las afueras de la ciudad.

En el trabajo presente se utilizarón grabaciones de ruido pertenecientes a este corpus para probar y evaluar el desempeño del algoritmo de detección de actividad acústica propuesto. En la *subsección 5.2.3* se especifica que grabaciones fueron utilizadas y como se usaron.

Una descripción más detallada del corpus y su uso se pueden encontrar en [42].

5.2.3. Pruebas del algoritmo y resultados

El algoritmo de detección de actividad acústica implementado y descrito en la *Sección 4.4* de este trabajo se probó bajo diferentes condiciones y se evaluó su desempeño utilizando los parámetros descritos en la *Subsección 5.2.1*. Las pruebas realizadas y los resultados obtenidos en las mismas se describen a continuación.

Se realizaron 18 pruebas con las que se valoró el desempeño del algoritmo en diferentes ambientes de ruido y variando distintos parámetros del algoritmo. Las pruebas se dividieron en dos grupos de nueve pruebas cada uno, a cada grupo de pruebas le corresponde un valor distinto del parámetro k_d utilizado para ajustar el umbral de decisión del algoritmo como se describe en la *Subsección 3.3.1*. Al primer grupo le corresponde un valor de $k_d = 0.2$ y al segundo grupo $k_d = 0.1$.

Los grupos contienen nueve pruebas del algoritmo en diferentes ambientes ruidosos. La forma en que se realizaron las nueve pruebas del grupo fue la siguiente: en un cuarto silencioso se reprodujeron, mediante un par de altavoces, tres de las secuencias de ruido pertenecientes al corpus *QUT-NOISE-TIMIT* [42] descrito en la subsección 5.2.2. Las secuencias de ruido ambiental escogidas fueron la grabada en la cafetería al aire libre, la proveniente de la sala dentro de la casa y la de la calle en las afueras de la ciudad. Se programó al DSP para que produjera una señal de 5.3s de duración durante la reproducción de las señales de ruido. Esta señal constaba de una señal digital que salía a través del GPIO12 del DSP, a su vez este GPIO se encontraba conectado a un zumbador que generaba una señal acústica. La señal del GPIO tenía una duración de 5.35s, los primeros 3.22s la señal del GPIO tenía un valor de "0" lógico y el zumbador no producía sonido; el segundo siguiente el GPIO poseía un valor de "1" lógico y el zumbador generaba sonido; finalmente el GPIO regresaba al "0" lógico el tiempo restante

5.2. DESEMPEÑO DEL ALGORITMO DE DETECCIÓN DE ACTIVIDAD ACÚSTICA 73

de la señal y no se producía sonido con el zumbador. Para cada señal de ruido escogida se establecieron tres diferentes SNR con la señal producida por el zumbador, siendo el segundo de sonido del zumbador la señal a detectar. Las SNR establecidas para cada señal de ruido fueron 10dB (alta), 5dB(media) y 0dB(baja).

Resumiendo lo descrito en los párrafos anteriores: se realizaron 18 pruebas divididas en dos grupos de nueve pruebas, al primer grupo le correspondía $k_d = 0.2$ y al segundo grupo $k_d = 0.1$, Dentro de cada grupo se tenían nueve pruebas agrupadas en tres grupos de tres pruebas: tres por locación de ruido ambiental y con diferente SNR.

Se realizaron diversos programas en lenguaje C que se cargaron al DSP y que sirvieron para llevar a cabo las pruebas. El primer programa calculaba la SNR promedio durante el intervalo de 5.3s en el que el DSP generaba la señal reproducida por el zumbador. El segundo programa se encargaba de medir el tiempo en que iniciaban y terminaban los pulsos de la señal de detección de actividad acústica. A partir de estos tiempos y tomando como referencia los tiempos de duración descritos para la señal del GPIO12 se calcularon los parámetros de desempeño del algoritmo.

Los resultados obtenidos para cada una de las 18 pruebas, representan los valores promedio de 10 iteraciones bajo las mismas condiciones.

Los resultados de las pruebas obtenidas se pueden observar en el *Cuadro 5.1* y en el *Cuadro 5.2*.

Detector lineal basado en energía con umbral adaptable					
k=0.2					
Ambiente	SNR(dB)	% FEC	% MSC	% OVER	% NDS
CAFE	10	0.92625	0.00000	0.12084	0.66566
CAFE	5	1.49738	49.99969	0.53483	0.57257
CAFE	0	0.00000	99.99938	0.00000	0.34621
SALA	10	0.58829	0.00000	1.01200	0.58053
SALA	5	0.00000	99.99938	0.00000	0.09321
SALA	0	2.29946	97.64644	0.00000	0.39947
CALLE	10	0.53482	0.00000	0.13318	0.13316
CALLE	5	3.10583	1.90412	0.14649	0.34179
CALLE	0	1.03346	0.00000	0.12090	0.14644

Cuadro 5.1: Parámetros de desempeño del algoritmo de detección para $k_d = 0.2$

El *Cuadro 5.1* muestra que el algoritmo logra delimitar correctamente y con precisión los extremos de la señal de interés, ya que el porcentaje de FEC y OVER es bajo (<5%). Por otra parte, el algoritmo muestra un desempeño pobre en entornos con una SNR media o baja ya que para los entornos de la sala y la cafetería falló en detectar la actividad de la señal de interés en la mayoría de los casos (MSC>49%).

El *Cuadro 5.2* muestra que el algoritmo obtuvo un mejor desempeño en cuanto al número de detecciones existosas de actividad de la señal que cuando se utilizó $k_d = 0.2$, ya que el MSC

Detector lineal basado en energía con umbral adaptable k=0.1					
Ambiente	SNR(dB)	% FEC	% MSC	% OVER	% NDS
CAFE	10	0.49810	0.00000	0.25387	1.25141
CAFE	5	0.49793	0.00000	0.22720	1.71737
CAFE	0	1.13892	25.51955	6.38552	2.68923
SALA	10	0.49874	0.00000	0.25403	0.26628
SALA	5	0.55660	0.05346	0.59930	24.00426
SALA	0	1.57422	0.00000	2.29075	21.50082
CALLE	10	0.45084	0.00000	0.25354	3.03533
CALLE	5	0.50822	0.10693	0.22553	11.78196
CALLE	0	1.25566	5.51268	1.46828	12.72917

Cuadro 5.2: Parámetros de desempeño del algoritmo de detección para $k_d = 0.1$

fue menor en los entronos más ruidosos. Por otra parte el OVER y el NDS fue mayor en la mayoría de los casos por lo que presenta una mayor cantidad de alertas falsas que cuando se utilizó $k_d = 0.2$.

Después de obtener los parámetros de desempeño del algoritmo mostrados en el *Cuadro 5.1* y en el *Cuadro 5.2*, se obtuvo la tasa de fallas (MR) y la tasa de falsa alarma (FAR) del algoritmo para todos los casos en la forma descrita en la *Subsección 5.2.1*. En consecuencia, también se obtuvo la tasa de error total medio para todos los casos. Esto se muestra en las *Figuras 5.3 y 5.4*.

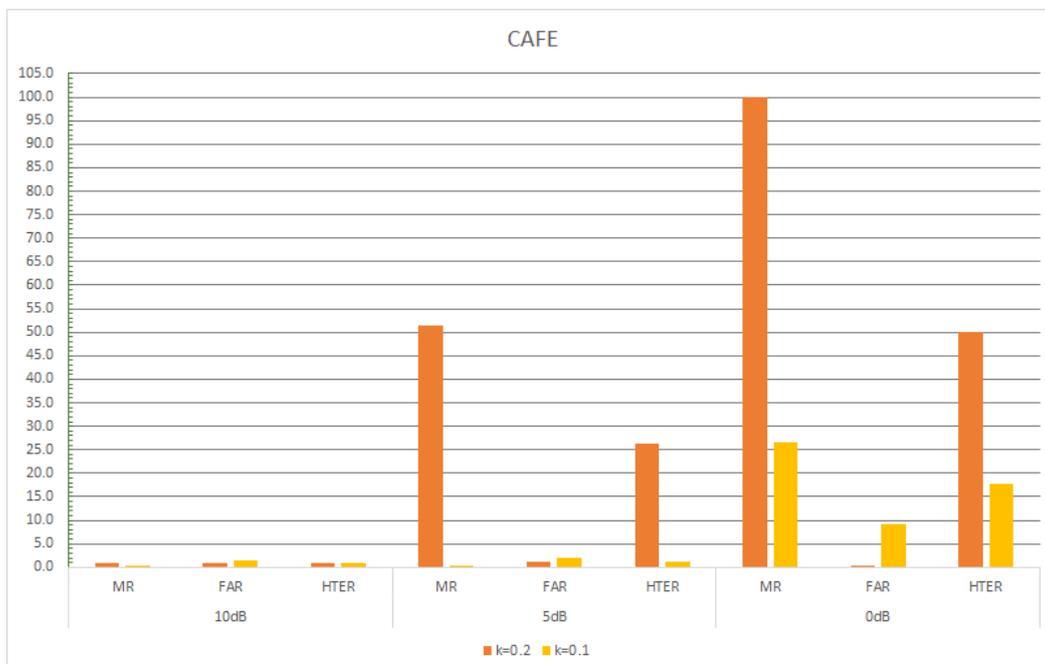


Figura 5.3: MR, FAR y HTER para el ambiente cafetería

5.2. DESEMPEÑO DEL ALGORITMO DE DETECCIÓN DE ACTIVIDAD ACÚSTICA 75

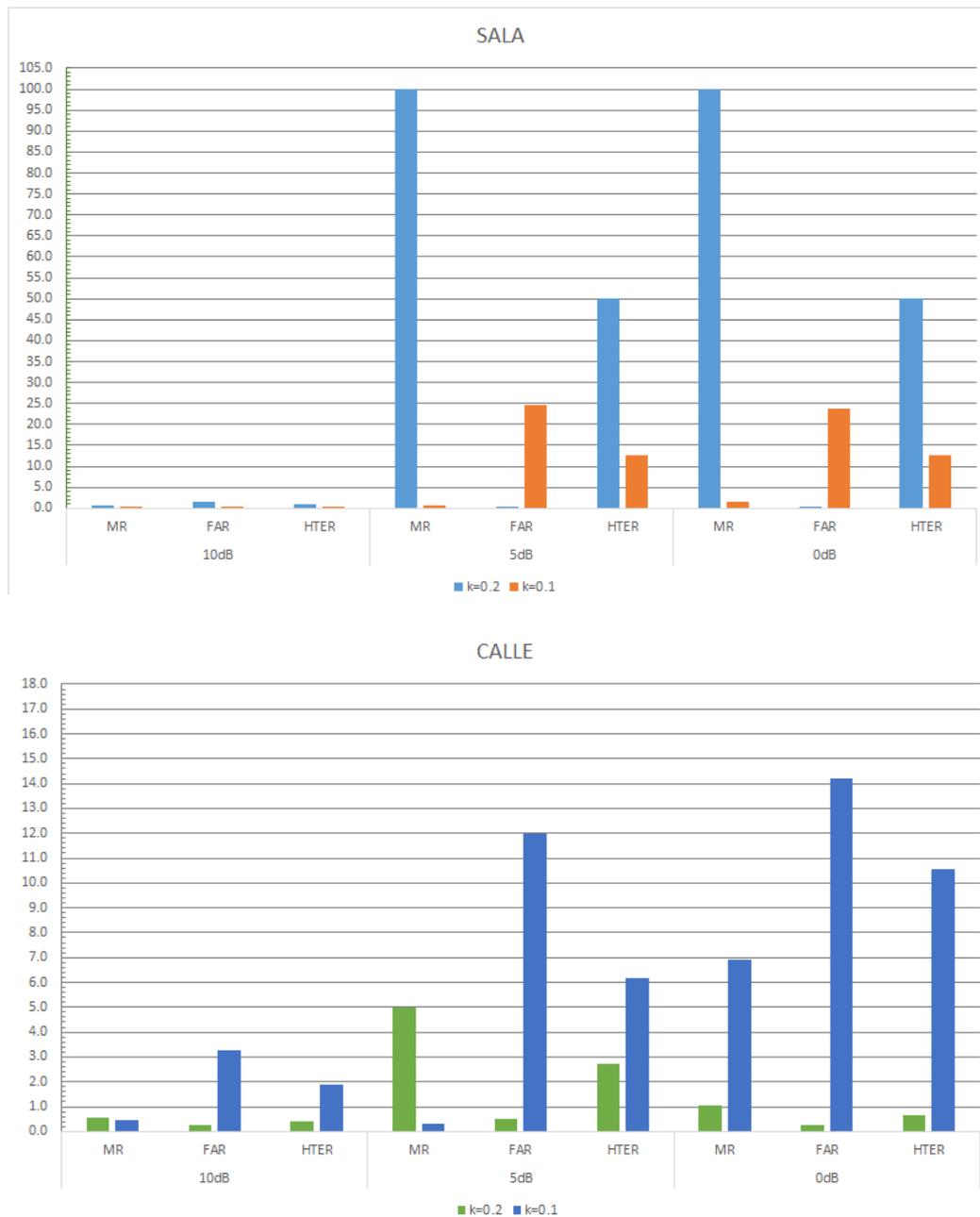


Figura 5.4: MR, FAR y HTER para el ambiente sala y calle

Se observa que en la cafetería y la sala el error (HTER) fue menor utilizando $k_d = 0.1$, pero para el ambiente de calle el error (HTER) fue menor utilizando $k_d = 0.2$. Esto quiere decir que a pesar de que se tenga una misma SNR, el desempeño del algoritmo depende en gran medida de la forma en que varía el ruido de fondo del ambiente en el que se encuentra.

Los resultados muestran que para ambientes con un SNR alto, aumentar k_d representa una reducción de la FAR, el OVER y el NDS, pero para ambientes con SNR medio o bajo, utilizar un k_d disminuye el MSC con lo que se obtiene mayor cantidad de detecciones acertadas. A menor k_d mayor será la tasa de falsa alarma y ruido detectado como señal, pero también se mejorará el acierto con el que se detecta la señal de interés, es decir menor MSC y FEC.

5.3. Movimiento de la cámara

El movimiento de la cámara implica el control del movimiento del servomotor. Los resultados obtenidos y expuestos en esta sección provienen de la caracterización del servomotor y el algoritmo de control de la posición y la velocidad del mismo.

5.3.1. Control de posición angular

La *Sección 4.5* señala como se realizó la caracterización del servomotor para obtener una posición angular deseada, en seguida se muestran los resultados obtenidos de dicha caracterización en forma gráfica.

Los datos obtenidos y correspondientes a la *Figura 5.5* se obtuvieron sin colocar una carga mecánica al motor. El servomotor cuenta con un lazo cerrado de control interno, este lazo se encarga de mover al motor y realizar los ajustes necesarios hasta obtener la posición angular deseada independientemente de la carga que se le coloque. Por supuesto, no es posible rebasar los límites de potencia y "par" del motor. Las curvas de estas dos variables asociadas al servomotor utilizado se muestran en el *Capítulo 7*.

La *Figura 5.5* muestra los resultados obtenidos de cada una de las iteraciones realizadas durante la caracterización, *Sección 4.5*. Los círculos mostrados en la parte inferior de la *Figura 5.5* indican a que iteración pertenece cada conjunto de puntos. El eje horizontal de la gráfica representa la posición angular θ a la que se desplazaba el eje del servomotor y el eje vertical indica el ciclo de trabajo, D , de la correspondiente señal de PWM utilizada para mover al servomotor a la posición angular θ .

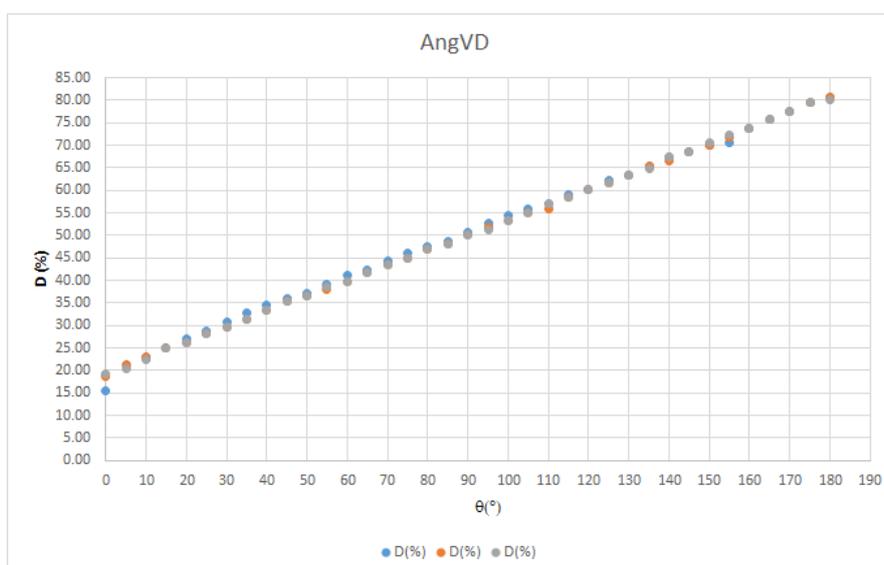


Figura 5.5: Resultados de tres iteraciones para obtener la posición angular

La gráfica de la *Figura 5.6* indica que la posición angular del servomotor es directamente proporcional al ciclo de trabajo y que sigue un comportamiento lineal en ese intervalo de posiciones angulares. La línea punteada es el modelo matemático del comportamiento obtenido mediante regresión lineal. Como se mencionó en la *Sección 4.5* este modelo se utilizó dentro del algoritmo de control para mover el servomotor.

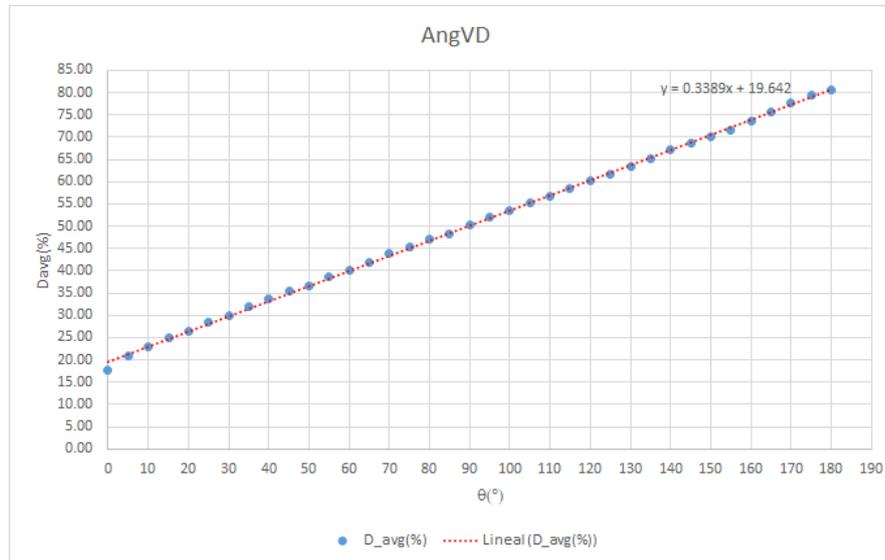


Figura 5.6: Promedio de las tres iteraciones para obtener la posición angular

5.3.2. Control de velocidad angular

El algoritmo de control del servomotor que se implementó en el DSP permite ajustar la posición y también la velocidad angular del mismo en el intervalo de $[10 - 170]^\circ$.

La velocidad con la que gira el motor depende de la carga que se tenga que mover y de la forma en que se realiza el control del movimiento. Como se describió en la *Subsección 4.5.2* el movimiento del servo se hace modificando iterativamente la posición angular, con lo que se recalcula y ajusta el ciclo de trabajo de la señal de control (PWM). Cada vez que se modifica la posición angular, y por ende el ciclo de trabajo, se da un periodo de tiempo de espera (retraso) entre cada modificación. La velocidad con la que se mueve el servomotor depende de la duración del retraso.

Se realizó un conjunto de pruebas para obtener la relación entre la duración del retraso y la velocidad angular del motor. Las pruebas realizadas consistieron en lo expuesto en seguida:

Se restringió el movimiento del motor en un intervalo de $[0 - 47]^\circ$. En las posiciones angulares extremas (0 y 47°) se colocaron dos detectores que producían un valor lógico de salida. Sobre el servomotor se montó una flecha conductora, la cual era perpendicular al eje de salida y se encontraba a la altura del mismo. Esta flecha estaba conectada a una fuente de $4.6V$ y las salidas de los detectores estaban conectadas a un compuerta OR [43]. Cuando la flecha hacía contacto físico con alguno de los detectores, la salida de la compuerta presentaba un valor lógico alto ("1"), cuando no hacía contacto físico se presentaba un valor lógico bajo ("0"). Mediante

software y utilizando el DSP se movió al servomotor a la posición inicial (0°) donde la flecha hacía contacto con uno de los detectores. Posteriormente, el servomotor se movía a la posición final (47°) donde se encontraba el otro detector. En el intervalo de posiciones angulares donde la flecha conductora no tenía contacto con los detectores, la salida se encontraba en un nivel bajo, es decir, a la salida de la compuerta se producía un pulso en bajo cuya duración era aproximadamente igual al tiempo que le tomaba al motor en realizar un recorrido entre los dos detectores. A partir del tiempo que duraba el pulso y al desplazamiento angular realizado se calcularon las velocidades angulares promedio con la que se movía el servomotor. Cuando el motor tenía como carga a la cámara y el retraso de tiempo era nulo, la velocidad angular del servomotor era $\omega = 3.17976 \text{ rad/s}$.

La gráfica de la *Figura 5.7 (a)* muestra la relación entre la duración del retraso y la velocidad angular promedio a la que se movía el motor cuando tenía acoplado la cámara como carga. Para la obtención de esta curva se fijó un incremento unitario en la posición y solamente se varió el retraso de tiempo. La curva punteada representa el modelo matemático con el que se aproximó el comportamiento de la velocidad en términos de la duración del retraso de tiempo. La gráfica de la *Figura 5.7 (b)* muestra la relación entre la duración del retraso y la velocidad angular promedio a la que se movía el motor utilizando escalas logarítmicas.

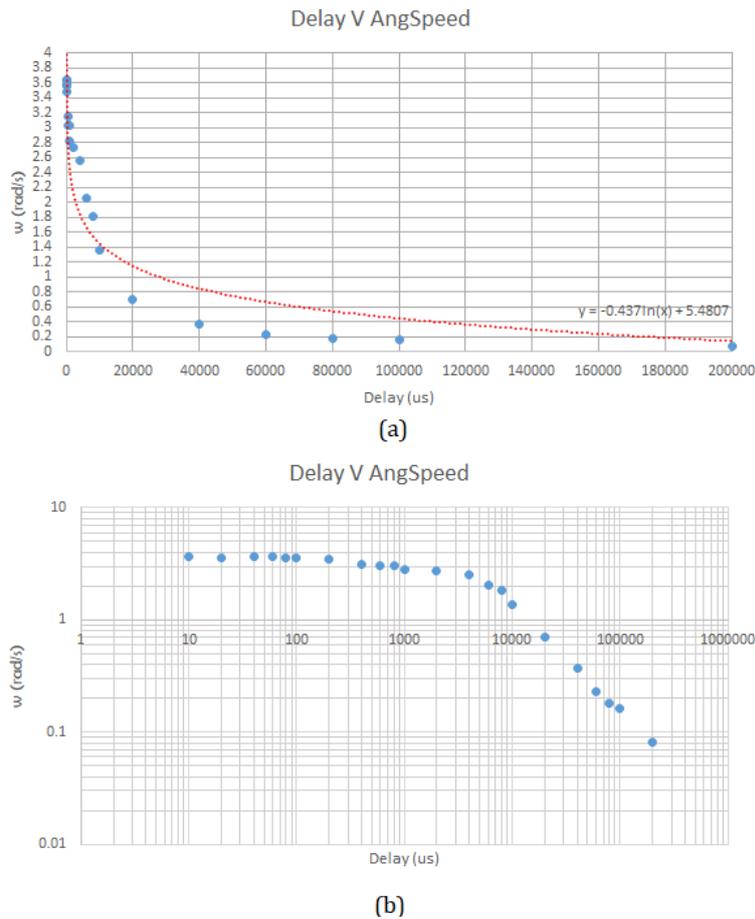
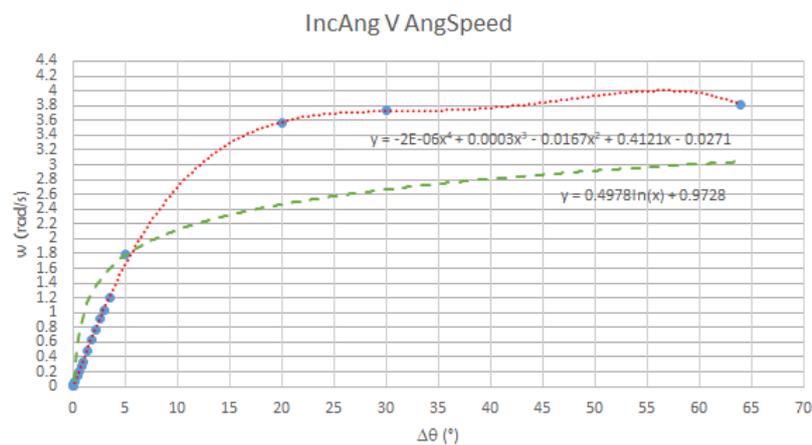


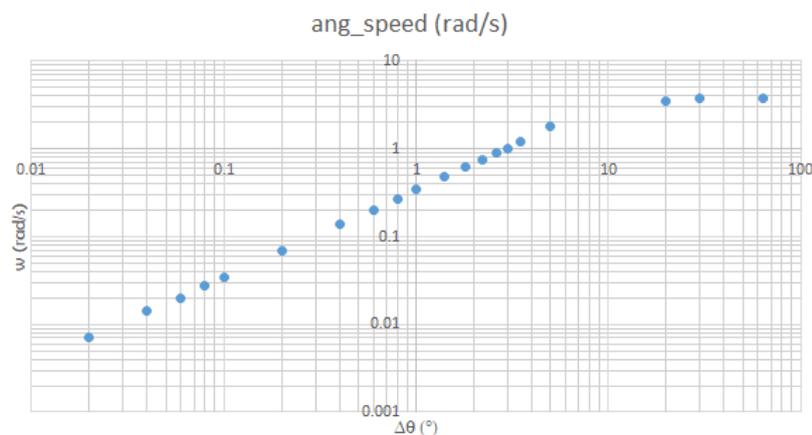
Figura 5.7: Velocidad angular en función del retraso de tiempo

La velocidad angular del motor también se podía controlar modificando el tamaño del paso o el incremento en la posición angular. Las pruebas realizadas para obtener la relación entre el incremento en la posición angular y la velocidad angular del motor se llevaron a cabo siguiendo la misma metodología que las hechas para obtener la relación entre el retraso y la velocidad angular; se utilizaron los mismos detectores y el mismo intervalo de desplazamiento angular $[0 - 47]^\circ$.

La gráfica de la *Figura 5.8 (a)* muestra la relación entre el incremento de la posición angular y la velocidad angular promedio a la que se movía el motor cuando tenía acoplado la cámara como carga. Para la obtención de esta curva se fijó un tiempo de retraso de 50ms y se varió el incremento con el que se modificaba la posición angular en cada iteración dentro del algoritmo de control. Las curvas punteadas corresponden a modelos matemáticos que aproximaban el comportamiento de la velocidad en términos del incremento en la posición. Se obtuvieron las mejores aproximaciones utilizando una función logarítmica y un polinomio de Taylor de cuarto orden. También se muestra la curva obtenida usando escalas logarítmicas *Figura 5.8 (b)*.



(a)



(b)

Figura 5.8: Velocidad angular en función del incremento de posición angular

Existen varios intervalos en los que el comportamiento de las curvas *Figura 5.7 (a)* y *Figura 5.8 (a)* se puede aproximar a un comportamiento lineal. La *Figura 5.9* muestra uno de estos intervalos para la curva de retraso contra velocidad angular y la *Figura 5.10* muestra un intervalo para la curva de incremento angular contra velocidad angular.

La elección de los intervalos presentados en la *Figura 5.9* y en la *Figura 5.10* se debe a que los valores utilizados en la realización del sistema de vigilancia se encontraban comprendidos en esos mismos.

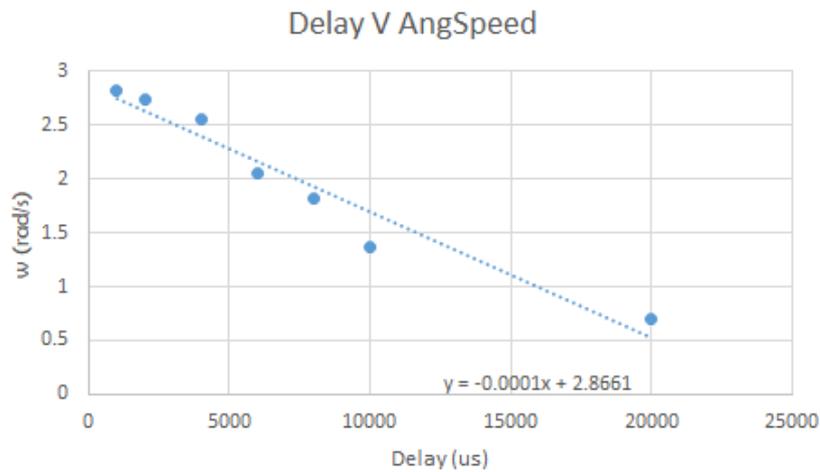


Figura 5.9: Intervalo de comportamiento lineal. Retraso (Delay)

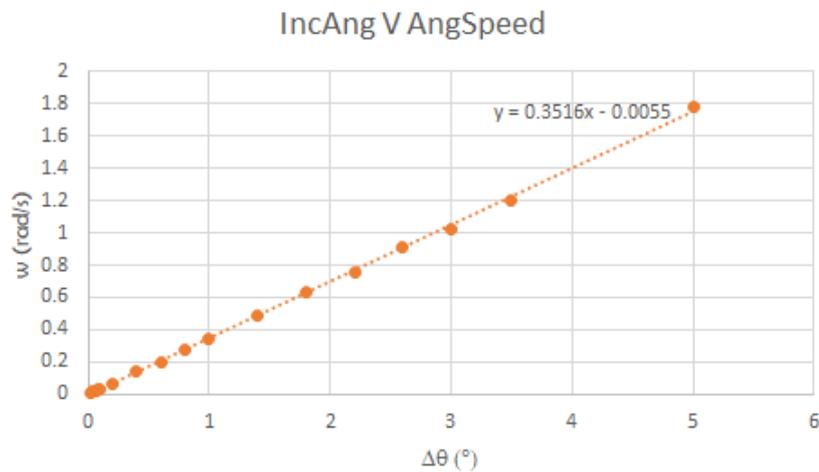


Figura 5.10: Intervalo de comportamiento lineal. Incremento angular ($\Delta\theta$)

La diferencia entre los intervalos a elegir es la pendiente de la recta de aproximación. En los escogidos se tiene una pendiente mayor en las rectas de aproximación, esta pendiente representa a la aceleración angular y al ser mayor se pueden tener mayores cambios en la velocidad angular. Además en estos intervalos la aproximación lineal es más exacta.

El algoritmo de control implementado mantenía un valor constante tanto para el incremento angular como para la duración del retraso de tiempo con lo que el motor giraba a una velocidad angular constante. Con base en las curvas obtenidas se puede modificar el algoritmo de control para generar movimientos acelerados, considerando que no se modifique la carga.

5.4. Resumen

En este capítulo se presentó el sistema de vigilancia completo y un conjunto de pruebas realizadas que permitían evaluar el desempeño de cada una de sus partes. Se mostró como se montó el sistema de vigilancia en su totalidad y como están conetadas las partes o etapas que lo conforman. Se presentaron los resultados obtenidos de las pruebas de desempeño y se obtuvieron modelos matemáticos que permitían controlar el movimiento del servomotor con la cámara de vigilancia.

Capítulo 6

Conclusiones

6.1. Conclusiones generales

El sistema de vigilancia diseñado permite detectar estruendos o ruidos fuertes dentro del entorno que se está monitoreando y con base en la detección se mueve una cámara para visualizar el ambiente a vigilar. El sistema está conformado por cuatro etapas y el desempeño del sistema completo depende principalmente de los algoritmos utilizados para la detección de actividad y control del servomotor.

El algoritmo de detección implementado se adapta a las condiciones de ruido ambiental y los resultados obtenidos mediante los parámetros objetivos de desempeño muestran que el sistema tiene un mejor funcionamiento en ambientes que cuentan con una SNR media o alta ($SNR > 4$). La HTER es inversamente proporcional a la SNR, es decir, a menor SNR mayor error presente en el sistema.

El funcionamiento del sistema está altamente afectado por las condiciones de ruido ambiental. El establecimiento de un correcto umbral de decisión que permite adaptarse a las condiciones ambientales es indispensable en el sistema.

El algoritmo de control de movimiento del servomotor permite obtener desplazamientos angulares precisos, lo cual es de gran utilidad cuando se necesita llegar a una posición angular específica. La velocidad del movimiento y el control de la misma dependerán de la carga que se coloque al motor, un control preciso de la velocidad se puede obtener manteniendo la carga constante y utilizando el motor en intervalos angulares de movimiento donde la velocidad presente un comportamiento lineal.

Los DSP permiten realizar aplicaciones en tiempo real debido a sus altas velocidades de procesamiento. El procesamiento de la señal se llevaba a cabo en intervalos cortos de tiempo, lo cual permitía al sistema operar en tiempo real a pesar de las limitaciones en memoria.

6.2. Trabajo a futuro

El hardware y software mostrado e implementado en este trabajo puede formar parte de otros sistemas más complejos o se pueden usar los algoritmos como parte de otros proyectos.

6.2.1. Para el algoritmo de detección de actividad acústica

La detección de actividad acústica usualmente es la primera etapa en sistemas de estimación de dirección ángulo de arribo o reconocimiento de voz y sonidos. Los algoritmos de detección de actividad son ampliamente utilizados en telefonía y otros sistemas de telecomunicaciones. Se puede modificar el algoritmo de detección de actividad para que utilice ventanas diferentes a la rectangular o utilizar el doble umbral de decisión y comparar el desempeño obtenido.

6.2.2. Para el algoritmo de control del servomotor

El algoritmo de control de posición y velocidad puede incorporarse en sistemas para mover mecanismos de forma precisa, incluyendo aplicaciones de robótica y antenas.

Se propone modificar el algoritmo para el motor sea capaz de producir movimientos acelerados y que se pueda controlar la posición a la que se desee mover al servomotor de forma remota.

6.2.3. Para el sistema de vigilancia completo

El sistema de vigilancia está limitado en la transmisión de video debido a que se conectó directamente la cámara al monitor LCD. Se propone agregar un sistema que permita procesar y transmitir el video inalámbricamente y que permita su visualización a través de una dirección IP. Además se propone la modificación de los circuitos de acondicionamiento para que ocupen menor espacio físico e integrar las fuentes de alimentación.

Capítulo 7

Anexos

7.1. Características generales del DSP TMS320F28377S

El TMS320F28377S pertenece a la familia de microcontroladores MCU C2000 de *Texas Instruments TM*. Los microcontroladores de la familia C2000 son de 32 bits y están optimizados para realizar aplicaciones de control en tiempo real; se enfocan y pueden usarse para el control de motores, inversores y dispositivos que involucran electrónica de potencia y para el sensado y procesamiento digital de señales.

Esta familia de MCUs tiene buses optimizados para la transmisión de datos y permite el uso de interrupciones generadas por el CPU y una gran diversidad de periféricos incluidos.

El núcleo de la familia C2000 es el CPU C28x, este es un CPU de 32 bits con aritmética de punto flotante que puede operar en frecuencias hasta 200 MHz. Además cuenta con varias unidades dedicadas que le permiten optimizar el procesamiento de datos. Las unidades dedicadas incluidas en el CPU son las siguientes:

- FPU (Unidad de punto flotante): Permite ejecutar instrucciones, operar registros y realizar cálculos en punto flotante con precisión simple, bajo el estándar 754 de IEEE.
- TMU (Unidad matemática trigonométrica): Permite acelerar y optimizar la ejecución de algoritmos con operaciones trigonométricas como la FFT y otras transformadas.
- VCU (Unidad Viterbi, complejos y CRC): Agrega registros e instrucciones para reducir el tiempo de proceso en operaciones con números complejos.

Los CPU de esta familia, además de tener las unidades de procesamiento dedicadas, también cuentan con un coprocesador acelerador en tiempo real (CLA). Este procesador también es de 32 bits punto flotante y puede trabajar a las mismas frecuencias que el CPU principal. El CLA responde a interrupciones de los periféricos y trabaja en paralelo al CPU principal.

Algunas de las características del CPU del DSP utilizado, TMS320F28377S, son las siguientes:

- Arquitectura tipo Harvard modificada.
- Tiene una unidad lógico aritmética (ALU) de 32 bits.
- Puede ejecutar algunas instrucciones de lectura y escritura en memoria y multiplicaciones acumuladas en un ciclo de reloj.
- Tiene un Pipeline de 8 niveles que permite el solapamiento de instrucciones en diferentes niveles de ejecución.
- Cuenta con 1MB (512kW) de memoria flash y 164 kB (82kW) de memoria RAM estática (SRAM).
- Permite la protección de código y registros.
- Tiene un módulo que permite configurar y manejar hasta 196 interrupciones producidas por los periféricos integrados o el CPU.

Los microcontroladores de esta familia cuentan con diversos periféricos analógicos y de control integrados para darle mayor funcionalidad al sistema. Algunos de ellos se listan a continuación:

- 16 ADCs configurables a 16 bits o a 12 bits, con entradas simples o diferenciales, bloques de post procesamiento y hasta 16 canales de conversión por ADC.
- 3 convertidores digital-analógico (DAC) con referencia ajustable.
- Periféricos de control mejorados: 12 módulos ePWM con salida dual, es decir, hasta 24 salidas de PWM de las cuales 16 se pueden configurar para operar en alta resolución. Seis módulos de captura mejorados (eCAP). Tres módulos de generación de pulsos de codificación en cuadratura (eQEP). Ocho canales de entrada a módulos de filtrado Sigma-Delta.
- Periféricos de comunicaciones: cuentan con soporte para USB 2.0 e interfaces para puertos paralelos universales (uPP) de 12 pines. Incluyen dos módulos CAN, tres puertos SPI de alta velocidad y dos puertos seriales multicanal (McBSPs). Dos interfaces I2C y 4 interfaces de comunicaciones seriales UART/SCI.
- Puertos de entrada/salida de propósito general (GPIOs).

La cantidad de periféricos disponibles para cada dispositivo varía de acuerdo al dispositivo utilizado y al empaquetado del mismo, pero incluyen la mayoría de los listados previamente. El DSP utilizado incluía todos los mencionados con ciertas restricciones en cada uno.

7.2. Curvas características del motor

La carga mecánica que se puede mover cuando se coloca sobre el eje de un motor está relacionada al "par" y a la potencia del mismo, además de otros factores mecánicos que influyen. A su vez, el "par" y la potencia dependen de la velocidad angular con la que gire el eje del motor.

Para cada motor, existe una curva específica de Par/velocidad angular y de Potencia/velocidad angular. El "par" es inversamente proporcional a la velocidad del eje de salida [17], es decir, existe un intercambio entre que tan rápido gira el eje del motor y cuanto "par" genera el motor. La curva de par/velocidad angular se puede obtener analíticamente utilizando las ecuaciones mostradas a continuación.

$$\tau_{mot} = \tau_s - \left(\frac{\tau_s}{\omega_n} \right) \omega \quad (7.1)$$

Donde τ_s es el "par" de arranque y se presenta cuando se energiza al motor después de estar en reposo y tener conectada la carga. En este punto el "par" es el máximo ($\tau_{mot} = \tau_s$) pero el eje del motor no se mueve ($\omega = 0$), ω_n es la velocidad angular sin carga y es la máxima velocidad angular alcanzada por el motor, en esta velocidad ($\omega = \omega_n$) el "par" en el eje de salida motor es nulo.

La *Ecuación 7.1* representa un modelo lineal con el que se puede hacer una buena aproximación al verdadero comportamiento. La *Figura 7.1* muestra la aproximación de la curva par/velocidad angular para el motor utilizado (HITECH HS-422 [37]); los valores de τ_s y ω_n se obtuvieron de la hoja de datos del motor [37].

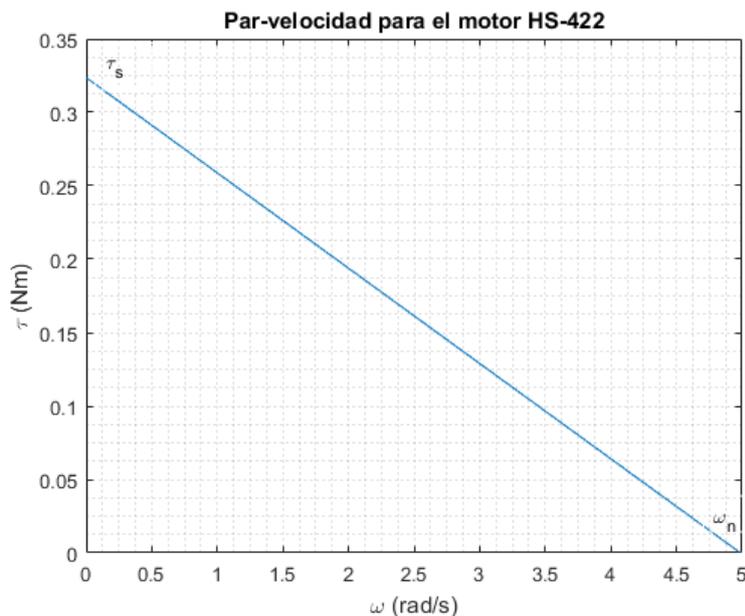


Figura 7.1: Curva característica de velocidad angular contra "par" para el motor utilizado

Esta curva representa adecuadamente el comportamiento del eje de salida del motor o una salida simple reducida mediante engranes. Si se agregan demás elementos mecánicos la curva cambia sus características.

La potencia mecánica producida por el motor se puede definir como el producto del "par" generado por el motor y la velocidad angular del eje de salida del mismo

$$P_{mot} = \tau_{mot} \omega \quad (7.2)$$

Si se sustituye la *Ecuación 7.1* en 7.2 se obtiene un modelo analítico de la potencia mecánica del motor en función de la velocidad angular del eje de salida del mismo:

$$P_{mot} = - \left(\frac{\tau_s}{\omega_n} \right) \omega^2 + \tau_s \omega \quad (7.3)$$

La *Figura 7.2* muestra la aproximación de la curva potencia/velocidad angular para el motor utilizado (HITECH HS-422 [37]).

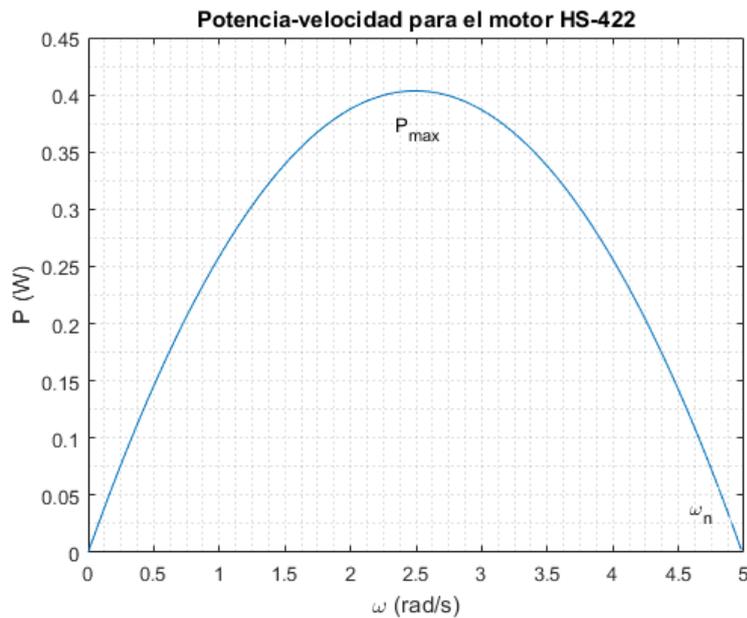


Figura 7.2: Curva característica de velocidad angular contra potencia para el motor utilizado

Se puede obtener la máxima potencia mecánica producida por el motor si se deriva la *Ecuación 7.3* y se aplica el criterio de la primera derivada. La máxima potencia se da cuando $\omega = \omega_n/2$, para el motor utilizado $P_{max} = 0.4034$ [W].

Glosario

- algoritmo** conjunto de instrucciones o pasos a seguir que permiten solucionar un problema, realizar cálculos y procesar datos. 1
- aliasing** traslape o repetición de componentes frecuenciales de la señal digitalizada que hace irreconocible un componente en frecuencia de la señal. 38
- carga** elemento del sistema que solicita potencia. 16
- ciclo de trabajo** razón entre el tiempo que una señal cuadrada periódica está activa y el periodo de la misma. 13
- corriente de encendido** corriente de entrada instantánea de magnitud elevada consumida por una máquina o equipo eléctrico al momento del encendido. 17
- detección de actividad acústica** técnica empleada en procesamiento de audio en la que se detecta e indica la presencia de actividad sonora o la ausencia de la misma. 5
- energía en tiempo corto** suma del cuadrado de cada muestra dentro del segmento en tiempo corto. Es análoga a la potencia de la señal en ese intervalo de tiempo. 41
- frame** bloques de corta duración que agrupan un conjunto de datos pertenecientes o parte de una señal. 40
- frecuencia de muestreo o frecuencia de Nyquist** tasa a la que se obtienen o toman las muestras de una señal. Tasa de muestras obtenidas por segundo. 37
- función de transferencia** relación entre la salida y la entrada de un sistema. 26
- lazo abierto** sistemas en los que el control no está afectado por la salida producida, en estos sistemas no se retroalimenta la salida ni se compara con la entrada. 7
- lazo cerrado** sistemas que obtienen la diferencia entre la señal de salida y de entrada y alimentan al controlador con ésta. La diferencia genera una señal de error que se busca reducir mediante acciones desencadenadas por el controlador. 7
- microcontroladores** dispositivos electrónicos de muy alto grado de integración que permiten ejecutar un gran conjunto de instrucciones lógicas y aritméticas para realizar tareas. Son de tamaño reducido, integrados en un solo chip y están conformados por un núcleo de procesamiento, elementos de memoria, buses y periféricos. 14

- muestreo** proceso de conversión de una señal en el tiempo continuo a una señal en el tiempo discreto. 37
- ondas mecánicas** perturbaciones en un medio a través del que viajan. Ondas que necesita un medio para su propagación. 5
- posición angular** magnitud angular que representa el ángulo que se forma entre el vector de posición de un cuerpo y un eje de referencia. 23
- procesamiento digital** es el tratamiento que se le da a datos representados de forma discreta. 2
- PWM** modulación por ancho de pulso. Forma de representar una señal analógica utilizando una señal digital. Modulación que utiliza señales cuadradas periódicas en las que se varía la duración del pulso cuadrado. 13
- relación señal a ruido** medida logarítmica que relaciona la potencia de la señal deseada y el ruido añadido a ésta. Relación entre la potencia de una señal y la potencia del ruido añadido a ésta. 69
- retroalimentación** recuperación de información acerca del proceso que se está controlando. Proceso en el que la salida de un sistema se retorna y compara con la entrada. 7
- servomotor** motor colocado dentro de un lazo de control cerrado (retroalimentado). 4
- señales** variable o cantidad física que provee información sobre el estado o evolución de un sistema o fenómeno físico y se puede modelar matemáticamente [44]. 2
- sistema** todo aquello utilizado para procesar señales. Se pueden describir a través de diversas formas: mediante su función de transferencia, respuesta en frecuencia, respuesta al impulso, usando variables de estados y más. 26
- tiempo real** sistema digital que opera bajo restricciones temporales y que presenta una respuesta antes de la restricción establecida. 2
- transductor** dispositivo que transforma una forma de energía a otra forma. 8
- umbral** valor mínimo de la energía de la señal a partir del cual se toma una decisión. 43
- velocidad angular** tasa de rotación alrededor de un eje usualmente expresada en revoluciones por minuto o radianes por segundo. 11
- voltaje de retroalimentación** voltaje asociado a la salida de un sistema que regresa y se compara con la entrada para generar un error. 24
- voltaje promedio** el cociente del área debajo de la señal de voltaje con respecto al tiempo. 16

7.3. Códigos de los programas

7.3.1. Código del sistema completo

```

/*
 * VADServo2Mod.c
 *
 * Created on: 02/05/2020
 * Author: Yves
 */

#include "F28x_Project.h"

#define BUFFER_SIZE      256                //Tamaño ventana
      (buffer) de datos
#define FRAMES_NOISE_IN  80                //Número de ventanas
      iniciales de ruido (~100mS para Fs=48kHz)
float x[BUFFER_SIZE];                    //Vector para almacenar
      datos correspondientes a ventanas de la señal de entrada
float resultADC;                          //Variable para convertir
      resultados del ADC a punto flotante
Uint16 bufferIndex=0;                     //Índice para llenar
      ventana de datos
volatile Uint16 bufferFull=1;             //Bandera de término de
      llenado de ventana
volatile Uint16 VADFlag = 0;              //Para señal de
      detección de actividad

//
// Prototipos de funciones
//
void configInit(void);                    //Configura WatchDog,
      reloj de sistema (PLL) y memorias
void configADCA(float Fs);                //Configura uso del ADC
void configEPWM2_Samp(float Fs);          //Configura módulo de PWM
      para disparar conversión de ADC
void configEPWM7(float F_PWM,unsigned char COUNT_DIR); //Configura módulo de PWM
      para señal de control del servomotor
void initialADCA(Uint16 channel);          //Configura SOC del ADC
void ADCAOInt(void);                       //Configura
      interrupciones del ADC
void startADCA(void);                       //Inicia conversión del
      ADC
float energyFrame(float *x);               //Ventana de datos en
      punto flotante
void servoTurn(float POS);                 //Control de posición del
      servomotor

```

```

interrupt void adca1_isr(void); //Subrutina de
    interrupcion para obtener resultados de conversion y mover a arreglo

void main(void){

    float Fsamp = 48000; //Frecuencia de muestreo
        estandar para audio
    float E_r = 0; //Para calcular energia
        por ventana
    float p = 0.25; //Parametros algoritmo VAD
    float kd = 0.2;
    float Thr,ThrF,eFrame;
    float F_PWM7 = 320; //Frecuencia senial de
        control del servo
    float POSITION = 10; //Establece posicion
        angular inicial
    int i=0;
    unsigned char COUNT_DIR = 0x0; //0x0:UP, 0x1:DOWN,
        0x2:UP/DOWN, 0x3:FREEZE

    configInit(); //Deshabilita WatchDog,
        habilita perifericos, calibra ADC, configura reloj del sistema a 190 MHz

    InitGpio(); //Escribe 0 en todos los
        registros de control y de datos de GPIOs (reset)

    InitEPwm7Gpio(); //Deshabilita PUR de
        GPIO12 y GPIO13.Configura GPIO12 y GPIO13 como EPWM7A y EPWM7B

    DINT; //Deshabilita
        interrupciones de forma global

    InitPieCtrl(); //Inicializa los
        registros de control del PIE,

    IER = 0x0000; //Deshabilita las
        interrupciones del CPU y limpia las banderas de interrupcion
    IFR = 0x0000;

    InitPieVectTable(); //Inicializa la tabla de
        vectores del PIE para poder ligar ISR

    /*****
    * Configuracion modulo ePWM7
    *****/
    configEPWM7(F_PWM7,COUNT_DIR); //Configuracion inicial
        del modulo, ver funcion a detalle

    configADCA(Fsamp); //Configura reloj para
        conversion, resolucion, modo del ADC y senial de PWM para disparo

```

```

initialADCA(2); //Configura canal a
                convertir, ventana de adquisicion y disparo del ADC

ADCA0Int(); //Configura
            interrupciones del ADC: liga subrutina de interrupcion, habilita INT y

CpuSysRegs.PCLKCRO.bit.TBCLKSYNC = 1; //Bases de tiempo de
                todos los modulos PWM empiezan a contar, habilita inicio de conteo PWMs

EINT; //Habilita interrupciones
    de forma global
ERTM; //Habilita interrupciones
    en tiempo real

/*****
 * Establecimiento de umbral de ruido inicial
 *****/

for(i=0;i<FRAMES_NOISE_IN;i++){

    startADCA(); //Inicia operacion ADC:
                Habilita cuenta PWM y pulso de EOC

    /*****
     * Espera mientras ePWM causa conversiones del ADC, las cuales generan
     * interrupciones que llenan buffer de datos,
     * cuando el buffer se llena se activa bandera bufferFull
     *****/

    while(!bufferFull);
    bufferFull = 0; //Limpia bandera de
                buffer lleno

    E_r = energyFrame(x) + E_r; //Calcula energia de la
                ventana y acumula para promediar al final
}

Thr = E_r/FRAMES_NOISE_IN; //Establece umbral de
                ruido inicial

while(1){ //Ciclo infinito,
    mantiene trabajando al DSP
    bufferIndex=0;

```

```

while(!bufferFull); //Llena buffer o ventana
  de datos
bufferFull = 0; //Limpia la bandera de
  buffer lleno
eFrame = energyFrame(x); //Calcula energia de
  ventana

/*****
 * Decision de deteccion de actividad en base a la energia calculada
 *****/

if(eFrame>Thr){ //Ventana con actividad

  VADFlag = 1; //Actividad detectada
  EPwm7Regs.AQCTLA.bit.ZRO = AQ_SET; //Salida PWM en alto
  cuando contador llega a 0, TBCTR=0
  EPwm7Regs.AQCTLA.bit.CAU = AQ_CLEAR; //Salida PWM en bajo
  cuando se da un evento, es decir, salida en bajo cuando TBCTR=CMPA
}else{ //Ventana sin actividad

  ThrF = (1-p)*Thr+(p*eFrame); //Establece nuevo umbral
  de decision o de ruido
  Thr = ThrF+ThrF*kd;
}

if(VADFlag==1){ //Mueve motor si existe
  actividad

  EPwm2Regs.ETSEL.bit.SOCAEN = 0; //Deshabilita SOCA
  (conversion ADC)
  EPwm2Regs.TBCTL.bit.CTRMODE = 3; //Congela contador

/*****
 * Movimiento del servomotor
 *****/
while(POSITION<170){ //Movimiento a traves de
  modificar posicion iterativamente

  servoTurn(POSITION); //Ajusta ciclo de trabajo
  D de PWM

  POSITION = POSITION + 1; //Incrementa pos angular

  DELAY_US(15000); //Retraso para ajustar D
  del PWM
}

DELAY_US(200000);

```

```

        while(POSITION>10){

            POSITION = POSITION - 1;

            servoTurn(POSITION);

            DELAY_US(15000);
        }

        VADFlag = 0; //Establece fin del
                    movimiento del motor, reinicia senial de actividad

        startADCA(); //Inicia SOCA (conversion
                    ADC)

        EPwm7Regs.AQCTLA.bit.ZRO = AQ_NO_ACTION; //No genera salida de PWM
        EPwm7Regs.AQCTLA.bit.CAU = AQ_NO_ACTION;

    }
}

void configInit(void){
    EALLOW; //Habilita
            escritura a registros protegidos

    memcpy(&RamfuncsRunStart, &RamfuncsLoadStart, (size_t)&RamfuncsLoadSize);
    //Copia configuracion de Flash a RAM, permite usar DELAY_US(A)

    ClkCfgRegs.PERCLKDIVSEL.bit.EPWMCLKDIV = 1; //Por
            defecto 1h:EPWMCLK=PLLSYSCLK/2,0h:EPWMCLK=PLLSYSCLK

    InitSysCtrl(); //Funcion
                    modificada, configura reloj del sistema a 190MHz y usa el OSC22 del MCU

    EDIS;
}

void configADCA(float Fs){
    EALLOW;

    AdcaRegs.ADCCTL2.bit.PRESCALE = 7;
    //Prescalador del reloj de ADC. SYSCLK/4 = 190MHz/4.5=42.22 MHz
    AdcSetMode(ADC_ADCA, ADC_RESOLUTION_12BIT, ADC_SIGNALMODE_SINGLE); //Usa ADCA,
    resolucion de 12 bits, modo simple

    AdcaRegs.ADCCTL1.bit.INTPULSEPOS = 1;
    //Interrupcion al final de la conversion. Es decir, cuando el resultado esta

```

```

    en memoria

AdcaRegs.ADCCTL1.bit.ADCPWDNZ = 1;           //Enciende
    ADC

DELAY_US(1000);                               //Retraso
    1ms para encender ADC

configEPWM2_Samp(Fs);
    //Configuracion de PWM2 para disparar SOC ADC

EDIS;
}

void configEPWM2_Samp(float Fs){

    Uint16 EPWM2_TIMER_TBPRD = 0;
    Uint16 EPWM2_CMPA = 0;
    float TBCLK = 95000000;                   //Modificar
        de acuerdo a valores de registros de TBCTL posteriores

    EALLOW;
    EPwm2Regs.TBCTL.bit.CTRMODE = 0x0;        //Conteo
        ascendente
    EPwm2Regs.TBCTL.bit.HSPCLKDIV = TB_DIV1;
        //TBCLK=EPWMCLK/(HSPCLKDIV*CLKDIV)=95MHz/[(1*1)]=95MHz/(1*2)=95MHz
    EPwm2Regs.TBCTL.bit.CLKDIV = TB_DIV1;     //Base de
        tiempo trabaja a 95MHz

    //Configura GPIOs para ver PWM que dispara ADC, es decir verifica Fs
    GpioCtrlRegs.GPAPUD.bit.GPIO2 = 1;
        //Deshabilita pull-ups en GPIO2

    GpioCtrlRegs.GPAMUX1.bit.GPIO2 = 1;      //Configura
        GPIO2 como EPWM2A

    //Configuracion de evento(s) que dispara(n) conversion
    EPwm2Regs.ETSEL.bit.SOCAEN = 0;          //
        Deshabilita pulso de SOC en ADCA
    EPwm2Regs.ETSEL.bit.SOCASEL = 4;         // Habilita
        conversion cuando TBCTR=CMPA
    EPwm2Regs.ETPS.bit.SOCAPRD = 1;         // Genera
        pulso de conversion con 1 evento

    EPWM2_TIMER_TBPRD = (Uint16) (TBCLK/Fs)-1; //Calcula
        periodo de senial PWM
    EPWM2_CMPA = (Uint16) ((TBCLK/Fs)-1)*0.5;

```

```

EPwm2Regs.CMPA.bit.CMPA = EPWM2_CMPA;           // PWM
    D=50. Cuando realiza la comparacion se habilita la conversion del ADC
EPwm2Regs.TBPRD = EPWM2_TIMER_TBPRD;           //
    Configura frecuencia de muestreo
EPwm2Regs.AQCTLA.bit.ZRO = AQ_SET;             // Salida
    en alto cuando contador llega a 0, TBCTR=0
EPwm2Regs.AQCTLA.bit.CAU = AQ_CLEAR;           //Salida en
    bajo cuando se da un evento, es decir, salida en bajo cuando TBCTR=CMPA

EPwm2Regs.TBCTL.bit.CTRMODE = 3;               // Congela
    cuenta
EDIS;
}

//
// Configura ventana de adquisicion y disparo del ADC
//
void initialADCA(Uint16 channel){
    Uint16 acqps=77;                             //Variable
        para ventana de adquisicion
    /*****
    * Determina la duracion de la ventana de adquisicion
    (tiempo de carga del capacitor de muestreo)
    basandose en la resolucion (12 bits). Para el circuito de
    entrada utilizado y esta
    tarjeta t=411.4957ns,aqps=20-1 ciclos de reloj
    *****/

    EALLOW;
    AdcaRegs.ADCSOCOCTL.bit.CHSEL = channel;      //SOCO
        convierte pin especificado por channel
    AdcaRegs.ADCSOCOCTL.bit.ACQPS = acqps;        //Ventana
        de muestreo de 16 ciclos de reloj
    AdcaRegs.ADCSOCOCTL.bit.TRIGSEL = 7;         //Disparo
        de conversion mediante EPWM2
    EDIS;
}

void ADCAOInt(void){

    EALLOW;
    PieVectTable.ADCA1_INT = &adca1_isr;        //Liga
        subrutina de interrupcion del ADC_A a la tabla de interrupciones

    PieCtrlRegs.PIEIER1.bit.INTx1 = 1;          //Habilita
        la interrupcion del PIE INT1.1
    AdcaRegs.ADCINTSEL1N2.bit.INT1SEL = 0;       //EOCO
        dispara interrupcion del ADC_A
    AdcaRegs.ADCINTSEL1N2.bit.INT1E = 1;         //Habilita
        INT1 del ADC_A

```

```

AdcaRegs.ADCINTFLGCLR.bit.ADCINT1 = 1; //Limpia
    bandera de interrupcion

IER |= M_INT1; //Habilita
    interrupciones del grupo 1, donde se encuentra la INT1 del ADC_A

EDIS;

}

//Inicia operacion del ADCA
void startADCA(){
    EPwm2Regs.ETSEL.bit.SOCAEN = 1; //Habilita pulso de
        SOC en ADCA
    EPwm2Regs.TBCTL.bit.CTRMODE = 0; //Descongela cuenta de
        TB, conteo ascendente
}

void configEPWM7(float F_PWM,unsigned char COUNT_DIR){

    Uint16 EPWM7_TIMER_TBPRD = 0;
    float MAX_FPWM = 666;
    float TBCLK = 11875000;

    if ( F_PWM > MAX_FPWM ) { //F_maxServo = 666
        Hz.Regresa si F_PWM
        return;
    }

    EPwm7Regs.TBCTL.bit.CTRMODE = COUNT_DIR; //Conteo ascendente
    EPwm7Regs.TBCTL.bit.HSPCLKDIV = TB_DIV2;
        //TBCLK=EPWMCLK/(HSPCLKDIV*CLKDIV)=95MHz/[(2*4)]=95MHz/(2*4)=11.875MHz
    EPwm7Regs.TBCTL.bit.CLKDIV = TB_DIV4;

    EPWM7_TIMER_TBPRD = (Uint16) (TBCLK/F_PWM)-1;

    EPwm7Regs.TBPRD = EPWM7_TIMER_TBPRD; //Define el periodo
        del contador de base de tiempo (TBPRD)

        //f_pwm =
            TBCLK/(TBPRD+1) =
            11.25MHz/35156 =
            333 Hz

    EPwm7Regs.TBCTR = 0x0000; //Reinicia contador
    EPwm7Regs.CMPA.bit.CMPA = 0x0 ; //Ajusta valor de
        comparador para modificar ciclo de trabajo

        //Ciclo de trabajo
            arbitrario

    EPwm7Regs.AQCTLA.bit.ZRO = AQ_NO_ACTION; //Salida en alto
        cuando contador llega a 0, TBCTR=0

```

```

EPwm7Regs.AQCTLA.bit.CAU = AQ_NO_ACTION;           //Salida en bajo
              cuando se da un evento, es decir, salida en bajo cuando TBCTR=CMPA
}

float energyFrame(float *x){                          //Funcion para
  calcular la energia de una ventana
  int j;
  float E=0.0;
  float N;

  N = (float)BUFFER_SIZE;

  for(j=0;j<BUFFER_SIZE;j++){                        //Calcula energia de
    la ventana de datos
    E=x[j]*x[j]+E;
  }

  E = E/N;                                           //Calcula energia
  promedio

  return E;
}

void servoTurn(float POS){

  Uint16 EPWM7_CMPA = 0;
  float PWM_PRD = 0;
  float tH = 0;
  float D = 0;
  float m = 0.003389442;                             //Constantes obtenidos
              mediante modelo de regresion lineal
  float b = 0.19641706;

  PWM_PRD =(float) EPwm7Regs.TBPRD;                 //Recalcula y ajusta
              ciclo de trabajo de senial de PWM
  D = m*POS + b;
  tH = PWM_PRD*D;
  EPWM7_CMPA = (Uint16) tH;

  //EPwm7Regs.TBCTR = 0x0000;                       //Reinicia contador
  EPwm7Regs.CMPA.bit.CMPA = EPWM7_CMPA;            //Ajusta valor de
              comparador para modificar ciclo de trabajo
  EPwm7Regs.AQCTLA.bit.ZRO = AQ_SET;               //Salida en alto
              cuando contador llega a 0, TBCTR=0
  EPwm7Regs.AQCTLA.bit.CAU = AQ_CLEAR;             //Salida en bajo
              cuando se da un evento, es decir, salida en bajo cuando TBCTR=CMPA
}

```

```

interrupt void adca1_isr(void){

    resultADC = (float) AdcaResultRegs.ADCRESULT0;           //Lee valores
        resultados de la conversion analogico-digital
    x[bufferIndex++] = (resultADC*3/4096)-1.6;               //Convierte valores
        del ADC a valores de voltaje representado en fp

    if(BUFFER_SIZE <= bufferIndex)                          //Revisa si se lleno
        el buffer o ventana, activa bandera si se llena
    {
        bufferIndex = 0;
        bufferFull = 1;
    }

    AdcaRegs.ADCINTFLGCLR.bit.ADCINT1 = 1;                  //Limpia bandera de
        interrupcion

    //
    // Revisa si ocurrio sobreflujo
    //
    if(1 == AdcaRegs.ADCINTOVF.bit.ADCINT1)
    {
        AdcaRegs.ADCINTOVFCLR.bit.ADCINT1 = 1;              //Limpia bandera de
            sobreflujo de INT1
    }

    PieCtrlRegs.PIEACK.bit.ACK1 = 1;                        //Limpia acknowledge
        de interrupcion, permite que se propaguen mas interrupciones
}

```

7.3.2. Código para pruebas del detector

```

/*
 * pruebaVAD.c
 *
 * Created on: 27 abr. 2020
 * Author: Yves
 */

#include "F28x_Project.h"

#define BUFFER_SIZE      256                               //Tamaño ventana (buffer)
    de datos
#define FRAMES_NOISE_IN  60                               //Número de ventanas
    iniciales de ruido (320ms para Fs=48kHz)
#define FRAMES_TEST_5S   938                             //Número de ventanas para
    prueba VAD (5s aprox para Fs=48kHz)

```

```

#define FRAMES_VAD_BEGIN    563                //Numero ventanas inicio de
    senial (3s aprox para Fs=48kHz)
#define FRAMES_VAD_END      750                //Numero ventanas fin
    senial (4s aprox para Fs=48kHz)

float x[BUFFER_SIZE];                          //Vector para almacenar
    datos correspondientes a ventanas de la senial de entrada
float resultADC;                                //Para convertir resultado
    ADC a fp
Uint16 bufferIndex=0;                          //Indice para llenar
    ventana de datos
volatile Uint16 bufferFull=1;                  //Bandera de termino de
    llenado de ventana
volatile Uint16 VADFlag = 0;                  //Para senial de deteccion
    de actividad

void configInit(void);                          //Configura WatchDog,
    reloj de sistema (PLL) y memorias
void configADCA(float Fs);                     //Configura uso del ADC
void configEPWM2_Samp(float Fs);               //Configura modulo de PWM
    para disparar conversion de ADC
void initialADCA(Uint16 channel);              //Configura SOC del ADC
void ADCAOInt(void);                           //Configura
    interrupciones del ADC
void startADCA(void);                          //Inicia conversion del
    ADC
float energyFrame(float *x);                   //Ventana de datos en
    punto flotante

interrupt void adca1_isr(void);                //Subrutina de
    interrupcion para obtener resultados de conversion y mover a arreglo

void main(void){
    float Fsamp = 48000;                        //Frecuencia de muestreo
        estandar para audio
    float E_r = 0;                              //Para calcular energia
        por ventana
    float p = 0.25;                             //Parametros algoritmo VAD
    float kd = 0.2;
    float Thr,ThrF,eFrame;

    Uint32 VADPosCtr = 0;                       //Contador de frames con
        actividad detectados
    Uint32 VADNegCtr = 1;                       //Contador de frames sin
        actividad detectados
    Uint32 VADPosArray[170];                   //Guarda cuenta TIMER
        cuando se da el flanco de subida
    Uint32 VADNegArray[170];                   //Guarda cuenta TIMER
        cuando se da el flanco de bajada

```

```

int i=0; //Para ciclo
int ArrayIndexPos = 0; //Indice arreglo de
    tiempos (cuentas TIMER) cuando hay flancos de subida
int ArrayIndexNeg = 0; //Indice arreglo de
    tiempos (cuentas TIMER) cuando hay flancos de bajada

for(i=0;i<128;i++){ //Inicializacion de
    arreglos
    VADPosArray[i]=0;
    VADNegArray[i]=0;
}

configInit(); //Deshabilita WatchDog,
    enciende perifericos, calibra ADC, configura reloj del sistema a 190 MHz

InitGpio(); //Escribe 0 en todos los
    registros de control y de datos de GPIOs (reset)

EALLOW;
GpioCtrlRegs.GPADIR.bit.GPIO12 = 1; //Configura GPIO12 como
    salida
EDIS;

DINT; //Deshabilita
    interrupciones de forma global

InitPieCtrl(); //Inicializa los
    registros de control del PIE

IER = 0x0000; //Deshabilita las
    interrupciones del CPU y limpia las banderas de interrupcion
IFR = 0x0000;

InitPieVectTable(); //Inicializa la tabla de
    vectores del PIE para poder ligar ISR

InitCpuTimers(); //Inicializa Timers de la
    tarjeta: provee senial de reloj, recarga periodo y congela cuenta

ConfigCpuTimer(&CpuTimer0, 190, 5500000); //Configura al Timer0 con
    un periodo de 5.5s, freq=190MHz,PRD=5 500 000 uS

CpuTimer0Regs.TCR.all = 0x0031; //Deshabilita INT del
    timer, detenido por software, recarga PRD y congela cuenta

configADCA(Fsamp); //Configura reloj para
    conversion, resolucion, modo del ADC y senial de PWM para disparo del ADC

```

```

initialADCA(2); //Configura canal a
                convertir, ventana de adquisicion y disparo del ADC

ADCA0Int(); //Configura
            interrupciones del ADC: liga subrutina de interrupcion, habilita INT y

CpuSysRegs.PCLKCRO.bit.TBCLKSYNC = 1; //Bases de tiempo de
                todos los modulos PWM empiezan a contar, habilita inicio de conteo PWMs

EINT; //Habilita interrupciones
    de forma global
ERTM; //Habilita interrupciones
    en tiempo real

/*****
 * Establecimiento de umbral de ruido inicial
 *****/

for(i=0;i<FRAMES_NOISE_IN;i++){

    startADCA(); //Inicia operacion ADC:
                Habilita cuenta PWM y pulso de EOC

    /*****
     * Espera mientras ePWM causa conversiones del ADC, las cuales generan
     * interrupciones que llenan buffer de datos,
     * cuando el buffer se llena se activa bandera bufferFull
     *****/

    while(!bufferFull);
    bufferFull = 0; //Limpia bandera de
                buffer lleno

    E_r = energyFrame(x) + E_r; //Calcula energia de la
                ventana y acumula para promediar al final
}

Thr = E_r/FRAMES_NOISE_IN; //Establece umbral de
                ruido inicial

GpioDataRegs.GPACLEAR.bit.GPIO12 = 1; //GPIO12 con salida en
                nivel bajo
CpuTimer0Regs.TCR.all &= 0xFFCF; //Empieza cuenta del TIMER

/*****
 * Se produce una senial digital a traves del GPIO12 el cual se conecta a un
    zumbador.
 * La senial dura 5s, los primeros 3s la senial se encuentra en nivel bajo
 *****/

```

```

* Posteriormente la senial se mantiene en nivel alto durante 1s
* Finalmente la senial regresa a un nivel bajo el tiempo restante (1s)
*****/

for(i=0;i<FRAMES_TEST_5S;i++){
    bufferIndex=0; //Reinicia indice de
    buffer, regresa al inicio del buffer o ventana

    while(!bufferFull); //Llena buffer o
    ventana de datos
    bufferFull = 0; //Limpia la bandera de
    buffer lleno
    eFrame = energyFrame(x); //Calcula energia de
    ventana

/*****
* La variable i indica el numero de ventanas de la senial que se han
    procesado.
* Es decir, el tiempo que ha durado la senial
* Cuando el numero de ventanas sobrepasa una duracion de 3s pero no ha
* alcanzado una duracion de 4s se produce una salida en alto en el GPIO12.
* Todo el demas tiempo de la senial la salida en el GPIO12 se mantiene en
    bajo
*****/

if((i>=FRAMES_VAD_BEGIN)&&(i<FRAMES_VAD_END)){
    GpioDataRegs.GPASET.bit.GPIO12 = 1; //GPIO12 en alto
}else{
    GpioDataRegs.GPACLEAR.bit.GPIO12 = 1; //GPIO12 en bajo
}

if(eFrame>Thr){ //Ventana con actividad

    CpuTimer0Regs.TCR.all |= 0x0010; //Congela cuenta
    VADPosCtr++; //Incrementa contador de
    frames con actividad en senial de deteccion

/*****
* Si se detecta actividad se guarda el valor de la cuenta del TIMER
    cuando se dio
* la deteccion de la primera ventana con actividad, es decir un flanco
    de subida
* en la senial de deteccion
*****/
if(VADPosCtr==1){
    VADPosArray[ArrayIndexPos++] = CpuTimer0Regs.TIM.all; //Guarda valor
    de la cuenta del TIMER

```

```

        VADNegCtr = 0; //Reinicia contador de
            ventanas sin actividad
        CpuTimer0Regs.TCR.all &= 0xFFCF; //Descongela contador,
            continua cuenta
    }else{
        VADNegCtr = 0; //Mantiene contador de
            ventanas sin actividad en 0
        CpuTimer0Regs.TCR.all &= 0xFFCF; //Descongela contador,
            continua cuenta
    }
    /*****
    * Si no se detecta actividad se guarda el valor de la cuenta del TIMER
    * cuando se dio
    * la deteccion de la primera ventana sin actividad, es decir un flanco
    * de bajada
    * en la senial de deteccion
    *****/
}else{
    CpuTimer0Regs.TCR.all |= 0x0010; //Congela cuenta
    VADNegCtr++; //Incrementa contador de
        frames sin actividad en senial de deteccion

    if(VADNegCtr==1){
        VADNegArray[ArrayIndexNeg++] = CpuTimer0Regs.TIM.all; //Guarda valor
            de la cuenta del TIMER
        VADPosCtr = 0; //Reinicia contador de
            ventanas con actividad
        CpuTimer0Regs.TCR.all &= 0xFFCF;
    }else{
        VADPosCtr = 0;
        CpuTimer0Regs.TCR.all &= 0xFFCF;
    }
    ThrF = (1-p)*Thr+(p*eFrame); //Establece nuevo umbral
        de decision o de ruido
    Thr = ThrF+ThrF*kd;

}

asm(" ESTOPO"); //Detiene ejecucion al
    termino de la senial

}

void configInit(void){
    EALLOW; //Habilita
        escritura a registros protegidos

    memcpy(&RamfuncsRunStart, &RamfuncsLoadStart, (size_t)&RamfuncsLoadSize);
    //Copia configuracion de Flash a RAM, permite usar DELAY_US(A)

```

```

ClkCfgRegs.PERCLKDIVSEL.bit.EPWMCLKDIV = 1; //Por
    defecto 1h:EPWMCLK=PLLSYSCLK/2,0h:EPWMCLK=PLLSYSCLK

InitSysCtrl(); //Funcion
    modificada, configura reloj del sistema a 190MHz y usa el OSC22 del MCU

EDIS;
}

void configADCA(float Fs){
    EALLOW;

    AdcaRegs.ADCCTL2.bit.PRESCALE = 7;
        //Prescalador del reloj de ADC. SYSCLK/4 = 190MHz/4.5=42.22 MHz
    AdcSetMode(ADC_ADCA, ADC_RESOLUTION_12BIT, ADC_SIGNALMODE_SINGLE); //Usa ADCA,
        resolucion de 12 bits, modo simple

    AdcaRegs.ADCCTL1.bit.INTPULSEPOS = 1;
        //Interrupcion al final de la conversion. Es decir, cuando el resultado esta
        en memoria

    AdcaRegs.ADCCTL1.bit.ADCPWDNZ = 1; //Enciende
        ADC

    DELAY_US(1000); //Retraso
        1ms para encender ADC

    configEPWM2_Samp(Fs);
        //Configuracion de PWM2 para disparar SOC ADC

    EDIS;
}

void configEPWM2_Samp(float Fs){

    Uint16 EPWM2_TIMER_TBPRD = 0;
    Uint16 EPWM2_CMPA = 0;
    float TBCLK = 95000000; //Modificar
        de acuerdo a valores de registros de TBCTL posteriores

    EALLOW;
    EPwm2Regs.TBCTL.bit.CTRMODE = 0x0; //Conteo
        ascendente
    EPwm2Regs.TBCTL.bit.HSPCLKDIV = TB_DIV1;
        //TBCLK=EPWMCLK/(HSPCLKDIV*CLKDIV)=95MHz/[(1*1)]=95MMHz/(1*2)=95MHz
    EPwm2Regs.TBCTL.bit.CLKDIV = TB_DIV1; //Base de
        tiempo trabaja a 95MHz

```

```

//Configura GPIOs para ver PWM que dispara ADC, es decir verifica Fs
GpioCtrlRegs.GPAPUD.bit.GPIO2 = 1;
    //Deshabilita pull-ups en GPIO2

GpioCtrlRegs.GPAMUX1.bit.GPIO2 = 1;                                     //Configura
    GPIO2 como EPWM2A

//Configuracion de evento(s) que dispara(n) conversion
EPwm2Regs.ETSEL.bit.SOCAEN = 0;                                         //
    Deshabilita pulso de SOC en ADCA
EPwm2Regs.ETSEL.bit.SOCASEL = 4;                                         // Habilita
    conversion cuando TBCTR=CMPA
EPwm2Regs.ETPS.bit.SOCAPRD = 1;                                         // Genera
    pulso de conversion con 1 evento

EPWM2_TIMER_TBPRD = (Uint16) (TBCLK/Fs)-1;                               //Calcula
    periodo de senial PWM
EPWM2_CMPA = (Uint16) ((TBCLK/Fs)-1)*0.5;

EPwm2Regs.CMPA.bit.CMPA = EPWM2_CMPA;                                   // PWM
    D=50. Cuando realiza la comparacion se habilita la conversion del ADC
EPwm2Regs.TBPRD = EPWM2_TIMER_TBPRD;                                     //
    Configura frecuencia de muestreo
EPwm2Regs.AQCTLA.bit.ZRO = AQ_SET;                                       // Salida
    en alto cuando contador llega a 0, TBCTR=0
EPwm2Regs.AQCTLA.bit.CAU = AQ_CLEAR;                                     //Salida en
    bajo cuando se da un evento, es decir, salida en bajo cuando TBCTR=CMPA

EPwm2Regs.TBCTL.bit.CTRMODE = 3;                                         // Congela
    cuenta
EDIS;
}
//
// Configura ventana de adquisicion y disparo del ADC
//
void initialADCA(Uint16 channel){
    Uint16 acqps=77;                                                       //Variable
    para ventana de adquisicion
/*****
 * Determina la duracion de la ventana de adquisicion
 (tiempo de carga del capacitor de muestreo)
 basandose en la resolucion (12 bits). Para el circuito de
 entrada utilizado y esta
 tarjeta t=411.4957ns,aqps=20-1 ciclos de reloj
 *****/
    EALLOW;
    AdcaRegs.ADCSOCCTL.bit.CHSEL = channel;                               //SOC0
    convierte pin especificado por channel

```

```

AdcaRegs.ADCSOCOCTL.bit.ACQPS = acqps; //Ventana
    de muestreo de 16 ciclos de reloj
AdcaRegs.ADCSOCOCTL.bit.TRIGSEL = 7; //Disparo
    de conversion mediante EPWM2
EDIS;
}

void ADCAOInt(void){

    EALLOW;
    PieVectTable.ADCA1_INT = &adca1_isr; //Liga
        subrutina de interrupcion del ADC_A a la tabla de interrupciones

    PieCtrlRegs.PIEIER1.bit.INTx1 = 1; //Habilita
        la interrupcion del PIE INT1.1
    AdcaRegs.ADCINTSEL1N2.bit.INT1SEL = 0; //EOCO
        dispara interrupcion del ADC_A
    AdcaRegs.ADCINTSEL1N2.bit.INT1E = 1; //Habilita
        INT1 del ADC_A
    AdcaRegs.ADCINTFLGCLR.bit.ADCINT1 = 1; //Limpia
        bandera de interrupcion

    IER |= M_INT1; //Habilita
        interrupciones del grupo 1, donde se encuentra la INT1 del ADC_A

    EDIS;

}
//Inicia operacion del ADCA
void startADCA(){
    EPwm2Regs.ETSEL.bit.SOCAEN = 1; //Habilita pulso de
        SOC en ADCA
    EPwm2Regs.TBCTL.bit.CTRMODE = 0; //Descongela cuenta de
        TB, conteo ascendente
}

float energyFrame(float *x){ //Funcion para
    calcular la energia de una ventana
    int j;
    float E=0.0;
    float N;

    N = (float)BUFFER_SIZE;

    for(j=0;j<BUFFER_SIZE;j++){ //Calcula energia de
        la ventana de datos
        E=x[j]*x[j]+E;
    }
}

```

```

    E = E/N; //Calcula energia
    promedio

    return E;
}

interrupt void adca1_isr(void){

    resultADC = (float) AdcaResultRegs.ADCRESULT0; //Lee valores
    resultados de la conversion analogico-digital
    x[bufferIndex++] = (resultADC*3/4096)-1.6; //Convierte valores
    del ADC a valores de voltaje representado en fp

    if(BUFFER_SIZE <= bufferIndex) //Revisa si se lleno
    el buffer o ventana, activa bandera si se llena
    {
        bufferIndex = 0;
        bufferFull = 1;
    }

    AdcaRegs.ADCINTFLGCLR.bit.ADCINT1 = 1; //Limpia bandera de
    interrupcion

    //
    // Revisa si ocurrio sobreflujo
    //
    if(1 == AdcaRegs.ADCINTOVF.bit.ADCINT1)
    {
        AdcaRegs.ADCINTOVFCLR.bit.ADCINT1 = 1; //Limpia bandera de
        sobreflujo de INT1
    }

    PieCtrlRegs.PIEACK.bit.ACK1 = 1; //Limpia acknowledge
    de interrupcion, permite que se propaguen mas interrupciones
}

```

Bibliografía

- [1] Norman S. Nise. *Sistemas de control para ingeniería*. CECSA (Compañía Editorial Continental), Distrito Federal, México, 2006.
- [2] Magnetic Innovations. What is a direct current motor?, 2019.
- [3] Larry Escobar Salguero. *Arquitecturas de DSP TMS320F28xxx y aplicaciones*. Facultad de Ingeniería, CU CDMX, México, 2014.
- [4] Real Academia Española. *Diccionario de la lengua española (23a edición)*. S.L.U. ESPASA LIBROS, 2014.
- [5] Josh Lauer. Surveillance history and the history of new media: An evidential paradigm . *SAGE journals*, 14(4):566–582, 2011.
- [6] Louis E. Freznel Jr. *Principles of electronic communication systems*. Mc Graw Hill Education, Penn Plaza, New York, 2016.
- [7] Lubos Ovsenik, Anna Kolesárová, and Ján Turán. Video surveillance systems. *Acta Electrotechnica et Informatica*, 10:46–53, 01 2010.
- [8] Julio Muñoz. *Sistemas de seguridad*. Paraninfo, Madrid, 2000.
- [9] George W. Younkin. *Industrial Servo Control Systems*. Taylor and Francis Inc., Bosa Roca, US, 2002.
- [10] Stephen M.Tobin. *DC Servos. Application and design with Matlab*. Taylor and Francis Group, Boca Raton, US, 2017.
- [11] H.D Young and Roger A. Freedman. *Física universitaria, volumen 1*. Pearson educación, Estado de México, Mexico, 2009.

- [12] Texas Instruments. *TMS320F2837xS Microcontrollers. Technical Reference Manual SPRUH5G*. Texas Instruments, US, 2014. Rev. 2019.
- [13] Muhammad H. Rashid. *Microelectronic Circuits. Analysis and design*. Cengage Learning, West Florida,US, 2011.
- [14] Daniel W. Hart. *Power Electronics*. McGraw Hill, NY,US, 2011.
- [15] Dennis G. Zill and Warren s. Wright. *Ecuaciones diferenciales con problemas con valores en la frontera (4va Edición)*. Cengage Learning, Inc., Santa Fe, DF, México, 2015.
- [16] Benjamin C. Kuo. *Sistemas de control automático*. Prentice Hall Hispanoamérica, Estado de México, Mexico, 1996.
- [17] Austin Hughes. *Electric motors and drives. Fundamentals, types and applications*. Elsevier, Oxford, England, 2006.
- [18] Dr.Guy Drijkoningen. Introduction to reflection seismics. appendix c: Derivation of the 1d wave equation. *TU Delft OpenCourseWare*, 02 2016.
- [19] J. Marsden and A. Tromba. *Vector Calculus*. Macmillan Learning, Madison Avenue, NY, USA, 2012.
- [20] R. Benintendi. *Process Safety Calculations*. Elsevier Science, Cambridge, MA, USA, 2017.
- [21] Lawrence E. Kinsler. *Fundamentals of acoustics*. John Wiley and sons, Inc, NY, US, 2000.
- [22] Preeti Rao. *Audio signal processing. Chapter in Speech, Audio, Image and Biomedical Signal Processing using Neural Network*. Springer-Verlag, Bombay, India, 2007.
- [23] Prof. Ilya Pollak. Ece 438 digital signal processing with applications. lecture notes, section 2.2: Speech processing. *Purdue University. School of Electrical and Computer Engineering*, 08 2004.
- [24] X. Yang, B. Tan, J. Ding, J. Zhang, and J. Gong. Comparative study on voice activity detection algorithm. pages 599–602, June 2010.

- [25] B. Sklar. *Digital communications: fundamentals and applications*. Prentice-Hall PTR, Upper Saddle River, NJ, USA, 2001.
- [26] John G. Proakis and Dimitris K. Manolakis. *Digital Signal Processing (4th Edition)*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 2006.
- [27] J. Watkinson. *An Introduction to Digital Audio*. Focal Press, Jordan Field, Oxford, UK, 2002.
- [28] Lawrence R. Rabiner and Ronald W. Schafer. *Introduction to Digital Speech Processing*. Now Publishers Inc., Hanover, MA, USA, 2007.
- [29] M. Jalil, F. A. Butt, and A. Malik. Short-time energy, magnitude, zero crossing rate and autocorrelation measurement for discriminating voiced and unvoiced segments of speech signals. pages 208–212, May 2013.
- [30] Sakhnov Kirill, Verteletskaya Ekaterina, and B. Simak. Approach for energy-based voice detector with adaptive scaling factor. *IAENG International Journal of Computer Science*, 36, 11 2009.
- [31] Seshashyama Sameeraj Meduri and Rufus Ananth. A survey and evaluation of voice activity detection algorithms. Master's thesis, Blekinge Tekniska Högskola, 2011.
- [32] Texas Instruments. *TMS320F2837xS Microcontrollers. DataSheet SPRS881H*. Texas Instruments, US, 2014. Rev. 2020.
- [33] Texas Instruments. *LAUNCHXL-F28377S overview. User's Guide SPRUI25D*. Texas Instruments, US, 2015. Rev. 2019.
- [34] Kwan Chi Kao, editor. *5 - Electrets*. Academic Press, San Diego, 2004.
- [35] John Caldwell. *Single-Supply, Electret Microphone Pre Amplifier Reference Design*. Texas Instruments, US, 2015.
- [36] Larry Escobar Salguero et al. *Manual de algoritmos y aplicaciones de procesamiento digital de señales empleando la familia TMS320F2837xS*. Facultad de Ingeniería, CU CDMX, México, 2019.

- [37] Hitec. *Announced specification of HS-422 Standar Deluxe Servo*. Hitec, 2014.
- [38] Steren. *CCD CCTV color camera*. Steren, 2007.
- [39] D. K. Freeman, G. Cosier, C. B. Southcott, and I. Boyd. The voice activity detector for the pan-european digital cellular mobile telephone service. pages 369–372 vol.1, 1989.
- [40] F. Beritelli, S. Casale, and A. Cavallaero. A robust voice activity detector for wireless communications using soft computing. *IEEE Journal on Selected Areas in Communications*, 16(9):1818–1829, 1998.
- [41] Simon Graf, Tobias Herbig, Markus Buck, and Gerhard Schmidt. Features for voice activity detection: a comparative analysis. *EURASIP Journal on Advances in Signal Processing*, 2015:91, 11 2015.
- [42] David Dean, Sridha Sridharan, Robert Vogt, and Michael Mason. The qut-noise-timit corpus for evaluation of voice activity detection algorithms. pages 1–4, 2010.
- [43] Renesas Technology Corp. *HD74LS32 Quadruple 2-Input Positive OR Gate Rev. 2*. Renesas Technology Corp, Japan, 2005.
- [44] Larry Escobar Salguero. *Conceptos básicos de procesamiento digital de señales*. Facultad de Ingeniería, CU CDMX, México, 2009.